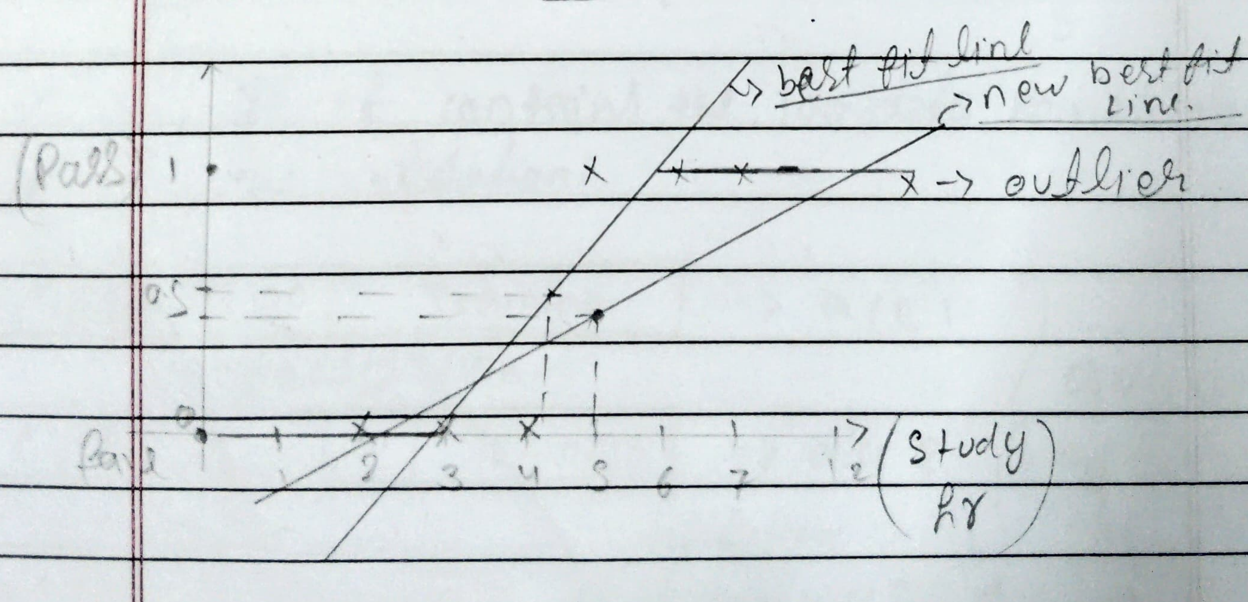
# Logistic regression
## (Binary classification)

**dataset**

| Study hr | Pass / Fail (Binary category) |
|---|---|
| 2 | Fail |
| 3 | fail |
| 4 | Fail |
| 5 | Pass |
| 6 | Pass |
| 7 | Pass |

I/P $\xrightarrow{\text{Study hr}}$ $\boxed{\text{machine model}}$ $\longrightarrow$ O/P Pass/Fail



In the given plot of fail / Pass v/s Study hr is solved by the Linear Regression so in this if my value or input is

$$\Rightarrow \begin{cases} \text{if } x < 0.5 \rightarrow \text{fail } (0) \\ \quad\; x > 0.5 \rightarrow \text{Pass } (1) \end{cases}$$

↳ This is condition for binary classification.

Let's assume an outlier 12 hr and when we are drawing the new best fit line with respect to outlies the line sifts and then if we input the 5 hr then the value is coming $< 0.5$ means it is going to give me the false or fail output but in our training data if the input is 5 hr then the output is Pass.

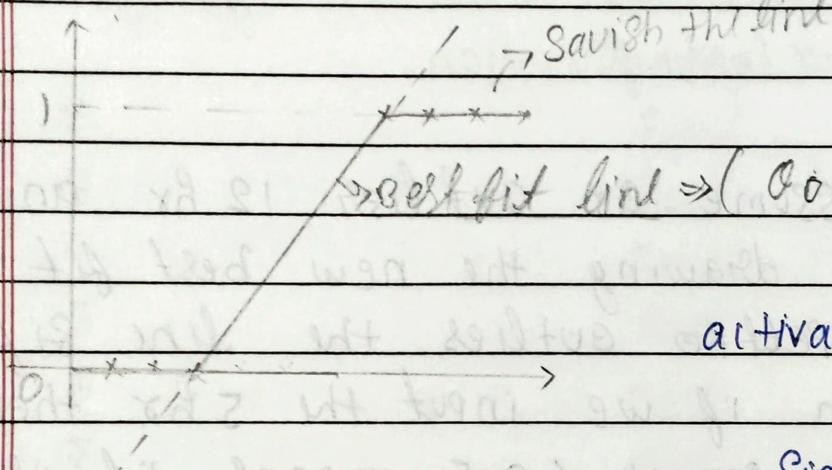
That is the problem with ~~ost~~ Linear Regression using in classification problem.

⇒ and two major Problem with the Linear Regression

1. In prasence of outlier the Best fit line chang
2. The value is come $< 0$ and $> 1$ (not possible).

⇒ So to encounter these thing or problems we use the Logistic Regression not Linear Regression.

⇒ How Logistic Regression solves classification problem.




activation function

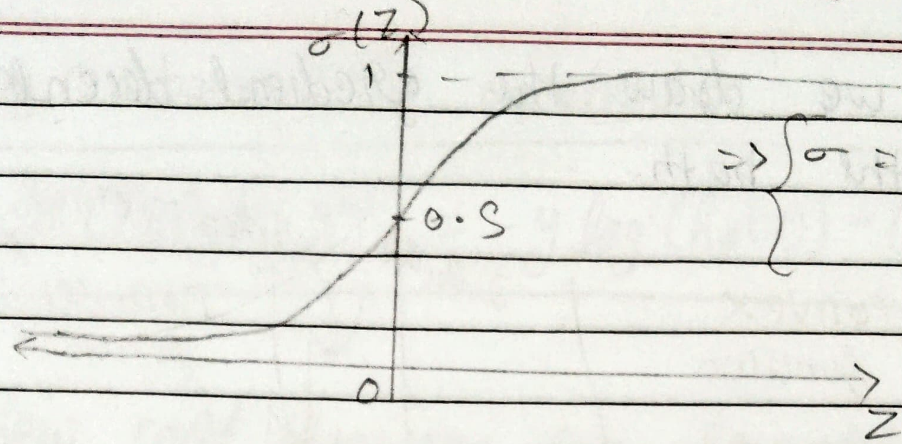
⇓

Sigmoid

$$h_\theta(x) = \sigma(\underbrace{\theta_0 + \theta_1 x_1}_{z})$$

after the sigmoid ⇒ $h_\theta(x) = \sigma(z) \Rightarrow \dfrac{1}{1+e^{-z}}$

# $$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1)}}}$$




$$\Rightarrow \left\{ \sigma = \frac{1}{1+e^{-z}} \right\}$$

#
$$\left\{ \begin{array}{l} \text{if } \left\{ \begin{array}{l} z > 0 \Rightarrow O/P \Rightarrow 1 \\ z < 0 \Rightarrow O/P \Rightarrow 0 \end{array} \right\} \\ \qquad \text{or} \\ \text{if } \left\{ \begin{array}{l} \sigma(z) > 0.5 \Rightarrow O/P \Rightarrow 1 \\ \sigma(z) < 0.5 \Rightarrow O/P \Rightarrow 0 \end{array} \right\} \end{array} \right\}$$

$\Rightarrow$ <u>Linear Regression cost function:</u>

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x)^{(i)} - y^{(i)} \right)^2$$

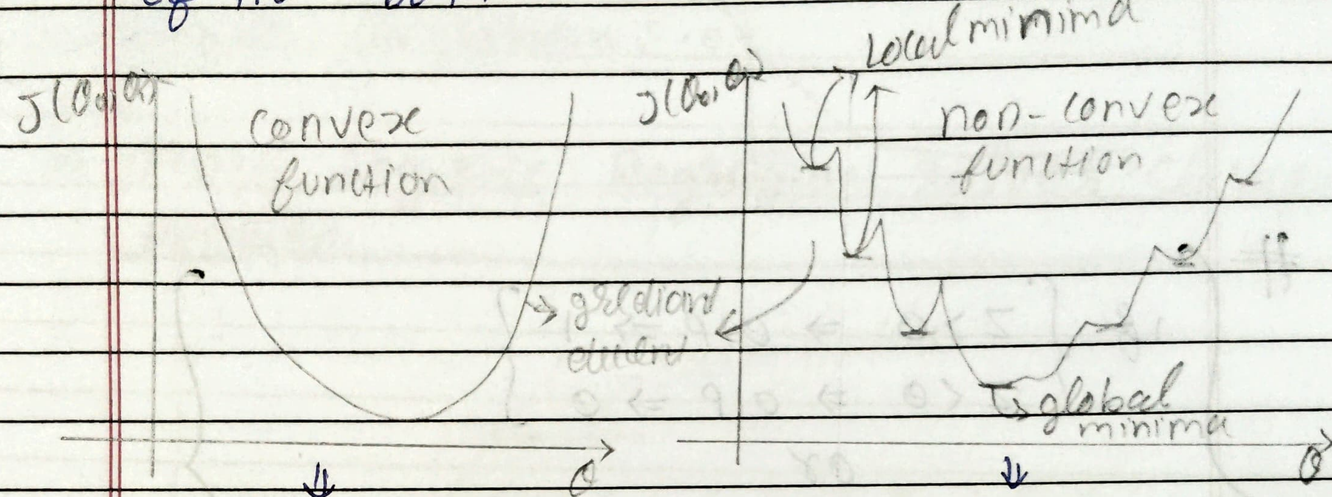
$$\Rightarrow h_\theta(x) = \theta_0 + \theta_1 x_1$$

$\Rightarrow$ <u>Logistic Regression cost function:</u>

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x)^{(i)} - y^{(i)} \right)^2$$

$$\Rightarrow h_\theta(x) = \frac{1}{1+e^{-z}}, \quad z = \theta_0 + \theta_1 (x_1)$$

if we draw the gredient decent curve of the both.




Linear Regression of $J(\theta_0, \theta_1)$

Logistic Regression of $J(\theta_0, \theta_1)$

⇒ Let's modify the $J(\theta_0, \theta_1)$ for Logistic Regression:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left( h_\theta(x)^{(i)} - y^{(i)} \right)^2$$

Let $cost\left(h_\theta(x)^{(i)}, y^{(i)}\right)$

$$cost\left(h_\theta(x)^{(i)}, y^{(i)}\right) = \begin{cases} -\log\left(h_\theta(x)\right) & \text{if } y = 1 \\ -\log\left(1 - h_\theta(x)\right) & \text{if } y = 0 \end{cases}$$

⇓

log loss.

by combaining the equations.

\# $$\boxed{cost(h_\theta(x)^{(i)}, y^{(i)}) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))}$$

New cost function for logaritic Regression.

$$\boxed{J(\theta_0, \theta_1) = \frac{-1}{2m} \sum_{i=1}^{m} \left[ \left( y^{(i)} \log (h_\theta(x)^{(i)}) \right) - (1-y^{(i)}) \log (1-h_\theta(x)^{(i)}) \right]}$$

so minimize the cost function $J(\theta_0, \theta_1)$ by changing $\theta_0$ & $\theta_1$

⇒ Convergence algorithm.

Repeat

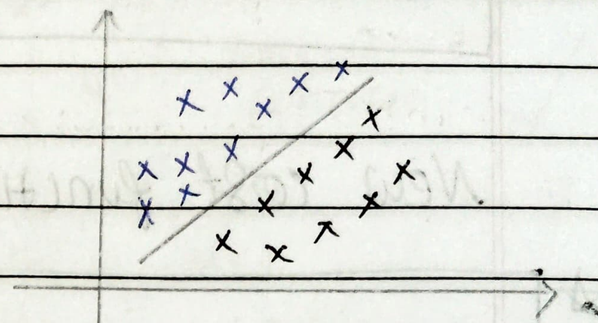
learning rate

cost function

$$\left\{ \theta_j \approx \theta_j - \alpha \frac{\partial (J(\theta_0, \theta_1))}{\partial \theta_j} \right\}$$

=> Performence matrix, Accuracy, Precision, Recall And F-Beta score.

1. Confusion matrix
2. Accuracy
3. Precision
4. Recall
5. F-Beta score



Logistic Regression

**Dataset**

Input — actual O/P — Predicted O/P

| $x_1$ | $x_2$ | $y$ | $\hat{y}$ |
|---|---|---|---|
| — | — | 0 | 1 |
| — | — | 1 | 1 |
| — | — | 0 | 0 |
| — | — | 1 | 1 |
| — | — | 1 | 1 |
| — | — | 0 | 1 |
| — | — | 1 | 0 |

① **Confusion matrix :** → actual values

True Positive (TP)

| Predicted values | 1 | 0 |
|---|---|---|
| 1 | 3 | 2 |
| 0 | 1 | 1 |

→ False Positive (FP)

→ True Negative (TN)

False Negative (FN)

| | 1 | 0 |
|---|---|---|
| 1 | TP | FP |
| 0 | FN | TN |

$$\Rightarrow \text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Dataset
↓
1000 data Points → 900 → 1, 100 → 0 ⇒ Imbalence data

Imp

So In case of Imbalence dataset we can't use the directly accuracy formula

So we have to use here first Precision and Recall matrices.

(iii) **Precision** $\Rightarrow \dfrac{TP}{TP+FP}$

| | 1 | 0 |
|---|---|---|
| 1 | TP | FP |
| 0 | FN | TN |

$\Downarrow$

It meant that out of all the actua values how many are correctly predicted.

(iv) **Recall** $\Rightarrow \dfrac{TP}{TP+FN}$

| | 1 | 0 |
|---|---|---|
| 1 | TP | FP |
| 0 | FN | TN |

$\Downarrow$

It means that out of all the predicted value how many are correctly predicted.

(v) F- Beta score $\Rightarrow (1+\beta^2) \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

$\Downarrow$

{ If FP & FN are both important then $\beta \Rightarrow 1$ }

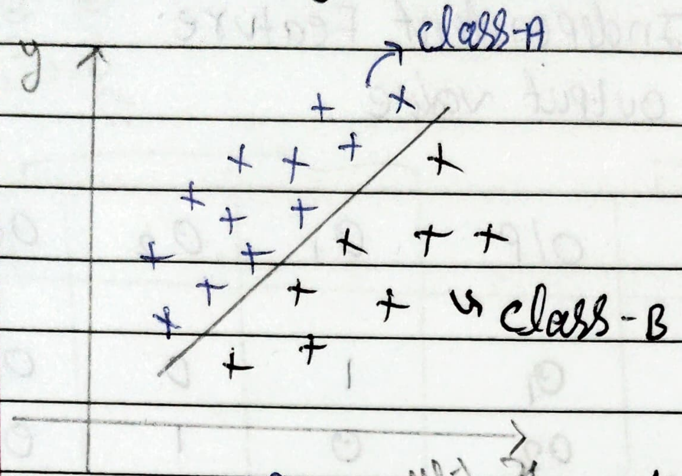
$\Downarrow$

$F_1\text{-score} = (1+1^2) \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

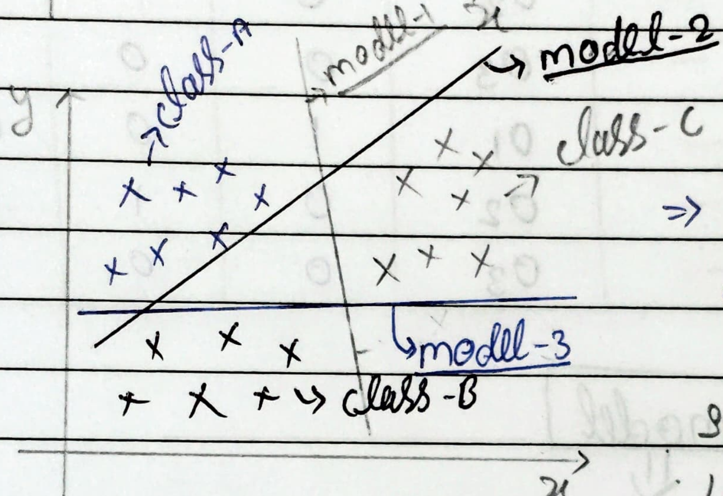
$\Downarrow$

$$\boxed{F1\text{-score} = 2 \times \left( \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right)}$$

# Logistic Regression (One V/S Rest) → OVR




⇒ Binary classification




⇒ multiclass classification

⇓

solving with Logistic Regression using (OVR)

So to solve these type of multiclass classification using logistic Regression ⇒

we have to make the 3-line to saparate it and the classify it in two category. like ⇒

$$\left\{ \begin{array}{l} \text{model-1} \Rightarrow \text{Binary classification} \\ \text{model-2} \Rightarrow \text{Binary classification} \\ \text{model-3} \Rightarrow \text{Binary classification} \end{array} \right\} \Rightarrow \text{Join or combain it togethor.}$$

Dataset =>

3 → Independent Feature

I/P 3 → output value

One Hot Encoding

| $f_1$ | $f_2$ | $f_3$ | O/P | $O_1$ | $O_2$ | $O_3$ |
|---|---|---|---|---|---|---|
| — | — | — | $O_1$ | 1 | 0 | 0 |
| — | — | — | $O_2$ | 0 | 1 | 0 |
| — | — | — | $O_3$ | 0 | 0 | 1 |
| — | — | — | $O_1$ | 1 | 0 | 0 |
| — | — | — | $O_2$ | 0 | 1 | 0 |
| — | — | — | $O_3$ | 0 | 0 | 1 |

$\boxed{\text{model}}$

⇓

model-1 => I/P => $\{f_1, f_2, f_3\}$ , O/P => $\{O_1\}$
model-2 => I/P => $\{f_1, f_2, f_3\}$ , O/P => $\{O_2\}$
model-3 => I/P => $\{f_1, f_2, f_3\}$ , O/P => $\{O_3\}$

New Data =>
→ modell-1 => 0.25
→ model-2 => 0.20
→ modell-3 => 0.55

Probabilities => [0.25, 0.20, 0.55]
So the model-3 gives the higher Probability w.r.t others, so the final O/P class is $O_3$