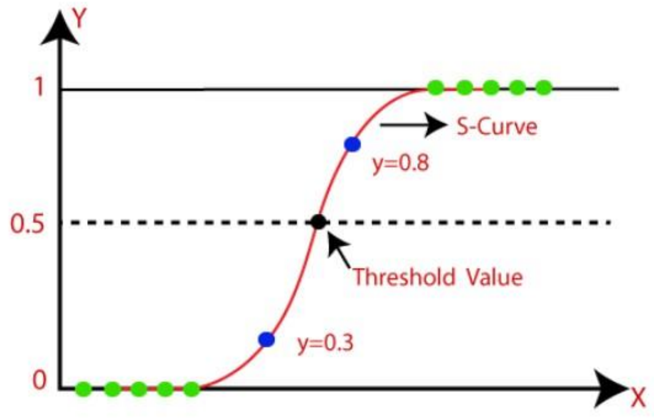
Machine Learning
Experiment 2

SAP ID: 60004200107 Name: Kartik Jolapara
Division: B Batch: B1

AIM
To implement Logistic Regression.

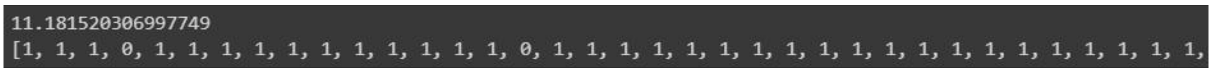
THEORY
Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. It predicts the output of a categorical dependent variable. 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 like 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.
It is a significant machine learning algorithm because it can 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 –



CODE import
pandas as pd import

```
math as m import
random as r import
numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('pima-indians-diabetes.csv')

x = df.iloc[ : , 0]
y = df.iloc[ : , -
1]

w = 1 alpha
= 0.01

epoch = 0 while
epoch <= 1000:
  sigmoidal_func = list(map(lambda x1 : (1/(1 + m.exp((-1) * (x1 * w)))), x))
sumation = list(map(lambda y1, y2, x1 : (y1 - y2) * x1, y, sigmoidal_func,
x))   total = sum(sumation)   gradient =  alpha * total   w += gradient   epoch
+= 1
print(w)

y_pred = list(map(lambda x1 : 1 if (1/(1 + m.exp((-1) * (x1 * w)))) > 0.5 else 0, x))
print(y_pred)
```

OUTPUT



```
11.181520306997749
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

CONCLUSION

Thus, we have successfully implemented Logistic Regression.