In [1]:
```
#import libraries
```

In [106]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [61]:
```
#load the data
```

In [6]:
```
titanic_data=pd.read_csv('train.csv')
```

In [7]:
```
len(titanic_data)
```

Out[7]:
```
891
```

In [8]:
```
titanic_data.head()
```

Out[8]:

	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 [9]:
```
titanic_data.index
```

Out[9]:
```
RangeIndex(start=0, stop=891, step=1)
```

In [10]:
```
titanic_data.columns
```

Out[10]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [11]:
```
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 [12]:
```
titanic_data.dtypes
```

Out[12]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [13]:
```
titanic_data.describe()
```

Out[13]:

	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

data analysis

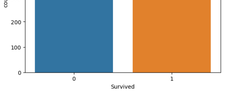
In [14]:
```
#countplot of survived vs not survived
```

In [15]:
```
titanic_data['Survived'].value_counts()
#finding no. of people survived or not survived
```

Out[15]:
```
0    549
1    342
Name: Survived, dtype: int64
```

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

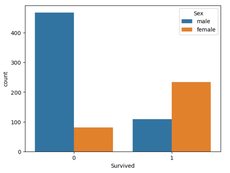
Out[16]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



In [17]:
```
#male vs female
```

In [18]:
```
sns.countplot(x='Survived',data=titanic_data,hue='Sex')
```

Out[18]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



In [19]:
```
#check for null
```

In [20]:
```
titanic_data.isna()
```

Out[20]:

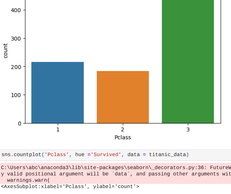
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

In [21]:
```
#making a count plot for the pclass column
sns.countplot('Pclass', data = titanic_data)
```

```
C:\Users\abc\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

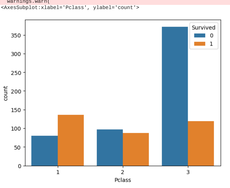
Out[21]:
```
<AxesSubplot:xlabel='Pclass', ylabel='count'>
```



In [22]:
```
sns.countplot('Pclass', hue ='Survived', data = titanic_data)
```

```
C:\Users\abc\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[22]:
```
<AxesSubplot:xlabel='Pclass', ylabel='count'>
```



In [23]:
```
titanic_data['Sex'].value_counts()
```

Out[23]:
```
male      577
female    314
Name: Sex, dtype: int64
```

In [24]:
```
titanic_data['Embarked'].value_counts()
```

Out[24]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [78]:
```
#checking missing  values
titanic_data.isna().sum()
```

Out[78]:
```
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 [25]:
```
#converting categorical columns
titanic_data.replace({'sex':{'male':0,'female':1},' Embarked':{'s':0,'c':1,'Q':2}},inplace =True)
```

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

Out[26]:

	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 [27]:
```
#separating features and target
X = titanic_data.drop(columns = ['PassengerId','Name','Ticket','Survived'],axis =1)
Y = titanic_data['Survived']
```

In [28]:
```
print(X)
```

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

[891 rows x 8 columns]
```

In [29]:
```
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
```

In [30]:
```
X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state =2)
```

In [31]:
```
print(X.shape, X_train.shape, X_test.shape)
```

```
(891, 8) (712, 8) (179, 8)
```

In [106]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: