Simple Linear Regression

Used for Supervised ML

Regression problems

Dataset

Weight (Independent Feature)	Height (Dependent/output Feature)
74	170
80	180
75	175.5

Now we have to train the model that takes the weight and predict height

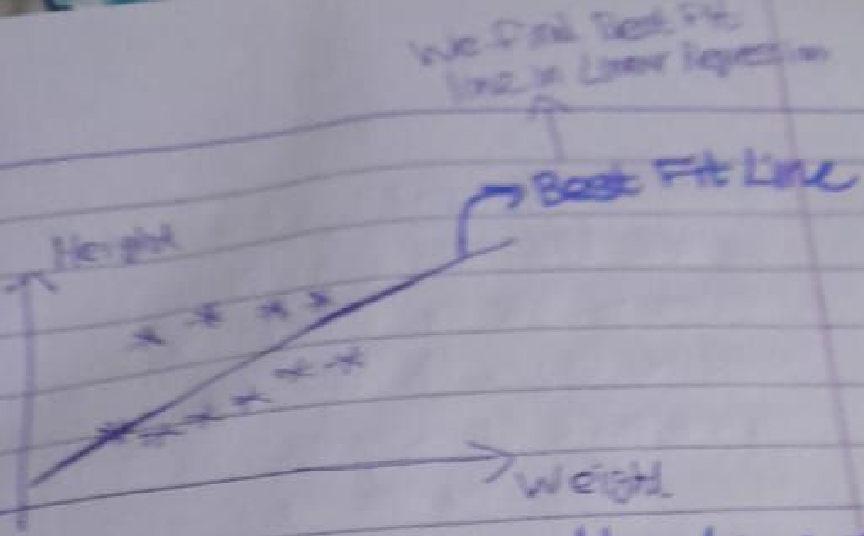
Weight → [Model] → Height

↑

Simple Linear Regression

⇒ When we have one independent feature we call it Simple Linear Regression.

⇒ When we have more than one independent features we call it multiple linear regression.



⇒ Understanding Simple Linear Regression Equations:

Straight Line Equations:

$y = mx + c$ or

$y = \beta_0 + \beta_1 x$ or

$\boxed{h(x) = \theta_0 + \theta_1 x}$ → We will use this one

Independent Feature

Now our equation is

$$\boxed{h(x) = \theta_0 + \theta_1 x}$$

where **x =** Independent Feature

θ_0 = Intercept

∴ Intercept means when x is 0 what is the value of y

θ_1 = Slope or Coefficient

∴ Slope is with the unit movement in x-axis what is the unit movement in y-axis

$x \Rightarrow$ My data points with respect to x-axis

$h^{\circ}(x) \Rightarrow \hat{y} \Rightarrow$ Predicted Point

Error : $(y - \hat{y})$

Where

y: Actual output in dataset

$\hat{y}$: Predicted output point

We have to create **Best Fit Line** in which the **error** is minimal.

⇛ Cost Function:

$$\underbrace{J(\overset{\text{Intercept}}{\theta_0}, \overset{\text{Slope}}{\theta_1}) = \frac{1}{2m} \sum_{i=1}^{m} (\overset{\text{Predicted output}}{h_\theta(x)^{(i)}} - \overset{\text{True output}}{y^{(i)}})^2}_{\text{Mean Squared Error}}$$

Final aim:

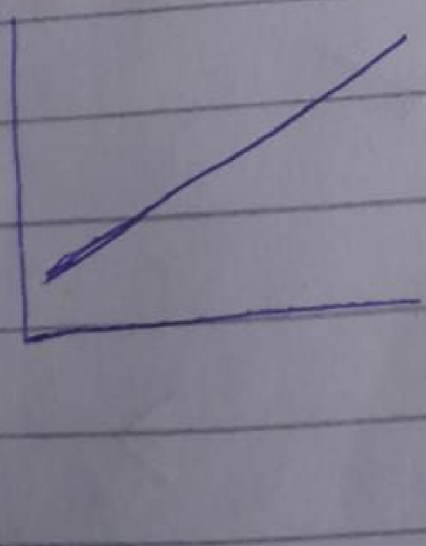
Our aim is to minimize the cost function given by

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=0}^{m} (h_\theta(x)^{(i)} - y^{(i)})^2$$

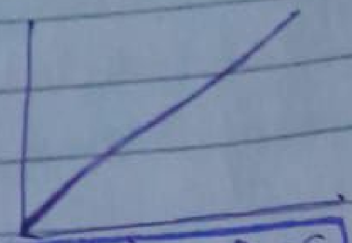
To minimize it we have to change θ_0 (intercept) and θ_1 (slope)

Equation of straight line

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



Equation of straight line passing through origin:



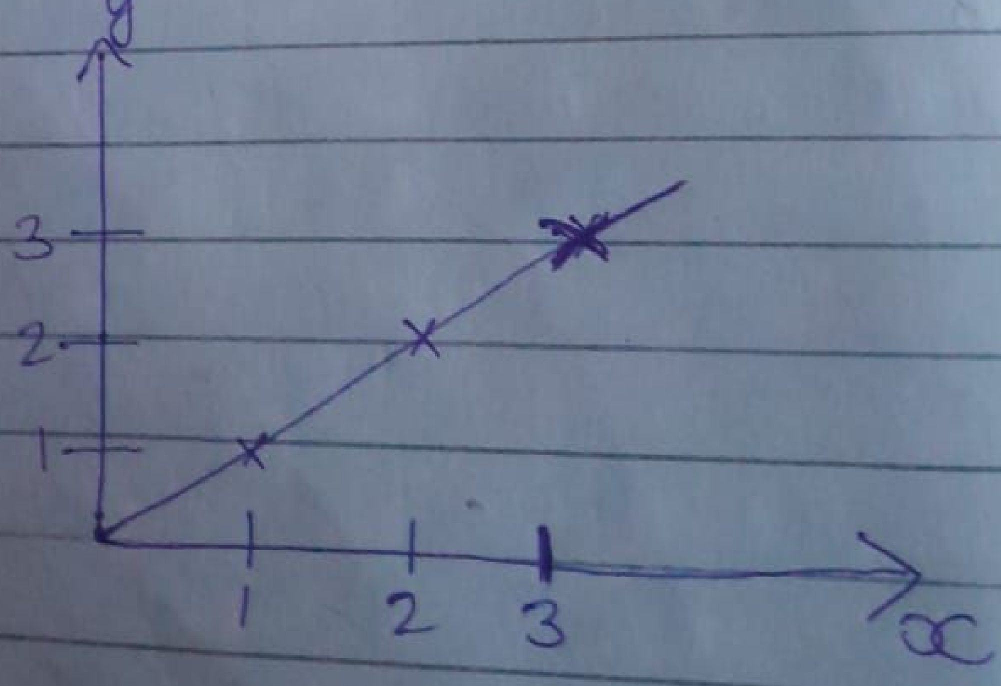
$\boxed{h_\theta(x) = \theta_1 x}$

As $\theta_0 \Rightarrow$ Intercept $= 0$

Let we have a dataset

x	y
1	1
2	2
3	3

And we have to create best fit line for it



Now as we said it is passing through origin so our equation will be

$h_\theta(x) = \theta_1 x$

Let $\theta_1 = 1$ {slope}

At $x = 1$

$h_\theta(x) = (1)(1) = 1$ when $x = 1$

$h_\theta(x) = 2$ when $x = 2$

$h_\theta(3) = 3$ when $x = 3$

I have drawn the line according to these $h_\theta(x)$ points

Now Let's find cost function of this line

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

∴ m = total points given = 3

$$J(\theta_1) = \frac{1}{2\times3} \sum_{i=1}^{3} (h_\theta(x)^{(i)} - y^{(i)})^2$$

$$J(\theta_1) = \frac{1}{6} \left[(1-1)^2 + (2-2)^2 + (3-3)^2 \right]$$

$$J(\theta_1) = \frac{1}{6}(0)$$

$\boxed{J(\theta_1) = 0}$ $\boxed{\text{when } \theta_1 = 1}$

My cost function is 0 means there is no error so best fit line

Now let's change our θ_1 (slope) value

Let $\theta_1 = 0.5$

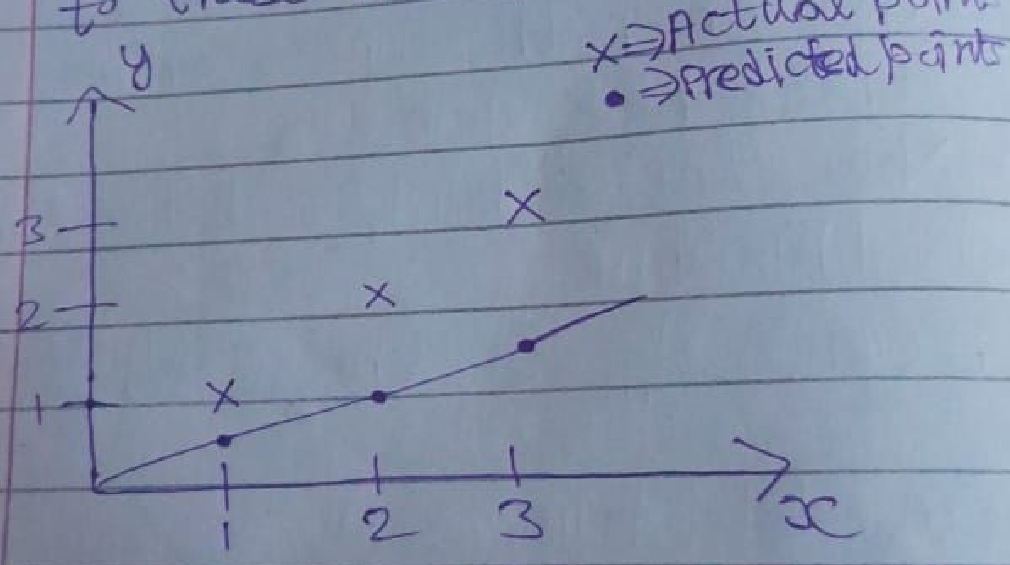
As $h_\theta(x) = \theta_1 x$

$h_\theta(x) = 0.5$ if $x = 1$

$h_\theta(x) = 1$ if $x = 2$

$h_\theta(x) = 1.5$ if $x = 3$

Now let's draw line according to these $h_\theta(x)$ predicted points



Now let's calculate cost function for this

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

$$J(\theta_1) = \frac{1}{2 \times 3} \left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right]$$

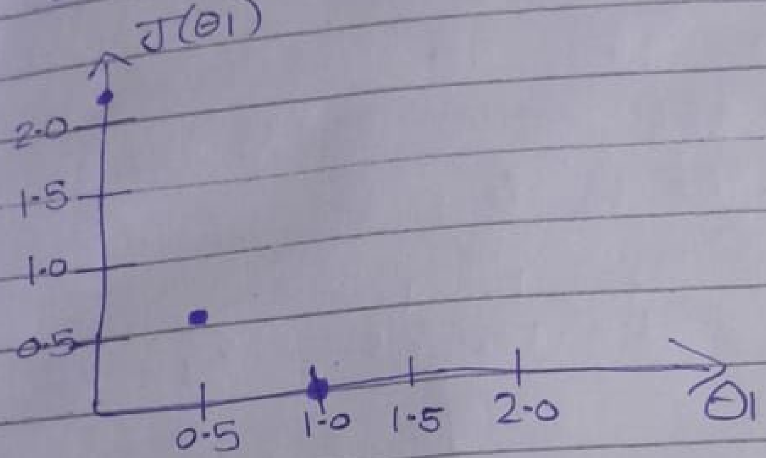
$$\boxed{J(\theta_1) = 0.58 \text{ for } \theta_1 = 0.5}$$

We know,

$J(\theta_1) = 0$ when $\theta_1 = 1$

$J(\theta_1) = 0.58$ when $\theta_1 = 0.5$

Now let's also draw a graph between θ_1 and $J(\theta_1)$



Now let

$\theta_1 = 0$

$h_\theta(x) = 0$ if $x = 1$

$h_\theta(x) = 0$ if $x = 2$

$h_\theta(x) = 0$ if $x = 3$

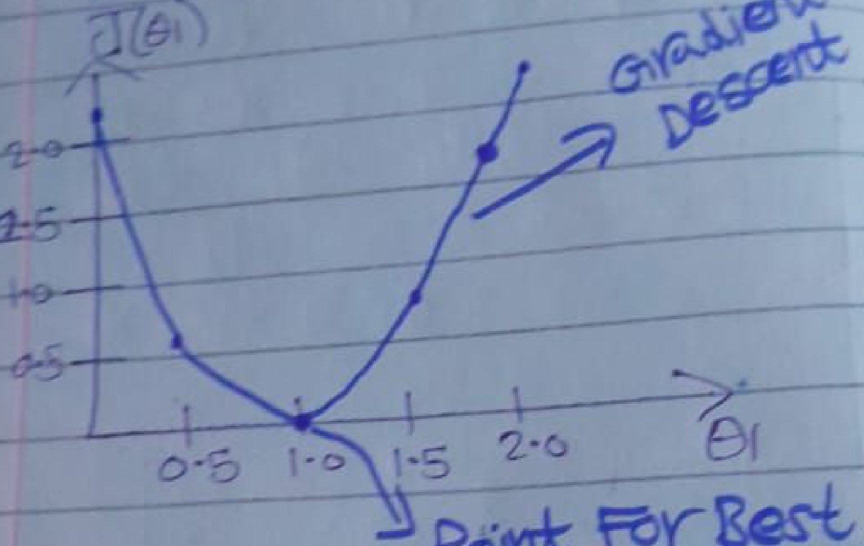
$$J(\theta_1) = \frac{1}{2 \times 3} \left[(0-1)^2 + (0-2)^2 + (0-3)^2 \right]$$

$$J(\theta_1) = \frac{14}{6}$$

$$\boxed{J(\theta_1) = 2.33 \text{ for } \theta_1 = 0}$$

As we have to pick the best fit line with minimal error so we will pick line with $\boxed{\theta_1 = 1}$

Now if we keep on trying with different θ_1 values and find cost we will get graph like this



Point For Best Fit line called Global Minima (error is minimized)

⇒ Convergence Algorithm

- The main aim of convergence algorithm is to optimize the changes of θ_1 values to find best fit line with minimal cost function (error)

Statement:

Convergence algorithm states that:

Repeat until convergence (converge/reach at global minima)

$$\{$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_j)$$

$$\}$$

Now this

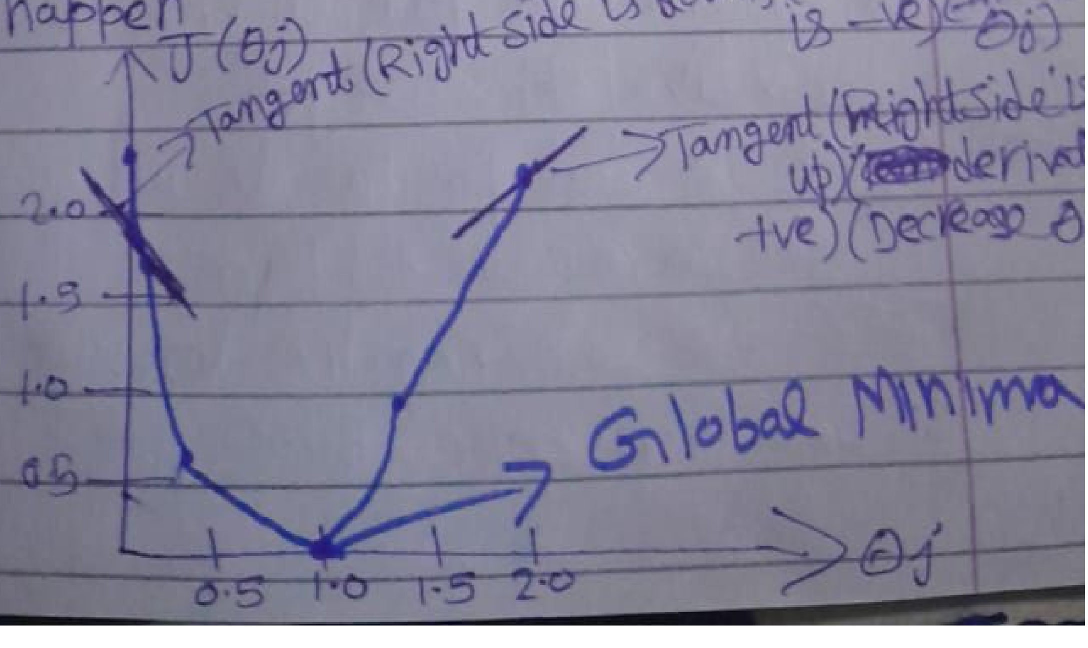
$\frac{\partial}{\partial \theta_j} J(\theta_j) \Rightarrow$ Derivative $\Rightarrow$ Slope

α = Learning Rate

In learning rate we select a smaller value usually

$\boxed{\alpha = 0.001} \Rightarrow$ Used in SKlearn library

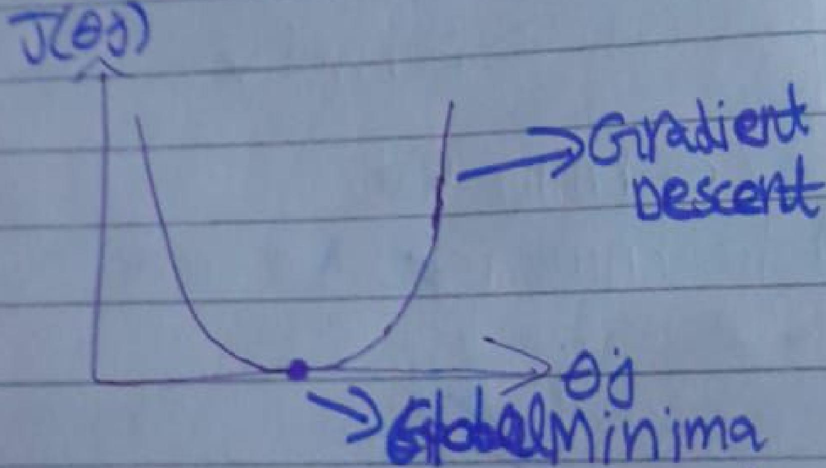
Learning Rate controls the speed at which convergence should happen



Final Conclusion:

① Gradient Descent

Till Now we are taking case that the line pass through origin so $\theta_0 = 0$ (intercept) and θ_1 (changable) (slope). For this case our gradient descent is 2D



And our aim is to reach Global Minima using Convergence theorem depending on the derivative/slope we get

⇒ If we get positive derivative we decrease θ_j by α

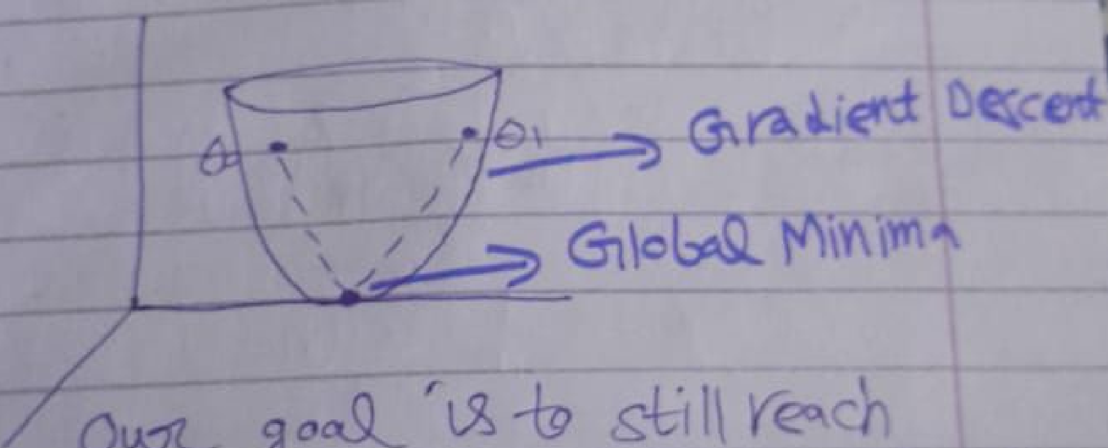
⇒ If we get negative derivative we increase θ_j by α

But le
line is
origin
hal
Now
Now
be
thi

But let's consider now the line is not passing through origin so equation will be

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

Now θ_0 and θ_1 both are changable
Now our gradient descent will be 3D and look something like this



Our goal is to still reach global minima. Now convergence algorithm will be:

Convergence Algorithm:

Repeat until converjence
{

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

}

where we know $J(\theta_0, \theta_1)$ is cost funct

$$J(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=0}^{m}\left(h_\theta(x)^{(i)}-y^{(i)}\right)^2$$

So,

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

For $j=0$

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

$$\therefore \frac{\partial}{\partial x}x^n=n\,x^{n-1}$$

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

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

$\therefore$

$$\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1)=\frac{\partial}{\partial\theta_j}\left(\frac{1}{2m}\sum_{i=1}^{m}\left[(\theta_0+\theta_1 x)^{(i)}-y^{(i)}\right]^2\right)$$

For $j=0$

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

$$\therefore \frac{\partial}{\partial x}x^n=n\,x^{n-1}$$

$$\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1)=\frac{1}{\cancel{2}m}\cancel{2}\sum_{i=0}^{m}\left[(\theta_0+\theta_1 x)^{(i)}-y^{(i)}\right]$$

$$\boxed{\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=0}^{m} \left[(\theta_0 + \theta_1 x)^{(i)} - y^{(i)} \right]}$$

For $j=1$

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

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{2m} \times 2 \sum_{i=1}^{m} \left[(\theta_0 + \theta_1 x)^{(i)} - y^{(i)} \right] (x)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left[(\theta_0 + \theta_1 x)^{(i)} - y^{(i)} \right] (x^{(i)})$$

- Convergence Theorem

Repeat until convergence

$\{$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)} \right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x)^{(i)} - y^{(i)} \right) x^{(i)}$$

$\}$

⇒ Multiple Linear Regression:

Let a data set

House Pricing Dataset

$\underbrace{\text{No. of Rooms} \quad \text{Size of house} \quad \text{Location}}_{\text{Independent Features}} \quad \underbrace{\text{Price}}_{\text{output Features}}$

- When we have more than one independent features we call/use Multiple Linear Regression
- Equation of best fit line for multiple linear regression will be:

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x + \theta_3 x$$

$\theta_0 \Rightarrow$ Intercept

$\theta_1, \theta_2, \theta_3 \Rightarrow$ Coefficients/slopes

⇒ Performance Matrix used in Linear Regression

① R squared

② Adjusted R squared

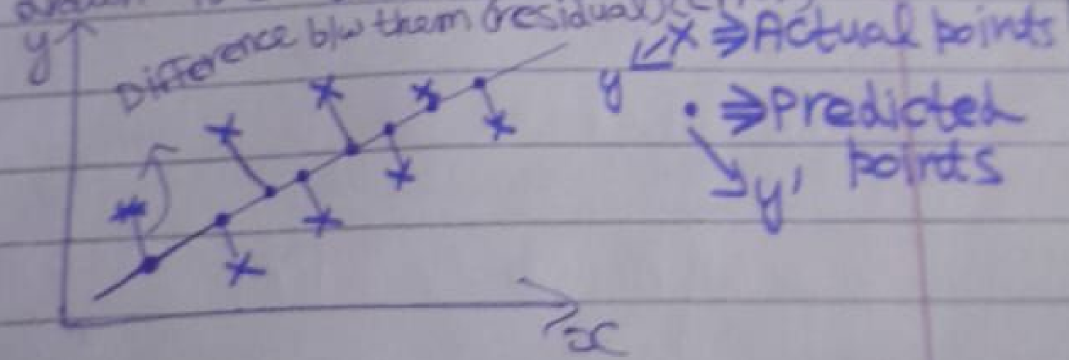
① Rsquared

$$\text{Rsquared} = 1 - \frac{SS_{res}}{SS_{total}} \quad \text{——(i)}$$

(sum of square residual)

(sum of square total)

Let we have a dataset and we drawn it's best fit line



so,

$$\text{Rsquared} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

according to y-axis

where $\bar{y}$ is the average of y points ↑

we know $\sum (y_i - \hat{y})^2$ is small number

and $\sum (y_i - \bar{y})^2$ is large number

so,

$$R\text{squared} = 1 - \frac{\text{small number}}{\text{large number}}$$

Rsquared $= 1 -$ small number

So Rsquared ≈ 1

means Rsquared can be 0.75, 0.8 the more Rsquared is closer to 1 the more the performance of linear Regression

(2) Adjusted R square

Let Dataset

Independent feature	Dependent/ output feature
Size of house	Price

We know as size of house increases Price also increases means there is a **direct positive correlation** b/w them

⇒ In this case if we calculate Rsquared Let

Rsquared $= 75\% = 0.75$

Now if we add another independent feature No. of Rooms we know
No. of Rooms ↑ Price ↑ so
direct positive correlation
This time if we calculate:
Rsquared = 80% = 0.80
Now if we again add another independent feature Location
we know Location ↑ Price ↑ so
direct positive correlation
means Rsquared again going to increase let this time
Rsquared = 85% = 0.85

Now if we add another independent feature Gender and we know it is not going to affect the Price
means no correlation
But even if now we again calculate Rsquared it again increases but by a small number
Rsquared = 87% = 0.87

Problem of R squared:
Even if there is no correlation b/w independent feature and output feature the R squared value still increase means as more no. of features it will increase.

- To solve this problem we use **Adjusted R squared**

Adjusted R squared:

$$\text{Adjusted R squared} = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$

where

N ⇒ No. of data points
P ⇒ No. of Independent Features
R^2 ⇒ R squared

★ In adjusted R squared if feature added has correlation with output feature it's value increases and if no correlation it's value decreases.

⇒ MSE, MAE, RMSE
- MSE ⇒ Mean Squared Error
- MAE ⇒ Mean Absolute Error
- RMSE ⇒ Root Mean Squared Error

Let dataset

Experience (Independent) Salary (Dependent).

We have different cost functions mentioned above to calculate the error

$y \leftarrow \times \Rightarrow$ Actual
$y' \leftarrow \bullet \Rightarrow$ predicted

① MSE (Mean Squared Error)

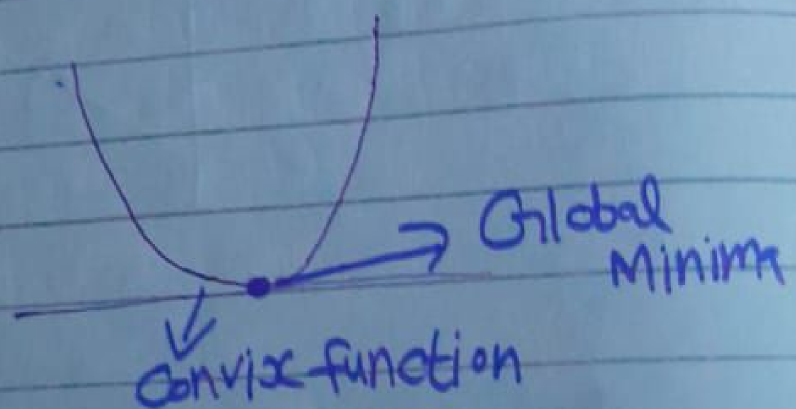
$$MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}$$

- This cost function should be reduced

• Advantages of MSE

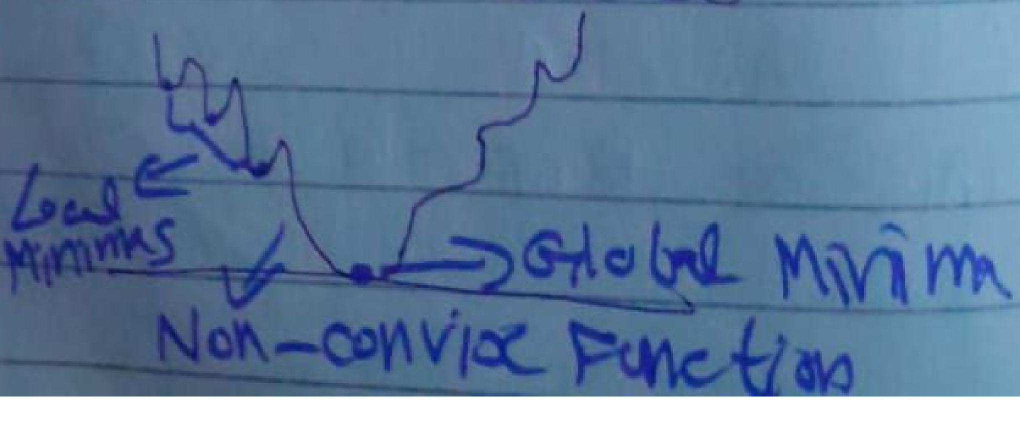
1. Differentiable
2. It has one local and one global minima (So convergence can't go stuck at a local minima)
3. Converges faster

The reason of all these advantages are: because in MSE $(y_i-\hat{y})^2$ is quadratic equation when we plot it it is like:



So it has only one local and global minima and is differentiable at all points

Where in other case



* It has many local minimas so it's convergence can get stuck

• Disadvantages of MSE

① Not Robust to outliers (because it penalize the outliers means extend line to get difference b/w them due to which MSE get increased)

② When we take square while taking MSE the units change like y is price whose unit is lakh when we take $(y_i-\hat{y}_i)^2$ it becomes $lakh^2$. Means it is not in the same unit

② Mean Absolute Error (MAE):

(absolute or mode of $y_i-\hat{y}$)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}|$$

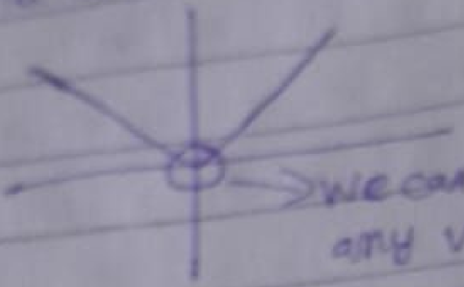
Advantages of MAE:

① Robust to outliers (As MAE don't increase much)

② It will be in the same unit

Disadvantages of MAE:

In MAE as there is no squaring we will get this kind of curve



→ we cannot differentiate any value at zero

So we use subgradients

① Convergence usually takes more time. (optimization i.e converting to subgradients is complex task)

② Time Consuming

3. RMSE (Root Mean Squared Error):

$RMSE = \sqrt{MSE}$

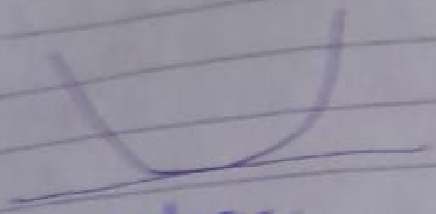
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2}$$

Advantages of RMSE

1. Same Unit
2. Differentiable

As we are taking square and

so



- Disadvantage:
  1. Not robust to outliers

⇒ Overfitting and Underfitting (Bias and Variance)

1. Training dataset
2. Test dataset
3. Validation dataset

Let I have a dataset

Dataset
$\boxed{\text{1000 datapoint}}$

and we have to train a model:

⇒ We will split our entire dataset to two important parts:

Dataset (1000 datapoints) → Training Dataset (700) → Tra Mo

Dataset (1000 datapoints) → Test Dataset (300) → Test Mod

Let we split by 7:3

⇒ Now we take our training dataset and further split it into 2 parts

Training dataset → Train → Train the model

Training dataset → Validation ⇒ Hyperparameter Tunning your model

Let I have a model which

Train data → Model gives very good accuracy → Low Bias

Test data → Model gives very good accuracy → Low variance

* This is the model we want and called **Generalized Model**

But let I have a model which

Train data → Very good Accuracy (90%) → Low Bias

Test data → Bad Accuracy (50%) → High Variance

⇓

Model is overfitting

* Model is overfitting when model is trained well with trained dataset but giving bad accuracy for new test data

Now let we have model which:

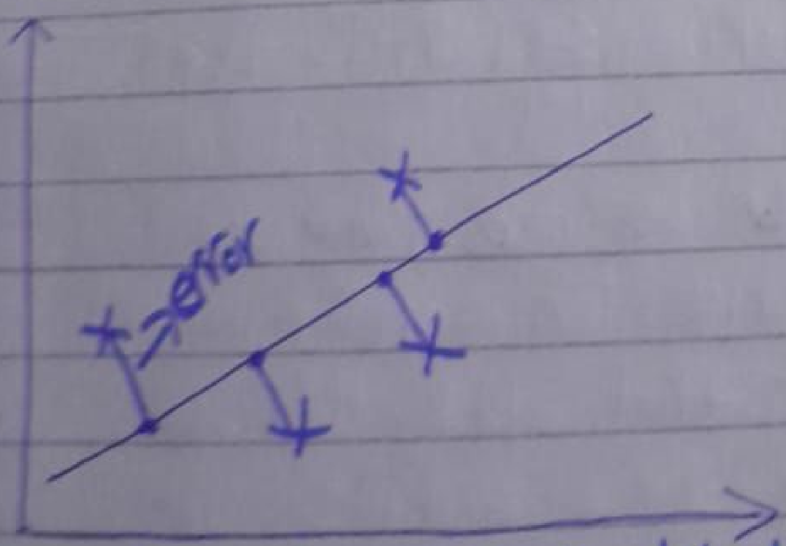
Train Data → Model accuracy is low → High Bias
Test Data → Model accuracy is low → High Variance

⇓

Model is underfitting
(Model is not trained well with training data)

Generalized Model → Low Bias, Low Variance
Overfitting → Low Bias, High Variance
Underfitting → High Bias, High Variance

⇒ Linear Regression using OLS (Ordinary Least Square)



x = Actual correct datapoint
• = Predicted datapoints

* The aim of OLS (Ordinary Least Square) is to reduce the error and it provides the formula to calculate the formula to calculate **slope** and **intercept** for **Best fit line**

The equation for line is

$$\boxed{h_\theta(x) = \beta_0 + \beta_1 x}$$

$\beta_0 \Rightarrow$ Intercept

$\beta_1 \Rightarrow$ Slope

Ordinary Least Square

$$S(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h_\theta(x)\right)^2$$

$$S(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (\beta_0 + \beta_1 x_i)\right)^2$$

$$S(\beta_0, \beta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \beta_0 - \beta_1 x_i\right)^2 \quad \text{— (i)}$$

Now we are going to derive formula for β_0 and β_1

For that apply derivative on b·s of equ (i)

$$\frac{\partial}{\partial \beta_0} S(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_0}\left(\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2\right)$$

$\because$

$$\frac{\partial}{\partial x}(x^2) = 2x^{2-1} \frac{\partial}{\partial x}(x)$$

$$\frac{\partial}{\partial \beta_0}(\beta_0, \beta_1) = \frac{1}{n}(2) \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)$$

$$\left(\frac{\partial}{\partial \beta_0}(y_i) - \frac{\partial}{\partial \beta_0}(\beta_0) - \frac{\partial}{\partial \beta_0}(\beta_1 x_i)\right)$$

$$\frac{\partial}{\partial \beta_0}(\beta_0, \beta_1) = \frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(0 - 1 - 0)$$

$$\frac{\partial}{\partial \beta_0}(\beta_0, \beta_1) = \frac{-2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)$$

Equate it to zero

$$\frac{-2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0 \quad \text{—— (i)}$$

$$\frac{\partial}{\partial \beta_1} S(\beta_0, \beta_1) = \frac{\partial}{\partial \beta_1}\left(\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2\right)$$

$$\frac{\partial}{\partial \beta_1} S(\beta_0, \beta_1) = \frac{1}{n}(2) \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)$$

$$\left(\frac{\partial}{\partial \beta_1}(y_i) - \frac{\partial}{\partial \beta_1}(\beta_0) - \frac{\partial}{\partial \beta_1}(\beta_1 x_i)\right)$$

$$\frac{\partial}{\partial \beta_1} S(\beta_0, \beta_1) = \frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(0 - 0 - x_i)$$

$$\frac{\partial}{\partial \beta_1} S(\beta_0, \beta_1) = \cancel{\frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(-x_i)}$$

$$\cancel{-\frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(+x_i) = 0} \quad \text{—— (2)}$$

Now by taking derivative of β_0 and β_1 we got two equations

$$-\frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0 \quad \text{—— (1)}$$

$$-\frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(x_i) = 0 \quad \text{—— (2)}$$

Take equ(i)

$$-\frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$-\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = \frac{0 \times n}{2}$$

$$-\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0$$

Apply limit (summation) inside

$$-\sum_{i=1}^{n} y_i + n \times \beta_0 + \beta_1 \sum_{i=1}^{n} x_i = 0$$

$$n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$\beta_0 = \frac{\sum_{i=1}^{n} y_i}{n} - \frac{\beta_1 \sum_{i=1}^{n} x_i}{n}$$

$$\therefore \bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$$

So

$$\boxed{\beta_0 = \bar{y} - \beta_1 \bar{x}} \longrightarrow \text{Intercept}$$

Now let's take equ(ii)

$$-\frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(x_i) = 0$$

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(x_i) = \frac{-0 \times n}{2}$$

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)(x_i) = 0$$

$$\sum_{i=1}^{n} x_i y_i - \beta_0 \sum_{i=1}^{n} x_i - \beta_1 \sum_{i=1}^{n} x_i^2 = 0$$

$$\sum_{i=1}^{n} (x_i y_i - \beta_0 x_i - \beta_1 x_i^2) = 0$$

$$\therefore \beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\sum_{i=1}^{n} (x_i y_i - (\bar{y} - \beta_1 \bar{x}) x_i - \beta_1 x_i^2) = 0$$

$$\sum_{i=1}^{n} (x_i y_i - x_i \bar{y} + \beta_1 \bar{x} x_i - \beta_1 x_i^2) = 0$$

$$\sum_{i=1}^{n} (y_i - \bar{y} + \beta_1 \bar{x} - \beta_1 x_i) x_i = 0$$

$$\sum_{i=1}^{n} \left((y_i - \bar{y}) + \beta_1 (\bar{x} - x_i) \right) = 0$$

$$\sum_{i=1}^{n} (y_i - \bar{y}) + \beta_1 \sum_{i=1}^{n} (\bar{x} - x_i) = 0$$

$$\beta_1 \sum_{i=1}^{n} (\bar{x} - x_i) = - \sum_{i=1}^{n} (y_i - \bar{y})$$

$$\beta_1 = \frac{- \sum_{i=1}^{n} (y_i - \bar{y})}{\sum_{i=1}^{n} (\bar{x} - x_i)}$$

$$\beta_1 = \frac{+ \sum_{i=1}^{n} (y_i - \bar{y})}{+ \sum_{i=1}^{n} (x_i - \bar{x})}$$

$$\boxed{\beta_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})}} \Rightarrow \text{Slope/ Coefficient}$$

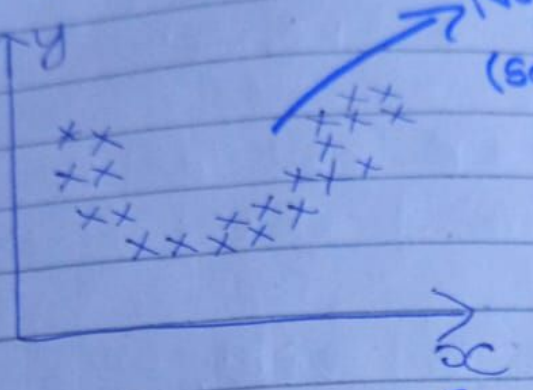
So by OLS (Ordinary Least Squared)

Intercept $\Rightarrow \boxed{\beta_0 = \bar{y} - \beta_1 \bar{x}}$

Slope/coefficient $\Rightarrow \boxed{\beta_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})}}$

$\Rightarrow$ P

⇒ Polynomial Regression:



$h_\theta(x) = \beta_0 + \beta_1 x$ — Simple Linear Regression

$h_\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ — Multiple Linear Regression

★ When we have non-linear relationships we use Polynomial Regression.

Degrees in Polynomial:

⇒ Simple Polynomial Regression (1 input and 1 output feature)

- **Degree = 0**

  $h_\theta(x) = \beta_0 \times x^0$

- **Degree = 1**

  $h_\theta(x) = \beta_0 \times x^0 + \beta_1 \times x^1$

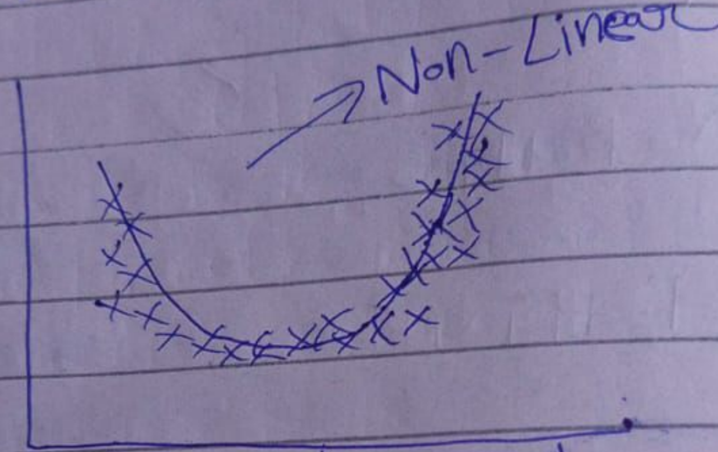
- **Degree = 2**

  $h_\theta(x) = \beta_0 \times x^0 + \beta_1 \times x^1 + \beta_2 \times x^2$

Degree=n

$$h_\theta(x) = \beta_0 \times x^0 + \beta_1 \times x^1 + \beta_2 \times x^2 + \beta_3 \times x^3 + \dots + \beta_n \times x^n$$

* Now for best fit line we have to select degrees



* We have discussed Simple Polynomial Regression till now. What if we have 2 independent and 1 dependent feature

Degree=1 $\Rightarrow h_\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

Degree=2

$$h_\theta(x) = \beta_0 \times x^0 + \beta_1 \times x_1^1 + \beta_2 \times x_2^1 + \beta_3 \times x_1^2 + \beta_4 \times x_2^2$$