Bharat Intern

Task - 2 (Titanic Classification)

Done by Cherukuri Krishna Lathvik

Importing the Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Loading the dataset

In [4]:
```
df = pd.read_csv("tested.csv")
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN

In [5]:
```
df.shape
```

Out[5]:
```
(418, 12)
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [10]:
```
df.describe()
```

Out[10]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [7]:
```
df.isnull().any()
```

Out[7]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare            True
Cabin           True
Embarked       False
dtype: bool
```

In [8]:
```
df['Survived'].value_counts()
# 0 means Not Survived
# 1 means Survived
```

Out[8]:
```
0    266
1    152
Name: Survived, dtype: int64
```

In [9]:
```
df['Sex'].value_counts()
```

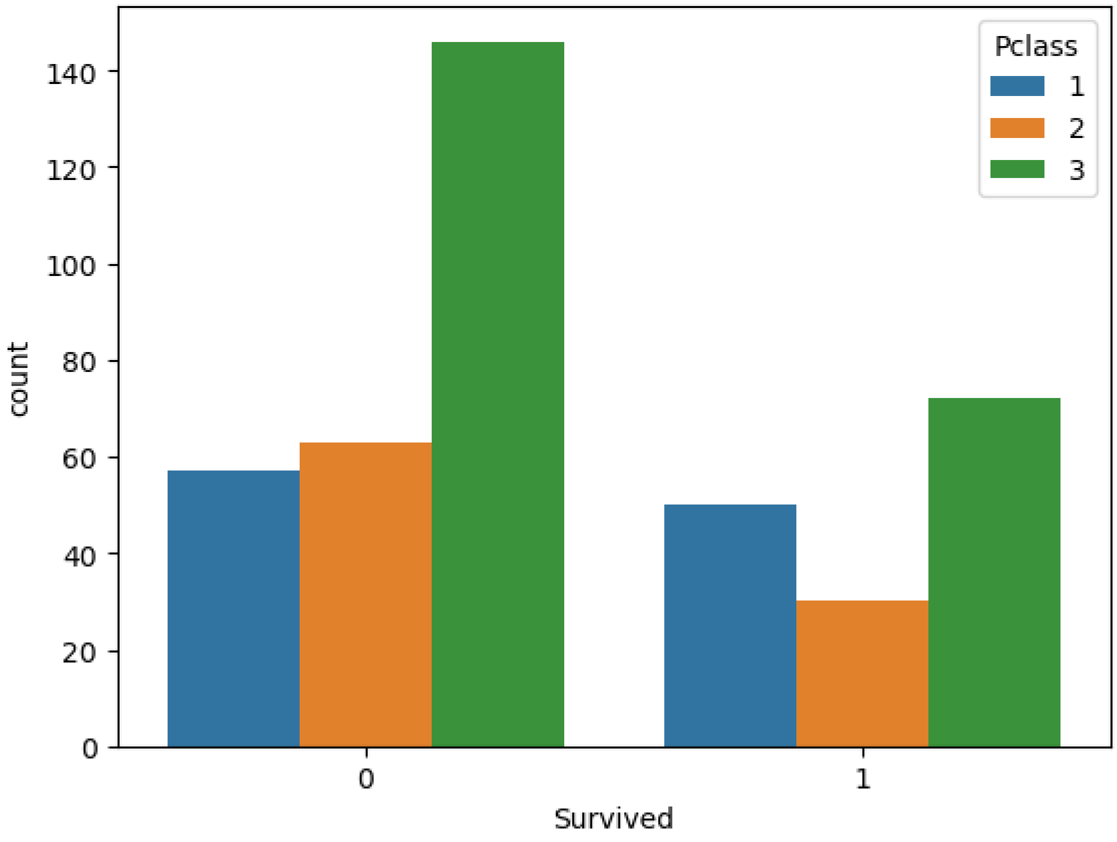
```
Out[9]: male      266
        female    152
        Name: Sex, dtype: int64
```

Data Visualisation

Visualising the count of Survivals with respect to Pclass parameter

```
In [13]: sns.countplot(x=df['Survived'], hue=df['Pclass'])
```

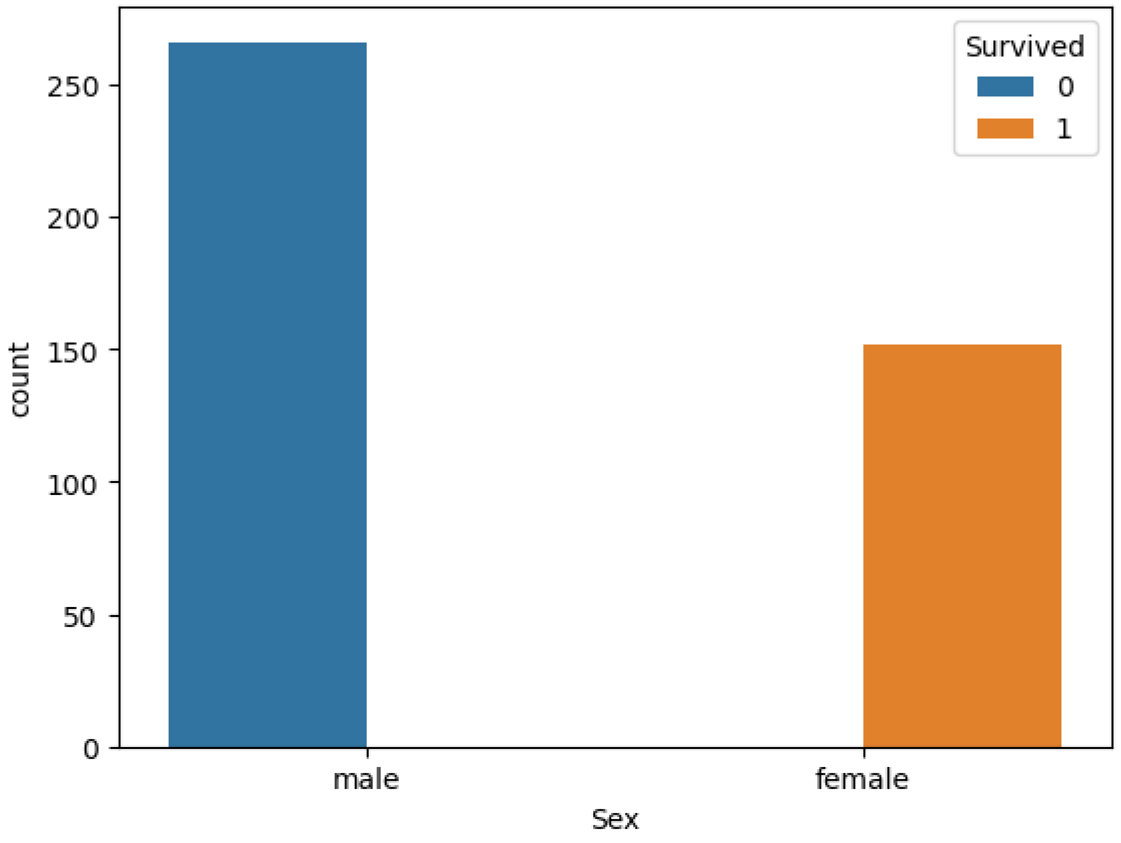
```
Out[13]: <Axes: xlabel='Survived', ylabel='count'>
```



Visualising the count of Survivals with respect to Gender parameter

```
In [13]: sns.countplot(x=df['Sex'], hue=df['Survived'])
```

```
Out[13]: <Axes: xlabel='Sex', ylabel='count'>
```



In [14]:
```
df['Sex']
```

Out[14]:
```
0        male
1      female
2        male
3        male
4      female
        ...  
413      male
414    female
415      male
416      male
417      male
Name: Sex, Length: 418, dtype: object
```

In [15]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()

df['Sex'] = labelencoder.fit_transform(df['Sex'])
df.head()
# Male = 1
# Female = 0
```

Out[15]:

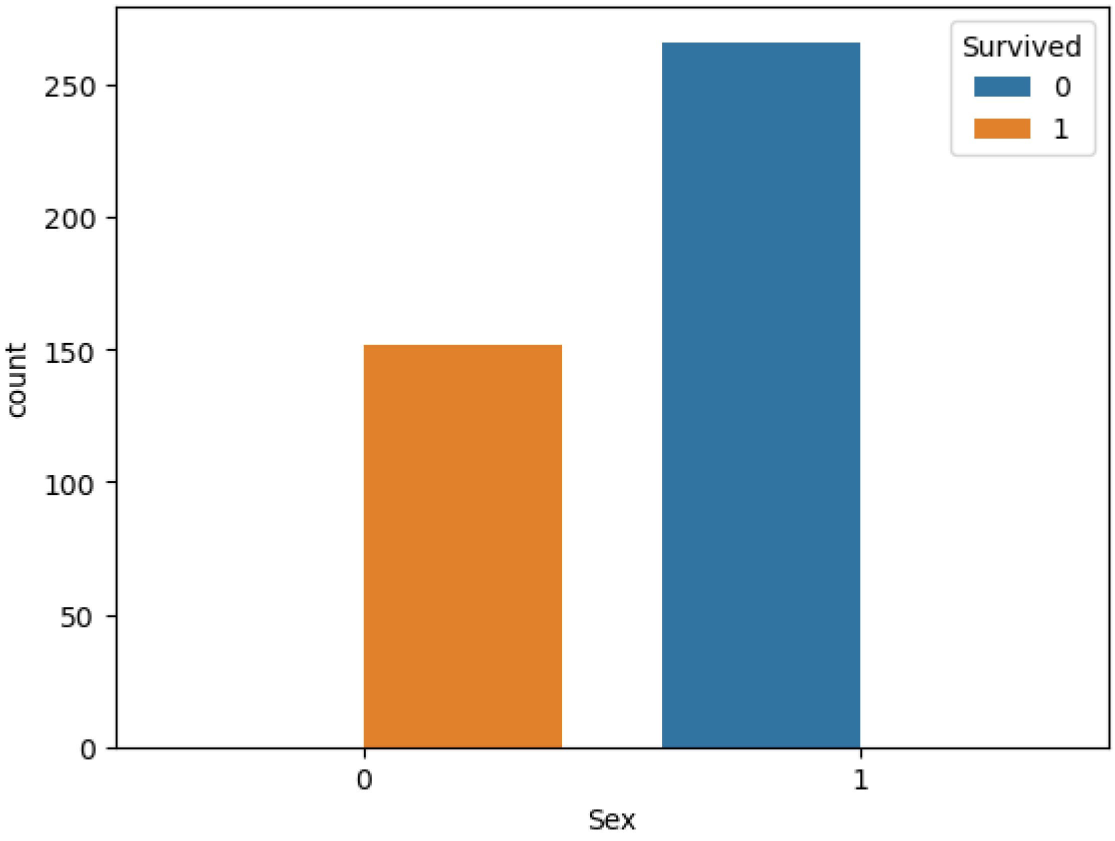
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	En
0	892	0	3	Kelly, Mr. James	1	34.5	0	0	330911	7.8292	NaN	
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	47.0	1	0	363272	7.0000	NaN	
2	894	0	2	Myles, Mr. Thomas Francis	1	62.0	0	0	240276	9.6875	NaN	
3	895	0	3	Wirz, Mr. Albert	1	27.0	0	0	315154	8.6625	NaN	
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	22.0	1	1	3101298	12.2875	NaN	

In [16]:
```
df['Sex'], df['Survived']
```

Out[16]:
```
(0      1
 1      0
 2      1
 3      1
 4      0
       ..
 413    1
 414    0
 415    1
 416    1
 417    1
 Name: Sex, Length: 418, dtype: int32,
 0      0
 1      1
 2      0
 3      0
 4      1
       ..
 413    0
 414    1
 415    0
 416    0
 417    0
 Name: Survived, Length: 418, dtype: int64)
```

In [17]:
```
sns.countplot(x=df['Sex'], hue=df['Survived'])
```

Out[17]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



In [18]:
```
df.isna().sum()
```

Out[18]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

Dropping the Age column

In [19]:
```
if "Age" in df.columns:
    df.drop("Age", axis=1, inplace=True)
```

In [20]:
```
new_df = df
new_df.head()
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	892	0	3	Kelly, Mr. James	1	0	0	330911	7.8292	NaN	
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	0	1	0	363272	7.0000	NaN	
2	894	0	2	Myles, Mr. Thomas Francis	1	0	0	240276	9.6875	NaN	
3	895	0	3	Wirz, Mr. Albert	1	0	0	315154	8.6625	NaN	
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	0	1	1	3101298	12.2875	NaN	

Training the Model

In [21]:
```
X = df[['Pclass', 'Sex']]
Y = df['Survived']
```

In [22]:
```
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_s
```

In [23]:
```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state=0)
log.fit(X_train, Y_train)
```

Out[23]:
```
▾ LogisticRegression
LogisticRegression(random_state=0)
```

Model Prediction

In [26]:
```
pred = print(log.predict(X_test))
```

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

In [27]:
```
print(Y_test)
```

```
360    0
170    0
224    1
358    0
309    1
      ..
100    1
7      0
22     1
68     0
328    0
Name: Survived, Length: 84, dtype: int64
```

In [42]:

```
import warnings
warnings.filterwarnings("ignore")

res = log.predict([[2,0]])

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

```
Survived
```