Experiment No. 2
Analyze the Titanic Survival Dataset and apply appropriate regression technique
Date of Performance: 02/08/2023
Date of Submission: 10/08/2023

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:

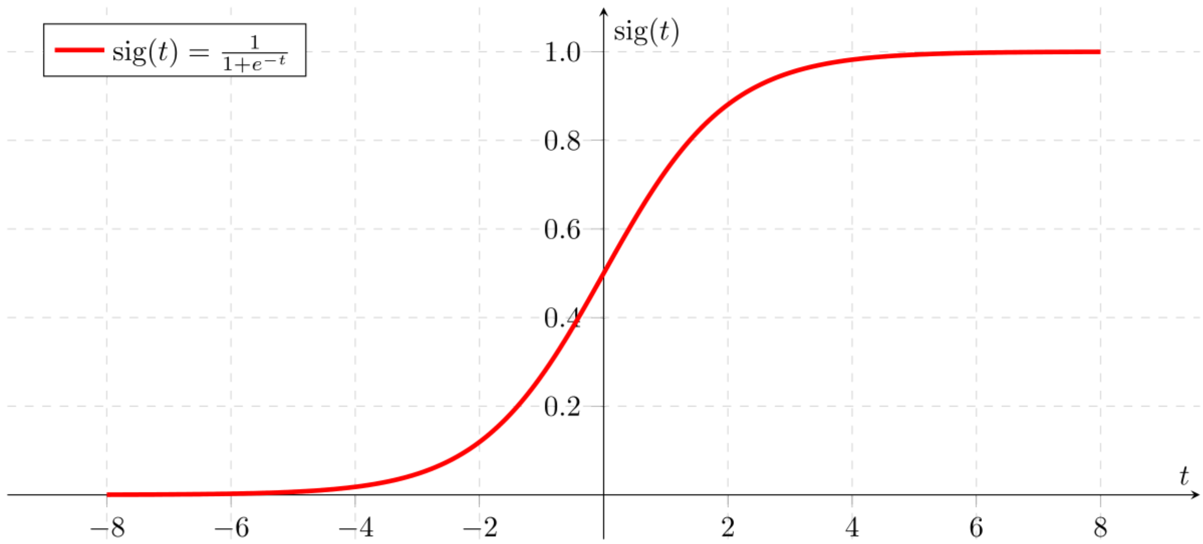
Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical and is binary in nature. In order to perform binary classification the logistic regression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

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
train = pd.read_csv("train.csv")
```

```
train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2

```
train.count()
```

```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            714
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          204
Embarked       889
dtype: int64
```

```
train[train['Sex'].str.match("female")].count()
```

```
PassengerId    314
Survived       314
Pclass         314
Name           314
Sex            314
Age            261
SibSp          314
Parch          314
Ticket         314
Fare           314
Cabin           97
Embarked       312
dtype: int64
```

```
train[train['Sex'].str.match("male")].count()
```

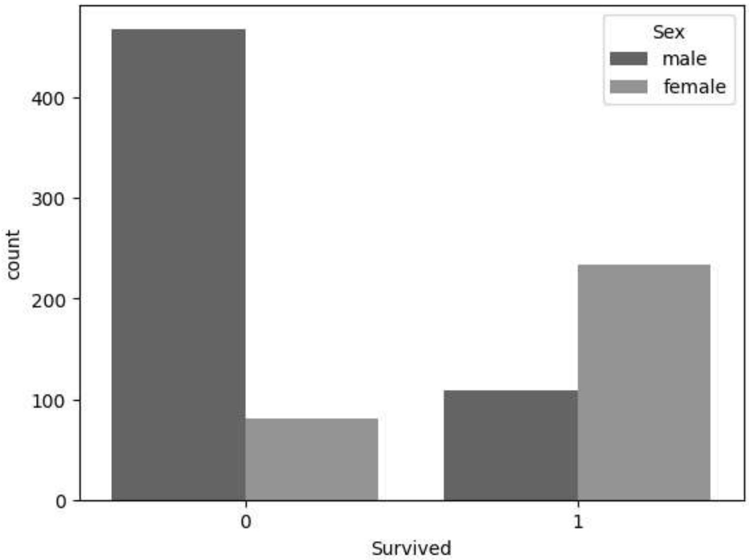
```
PassengerId    577
Survived       577
Pclass         577
Name           577
Sex            577
Age            453
SibSp          577
Parch          577
Ticket         577
Fare           577
Cabin          107
Embarked       577
dtype: int64
```

```
sns.countplot(x='Survived', hue='Pclass', data=train)
```

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

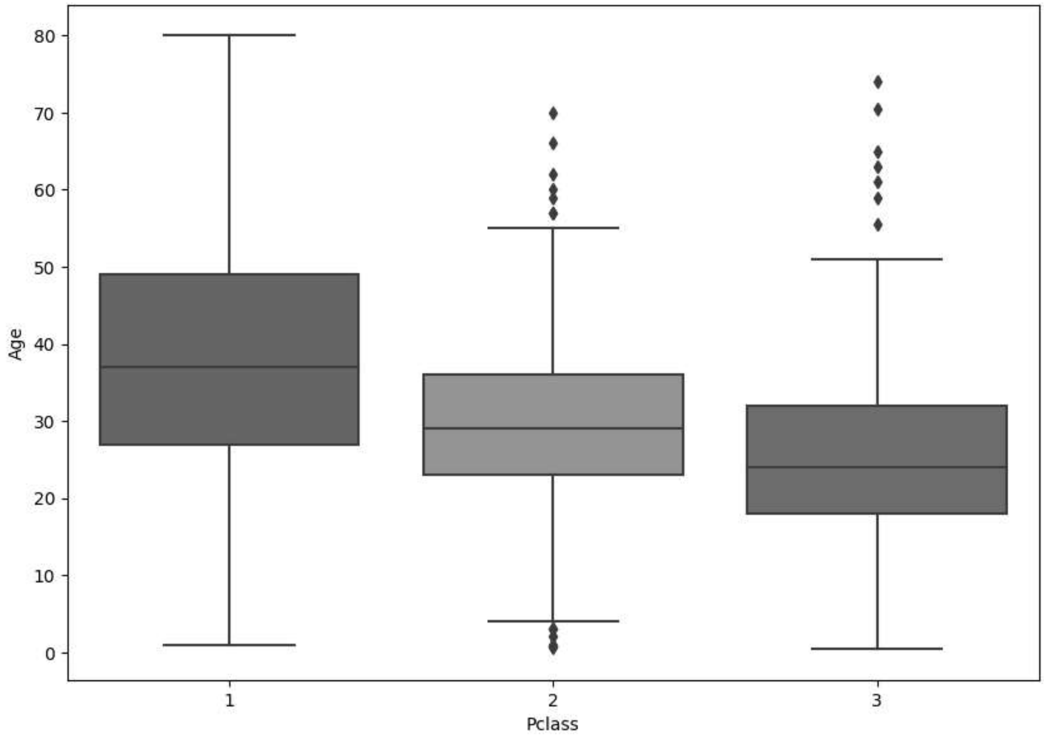

```
sns.countplot(x='Survived', hue='Sex', data=train)
```

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




```
plt.figure(figsize=(10,7))
sns.boxplot(x='Pclass',y='Age',data=train)
```

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




```
def add_age(cols):
    Age = cols[0]
```

```
    Pclass = cols[1]
    if pd.isnull(Age):
        return int(train[train["Pclass"] == Pclass]["Age"].mean())
    else:
        return Age
```

```
train["Age"] = train[["Age", "Pclass"]].apply(add_age,axis=1)
```

```
train.drop("Cabin",inplace=True,axis=1)
```

```
train.dropna(inplace=True)
```

```
pd.get_dummies(train["Sex"])
```

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1
...	...	...
886	0	1
887	1	0
888	1	0
889	0	1
890	0	1

889 rows × 2 columns

```
sex = pd.get_dummies(train["Sex"],drop_first=True)
```

```
embarked = pd.get_dummies(train["Embarked"],drop_first=True)
embarked = pd.get_dummies(train["Pclass"],drop_first=True)
```

```
train.drop(["PassengerId","Pclass","Name","Sex","Ticket","Embarked"],axis=1,inplace=True)
```

```
X = train.drop("Survived",axis=1)
y = train["Survived"]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 101)
```

```
from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
predictions = logmodel.predict(X_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.67      0.94      0.78       163
           1       0.75      0.26      0.39       104

    accuracy                           0.68       267
   macro avg       0.71      0.60      0.58       267
weighted avg       0.70      0.68      0.63       267
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, predictions)
```

```
array([[154,   9],
       [ 77,  27]])
```

✓ 0s completed at 8:28 PM

Conclusion:

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

   Sex:

   Historical records from the Titanic disaster show a strong bias towards saving women first. Including 'Sex' as a feature can capture this significant factor in survival.

   Age:

   Age can be an important factor in survival as children and elderly passengers may have been given priority during the evacuation. Additionally, the ability to navigate the ship or lifeboats could be age-dependent.

   Pclass (Passenger Class):

   Passenger class is often used as a proxy for socio-economic status. It is reasonable to assume that passengers in higher classes had better access to lifeboats and were more likely to survive.

   Fare:

   Fare paid might be correlated with passenger class and, therefore, socio-economic status. It could be an indicator of the passenger's priority for access to lifeboats and safety.

   SibSp (Number of Siblings/Spouses) and Parch (Number of Parents/Children):

   Family size could influence survival decisions. Passengers might have tried to stay together during the evacuation, or individuals with more family members might have been given priority.

   Embarked (Port of Embarkation):

   While the correlation may not be as strong as some other features, the port of embarkation could still be a factor. It might be associated with passenger demographics and their position on the ship.

   Cabin (Cabin Number):

   Cabin location on the ship might have influenced survival. Passengers in cabins closer to lifeboats or above the waterline may have had better chances.

   Ticket (Ticket Number):

   Although not always included due to its complexity, the ticket number might contain some information about the passenger's location on the ship or other relevant details.

Name:

While the name itself may not be a direct predictor, extracting titles (e.g., Mr., Mrs., Miss) from names could provide additional information about the passenger's gender and marital status, which might be relevant to survival.

2. Comment on the accuracy obtained.

An accuracy rate of 70% signifies that the model's predictions are in agreement with the real outcomes found in the dataset. This metric assesses overall correctness.

Precision, recall, and F1-score metrics offer supplementary insights into the model's performance for each class. These metrics delve deeper into the model's ability to distinguish between positive and negative outcomes, highlighting aspects like false positives, false negatives, and the balance between precision and recall