LOGISTIC REGRESSION

Pre-Requisite : Data should be linearly (or almost) separable

Perceptron Trick

$$W_{new} = W_{old} + \eta (Y_i - \hat{Y}) X_i$$

η (learning rate)

1000 times iteration करके Best Line

BUT Problem is overfitting on training data and under-performance on test data.

⇓

Logistic Regression

why because it works only when something is wrongly classified, if it's correctly classified we do nothing.

- If mis-classified, we were moving the line towards that point (PULL) else do nothing.

Now in Logistic Regression

If point has been correctly classified, then PUSH the line, if point has been incorrectly classified then PULL the line ⇒ llgbm will be achieved.

	Magnitude	
PULL	low	Misclassified Pt Near to Line
	High	" " far off " "
PUSH	High	Classified Pt Near to Line
	low	" " far off " "

<u>Till Now</u> $\quad W_n = W_0 + \eta \underbrace{(Y_i - \hat{Y}_i)}_{\hookrightarrow 0 \text{ or } 1} X_i$

But if it is 0 or 1; then again we will reach to Perceptron trick $\Rightarrow$ i don't want 0 or 1 rather I want some other number

$\Downarrow$

Y_i is fixed it's either 0 or 1

$\hat{Y}_i$ = we are calculating; using

$\rightarrow$ we can only change this.

$\Downarrow$

USE SIGMOID f^n to calculate $\hat{Y}_i$

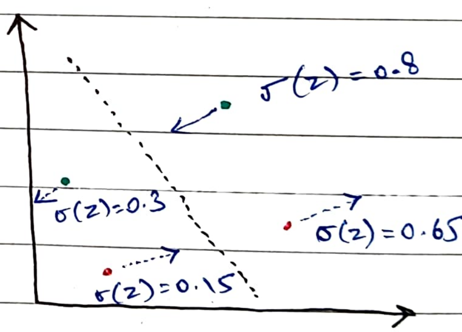
How were we calculating, $\hat{Y}_i$

$$\hat{Y}_i = \sum_{i=0}^{n} w_i x_i$$ then we were using step $f^n = \begin{cases} >0 & 1 \\ <0 & 0 \end{cases}$

<u>Now</u> using <u>sigmoid</u>

$$\hat{Y}_i = \sigma(z)$$

where $z = \sum w_i x_i$



e.g.,

Y_i	$\hat{Y}_i$	$Y_i - \hat{Y}_i$
1	0.8	0.2
0	0.65	-0.65
1	0.3	0.7
0	0.15	-0.15

$W_n = W_0 + \eta(0.2) X_i$ (Push)

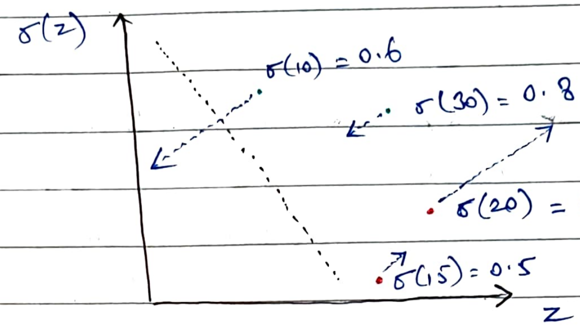
$W_n = W_0 - \eta(0.65) X_i$ (Pull)

$W_n = W_0 + \eta(0.7) X_i$ (~~Push~~ Pull)

$W_n = W_0 - \eta(0.15) X_i$ (Push)

adding something $\Rightarrow$ line पॉइंट की ओर

we want to test magnitude



Y_i	$\hat{Y}_i$	$Y_i - \hat{Y}_i$
1	0.6	0.4
1	0.8	0.2
0	0.7	-0.7
0	0.5	-0.5

$W_n = W_0 + \eta (Y_i - \hat{Y}_i) X_i$ where $\hat{Y}_i = \sigma(z)$ and $\sigma(z) = \Sigma w_i X_i$

$$\left.\begin{aligned} W_n &= W_0 + \eta(0.4) X_i \rightarrow (1) \\ W_n &= W_0 + \eta(0.2) X_i \rightarrow (2) \end{aligned}\right\}$$ clearly (1) will push more

$\Downarrow$

Near point, correctly classified

$\Rightarrow$ MORE FORCE

$$\left.\begin{aligned} W_n &= W_0 - \eta(0.7) X_i \rightarrow (3) \\ W_n &= W_0 - \eta(0.5) X_i \rightarrow (4) \end{aligned}\right\}$$ eq (3) will pull more

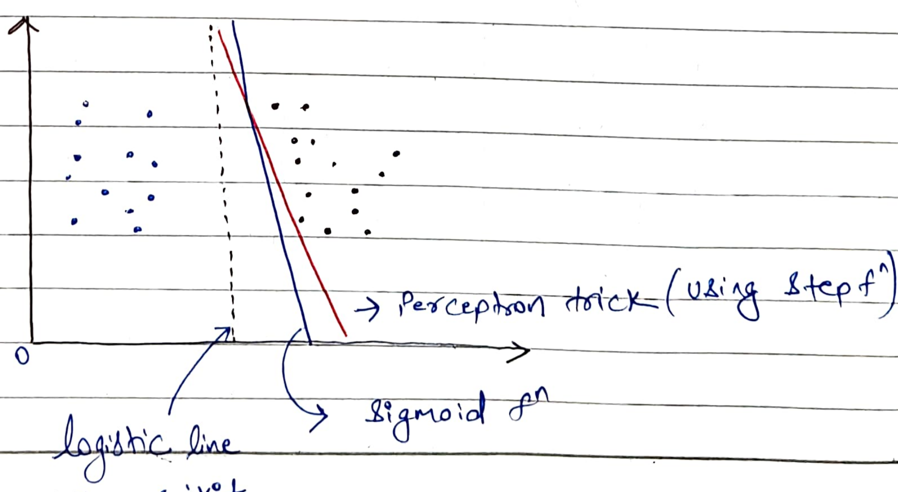
$\Downarrow$

Far point, incorrectly classified

$\Rightarrow$ MORE FORCE

STILL, Not matching ; logistic Regression with Scikit learn method

CRUX



We have got a random line, But to find Best fit line

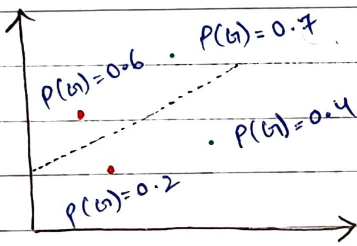
$\Downarrow$

Minimise error

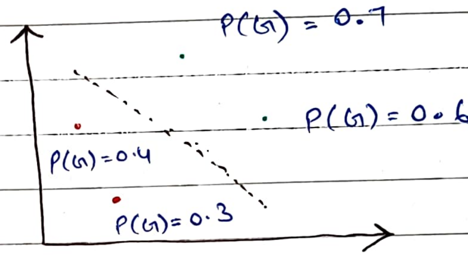
$\Downarrow$

find loss f^n or error f^n

How to find loss function ?



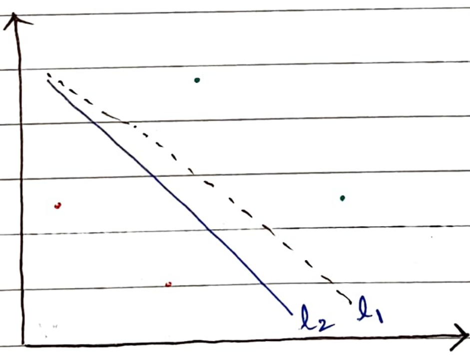
model-1



model-2

visually this model is better

But, visually may not be always Possible



out of line-1 & line-2 which is better ?

$\Downarrow$

loss f^n can tell

$\Downarrow$

To fetch loss f^n use MLE (Maximum Likelihood)

$\Downarrow$

multiply the Prob of all the actual data point and whichever gives max Prob is the best

$$\text{model-1} = \underset{P(G)}{0.7} \times \underset{P(G)}{(0.4)} \times \underset{P(R)}{(0.4)} \times \underset{P(R)}{(0.8)}$$

$$= 0.0889$$

$$\text{model-2} = 0.7 \times (0.6) \times (0.6) \times (0.7)$$

$$= 0.1764$$

Model-2 better

$\because MLE_2 > MLE_1$

This gives us Answer
BUT, Problem is after multiplication numbers will be vv small incase there are even 1000 points.

$\Downarrow$

use summation ~~after~~ using log

$\because \log(ab) = \log a + \log b$

Model 1 = $0.7 \times 0.4 \times 0.4 \times 0.8$

$\log(\text{Max}) = \log(0.7) + \log(0.4) + \log(0.4) + \log(0.8)$

Problem : $\log x = -ve$ when $x \in (0,1)$
to make $\log(\max)$ +ve

$\Downarrow$

CROSS ENTROPY

$\log(\text{Max}) = -\log(0.7) - \log(0.4) - \log(0.4) - \log(0.8)$

In MLE we were maximising
while In Cross Entropy we need to minimise

~~$\because \log(0.7) > \log(0.9)$~~ $\because f(x)$ $\uparrow$ing cont f^n
$\Rightarrow \log(f(x))$ is also $\uparrow$ing cont. f^n
$\Rightarrow$ minimise $-\log(f(x))$

can I write loss f^n as

$L = -\log(\hat{Y}_1) - \log(\hat{Y}_2) - \log(\hat{Y}_3) - \log(\hat{Y}_4)$
✓ ✓

NO $\because$ here we are calculating the Prob of actual event ~~not of Predicted~~
Red $\forall$ $\exists$ Red
Green $\forall$ $\exists$ Green

Hence;

$$\text{loss } f^n = \sum_{i=1}^{n} -Y_i \log(\hat{Y}_i) - (1-Y_i)\log(1-\hat{Y}_i)$$

$\because$ we want to find Avg Error

$$\boxed{\text{loss } f^n = -\frac{1}{n} \sum_{i=1}^{n} Y_i \log(\hat{Y}_i) + (1-Y_i)\log(1-\hat{Y}_i)}$$

↳ log loss error
↳ Binary cross Entropy

$\nexists$ any close form sol^n unlike, in linear Regression
$\Rightarrow$ Minimize error using Gradient Discent.

GRADIENT DISCENT

Hypothetical e.g.,
rows = m
$col^n = n$

$$\begin{matrix} x_{11} & x_{12} & \cdots & x_{1n} & y_1 \\ x_{21} & x_{22} & \cdots & x_{2n} & y_2 \\ \vdots & & & & \\ x_{m1} & x_{m2} & \cdots & x_{mn} & y_m \end{matrix}$$

we know If we have n input column
then we have $(n+1)$ coefficient
assuming coefficients are $(w_0, w_1, w_2, \ldots, w_n)$

$$\hat{y}_i = \sigma(z) \qquad \text{where} \quad z = \sum w_i x_i$$

$$\hat{y}_1 = \sigma(w_0 + w_1 x_{11} + w_2 x_{12} + w_3 x_{13} + \cdots + w_n x_{1n})$$

$$\hat{y}_2 = \sigma(w_0 + w_1 x_{21} + w_2 x_{22} + \cdots + w_n x_{2n})$$

$\vdots$

$\hat{y}_m$

writing in terms of matrices

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} \sigma(w_0 + w_1 x_{11} + w_2 x_{12} + \dots + w_n x_{1n} \\ \sigma(w_0 + w_1 x_{21} + w_2 x_{22} + \dots + w_n x_{2n} \\ \vdots \\ \sigma(w_0 + w_1 x_{m1} + \dots + w_n x_{mn} \end{bmatrix}$$

Taking σ common and then separating w_i and x_{ij} in dot Product format

$$\hat{Y} = \sigma\left(\begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & & & & \\ 1 & x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \right)$$

$$\boxed{\hat{Y} = \sigma(XW)}$$

From loss f^n we know

$$L = -\frac{1}{m} \sum_{i=1}^{m} Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)$$

$$L = -\frac{1}{m} \left[\sum_{i=1}^{m} Y_i \log(\hat{Y}_i) + \sum_{i=1}^{m} (1 - Y_i) \log(1 - \hat{Y}_i) \right]$$

our aim is to write it in matrix form.

WAL

$$\sum_{i=1}^{m} Y_i \log(\hat{Y}_i) = Y_1 \log \hat{Y}_1 + Y_2 \log \hat{Y}_2 + \dots + Y_m \log \hat{Y}_m$$

$$= [Y_1 \ Y_2 \dots Y_m] \begin{bmatrix} \log(\hat{Y}_1) \\ \log(\hat{Y}_2) \\ \vdots \\ \log(\hat{Y}_m) \end{bmatrix}$$

$$= \underbrace{[Y_1 \ Y_2 \dots Y_m]}_{Y} \log \left(\underbrace{\begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_m \end{bmatrix}}_{\hat{Y}} \right)$$

$$= Y \log \hat{Y}$$

$$= Y \log(\sigma(XW))$$

Now my loss f^n becomes

$$L = \frac{-1}{m} \left[Y \log \hat{Y} + (1-Y) \log(1-\hat{Y}) \right]$$

where $\hat{Y} = \sigma(XW)$

we can't solve this
⇒ use Gradient descent to find [W] such that loss f^n is minimum

loss function in Matrix form

$$L = -\frac{1}{m}\left[Y \log(\sigma(WX)) + (1-Y)\log(1-\sigma(WX))\right]$$

To minimise L, we will use gradient descent to find all the coefficient

- we will initialise [W] with any value

$$W = [\qquad\qquad]$$

- Run loop ; for i in epochs

$$W = W - \eta \frac{\Delta L}{\Delta W}$$

Learning rate

where

$$\frac{\Delta L}{\Delta W} = \left[\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \ldots \frac{\partial L}{\partial w_n}\right]$$

Now our aim is to find $\frac{\Delta L}{\Delta W}$

$$L = -\frac{1}{m}\left[\underbrace{Y \log \hat{Y}}_{I} + \underbrace{(1-Y)\log(1-\hat{Y})}_{II}\right]$$

taking I^{st}

$$\frac{dL}{dW} = Y \frac{1}{\hat{Y}} \frac{d(\hat{Y})}{dW}$$

$\because \hat{Y} = \sigma(WX)$

$\& \frac{d}{dW}(\sigma(WX)) = \sigma(WX)[1-\sigma(WX)] X$

$$= \frac{Y}{\hat{Y}} \sigma(WX)[1-\sigma(WX)] \cdot X$$

$$\frac{dL}{dW} = Y(1-\hat{Y})X$$

Taking derivative of II

$$\frac{d}{dw}(1-Y)\log(1-\hat{Y}) = \frac{(1-Y)}{(1-\hat{Y})}\frac{d}{dw}(1-\hat{Y})$$

$$\Rightarrow \frac{1-Y}{(1-\hat{Y})}(-)\frac{d}{dw}\sigma(wx)$$

$$\Rightarrow -\frac{(1-Y)}{(1-\hat{Y})}\left[\sigma(wx)(1-\sigma(wx))\right]\frac{d}{dL}(wx)$$

$$\Rightarrow -\frac{(1-Y)}{(1-\hat{Y})}\hat{Y}(1-\hat{Y})\,X$$

$$\Rightarrow -\hat{Y}(1-Y)X$$

Combining I and II

$$\frac{dL}{dw} = -\frac{1}{m}\left[Y(1-\hat{Y})X - \hat{Y}(1-Y)X\right]$$

$$= -\frac{1}{m}\left[Y - Y\hat{Y} - \hat{Y} + \hat{Y}Y\right]X$$

$$\boxed{\frac{\Delta L}{\Delta W} = -\frac{1}{m}\left[Y-\hat{Y}\right]X}$$

Now Gradient descent

$$\boxed{W = W + \eta\frac{1}{m}(Y-\hat{Y})X}$$

$$W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}_{(n+1),1} \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & & & & \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{(m,n+1)} \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{bmatrix}_{m,1} \quad \hat{Y} = \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_M \end{bmatrix}_{m,1}$$