```
Task-1 TITANIC SURVIVAL PREDICTION

Author: Bandana Prakash
Batch :June
Domain: Data Science
Aim: To build a model that predicts whether a passenger on Titanic Survived or not.
```

IMPORTING IMPORTANT LIBRARIES

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING DATASET

In [3]:
```
train = pd.read_csv('Titanic.csv')
train.head(10)
```

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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

In [4]:
```
train.shape
```

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

In [5]:
```
train.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

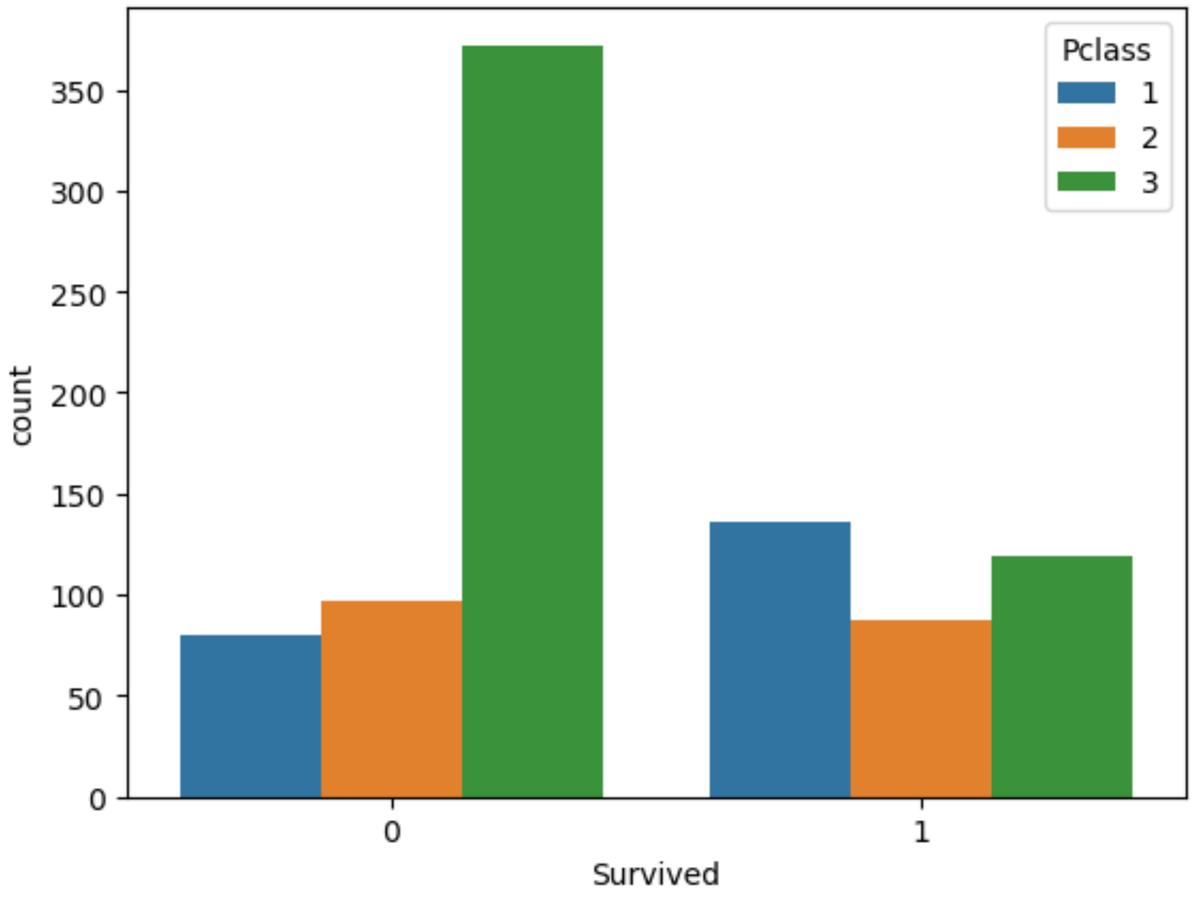
From above cell it is clear there there are few missing values in age column

In [6]:
```
train['Survived'].value_counts()
```

Out[6]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

In [7]:
```
#let's visualize the count of survivals wrt pclass
sns.countplot(x=train['Survived'],hue=train['Pclass'])
```

Out[7]:
```
<Axes: xlabel='Survived', ylabel='count'>
```

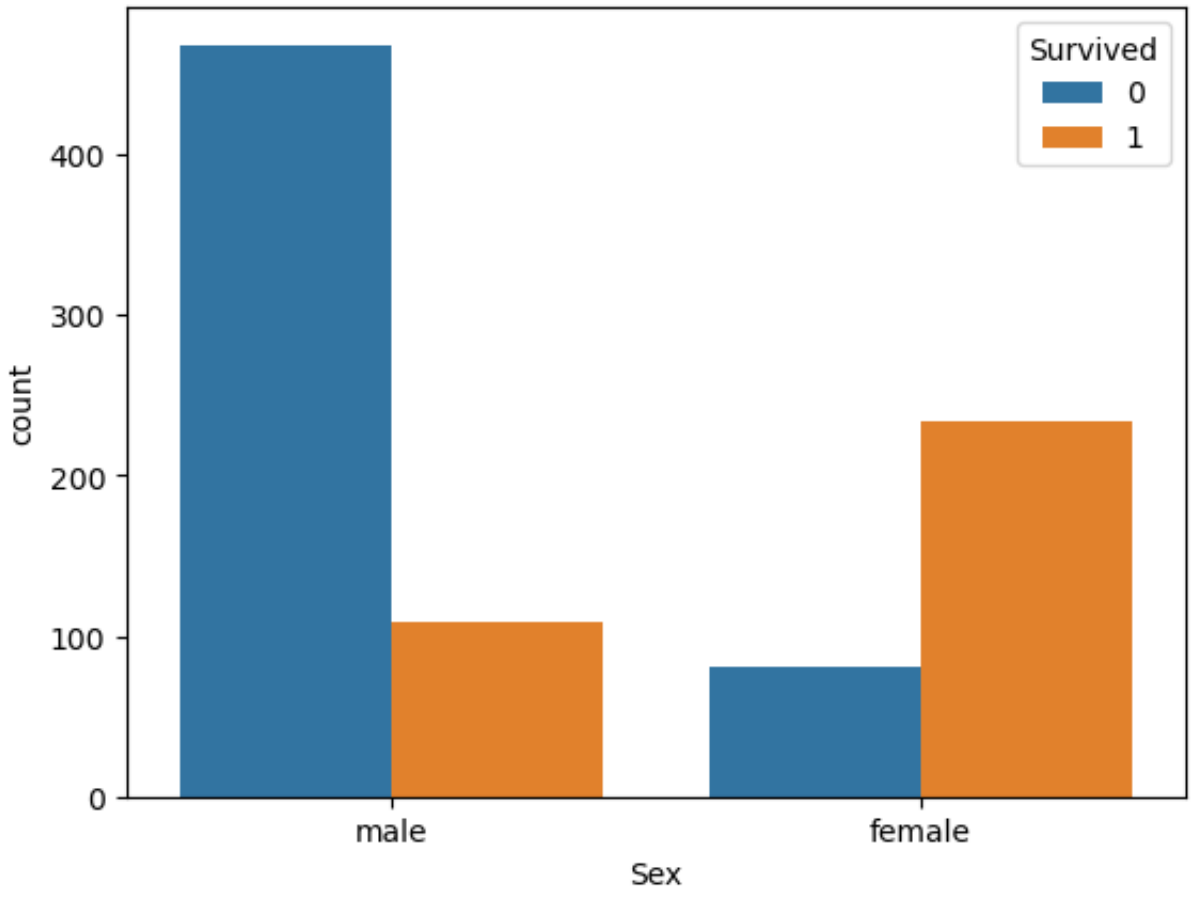


In [8]:
```
train["Sex"]
```

Out[8]:
```
0        male
1      female
2      female
3      female
4        male
        ...  
886      male
887    female
888    female
889      male
890      male
Name: Sex, Length: 891, dtype: object
```

In [9]:
```
#let's visualize the count of survivals wrt Gender
sns.countplot(x=train['Sex'],hue=train['Survived'])
```

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



In [10]:
```
#Look at survival rate by sex
train.groupby('Sex')[['Survived']].mean()
```

Out[10]:

	Survived
Sex	
female	0.742038
male	0.188908

In [11]:
```
train['Sex'].unique()
```

Out[11]:
```
array(['male', 'female'], dtype=object)
```

In [12]:
```
from sklearn.preprocessing import LabelEncoder
LabelEncoder = LabelEncoder()

train['Sex']= LabelEncoder.fit_transform(train['Sex'])

train.head()
```

Out[12]:

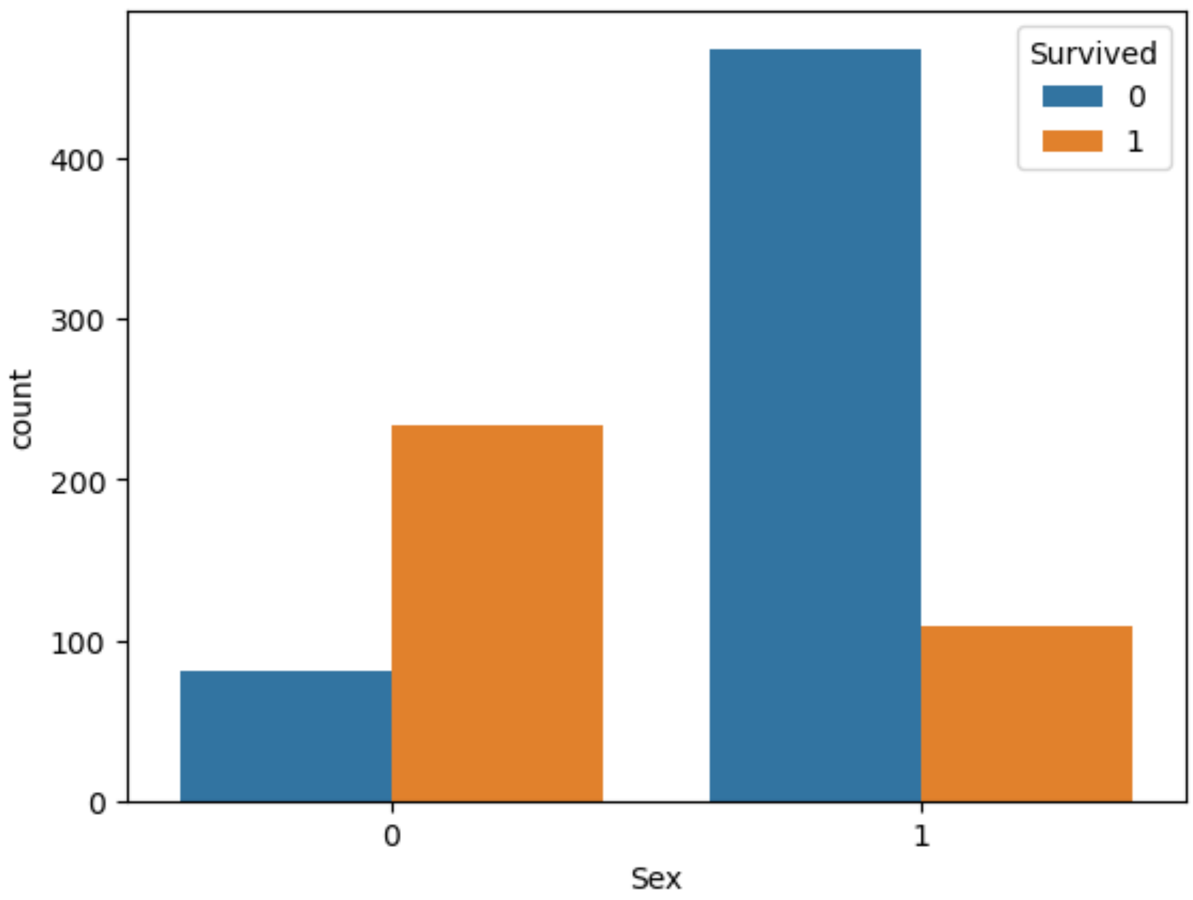
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250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

In [13]:
```
train['Sex'],train['Survived']
```

Out[13]:
```
(0      1
 1      0
 2      0
 3      0
 4      1
       ..
 886    1
 887    0
 888    0
 889    1
 890    1
 Name: Sex, Length: 891, dtype: int64,
 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 [14]:
```
sns.countplot(x=train['Sex'],hue=train["Survived"])
```

Out[14]:
```
<Axes: xlabel='Sex', ylabel='count'>
```



In [15]:
```
train.isna().sum()
```

Out[15]:
```
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 [16]:
```
train = train.drop(['Age'], axis=1)
```

In [17]:
```
train_final = train
train_final.head(10)
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	1	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	1	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	1	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	1	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	0	1	0	237736	30.0708	NaN	C

MODEL TRAINING

In [18]:
```
X = train[['Pclass','Sex']]
Y = train['Survived']
```

In [19]:
```
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_state= 0)
```

In [20]:
```
from sklearn.linear_model import LogisticRegression

log = LogisticRegression( random_state = 0 )
log.fit(X_train, Y_train)
```

Out[20]:
```
LogisticRegression(random_state=0)
```

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.

MODEL PREDICTION

In [21]:
```
pred = print(log.predict(X_test))
```

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

In [22]:
```
print( Y_test )
```

```
495    0
648    0
278    0
31     1
255    1
      ..
780    1
837    0
215    1
833    0
372    0
Name: Survived, Length: 179, dtype: int64
```

In [23]:
```
import warnings
warnings.filterwarnings("ignore")

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

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

```
Survived
```