In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

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

Out[2]:

	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
...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

891 rows × 12 columns

In [3]:
```
data.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 [4]:
```
data.shape
```

Out[4]:
```
(891, 12)
```

In [5]:
```
data.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]:
```
data.isnull().sum()
```

Out[6]:
```
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
```

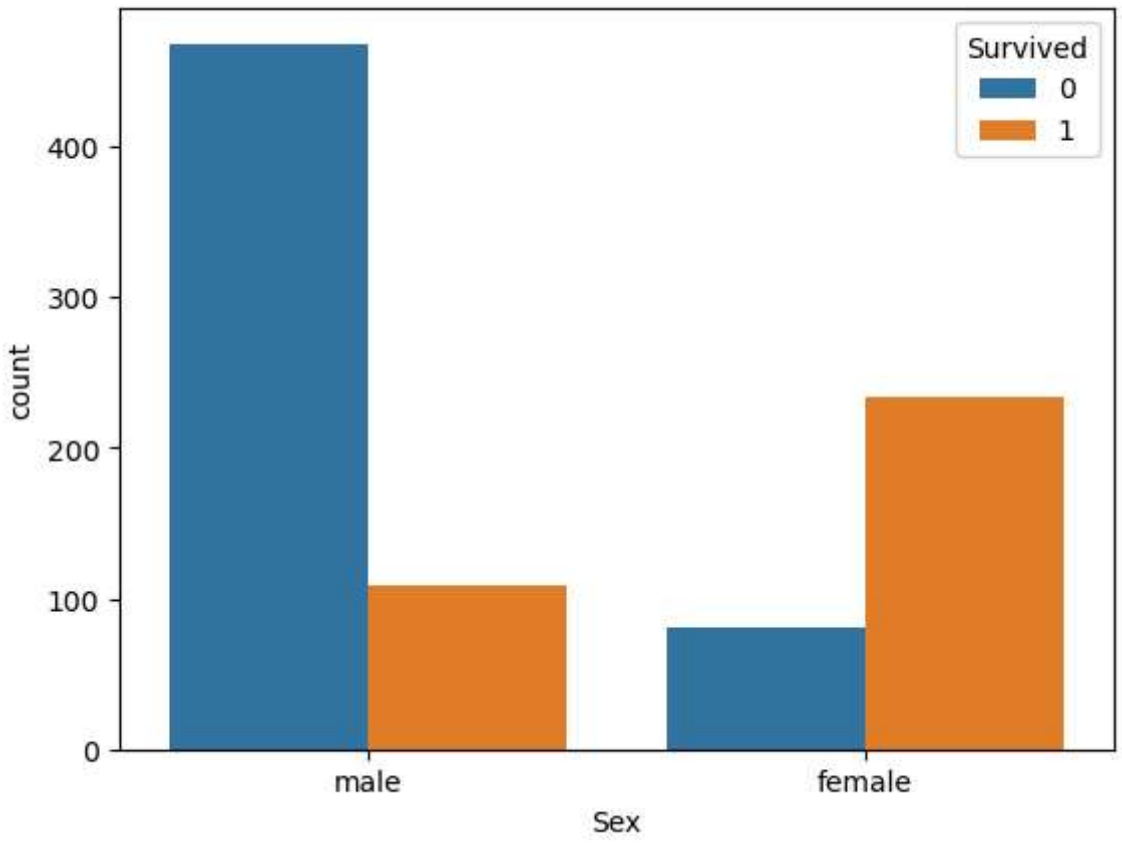
In [7]:
```
data.drop(columns=['Cabin','Name','PassengerId','Ticket'], inplace = True)
```

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

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

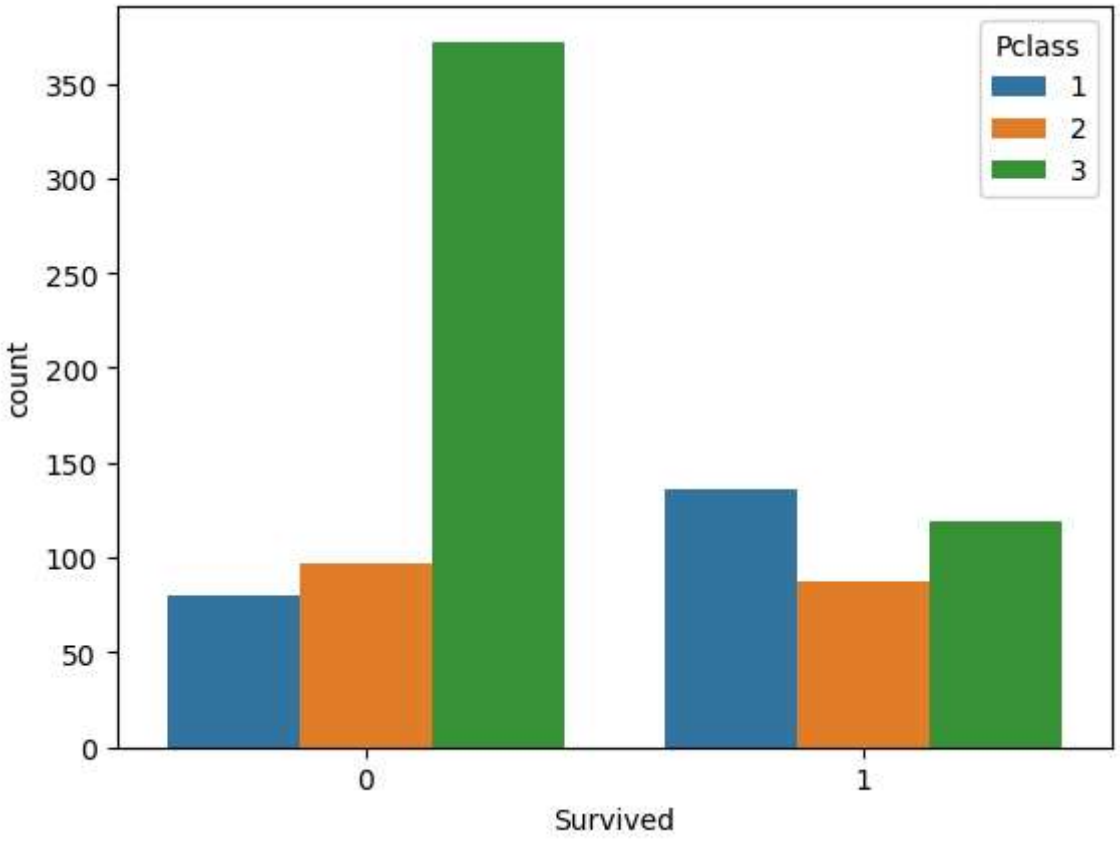
In [9]:
```
sns.countplot(x=data['Sex'], hue= data['Survived'] )
```

Out[9]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



In [10]:
```
sns.countplot(x = data['Survived'], hue = data['Pclass'])
```

Out[10]:
```
<Axes: xlabel='Survived', ylabel='count'>
```



In [11]:
```
import sklearn
from sklearn.preprocessing import LabelEncoder
df = LabelEncoder()
data['Sex'] = df.fit_transform(data['Sex'])
```

In [12]:
```
data.head()
```

Out[12]:

	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 [13]:
```
x = data[['Sex','Pclass']]
Y = data['Survived']
```

In [14]:
```
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, rand
```

In [15]:
```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state=42)
log.fit(x_train, Y_train)
```

Out[15]:
```
LogisticRegression(random_state=42)
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [16]:
```
pred = log.predict(x_test)
pred
```

Out[16]:
```
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 [17]:
```
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 [18]:
```
from sklearn.metrics import accuracy_score , precision_score , recall_score ,
accuracy = accuracy_score(Y_test, pred)
print("Accuracy : ", accuracy)
```

```
Accuracy :  0.7821229050279329
```

In [19]:
```
precision = precision_score(Y_test, pred, average = "micro")
print("precision :" , precision)
```

```
precision : 0.7821229050279329
```

In [20]:
```
recall = recall_score(Y_test, pred, average = "micro")
print("Recall : ", recall)
```

```
Recall :  0.7821229050279329
```

In [21]:
```
fscore = f1_score(Y_test, pred, average = "micro")
print("F1_Score " ,fscore)
```

```
F1_Score  0.7821229050279329
```

In [22]:
```
import warnings
warnings.filterwarnings("ignore")
res= log.predict([[0,0]])

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

```
survived
```

In [23]:
```
import warnings
warnings.filterwarnings("ignore")
res= log.predict([[2,1]])

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

```
Not survived
```

In [ ]:
```

```