1) Importing the Required Libraries

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

2) Read the data and Explore it

```
Titanic = pd.read_csv("/content/Titanic.csv")
Survival= pd.read_csv("/content/Survival.csv")
Titanic.describe(include="all")
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.

3)Analyzing the Data

```
print(Titanic.columns)
```

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

```
Titanic.describe(include = "all")
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.

```
print(pd.isnull(Titanic).sum())
```

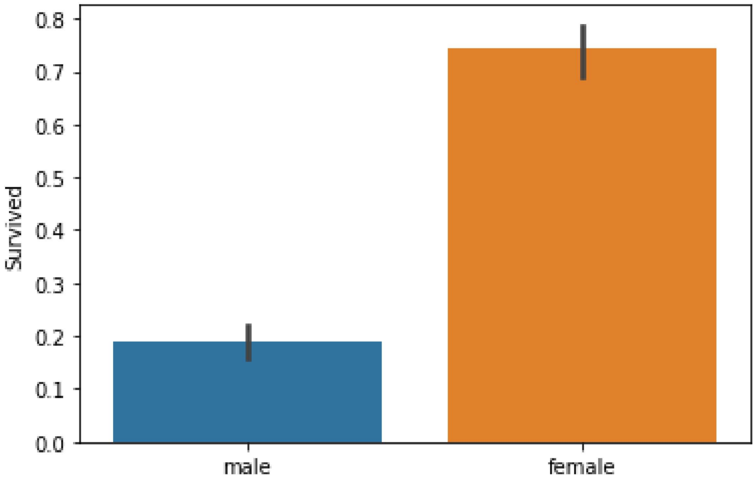
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

4) Data Visualization

Sex Feature

```
sns.barplot(x="Sex", y="Survived", data=Titanic)
print("Female Percentage who have Survived:",Titanic["Survived"][Titanic["Sex"] == 'female'].value_counts(normalize =
print("Male Percentage who have Survived:", Titanic["Survived"][Titanic["Sex"] == 'male'].value_counts(normalize = Tru
print("Chance For Female is More Than Male")
print("Female>Male")
```

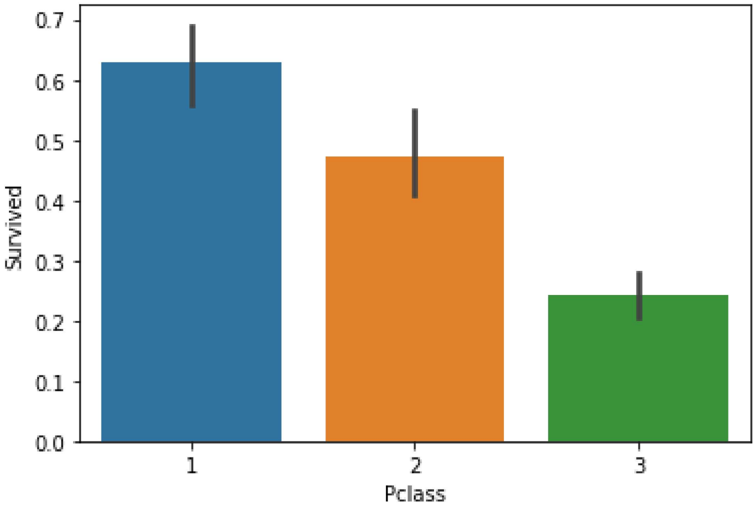
```
Female Percentage who have Survived: 74.20382165605095
Male Percentage who have Survived: 18.890814558058924
Chance For Female is More Than Male
Female>Male
```



Pclass Feature

```
sns.barplot(x="Pclass", y="Survived", data=Titanic)
print("Pclass =1 Percentage who survived:", Titanic["Survived"][Titanic["Pclass"] == 1].value_counts(normalize = True)[1
print("Pclass =2 Percentage who survived:", Titanic["Survived"][Titanic["Pclass"] == 2].value_counts(normalize = True)[
print("Pclass =3 Percentage who survived:", Titanic["Survived"][Titanic["Pclass"] == 3].value_counts(normalize = True)[
print("P1>P2>P3")
```

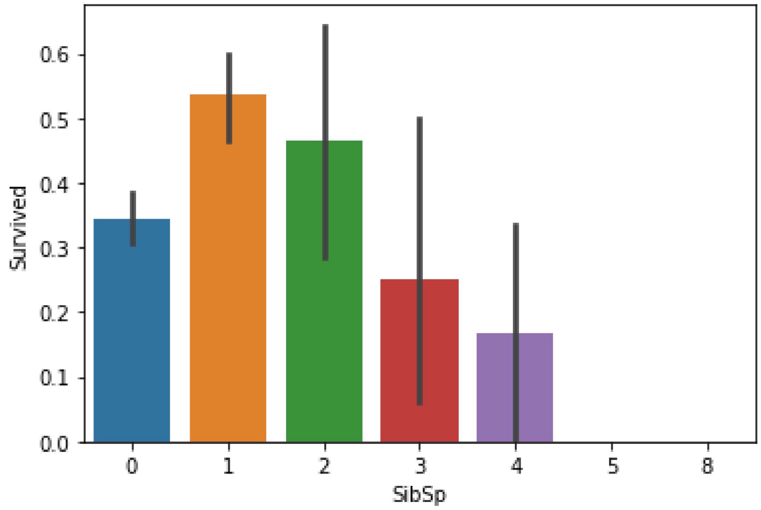
```
Pclass =1 Percentage who survived: 62.96296296296296
Pclass =2 Percentage who survived: 47.28260869565217
Pclass =3 Percentage who survived: 24.236252545824847
P1>P2>P3
```



SibSp Feature

```
sns.barplot(x="SibSp", y="Survived", data=Titanic)
print("SibSp = 0 Percentage who survived:", Titanic["Survived"][Titanic["SibSp"] == 0].value_counts(normalize = True)[
print("SibSp = 1  Percentage who survived:", Titanic["Survived"][Titanic["SibSp"] == 1].value_counts(normalize = True)[1
print("SibSp = 2 Percentage who survived:", Titanic["Survived"][Titanic["SibSp"] == 2].value_counts(normalize = True)[
print("SibSp(0)<(SibSp(1)&SibSp(2))")
```

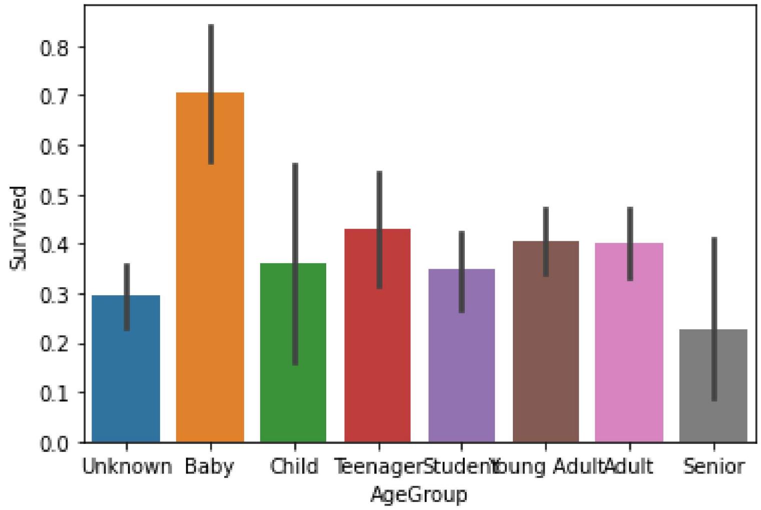
```
SibSp = 0 Percentage who survived: 34.53947368421053
SibSp = 1  Percentage who survived: 53.588516746411486
SibSp = 2 Percentage who survived: 46.42857142857143
SibSp(0)<(SibSp(1)&SibSp(2))
```



Age Feature

```
Titanic["Age"] = Titanic["Age"].fillna(-0.5)
Survival["Age"] = Survival["Age"].fillna(-0.5)
bins = [-1, 0, 5, 12, 18, 24, 35, 60, np.inf]
labels = ['Unknown', 'Baby', 'Child', 'Teenager', 'Student', 'Young Adult', 'Adult', 'Senior']
Titanic['AgeGroup'] = pd.cut(Titanic["Age"], bins, labels = labels)
Survival['AgeGroup'] = pd.cut(Survival["Age"], bins, labels = labels)
```

```
sns.barplot(x="AgeGroup", y="Survived", data=Titanic)
plt.show()
```



5) Cleaning Data

```
Titanic.describe(include="all")
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	
count	891.000000	891.000000	891.000000	891	891	891.000000	891.000000	891.
unique	NaN	NaN	NaN	891	2	NaN	NaN	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	
freq	NaN	NaN	NaN	1	577	NaN	NaN	
mean	446.000000	0.383838	2.308642	NaN	NaN	23.699966	0.523008	0.
std	257.353842	0.486592	0.836071	NaN	NaN	17.731181	1.102743	0.
min	1.000000	0.000000	1.000000	NaN	NaN	-0.500000	0.000000	0.
25%	223.500000	0.000000	2.000000	NaN	NaN	6.000000	0.000000	0.
50%	446.000000	0.000000	3.000000	NaN	NaN	24.000000	0.000000	0.
75%	668.500000	1.000000	3.000000	NaN	NaN	35.000000	1.000000	0.

Droping Cabin Feature

```
Titanic = Titanic.drop(['Cabin'], axis = 1)
Survival = Survival.drop(['Cabin'], axis = 1)
```

Droping Ticket Feature

```
Titanic = Titanic.drop(['Ticket'], axis = 1)
Survival = Survival.drop(['Ticket'], axis = 1)
```

Age Feature

```
combine = [Titanic,Survival]
```

```
for dataset in combine:
    dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.', expand=False)

pd.crosstab(Titanic['Title'], Titanic['Sex'])
```

Sex	female	male
Title		
Capt	0	1
Col	0	2
Countess	1	0
Don	0	1
Dr	1	6
Jonkheer	0	1
Lady	1	0
Major	0	2
Master	0	40
Miss	182	0
Mlle	2	0
Mme	1	0
Mr	0	517
Mrs	125	0
Ms	1	0
Rev	0	6
Sir	0	1

```
for dataset in combine:
    dataset['Title'] = dataset['Title'].replace(['Lady', 'Capt', 'Col',
    'Don', 'Dr', 'Major', 'Rev', 'Jonkheer', 'Dona'], 'Rare')
    dataset['Title'] = dataset['Title'].replace(['Countess', 'Lady', 'Sir'], 'Royal')
    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')

Titanic[['Title', 'Survived']].groupby(['Title'], as_index=False).mean()
```

	Title	Survived
0	Master	0.575000
1	Miss	0.702703
2	Mr	0.156673
3	Mrs	0.793651

```
title_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Royal": 5, "Rare": 6}
for dataset in combine:
    dataset['Title'] = dataset['Title'].map(title_mapping)
    dataset['Title'] = dataset['Title'].fillna(0)
```

```
Titanic.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embark
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.0	1	0	71.2833	

Predicting the missing age values

```
mr_age = Titanic[Titanic["Title"] == 1]["AgeGroup"].mode()
miss_age = Titanic[Titanic["Title"] == 2]["AgeGroup"].mode()
mrs_age = Titanic[Titanic["Title"] == 3]["AgeGroup"].mode()
master_age = Titanic[Titanic["Title"] == 4]["AgeGroup"].mode()
royal_age = Titanic[Titanic["Title"] == 5]["AgeGroup"].mode()
rare_age = Titanic[Titanic["Title"] == 6]["AgeGroup"].mode()
age_title_mapping = {1: "Young Adult", 2: "Student", 3: "Adult", 4: "Baby", 5: "Adult", 6: "Adult"}

for x in range(len(Titanic["AgeGroup"])):
    if Titanic["AgeGroup"][x] == "Unknown":
        Titanic["AgeGroup"][x] = age_title_mapping[Titanic["Title"][x]]

for x in range(len(Survival["AgeGroup"])):
    if Survival["AgeGroup"][x] == "Unknown":
        Survival["AgeGroup"][x] = age_title_mapping[Survival["Title"][x]]
```

```
age_mapping = {'Baby': 1, 'Child': 2, 'Teenager': 3, 'Student': 4, 'Young Adult': 5, 'Adult': 6, 'Senior': 7}
Titanic['AgeGroup'] = Titanic['AgeGroup'].map(age_mapping)
Survival['AgeGroup'] = Survival['AgeGroup'].map(age_mapping)

Titanic.head()
#Droping age just for now
Titanic = Titanic.drop(['Age'], axis = 1)
Survival = Survival.drop(['Age'], axis = 1)
```

Droping Name

```
Titanic = Titanic.drop(['Name'], axis = 1)
Survival = Survival.drop(['Name'], axis = 1)
```

Sex Feature

```
sex_mapping = {"male": 0, "female": 1}
Titanic['Sex'] = Titanic['Sex'].map(sex_mapping)
Survival['Sex'] = Survival['Sex'].map(sex_mapping)
```

```
Titanic.head()
```

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Fare	Embarked	AgeGroup	CabinB
0	1	0	3	0	1	0	7.2500	S	4.0	
1	2	1	1	1	1	0	71.2833	C	6.0	
2	3	1	3	1	0	0	7.9250	S	5.0	
3	4	1	1	1	1	0	53.1000	S	5.0	
4	5	0	3	0	0	0	8.0500	S	5.0	

```
embarked_mapping = {"S": 1, "C": 2, "Q": 3}
Titanic['Embarked'] = Titanic['Embarked'].map(embarked_mapping)
Survival['Embarked'] = Survival['Embarked'].map(embarked_mapping)
```

Droping Fare Values

```
Titanic = Titanic.drop(['Fare'], axis = 1)
```

```
Survival = Survival.drop(['Fare'], axis = 1)
```

```
Titanic.head()
```

	PassengerId	Survived	Pclass	Sex	SibSp	Parch	Embarked	AgeGroup	CabinBool	Title
0	1	0	3	0	1	0	1	4.0	0	1
1	2	1	1	1	1	0	2	6.0	1	3
2	3	1	3	1	0	0	1	5.0	0	2
3	4	1	1	1	1	0	1	5.0	1	3
4	5	0	3	0	0	0	1	5.0	0	1

```
Survival.head()
```

	PassengerId	Pclass	Sex	SibSp	Parch	Embarked	AgeGroup	CabinBool	Title
0	892	3	0	0	0	3	5.0	0	1
1	893	3	1	1	0	1	6.0	0	3
2	894	2	0	0	0	3	7.0	0	1
3	895	3	0	0	0	1	5.0	0	1
4	896	3	1	1	1	1	4.0	0	3

Splitting the data we got

```
from sklearn.model_selection import train_test_split

predictors = Titanic.drop(['Survived', 'PassengerId'], axis=1)
target = Titanic["Survived"]
x_train, x_val, y_train, y_val = train_test_split(predictors, target, test_size = 0.22, random_state = 0)
```

BY LOGISTIC REGRESSION MODEL

```
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()
logreg.fit(x_train, y_train)
y_pred = logreg.predict(x_val)
acc_logreg = accuracy_score(y_pred, y_val)
print(acc_logreg)
print("In Percentage:")
```

```
acc_logreg = round(accuracy_score(y_pred, y_val) * 100,4)
print(acc_logreg)
```

```
0.8071065989847716
In Percentage:
80.7107
```