**Logistic Regression Intuition**

This article discusses the basics of Logistic Regression and its implementation in Python. Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for a given set of features(or inputs), X.  
Contrary to popular belief, logistic regression is a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.



Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.  
The decision for the value of the threshold value is majorly affected by the values of [precision and recall.](https://www.geeksforgeeks.org/confusion-matrix-machine-learning/) Ideally, we want both precision and recall to be 1, but this seldom is the case.

In the case of a Precision-Recall tradeoff, we use the following arguments to decide upon the threshold:-  
**1. Low Precision/High Recall:** In applications where we want to reduce the number of false negatives without necessarily reducing the number of false positives, we choose a decision value that has a low value of Precision or a high value of Recall. For example, in a cancer diagnosis application, we do not want any affected patient to be classified as not affected without giving much heed to if the patient is being wrongfully diagnosed with cancer. This is because the absence of cancer can be detected by further medical diseases but the presence of the disease cannot be detected in an already rejected candidate.  
**2. High Precision/Low Recall:** In applications where we want to reduce the number of false positives without necessarily reducing the number of false negatives, we choose a decision value that has a high value of Precision or a low value of Recall. For example, if we are classifying customers whether they will react positively or negatively to a personalized advertisement, we want to be absolutely sure that the customer will react positively to the advertisement because otherwise, a negative reaction can cause a loss of potential sales from the customer.  
Based on the number of categories, Logistic regression can be classified as: 

1. **binomial:** target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.
2. **multinomial:** target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.
3. **ordinal:** it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232307/gif-5.gif

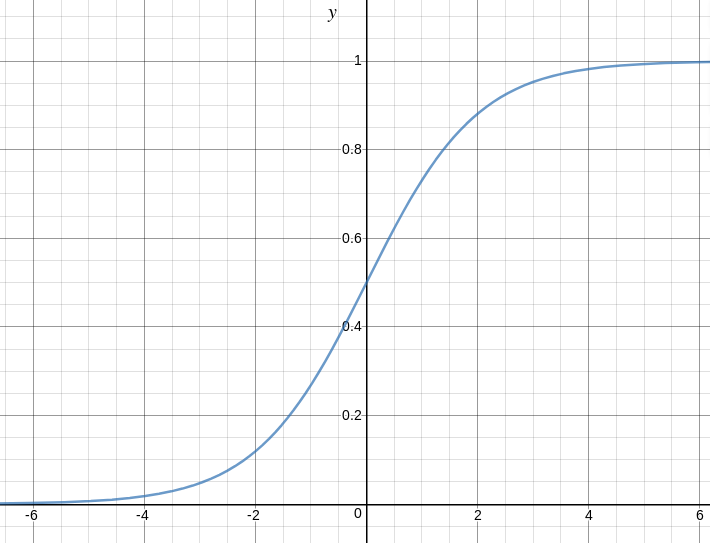
*The reason for taking= 1 is pretty clear now.*  
*We needed to do a matrix product, but there was no*  
*actualmultiplied toin original hypothesis formula. So, we defined= 1.*

Now, if we try to apply Linear Regression to the above problem, we are likely to get continuous values using the hypothesis we discussed above. Also, it does not make sense forto take values larger than 1 or smaller than 0.   
So, some modifications are made to the hypothesis for classification: 

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232331/gif-6.gif

where,

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232357/gif-7.gif

is called **logistic function** or the **sigmoid function**.   
Here is a plot showing g(z): 

We can infer from the above graph that:

* g(z) tends towards 1 as
* g(z) tends towards 0 as
* g(z) is always bounded between 0 and 1

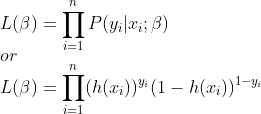
So, now, we can define conditional probabilities for 2 labels(0 and 1) forobservation as:

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232429/gif-8.gif

We can write it more compactly as:

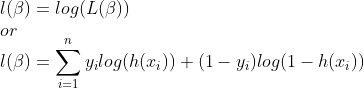
https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232455/gif-9.gif

Now, we define another term, **likelihood of parameters** as:



*Likelihood is nothing but the probability of data(training examples), given a model and specific parameter values(here,). It measures the support provided by the data for each possible value of the. We obtain it by multiplying allfor given.*

And for easier calculations, we take**log-likelihood:**



The **cost function** for logistic regression is proportional to the inverse of the likelihood of parameters. Hence, we can obtain an expression for cost function, J using log-likelihood equation as:

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232647/gif-12.gif

and our aim is to estimate so that cost function is minimized !!

**Using Gradient descent algorithm**

Firstly, we take partial derivatives ofw.r.t eachto derive the stochastic gradient descent rule(we present only the final derived value here):

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210722232711/gif-13.gif

Here, y and h(x) represents the response vector and predicted response vector(respectively). Also,is the vector representing the observation values forfeature.   
Now, in order to get min,

whereis called **learning rate** and needs to be set explicitly.