Simple Linear Regression

Supervised ML → Regression
Supervised ML → Classification

Dataset

↓ Independent features

Dependent feature ⇓ (Pass/Fail ⇐ o/p)

Play hours	Study hours	Pass/Fail
7	5	Pass
7	2	Fail
3	8	Pass

⇒ Classification.

Play hours, Study hours → Model → P/F

TRAIN ⇓ Model

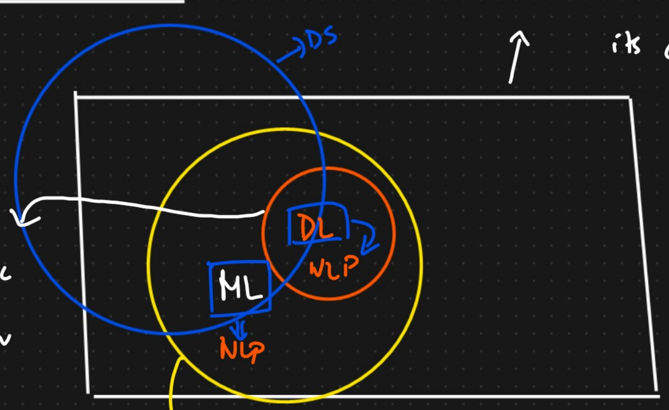
Model ⇓ Acc↑↑

House price prediction

No. of Rooms	House size	Price
—	—	150K
—	—	155K
—	—	140K

Continuous value ⇒ Regression problem Statement

AI Vs ML Vs DL Vs DS



Mimic
Human
Brain

Multi Layered Neural N/w

Statistical to perform explore, analyze, visualize and perform prediction.

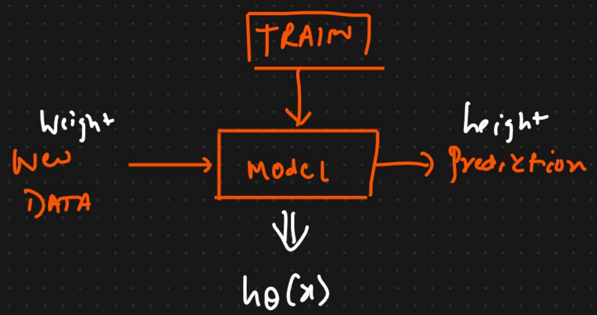
AI → Smart application that can perform its own task without any human intervention

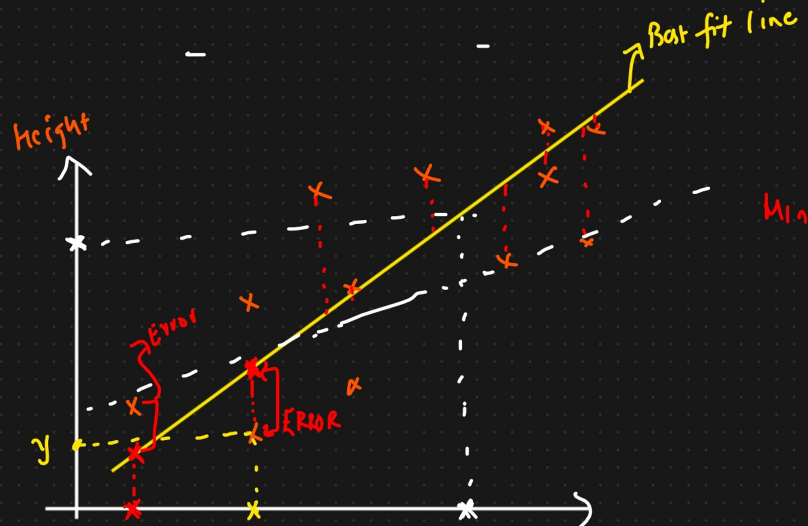
Eg: Self Driving Car, Chatgpt, Alexa, SIRI, Google Home

Eg:- Recommendation system
Weather prediction
Spam detection
Disease prediction

Simple Linear Regression

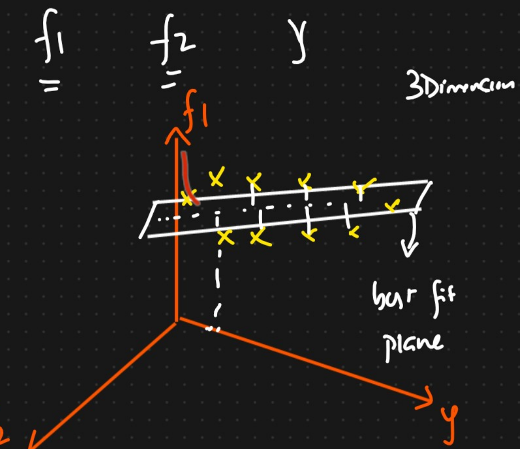
↓ Independent feature. Weight	Dependent feature ↓ Height (y) Actual value
74	170
80	180cm
75	175.5cm
–	–
–	–

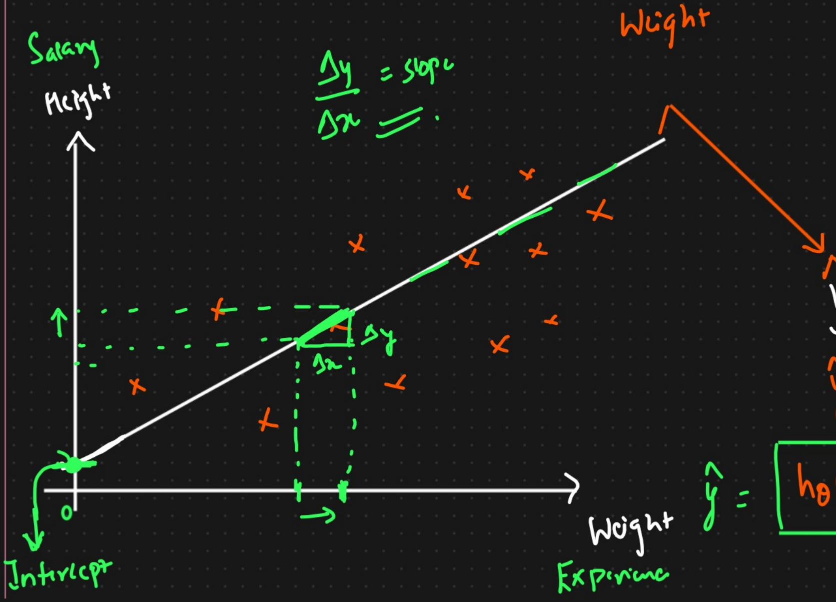




3D PLANE

Multiple Linear Regression

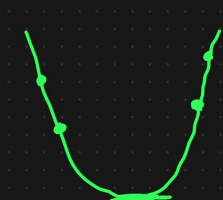




$\hat{y} = mx + c$

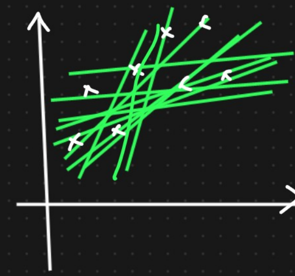
$\hat{y} = \beta_0 + \beta_1 x$

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



intercept = 0

$$\left\{ \begin{aligned} \theta_0 &= \text{Intercept} \\ \theta_1 &= \text{Slope or Coefficient} \end{aligned} \right\}$$



$\boxed{\theta_0 \;\&\; \theta_1}$

Cost function → n data

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} \big(y_i - \underbrace{h_\theta(x)}_{\hat{y}}\big)^2$$ [Mean Squared Error].

$$J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (\underbrace{y_i}_{\text{Actual}} - \underbrace{\hat{y}_i}_{\text{predicted.}})^2$$

n = no. of datapoints

y_i ⇒ Actual value

$h_\theta(x)$ = predicted value

Loss function = $(y_i - \hat{y}_i)^2$ {1 data point}.

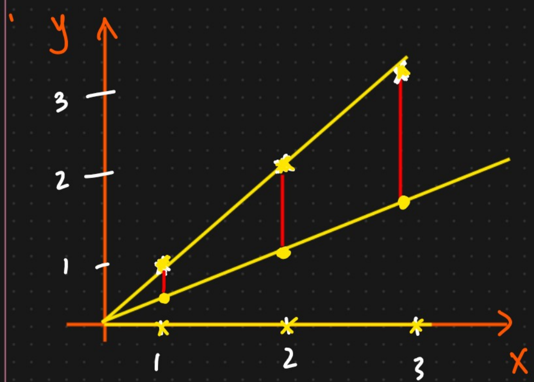
Final Aim

$$\underset{\theta_0, \theta_1}{\text{Minimize}} \; J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_\theta(x))^2 \downarrow\downarrow\downarrow$$

Optimization {Minimizing the Cost function}

DATASET

x	y
1	1
2	2
3	3

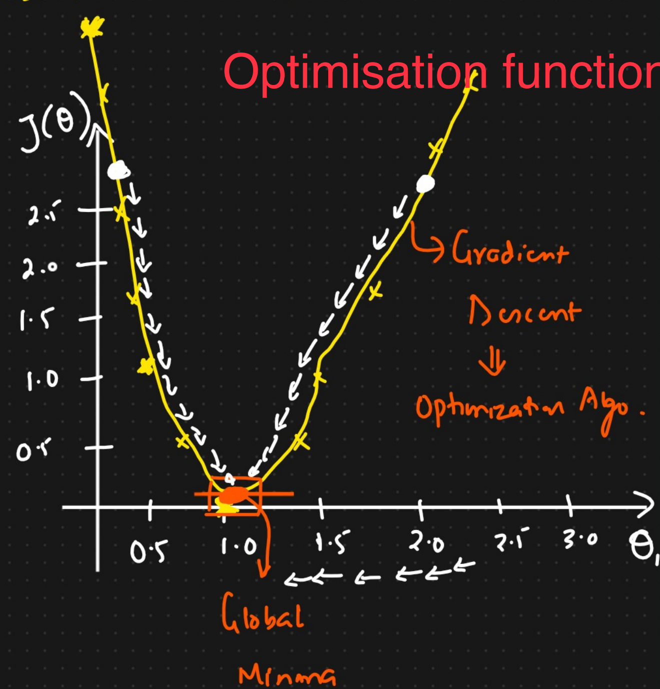


$h_\theta(x) = \theta_0 + \theta_1 x_1$

Consider $\theta_0 = 0$

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

	Let $\theta_1 = 1$	Let $\theta_1 = 0.5$	Let $\theta_1 = 0$
$x_1 = 1$	$h_\theta(x) = 1(1) = 1$	$h_\theta(x) = 0.5$	$h_\theta(x) = 0$
$x_1 = 2$	$h_\theta(x) = 1(2) = 2$	$h_\theta(x) = 1.0$	$h_\theta(x) = 0$
$x_1 = 3$	$h_\theta(x) = 1(3) = 3$	$h_\theta(x) = 1.5$	$h_\theta(x) = 0$



Cost function

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

$n = 3$

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

$$= 0$$

Cost fn $\quad \theta_1 = 0.5$

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

$$J(\theta_1) = 1.16$$

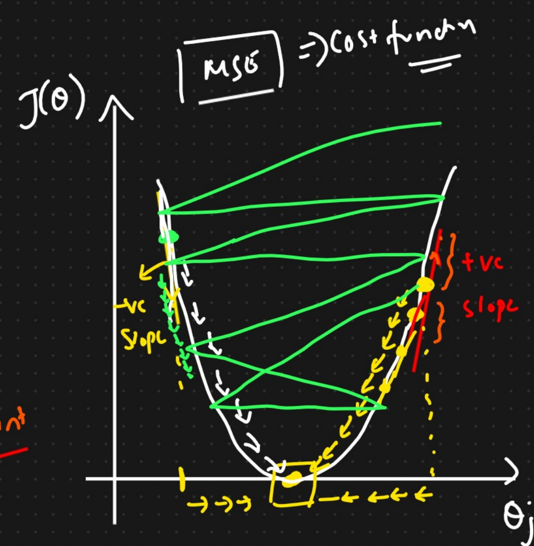
Convergence Algorithm

Repeat until Convergence Global Minima

{

$$\theta_j : \theta_j - \underbrace{\boxed{\alpha}}_{\text{Learning Rate}} \boxed{\frac{\partial}{\partial \theta_j} J(\theta_j)} \Rightarrow \text{Slope at a point}$$

}



$\Uparrow$ huge

$\theta_j := \theta_j - \alpha\,(\text{+ve value})$

$\boxed{\alpha = 0.01} \Leftarrow$

$\alpha \Rightarrow$ Learning Rate $\Rightarrow 1.00$

$\Downarrow$

Speed of Convergence.

$\theta_j := \theta_j - (\text{+ve value})$

$\theta_{new} << \theta_{old}$

$\theta_j := \theta_j - \alpha\,(-\text{ve value})$

$= \theta_j + \alpha\,(\text{+ve value})$

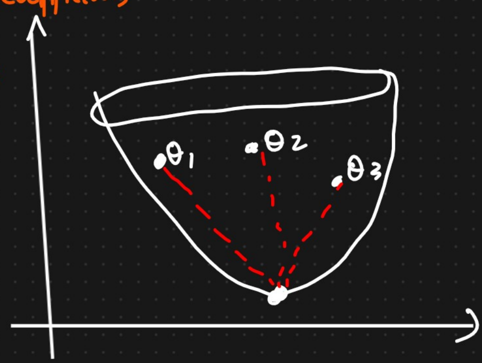
$\theta_{new} >> \theta_{old}$

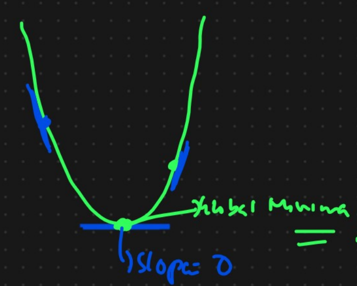
$f_1 \quad f_2 \quad f_3 \quad y$

$h_\theta(x) = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3$ {Multiple Linear Regression}

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

$\theta_0 \Rightarrow$ intercepts



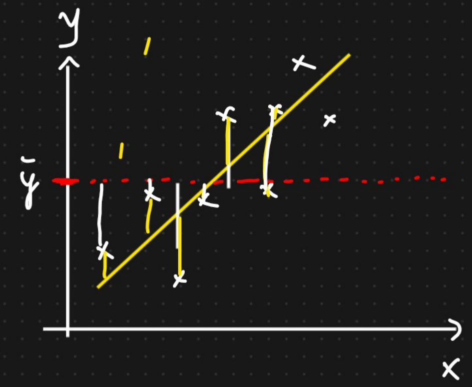


$\boxed{\theta_{new} = \theta_{old}}$

Performance Metrics

① R squared ② Adjusted R Squared

① R Squared



R squared = 1 − $\frac{SS_{Res} \text{ \{Best fit line\}}}{SS_{Total} \text{ \{Average of y\}}}$

SS_{Res} = Sum of Square Residuals or Errors

SS_{Total} = Sum of Squared Total

$$\boxed{R \text{ squared} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}}$$

⇒ Small value / Big value = 0.7 ⇓ 70% Accurate

y mean

f1 f2 f3 f4 f5 O/P

Highly corelated

R squared = 70%

75% ← More Score

76% ← Less corelated so less increase in score

Adjusted R Squared

1. **Model Complexity Control:** Regular R-squared tends to increase as more independent variables are added to the model, even if those variables don't truly improve the model's predictive power. This can lead to overfitting, where the model is too complex and doesn't generalize well to new data. Adjusted R-squared penalizes the inclusion of unnecessary variables by adjusting for model complexity.
2. **Better Model Comparison:** When comparing models with different numbers of independent variables, adjusted R-squared provides a fairer basis for comparison. It allows you to assess whether the addition of a new variable truly adds value to the model by taking into account the trade-off between model fit and complexity.
3. **Prevention of Overfitting:** By penalizing the inclusion of too many independent variables, adjusted R-squared helps prevent overfitting. Overfit models may perform well on the training data but poorly on new data because they've learned noise in the training data.

Size of house | No.of Rooms | Location | Gender | $\rightarrow$ Price

$R^2 = 70\%$, $R^2 = 75\%$ $R^2 = 78\%$ $R^2 = 79\%$

Adjusted R square < R squared

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

N = no.of data points

p = No.of independent predictors

$R^2 = 80\%$ $N = 11$ $p = 2$

$$\text{Adjusted R squared} = 1 - \frac{(1-0.8)(10)}{11-2-1} = 0.75 \Rightarrow 75\%$$

$p=2$ $R^2 = 80\%$ Adjusted $R^2 = 75\%$

$p=3$ $R^2 = 85\%$ Adjusted $R^2 = 78\%$

$p=4$ $R^2 = 86\%$ " " $= 76\%$

$\Downarrow$

Feature is not important