

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

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

```
In [3]: df.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

```
In [4]: 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 [5]: `df.describe()`

Out[5]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [6]: `df.drop(columns=['PassengerId', 'Name', 'Ticket', 'Cabin'], inplace=True)`

In [7]: `df.head()`

Out[7]:

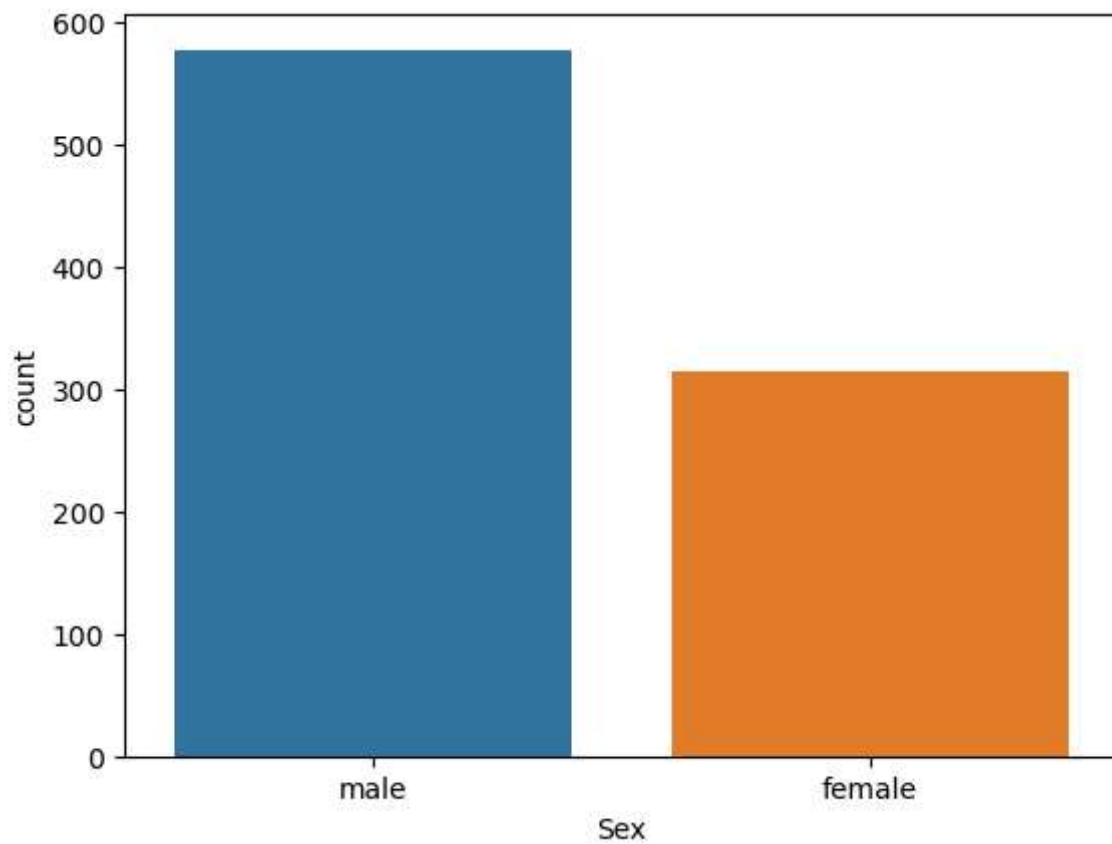
	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

```
In [8]: df['Survived'].value_counts()
```

```
Out[8]: Survived  
0      549  
1      342  
Name: count, dtype: int64
```

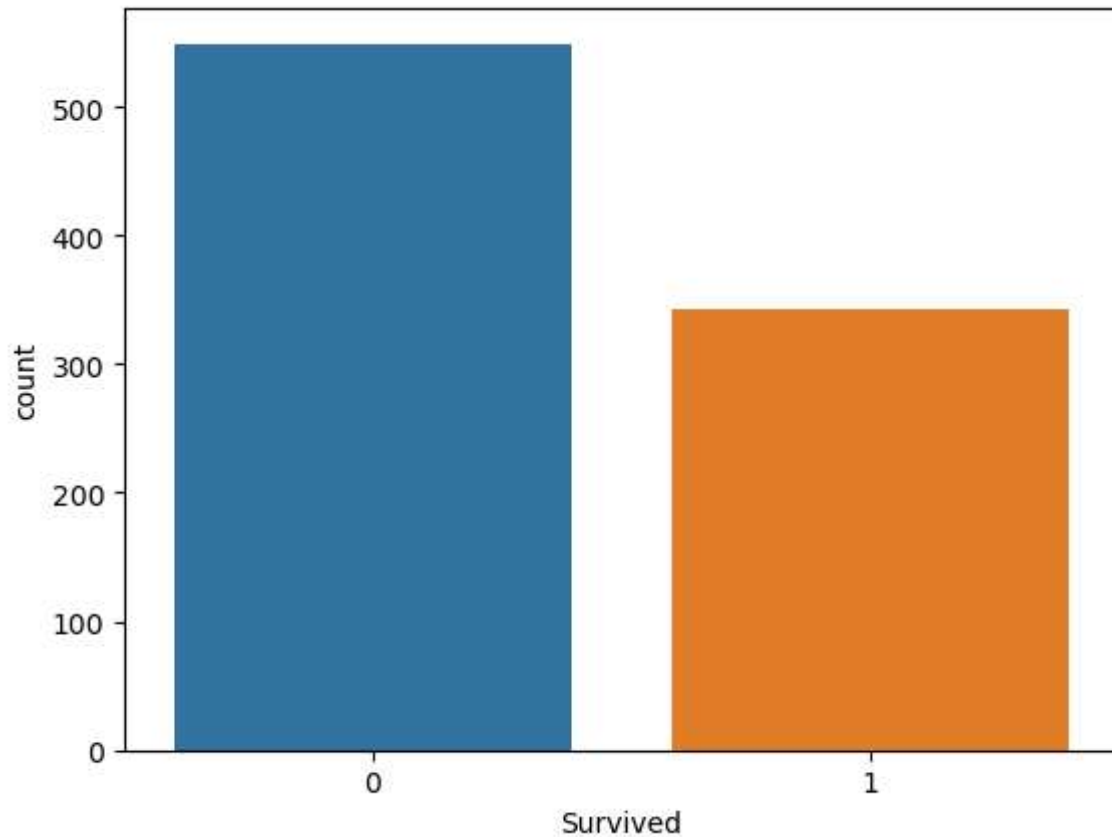
```
In [11]: import seaborn as sns  
sns.countplot(x=df['Sex'])
```

```
Out[11]: <Axes: xlabel='Sex', ylabel='count'>
```



```
In [12]: sns.countplot(x=df['Survived'])
```

```
Out[12]: <Axes: xlabel='Survived', ylabel='count'>
```



```
In [13]: import sklearn
from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()
df['Sex']= le.fit_transform(df['Sex'])
```

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

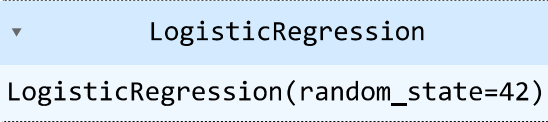
```
Out[14]:
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	1	22.0	1	0	7.2500	S
1	1	1	0	38.0	1	0	71.2833	C
2	1	3	0	26.0	0	0	7.9250	S
3	1	1	0	35.0	1	0	53.1000	S
4	0	3	1	35.0	0	0	8.0500	S

```
In [15]: x=df[['Sex', 'Pclass']]
y=df['Survived']
```

```
In [16]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(x,y, test_size=0.2, random_state
```

```
In [17]: from sklearn.linear_model import LogisticRegression
log= LogisticRegression(random_state = 42)
log.fit(X_train, y_train)
```

Out[17]: 

In [18]:

```
pred = log.predict(X_test)
pred
```

Out[18]:

```
array([0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
        1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
        1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1,
        0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
        0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
        1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0,
        0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
        0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
        1, 1, 1], dtype=int64)
```

In [19]:

```
print(y_test)
```

```
709    1
439    0
840    0
720    1
39     1
..
433    0
773    0
25     1
84     1
10     1
Name: Survived, Length: 179, dtype: int64
```

In [20]:

```
from sklearn.metrics import accuracy_score, precision_score , recall_score , f1_score
accuracy = accuracy_score(y_test, pred)
print("Accuracy:", accuracy)
pre = precision_score(y_test , pred , average = "micro")
print("The precision is:{}".format(pre))

rec = recall_score(y_test , pred , average = "micro")
print("The recall is:{}".format(rec))

f1 = f1_score(y_test , pred , average = "micro")
print("The f1_score is:{}".format(f1))
```

```
Accuracy: 0.7821229050279329
The precision is:0.7821229050279329
The recall is:0.7821229050279329
The f1_score is:0.7821229050279329
```

In [21]:

```
import warnings
warnings.filterwarnings("ignore")
res= log.predict([[0,0]])

if(res==0):
    print("Not survived")
```

```
else:  
    print("survived")
```

survived

```
In [22]: import warnings  
warnings.filterwarnings("ignore")  
res= log.predict([[2,1]])  
  
if(res==0):  
    print("Not survived")  
else:  
    print("survived")
```

Not survived

In []: