In [36]:

```
import numpy as np
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
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
```

In [2]:

```
#importing dataset with pd
titanic_data = pd.read_csv("D:\study\datasets\Titanic-Dataset.csv")
```

In [3]:

```
# first five rows
titanic_data.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [4]:

```
# last five rows
titanic_data.tail()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [5]:

```
titanic_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 [7]:

```
# number of rows and columns
titanic_data.shape
```

Out[7]:

```
(891, 12)
```

In [8]:

```
# counting number of null values
titanic_data.isnull().sum()
```

Out[8]:

```
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 [9]:

```
# describing the table
titanic_data.describe()
```

Out[9]:

	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 [11]:

```
# number of people who survived
titanic_data['Survived'].value_counts()
```

Out[11]:

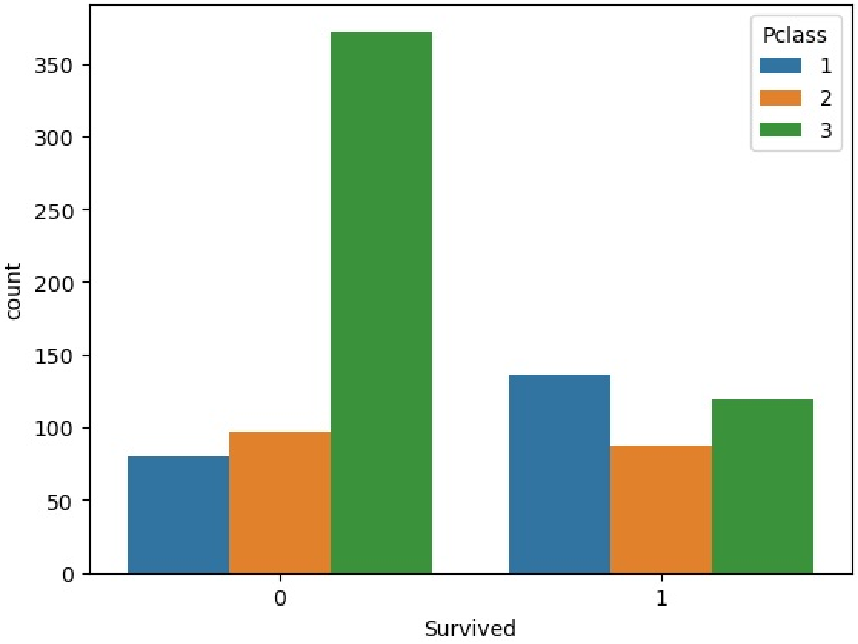
```
0    549
1    342
Name: Survived, dtype: int64
```

In [12]:

```
# visualising the nnumber of poeple who survived with countplot
sns.countplot(x = titanic_data['Survived'], hue = titanic_data['Pclass'])
```

Out[12]:

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

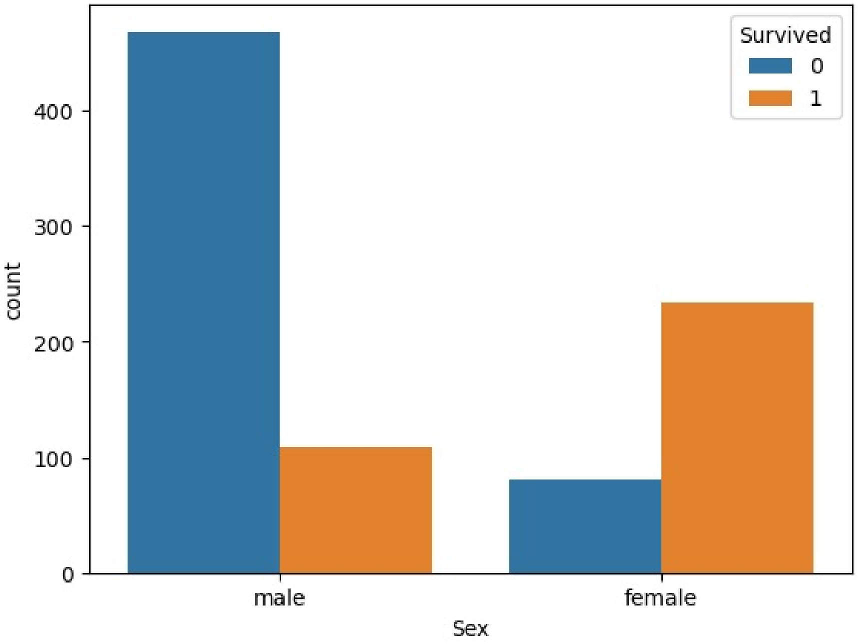


In [13]:

```
# visualising the number of people survived with respect to gender
sns.countplot(x = titanic_data['Sex'], hue = titanic_data['Survived'])
```

Out[13]:

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

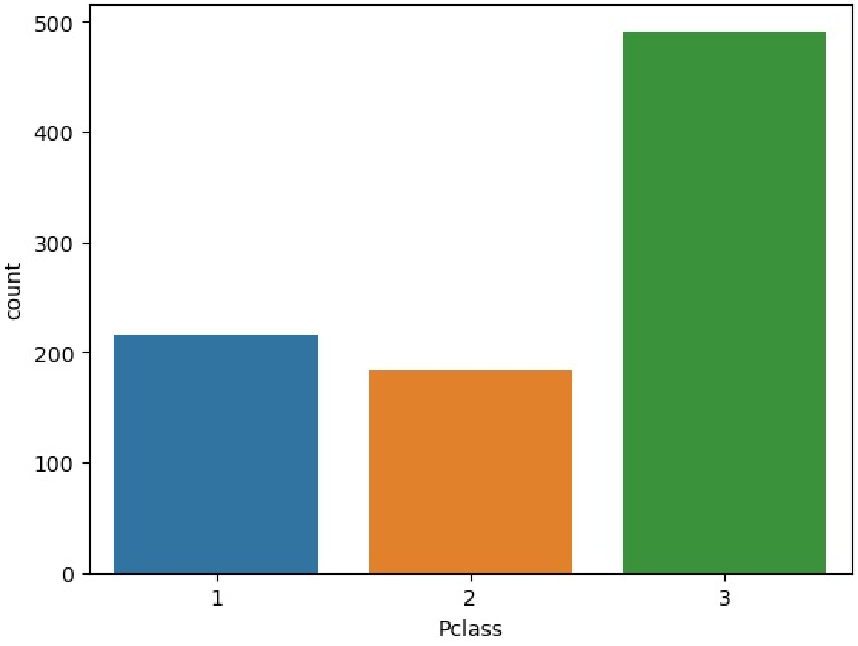


In [16]:

```
# visualising Pclass data with countplot
sns.countplot(x = 'Pclass', data = titanic_data)
```

Out[16]:

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



In [17]:

```
titanic_data['Sex'].value_counts()
```

Out[17]:

```
male      577
female    314
Name: Sex, dtype: int64
```

In [18]:

```
titanic_data['Embarked'].value_counts()
```

Out[18]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [19]:

```
# converting categorical Columns

titanic_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}}, inplace=True)
```

In [20]:

```python
titanic_data.head()
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	0.0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	C85	1.0
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	0.0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	C123	0.0
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	NaN	0.0

In [21]:

```python
x = titanic_data.drop(columns = ['PassengerId','Name','Survived','Ticket'], axis=1)
y = titanic_data['Survived']
```

In [22]:

```python
print(x)
```

```
     Pclass  Sex   Age  SibSp  Parch     Fare Cabin  Embarked
0         3    0  22.0      1      0   7.2500   NaN       0.0
1         1    1  38.0      1      0  71.2833   C85       1.0
2         3    1  26.0      0      0   7.9250   NaN       0.0
3         1    1  35.0      1      0  53.1000  C123       0.0
4         3    0  35.0      0      0   8.0500   NaN       0.0
..      ...  ...   ...    ...    ...      ...   ...       ...
886       2    0  27.0      0      0  13.0000   NaN       0.0
887       1    1  19.0      0      0  30.0000   B42       0.0
888       3    1   NaN      1      2  23.4500   NaN       0.0
889       1    0  26.0      0      0  30.0000  C148       1.0
890       3    0  32.0      0      0   7.7500   NaN       2.0

[891 rows x 8 columns]
```

In [23]:

```python
print(y)
```

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

Splitting data into train and test data

In [41]:

```python
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 2)
```

In [42]:

```python
print(x.shape)
```

```
(891, 8)
```

In [43]:

```python
print(x_train.shape)
```

```
(712, 8)
```

In [44]:

```python
print(x_test.shape)
```

```
(179, 8)
```

Model training

In [70]:

```python
print(x_train.isnull().sum())  # Check for missing values
x_train = x_train.dropna()
```

```
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Cabin       0
Embarked    0
dtype: int64
```

In [71]:

```python
titanic_data = titanic_data.drop(['Age'], axis =1)
```

In [72]:
```
titanic_data_final = titanic_data
```

In [73]:
```
titanic_data.head()
```

Out[73]:

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	1	0	A/5 21171	7.2500	NaN	0.0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	1	0	PC 17599	71.2833	C85	1.0
2	3	1	3	Heikkinen, Miss. Laina	1	0	0	STON/O2. 3101282	7.9250	NaN	0.0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	1	0	113803	53.1000	C123	0.0
4	5	0	3	Allen, Mr. William Henry	0	0	0	373450	8.0500	NaN	0.0

In [74]:
```
x = titanic_data[['Pclass','Sex']]
y = titanic_data['Survived']
```

In [75]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 2)
```

In [76]:
```
log = LogisticRegression(random_state = 0)
log.fit(x_train, y_train)
```

Out[76]:
```
▾        LogisticRegression
LogisticRegression(random_state=0)
```

Model prediction

In [77]:
```
pred = print(log.predict(x_test))
```

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

In [78]:
```
print(y_test)
```

```
707    1
37     0
615    1
169    0
68     1
      ..
89     0
80     0
846    0
870    0
251    0
Name: Survived, Length: 179, dtype: int64
```

In [79]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [82]:
```
res = log.predict([[2,0]])
if res==0:
    print("Not survived...")
else:
    print("Survived")
```

```
Not survived...
```

In [ ]: