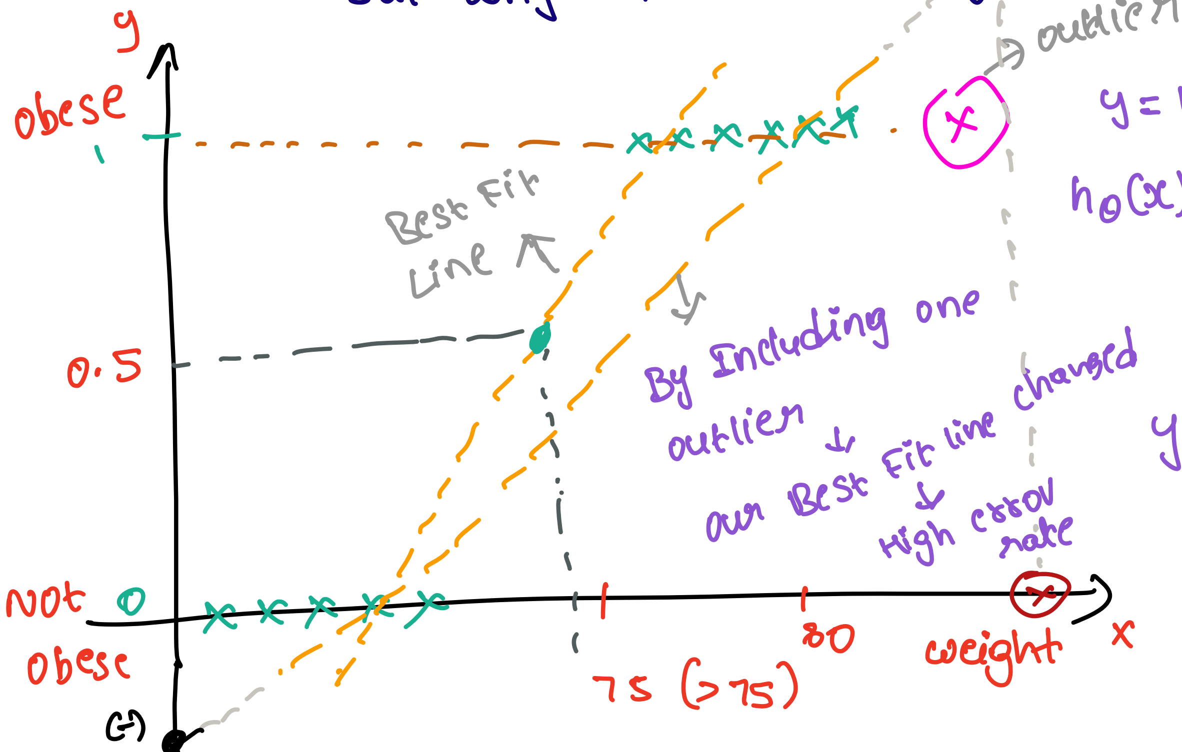
Logistic Regression

→ It is a Binary classifier
↓
But why its called Regression ≥1 y value



$y = mx + c$

$h_\theta(x) = \theta_0 + \theta_1 x$

$= \beta_0 + \beta_1 x$

$y = w^T x + b$

i.e, $h_\theta(x) \geq 0.5$

$1 < 0.5$

↓

By creating straight line like Linear Regression, we can solve this classification problem

↓

So, why we Require Logistic Regression then?

↓

For Eg we have one outlier in the Above

↓

our Best Fit line is completely changed with High Error rate

↓

In this scenarios we shouldn't use Linear Regression, we have to use Logistic Regression

Problems with Linear Regression :

→ whenever we have lot of datapoints and outliers, our Best Fit line keeps on changing giving high error rate

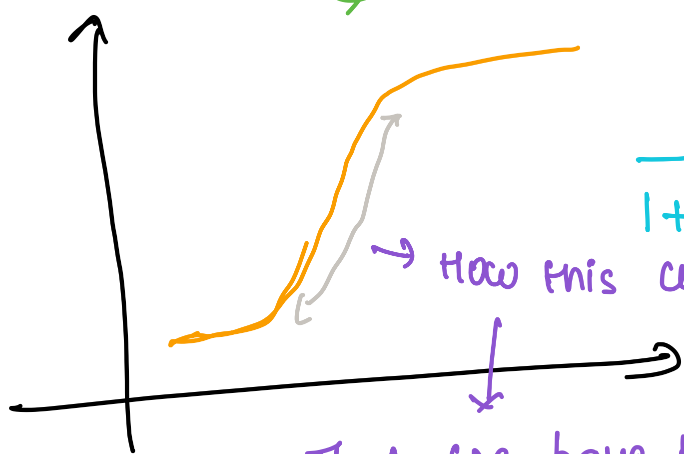
→ most of the Time values go above '1' and Below '0'

↓

In This scenarios For classification problems we use Logistic Regression

↓

Logistic regression uses sigmoid Function



$$\frac{1}{1+e^{-x}}$$

→ How this curve is created?

↓

That we have to understand

→ we shouldn't use Linear Regression for classification problems for the above reasons.

Logistic Regression Indepth Intuition :

→ understand with

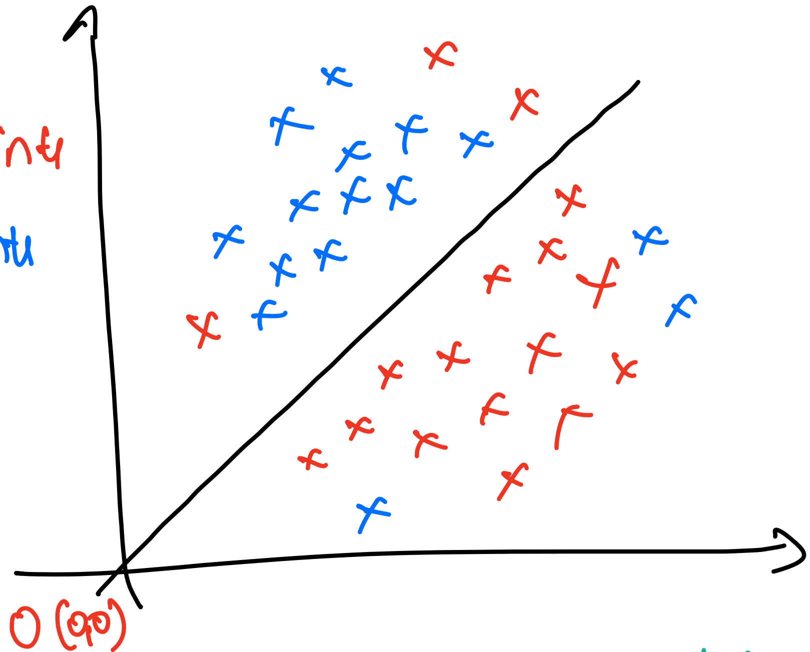
* Geometric Intuition
* Mathematics Intuition

→ Logistic Regression is mainly used for Binary classification

→ We can also modify Logistic Regression for multi-class classification

x → Negative points

x → Positive points



→ Logistic Regression can be Applied, If the data points are Linearly separable

↓

means data points can be divided with the help of a straight line

↓

In Linear regression also we are creating a Straight line

↓

what is the difference ?

↓

In Logistic Regression we are creating a Line (or) Plane, which will divide data points

Line can be written as :

$$y = mx + c$$

slope

data points

Intercept

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

$$y = w^T x + b$$

→ Here the Best Fit line is not created with the help of Linear Regression

↓

There is other way in Logistic Regression

↓ $y = \underline{m}x + c$

we need to modify this coefficient (m) which will give us the Best Fit line

↓

How this will happen in Logistic Regression ?

↓

<u>Assumptions in Logistic Regression</u>

+ve → +1 (All positive points are denoted as '+1')

−ve → −1 (All negative points are denoted as '−1')

→ From the above graph, Best Fit line is passing through origin

↓

So Intercept will Become Zero

$$y = w^T x + b = 0$$

$$\boxed{y = w^T x}$$

For All these data points Above plane

↓

The distance Between plane and data points is Always (+)ve

↓

Because slope of derivative is Always positive

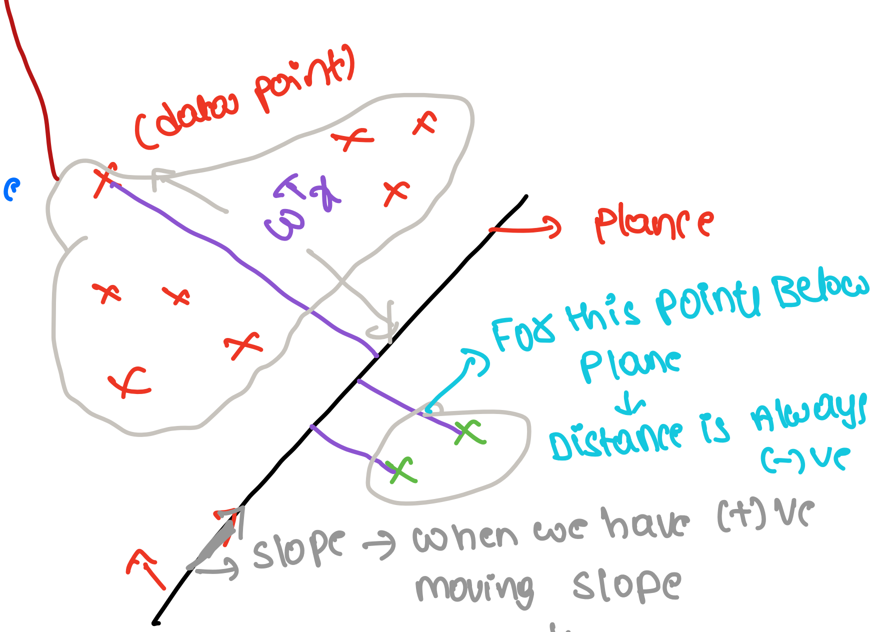
For Eg:

→ If we want to Find distance Between data point and plane

↓

This is the Formula

$$\frac{w^T x + b}{||w||}$$



slope → when we have (+)ve moving slope

↓

slope of Derivative is always positive

→ If we consider "w" as the unit vector

$$||w|| = 1$$

→ since Line is passing through origin

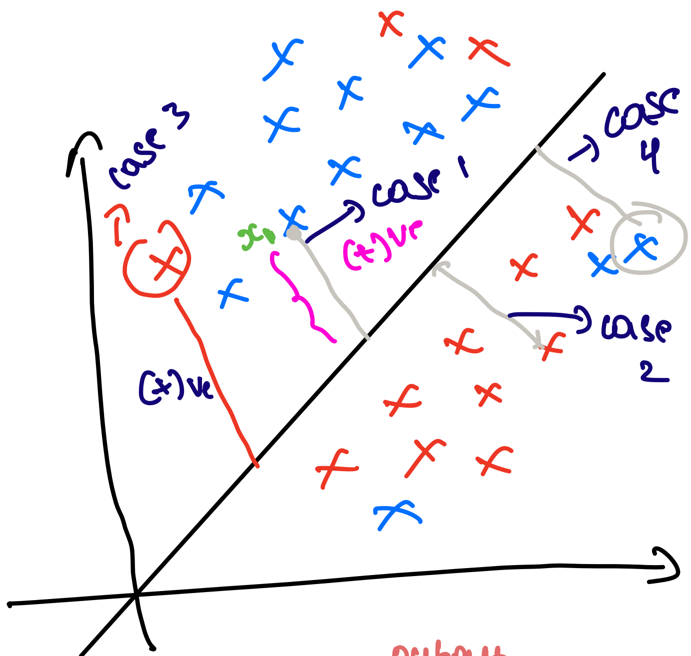
↓

Intercept, $b = 0$

$\boxed{= w^T x}$ → This nothing But, distance Between the data point and the plane

→ For many data points

$$\boxed{\sum_{i=1}^{n} w_i^T x_i}$$



Case : 1

$y_1 = +ve$ (since its Above plane)

$w^T x > 0$

→ $\boxed{y \times w^T x} > 0$

↓

If this value is (+)ve, we can clearly classify y_1 is +ve (easily we can classify)

output

Case : 2

$y = -1$, $w^T x < 0$

→ $\boxed{y \times w^T x} > 0$ → This particular point is Able to get classified correctly

$(-) \times (-) = (+)$

Case : 3

$y = -1$, $w^T x > 0$

$\boxed{y \times w^T x} < 0$ → when we have this value $(<) 0$

↓

means That particular datapoint is Incorrectly classified

↓

which is True in our case

Case : 4 :

$y = +1$, $w^T x = -ve$

$\boxed{y \times w^T x} < 0$ → Since this value is negative

↓

The datapoint is Incorrectly classified

Important :

$\boxed{\text{cost Function} \quad \sum_{i=1}^{n} y_i \, w_i^T x_i}$ → This should be maximum as possible

→ The Best Fit line To linearly separates all the data points

↓

we have to make sure that, the summation of all the points along with the distance should be maximum

↓

Because whenever $y \times w^T x > 0$, correctly classifying

$y \times w^T x < 0$, wrongly classifying

Cost Function for Best Fit Line [For Logistic Regression]

$$\max \sum_{i=1}^{n} y_i \, w_i^T x_i$$

y_i → we know this value

↓

$y = +1$, Above Plane

$y = -1$, Below Plane

x_i → we know this value

w_i^T → The only parameter we need to update is $\boxed{w_i^T}$

↓

Which is Actually the Coefficient (or) slope

↓

So we have to update this weight (or) coefficient (or) slope

↓

So that we get maximum of cost Function

Summary :

→ main Aim is to maximize the Below cost Function

→ cost Function (or) optimizer For Logistic Regression

$$\boxed{\max \sum_{i=1}^{n} y_i \, w_i^T x_i}$$

→ This should be maximum value for the best fit line in Logistic Regression

↓

The summation of y_i and the distance between plane and all of the datapoints ($w_i^T x_i$)

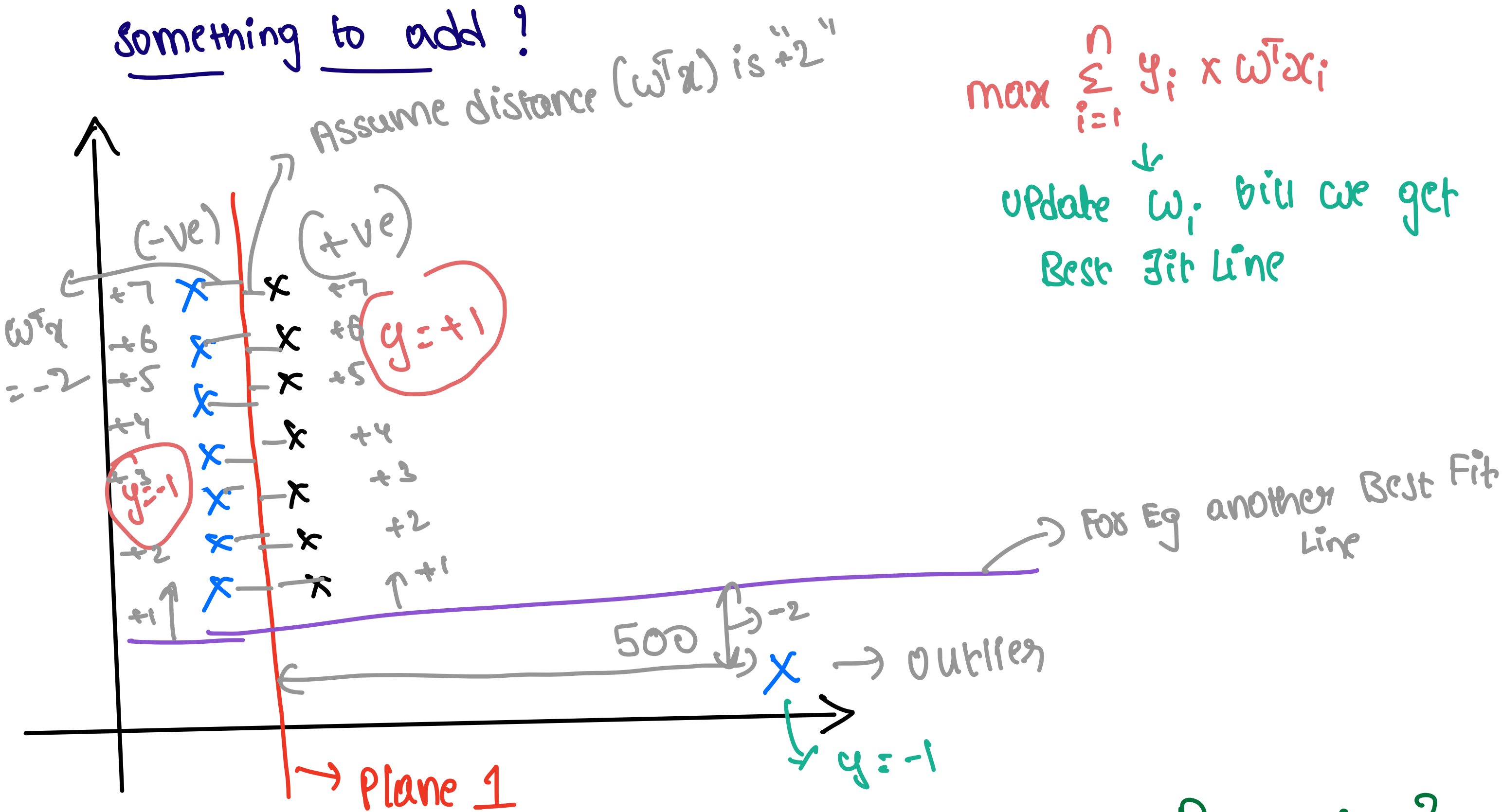
Training Data set

But still Above is not considered as optimizer, there is still something to add?

$$\max \sum_{i=1}^{n} y_i \times w^T x_i$$

↓

Update w_i till we get Best Fit Line



What happens if we have some outliers in Logistic Regression?

→ $w^T x = 2+2+2+2+2+2+2$
$\quad + 2+2+2+2+2+2+2$ -500 (outlier)
$\quad = -476$

↓

But here we are getting (-)ve value

↓

But our main Aim is max cost Function

↓

Impact of outliers

calculations for next Best Fit line [which is Horizontal]

→ $w^T = -6-5-4-3-2-1$
$\quad + 6+5+4+3+2+1 + (-1)(-2) = 2$ (which is (+)ve)

→ If we see Above:

* The First Fit line is Best, Because it divides data points Bit accurately visibly

[-480]

↓

But due to outlier we are getting Negative cost Function

* The Second Fit line is not Best, which we can clearly see, from the above graph

↓

But cost Function is (+2)

↓

So Ideally Model selects this [since max cost Function]

↓

This incorrect selection of Best Fit line is due to the Impact of outliers

So, How do we prevent this?

→ For this we have to modify our cost Function

new cost Function:

$$\max f\left(\sum_{i=1}^{n} y_i \times w^T x_i\right)$$

↓

This 'f' function is nothing But Sigmoid function

↓

$$\max \sum_{i=1}^{n} f(\underline{y_i \times w^T x_i})$$

↓

This value is passed through sigmoid Function

$$\boxed{\text{Sigmoid function} = \frac{1}{1+e^{-z}}}$$

→ $z = y_i \times w^T x_i$

→ The distance of outlier data point to the plane is "500"

From Above

$$\underset{\underset{-1}{\shortparallel}}{y_i} \times \underset{\underset{500}{\shortparallel}}{w^T x_i} = -500$$

→ very Big number

↓

on summation we got Big negative number

↓

we will pass this through sigmoid function

↓

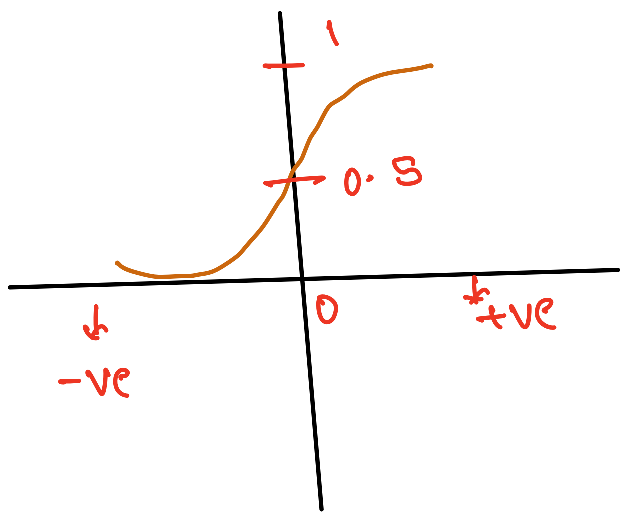
Sigmoid Function will Transfer this (-500)

↓ to

(0 to 1) range

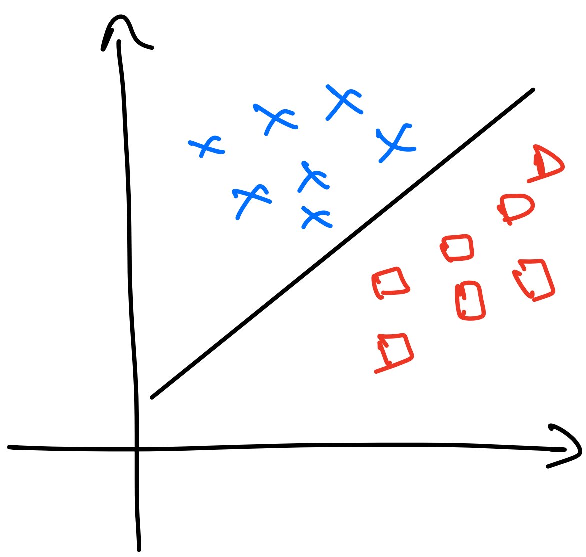
↓

Sigmoid Function is removing the Effect of outliers



How we can solve a multi class classification problem using Logistic Regression? (one v Rest) (or) (one v All) concept

Binary classifier:



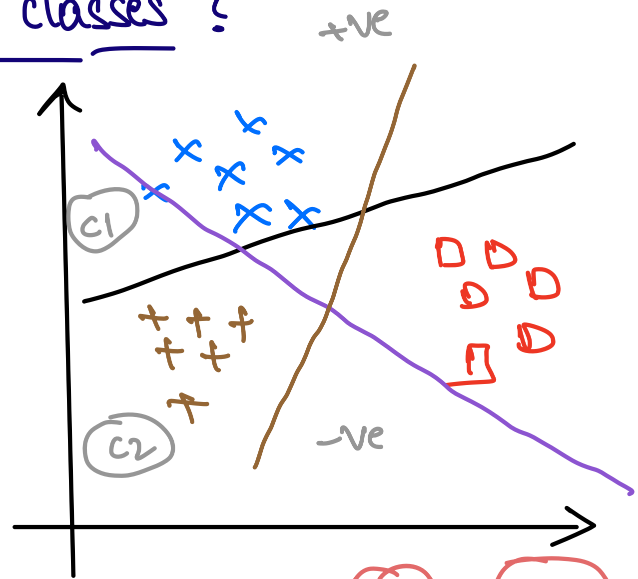
2 classes

Why if we have more than 2 classes?

→ One vs Rest concept

↓

By Grouping the classes



↳ M_1 model → C1, C2

↳ M_2 model

↳ M_3 model

f_1	f_2	f_3	o/p	O_1	O_2	O_3
I_1	I_2	I_3	O_1	+1	-1	-1
I_4	I_5	I_6	O_2	-1	+1	-1
I_7	I_8	I_9	O_3	-1	-1	+1
I_{10}	I_{11}	I_{12}	O_1	+1	-1	-1
I_{13}	I_{14}	I_{15}	O_2	-1	+1	-1
				↓ M_1	↓ M_2	↓ M_3

→ 'M_1' will give 'O_1' by taking all Input Features (f_1, f_2, f_3)

→ 'M_2' will give 'O_2' by taking all Input Features (f_1, f_2, f_3)

→ "M_3" will give 'O_3' by taking all Input features (f_1, f_2, f_3)

How this work on New Test Data :

→ For Every Data Point

↓

Model M_1, M_2, M_3 will Give '3' probabilities

↓

[0.20, 0.25, 0.55]

↓

O_3 having Highest probability

→ In Logistic Regression, For multiclass we have to set a parameter

↓

multiclass = 'ovr'

Why can't we use the cost Function used for Linear Regression for Logistic Regression

→ The cost Function for linear regression is "mean squared error"

→ If we use it for logistic regression

↓

The parameter function will become Non-convex

↓

Only if the function is convex

↓

Gradient descent lead to global minimum.

What is Logistic Regression Model?

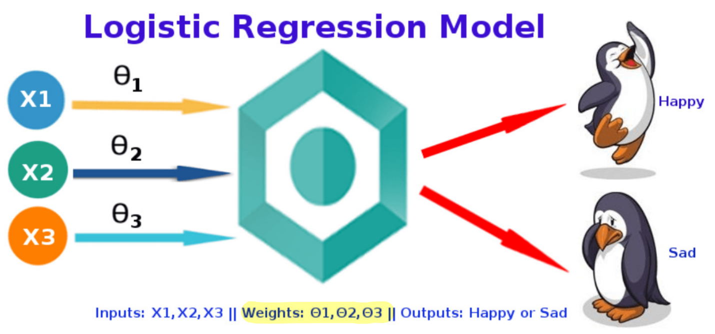
Logistic regression is a handy tool for picking between two choices. It's like predicting if it's going to rain or shine tomorrow based on today's weather.

Logistic regression estimates the relationship between a dependent variable and one or more independent variables and predicts a categorical variable versus a continuous one. Here are a few things to know about logistic regression:

- Logistic regression is a Machine Learning method used for classification tasks.
- It is a predictive analytic technique based on the probability idea.
- The dependent variable in logistic regression is binary (coded as 1 or 0).
- The goal is to discover a link between characteristics and the likelihood of a specific outcome.
- Logistic regression uses a more sophisticated cost function called the "Sigmoid function" or "logistic function" instead of a linear function.
- The logistic regression hypothesis limits the cost function to a value between 0 and 1, making linear functions unsuitable for this task.
- Logistic regression finds applications in various fields such as finance, marketing, healthcare, and social sciences, where it is employed to model and predict binary outcomes.

Logistic Regression uses cost Function called "Sigmoid" (or) "Logistic Function"
↓
Instead of Linear Function

Behind every great leader, there was an even greater logistician.



Logistic Regression is considered a regression model also. This model creates a regression model to predict the likelihood that a given data entry belongs to the category labeled "1." Logistic regression models the data using the sigmoid function, much as linear regression assumes that the data follows a linear distribution.

Why the Name Logistic Regression?

We call it logistic regression because of its special trick, the sigmoid function. Think of it as a secret formula that turns numbers into probabilities, helping us decide between two outcomes. Or, we can say 'Logistic Regression' since the technique behind it is quite similar to Linear Regression. The name "Logistic" originates from the Logit function, which plays a central role in this categorization approach.

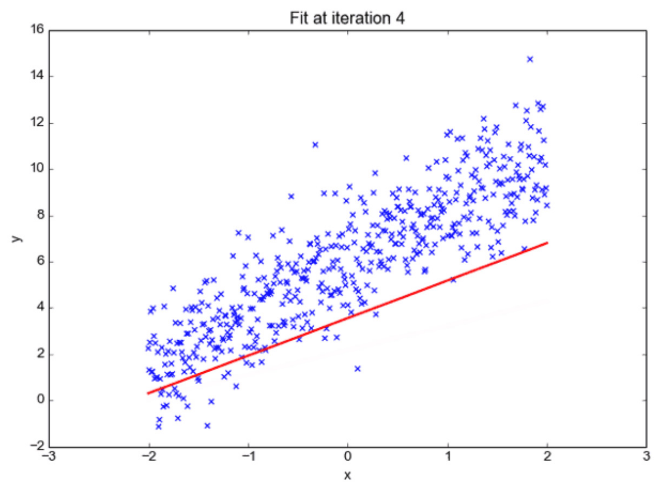
Why Can't we use Linear Regression Instead of Logistic Regression?

Before answering this question, we will explain from [Linear Regression]() concept, from the scratch then only we can understand it better. Although logistic regression is a sibling of linear regression, it is a classification technique, despite its name. Using linear regression for categories is like trying to fit a square peg into a round hole. It might work, but it won't give us accurate results like logistic regression does.

Mathematically, one can explain linear regression as follows:

- y = mx + c
- y – predicted value
- m – slope of the line
- x – input data
- c- Y-intercept or slope

We can forecast y values such as using these values. Now observe the below diagram for a better understanding,



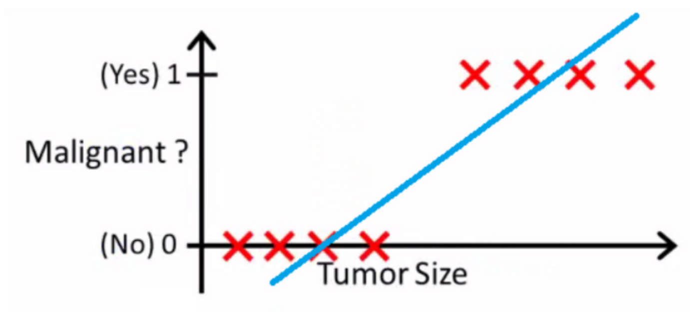
The x values are represented by the blue dots (the input data). We can now compute slope and y coordinate using the input data to ensure that our projected line (red line) covers most of the locations. We can now forecast any value of y given its x values using this line.

One thing to keep in mind about linear regression is that it only works with continuous data. If we want to include linear regression in our classification methods, we'll have to adjust our algorithm a little more. First, we must choose a threshold so that if our projected value is less than the threshold, it belongs to class 1; otherwise, it belongs to class 2.

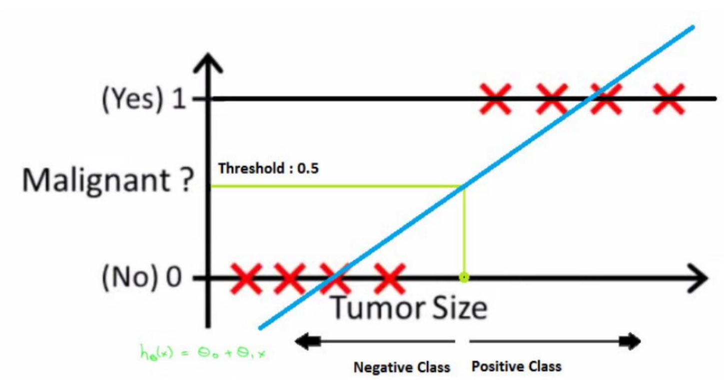
Now, if you're thinking, "Oh, that's simple, just create linear regression with a threshold, and hurray!, classification method," there's a catch. We must specify the threshold value manually, and calculating the threshold for huge datasets will be impossible. Furthermore, even if our anticipated values vary, the threshold value will remain the same. A logistic regression, on the other hand, yields a logistic curve with values confined to 0 and 1. In logistic regression, we generate the curve by using the natural logarithm of the target variable's "odds" rather than the probability, as in linear regression. Additionally, the predictors do not need to be regularly distributed or have the same variance in each group.

And Now the Question?

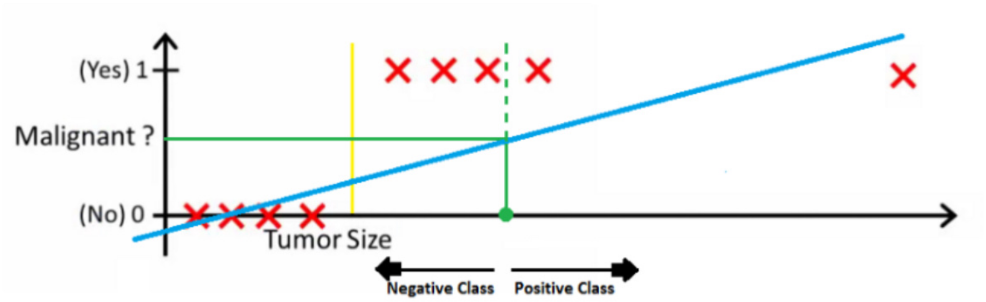
Our beloved person, Andrew Ng, explained this famous title question. Let's assume we have information about tumor size and malignancy. Because this is a classification issue, we can see that all the values will fall between 0 and 1. And, by fitting the best-found regression line and assuming a threshold of 0.5, we can do a very good job with the line.



We can select a point on the x-axis where all values to the left are considered negative, and all values to the right are regarded as positive.



But what if the data contains an outlier? Things would become shambles. For 0.5 thresholds, for example,



Even if we fit the best-found regression line, we won't be able to determine any point where we can distinguish classes. It will insert some instances from the positive class into the negative class. The green dotted line (Decision Boundary) separates malignant and benign tumors, however, it should have been a yellow line that clearly separates the positive and negative cases. As a result, even a single outlier can throw the linear regression estimates off. And it's here that logistic regression comes into play.

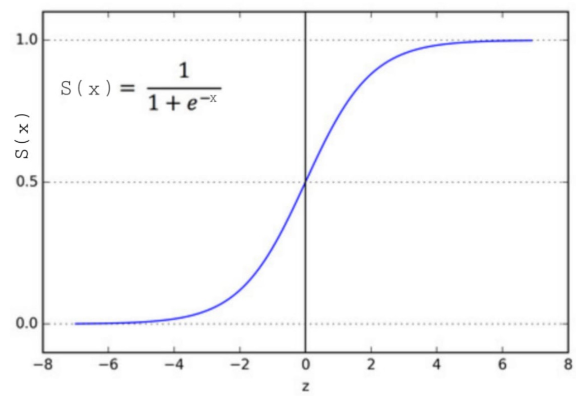
Logit function to Sigmoid Function

Moving from the logit function to the sigmoid function is like turning raw data into something we can actually use, sort of like turning a block of wood into a finely crafted sculpture.

Logistic Regression can be expressed as,

$$log\left(\frac{p(X)}{1-p(X)}\right) = \beta 0 + \beta 1X$$

where p(x)/(1-p(x)) is termed odds, and the left-hand side is called the logit or log-odds function. The odds are the ratio of the chances of success to the chances of failure. As a result, in Logistic Regression, a linear combination of inputs is translated to log(odds), with an output of 1.



The following is the inverse of the aforementioned function

$$p(X) = \frac{e^{\beta 0+\beta 1X}}{1 + e^{\beta 0+\beta 1X}}$$

This is the Sigmoid function, which produces an S-shaped curve. It always returns a probability value between 0 and 1. The Sigmoid function is used to convert expected values to probabilities. The function converts any real number into a number between 0 and 1. We utilize sigmoid to translate predictions to probabilities in machine learning.

The mathematically sigmoid function can be,

$$\boxed{f(x) = \frac{1}{1 + e^{-(x)}}} \rightarrow$$ sigmoid (or) Logistic Function

Types of Logistic Regression

Just like there are different types of pizza, there are different types of logistic regression: binary, multinomial, and ordinal. Each one serves a different purpose.

1. Binary Logistic Regression – two or binary outcomes like yes or no
2. Multinomial Logistic Regression – three or more outcomes like first, second, and third class or no class degree
3. Ordinal Logistic Regression – three or more like multinomial logistic regression but here with the order like customer rating in the supermarket from 1 to 5

Requirements for Logistic Regression

To use logistic regression, you need clean data, no big surprises between data points, and a straight line that shows the relationship between variables.

This model can work for all the datasets, but still, if you need good performance, then there will be some assumptions to consider,

- The dependant variable in binary logistic regression must be binary.
- Only the variables that are relevant should be included.
- The independent variables must be unrelated to one another. That is, there should be minimal or no multicollinearity in the model.
- The log chances are proportional to the independent variables.
- Large sample sizes are required for logistic regression.

Decision Boundary – Logistic Regression

The decision boundary in logistic regression is like a fence that separates the cats from the dogs. It's where we say, "This side is for cats, and that side is for dogs."

We can establish a threshold to predict the class to which a given data point belongs. The estimated probability obtained is then classified into classes based on this threshold.

If the predicted value is less than 0.5, categorize the particular student as a pass; otherwise, label it as a fail. There are two types of decision boundaries: linear and non-linear. To provide a complicated decision boundary, the polynomial order can be raised.

Why can't we use the cost function used for linearity for logistic regression?

The cost function for linear regression is mean squared error. If we use it for logistic regression, the parameter function will become non-convex. Only if the function is convex will gradient descent lead to a global minimum.

Cost function – Linear Regression Vs Logistic Regression

In linear regression, we measure how far our guesses are from the real thing. In logistic regression, we measure how close our guesses are to the probabilities we want.

Linear regression employs the Least Squared Error as the loss function, which results in a convex network, which we can then optimize by identifying the vertex as the global minimum. For logistic regression, however, it is no longer a possibility. Also, modifying the hypothesis results in a non-convex graph with local minimums when calculating Least Squared Error using the sigmoid function on raw model output.

What is cost function? Cost functions are used in machine learning to estimate how poorly models perform. Simply put, a cost function is a measure of how inaccurate the model is in estimating the connection between X and y. This is usually stated as a difference or separation between the expected and actual values. A machine learning model's goal is to discover parameters, weights, or a structure that minimizes the cost function.

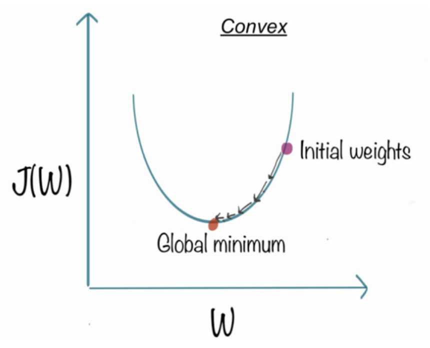
A convex function indicates there will be no intersection between any two points on the curve, but a non-convex function will have at least one intersection. In terms of cost functions, a convex type always guarantees a global minimum, whereas a non-convex type only guarantees local minima.

Linear Regression

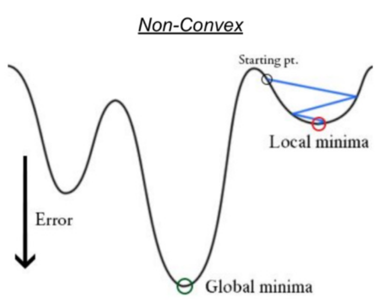
we measure how far our guesses are from the real thing

Logistic Regression

we measure how close our guesses are to the probabilities we want



convex

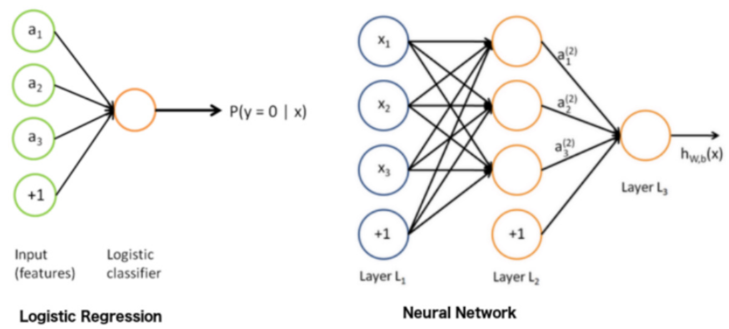


non - convex

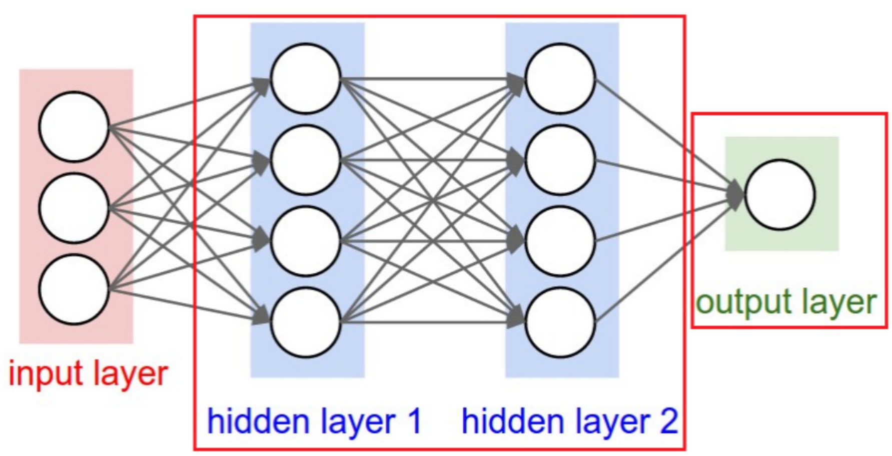
How Logistic Regression links with Neural Network?

Logistic regression and neural networks are like cousins. They share some similarities, but neural networks are like the big brother with more tricks up its sleeve.

We all know that Neural Networks are the foundation for Deep Learning. The best part is that Logistic Regression is intimately linked to Neural networks. Each neuron in the network may be thought of as a Logistic Regression; it contains input, weights, and bias, and you conduct a dot product on all of that before applying any non-linear function. Furthermore, a neural network's last layer is a basic linear model (most of the time). That can be understood by visualization as shown below,



Take a deeper look at the "output layer," and you'll notice that it's a basic linear (or logistic) regression: we have the input (hidden layer 2), the weights, a dot product, and finally a non-linear function, depends on the task. A helpful approach to thinking about neural networks is to divide them into two parts: representation and classification/regression. The first section (on the left) aims to develop a decent data representation that will aid the second section (on the right) is doing a linear classification/regression.



Hyperparameter Fine-tuning – Logistic Regression

Hyperparameter fine-tuning is like adjusting the knobs on a radio until you find the perfect station. It helps us optimize our model for peak performance.

There are no essential hyperparameters to adjust in logistic regression. Even though it has many parameters, the following three parameters might be helpful in fine-tuning for some better results,

Regularization (penalty) might be beneficial at times.

Penalty – *{'l1', 'l2', 'elasticnet', 'none'}, default='l2'*

The penalty strength is controlled by the C parameter, which might be useful.

C – float, default=1.0

With different solvers, you might sometimes observe useful variations in performance or convergence.

Solver – *{'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'*

Note: The algorithm to use is determined by the penalty: Solver-supported penalties:

- 'newton-cg' – ['l2', 'none']
- 'lbfgs' – ['l2', 'none']
- 'liblinear' – ['l1', 'l2']
- 'sag' – ['l2', 'none']
- 'saga' – ['elasticnet', 'l1', 'l2', 'none']

Advantages of Logistic Regression

Logistic regression has its perks, like being easy to understand, but it's not without its flaws, such as struggling with complex relationships in data.

1. Overfitting is less likely with logistic regression, although it can happen in high-dimensional datasets. In these circumstances, regularization (L1 and L2) techniques may be used to minimize over-fitting.
2. It works well when the dataset is linearly separable and has good accuracy for many basic data sets.
3. It is more straightforward to apply, understand, and train.
4. The inferences regarding the relevance of each characteristic are based on the anticipated parameters (trained weights). The association's orientation, positive or negative, is also specified.As a result, logistic regression can be utilized to determine the relationship between characteristics. Unlike decision trees or support vector machines, this technique permits models to be easily adjusted to accommodate new data. Stochastic gradient descent can be employed for data updating.
5. It is less prone to over-fitting in a low-dimensional dataset with enough training instances.
6. When the dataset includes linearly separable characteristics, Logistic Regression shows to be highly efficient.
7. It has a strong resemblance to neural networks. A neural network representation may be thought of as a collection of small logistic regression classifiers stacked together.

   → A neural network representation can be thought of as a collection of small logistic regression classifiers stacked together

8. The training time of the logistic regression method is considerably smaller than that of most sophisticated algorithms, such as an Artificial Neural Network, due to its simple probabilistic interpretation.
9. Multinomial Logistic Regression is the name given to an approach that may easily be expanded to multi-class classification using a softmax classifier.

Disadvantages of Logistic Regression

1. Logistic Regression is not advisable when the number of observations is fewer than the number of features, as this can lead to overfitting.
2. Because it creates linear boundaries, we won't obtain better results when dealing with complex or non-linear data.
3. It's only good for predicting discrete functions. As a consequence, the Logistic Regression model is constrained to having a dependent variable that is restricted to a discrete numerical set.
4. Logistic regression requires that there is little to no multicollinearity between independent variables.
5. Logistic regression needs a big dataset and enough training samples to identify all of the categories.
6. Because this method is sensitive to outliers, the presence of data values in the dataset that differs from the anticipated range may cause erroneous results.
7. Utilizing only significant and relevant features is crucial in constructing a model. Otherwise, the model's probabilistic predictions might be inaccurate, leading to a decline in its predictive value.
8. Complex connections are difficult to represent with logistic regression. More powerful and sophisticated algorithms, such as Neural Networks, often outperform this technique readily.
9. Because logistic regression has a linear decision surface, it cannot address nonlinear issues. In real-world settings, linearly separable data is uncommon. Consequently, transforming non-linear features becomes necessary, often achieved by augmenting the feature space to enable linear separation in higher dimensions.
10. Based on independent variables, a statistical analysis model seeks to predict accurate probability outcomes. On high-dimensional datasets, this may cause the model to be over-fit on the training set, overstating the accuracy of predictions on the training set, and so preventing the model from accurately predicting outcomes on the test set. This commonly occurs when training the model with a small amount of training data and numerous features. Exploring regularization strategies on high-dimensional datasets becomes essential to mitigate overfitting, although this complexity adds to the model. The model may be under-fit on the training data if the regularization parameters are too high.

Application of Logistic Regression

Logistic regression finds its way into many areas, from healthcare to finance. It helps with things like predicting diseases or figuring out which customers might leave

All use cases where data must be categorized into multiple groups are covered by Logistic Regression. Consider the following illustration:

1. Fraud detection in Credit card
2. Email spam or ham
3. Sentiment Analysis in Twitter analysis
4. Image segmentation, recognition, and classification – X-rays, Scans
5. Object detection through video
6. Handwriting recognition
7. Disease prediction – Diabetes, Cancer, Parkinson etc..