importing dependencies

In [2]:
```
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.metrics import accuracy_score,precision_score
```

Data collection and Processing

In [17]:
```
titanic_data=pd.read_csv("train.csv")
titanic_data.head()
```

Out[17]:

	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 [18]:
```
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 [19]:
```
titanic_data.shape
```

Out[19]:
```
(891, 12)
```

In [20]:
```
titanic_data["Pclass"].value_counts()
```

Out[20]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

In [21]:
```
titanic_data.isnull().sum()
```

Out[21]:
```
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
```

> Handling the missing values

In [22]:
```
titanic_data=titanic_data.drop(columns="Cabin",axis=1)
```

In [26]:
```
titanic_data.describe()
```

Out[26]:

	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 [27]:
```
titanic_data['Age'].fillna(titanic_data['Age'].mean(),inplace=True)
```

In [29]:
```
titanic_data['Embarked'].mode()
```

Out[29]:
```
0    S
Name: Embarked, dtype: object
```

In [38]:
```
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0],inplace=True)
```

In [39]:
```
titanic_data.isnull().sum()
```

Out[39]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

Data Analysis

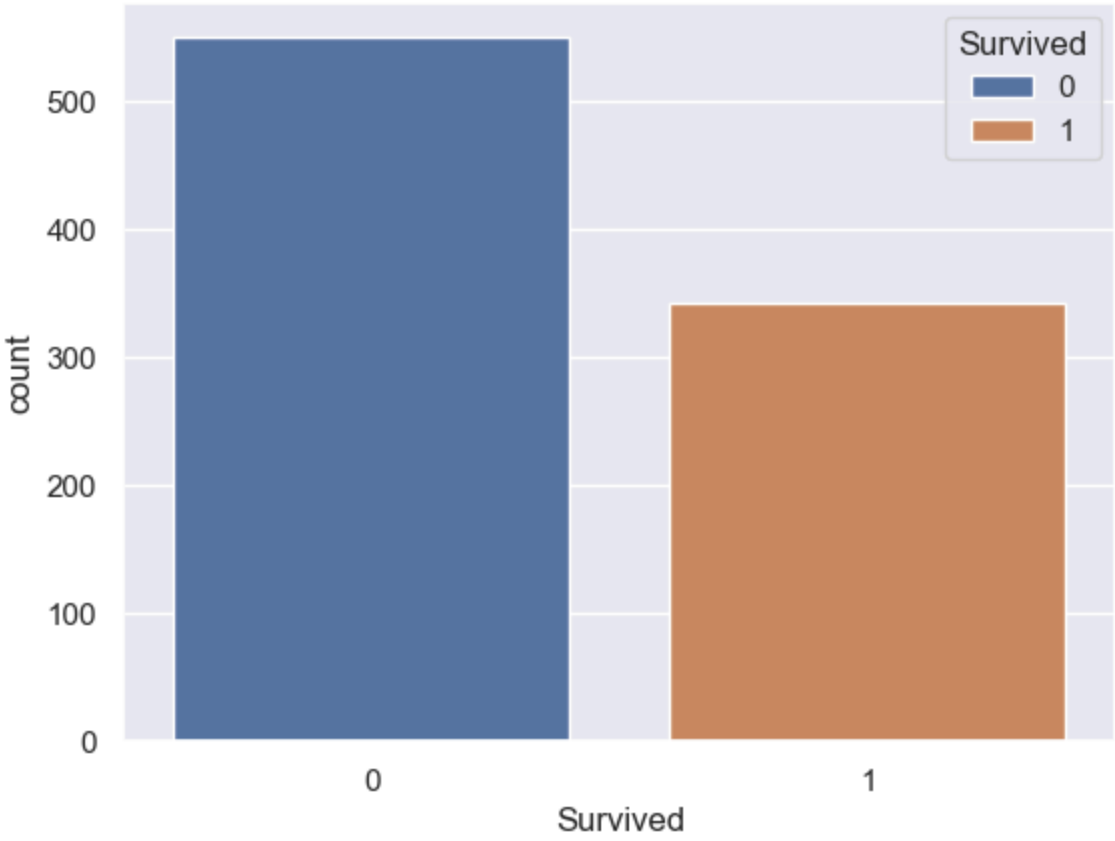
In [43]:
```
titanic_data['Survived'].value_counts()
```

Out[43]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

In [44]:
```
sns.set()
```

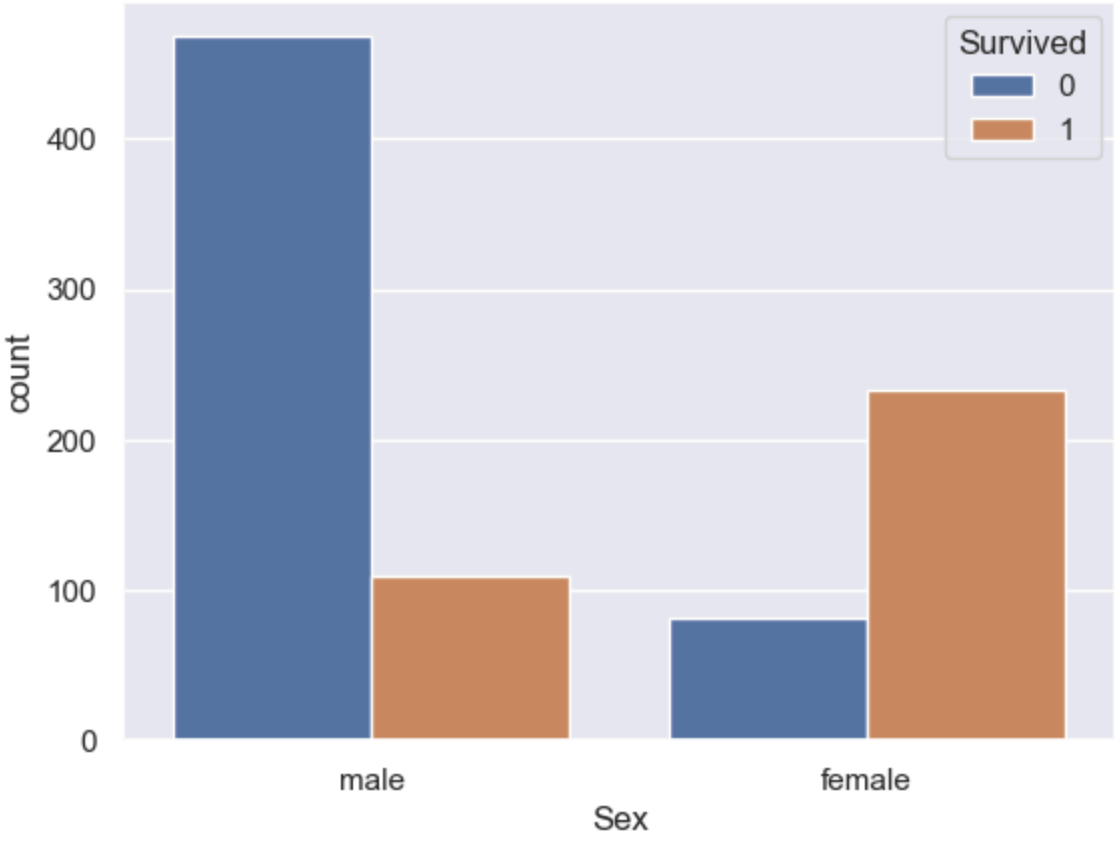
In [48]:
```
sns.countplot(x="Survived",data=titanic_data,hue="Survived")
```

Out[48]:
```
<Axes: xlabel='Survived', ylabel='count'>
```



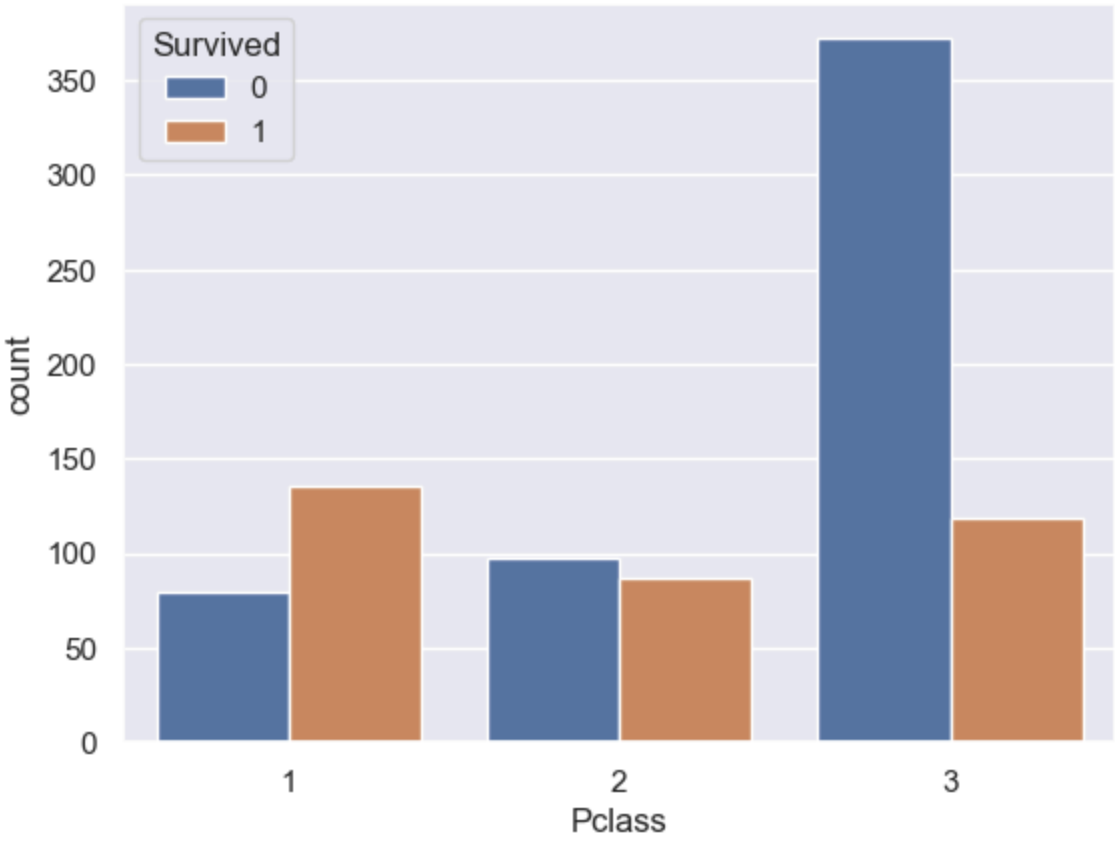
In [53]:
```
sns.countplot(x="Sex",data=titanic_data,hue="Survived")
```

Out[53]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



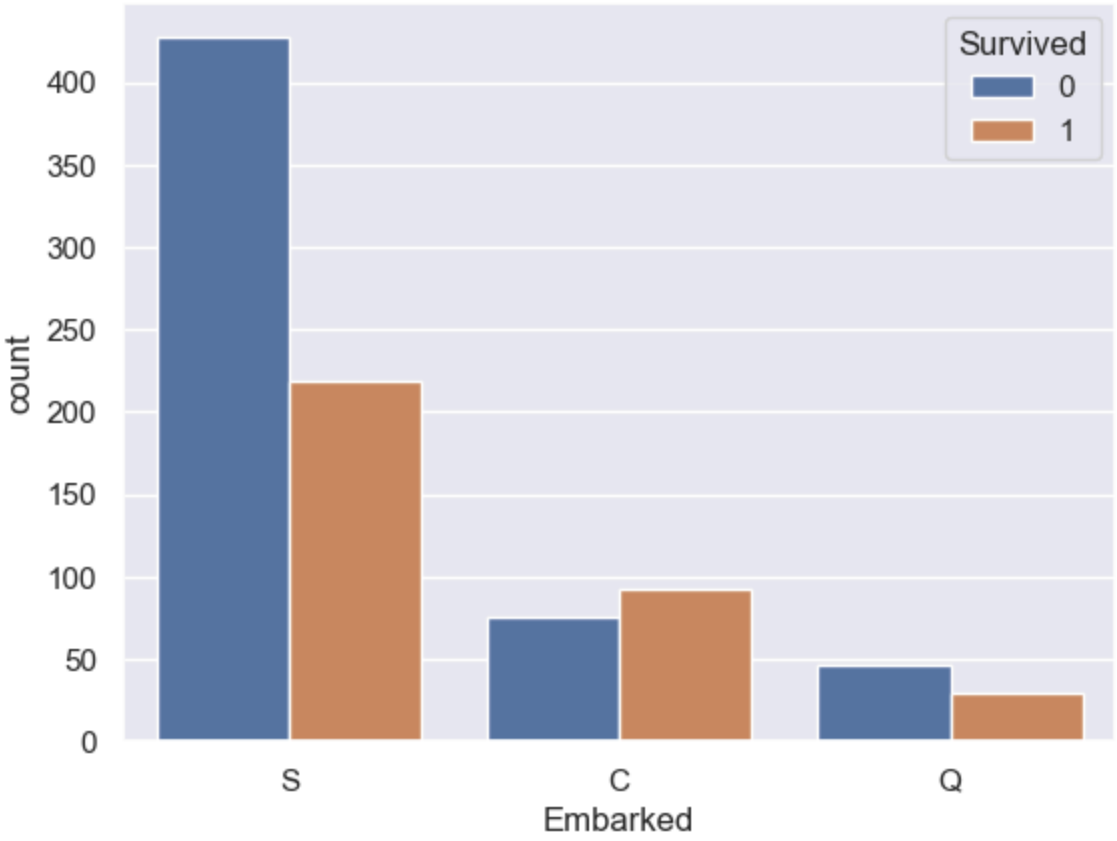
In [54]:
```
sns.countplot(x="Pclass",data=titanic_data,hue="Survived")
```

Out[54]: <Axes: xlabel='Pclass', ylabel='count'>



In [55]:
```
sns.countplot(x="Embarked",data=titanic_data,hue="Survived")
```

Out[55]:
```
<Axes: xlabel='Embarked', ylabel='count'>
```



One_hot Encoding

In [56]:
```
titanic_data['Sex'].value_counts()
```

Out[56]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

In [57]:
```
titanic_data['Embarked'].value_counts()
```

Out[57]:
```
Embarked
S    646
C    168
Q     77
Name: count, dtype: int64
```

In [61]:
```
titanic_data.replace({'Sex':{"male":0 , "female":1},"Embarked":{"S":0,"C":1,"Q":2}},i
```

In [62]:
```
titanic_data.head(3)
```

Out[62]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	I
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	

In [63]:
```
X= titanic_data.drop(columns=["PassengerId","Name","Ticket","Survived"],axis=1)
Y= titanic_data["Survived"]
```

In [65]:
```
X
```

Out[65]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	22.000000	1	0	7.2500	0
1	1	1	38.000000	1	0	71.2833	1
2	3	1	26.000000	0	0	7.9250	0
3	1	1	35.000000	1	0	53.1000	0
4	3	0	35.000000	0	0	8.0500	0
...	...	...	...	...	...	...	...
886	2	0	27.000000	0	0	13.0000	0
887	1	1	19.000000	0	0	30.0000	0
888	3	1	29.699118	1	2	23.4500	0
889	1	0	26.000000	0	0	30.0000	1
890	3	0	32.000000	0	0	7.7500	2

891 rows × 7 columns

In [67]:
```
Y
```

Out[67]:
```
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
```

In [69]:
```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2, random_state=2)
```

In [71]:
```
print(X.shape,x_train.shape,x_test.shape)
```

```
(891, 7) (712, 7) (179, 7)
```

Model training

In [72]:
```
model=LogisticRegression()
```

In [73]:
```
model.fit(x_train,y_train)
```

```
c:\Users\Soumaya\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\lin
ear_model\_logistic.py:469: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[73]:
```
LogisticRegression()
```

Model Evaluation

In [74]:
```
x_train_predictions=model.predict(x_train)
```

In [75]:
```
print(x_train_predictions)
```

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

In [78]:
```
training_data_accuracy=accuracy_score(y_train,x_train_predictions)
print(training_data_accuracy)
```

```
0.8075842696629213
```

In [79]:
```
x_test_predictions=model.predict(x_test)
```

In [81]:
```
print(accuracy_score(y_test,x_test_predictions))
```

```
0.7821229050279329
```

In [82]:
```
model.score(x_test,y_test)*100
```

Out[82]: 78.2122905027933

In [83]:
```
x_train.head(3)
```

Out[83]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
30	1	0	40.0	0	0	27.7208	1
10	3	1	4.0	1	1	16.7000	0
873	3	0	47.0	0	0	9.0000	0