# Logistic Regression

# Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems.

It is a predictive analysis algorithm and based on the concept of probability.

Some of the examples of classification problems are

- Classifying Emails: spam or not spam,
- Classifying Online transactions: Fraud or not Fraud,
- Classifying Tumor: Malignant or Benign.

Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

# Types of Logistic Regression

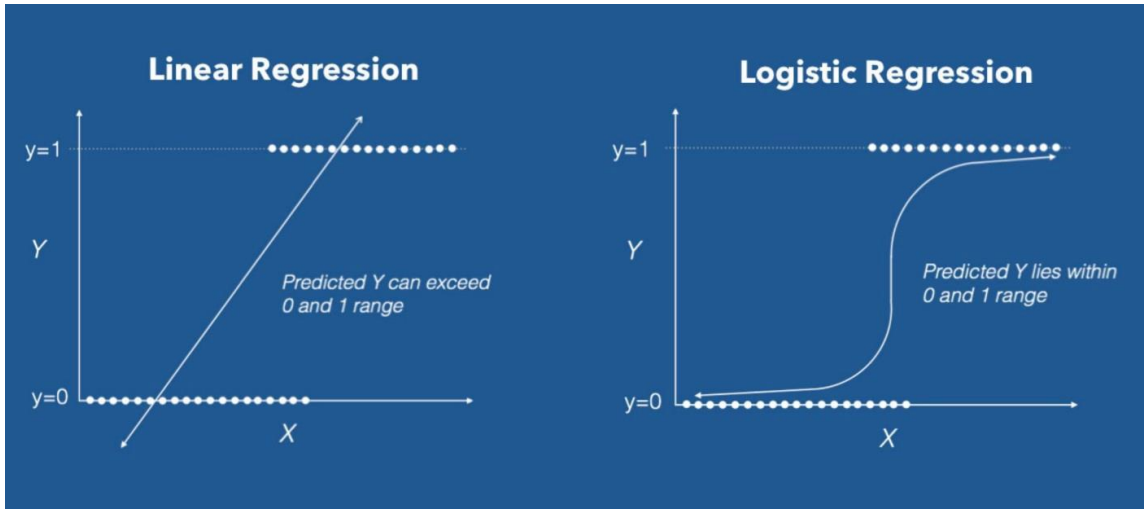
1. Binary Classification
   - Classifying Emails: spam or not spam,
   - Classifying Online transactions: Fraud or not Fraud,
   - Classifying Tumor: Malignant or Benign.

   Instead of our output vector y being a continuous range of values, it will only be 0 or 1.

   $y \in \{0,1\}$

1. Multi-Class Classification
   - Handwritten digit recognition
   - Diabetes Detection: No diabetes, Type-1 Diabetes, Type-2 Diabetes

# Linear Regression Vs. Logistic Regression




Our hypothesis should satisfy: $0 \leq h\theta(x) \leq 1$

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1.

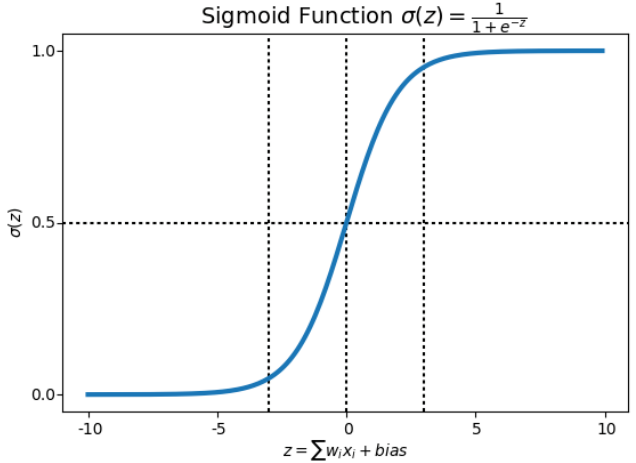
Linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

# Sigmoid Function

In order to map predicted values to probabilities, we use the Sigmoid function. Also, known as Logistic Function.

The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$z = 0, e^0 = 1 \Rightarrow g(z) = 1/2$$
$$z \to \infty, e^{-\infty} \to 0 \Rightarrow g(z) = 1$$
$$z \to -\infty, e^{\infty} \to \infty \Rightarrow g(z) = 0$$




$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\sigma(z) \geq 0.5 \text{ when } z \geq 0$$

# Hypothesis Representation

When using linear regression we used a formula of the hypothesis i.e.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n$$

For logistic regression we are going to modify it by applying sigmoid function over it i.e.

$$h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + ... + \theta_n x_n)$$

$$h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+...+\theta_n x_n)}}$$

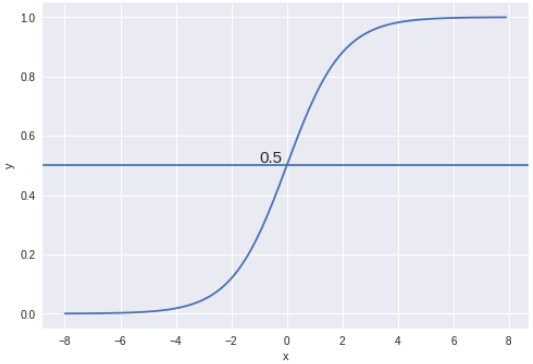
# Decision Boundary

# Decision Boundary

We expect our classifier to give us a set of outputs or classes based on probability when we pass the inputs through a prediction function and returns a probability score between 0 and 1.

For Example, We have 2 classes, let's take them like cats and dogs(1 — dog , 0 — cats). We basically decide with a threshold value above which we classify values into Class 1 and of the value goes below the threshold then we classify it in Class 2.

As shown in the above graph we have chosen the threshold as 0.5, if the prediction function returned a value of 0.7 then we would classify this observation as Class 1(DOG). If our prediction returned a value of 0.2 then we would classify the observation as Class 2(CAT).




$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$

$$h_\theta(x) < 0.5 \rightarrow y = 0$$

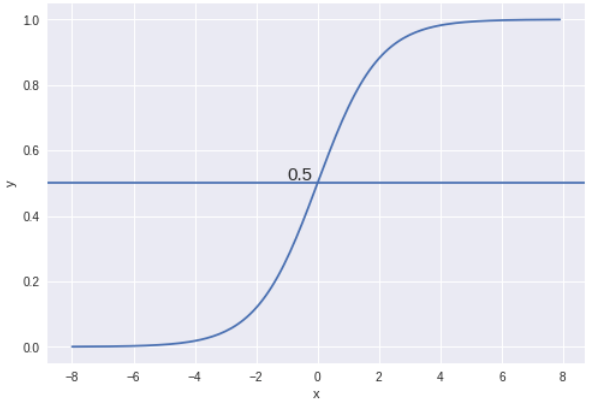
$$\theta^T x = \begin{pmatrix} \theta_0 \\ \theta_1 \\ . \\ . \\ \theta_n \end{pmatrix}^T \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ . \\ . \\ x_n \end{pmatrix}$$

$$\theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$
$$h_\theta(x) < 0.5 \rightarrow y = 0$$

$$h_\theta(x) = \sigma\left(\theta^T x\right) \geq 0.5 \text{ when } \theta^T x \geq 0$$

$$\theta^T x \geq 0 \Rightarrow y = 1$$
$$\theta^T x < 0 \Rightarrow y = 0$$



The decision boundary is the line that separates the area where y = 0 and where y = 1. It is created by our hypothesis function.

In this case, our decision boundary is a straight vertical line placed on the graph where $x_1 = 5$, and everything to the left of that denotes y = 1, while everything to the right denotes y = 0.

$$\theta = \begin{bmatrix} 5 \\ -1 \\ 0 \end{bmatrix}$$

$$y = 1 \; if \; 5 + (-1)x_1 + 0x_2 \geq 0$$

$$5 - x_1 \geq 0$$

$$-x_1 \geq -5$$

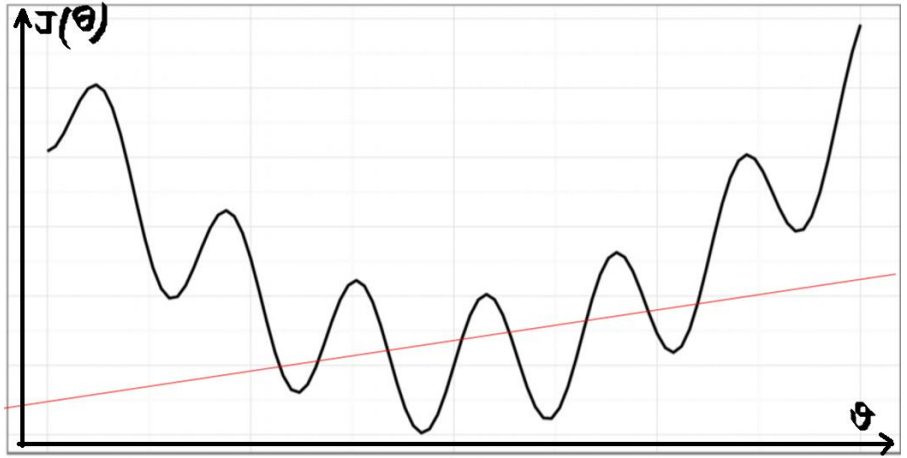
$$x_1 \leq 5$$

# Cost Function

# Why can’t we use Linear Regression Cost function?

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Cost function of Linear Regression




If we try to use the cost function of the linear regression in ‘Logistic Regression’ then it would be of no use as it would end up being a non-convex function with many local minimums, in which it would be very difficult to minimize the cost value and find the global minimum.

# Logistic Regression Cost Function

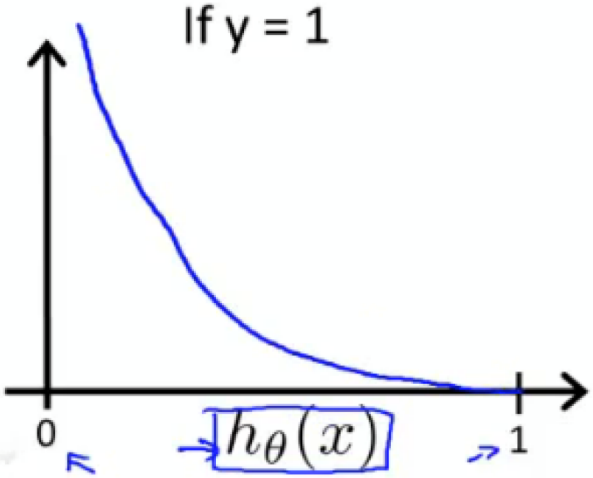
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

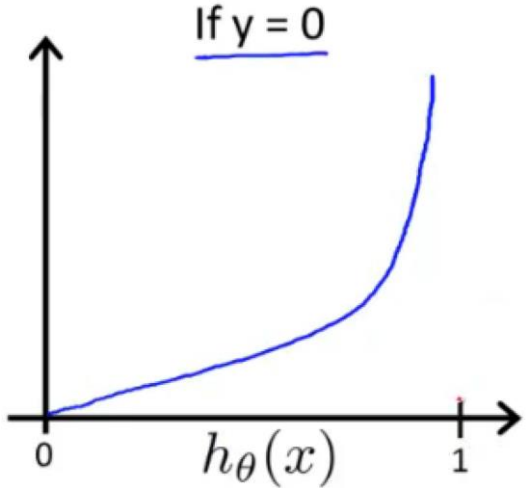
$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if y} = 1$$

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if y} = 0$$

# Understanding Cost Function

| Actual y | Predicted Class | Cost fn | Cost Value |
|---|---|---|---|
| 1 | 1 (h(x) is close to 1) | -log(h(x)) | ~0 |
| 1 | 0 (h(x) is close 0) | -log(h(x)) | ~ Infinity |
| 0 | 0 (h(x) is close 0) | -log(1-h(x)) | ~0 |
| 0 | 1 (h(x) is close 1) | -log(1-h(x)) | ~Infinity |




$\text{Cost}(h_\theta(x), y) = 0$ if $h_\theta(x) = y$

$\text{Cost}(h_\theta(x), y) \to \infty$ if $y = 0$ and $h_\theta(x) \to 1$

$\text{Cost}(h_\theta(x), y) \to \infty$ if $y = 1$ and $h_\theta(x) \to 0$

$$\text{Cost}(h_\theta(x), y) = -y \ \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

$$\text{Cost}(h_\theta(x), y) = -y \ \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

**VECTORIZED IMPLEMENTATION**

$$J(\theta) = \frac{1}{m} \left( - y^T \log(h) - (1 - y)^T \log(1 - h) \right)$$

$$\text{where } \ h = \sigma(X\theta)$$

# Using Gradient Descent

Remember that the general form of gradient descent is:

$$
\begin{aligned}
&Repeat\ \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&Repeat\ \{ \\
&\quad \theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \\
&\}
\end{aligned}
$$

The vectorized version;

$$\nabla J(\theta) = \frac{1}{m} \cdot X^T \cdot \left( g\left( X \cdot \theta \right) - \vec{y} \right)$$

# Practice Question

Consider the dataset given below, which categorizes an article either as *Technical (Class 1)* or *Non-Technical (Class 0)* based on the time spent in *reading (in Hours)* and the *number of sentences (in multiples of 1000)* in that article.

| Time (Hours) | Sentences (In multiples of 1000) | Article Type |
|---|---|---|
| 2.7 | 2.5 | 0 |
| 3 | 3 | 0 |
| 5.9 | 2.2 | 1 |
| 7.7 | 3.5 | 1 |

# Practice Question

- Using the above data, build a logistic regression model to predict the class of an article using gradient descent method. Assume $learning\ rate = 0.3$. Further, in the first iteration the value of the coefficients is 0, and the *bias* is 1. Use two iterations of the gradient descent process to learn the model parameters.

- Compute the error in prediction.

- Use the above model to predict the article type of an article which requires **`6.2 hours`** of reading time and contains **`3100 sentences`**.

# Working

$$X = \begin{bmatrix} x_0 & x_1 & x_2 \\ 1 & 2.7 & 2.5 \\ 1 & 3 & 3 \\ 1 & 5.9 & 2.2 \\ 1 & 7.7 & 3.5 \end{bmatrix} \qquad \overrightarrow{y} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \qquad \theta = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\theta = \theta - \frac{\alpha}{m}.\, X^T.\left(g(X.\,\theta) - \overrightarrow{y}\right) \qquad \alpha = 0.3$$

## First Iteration

Theta0 = 1, theta1=0,  theta2 = 0

| X0 | X1 | X2 | Y | Y_pred | Y_pred-y | (y_pred-Y)*x1 | (y-pred - y) * x2 |
|---|---|---|---|---|---|---|---|
| 1 | 2.7 | 2.5 | 0 | 0.73 | 0.73 | 1.971 | 1.83 |
| 1 | 3 | 3 | 0 | 0.73 | 0.73 | 2.19 | 2.19 |
| 1 | 5.9 | 2.2 | 1 | 0.73 | -0.27 | -1.59 | -0.59 |
| 1 | 7.7 | 3.5 | 1 | 0.73 | -0.27 | -2.08 | -0.94 |
| | | | | SUM | 0.92 | 0.491 | 2.49 |

# First Iteration

Y_pred = sigmoid ( x0*theta0 + x1* theta1 + x2*theta2)

Updating thetas, using the theta update equation given as:

Theta0 = theta0 – (alpha / m ) * SUM (y_pred - y)

Theta0 = 1 – (0.3/4)(0.92) = 0.931

Theta1 = theta0 – (alpha / m ) * SUM (y_pred - y)*x1

Theta1 = 0 – (0.3/4)(0.491) = -0.04

Theta2 = theta1 - (alpha / m ) * SUM (y_pred - y)*x2

Theta2 = 0 – (0.3/4)(2.491) = -0.19

## Second Iteration

Theta0=0.93, theta1=-0.04, theta2=-0.19

| X0 | X1 | X2 | Y | Y_pred | Y_pred-y | (y_pred-Y)*x1 | (y-pred - y) * x2 |
|---|---|---|---|---|---|---|---|
| 1 | 2.7 | 2.5 | 0 | 0.589 | 0.589 | 1.591 | 1.473 |
| 1 | 3 | 3 | 0 | 0.564 | 0.564 | 1.691 | 1.691 |
| 1 | 5.9 | 2.2 | 1 | 0.573 | -0.427 | -2.518 | -0.939 |
| 1 | 7.7 | 3.5 | 1 | 0.495 | -0.504 | -3.882 | -1.764 |
| | | | | SUM | 0.222 | -3.118 | 0.461 |

# Second Iteration

Updating thetas

Theta0 = theta0 – (alpha / m ) * S (y_pred - y)

Theta0 = 0.93 – (0.3/4)(0.222) = 0.913

Theta1 = theta0 – (alpha / m ) * S (y_pred - y)*x1

Theta1 = -0.04 – (0.3/4)(-3.118) = 0.196

Theta2 = theta1 - (alpha / m ) * S (y_pred - y)*x2

Theta2 = -0.19 – (0.3/4)(0.461) = -0.221

**Computing error in prediction**

| X0 | X1 | X2 | Y | Y_pred | Y_pred-y |
|---|---|---|---|---|---|
| 1 | 2.7 | 2.5 | 0 | 0.708 | 0.708 |
| 1 | 3 | 3 | 0 | 0.697 | 0.697 |
| 1 | 5.9 | 2.2 | 1 | -0.170 | -0.17 |
| 1 | 7.7 | 3.5 | 1 | -1.161 | -0.16 |
| | | | | SUM | 1.074 |

# c)

Use the above model to predict the article type of an article which requires **`6.2 hours`** of reading time and contains **`3100 sentences`**.

X0=1, x1=6.2, x2=3.1

Sigmoid(1*0.913 + 6.2*0.196 + 3.1*-0.221) = 0.808 > 0.5 which means Article Type = 1