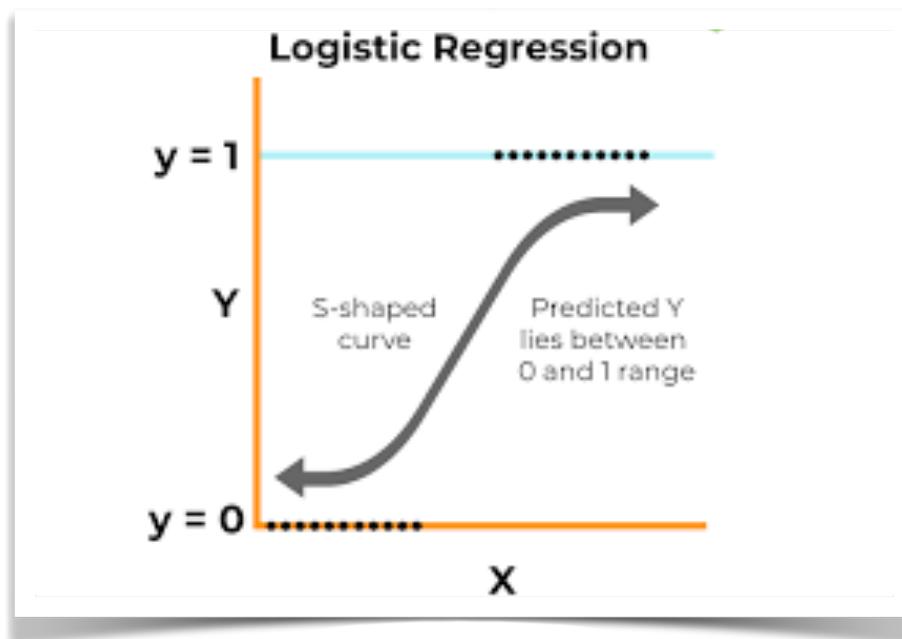


# Logistic Regression

[ Handwritten Notes (Hindi + English) ]

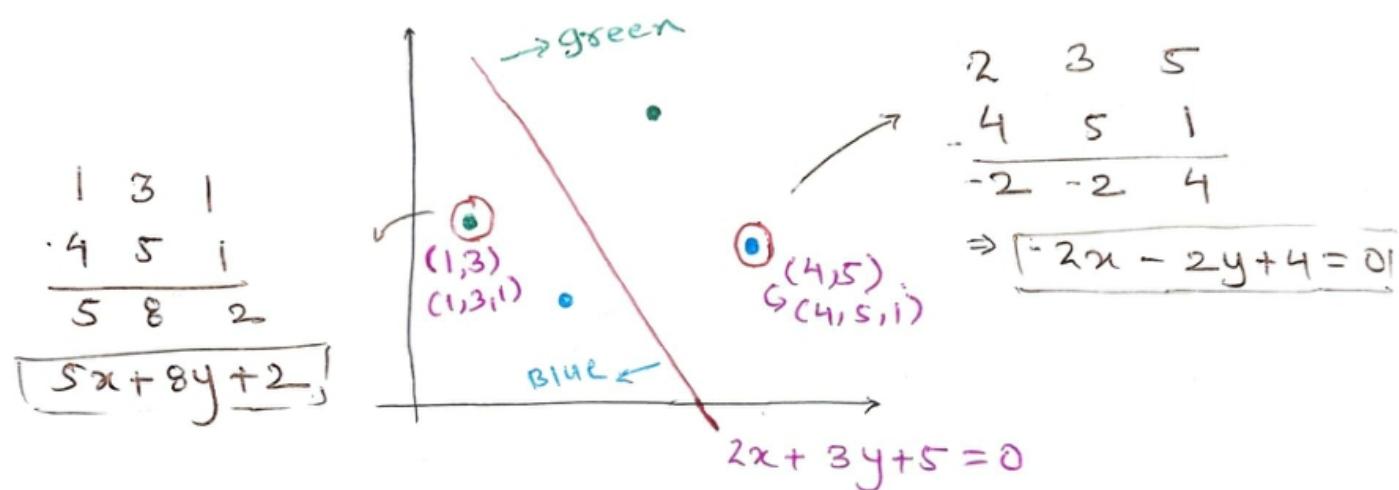
- Divyansh Waghmare



# Logistic Regression

- \* Condition  $\rightarrow$  data should be linearly or nearly linearly separable.

## 1. Perception Method.



There is two misclassified region

- \* The perception methods say that if we have to move linearly separable line in +ve region then you have to subtract all the coefficient of the line to the point's coeff which belongs to -ve region,

And if you have to move line in -ve region then you have to add all the coeff of line to the all coeff 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 eqn

$x_0$ : (1)	$(x_1)$	$(x_2)$	y
1	GPA	i.g	Placed.
1	7.5	81	1
1	8.9	log	1
1	7.0	81	0

$$Ax + By + C = 0$$

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

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

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

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

Let take example,

$$w_0 x_0 + w_1 x_1 + w_2 x_2$$

Calculated by model

if value  $> 0 \rightarrow 1 \quad \} \text{ step function}$

value  $< 0 \rightarrow 0$

$\Rightarrow$  Epoch  $\rightarrow 1000, \eta = 0.01$

for i in range(epoch):

Mean ye jo student hai  $\leftarrow$  randomly select a row uska placement nahi hoga par model bolta hua hai  $w_{new} = w_{old} - \eta [x_0, x_1]$

Mean ek student hai jiska placement hua par model bolta hua hoga,  $\leftarrow$   $\left[ \begin{array}{l} \text{if } x_i \in P \\ \text{and } \sum_{i=0}^2 w_i x_i > 0 \end{array} \right]$

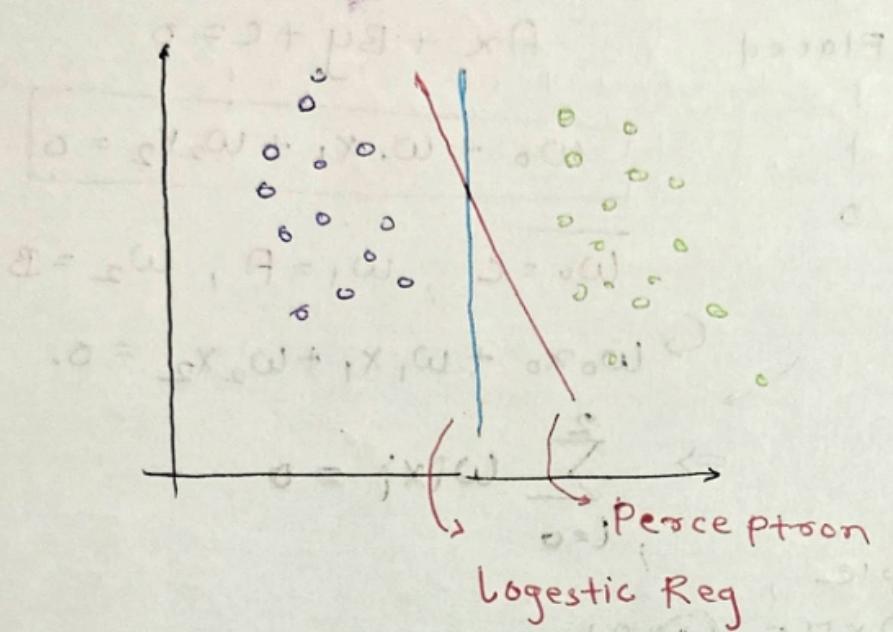
Mean ek student hai jiska placement hua par model bolta hua hoga,

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

generalize

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

## Problem with Perceptron



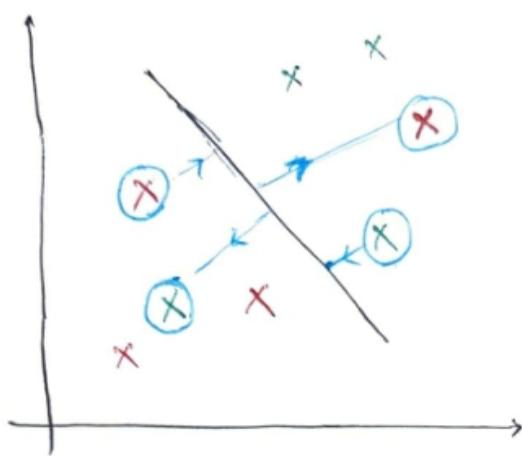
- \* The Perceptron method fails because method only works on miss classified Problem

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

Hume aise algorithm likhna padega  
Jo har points - pair lage means har point  
pair line ko move karaye so margin  
b/w classification is equal.

$$\mathbf{w}^T(\mathbf{x}_i - \mathbf{w}) + b = 0$$

# Modification of Step $f^n$ to Sigmoid $f^n$



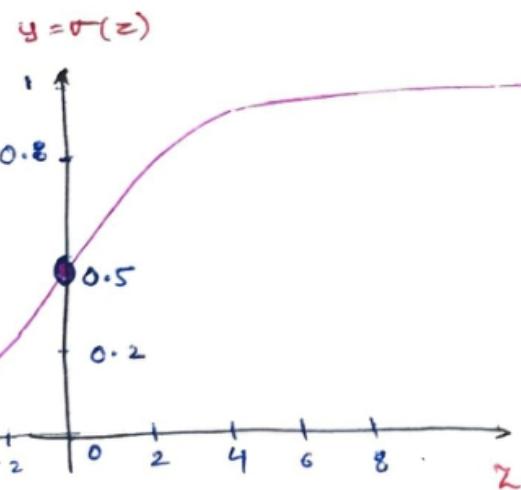
we know,

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

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

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 why we comes to sigmoid  $f^n$ .

## Sigmoid Function



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

+  $z$  is very high means  $\sigma$ .  $y$  tends to 1

+  $z = 0$  means  $\sigma = (0.5)$ .

means

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

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

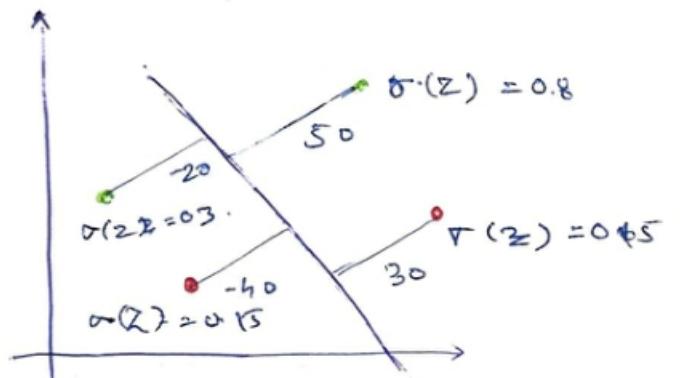
$$\sum w_i x_i < 0 \Rightarrow \sigma \leq 0.5$$

$$\boxed{f(\cdot) = P(\cdot)}$$

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

$$\boxed{1 - f(\cdot)/P(\cdot)}$$

$$z = \sum w_i x_i$$



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

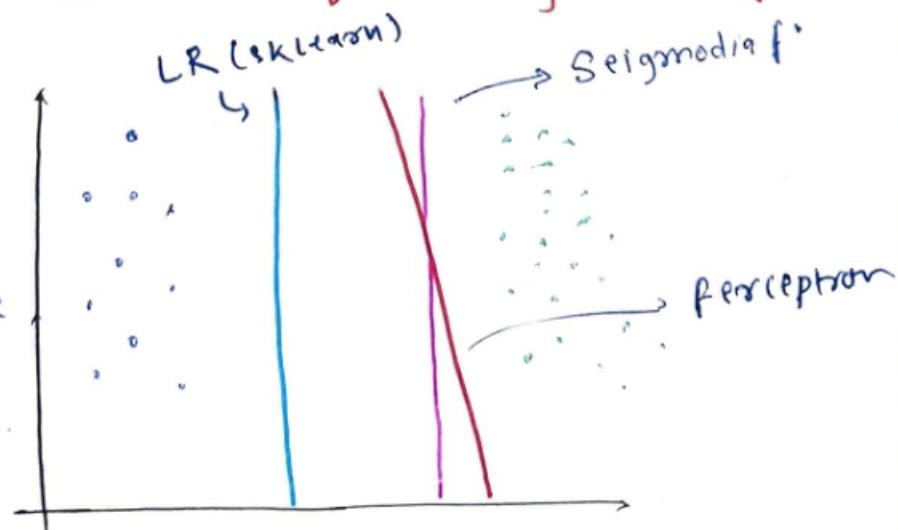
$y_i$	$\hat{y}_i$	$y_i - \hat{y}_i$
1	0.8	0.2
0	0.65	-0.65
1	0.3	0.7
0	0.15	-0.15

Means all points make effect on line

$\Leftrightarrow$  { no value is zero }  $\uparrow$

- \* 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. -

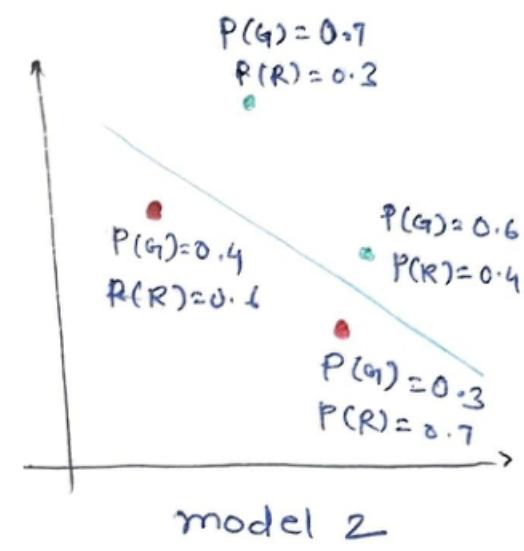
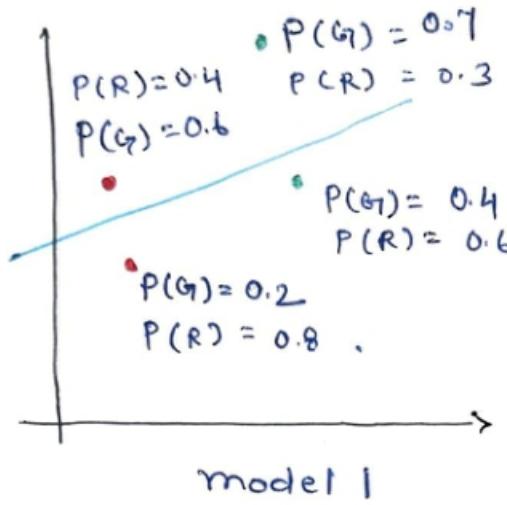
Still there is not a converge



# Maximum Likelihood

## & Binary Cross Entropy

Let's take two model.



here we can see that  $\text{Model 2} > \text{Model 1}$

Let prove with Maths.

$$\text{Maximum likelihood } (\text{M}_1) \rightarrow 0.7 \times 0.4 \times 0.4 \times 0.8 = 0.086$$

$$\text{Maximum likelihood } (\text{M}_2) \rightarrow 0.7 \times 0.6 \times 0.6 \times 0.7 = 0.176$$

$$\text{M}_2 > \text{M}_1$$

Means we need to select Model 2 because it has greater Maximum likelihood but we can see that we are multiplying 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.

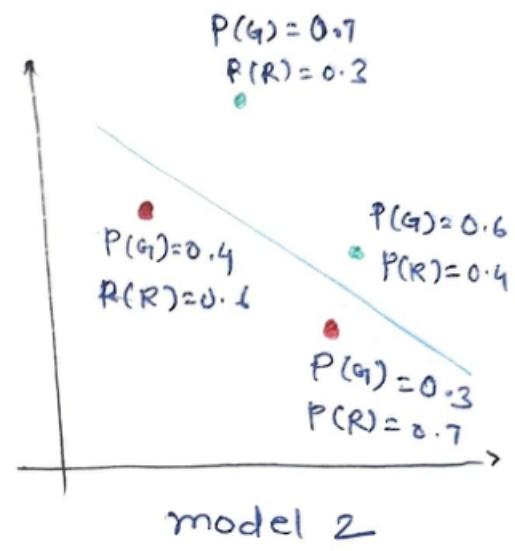
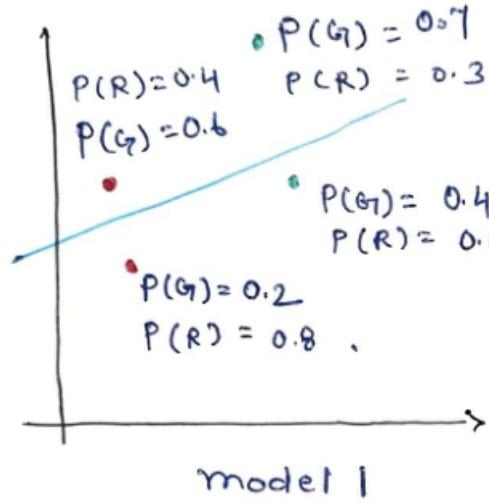
We know,

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

# Maximum Likelihood

## & Binary Cross Entropy

Let's take two model.



here we can see that  $\text{Model 2} > \text{Model 1}$

Let prove with Maths.

$$\text{Maximum likelihood } (M_1) \rightarrow 0.7 \times 0.4 \times 0.4 \times 0.8 = 0.086$$

$$\text{Maximum likelihood } (M_2) \rightarrow 0.7 \times 0.6 \times 0.6 \times 0.7 = 0.176$$

$$M_2 > M_1$$

Means we need to select Model 2 because it has greater Maximum likelihood but we can see that we are multiplying 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,

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

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 summation 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

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

$\downarrow$  This is wrong

because  $\hat{y}_i \rightarrow$  probability of both green & red

So,

$$L = \underbrace{-\frac{1}{n} \sum_{i=1}^n}_{\text{have both probil.}} \underbrace{y_i \log(\hat{y}_i)}_{\text{green}} + \underbrace{(1-y_i) \log(r\hat{y}_i)}_{\text{red}}$$

# Minimizing the Loss f<sup>n</sup> (Gradient Descent)

Let,

$$\begin{array}{ccccccc}
 & & x & & y & & \hat{y} \\
 & x_{11} & x_{12} & \cdots & x_{1n} & y_1 & \hat{y}_1 \\
 x_{21} & x_{22} & \cdots & x_{2n} & y_2 & & \{w_1, w_2, \dots, w_n\} \\
 \vdots & \vdots & & \vdots & \vdots & \vdots & w_0 \\
 \vdots & \vdots & & \vdots & \vdots & \vdots & \\
 x_{m1} & x_{m2} & \cdots & x_{mn} & y_m & &
 \end{array}$$

$\frac{\partial}{\partial w}$

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

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} \sigma(w_0 + w_1 x_{11} + w_2 x_{12} + \dots + w_n x_{1n}) \\ \vdots \\ \sigma(w_0 + w_1 x_{m1} + w_2 x_{m2} + \dots + w_n x_{mn}) \end{bmatrix}$$

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

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

We know, Loss f<sup>n</sup>

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

↪ Let's Write it into matrix form

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

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

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

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

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

$$\sum_{i=1}^m (1-y_i) \log(1-\hat{y}_i) = \overbrace{(1-Y) \log(1-\sigma(WX))}$$

Now Loss eq<sup>n</sup> becomes,

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

\* Now we have to make this f<sup>n</sup> min

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

for Minimizing the Loss f<sup>n</sup> we to <sup>practical</sup> clip it with its respective coefficient



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

$$\frac{\partial L}{\partial w} = \frac{d}{dw} y \log \hat{y} \Rightarrow y \frac{d}{dw} \log \hat{y} \Rightarrow \frac{y}{\hat{y}} \frac{d}{dw} (\hat{y})$$

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

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

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

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

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

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

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

$$\Rightarrow -\frac{(1-y)}{(1-\hat{y})} \hat{y}'(1-\hat{y}')x \frac{d}{dw} (wx)$$

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



so,

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

$$\left\{ \begin{array}{l} \frac{\Delta L}{\Delta w} \Rightarrow -\frac{1}{m} [y - \hat{y}]x \end{array} \right.$$

According to Gradient descent.

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

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}, x = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}_{(m, n+1)}$$

$$(y - \hat{y}) \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}_{(m, 1)} \frac{(y - \hat{y})x}{(m, 1)} \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

$$w_1 = (w_1) + \frac{n}{m} (y - \hat{y})x_{(1, n+1)}$$

$$w_{(n+1, 1)} = (w_{(n+1, 1)}) + \frac{1}{m} (y - \hat{y})x_{(1, n+1)}$$

$$x_{(1, n+1)}$$

## 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 doesn't tell nature of mistake

## Confusion Matrix.

1 → Sachme  
Heart disease - hai → 1

0 → Sachme  
Heart disease - nahi hai → 0

Actual

		Prediction.		
		1	0	→ 1 → heart disease Predicted
Actual	1	TRUE Positive (TP)	False Negative (FN)	0 → heart disease nahi predicted
	0	False Positive (FP)	TRUE Negative (TN)	

- \* Confusion matrix tells the nature of mistakes

- 1. Agar actual mai 1 and model bhi 1 → TP bole
- 2. Agar actual 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

Accuracy through CM → .

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

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

Type II ERROR → Ex:- Agar koi Aisa patient  
(FN) hai jise heart disease  
nhi hai aise model ne  
bola ~~actual~~ mai hai  
→ FN

- + When Accuracy is misleading?
- When there is imbalance data prediction or imbalance labels
- Ex. take Terrorist Example

In an airport there is only one terrorist and 999 people are G.P so.

CF	0	1
0	0	1
1	0	<u>999</u> 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%.

$$FP_A > FP_B \\ 30 > 10$$

	Sent to spam	No sent to spam
Spam	100	190 FN
Not spam	10 FP	700

Acc → 80%.

$$FN_A < FN_B$$

FP → Model ne not spam email ko spam bol deya

FN → Model ne spam wale email ko not spam boldey

Here  $FP_A > FP_B$ .

+ More Precise

So 2<sup>nd</sup> model is performing well

→ So precision 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

$$Acc = 90\%$$

$$A \quad F_{NA} < F_{NB}$$

	Detected cancer	No cancer
Has cancer	1000	500 FN
No cancer	500 FP	8000

$$Acc = 90\%$$

B

FP  $\rightarrow$  Model ne cancer detect kya par actual mai cancer nahi tha.

$\textcircled{FN}$   $\rightarrow$  Model ne cancer nahi bolo par actual mai cancer tha.

$$\Rightarrow F_{NA} < F_{NB}$$

so we select model I.

Recall  $\rightarrow$  Jitne logo ko sahi mai cancer tha unme se kitne ko sahi se detect kyu

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

- \* If TYPE I error is more dangerous then select high precision model
- \* If TYPE II error then Recall model.

# Multi-class Precision and Recall

	Dog	Cat	Rabbit	
Dog	25	5	10	$\rightarrow 40$
Cat	0	30	4	$\rightarrow 34$
Rabbit	4	10	20	$\rightarrow 34$
	29	45	34	

Precision =  $\frac{\text{TP}}{\text{TP} + \text{FP}}$   $\Rightarrow$   $P_D = \frac{25}{29}$ ,  $P_C = \frac{30}{45}$ ,  $P_R = \frac{20}{34}$

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

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

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

SAME with Recall

## F1 Score

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

- + Now we can't identify the model is precise or more Recall.
- + In these type of situation F1 score is given more importance.
- + F1 is combo of precision and Recall

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

→ humara jo F1 score Rahengा ~~to~~ %  
precision & Recall mai us taraf tilted  
Raheng jiska value kam hai.

# Soft MAX REGRESSION

or

# MULTINOMIAL REGRESSION

Let take example

CGPA	iq	place
7.0	71	0 → No
8.5	85	1 → Yes
9.5	95	2 → opt out

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

## Softmax Function

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{j=1}^k e^{z_j}} \quad k = \text{Number of classes}$$

(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	one hot encoding
		0 1 2	0 1 2 1 0 0 0 1 0 0 0 1

Now create 3 datasets

→ Three columns in output.

CGPA	iq	0(Place)

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

CGPA	iq	1(Place)

L.R. M2 ( $w_1^1, w_2^1, w_3^1$ )

CGPA	iq	2(Place)

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

$m_1$  (yes)

$(w_1^1, w_2^1, w_0^1)$

$m_2$  (No)

$(\bar{w}_1^2, \bar{w}_2^2, \bar{w}_0^2)$

$m_3$

(opt out)



Let take a point  
 $s_x = \{y, y_0\}$ .

$$z_3 = 7w_1^{(3)} + 70w_2^{(3)} + w_0^{(3)}$$

$$z_1 = 7w_1^{(1)} + 70w_2^{(1)} + w_0^{(1)}$$



$$z_2 = 7w_1^{(2)} + 70w_2^{(2)} + w_0^{(2)}$$

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

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

Let  $\rightarrow 0.40$

$\rightarrow 0.35$

0.25

Iska placement hoga.



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

$\rightarrow$  So we will modify the loss  $F^*$  so that it  
work on softmax reg.

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

$$L_{(\text{softmax})} = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log(\hat{y}_k^{(i)})$$

$x_1$	$x_2$	$y$	$y_{k=1}$	$y_{k=2}$	$y_{k=3}$
$x_{11}$	$x_{12}$	1	1	0	0
$x_{21}$	$x_{22}$	2	0	1	0
$x_{31}$	$x_{32}$	3	0	0	1

Now, in Loss fn (softmax)

$$y_1^{(1)} \log(\hat{y}_1^{(1)}) + \frac{y_2^{(1)} \log(y_2^{(1)}) + y_3^{(1)} \log(\hat{y}_3^{(1)})}{0}$$

✓ +

$$\frac{y_1^{(2)} \log(\hat{y}_1^{(2)})}{0} + \frac{y_2^{(2)} \log(y_2^{(2)})}{0} + \frac{y_3^{(2)} \log(\hat{y}_3^{(2)})}{0}$$

$$\frac{y_1^{(3)} \log(\hat{y}_1^{(3)})}{0} + \frac{y_2^{(3)} \log(y_2^{(3)})}{0} + \frac{y_3^{(3)} \log(\hat{y}_3^{(3)})}{0}$$

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

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

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

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

