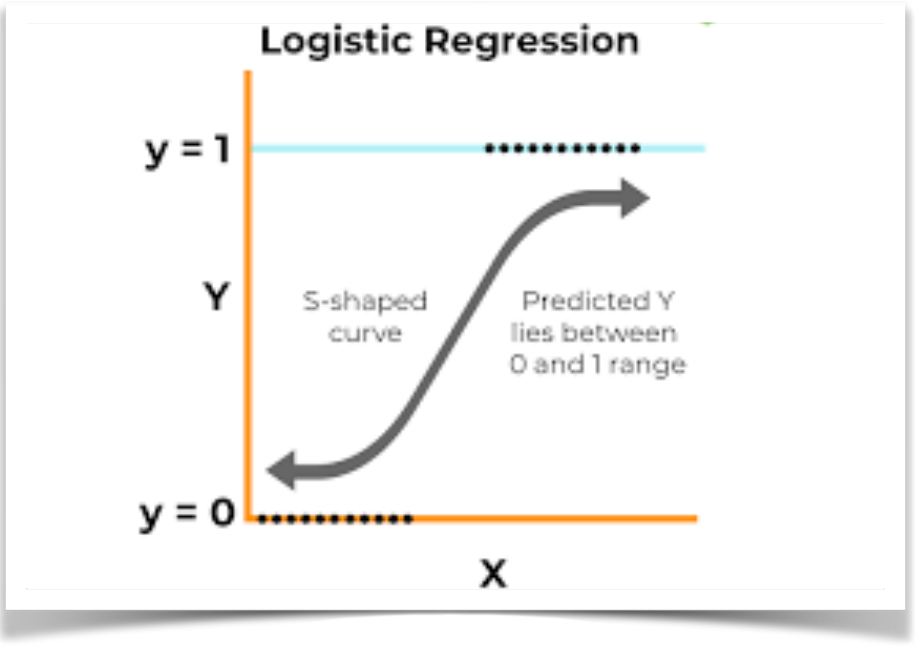
Logistic Regression

[Handwritten Notes (Hindi + English)]

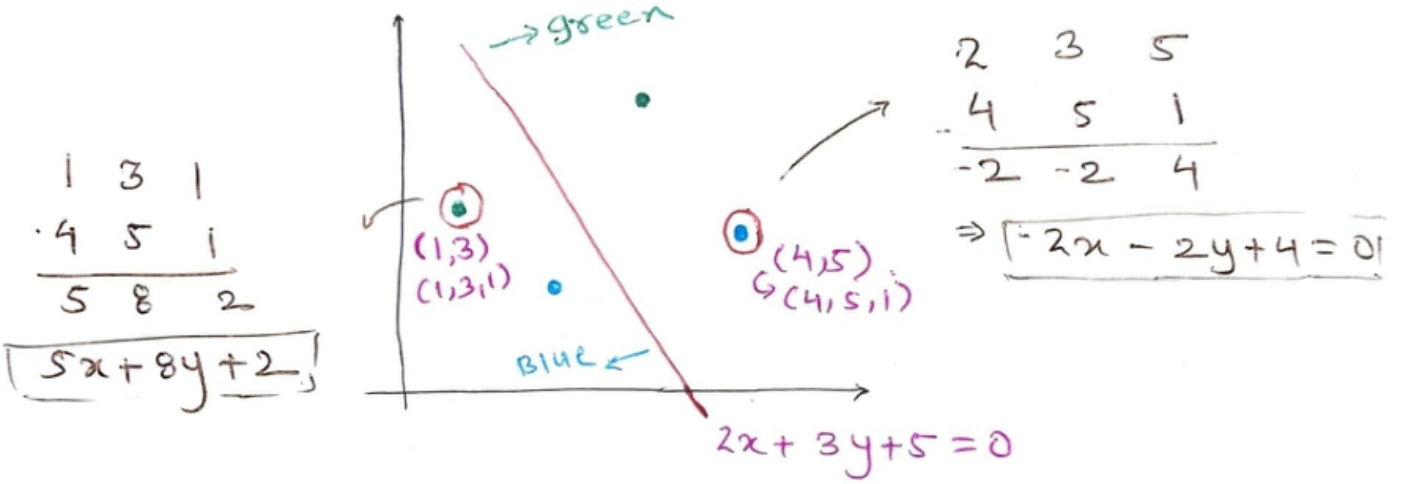
- Divyansh Waghmare



Logistic Regression

* condition → data should be linearly or nearly linearly seprable.

1. Perceptron Method.



There is two misclassified region

* The perception methods say that if we have to move linearly seprable line in +ve region then you have to substract all the cofficient of the line to the point's coff which belongs to -ve region,

And if you have to move line in -ve region then you have to add all the coff of line to the all coff of that point which belong to +ve region.

$$\{ \text{New\_coff} = \text{Coff\_line} - \eta \text{ Coff\_Point}$$

↳ it gradually ↑ with epoch

Let take example to form general eq^n

x_0	(x_1)	(x_2)	y
	CGPA	I.Q	Placed
1	7.5	81	1
1	8.9	109	1
1	7.0	81	0

$Ax + By + C = 0$

$$\boxed{w_0 + w_1 x_1 + w_2 x_2 = 0}$$

$w_0 = C$, $w_1 = A$, $w_2 = B$

$\hookrightarrow w_0 x_0 + w_1 x_1 + w_2 x_2 = 0$

$$\Rightarrow \sum_{i=0}^{2} w_i x_i = 0$$

Let take example,

$w_0 x_0 + (w_1) \times 7.5 + (w_2) \times 81$

calculated by model

if value $> 0 \rightarrow 1$
value $< 0 \rightarrow 0$
} step function

$\Rightarrow$ epoch $\rightarrow$ 1000 ; $\eta = 0.01$

For i in range (epoch):

+ randomly select a row

Mean ye jo student hai uska placement nahi hua par model bola hua hai

$\leftarrow$ If $x_i \in N$ and $\sum_{i=0}^{2} w_i x_i > 0$: (+N)

$w_{new} = w_{old} - \eta [x_0 x_1 x_2]$

x_i (particular row)

$\leftarrow$ If $x_i \in P$ and $\sum_{i=1}^{2} w_i x_i < 0$

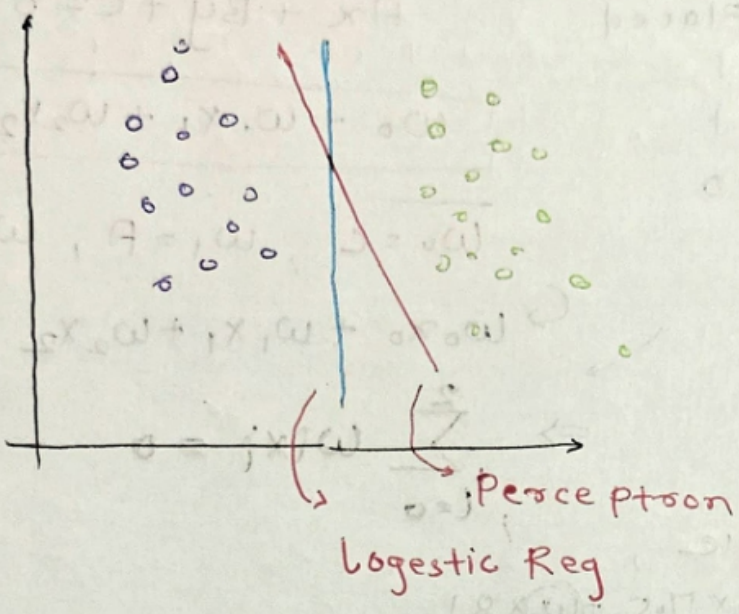
Mean ek student hai jiska placement hua par model bola nahi hua.

$w_{new} = w_{old} + \eta x_i$

$\hookrightarrow$ generalize

$$w_n = w_0 + \eta (y_i - \hat{y_i}) x_i$$

Problem with Perceptron

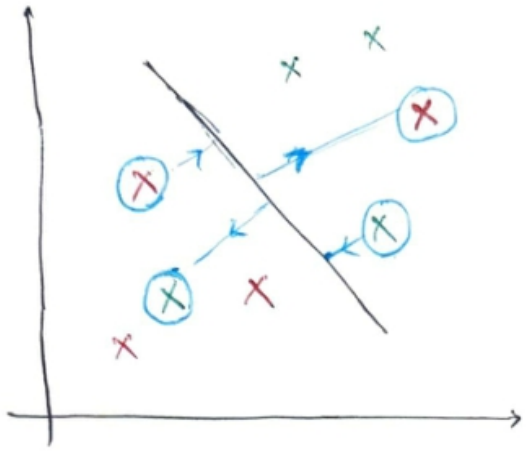


* The perceptron method fails because method only works on miss classified Problem

Agar point mis-classified hua to vo line ko apne taraf kichenga par ek bar sab classification hone k bad vo line move hone bond ho jayege, so overfitting ka case badenga,

hume aise algorithm likhna padega jo har points - pai lage means har point pai line ko move karvaye so margin b/w classification is equal:

Modification of Step f^n to {Sigmoid f^n}



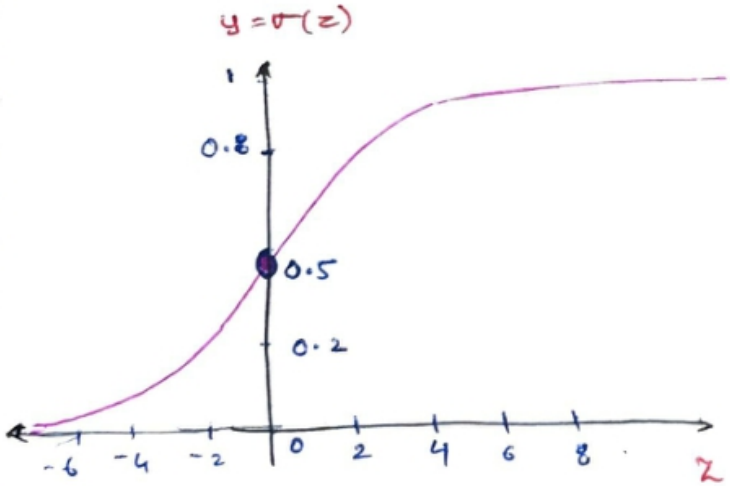
we know,

$$w_n = w_o + \eta(y_i - \hat{y})x_i$$

y_i	$\hat{y}_i$	$y_i - \hat{y}_i$
1	1	0
0	0	0
1	0	1
0	1	-1

we can see that in these two points, points are already in right section classification, means it doesn't make any change in the movement of line that's ~~the~~ we comes to sigmoid f^n.

Sigmoid Function



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

* z is very high means (σ) y thends to 1
* z = 0 means σ = (0.5)

means

$$\sum w_i x_i > 0 \Rightarrow \sigma \geq 0.5$$

$$\sum w_i x_i = 0 \Rightarrow \sigma = 0.5$$

$$\sum w_i x_i < 0 \Rightarrow \sigma < 0.5$$

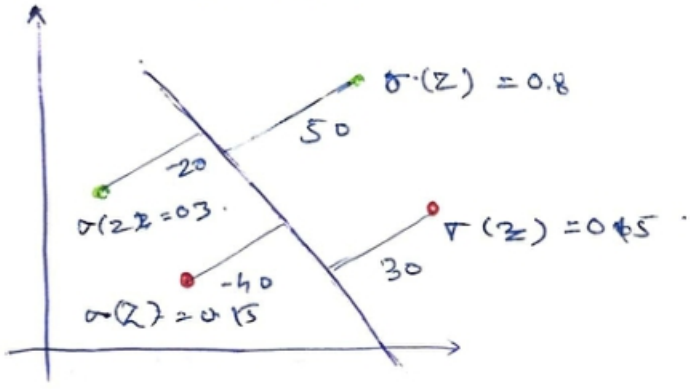
$\sigma() = P()$

* Sigmoid f" kehta hai ki jo value $\sigma(\sum w_i x_i)$ hoge vo cheeze uske hone ki probability hoge

  nai hone ki probability

$1 - \sigma()/P()$

$z = w_i x_i$



$w_n + \eta (y_i - \hat{y}_i) x_i$

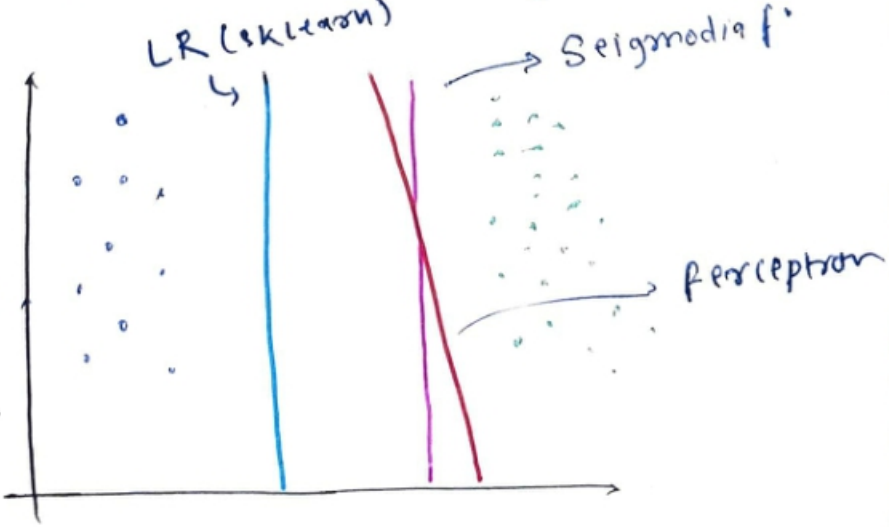
y_i	$\hat{y}_i$	$y_i - \hat{y}_i$
1	0.8	0.2
0	0.65	-0.65
1	0.3	0.7
0	0.15	-0.15

⇐ { no value is zero }

[Means all points make effect on line]

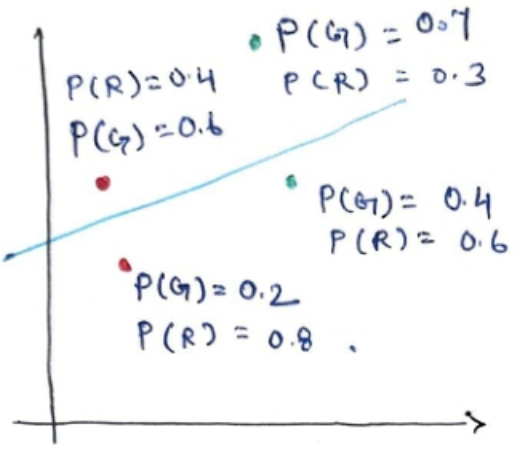
* If a point is closed to line then it will produce more effect on line, if correctly classified the push effect is more, if wrong then pull is more. -|

Still there is not a convegnie

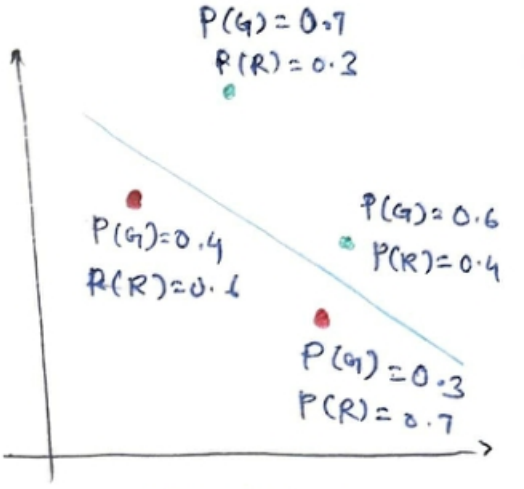


Maximum Likehood & Binary Cross Entropy

Let's take two model.



model 1



model 2

here we can see that model 2 > model 1

Let prove with Maths.

Maximum likelihood (M_1) → $0.7 \times 0.4 \times 0.4 \times 0.8 = 0.086$

Maximum likelihood (M_2) → $0.7 \times 0.6 \times 0.6 \times 0.7 = 0.176$

$$M_2 > M_1$$

Means we need to select Model 2 because it has greater Maximum likelihood but we can see that we are multipling smaller numbers so if there is large number in model then there MLH will be even more smaller.

So what we do,

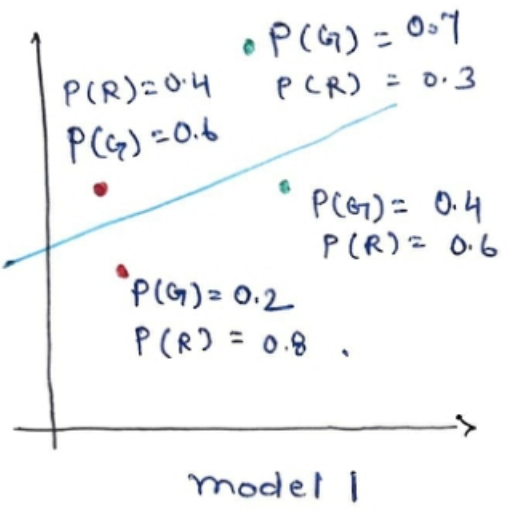
we will take log of Maximum likelihood.

we know,

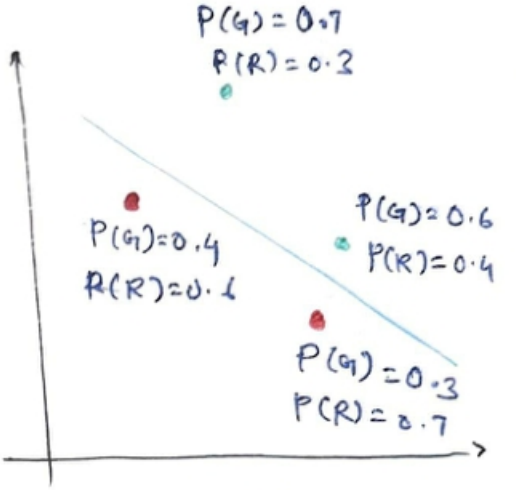
$$\log(ab) = \log a + \log b$$

Maximum Likehood & Binary Cross Entropy

Let's take two model.



model 1



model 2

here we can see that model 2 > model 1

Let prove with Maths.

Maximum liklehood (M_1) → $0.7 \times 0.4 \times 0.4 \times 0.8 = 0.086$

Maximum liklehood (M_2) → $0.7 \times 0.6 \times 0.6 \times 0.7 = 0.176$

$$M_2 > M_1$$

Means we need to select Model 2 because it has greater Maximum likelihood but we can see that we are multipling smaller number so if there is large number in model then there MLH will be even more smaller.

So what we do,

we will take log of Maximum likelihood.

we know,

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

Since the log of number b/w 0 -1 (0.1 —) is -ve so we take -ve sum of maximum likelihood.

$$\log(\max) = -\log(0.7) - \log(0.4) - \log(0.4) - \log(0.8.$$

And this is called Cross entropy!

taking the "The sumisation of -ve log of Maximum likelihood is called Cross entropy

* we have to minimize the cross entropy Because log for smaller number big

$$\log(0.1) > \log(0.8)$$

* we have to select that model who has lesser cross entropy or minimum cross entropy.

→ How can we write it into loss function

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

This is wrong because $\hat{y}_1$ → probality of both green & red

So,

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

have both probil.

Minimizing the Loss f^n

(Gradient Descent)

Let,

X				Y	$\hat{y}$
x_{11}	x_{12}	---	x_{1n}	y_1	$\hat{y}_1$
x_{21}	x_{22}	---	x_{2n}	y_2	
⋮	⋮		⋮	⋮	
x_{m1}	x_{m2}	---	x_{mn}	y_m	

$\{w_1, w_2, \cdots w_n\}$
w_0

1st row

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

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

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

(X) (W)

$$\boxed{\hat{y} \Rightarrow \sigma(XW)}$$

We know, Loss f^n

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

↳ Let's write it into matrix form

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

$$\sum_{i=1}^{m} y_i \log(\hat{y}_i) = y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 \; - - - \; y_m \log \hat{y}_m$$

$$\Rightarrow [y_1 \; y_2 \; y_3 \; \cdots \; y_n] \log \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \cdot \\ \hat{y}_n \end{bmatrix}$$

Y $\qquad$ $\hat{Y}$

$$\Rightarrow Y \log \hat{Y}$$

$$\Rightarrow \boxed{Y \log(\sigma(XW))}$$

$$\sum_{i=1}^{n} (1-y_i)\log(1-\hat{y}_i) = \boxed{(1-y)\log(1-\sigma(wx))}$$

Now Loss eq^n becames,

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

(W: minimum)

* Now we have to make this f^n min

$$w_4 = w_0 - \frac{\Delta L}{\Delta w} = \left[\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2} \; - - \; \frac{\partial L}{\partial w_4}\right]$$

For Minimizing the Loss f^n we to partial diff it with its respective coefficient

↳

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

$$\frac{dL}{dw} = \frac{d}{dw} y \log \hat{y} \Rightarrow y \frac{d}{dw} \log \hat{y} \Rightarrow \frac{y}{\hat{y}} \frac{d}{dL}(\hat{y})$$

$$\Rightarrow \frac{y}{\hat{y}} \frac{d}{dL} \sigma(wx)$$

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

$$\Rightarrow \frac{y}{\hat{y}} \hat{y}(1-\hat{y}) x = y(1-\hat{y})x$$

$$\Rightarrow \boxed{y(1-\hat{y})x}$$

2nd

$$\frac{d}{dw}(1-y)\log(1-\hat{y}) \Rightarrow (1-y)\frac{d}{dw}\log(1-\hat{y})$$

$$\Rightarrow \frac{(1-y)}{(1-\hat{y})} \frac{d}{dw}[1-\hat{y}]$$

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

$$\Rightarrow -\frac{(1-y)}{(1-\hat{y})} \hat{y}(1-\hat{y}) x$$

$$\Rightarrow \boxed{-\hat{y}(1-y)x}$$

↳

So,

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

$$= \frac{-1}{m}\left[y - y\hat{y} - \hat{y} + y\hat{y}\right]x$$

$$\frac{\Delta L}{\Delta w} \Rightarrow \frac{-1}{m}\left[y - \hat{y}\right]x$$

According to Gradient descent.

$$w_{New} = w_{Old} + \eta \frac{1}{m}(y - \hat{y})x$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}, \quad x = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & & & & \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

$\rightarrow (n+1, 1)$ $\qquad$ $(m, n+1)$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$(m, 1)$ $\qquad$ $(m, 1)$

$$w_1 = w + \frac{\eta}{m}(y - \hat{y})x$$

$(n+1, 1)$ $\quad$ $(n+1, 1)$ $\quad$ $(1, m)\ ((m, (n+1)))$

$(1, n+1)$

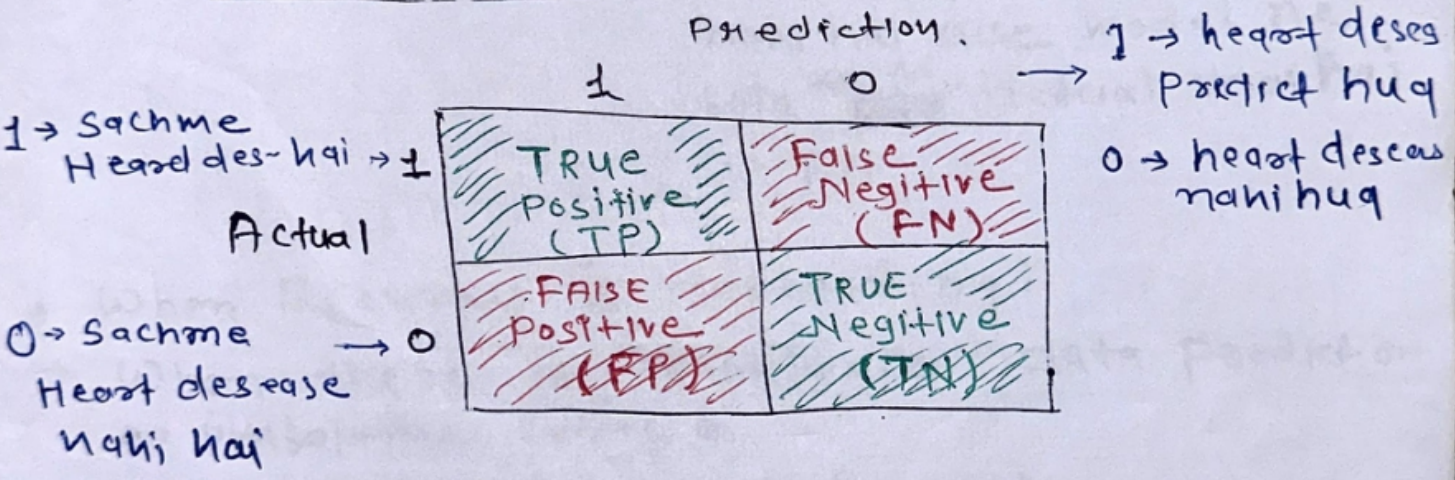
Accuracy:-

$$\text{Accuracy} = \frac{\text{Right Prediction}}{\text{Total Num of Prediction}}$$

* It is the problem in which Accuracy is dependent.

* Problem with Accuracy, it dosen't tell nature of mistake

↳ Confusion Matrix.



* Confusion matrix tells the nature of mistakes

1. Agar actually mai 1 and model bhi 1 bole → TP
2. Agar actuall mai 0 per model 1 bole → FP
3. Agar actual mai 1 hai par model 0 bole → FN
4. Agar actual mai 0 hai or model 0 bole → TN

Accuray through CM →.

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

TYPE I ERROR (FP) → . Ex :- Agar koi Aisa patient hai jise heart disease hai Aisa model ne bola par actual mai nahi hai → FP.

TYPE II ERROR (FN) → Ex :- Agar kai Aisa patient hai jise heart disease nahi hai aise model ne bola ~~Par~~ fer actual mai hai → FN

* When Accuray is misleading?

→ When there is imbalance data prediction or imbalance labels.

→ Ex. take Terrorist Example

In an airport there is only one terrorist and ggg people are G.P so.

CF →

	0	1
0	0	1
1	0	999

TP.

So,

$$Acc = \frac{999}{999 + 1 + 0 + 0}$$

$$= 99\%$$

Acc. Normal People

Precision

* Case study of Email Spam classifier.

	Sent to spam	No sent to spam
Spam	100	170 FN
Not spam	30 FP	700

Acc → 80%.

	Sent to spam	No sent to spam
spam	400	190 FN
Not spam	10 FP	700

Acc → 80%.

$FP_A > FP_B$ | $FN_A < FN_B$

$30 > 10$

FP → Model ne not spam email ko spam bol deya

FN → Model ne spam wale email ko not spam boldey

Here $FP_A > FP_B$.

So 2nd model is performing well

* More precise

* So presion is the proportion of predicted positive to the truly positive

$$P = \frac{TP}{TP + FP}$$

RECALL

	Detected cancer	Not detected
Has cancer	1000	200 FN
No cancer	800 FP	8000

Acc = 90%.

A

	Detect cancer	Not cancer
Has cancer	1000	500 FN
No cancer	500 FP	8000

Acc = 90%.

B

$FN_A < FN_B$

FP → Model ne cancer detect keya par actual mai cancer nahi tha.

FN → Model ne cancer nahi bola par Actual mai cancer tha.

Red alert.

$\Rightarrow FN_A < FN_B$.

So we select model I.

Recall ⇒ Jitne logo ko sahi mai cancer tha. Unme se kitno ko sahi se detect kiya

$$\Rightarrow \frac{TP}{TP + FN}$$

* If TYPE I error is more dengerous then select high Precision model
* If TYPE II error then ↑ Recall model.

Multi-class Precision and Recall

	Dog	Cat	Rabbit	
Dog	25	5	10	→ 40
Cat	0	30	4	→ 34
Rabbit	4	10	20	→ 34
	↓ 29	↓ 45	↓ 34	

Precision

$\Rightarrow P_D = \frac{25}{29}$, $P_C = \frac{30}{45}$, $P_R = \frac{20}{34}$

Macro precission = $\frac{P_D + P_C + P_R}{3} = 0.70$

Weighted Prec = $\frac{40}{108} \times 0.86 + \frac{34}{108} \times 0.66 + \frac{34}{108} \times 0.58$

$\Rightarrow 0.71$

SAME with Recall

F1 Score

Let we have take to identify cat. & dog if we identify dog as 1 and cat as 0 wat if our model identify dog as 0 and cat as 1

* Now we cant identify the model is precise or More Recall.
* In these type of situtions F1 score is given more importance.
* F1 is combo of Precision and Recall

$$F1\ score = \frac{2PR}{P+R}.$$

→ humara jo F1 score Ranenga ~~to~~ vo Precision & Recall mai us ~~k~~ taraf tilted Raheng jiska value kam hai.

Soft MAX REGRESSION

or

MULTINOMIAL Regression

Let take example

cgpa	iq	place
7.1	71	0 → No
8.5	85	1 → Yes
9.5	95	2 → opt out

* If there is more then one category in the output then we go for Softmax regression.

Softmax f^n

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

k = number of classes

(yes)

$$\sigma(z_1) = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

(No)

$$\sigma(z_2) = \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}}, \quad \sigma(z)_3 = \frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

cgpa	iq	place
		0
		1
		2

→ one HOT encoding

=>

0	1	2
1	0	0
0	1	0
0	0	1

→ Three Clowmn in output.

Now create 3 Datase

cgpa	id	0(Place)

L.R M1 $(w_1^{(0)}, w_2^{(0)}, w_3^{(0)})$

cage	id	1(Place)

L.R. M2 (w_1', w_2', w_3')

cgpa	iq	2

L.R. M3 (w_1^3, w_2^3, w_3^3)

M_1 (yes) (w_1', w_2', w_0')

M_2 (NO) (w_1^2, w_2^2, w_0^2)

M_3 (opt out) (w_1^3, w_2^3, w_0^3)

Let take a point
$S_x = \{7, 70\}$.

$Z_3 = 7w_1^{(3)} + 70w_2^{(3)} + w_0^{(3)}$

$Z_1 = 7 \times w_1^{(1)} + 70 \times w_2^{(1)} + w_0^{(1)}$

$Z_2 = 7 \times w_1^{(2)} + 70 \times w_2^{(2)} + w_0^{(2)}$

$$\nabla(y) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\nabla(N) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\nabla(o) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

Let → 0.40 → 0.35 0.25

↳ iska placement honga.

Warning For Big dataset this approach slow Because we have to perform ok LR on all the classes.

→ So we will modify the loss f^n so that it work on Softmax reg.

$$L_{(ori)} = \frac{-1}{m} \sum_{i=1}^{m} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

$$L_{(softmax)} = \frac{-1}{m} \sum_{i=1}^{m} \sum_{i=1}^{k} y_k^{(i)} \log(y_k^{(i)})$$

x_1	x_2	y	$y_{k=1}$	$y_{k=2}$	$y_{k=3}$
x_{11}	x_{12}	1	1	0	0
x_{21}	x_{22}	2	0	1	0
x_{22}	x_{32}	3	0	0	1

Now, in Loss fn (softmax)

$$y_1^{(1)} \log(\hat{y}_1^{(1)}) + y_2^{(1)} \log(\hat{y}_2^{(1)}) + y_3^{(1)} \log(\hat{y}_3^{(1)})$$

(✓ , 0 , 0)

$$+ \; y_1^{(2)} \log(\hat{y}_1^{(2)}) + y_2^{(2)} \log(\hat{y}_2^{(2)}) + y_3^{(2)} \log(\hat{y}_3^{(2)})$$

(0 , ✓ , 0)

$$+ \; y_1^{(3)} \log(\hat{y}_1^{(3)}) + y_2^{(3)} \log(\hat{y}_2^{(3)}) + y_3^{(3)} \log(\hat{y}_3^{(3)})$$

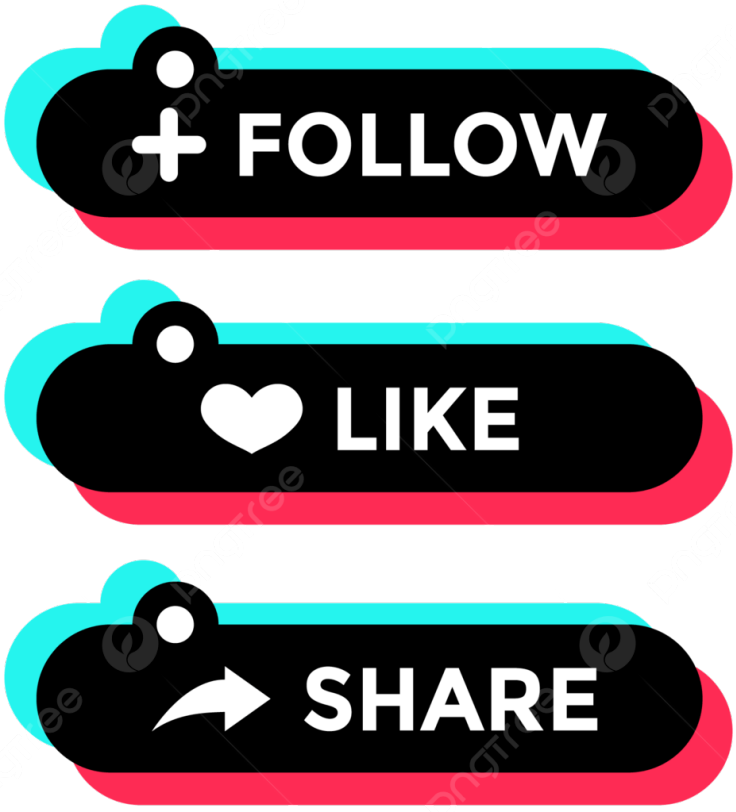
(0 , 0 , ✓)

$$L = y_1^{(1)} \log(\hat{y}_1^{(1)}) + y_2^{(2)} \log(\hat{y}_2^{(2)}) + y_3^{(3)} \log(\hat{y}_3^{(3)})$$

$$\hat{y}_1^{(1)} = \sigma(w_1^{(1)} x_{11} + w_2^{(1)} x_{12} + w_0)$$

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

$$\hat{y}_3 = \sigma(w_1^{(3)} x_{31} + w_2^{(3)} x_{32} + w_0^{(3)})$$


FOLLOW
LIKE
SHARE