120A3051
Shreya Idate
Batch: E3

EXPERIMENT NO. 7

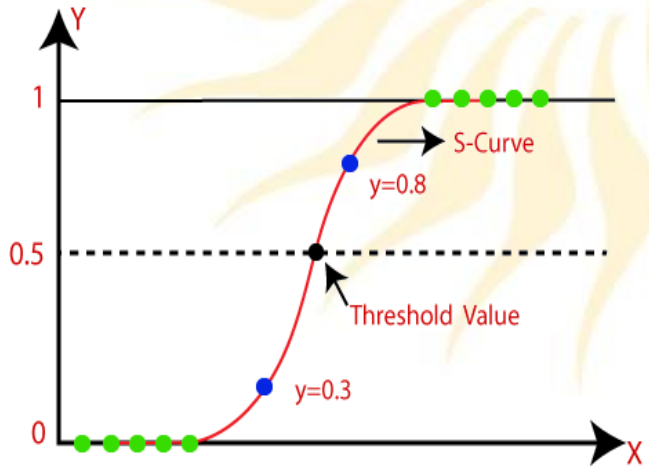
Aim: Regression Analysis

a. Perform Logistic Regression to find out relation between variables.
b. Apply regression Model techniques to predict the data on above dataset

Theory:

There are various kinds of regression techniques available to make predictions. These techniques are mostly driven by three metrics (number of independent variables, type of dependent variables and shape of regression line).
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.
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. It 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:



In Logistic regression, the independent variables should not be correlated with each other i.e. no multi collinearity. However, we have the options to include interaction effects of categorical variables in the analysis and in the model.
If the values of dependent variable is ordinal, then it is called as Ordinal logistic regression
If dependent variable is multi class then it is known as Multinomial Logistic regression.

Program:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LogisticRegression
```

```
x = np.arange(10).reshape(-1, 1)
y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

```
x
```

```
array([[0],
       [1],
       [2],
       [3],
       [4],
       [5],
       [6],
       [7],
       [8],
       [9]])
```

```
y
```

```
array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

```
model = LogisticRegression(solver='liblinear', random_state=0)
```

```
model.fit(x, y)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=0, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

```
model.classes_
```

```
array([0, 1])
```

```
model.intercept_
```

```
array([-1.04608067])
```

```
model.coef_
```

```
array([[ 0.51491375]])
```

```
model.predict_proba(x)
```

```
array([[ 0.74002157,  0.25997843],
       [ 0.62975524,  0.37024476],
       [ 0.5040632 ,  0.4959368 ],
       [ 0.37785549,  0.62214451],
       [ 0.26628093,  0.73371907],
       [ 0.17821501,  0.82178499],
       [ 0.11472079,  0.88527921],
       [ 0.07186982,  0.92813018],
       [ 0.04422513,  0.95577487],
       [ 0.02690569,  0.97309431]])
```

```
model.predict(x)
```

```
array([0, 0, 0, 1, 1, 1, 1, 1, 1, 1])
```

```
model.score(x, y)
```

```
0.90000000000000002
```

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y, model.predict(x)))
```

```
             precision    recall  f1-score   support

          0       1.00      0.75      0.86         4
          1       0.86      1.00      0.92         6

avg / total       0.91      0.90      0.90        10
```

Conclusion: Successfully performed logistic regression.