

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

In [ ]: IMPORTING DATASET

In [4]: import os

In [6]: os.getcwd()

Out[6]: 'C:\\Users\\shruw'

In [7]: os.chdir('C:\\Users\\shruw\\Desktop')

In [8]: df=pd.read_csv("Titanic-Dataset.csv")

In [9]: df.head()

Out[9]:
   PassengerId  Survived  Pclass
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
      Name               Sex  Age  SibSp  Parch    Ticket   Fare  Cabin Embarked
0  Braund, Mr. Owen Harris   male  22.0    1    0      A/5 21171   7.2500   NaN        S
1  Cumings, Mrs. John Bradley (Florence Briggs Th... female  38.0    1    0      PC 17599  71.2833   C85        C
2  Heikkinen, Miss. Laina    female  26.0    0    0  STON/O2. 3101282   7.9250   NaN        S
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0    1    0      113803  53.1000  C123        S
4  Allen, Mr. William Henry   male  35.0    0    0      373450   8.0500   NaN        S

In [12]: df.tail()

Out[12]:
   PassengerId  Survived  Pclass
886           887         0       2
887           888         1       1
888           889         0       3
889           890         1       1
890           891         0       3
      Name               Sex  Age  SibSp  Parch    Ticket   Fare  Cabin Embarked
886  Montvila, Rev. Juozas    male  27.0    0    0      211536  13.00   NaN        S
887  Graham, Miss. Margaret Edith  female  19.0    0    0      112053  30.00   B42        S
888  Johnston, Miss. Catherine Helen "Carrie"  female  NaN    1    2  W./C. 6607   23.45   NaN        S
889  Behr, Mr. Karl Howell    male  26.0    0    0      111369  30.00  C148        C
890  Dooley, Mr. Patrick     male  32.0    0    0      370376   7.75   NaN        Q

In [13]: df.shape

Out[13]: (891, 12)

In [14]: df.describe()

Out[14]:
   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 [15]: df['Survived'].value_counts()

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

In [16]: #Let's visualize the count of survivals wrt pclass
sns.countplot(x=df['Survived'],hue=df['Pclass'])

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

In [17]: df["Sex"]

Out[17]:
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 [19]: #Let's visualize the count of survivals write
sns.countplot(x=df['Sex'],hue=df['Survived'])

Out[19]:
<AxesSubplot: xlabel='Sex', ylabel='count'>

In [20]: #Look at survival rate by sex
df.groupby('Sex')[['Survived']].mean()

Out[20]:
Survived
Sex
female  0.742038
male    0.188908

In [21]: df['Sex'].unique()

Out[21]: array(['male', 'female'], dtype=object)

In [23]: from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['Sex']=labelencoder.fit_transform(df['Sex'])
df.head()

Out[23]:
   PassengerId  Survived  Pclass
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
      Name               Sex  Age  SibSp  Parch    Ticket   Fare  Cabin Embarked
0  Braund, Mr. Owen Harris    1  22.0    1    0      A/5 21171   7.2500   NaN        S
1  Cumings, Mrs. John Bradley (Florence Briggs Th...    0  38.0    1    0      PC 17599  71.2833   C85        C
2  Heikkinen, Miss. Laina    0  26.0    0    0  STON/O2. 3101282   7.9250   NaN        S
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    0  35.0    1    0      113803  53.1000  C123        S
4  Allen, Mr. William Henry    1  35.0    0    0      373450   8.0500   NaN        S

In [24]: df['Sex'],df['Survived']

Out[24]:
(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: int32,
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 [26]: sns.countplot(x=df['Sex'],hue=df["Survived"])

Out[26]:
<AxesSubplot: xlabel='Sex', ylabel='count'>

In [27]: df.isna().sum()

Out[27]:
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 [28]: #After dropping non required column
df=df.drop(['Age'],axis=1)

In [29]: df_final=df
df_final.head(10)

Out[29]:
   PassengerId  Survived  Pclass
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
5            6         0       3
6            7         0       1
7            8         0       3
8            9         1       3
9           10         1       2
      Name               Sex  SibSp  Parch    Ticket   Fare  Cabin Embarked
0  Braund, Mr. Owen Harris    1     1    0      A/5 21171   7.2500   NaN        S
1  Cumings, Mrs. John Bradley (Florence Briggs Th...    0     1    0      PC 17599  71.2833   C85        C
2  Heikkinen, Miss. Laina    0     0    0  STON/O2. 3101282   7.9250   NaN        S
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    0     1    0      113803  53.1000  C123        S
4  Allen, Mr. William Henry    1     0    0      373450   8.0500   NaN        S
5  Moran, Mr. James          1     0    0      330877   8.4583   NaN        Q
6  McCarthy, Mr. Timothy J    1     0    0      17463   51.8625   E46        S
7  Palsson, Master. Gosta Leonard    1     3     1      349909  21.0750   NaN        S
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)    0     0     2      347742  11.1333   NaN        S
9  Nasser, Mrs. Nicholas (Adele Achem)    0     1     0      237736  30.0708   NaN        C

Model training

In [30]: x=df[['Pclass','Sex']]
y=df['Survived']

In [33]: 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 [34]: from sklearn.linear_model import LogisticRegression
log=LogisticRegression(random_state=0)
log.fit(X_train,Y_train)

Out[34]: LogisticRegression(random_state=0)

In [35]: pred=print(log.predict(X_test))

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

In [36]: print(Y_test)

495     0
648     0
278     0
31      1
255     1
..
789     1
837     0
215     1
833     0
372     0
Name: Survived, Length: 179, dtype: int64

In [37]: import warnings
warnings.filterwarnings("ignore")
res=log.predict([[2,1]])
if(res==0):
    print("So Sorry! Not Survived")
else:
    print("Survived")

So Sorry! Not Survived

In [ ]:
```