

## Logistics Regression

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.

A logistic regression model predicts a **dependent data variable** by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college. These binary outcomes allow straightforward decisions between two alternatives.

Probability is the likelihood or chance of event occurring. The number of way of achieving success. The total number of possible outcome.

In binary classification, the dependent variable (also called the target variable) can take only two possible values, typically represented as 0 and 1. The logistic regression model predicts the probability that the dependent variable will be 1, given the values of the independent variables.

The logistic regression model equation is as follows:

$$p = 1 / (1 + e^{(-z)})$$

where:

- p is the probability of the dependent variable being 1,
- z is the linear combination of the independent variables and their corresponding coefficients.

$$y = \frac{e^{(b_0 + b_1X)}}{1 + e^{(b_0 + b_1X)}}$$

## Create and load dataset

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

```
dataset = pd.read_csv('LR.csv')
X=dataset.iloc[:,[0,1]].values
y=dataset.iloc[:,2].values
```

```
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)
```

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

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

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

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

