Register No: 21MIS1044

Name: Srinivasan JP

Logistic regression

In [ ]:
```
import numpy
from sklearn import linear_model
```

Creating the X and Y dataset here x is independent variable and y is dependent variable with only 2 classes

In [ ]:
```
X = numpy.array([3.78, 2.44, 2.09, 0.14, 1.72, 1.65, 4.92, 4.37, 4.96, 4.52, 3.69, 5.8
y = numpy.array([0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

Using the sklearns predefind modle the x and y is trained

In [ ]:
```
logr = linear_model.LogisticRegression()
logr.fit(X, y)
```

Out[ ]:
```
LogisticRegression()
```

The logistic function (also known as the sigmoid function) is used to transform the linear combination of features into probabilities:

$P(y=1 \mid x) = 1 / (1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)})$

In [ ]:
```
def logit2prob(logr, X):
    log_odds = logr.coef_ * X + logr.intercept_
    odds = numpy.exp(log_odds)
    probability = odds / (1 + odds)
    return probability
```

In [ ]:
```
odds = logit2prob(logr, X)
numpy.set_printoptions(precision=2)
for i in odds:
    if(i<0.5):
        print(i*100,"False", sep="---->")
    else:
        print(i*100,"True", sep="---->")
```

[60.75]---->True
[19.27]---->False
[12.77]---->False
[0.96]---->False
[8.04]---->False
[7.34]---->False
[88.36]---->True
[77.9]---->True
[88.92]---->True
[81.29]---->True
[57.72]---->True
[96.66]---->True