Assignment No: 5

Aim/Problem Statement:- Data Analytics II

Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

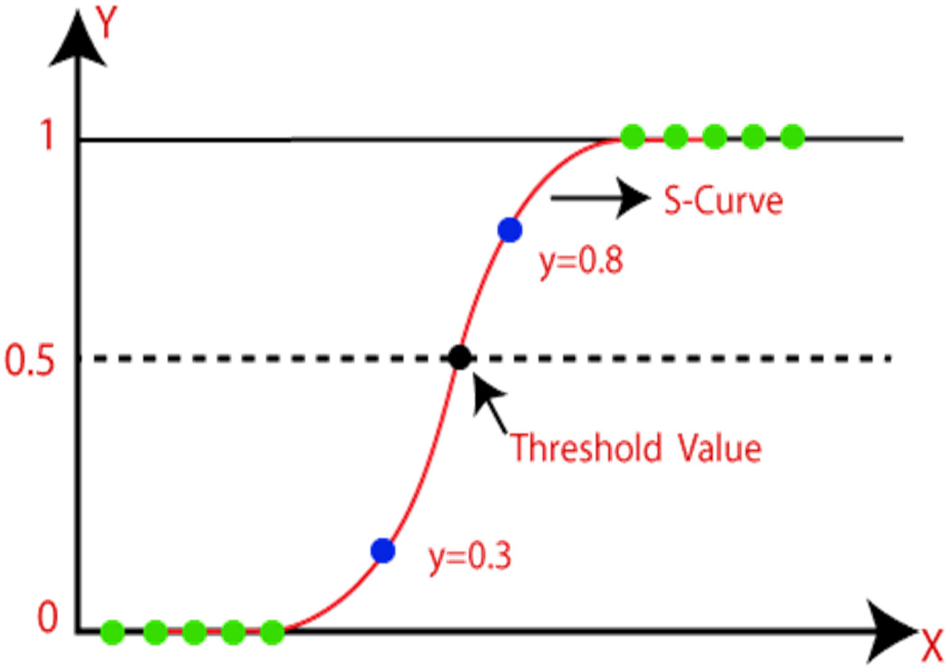
Theory:-

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



Note: *Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.*

Logistic Function (Sigmoid Function):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

Logistic Regression Equation:

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}$$; 0 for y= 0, and infinity for y=1

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

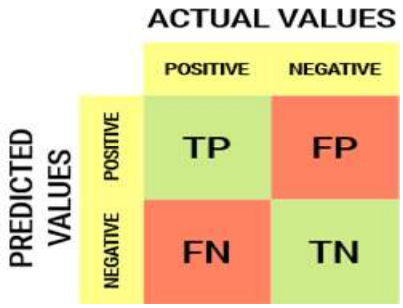
Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.
For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:



The target variable has two values: **Positive** or **Negative**
The **columns** represent the **actual values** of the target variable
The **rows** represent the **predicted values** of the target variable

True Positive (TP)
The predicted value matches the actual value
The actual value was positive and the model predicted a positive value

True Negative (TN)
The predicted value matches the actual value
The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error
The predicted value was falsely predicted
The actual value was negative but the model predicted a positive value
Also known as the **Type 1 error**

False Negative (FN) – Type 2 error
The predicted value was falsely predicted
The actual value was positive but the model predicted a negative value
Also known as the **Type 2 error**

Accuracy is calculated as below

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision tells us how many of the correctly predicted cases actually turned out to be positive. Here's how to calculate Precision:

$$Precision = \frac{TP}{TP + FP}$$

This would determine whether our model is reliable or not.
Recall tells us how many of the actual positive cases we were able to predict correctly with our model.
And here's how we can calculate Recall:

$$Recall = \frac{TP}{TP + FN}$$

Algorithm:-

Step 1: Download the data set of Social_Network_Ads
(https://www.kaggle.com/)

Step 2: Importing Libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Step 3: Importing Data

```
dataset = pd.read_csv('Social_Network_Ads.csv')
#select only age and salary as the features
x = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

Step 4: Perform splitting for training and testing. We will take 75% of the data for training, and test on the remaining data

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Step 5: Scale the features to avoid variation and let the features follow a normal distribution

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Step 6:Fit the Model

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
```

Step 7: Predict the labels of test data

```
y_pred = classifier.predict(X_test)
```

Step 8: Evaluate performance of the model

```
From sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

Step 9: Evaluate Accuracy depending on Confusion Matrix

```
#Accuray=(TN+TP)/Total
```

Step 10: Evaluate error rate

```
#Error_rate=(FN+FP)/Total
```

Step 11: Compute Precision and Recall

```
cl_report=classification_report(y_test,y_pred)

cl_report
```

Input: Social_Network_Ads.csv dataset

Output: Classification using Logistic Regression, Computed Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Conclusion:- Implemented logistic regression to perform classification on Social_Network_Ads.csv dataset using python

Questions:

1) What is logistic regression
2) How it is different from linear regression
3) What are the types of logistic regression
4) What are the limitations of logistic regression
5) List application where logistic regression can be applied