Linear Regression

Simple Linear Regression

Linear Regression

Experience in Years	0	2	4	5	6
Salary	2,00,000	4,00,000	8,00,000	10,00,000	12,00,000

What would be the **salary** of a person with **3 years of Experience?**

~ ₹ 650000 per Year

we can find the values using Y = mX + x

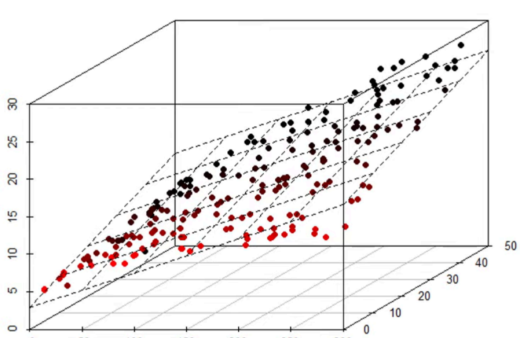
Thus , the model tries to make a straight line between the points and makes predictions based on it.

Multiple Linear Regression

What if there are more than 2 Variables?

Multiple Linear Regression

Multiple linear regression is a model for predicting the value of one dependent variable based on two or more independent variables.



Simple Linear Regression	$y = b_0 + b_1 * x_1$
Multiple Linear Regression	$y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$

Linear regression advantages and disadvantages:

Advantages:

1. Very simple to implement
2. Performs well on data with linear relationship

Disadvantages:

1. Not suitable for data having non-linear relationship
2. Underfitting issue
3. Sensitive to Outliers

Math behind linear regression

$Y = mX + c$

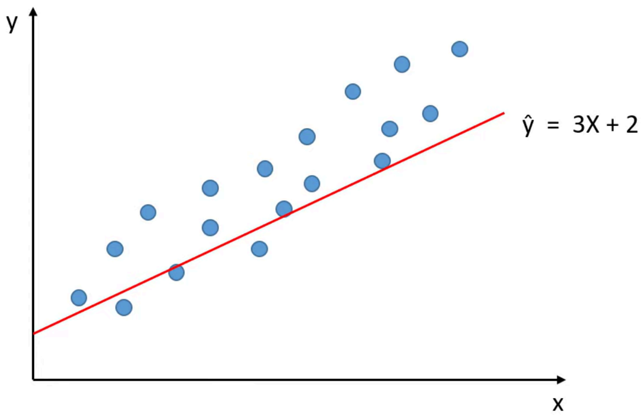
when $x=0$, then $Y= c$. Thus, c is value of y when x is zero.
and slope = tan theta = $dy/dx = y2 - y1/ x2 - x1$

How linear regression finds best fit line

This model uses Loss function to calculate the best fit.

Loss function :

Randomly assigned Parameters: m = 3; c = 2



x	y	ŷ
2	10	8
3	14	11
4	18	14
5	22	17
6	26	20

x	y	ŷ
2	10	8
3	14	11
4	18	14
5	22	17
6	26	20

$$\textbf{\textit{Loss}} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Loss = [$(10 - 8)^2 + (14 - 11)^2 + (18 - 14)^2 + (22 - 17)^2 + (26 - 20)^2$] / 5

Loss = [4 + 9 + 16 + 25 + 36] / 5

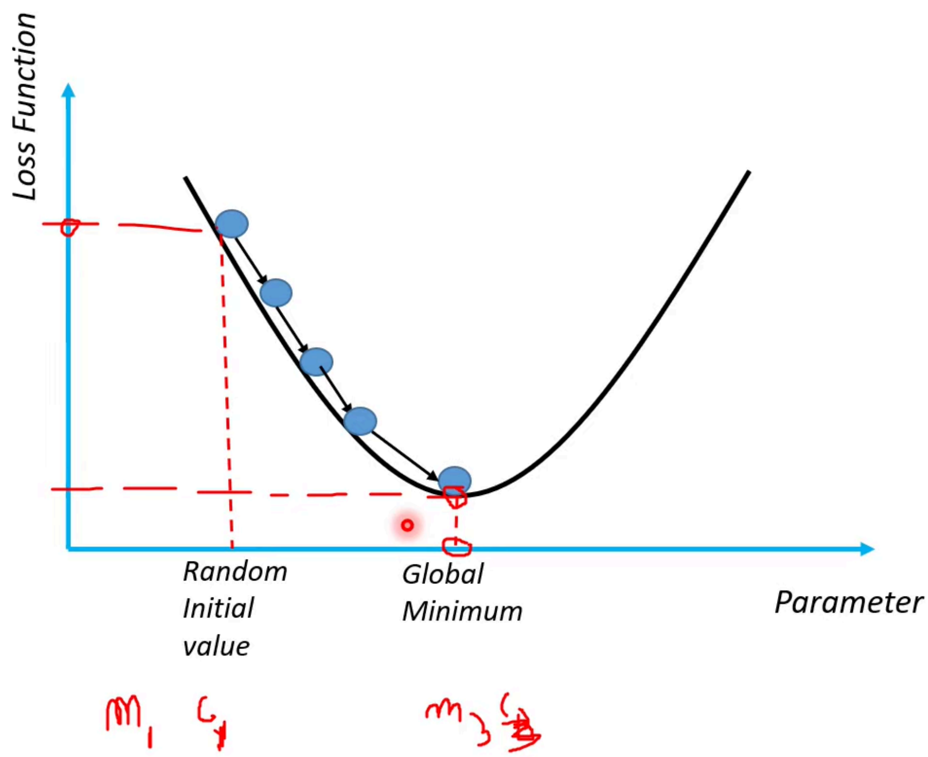
Loss = 18

Low loss value --> high accuracy

The loss of 18 is very high. Thus, the model will go through multiple iterations to find the best fit by using loss function.

Gradient descent for linear regression

Gradient Descent



Gradient descent is used in multiple algorithms , not only linear regression.

Gradient Descent is an optimization algorithm used for minimizing the loss function in various machine learning algorithms. It is used for updating the parameters of the learning model.

$$m = m - LD_m$$

$$c = c - LDc$$

m --> slope
c --> intercept
L --> Learning Rate

D_m --> Partial Derivative of loss function with respect to m

D_c --> Partial Derivative of loss function with respect to c

$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}\left(y_i - y_{i\ pred}\right)^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}\left(y_i - (mx_i + c)\right)^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}\left(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic\right)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i\left(y_i - (mx_i + c)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i\left(y_i - y_{i\ pred}\right)$$

$$D_c = \frac{\partial(Cost\ Function)}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=0}^{n}\left(y_i - y_{i\ pred}\right)^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}\left(y_i - (mx_i + c)\right)^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}\left(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic\right)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}\left(y_i - (mx_i + c)\right)$$

$$\frac{-2}{n}\sum_{i=0}^{n}\left(y_i - y_{i\ pred}\right)$$