This article was published as a part of the Data Science Blogathon

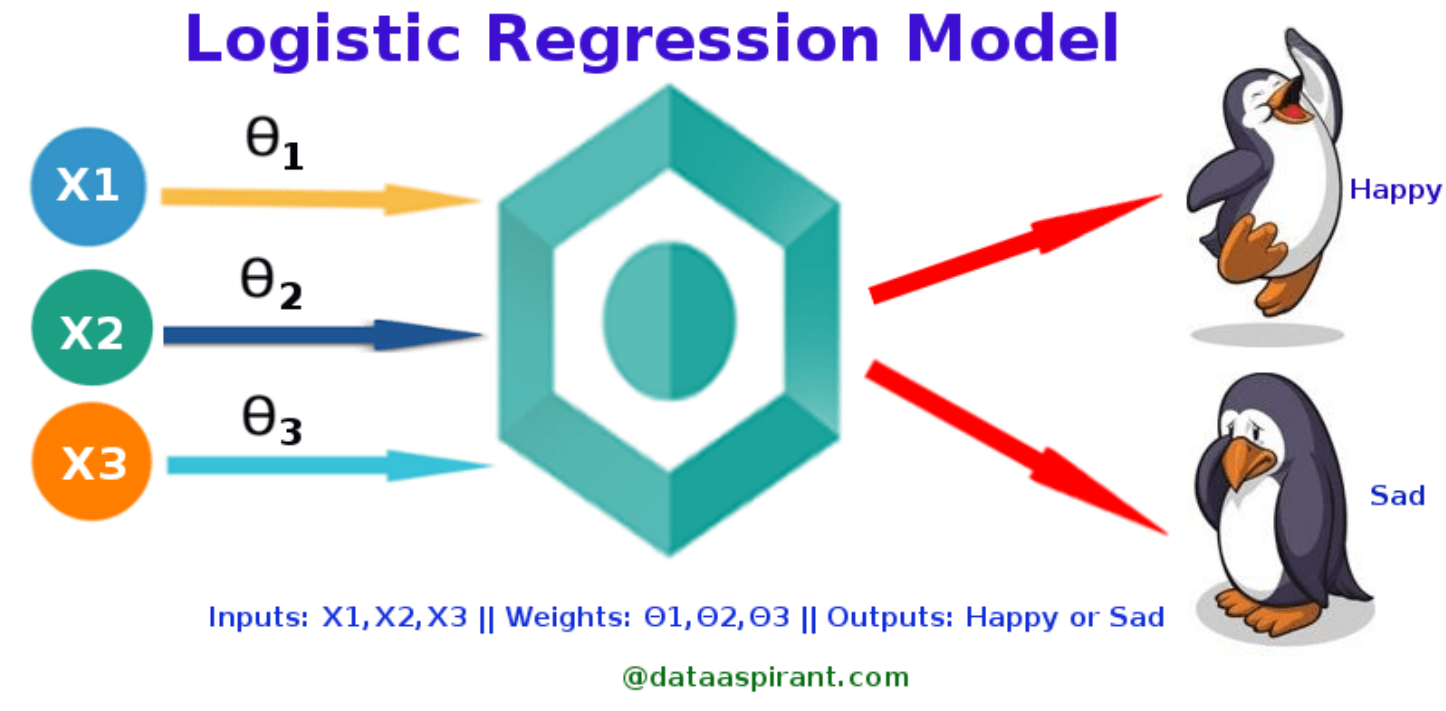


Image Source: dataaspirant.com

Introduction

What is Logistic Regression? How is it different from Linear Regression? Why regression word is used here if this is a classification problem? What is the use of MLE in Logistic regression? From where did the Loss function come? How does Gradient Descent work in Logistic Regression? What is an odd's ratio?

Well, these were a few of my doubts when I was learning Logistic Regression. To find the math behind this, I plunged deeper into this topic only to find myself a better understanding of the Logistic Regression model. And in this article, I will try to answer all the doubts you are having right now on this topic. I will tell you the math behind this regression model.

Table of contents

What is Logistic Regression?

> **Logistic regression** is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

I found this definition on google and now we'll try to understand it. Logistic Regression is another statistical analysis method borrowed by Machine Learning. It is used when our dependent variable is dichotomous or binary. It just means a variable that has only 2 outputs, for example, A person will survive this accident or not, The student will pass this exam or not. The outcome can either be yes or no (2 outputs). This regression technique is similar to linear regression and can be used to predict the **Probabilities** for classification problems.

Why do we use Logistic Regression rather than Linear Regression?

If you have this doubt, then you're in the right place, my friend. After reading the definition of logistic regression we now know that it is only used when our dependent variable is binary and in linear regression this dependent variable is continuous.

The second problem is that if we add an outlier in our dataset, the best fit line in linear regression shifts to fit that point.

Now, if we use linear regression to find the best fit line which aims at minimizing the distance between the predicted value and actual value, the line will be like this:

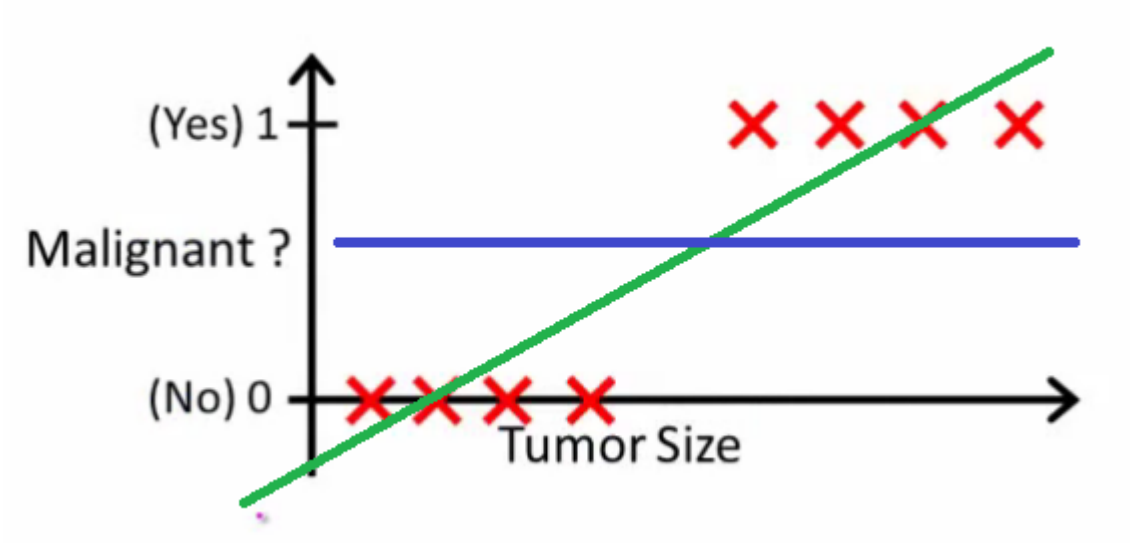


Image Source:[towardsdatascience.com]

Here the threshold value is 0.5, which means if the value of h(x) is greater than 0.5 then we

now this best fit line will shift to that point. Hence the line will be somewhat like this:

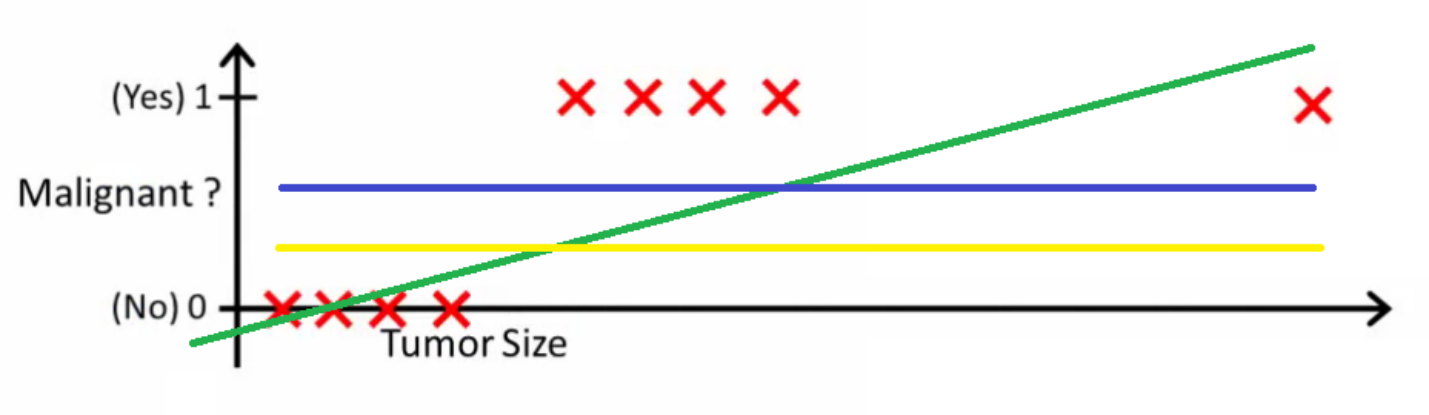


Image Source:[towardsdatascience.com]

Do you see any problem here? The blue line represents the old threshold and the yellow line represents the new threshold which is maybe 0.2 here. To keep our predictions right we had to lower our threshold value. Hence we can say that linear regression is prone to outliers. Now here if h(x) is greater than 0.2 then only this regression will give correct outputs.

Another problem with linear regression is that the predicted values may be out of range. We know that probability can be between 0 and 1, but if we use linear regression this probability may exceed 1 or go below 0.

To overcome these problems we use Logistic Regression, which converts this straight best fit line in linear regression to an S-curve using the sigmoid function, which will always give values between 0 and 1. How does this work and what's the math behind this will be covered in a later section?

If you want to know the difference between logistic regression and linear regression then you refer to this article.

Logistic Function

You must be wondering how logistic regression squeezes the output of linear regression between 0 and 1. If you haven't read my article on Linear Regression then please have a look at it for a better understanding.

Well, there's a little bit of math included behind this and it is pretty interesting trust me.

Let's start by mentioning the formula of logistic function:

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

How similar it is too linear regression? If you haven't read my article on Linear Regression, then please have a look at it for a better understanding.

We all know the equation of the best fit line in linear regression is:

$$y = \beta_0 + \beta_1 x$$

Let's say instead of y we are taking probabilities (P). But there is an issue here, the value of (P) will exceed 1 or go below 0 and we know that range of Probability is (0-1). To overcome this issue we take ***"odds"*** of P:

$$P = \beta_0 + \beta_1 x$$

$$\frac{P}{1-P} = \beta_0 + \beta_1 x$$

Do you think we are done here? No, we are not. We know that odds can always be positive which means the range will always be (0,+∞). Odds are nothing but the ratio of the probability of success and probability of failure. Now the question comes out of so many other options to transform this why did we only take ***'odds'***? Because odds are probably the easiest way to do this, that's it.

The problem here is that the range is restricted and we don't want a restricted range because if we do so then our correlation will decrease. By restricting the range we are actually decreasing the number of data points and of course, if we decrease our data points, our correlation will decrease. It is difficult to model a variable that has a restricted range. To control this we take the ***log of odds*** which has a range from (-∞,+∞).

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 x$$

If you understood what I did here then you have done 80% of the maths. Now we just want a function of P because we want to predict probability right? not log of odds. To do so we will multiply by ***exponent*** on both sides and then solve for P.

$$\exp[\log(\frac{p}{1-p})] = \exp(\beta_0 + \beta_1 x)$$

$$e^{\ln[\frac{p}{1-p}]} = e^{(\beta_0 + \beta_1 x)}$$

$$\frac{p}{1-p} = e^{\left(\beta_0 + \beta_1 x\right)}$$

$$p = e^{\left(\beta_0 + \beta_1 x\right)} - pe^{\left(\beta_0 + \beta_1 x\right)}$$

$$p = p[\frac{e^{\left(\beta_0 + \beta_1 x\right)}}{p} - e^{\left(\beta_0 + \beta_1 x\right)}]$$

$$1 = \frac{e^{\left(\beta_0 + \beta_1 x\right)}}{p} - e^{\left(\beta_0 + \beta_1 x\right)}$$

$$p[1 + e^{\left(\beta_0 + \beta_1 x\right)}] = e^{\left(\beta_0 + \beta_1 x\right)}$$

$$p = \frac{e^{\left(\beta_0 + \beta_1 x\right)}}{1 + e^{\left(\beta_0 + \beta_1 x\right)}}$$

Now dividing by $e^{\left(\beta_0 + \beta_1 x\right)}$, *we will get*

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad \textit{This is our sigmoid function.}$$

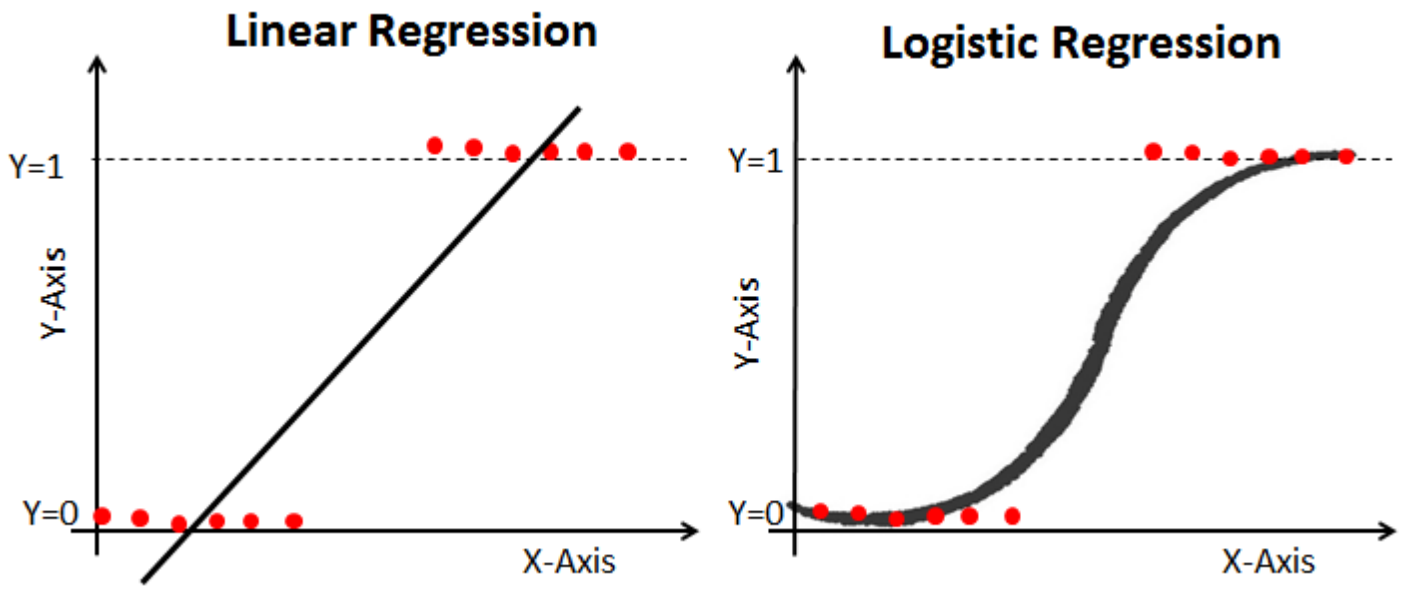


Image Source www.datacamp.com

Cost Function in Logistic Regression

In linear regression, we use the Mean squared error which was the difference between y_predicted and y_actual and this is derived from the maximum likelihood estimator. The graph of the cost function in linear regression is like this:

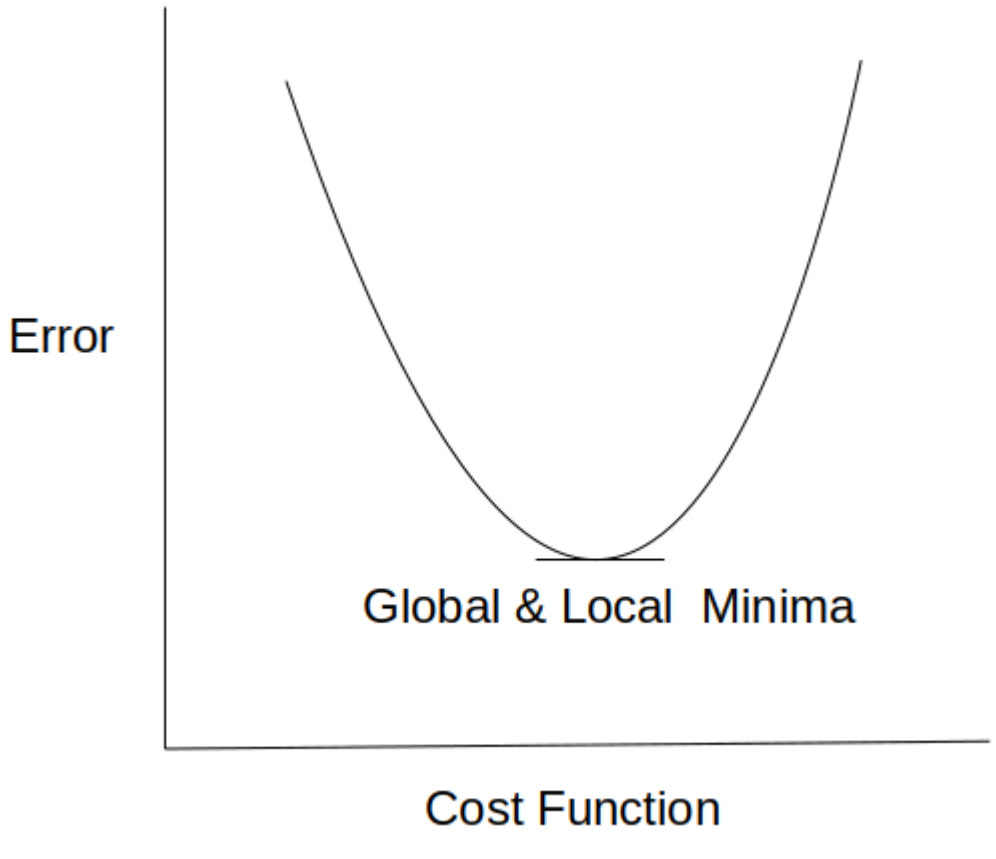
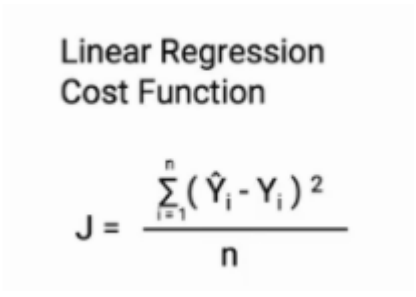
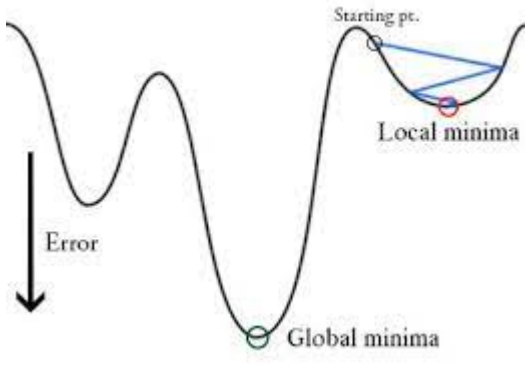


Image Source: https://dchandra.com/



In logistic regression Yi is a non-linear function ($\hat{Y}=1/1+ e^{-z}$). If we use this in the above MSE equation then it will give a non-convex graph with many local minima as shown



The problem here is that this cost function will give results with local minima, which is a big problem because then we'll miss out on our global minima and our error will increase.

In order to solve this problem, we derive a different cost function for logistic regression called ***log loss*** which is also derived from the *maximum likelihood estimation* method.

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -(y_i * \log(\hat{Y}_i) + (1 - y_i) * \log(1 - \hat{Y}_i))$$

In the next section, we'll talk a little bit about the maximum likelihood estimator and what it is used for. We'll also try to see the math behind this log loss function.

What is the use of Maximum Likelihood Estimator?

The main aim of MLE is to find the value of our parameters for which the likelihood function is *maximized*. The likelihood function is nothing but a joint pdf of our sample observations and joint distribution is the multiplication of the *conditional probability* for observing each example given the distribution parameters. In other words, we try to find such that plugging these estimates into the model for P(x), yields a number close to one for people who had a malignant tumor and close to 0 for people who had a benign tumor.

Let's start by defining our likelihood function. We now know that the labels are binary which means they can be either yes/no or pass/fail etc. We can also say we have two outcomes success and failure. This means we can interpret each label as Bernoulli random variable.

> A random experiment whose outcomes are of two types, success S and failure F, occurring with probabilities p and q respectively is called a Bernoulli trial. If for this experiment a random variable X is defined such that it takes value 1 when S occurs and 0 if F occurs, then X follows a Bernoulli Distribution.

$$Y \sim Ber(P)$$

Where P is our sigmoid function

$$P[Y = y|X = x] = \sigma(\theta^T x^i)^y (1 - \sigma(\theta^T x^i)^{1-y}$$

where **σ**(θ^T*x^i) is the sigmoid function. Now for n observations,

$$L(\theta) = \prod_{1}^{n} \sigma(\theta^T x^i)^y (1 - \sigma(\theta^T x^i)^{1-y}$$

We need a value for theta which will maximize this likelihood function. To make our calculations easier we multiply the log on both sides. The function we get is also called the log-likelihood function or sum of the log conditional probability

$$log(L(\theta)) = \sum_{1}^{n} y * log[\sigma(\theta^T x^i)] + (1 - y) * log[(1 - \sigma(\theta^T x^i)]$$

In machine learning, it is conventional to minimize a loss(error) function via gradient descent,

so that we use gradient descent. We'll talk more about gradient descent in a later section and then you'll have more clarity. Also, remember,

> *max[log(x)] = min[-log(x)]*

The negative of this function is our ***cost function*** and what do we want with our cost function? That it should have a minimum value. It is common practice to minimize a cost function for optimization problems; therefore, we can invert the function so that we minimize the negative log-likelihood (NLL). So in logistic regression, our cost function is:

$$-log[L(\theta)] = -\sum_{1}^{n} y * log[\sigma(\theta^T x^i)] + (1-y) * log(1-\sigma(\theta^T x^i)$$

Here y represents the actual class and log(**σ(**θ^T*x^i**))** is the probability of that class.

- p(y) is the probability of 1.
- 1-p(y) is the probability of 0.

Let's see what will be the graph of cost function when y=1 and y=0

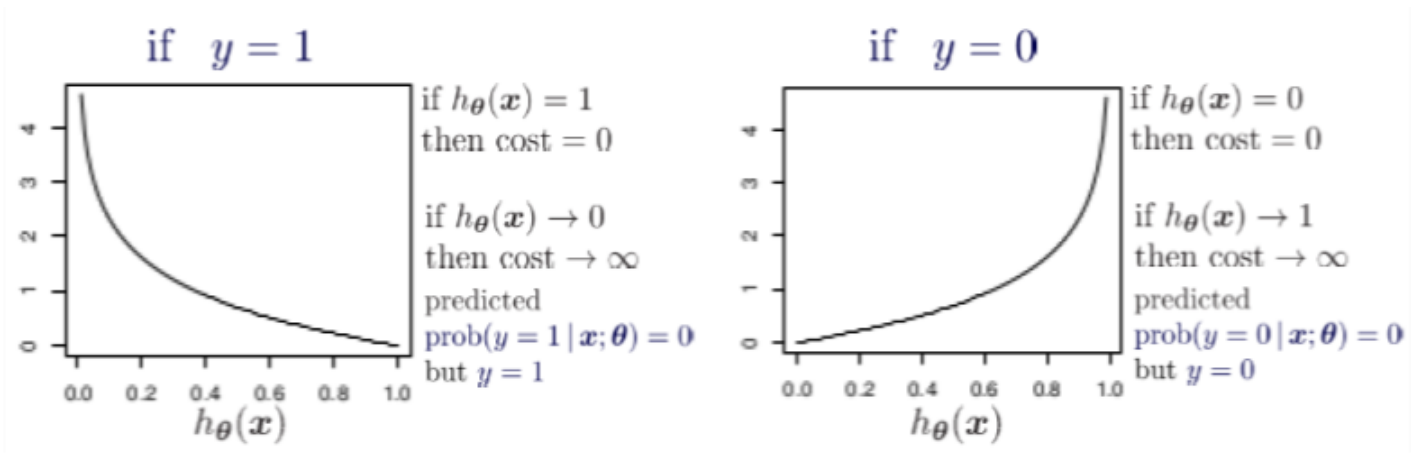


Image Source: ***https://medium.com/***

If we combine both the graphs, we will get a convex graph with only 1 local minimum and now it'll be easy to use gradient descent here.

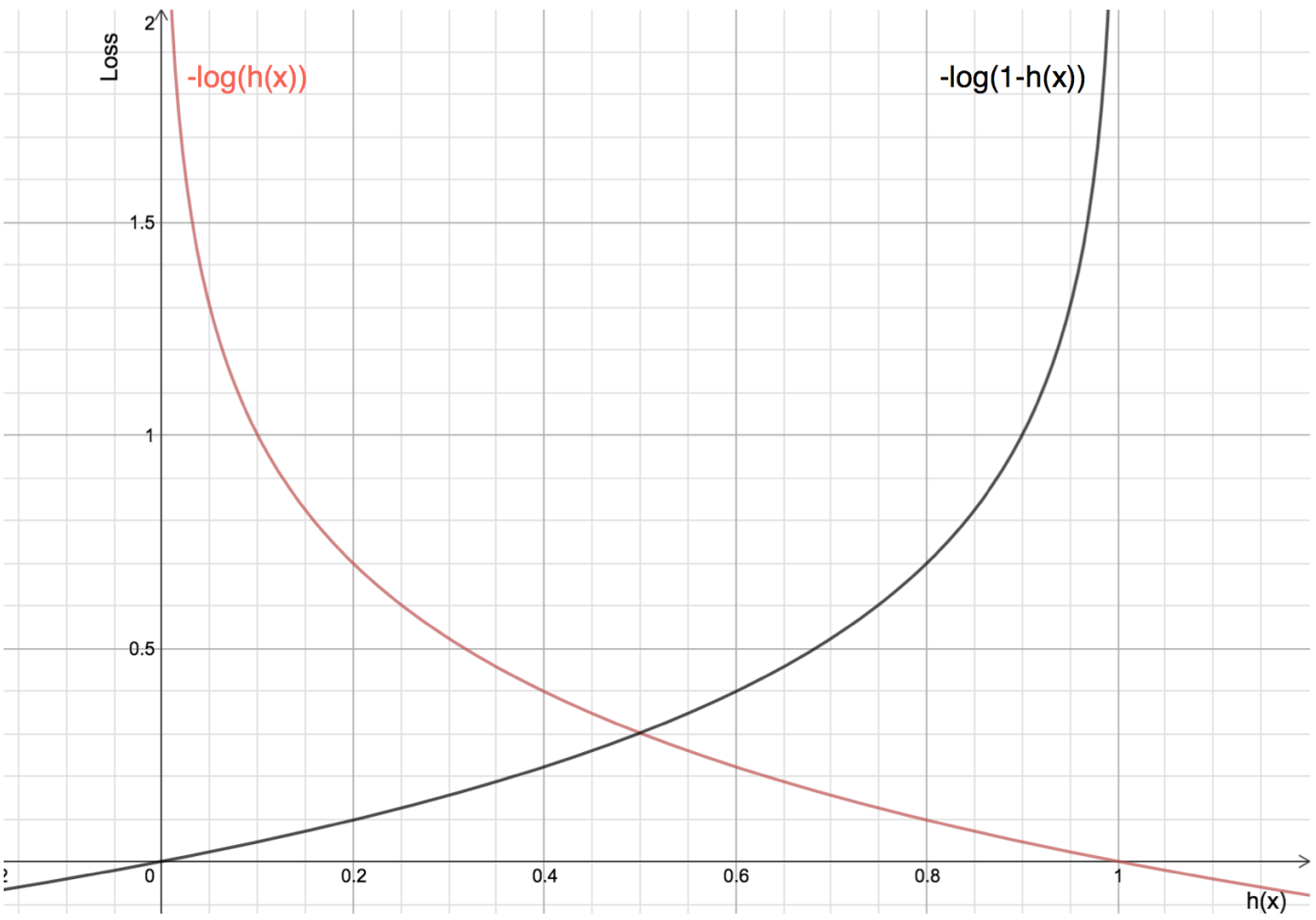


Image Source: ***https://www.analyticsvidhya.com/***

The red line here represents the 1 class (y = 1), the right term of cost function will vanish. Now

The black line represents 0 class (y=0), the left term will vanish in our cost function and if the predicted probability is close to 0 then our loss function will be less but if our probability approaches 1 then our loss function reaches infinity.

$$\text{Cost}(\text{h}_\Theta(x), y) = \begin{cases} -log(h_\Theta(x)) & if \quad y = 1 \\ -log(1 - h_\Theta(x)) & if \quad y = 0 \end{cases}$$

This cost function is also called log loss. It also ensures that as the probability of the correct answer is maximized, the probability of the incorrect answer is minimized. Lower the value of this cost function higher will be the accuracy.

Gradient Descent Optimization

In this section, we will try to understand how we can utilize *Gradient Descent* to compute the minimum cost.

Gradient descent changes the value of our weights in such a way that it always converges to minimum point or we can also say that, it aims at finding the optimal weights which minimize the loss function of our model. It is an iterative method that finds the minimum of a function by figuring out the slope at a random point and then moving in the opposite direction.

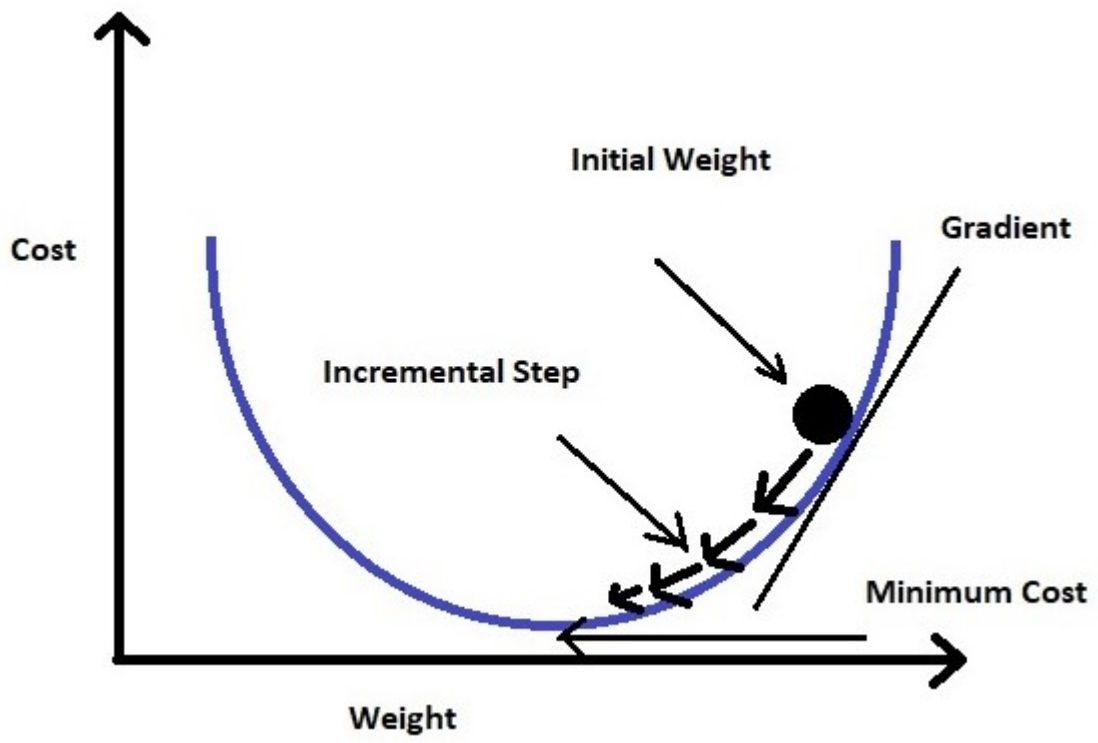


Image Source: www.analyticsvidhya.com/

> The intuition is that if you are hiking in a canyon and trying to descend most quickly down to the river at the bottom, you might look around yourself 360 degrees, find the direction where the ground is sloping the steepest, and walk downhill in that direction.

At first gradient descent takes a random value of our parameters from our function. Now we need an algorithm that will tell us whether at the next iteration we should move left or right to reach the minimum point. The gradient descent algorithm finds the slope of the loss function at that particular point and then in the next iteration, it moves in the opposite direction to reach the minima. Since we have a convex graph now we don't need to worry about local minima. A convex curve will always have only 1 minima.

We can summarize the gradient descent algorithm as:

$$\frac{\partial J(\theta)}{}$$

Here alpha is known as the learning rate. It determines the step size at each iteration while moving towards the minimum point. Usually, a lower value of *"alpha"* is preferred, because if the learning rate is a big number then we may miss the minimum point and keep on oscillating in the convex curve

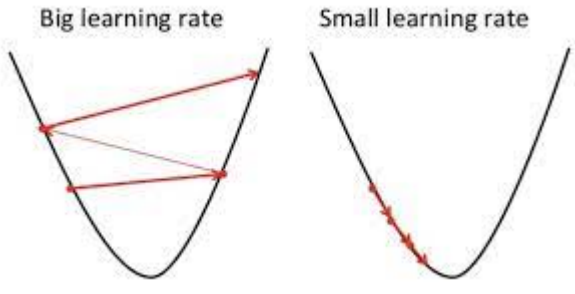


Image Source : https://stackoverflow.com/

Now the question is what is this derivative of cost function? How do we do this? Don't worry, In the next section we'll see how we can derive this cost function w.r.t our parameters.

Derivation of Cost Function:

Before we derive our cost function we'll first find a derivative for our sigmoid function because it will be used in derivating the cost function.

$$\frac{d\sigma(x)}{dx} = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right) = \frac{d}{dx}(1+e^{-x})^{-1}$$

$$\Rightarrow -(1+e^{-x})^{-2} \times \frac{d}{dx}(1+e^{-x})$$

$$\Rightarrow -(1+e^{-x})^{-2} \times [0 + \frac{d}{dx}(e^{-x})]$$

$$\Rightarrow -(1+e^{-x})^{-2} \times [e^{-x} \times \frac{d(-x)}{dx}]$$

$$\Rightarrow (1+e^{-x})^{-2} \times [e^{-x} \times 1]$$

$$\Rightarrow e^{-x}(1+e^{-x})^{-2}$$

$$\Rightarrow \frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}+1-1}{(1+e^{-x})(1+e^{-x})}$$

$$\Rightarrow \frac{(1+e^{-x})-1}{(1+e^{-x})(1+e^{-x})} = \frac{1}{(1+e^{-x})}[\frac{(1+e^{-x})}{(1+e^{-x})} - \frac{1}{(1+e^{-x})}]$$

$$\Rightarrow \frac{1}{(1+e^{-x})}[1 - \frac{1}{(1+e^{-x})}]$$

Now, we will derive the cost function with the help of the chain rule as it allows us to calculate complex partial derivatives by breaking them down.

Step-1: Use chain rule and break the partial derivative of log-likelihood.

$$-\frac{\partial LL(\theta)}{\partial \theta_j} = -\frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial \theta} \qquad where\ p = \sigma\left[\theta^T x\right]$$

$$= -\frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \theta_j} \qquad where\ z = \theta^T x$$

Step-2: Find derivative of log-likelihood w.r.t p

We know,

$$LL(\theta) = y\ \log(p) + (1-y)\log(1-p) \qquad \text{where } p = \sigma\left[\theta^T x\right]$$

$$\frac{\partial LL(\theta)}{\partial p} = \frac{y}{p} + \frac{(1-y)}{(1-p)}$$

Step-3: Find derivative of ***'p'*** w.r.t ***'z'***

$$p = \sigma(z)$$

$$\frac{\partial p}{\partial z} = \frac{\partial[\sigma(z)]}{\partial z}$$

$$We\ know\ the\ derivative\ of\ sigmoid\ function\ is\ \sigma\left[\theta^T x\right]\left[1-\sigma\left(\theta^T x\right)\right]$$

$$\Rightarrow \frac{\partial p}{\partial z} = \sigma[z][1-\sigma(z)]$$

Step-4: Find derivate of z w.r.t θ

$$z = \theta^T x \qquad \text{as previously defined}$$

$$\frac{\partial z}{\partial \theta_j} = x_j$$

Step-5: Put all the derivatives in equation 1

$$-\frac{\partial LL(\theta)}{\partial \theta_j} = -\frac{\partial LL(\theta)}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \theta_j}$$

$$-\frac{\partial LL(\theta)}{\partial \theta_j} = -\left[\frac{y}{p} + \frac{(1-y)}{(1-p)}\right].\sigma(z)\sigma(1-(z)).x_j$$

$$= -\left[\frac{y}{p} + \frac{(1-y)}{(1-p)}\right].p[1-p].x_j \qquad since\ p = \sigma[z]$$

$$= -[y(1-p) - p(1-y)].x_j$$

$$= -[y-p].x_j$$

$$\Rightarrow [p-y].x_j = [\sigma(\theta^T x) - y].x_j$$

Hence the derivative of our cost function is:

$$\theta_{new} = \theta_{old} - \alpha\left[\sigma\left(\theta^T x\right) - y\right].x_j$$

Now since we have our derivative of the cost function, we can write our gradient descent algorithm as:

If the slope is negative (downward slope) then our gradient descent will add some value to our new value of the parameter directing it towards the minimum point of the convex curve.

Frequently Asked Questions

Q1. What is logistic regression in simple terms?

A. Logistic regression is a statistical method for binary classification. It models the relationship between a dependent binary variable (like yes/no or 1/0) and one or more independent variables by estimating the probability of the binary outcome. It uses a logistic function to transform linear combinations of the independent variables, making it suitable for predicting probabilities and classifying data into two categories based on a threshold.

Q2. What are the 3 types of logistic regression?

A. The three main types of logistic regression are:

1. **Binary Logistic Regression:** Used for binary classification, where the dependent variable has only two possible outcomes.
2. **Multinomial Logistic Regression:** Applied when the dependent variable has more than two categories, but they are not ordered.
3. **Ordinal Logistic Regression:** Used when the dependent variable is ordinal, meaning it has ordered categories, but the intervals between them are not necessarily equal.

Endnote

To summarise, in this article we learned why linear regression doesn't work in the case of classification problems. Also, how MLE is used in logistic regression and how our cost function is derived.

In the next article, I will explain all the interpretations of logistic regression. And how we can check the accuracy of our logistic model.

Let me know if you have any queries in the comments below.

About the Author

I am an undergraduate student currently in my last year majoring in Statistics (Bachelors of Statistics) and have a strong interest in the field of data science, machine learning, and artificial intelligence. I enjoy diving into data to discover trends and other valuable insights about the data. I am constantly learning and motivated to try new things.

I am open to collaboration and work.

For any **doubt and queries,** feel free to contact me on Email

Connect with me on LinkedIn and Twitter

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

blogathon logistic regression

About the Author

Anshul Saini

I am an undergraduate student currently in my last year majoring in Statistics (Bachelors of Statistics) and have a strong interest in the