Experiment No. 2
Analyze the Titanic Survival Dataset and apply appropriate regression technique
Date of Performance:
Date of Submission:

Aim: Analyze the Titanic Survival Dataset and apply appropriate Regression Technique.

Objective: Able to perform various feature engineering tasks, apply logistic regression on the given dataset and maximize the accuracy.

Theory:

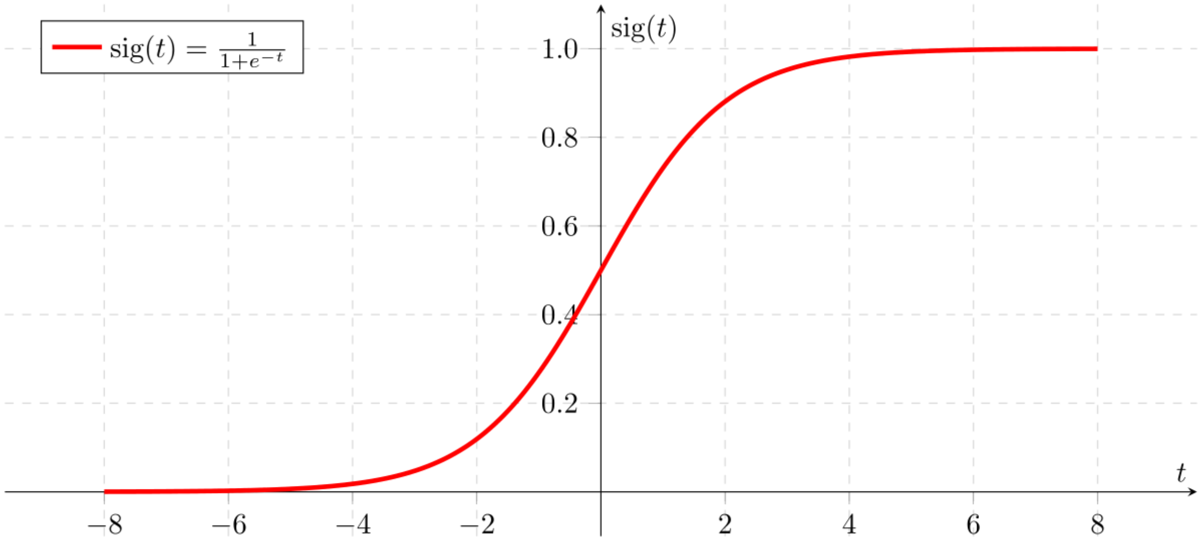
Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical and is binary in nature. In order to perform binary classification, the logistic regression techniques make use of Sigmoid function.

For example,

To predict whether an email is spam (1) or (0)

Whether the tumor is malignant (1) or not (0)

Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be classified as not malignant which can lead to serious consequence in real time.



From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

Dataset:

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this challenge, we ask you to build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (ie name, age, gender, socio-economic class, etc).

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)

1st = Upper, 2nd = Middle, 3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...,

Sibling = brother, sister, stepbrother, stepsister

Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...

Parent = mother, father

Child = daughter, son, stepdaughter, stepson

Some children travelled only with a nanny, therefore parch=0 for them.

Conclusion:

1. What are features have been chosen to develop the model? Justify the features chosen to determine the survival of a passenger.

   When developing a model to predict passenger survival, various features (also known as variables or attributes) can be considered. These features should be chosen based on their potential relevance to determining survival outcomes. Features considered while predicting the model were:

   1) **Age**: Age can play a significant role in survival, as children and elderly individuals might have received priority during evacuation due to their vulnerability.

   2) Sex: Historically, women and children were often given priority in evacuation situations. As such, gender can be an important feature as it might have influenced survival rates.

   3) Passenger Class (Pclass): Higher-class passengers might have had easier access to lifeboats and other safety measures.

   4) Fare: Fare could be correlated with passenger class and, indirectly, socio-economic status.

   5) PassengerId: To keep track of total number of passengers (unique Id).

   6) Ticket: Ticket number to identify the cabins of passenger.

8) Embarked: Port of embarkation

2. Comment on the accuracy obtained.

   Logistic regression is a commonly used algorithm for binary classification tasks, making it suitable for predicting survival outcomes in the Titanic dataset, where the target variable is binary (survived or not survived). The dataset was splitted in training and testing validation in the ratio 70:30. Here the accuracy obtained before feature engineering is 80.6 % and after feature engineering the accuracy obtained is 83.58 %.

Code:

Titanic Survival

Imports

```
import pandas as pd
import numpy as np
```

Reading CSV Files

```
df=pd.read_csv("/content/Titanic-Dataset.csv")
```

Data Preprocessing

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embark
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	

```
df.isnull().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
```

```
df.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
```

```
df.describe()
```

	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

```
df["Age"].fillna(value = df["Age"].mean(), inplace=True)
df["Embarked"].fillna(df["Embarked"].value_counts().idxmax(), inplace=True)
```

```
df['Sex'] = df['Sex'].replace({'male':0, 'female':1})
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

```
df.isnull().sum()
```

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

```
from sklearn.preprocessing import LabelEncoder

labelencoder_X = LabelEncoder()
```

```
df["Name"] = labelencoder_X.fit_transform(df["Name"])
df["Embarked"] = labelencoder_X.fit_transform(df["Embarked"])
df["Cabin"] = labelencoder_X.fit_transform(df["Cabin"])
df["Ticket"] = df["Ticket"].astype(str)
df["Ticket"] = labelencoder_X.fit_transform(df["Ticket"])
```

```
x = df.drop("Survived", axis=1)
y = df["Survived"]
```

Algorithms

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
```

```
model = LogisticRegression(max_iter=5000)
model.fit(x_train,y_train)
p_predict = model.predict(x_test)
```

Accuracy

```
print("The accuracy is", round(accuracy_score(p_predict, y_test) * 100,2))
```

```
     The accuracy is 80.6
```

Feature Engineering

```
df.drop(["Cabin"],axis=1,inplace=True)
df.drop(["Name"],axis=1,inplace=True)
df.drop(["Parch"],axis=1,inplace=True)
df.drop(["SibSp"],axis=1,inplace=True)
```

```
x = df.drop("Survived", axis=1)
y = df["Survived"]
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
```

```
model = LogisticRegression(max_iter=5000)
model.fit(x_train,y_train)
p_predict = model.predict(x_test)
```

```
print("The accuracy is", round(accuracy_score(p_predict, y_test) * 100,2))
```

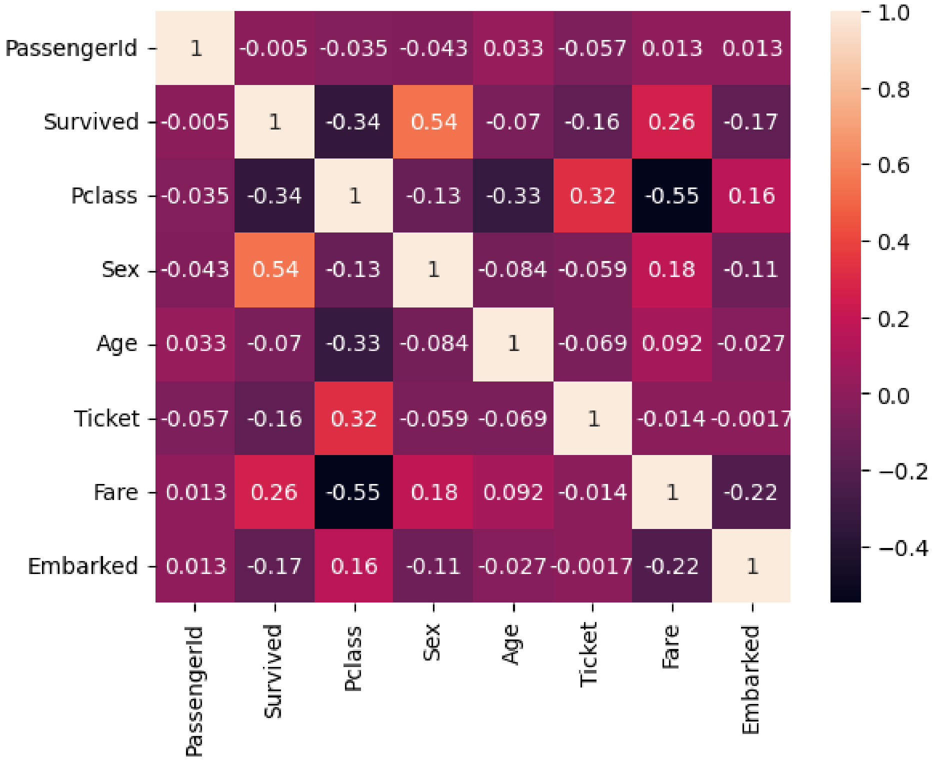
```
     The accuracy is 83.58
```

Data Visualization

```
import matplotlib.pyplot as plt
import seaborn as sb
```

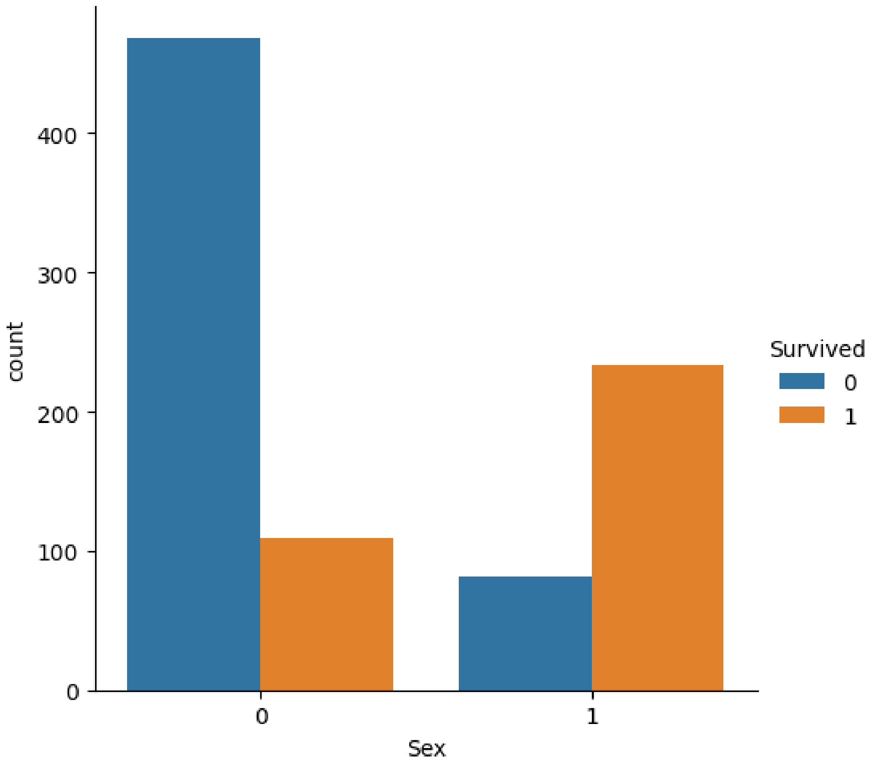
```
corr=df.corr()
sb.heatmap(corr,annot=True)
```

```
<Axes: >
```



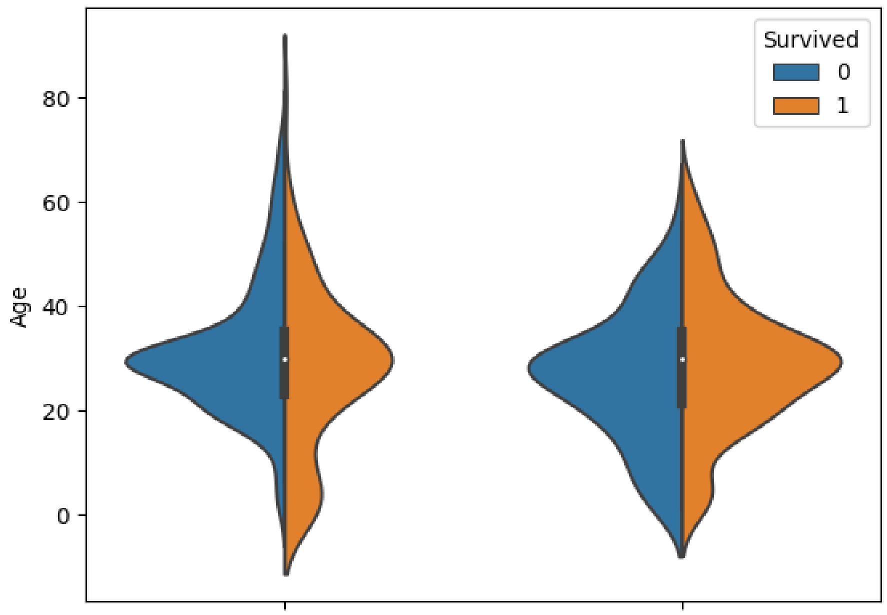
```
sb.catplot(x ="Sex", hue ="Survived",
kind ="count", data = df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7d073bedbaf0>
```



```
sb.violinplot(x ="Sex", y ="Age", hue ="Survived",
data = df, split = True)
```

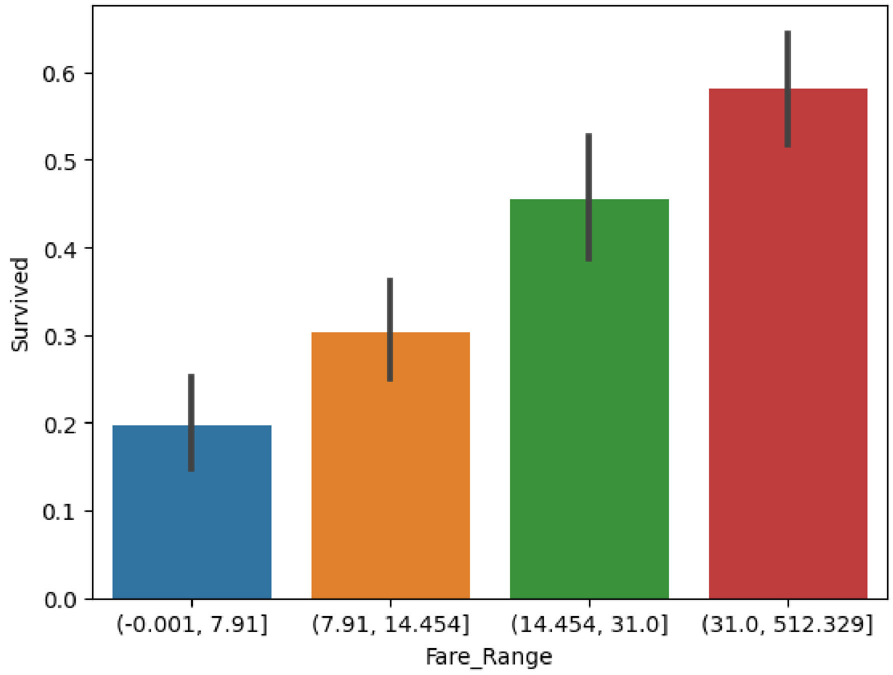
```
<Axes: xlabel='Sex', ylabel='Age'>
```



```
df['Fare_Range'] = pd.qcut(df['Fare'], 4)

sb.barplot(x ='Fare_Range', y ='Survived',
data = df)
```

```
<Axes: xlabel='Fare_Range', ylabel='Survived'>
```


Survived
(-0.001, 7.91]
(7.91, 14.454]
(14.454, 31.0]
(31.0, 512.329]
Fare_Range