Logistic Regression

To solve Classification → Binary classification → O/P → 2 categories
→ Multiclass classification → O/P → >2 categories

Dataset

Independent ↓ | O/P or Dependent feature ↓

No. of play hours		Pass/Fail (y_i)
9	0	Fail
8	0	Fail
7	0	Fail
6	0	Fail
5	1	Pass
4	1	Pass
3	1	Pass
2	1	Pass

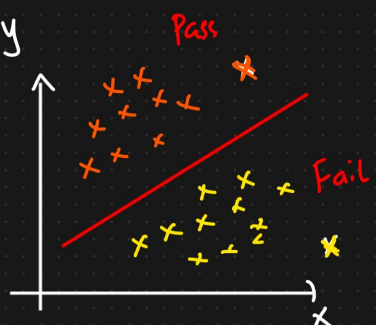
$\hat{y}_i$

TRAIN ↓

New data → Model → Pass/Fail

No. of play hours

↓ Accuracy ↑↑

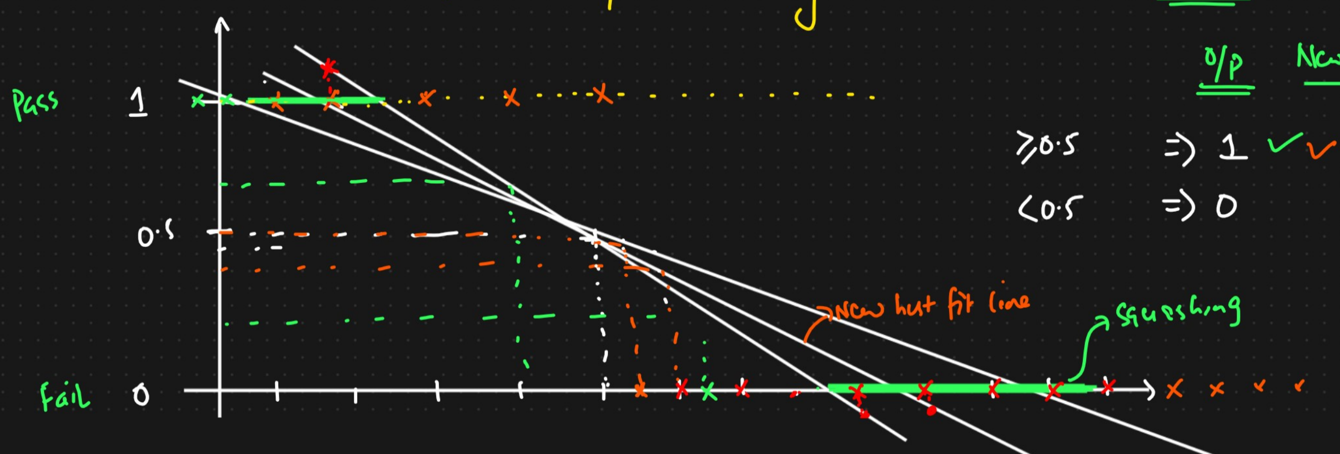


Can we solve this classification problem using Regression?

1 or 0

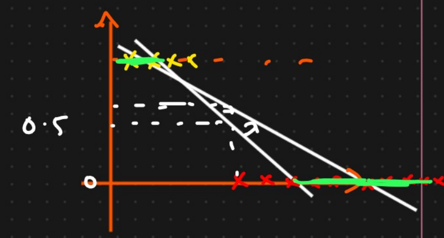
Threshold = 0.5

	O/P	New DATA
≥ 0.5	⇒ 1 ✓✓	
< 0.5	⇒ 0	



0 1 2 3 4 5 6 7 8 9 10 11 12 13 No of play hours

Why we cannot used Linear Regression for classification



① Best fit line changes because of outliers ⟶ prediction goes wrong

② The outcomes comes >1 and <0

⇓

Logistic Regression

⇓

$\boxed{0 \text{ to } 1}$ ⟹ Squashing Technique

How Logistic Regression Solves Classification Problem?

No of play hours

$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$ ⟹ Best fit line

⇓

Sigmoid Activation function

⇓

0 to 1

$$\boxed{\sigma = \frac{1}{1+e^{-z}}}$$ ⟹ 0 to 1

Squashing the best fit line

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

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

⇓

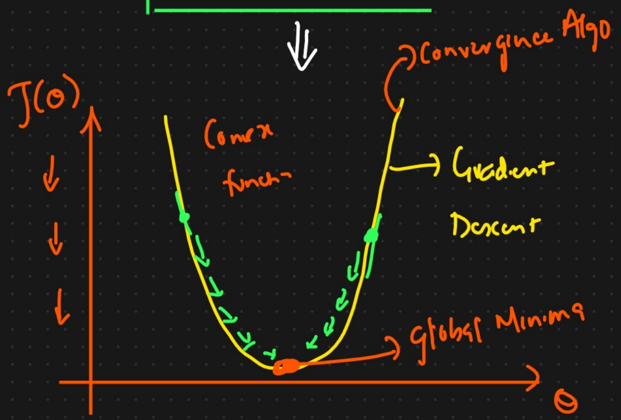
Prediction.

Linear Regression Cost function

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

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1}$$

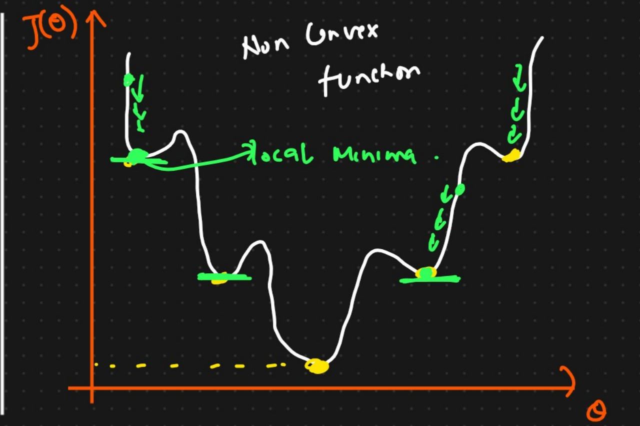
$$\Downarrow$$



Logistic Regression Cost function

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

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1)}}} \Rightarrow 0 \text{ to } 1$$



Log Loss function

$h_\theta(x)_i$ = predicted value = $\hat{y}$

$y_i \Rightarrow$ Actual Data.

$$\boxed{J(\theta_0,\theta_1) = -y_i \log\left(h_\theta(x)_i\right) - (1-y_i)\log\left(1-h_\theta(x)\right)} \Rightarrow \text{Cost function.}$$

if $y_i = 1$

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1)}}} \Rightarrow \hat{y} \Rightarrow \text{predicted point}$$

$$J(\theta_0,\theta_1) = \begin{cases} -\log\left(h_\theta(x)_i\right) & \text{if } y=1 \\ -\log\left(1-h_\theta(x)\right) & \text{if } y=0 \end{cases}$$

Final Aim: Minimize Cost function $J(\theta_0,\theta_1)$ by changing θ_0 & θ_1

Convergence Algorithm

Repeat until convergence

{

$$\theta_j : \theta_j - \alpha \frac{\partial}{\partial \theta} J(\theta_0, \theta_1)$$

}