# Simple Linear Regression

Supervised ML → Regression

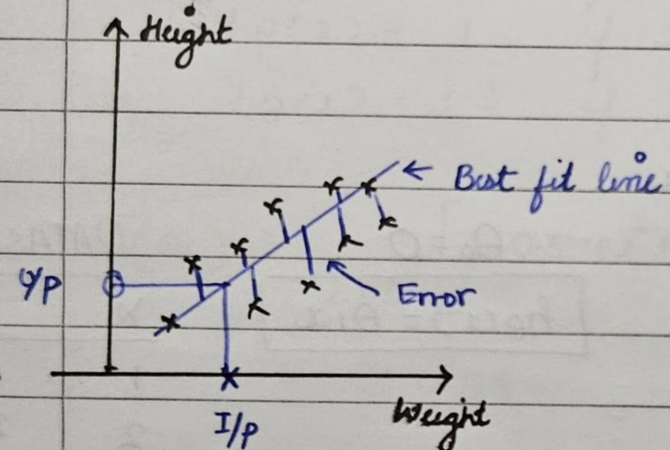
**Dataset:** ↗ Independent ↗ dependent

| Weight | Height |
|---|---|
| 74 | 170 cm |
| 80 | 180 cm |
| 75 | 175 cm |
| – | – |

TRAIN ↓

New weight → Model → Height

↑ Simple Linear Regression




$y = mx + c$

$y = \beta_0 + \beta_1 x$

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

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

$h_\theta(x)$ ⇓ $\{\hat{y}\}$ ↓ Predicted points

$\theta_0$ = Intercept

$\theta_1$ = slope or coefficient

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

## Cost function :

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

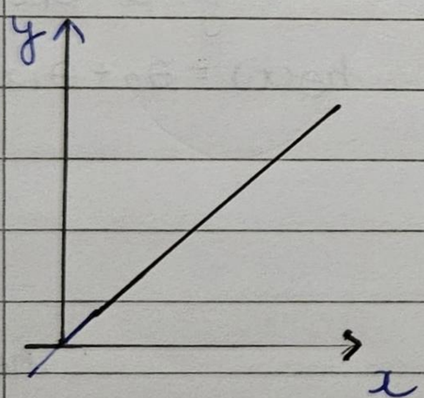
$h_\theta(x)^{(i)}$ ↓ Predicted

$y^{(i)}$ ↓ True o/p

## Final Aim :

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

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

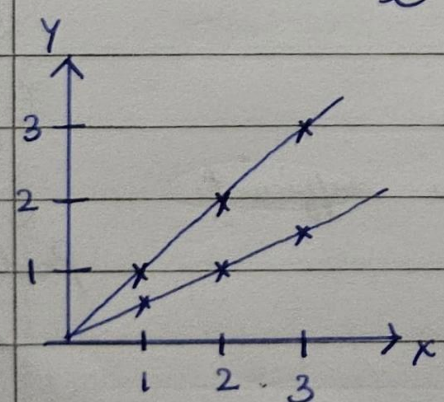



$\theta_0 = 0$

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

DATASET

| x | y |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |




let $\theta_1 = 1$ {slope}

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

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

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

__/__/__

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

$$= \frac{1}{2 \times 3} \sum_{i=1}^{3} \left( (1-1)^2 + (2-2)^2 + (3-3)^2 \right)$$

$$= 0$$

Now let say, $\theta_1 = 0.5$

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

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

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

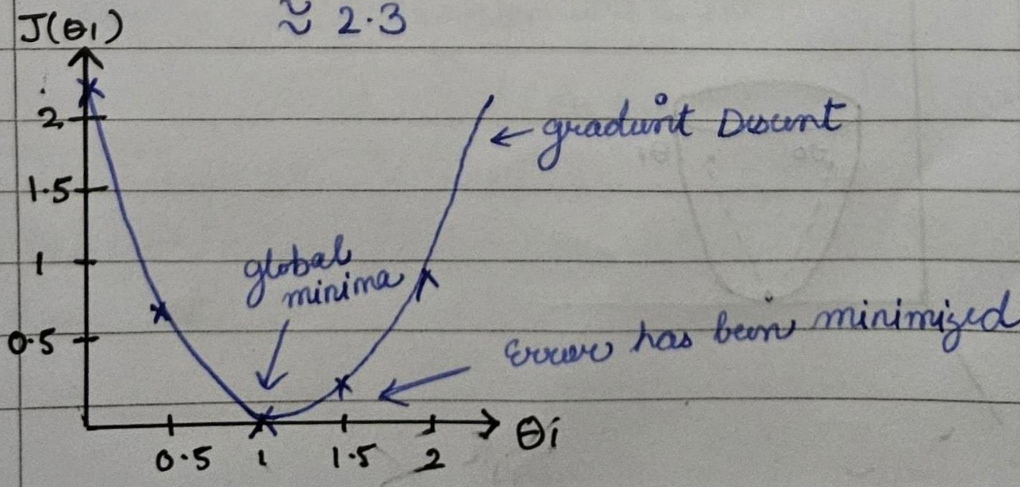
$$J(\theta_1) = \frac{1}{2 \times 3} \left[ (0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right]$$

$$= \simeq 0.58$$

Now if $\theta_1 = 0$

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

$$\simeq 2.3$$



__/__/____

## Convergence Algorithm :

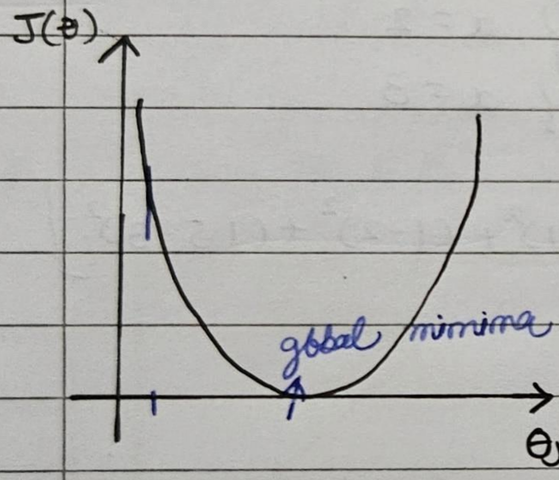
Optimize the changes of $\theta_1$ values.

Repeat until Convergence

$$\{$$

$$\theta_j = \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_j)} \rightarrow \text{-ve or +ve depends}$$

$$\}$$

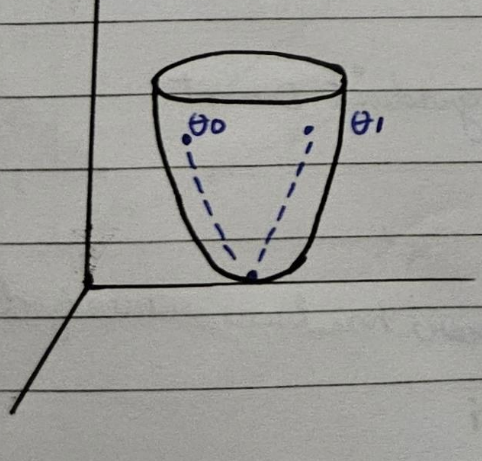



$\alpha$ = Learning rate (Should select a smaller value)

$\alpha = 0.001$

Learning rate controls the speed of convergence.

Final Conclusion :



__/__/__

Convergence Algorithm :

repeat until Convergence

{

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

}

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

if $J=0$

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

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

if $J=1$

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

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

$$\boxed{\frac{\partial}{\partial \theta_1} [\theta_0 + \theta_1 x] = x}$$

Repeat until convergence

{

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

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

}

→ **Multiple Linear Regression:**

House Pricing Dataset :

$\underbrace{\text{No. of rooms} \quad \text{Size of House} \quad \text{Location}}_{\text{Independent features}}$    Price (o/p feature)

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ {Multiple Linear Regression}

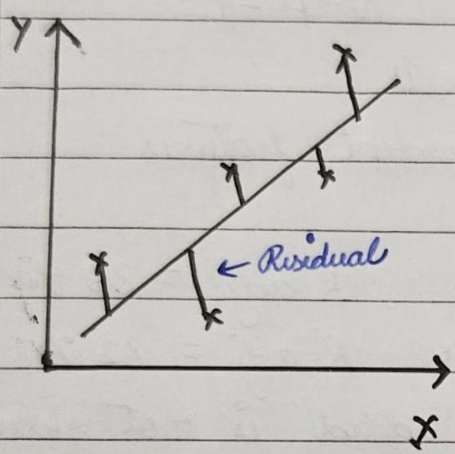
$\theta_1, \theta_2, \theta_3$ = Coefficient

$\theta_0$ = intercept

_/_/_

# Performance Metrics in Linear Regression

1. R-Squared $= 1 - \frac{SS\ Res}{SS\ Total}$




$$= 1 - \frac{\Sigma (y_i - \hat{y_i})^2}{\Sigma (y_i - \bar{y_i})^2}$$

$$= 1 - \frac{\text{small number}}{\text{bigger number}}$$

2. Adjusted R-squared

Size of house ↑ Price ↑

(+ve corelation)

Dataset

Size of the House    Price

let R - squared = 75 % = 0.75

if we add one more feature No. of bedrooms then if No. of bedrooms ↑ Price ↑

R-squared = 80 % = 0.80

if we add one more feature like "gender" which does not make sense with price then also R-squared value will increase.

R- squared = 83 % = 0.83

This is the problem with R-squared.

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

N = No. of data points
P = No. of Independent features

$$\begin{cases} P=2 & R^2 = 90\% & R^2 adj = 86\% \\ P=3 & R^2 = 92\% & R^2 adj = 82\% \\ & \uparrow \text{ if one feature added is not relevant} \end{cases}$$

3. Mean Squared Error:

(MSE) $= \sum_{i=1}^{n} \frac{(y-\hat{y})^2}{n}$ → Quadratic Equation

Advantages:

1. Differentiable
2. One local & one global minima
3. Converges faster

Disadvantages:

1. Not robust to outliers
2. It is no longer in the same units.

4. Mean Absolute Error:

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

Advantages:

1. Robust to outliers
2. It will be in the same unit

Disadvantage:

1. Convergence usually takes more time. Optimization is a complex task.
2. Time Consuming

5. Root Mean Squared Error :

$$RMSE = \sqrt{MSE}$$

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

Advantage:

1. Same Unit
2. Differentiable

Disadvantage:

1. Not robust to outliers.

## Overfitting and Underfitting (Bias and Variance)

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

<u>Dataset</u>

70% - 30%

1000 datapoints
- → Training dataset (700)
  - ↳ Model to train
- → Test dataset (300)
  - ↳ Testing model

TRAINING DATASET
- TRAIN
  - ↓ Train the model
- VALIDATION
  - ↓ Hyperparameter tuning your model

(90%) ↗ Low Bias

TRAIN → Very good accuracy

TEST → Very good accuracy

(85%) ⇑ ↘ Low variance

Generalized Model

Low Bias

Very good accuracy (90%)

Bad accuracy (50%) ↘ High variance

⇓

Model is overfitting

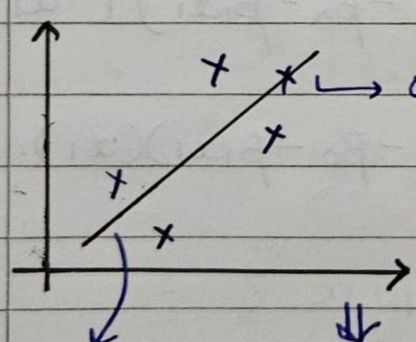
Train → Model accuracy is low [High Bias]

Test → Model accuracy is low [High Variance]

⇓

Model is underfitting

→ Linear Regression using OLS {Ordinary least Square}



↳ optimization

⇓

Loss Function

⇓

$h_\theta(x) = \beta_0 + \beta_1 x$ → OLS → apply formula & calculate $\beta_0$ and $\beta_1$

↑ intercept ↑ cofficient

Aim of the OLS is to reduce the error.

Ordinary Least Square  Error  MSE

$$S(\beta_0, \beta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_1)^2$$

Aim : $\beta_0$ & $\beta_1$

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = \frac{2}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_1)(0 - 1 - 0)$$

$$= -\frac{2}{n} \sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_1) = 0 \quad - (1)$$

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = \frac{2}{n} \sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_1)(-x_1) = 0 \quad - (2)$$

$$= -\frac{2}{n} \sum_{i=1}^{m} (y_i - \beta_0 - \beta_1 x_1)(x_1)$$

Eq$^n$ (1) $\rightarrow$

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

$$\Downarrow$$

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

_/_/_

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

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

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

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

Eq$^n$ 2 $\rightarrow$

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

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

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

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

$$\Downarrow$$

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

$$\sum_{i=1}^{m} \left( x_iy_i - (\bar{y} - \beta_1\bar{x})x_i - \beta_1 x_i^2 \right) = 0$$

$$\sum_{i=1}^{m} \left( x_iy_i - x_i\bar{y} + \beta_1\bar{x}x_i - \beta_1 x_i^2 \right) = 0$$

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

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

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

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

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

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

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

Cofficient

**OLS**

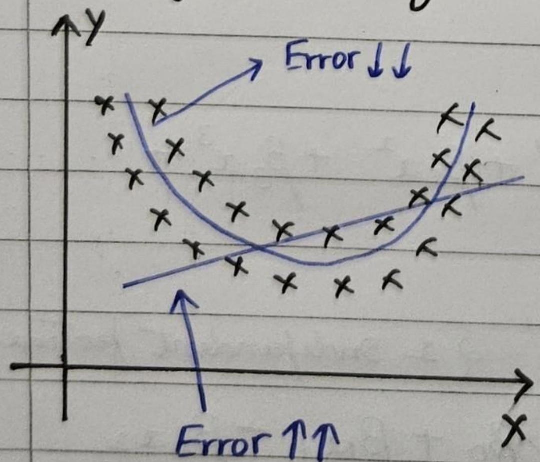
Intercept

↓

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

OLS $\approx$ Linear Regression (SK Learn)

→ Polynomial Regression




$h_\theta(x) = \beta_0 + \beta_1 x$

Simple Linear Regression

$h_\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

Multiple Linear Regression

__/__/__

## Polynomial Degrees:

### Simple Polynomial Regression {1 I/P and 1 O/P}

→ deg = 0

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

$h\theta(x) = \beta_0 \Rightarrow$ constant value

→ deg = 1

$h\theta(x) = \beta_0 \times x^0 + \beta_1 \times x^1$ → Simple Linear Reg.

→ deg = 2

$$h\theta(x) = \beta_0 \cdot x^0 + \beta_1 \cdot x^1 + \beta_2 \cdot x^2$$

→ deg = n

$$h\theta(x) = \beta_0 \cdot x^0 + \beta_1 x^1 + \beta_2 x^2 + \beta_3 x^3 + \ldots\ldots \beta_n x^n$$

{2 Independent feature}

* degree = 1, $h\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

* degree = 2, $h\theta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2$