**LOGISTIC REGRESSION.**

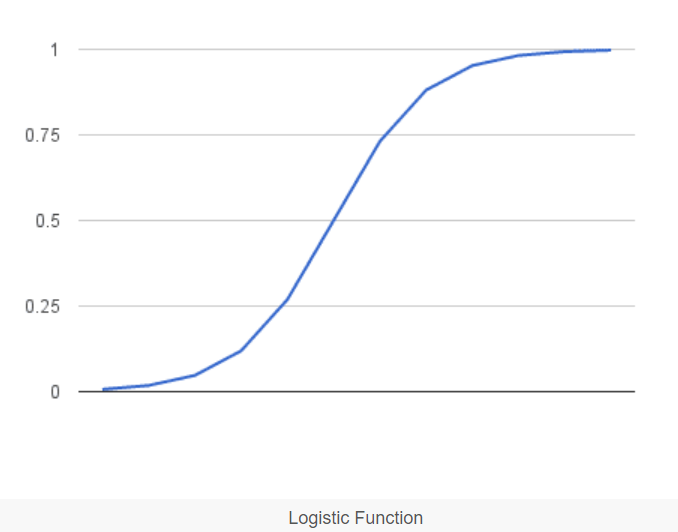
It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable ie. it is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance of belonging to a given class or not.

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 named for the function used at the core of the method, the **logistic function** also called the **sigmoid function**. It’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.



EXAMPLE CODE.

This code uses the diabetes dataset which has been attached to this file.

1. **Importing Libraries.**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

- Imports necessary libraries for data manipulation, visualization, logistic regression, and model evaluation.

2. **Loading and Exploring Data.**

data = pd.read\_csv('diabetes.csv')

data.head()

- Loads the diabetes dataset from a CSV file into a Pandas DataFrame and displays the first few rows.

3. **Transposing Data for Better Visualization.**

data.head().transpose()

- Transposes the data for better visualization by displaying columns as rows and vice versa.

4. **Descriptive Statistics of the Data.**

data.describe()

- Provides descriptive statistics of the numerical columns in the dataset, including count, mean, std (standard deviation), min, 25th percentile (Q1), median (50th percentile or Q2), 75th percentile (Q3), and max.

5. **Feature and Target Separation.**

X = data.drop("Outcome", axis=1)

y = data[["Outcome"]]

- Separates the features (`X`) and the target variable (`y`).

6. **Train-Test Split.**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=7)

- Splits the data into training and testing sets using `train\_test\_split`. 30% of the data is reserved for testing, and `random\_state` ensures reproducibility.

7. **Logistic Regression Model Training.**

model = LogisticRegression()

model.fit(X\_train, y\_train)

- Initializes a logistic regression model and trains it on the training data.

8. **Model Prediction and Evaluation.**

y\_predict = model.predict(X\_test)

model\_score = model.score(X\_test, y\_test)

- Predicts the target variable on the test set and calculates the accuracy score of the model on the test set.

9. **Printing Model Evaluation Metrics.**

print(model\_score)

print(metrics.confusion\_matrix(y\_test, y\_predict))

- Prints the accuracy score and the confusion matrix. The confusion matrix is a table that describes the performance of a classification model, showing the number of true positives, true negatives, false positives, and false negatives.