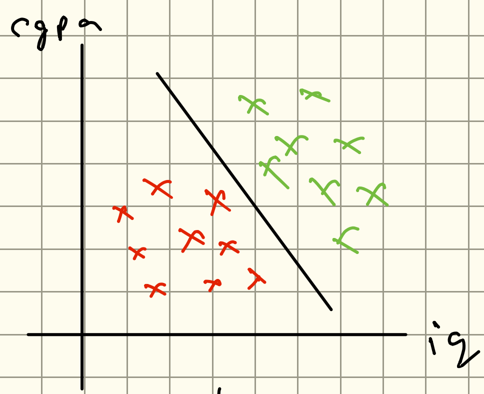
Logistic Regression - 01

* Introduction

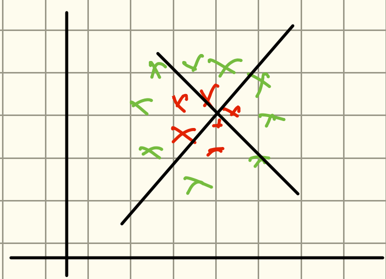
- → classification
- → linear model
  - → linear Regression
  - → logistic Regression → lines/planes → linear/sort of linear data.

cgpa | iq | placement



↓

I will use LR



↓

I will not use LR

Summary

Logistic Regression

↳ its name is Regression but it works like classification

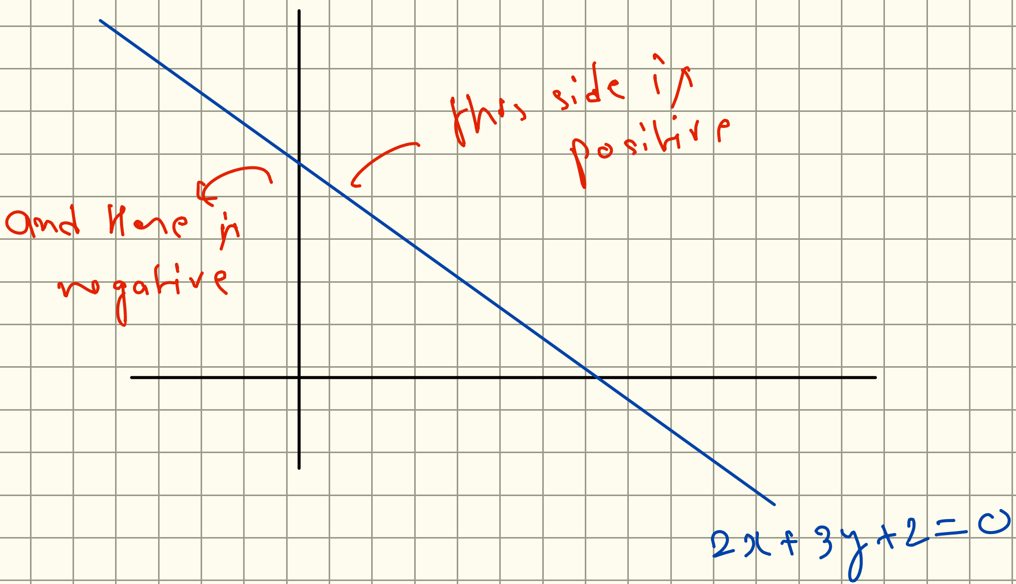
↳ it come under linear model

↳ Logistic Regression is always used for linear/sort of linear data.

Some Basic Geometry

① Every line has a positive side and a negative side.

$$\boxed{Ax + By + C = 0} \longrightarrow \text{general eq of line}$$



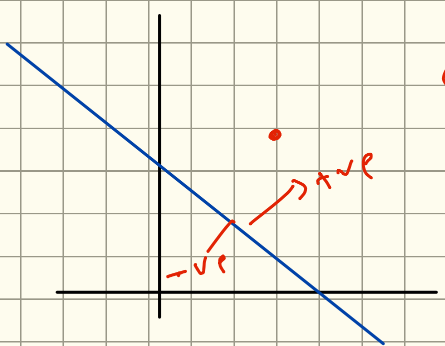
② How to find out if a given point lies on a given line?

$4x + 3y + 3 = 0$ $\quad$ $(5, 2)$

$$\underbrace{4 \times 5 + 3 \times 2 + 3}_{\downarrow\ (0)} = 0$$

$(0) \rightarrow$ if it will give zero then its given point lies on a given line.

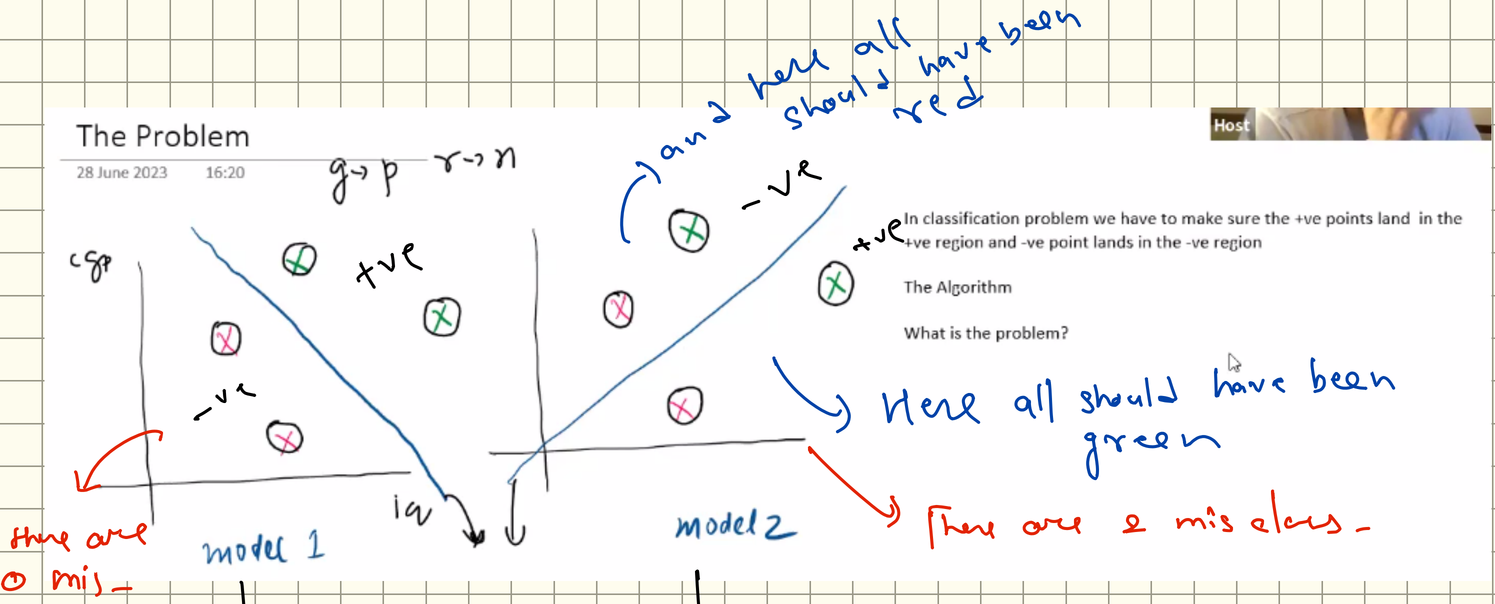
③ How to find out if a given point is on the positive side of the line or the negative side of the line



→ How do we know if it is on the +ve side or +ve side?

$Ax_1 + By_1 + C > 0 \longrightarrow$ +ve region

$Ax_1 + By_1 + C < 0 \longrightarrow$ -ve region



model 1 is better then model 2

↓

because there correctly classifi all the point

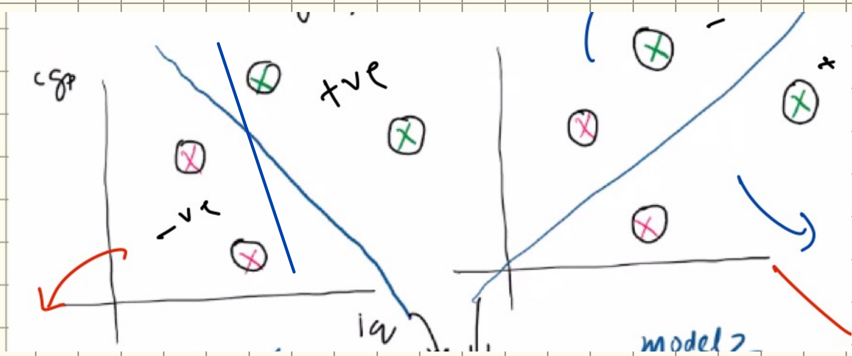
Where as model 2 is not good

↓

because it is doing certain mis classification

We made this decision based on the basics of misclassification. This means the model which makes fewer misclassification will be better.

#



→ First How will we write the code

↳ First, in coding we take a loop

Inside the loop we check:-

- if green come and it is on the positive side, and the equation is also positive, then it is a correct classification

→ $\boxed{\text{green}}$ → +ve
and → correct classi
$\boxed{Ax_1 + By_1 + C > 0}$ → +ve

- if red comes and it is on the negative side, but the equation comes positive, then it is a misclassification

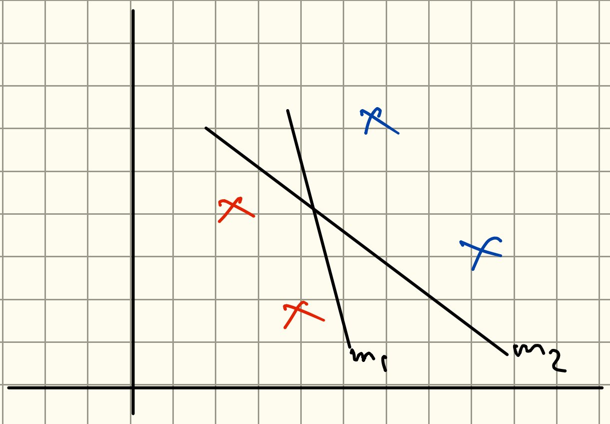
$\boxed{\text{red}}$ → -ve
→ misclassification
$\boxed{Ax_1 + By_1 + C > 0}$ → +ve

With this, we can easily build a model.

→ but the problem comes when there are two lines in the model. both line are correct, then how will we know which line is the right one?

→ At that point, Logistic Regression comes to solve this problem. before this, logistic regression was not there.

New problem formulation



⇒ Here we are using step function for the given prediction point. We put this point into the equation and calculate z.

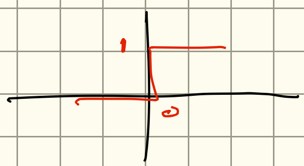
- if $z > 0$, then the output is 1
- other wise, the output is 0

This means we are only getting z (discrete output).

$$\boxed{\beta_0 + \beta_1 x_1 + \beta_2 x_2 = z}$$

Step

$z > 0$	1
$z < 0$	0



$$\boxed{\text{step}(z) = 0, 1}$$

but I want continuous value

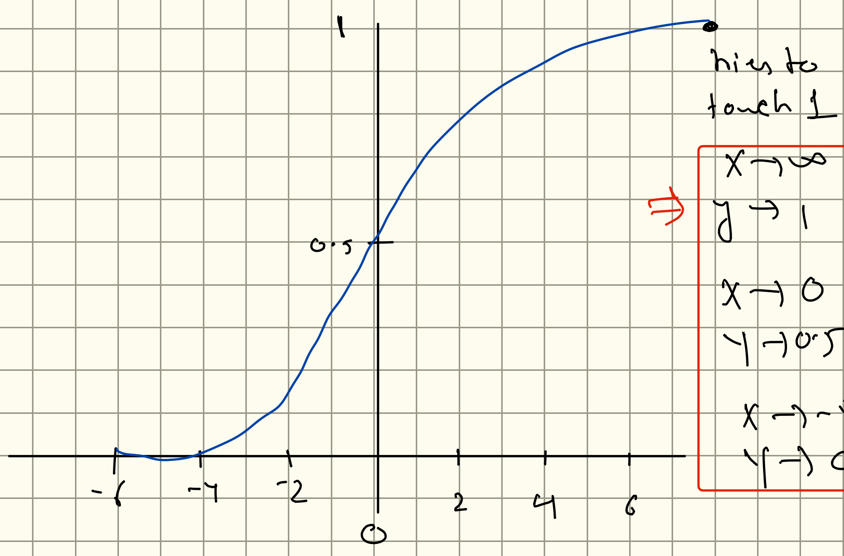
for example:-

- if the point is far, it value should be around 0.9
- if it is less far, then maybe 0.6
- if it is even closer, the maybe 0.2, etc.

That means we want to calculate the distance of the prediction point from the line

So I will not use step function because step function kaam kharab kar raha hai.

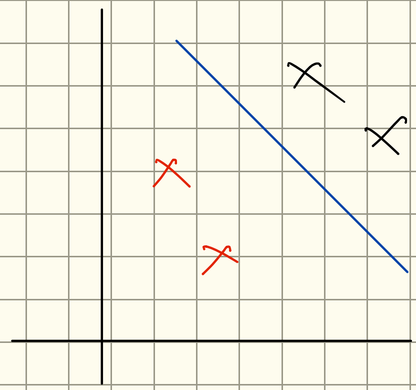
Sigmoid Function



$x \to \infty$
$\Rightarrow y \to 1$

$x \to 0$
$y \to 0.5$

$x \to -\infty$
$y \to 0$



$$\boxed{Y = \frac{1}{1+e^{-x}}}$$ Sigmoid function

⇒ **Now we use sigmoid**

First we calculate z just like before.

Now, instead of giving z into a step function, We give it to the sigmoid Function.

from sigmoid, We get a number between 0 and 1, depending on the value of z.

- if $z = 0$, the sigmoid $= 0.5$
  That means the point is on the line.
- This mean, if any point belongs to the line, its sigmoid value will be 0.5.
- For point above the line, Sigmoid will be greater then 0.5.
- For point below the line, sigmoid will be less then 0.5.

In this way, We converted the classification into a gradient (Continuous Value).

So now, for any given point we can say what is the probability of being positive.

for example

- if the probability of being positive is 0.6, then the probability of being negative will be 0.4.

This means we converted the whole pre-classification region into a probabilistic gradient.

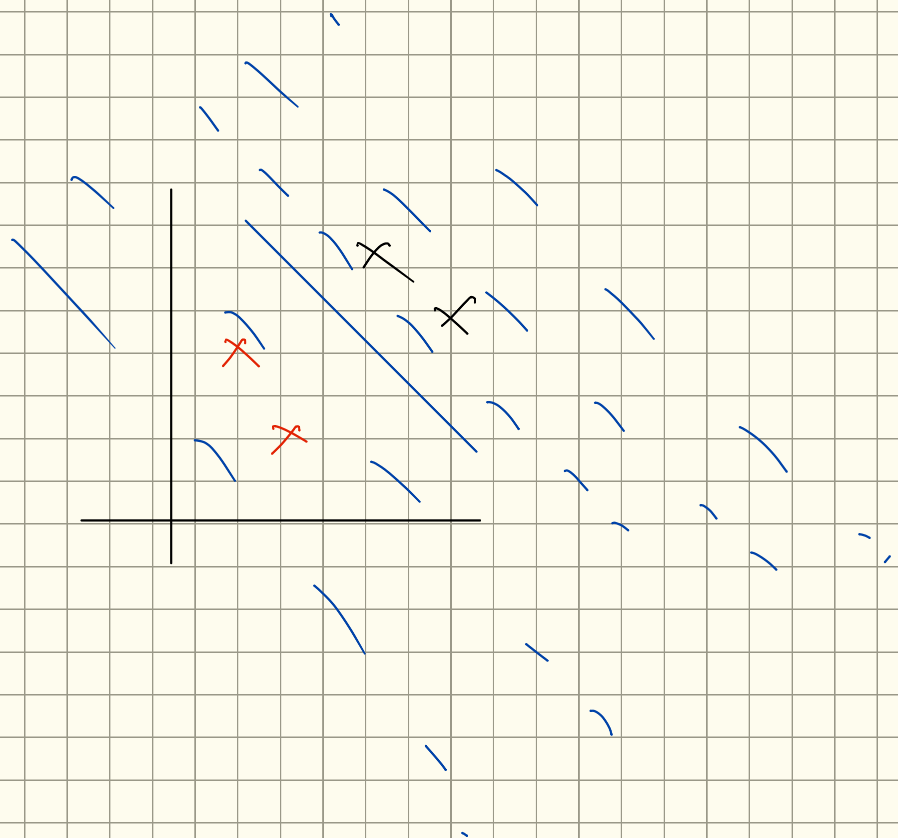
→ Probability of being positive above the line is > 0.5.

→ probability of being negative below the line is < 0.5.

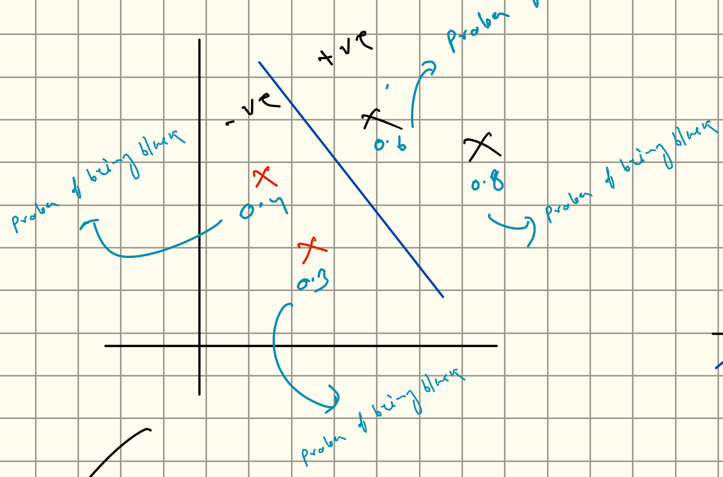
Earlier we had a problem:-

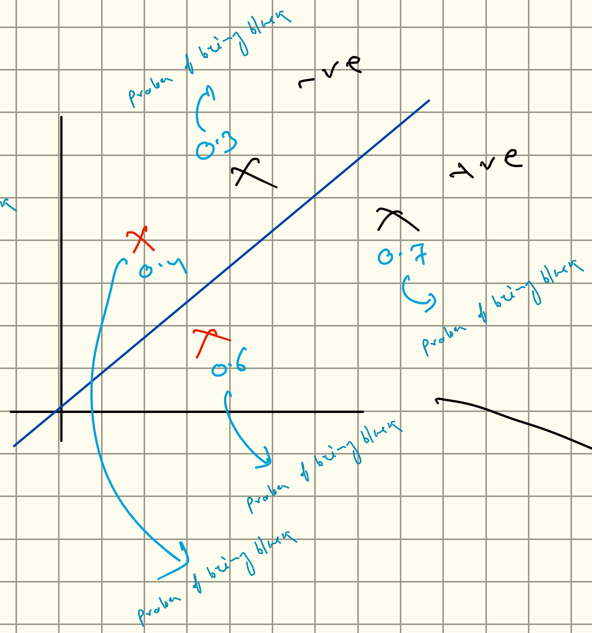
- Above the line we were directly saying 1
- Below the line we were directly saying 0.

Now with sigmoid, we solved this problem by giving probabilities instead of fixed 0 or 1.



Maximum Likelihood





⇒ The likelihood function is the product of the predicted probabilities for the actual class of each observation

$1-0.3$ $1-0.4$

$\boxed{0.8 \times 0.6 \times 0.7 \times 0.6 = 0.20}$

Probability

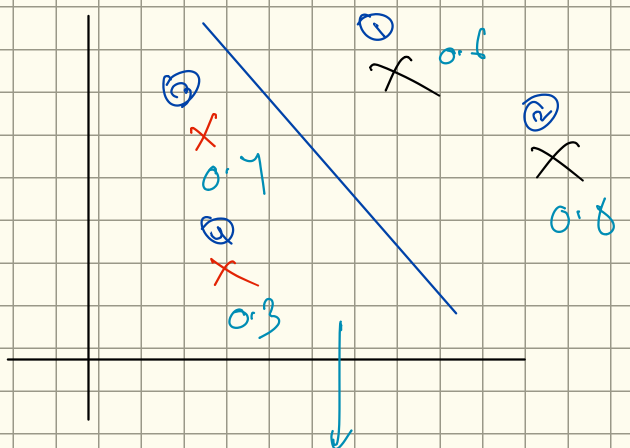
$\boxed{\text{Likelihood}}$

$\boxed{0.7 \times 0.4 \times 0.3 \times 0.6 = 0.05}$

Likelihood

⇒ Model 1 is better than Model 2 because the likelihood of Model 1 is greater than the likelihood of Model 2.

Log Loss



→ there are only 4 points

$= 0.20$

maximum likelihood $=$ $\boxed{0.8 \times 0.6 \times 0.6 \times 0.7}$ → Problem

log ↳

$$\log(ML) = \log(0.8 \times 0.6 \times 0.6 \times 0.7)$$

$$= \log 0.8 + \log 0.6 + \log 0.6 + \log 0.7$$

$$= \boxed{-\log 0.8 - \log 0.6 - \log 0.6 - \log 0.7}$$

positive

$\log(0-1)$

namesh −ve hota hai

isliye ham bahar se −ve use karenge

$$\boxed{= -\log(\hat{y}_1) - \log(\hat{y}_2) - \log(\hat{y}_3) - \log(\hat{y}_4)}$$ No

$\hat{y}_1 = P(\text{green})$ black

$1-\hat{y}_1 = P(\text{red})$

$\hat{y}_1 = \sigma(z) \rightarrow (P) \rightarrow$ getting green

$\hookrightarrow z = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

$$\boxed{-\log(\hat{y}_1) = -y_1 \log(\hat{y}_1) - (1-y_1)\log(1-\hat{y}_1)}$$

minimize

$$L = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$$

→ log loss error

↳ binary cross entropy

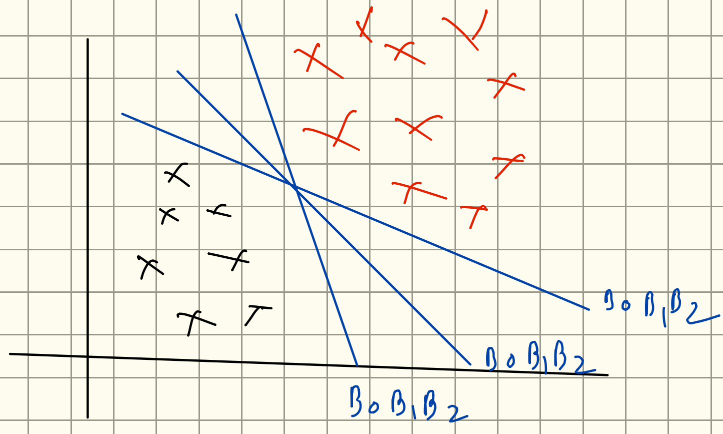
⇒ We use log because in a model we don't just have 4-5 points, we usually have thousand of points. if we multiply all the probability together, the result will become extremely small and close to zero. This can cause underflow problem. To avoid this, we take the log of probabilities.

Note :-

Earlier, we were maximizing likelihood directly, but when we use log, we actually minimize instead of maximize. This is because values of $-\log(P)$ works in reverse.

For example

$-\log(0.1)$ is 1 while $-\log(0.5)$ is 0.3, that means, the higher the probability, the smaller the $-\log$ value. So instead of maximizing likelihood, we minimize negative log-likelihood.



→ Out of the three lines, we will choose the one for logistic regression which gives the minimum loss

Gradient Descent

$$\boxed{L = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)}$$

	X_1	X_2	Y	$\hat{Y}_i$	
	28	13	0	0.63	→ 0.37
	15	12	1	0.37	→ 0.63
	.	.	.	.	
	.	.	.	.	
$100 = n$					

$\hat{Y}_1 \rightarrow p(\text{green})$ ↗ black

$(1-\hat{Y}_1) \rightarrow p(\text{red})$