Machine Learning Algorithms

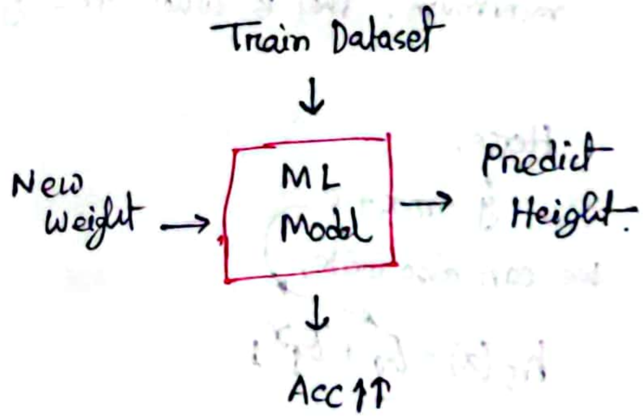
Simple Linear Regression (Algorithm)

Supervised ML
- → Regression O/P or Dependent Feature → Continuous
- → Classification Op or Dependent Feature → Categorical

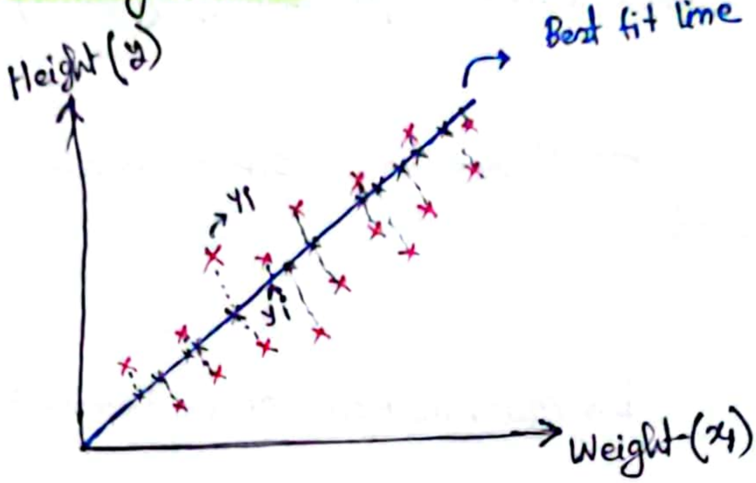
Problem statement: We have a dataset where there are 2 features. Weight (Independent) and Height (Dependent). After training our ML model which will be created using Simple linear regression algorithm, our model should be able to predict Height when a new input of weight is given.

Dataset

Weight	Height (Dependent)
74	170
80	180
75	175.5
—	—
—	—
—	—



Aim of simple linear regression:



In our simple linear regression model, we try to predict a best fit line in the data plot where we try to project data values ($\hat{y}_i$) but the actual values are (y_i). So, for each data value (y_i) there is an error ($y_i - \hat{y}_i$). Our aim is to fit the straight line in such a way that we can get the minimum error. $\Sigma(y_i - \hat{y}_i)$ would have to be minimum. That is what the goal of simple linear regression model is.

Here,

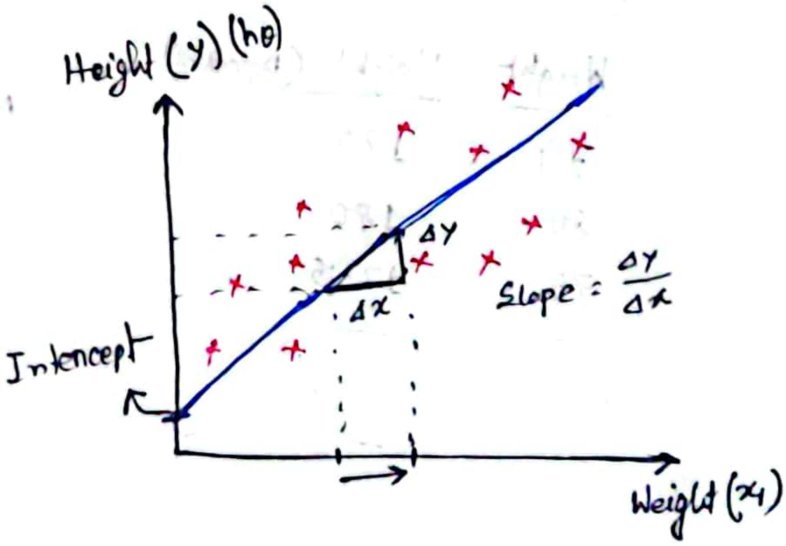
$\hat{y} = mx + c$

We can also write,

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

$\theta_0 \Rightarrow$ Intercept

$\theta_1 \Rightarrow$ Slope



As our data value of weight (x_1) are fixed, so we have to change (θ_0 and θ_1) in such a way that we can get the best fit line with minimum error.

So, the algorithm is kind of, we first intialize our θ_0 and θ_1 and draw the straight line for (x_1) datapoints. But after that we have to continously change our θ_0 and θ_1 values till we get the minimum error value.

The error is called cost Function.

Cost Function: [Error]

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

[n = num of datapoints
y_i = actual values
$\hat{y}_i$ = Predicted values]

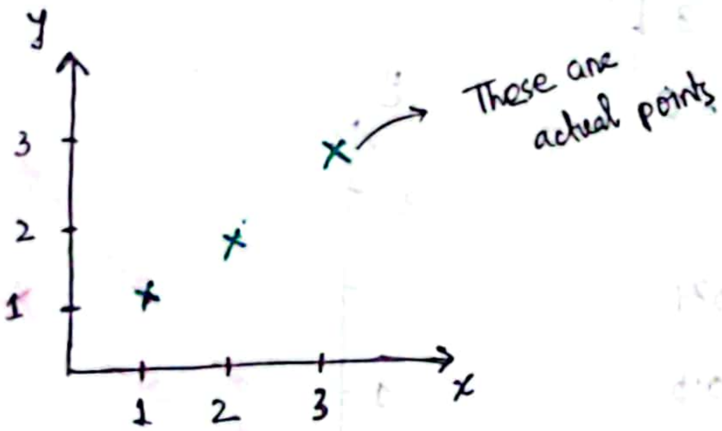
This cost function called Mean squared error.

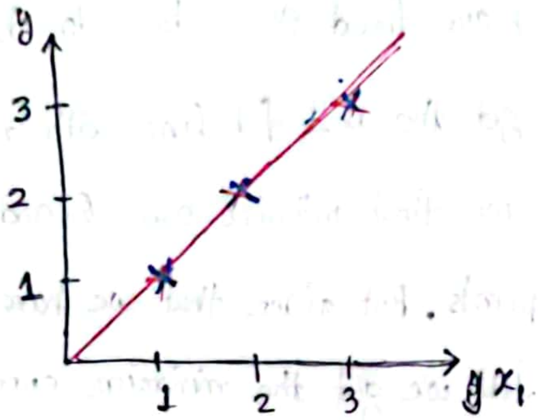
Our final aim is to minimize this cost function.

The process of Minimization: Suppose our dataset →

Dataset

x	y	$h_\theta(x)$
1	1	
2	2	
3	3	





Our straight line equation was

$$h\theta(x) = \theta_0 + \theta_1 x_1$$

Let's consider, $\theta_0 = 0$

$\therefore h\theta(x) = \theta_1 x_1$

(Means line will pass from origin).

Let, initialize $(\theta_0, \theta_1) = (0, 1)$

So, $h\theta(x) = \theta_0 + \theta_1 x_1$

$= 1x_1$

$\therefore$ for, $x = 1$, $h\theta(x) = 1$
$x = 2$, $h\theta(x) = 2$
$x = 3$, $h\theta(x) = 3$
$\rightarrow$ The predicted data from the actual data

$\therefore$ cost function $J(\theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h\theta(x)_i)^2$

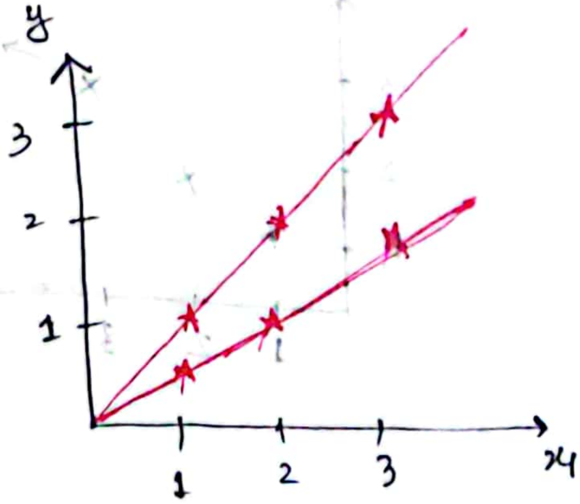
$$= \frac{1}{3} \left[(1-1)^2 + (2-2)^2 + (3-3)^2 \right]$$

$J(1) = 0$

Let, $(\theta_0, \theta_1) = (0, 0.5)$

$\therefore h\theta(x) = 0.5x_1$

for, $x = 1$, $h\theta(x) = 0.5$
$x = 2$, $h\theta(x) = 1$
$x = 3$, $h\theta(x) = 1.5$



Similarly, cost function for $\cancel{J(\theta_1)}$ $J(0.5) = 1.16$

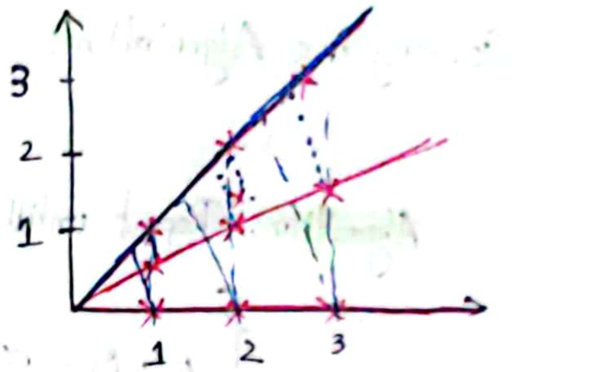
Now, let $(\theta_0, \theta_1) = (0,0)$

$\therefore h_\theta(x) = 0$

for, $x = 1$, $h_\theta(x) = 0$

for, $x = 2$, $h_\theta(x) = 0$

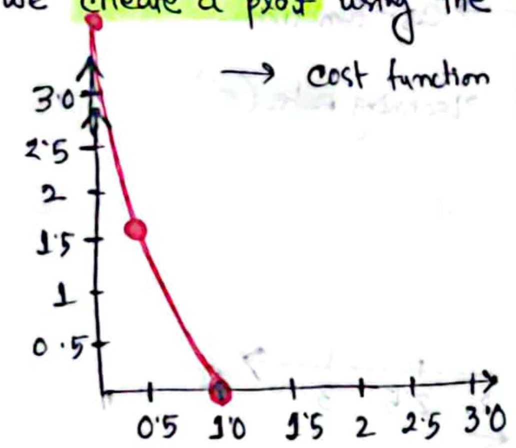
for, $x = 3$, $h_\theta(x) = 0$



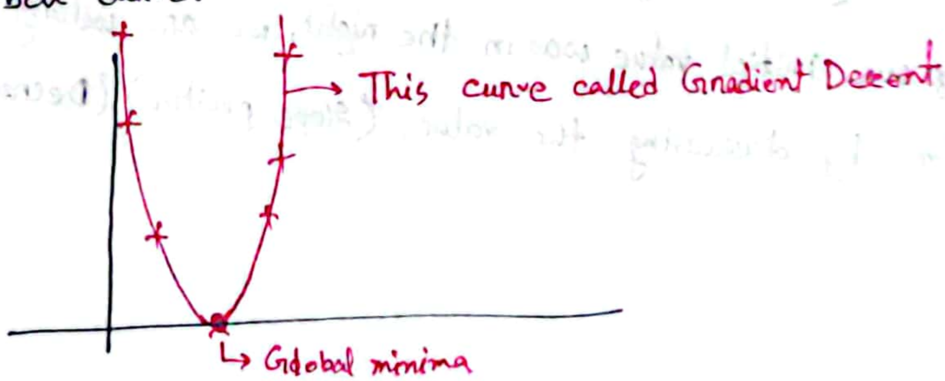
cost function for $J(0) = 4.66$

Now, if we create a plot using the cost function values →

→ cost function plot



If, we take some more points in θ_1, the cost function plot will become a bell curve. Our goal is to find the value of the global minima of the bell curve.



After intializing a value for (θ_0, θ_1), then we have to use the "convergence alogrithm" to reach to our global minima.

Convergence Algorithm: { Optimize the change of (θ_0, θ_1) to global Minima }

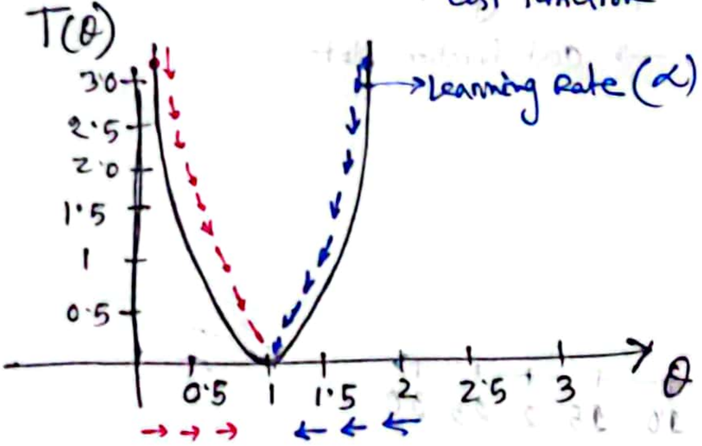
Algorithm: Repeat untill convergence

$$\left\{ \theta_j : \theta_j - \alpha \frac{d}{d\theta_j} J(\theta_j) \right\}$$

$\frac{d}{d\theta_j} J(\theta_j)$ → Slope

α → Learning Rate

θ_j → Cost function



Case1:
if our intial value was in the left, we are walking to the global minima by increasing the value (Slope ~~positiv~~ negative) (increasing θ values)

Case2:
If our initial value was in the right, we are walking to the gloable minima by decreasing the value. (Slope positive) (Decreasing θ values)

Learning Rate: It is the speed using which we are reaching towards global minima. It should not be very big or very small. Because if it is very big, then we can left behind the global minima and can go outside the range of the curve. If it is very slow, then it will take a lot time to reach the global minima.

So we have to make a balance in the learning rate.

Usually it is ~~(0.05)~~ (0.01)

$$\text{Learning Rate } (\alpha) = \text{Speed of Convergence}$$

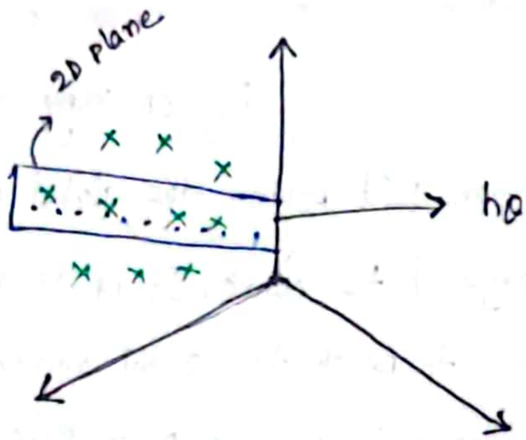
Multiple Linear Regression:

Previously we had one input feature (weight). What if we have more than one input feature.

House pricing dataset:

independent features		Dependent Features
No. of rooms (x_1)	Size of room (x_2)	Price (Y)
–	–	–
–	–	–
–	–	–
–	–	–

For, multiple (2) linear regression:



$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

θ_0 = intercept

θ_1, θ_2 = Slopes

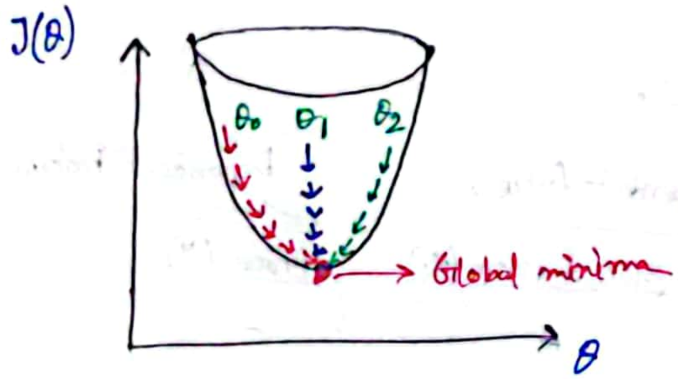
For multiple linear regression General equation:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots\cdots + \theta_n x_n$$

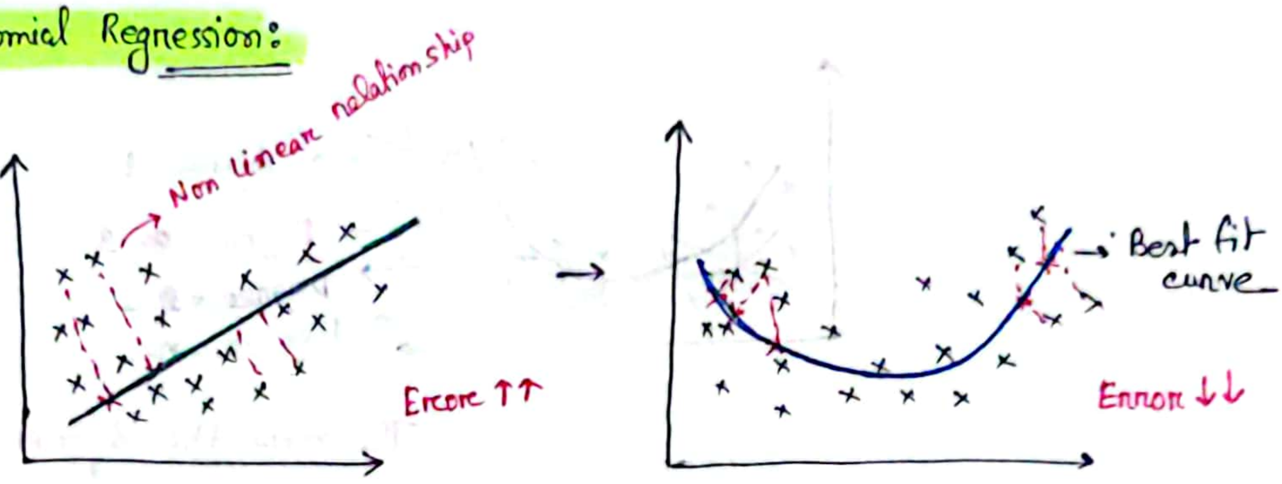
θ_0 = intercept

$\theta_1, \theta_2, \ldots \theta_n$ = Slopes.

The cost function plot for 2 independent features will be in 3D.



Polynomial Regression:



If the data in the dataset are non linear in relationship, we can't just make a best fit straight line Because, There will be great number of errors because of the distance of $(y_i - \hat{y})$. So, for this non-linear data we have to get a curve line to best fit the data and reduce the error. That's why we need polynomial regression here.

Polynomial Regression → Simple polynomial Regression

Polynomial Regression → Multiple polynomial Regression

↓

Polynomial Degree

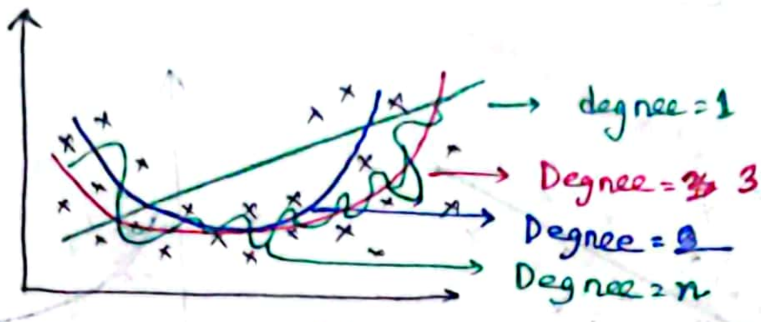
When, polynomial degree = 0, $h_\theta(x) = \theta_0 \times 1 = \theta_0 x_1^0$ = constant = intercept

Polynomial degree = 1, $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1$ = Simple Linear regression

Polynomial degree = 2, $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2$

⋮

Polynomial degree = n, $h_\theta(x) = \theta_0 x_1^0 + \theta_1 x_1^1 + \theta_2 x_1^2 + \theta_3 x_1^3 + \cdots + \theta_n x_1^n$



The more the degree,

the accurate it will fit the data

For multiple polynomial regression,

Equation will be like this,

Suppose the degree = 2

$$\boxed{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_3^2}$$