In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
df = pd.read_csv("Titanic-Dataset.csv")
```

In [3]:
```
df.columns
```

Out[3]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [4]:
```
df.head()
```

Out[4]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/ O2. 3101282 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.0 | 0 | 0 | 373450 |

In [5]:
```
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
```

In [6]:
```
df.describe()
```

Out[6]:

| | **PassengerId** | **Survived** | **Pclass** | **Age** | **SibSp** | **Parch** |
|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 |

In [7]:
```
df.drop("Cabin", axis=1, inplace=True)
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Embarked'],
      dtype='object')
```

In [9]:
```
df['Age'] = df['Age'].fillna(df['Age'].mean())
```

In [10]:
```
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
```

In [11]:
```python
df.dropna(inplace=True)
```

In [12]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 889 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  889 non-null    int64  
 1   Survived     889 non-null    int64  
 2   Pclass       889 non-null    int64  
 3   Name         889 non-null    object 
 4   Sex          889 non-null    object 
 5   Age          889 non-null    float64
 6   SibSp        889 non-null    int64  
 7   Parch        889 non-null    int64  
 8   Ticket       889 non-null    object 
 9   Fare         889 non-null    float64
 10  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 83.3+ KB
```

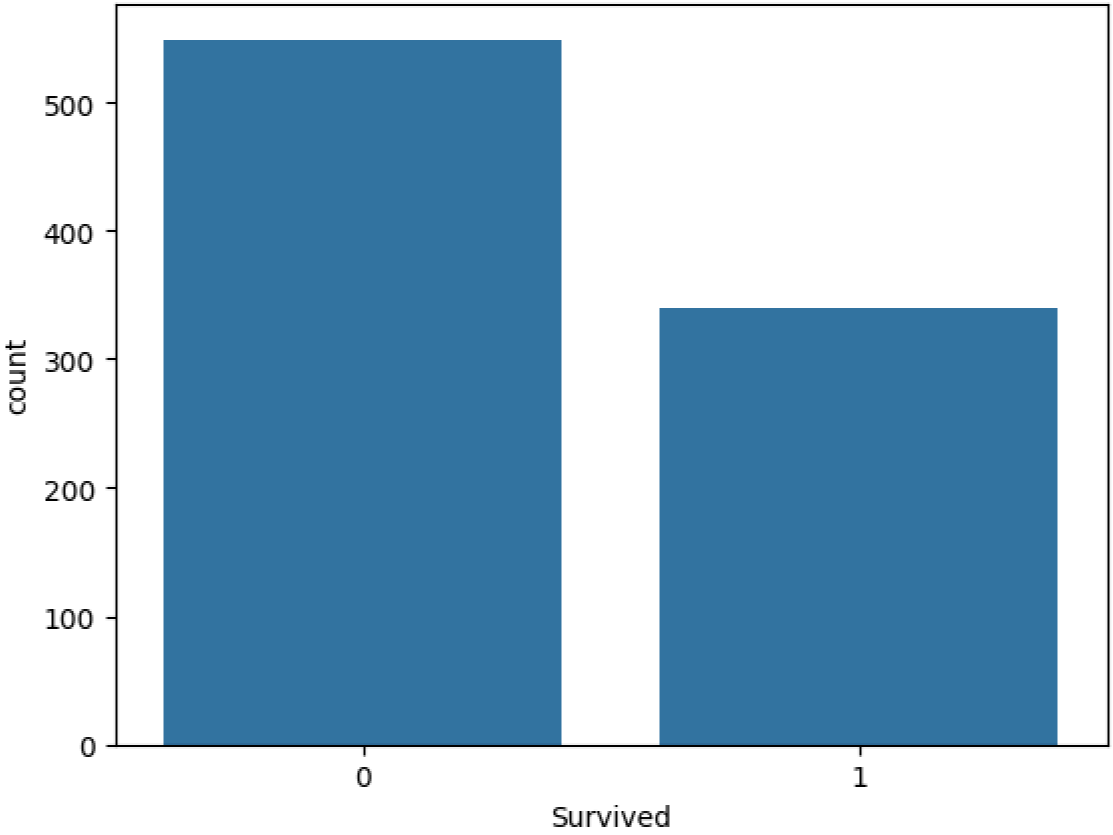
In [13]:
```python
df.columns
```

Out[13]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Embarked'],
      dtype='object')
```

In [14]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
```

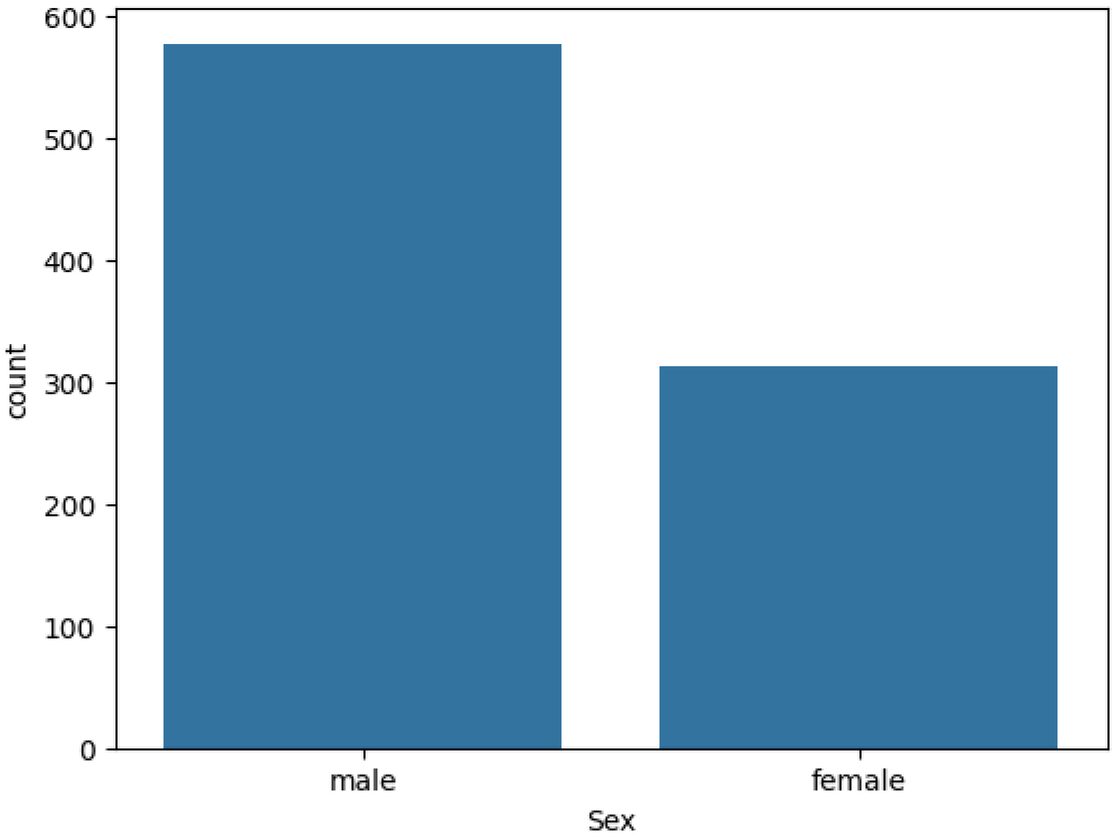
In [15]:
```python
sns.countplot(x='Survived',data=df)
```

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



```
In [16]: sns.countplot(x='Sex',data=df)
```

```
Out[16]: <Axes: xlabel='Sex', ylabel='count'>
```



In [17]:

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Sex'] = le.fit_transform(df['Sex'])
```

In [18]:

```
df.head()
```

Out[18]:

|  | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | 1 | 22.0 | 1 | 0 | A/5 21171 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 0 | 38.0 | 1 | 0 | PC 17599 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | 0 | 26.0 | 0 | 0 | STON/O2. 3101282 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | 0 | 35.0 | 1 | 0 | 113803 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | 1 | 35.0 | 0 | 0 | 373450 |

In [19]:

```
int_df = df.select_dtypes(include='int')

correlation = int_df.corr()

print(correlation)
```

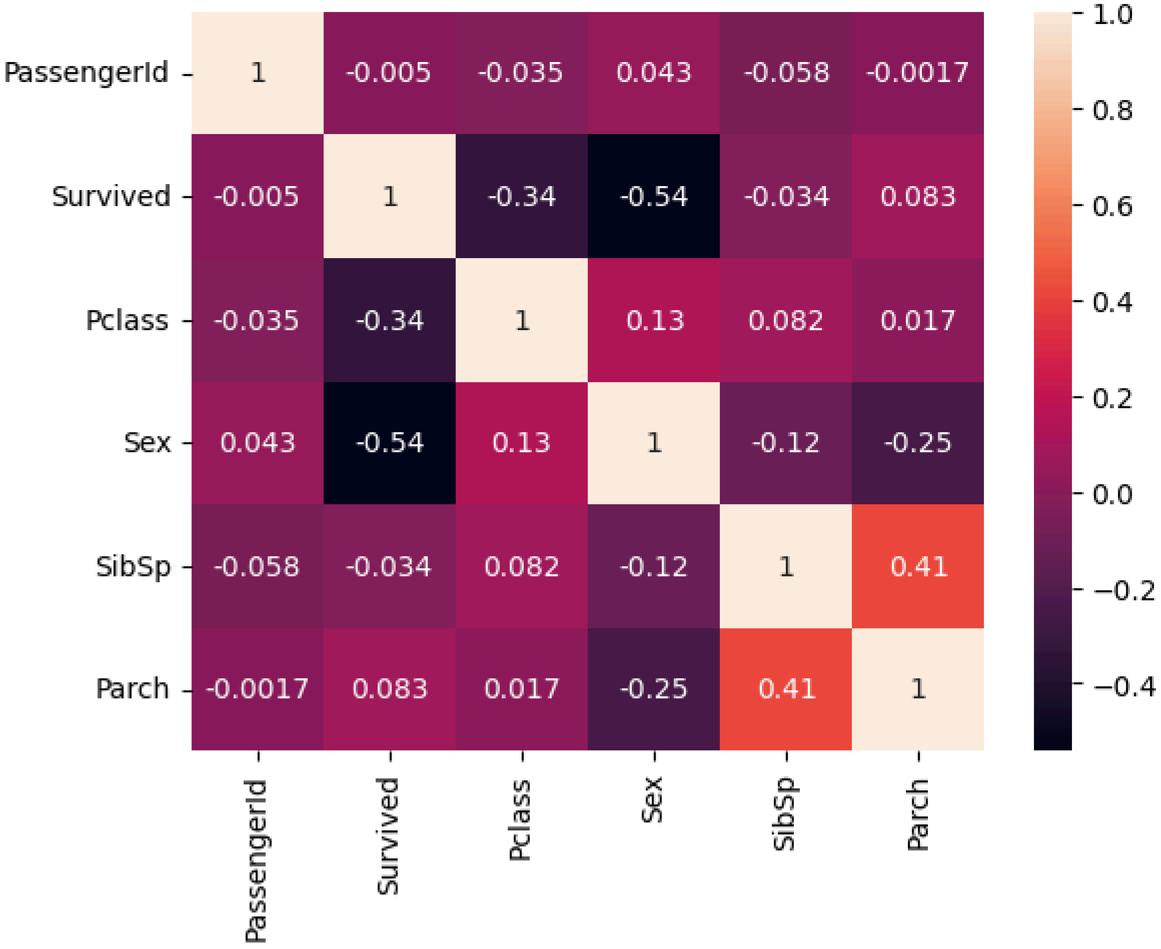
```
             PassengerId  Survived    Pclass       Sex     SibSp     Parch
PassengerId     1.000000 -0.005028 -0.035330  0.043136 -0.057686 -0.001657
Survived       -0.005028  1.000000 -0.335549 -0.541585 -0.034040  0.083151
Pclass         -0.035330 -0.335549  1.000000  0.127741  0.081656  0.016824
Sex             0.043136 -0.541585  0.127741  1.000000 -0.116348 -0.247508
SibSp          -0.057686 -0.034040  0.081656 -0.116348  1.000000  0.414542
Parch          -0.001657  0.083151  0.016824 -0.247508  0.414542  1.000000
```

In [20]:

```
sns.heatmap(correlation,annot=True)
```

Out[20]:

```
<Axes: >
```


PassengerId
Survived
Pclass
Sex
SibSp
Parch
1 -0.005 -0.035 0.043 -0.058 -0.0017
-0.005 1 -0.34 -0.54 -0.034 0.083
-0.035 -0.34 1 0.13 0.082 0.017
0.043 -0.54 0.13 1 -0.12 -0.25
-0.058 -0.034 0.082 -0.12 1 0.41
-0.0017 0.083 0.017 -0.25 0.41 1
PassengerId
Survived
Pclass
Sex
SibSp
Parch
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

```
In [21]: df.head()
```

Out[21]:

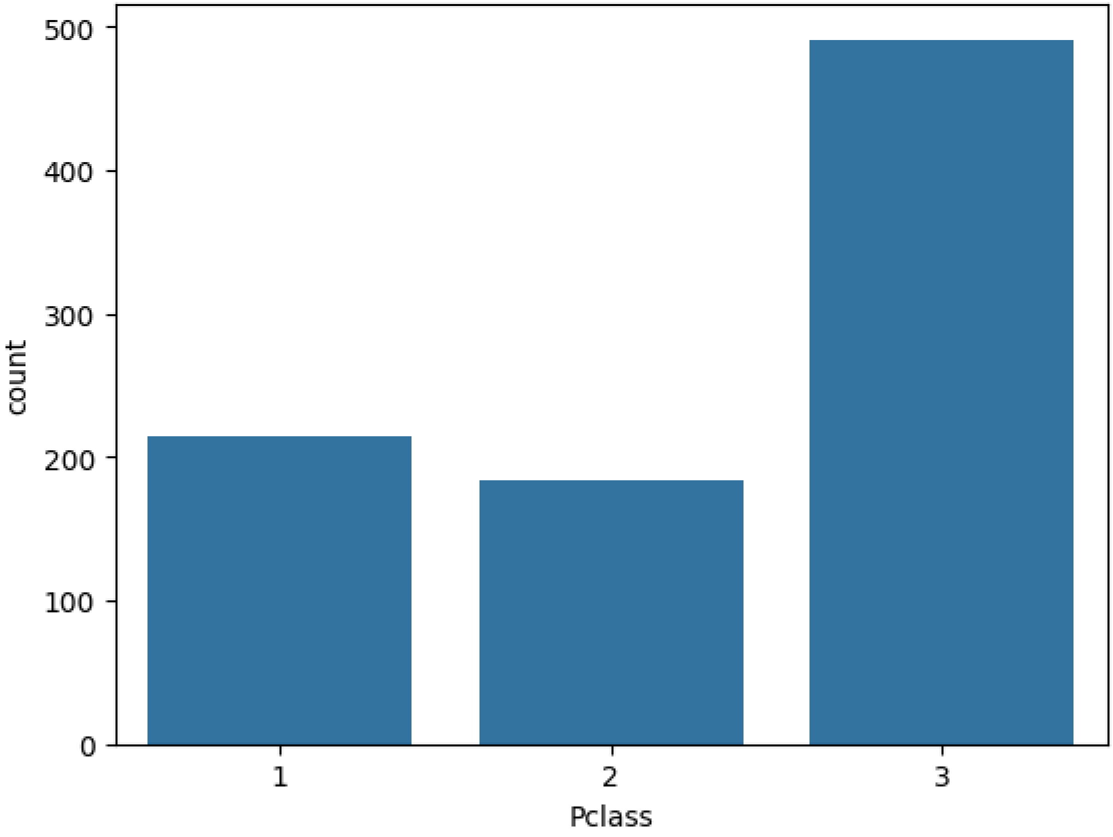
| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | 1 | 22.0 | 1 | 0 | A/5 21171 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 0 | 38.0 | 1 | 0 | PC 17599 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | 0 | 26.0 | 0 | 0 | STON/O2. 3101282 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | 0 | 35.0 | 1 | 0 | 113803 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | 1 | 35.0 | 0 | 0 | 373450 |

In [22]:

```
sns.countplot(x='Pclass',data=df)
```

Out[22]:

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



```
In [23]: df = pd.get_dummies(df, columns=['Embarked'], drop_first=True)
```

```
In [24]: df.head()
```

Out[24]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | 1 | 22.0 | 1 | 0 | A/5 21171 |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 0 | 38.0 | 1 | 0 | PC 17599 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | 0 | 26.0 | 0 | 0 | STON/O2. 3101282 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | 0 | 35.0 | 1 | 0 | 113803 |
| **4** | 5 | 0 | 3 | Allen, Mr. William Henry | 1 | 35.0 | 0 | 0 | 373450 |

In [25]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

In [26]:

```
df['Age'] = scaler.fit_transform(df[['Age']])
df['Pclass'] = scaler.fit_transform(df[['Pclass']])
```

In [27]:

```
df.head()
```

Out[27]:

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0.825209 | Braund, Mr. Owen Harris | 1 | -0.590495 | 1 | 0 | |
| **1** | 2 | 1 | -1.572211 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 0 | 0.643971 | 1 | 0 | |
| **2** | 3 | 1 | 0.825209 | Heikkinen, Miss. Laina | 0 | -0.281878 | 0 | 0 | 31 |
| **3** | 4 | 1 | -1.572211 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | 0 | 0.412509 | 1 | 0 | 1 |
| **4** | 5 | 0 | 0.825209 | Allen, Mr. William Henry | 1 | 0.412509 | 0 | 0 | 3 |

In [28]:

```
df.drop(['PassengerId', 'Ticket','Name','Fare'], axis=1, inplace=True)
```

In [29]:

```
df['Embarked_Q'] = df['Embarked_Q'].astype(int)
df['Embarked_S'] = df['Embarked_S'].astype(int)
```

In [30]:

```
df
```

Out[30]:

|  | Survived | Pclass | Sex | Age | SibSp | Parch | Embarked_Q | Embarked_ |
|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0.825209 | 1 | -0.590495 | 1 | 0 | 0 |  |
| **1** | 1 | -1.572211 | 0 | 0.643971 | 1 | 0 | 0 |  |
| **2** | 1 | 0.825209 | 0 | -0.281878 | 0 | 0 | 0 |  |
| **3** | 1 | -1.572211 | 0 | 0.412509 | 1 | 0 | 0 |  |
| **4** | 0 | 0.825209 | 1 | 0.412509 | 0 | 0 | 0 |  |
| **...** | ... | ... | ... | ... | ... | ... | ... |  |
| **886** | 0 | -0.373501 | 1 | -0.204724 | 0 | 0 | 0 |  |
| **887** | 1 | -1.572211 | 0 | -0.821957 | 0 | 0 | 0 |  |
| **888** | 0 | 0.825209 | 0 | 0.003524 | 1 | 2 | 0 |  |
| **889** | 1 | -1.572211 | 1 | -0.281878 | 0 | 0 | 0 |  |
| **890** | 0 | 0.825209 | 1 | 0.181046 | 0 | 0 | 1 |  |

889 rows × 8 columns

In [31]:

```
X=df.drop(columns=['Survived'],axis=1)
```

In [32]:

```
Y= df['Survived']
```

In [33]:

```
from sklearn.model_selection import train_test_split
```

In [34]:

```
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=
```

In [35]:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [36]:

```
model.fit(X_train,Y_train)
```

Out[36]:

▾ LogisticRegression ⓘ ?

Parameters

| | |
|---|---|
| penalty | 'l2' |
| dual | False |
| tol | 0.0001 |
| C | 1.0 |
| fit_intercept | True |
| intercept_scaling | 1 |
| class_weight | None |
| random_state | None |
| solver | 'lbfgs' |
| max_iter | 100 |
| multi_class | 'deprecated' |
| verbose | 0 |
| warm_start | False |
| n_jobs | None |
| l1_ratio | None |

In [37]:

```
X_train_prediction=model.predict(X_train)
```

In [38]:

```
print(X_train_prediction)
```

```
[0 1 1 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 0 0
 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 0 0
 1 0 1 0 0 1 0 0 1 0 0 1 1 0 1 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1
 1 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1
 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0
 1 0 0 0 1 0 1 0 0 0 1 1 1 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 0 0
 0 1 1 0 1 1 0 0 0 1 0 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 0
 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1
 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0
 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1
 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0
 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 1 1
 1 0 0 1 0 0 0 1 0 1 1 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 1 0 1 0 1 0 1 0
 0 1 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 0 0 0 0
 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 0
 1 1 1 0 0 1 1 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 1 0 1 1 0 0 1 0 0 1 0 0 0
 1 0 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0
 0 0 1 1 1 0 0 0]
```

```
In [39]: from sklearn.metrics import accuracy_score
         train_data_accuracy=accuracy_score(Y_train,X_train_prediction)
```

```
In [40]: train_data_accuracy
```

```
Out[40]: 0.8002812939521801
```

```
In [42]: print(f"Accuracy of Model: {int(train_data_accuracy*100)}%")
```

```
Accuracy of Model: 80%
```

```
In [ ]:
```