Simple linear Regression

Supervised ML
if output feature is known then supervised ml

- Regression — if o/p feature is continuous then regression supervised learning.
- Classification — if o/p feature is categorical then classification supervised learning.

Dataset

Input feature is independent.

↓ Independent features

Play hours	Study hours
7	5
7	2
3	8

Dependent feature — Output feature is dependent since it is the outcome obtained based on input features.

⇓

Pass/Fail ⇐ o/p

- Pass
- Fail
- Pass

⇒ Classification.

TRAIN ⇓

Play hours →
Study hours →
Model → P/F

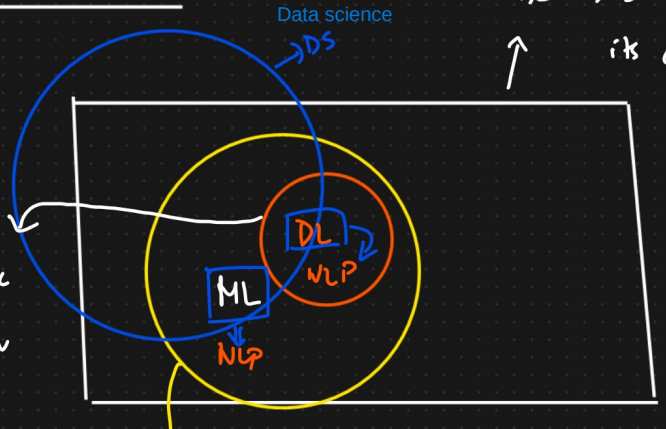
⇓ Acc↑↑

House price prediction

I/P No. of Rooms	I/P House size	⇓ Price O/P
–	–	150K
–	–	175K
–	–	140K

Continuous value ⇒ Regression problem statement

AI Vs ML Vs DL Vs DS



AI is super set where ML is subset of AI and DL is subset of ML.
Data science mix of everything in small portions

DL
Mimic human brain
Multi layered Neural Nw

ML
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

AI product.(Alexa, sophia)
Recommendation system is not a AI product it's just a feature/functionality (based on AI) of end AI product.

When recommendation system functionality is introduced to streaming product such as youtube, netflix then these products becomes AI powered products.

Privacy is just a myth. Our cell phone listens us all the time. If I am not well I will be taking with very low energy which my phone will observe and start showing ads related to doctors nearby.

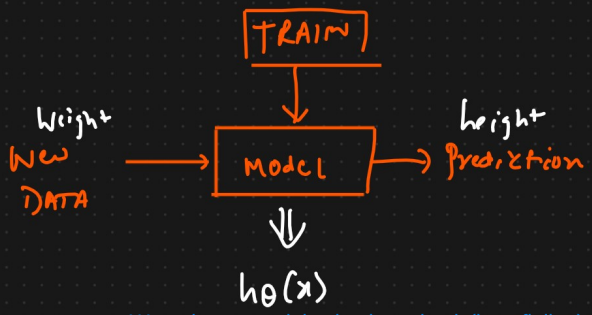
Krish sir shared that once he was taking about selling his phone and then he starts receiveing ads realted to mobile selling.

Simple Linear Regression

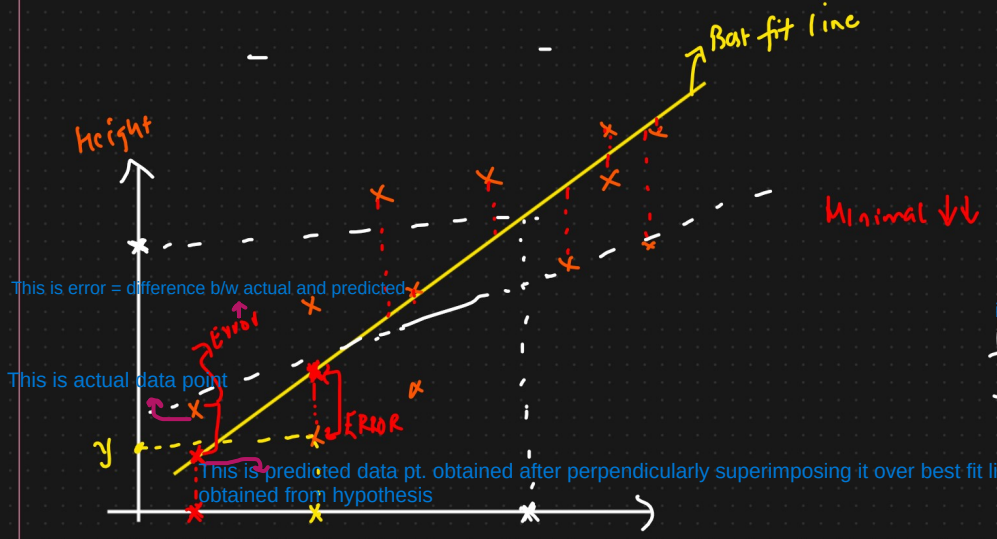
aim is to create a best fit line in such a manner that summation of all errors that is distance b/w actual and predicted data points is minimal.

Independent feature. ↓ Weight	Dependent feature ↓ Height (y) Actual value
74	170
88	180cm
75	175.5cm
—	—
—	—

Independent variable, dependent variable, coefficient of x(theta) all will be 1 in count. Whereas, in multiple regression it will be more than in 1 count



We train our model using hypothesis(best fit line) using dataset (independent and dependent var) and after that use this resultant trained model to make predictions based on new input features.



for 3 dimension(multiple regression) best fit will be a plane and not a best fit line.

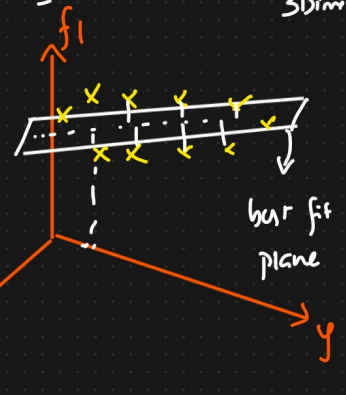
Multiple Linear Regression

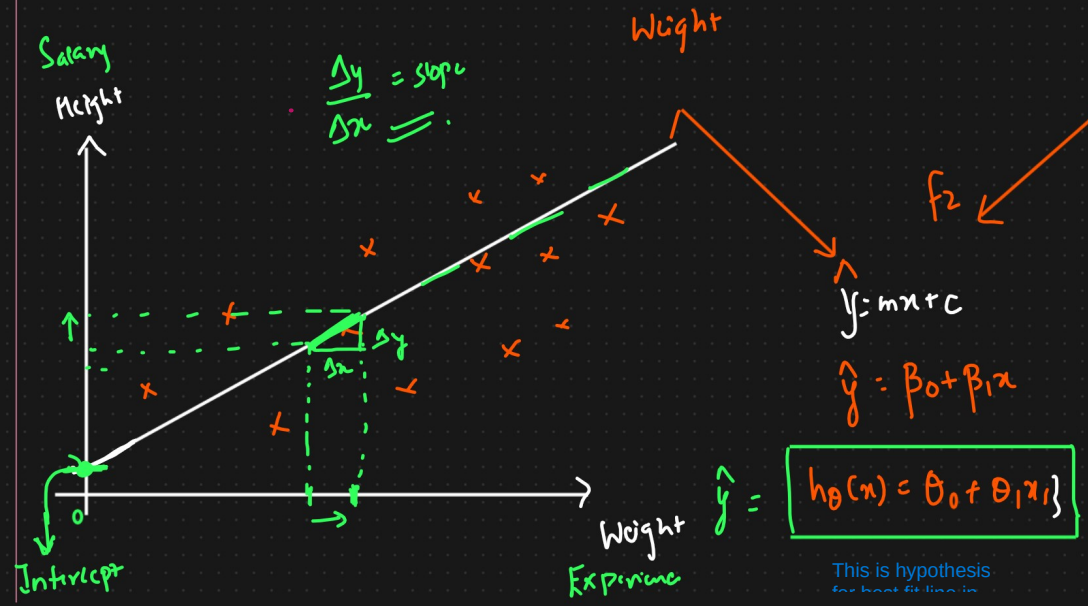
i/p f_1 =

i/p f_2 =

o/p y

3Dimension



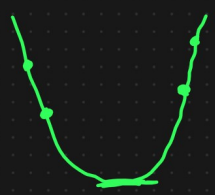


$y = mx + c$

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

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

This is hypothesis for best fit line in

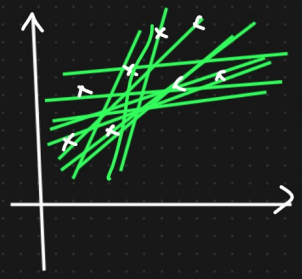


1. y hat or h theta x is prediction
2. Theta 0: Intercept in the point where best fit line is meeting y axis.
3. Theta 1: slope. with unit movement in x axis how much movement in y axis. For straight line slope is same for all data points.

intercept = 0

θ_0 = Intercept

θ_1 = Slope or Coefficient



θ_0 & θ_1

how to obtain best fit: Initialize theta 1 and theta 0 and then optimize it by adding/subtracting so that best fit is generated.

This will be done using gradient descent and convergence algo which is optimization alogo. discussed later in the same notes.

Cost function

calculating error for all datapoints is Cost function

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

[Mean Squared Error]

Cost function of simple linear regression is also called mean squared error (reason self explanatory just observe the formula)

$$J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

y_i ⇓ Actual

$\hat{y}_i$ ⇓ predicted.

n: no. of datapoints

y_i => Actual value

$h_\theta(x)$ = predicted value

Note error is obtained along y axis that's why along the y axis error is calculated.

loss function

$$= \left(y_i - \hat{y}_i\right)^2 \quad \{1 \text{ data point}\}.$$

calculating error for 1 datapoint is loss function

Final Aim

Final aim is to obtain learning parameter theta 0 and theta 1 for which cost function is minimal. This optimization of cost fun^ is done using gradient descent.

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

θ_0,θ_1

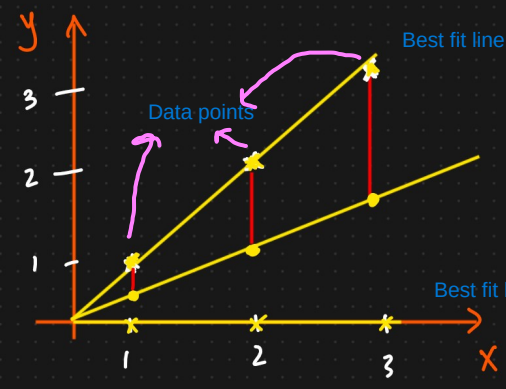
Optimization

{ Minimizing the Cost function }.

DATASET

x	y
1	1
2	2
3	3

Below we will under the optimization of cost function to obtain optimized learning parameters theta 0 and theta 1



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

Consider $\theta_0 = 0$

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

Let $\theta_1 = 1$ (Best fit line 1)

$x_1 = 1$ $\quad h_\theta(x) = 1(1) = 1$

$x_1 = 2$ $\quad h_\theta(x) = 1(2) = 2$

$x_1 = 3$ $\quad h_\theta(x) = 1(3) = 3$

Let $\theta_1 = 0.5$ (Best fit line 2)

$h_\theta(x) = 0.5$

$h_\theta(x) = 1.0$

$h_\theta(x) = 1.5$

Let $\theta_1 = 0$ (Best fit line 3)

$h_\theta(x) = 0$

$h_\theta(x) = 0$

$h_\theta(x) = 0$

Cost function

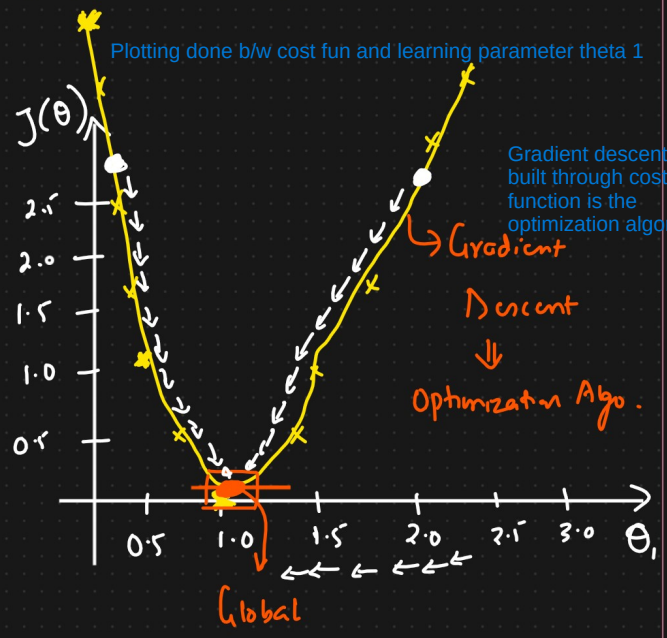
Cost fun for best fit line 1

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

$n = 3$

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

$$= 0$$



Gradient descent built through cost function is the optimization algorithm.

Global minima is the point of convergence (means theta new = theta old) or the point where value of cost function is minimum

Costfn

$\theta_1 = 0.5$

Cost fun for best fit line 2

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

$$J(\theta_1) = 1.16$$

Convergence Algorithm

Repeat until Convergence {

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

}

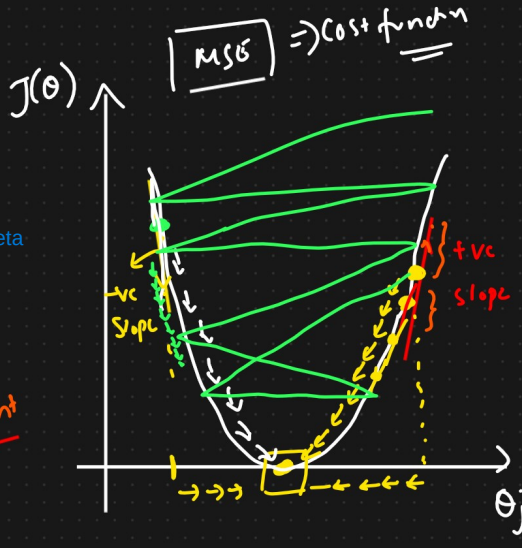
α → Learning Rate

Global Minima

$\Rightarrow$ Slope at a point

point if covergence is where theta new = theta old that is the point where slope is zero.

RMSE $\Rightarrow$ Cost function



When we are converging towards local minima starting from top right tip of inverted hill/bell curve

$\theta_j : \theta_j - \alpha$ (+ve value) ← huge

$\theta_j : \theta_j -$ (+ve value)

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

When we are converging towards local minima starting from top left tip of inverted hill/bell curve

$\theta_j : \theta_j - \alpha$ (-ve value)

$= \theta_j + \alpha$ (+ve value)

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

$\alpha = 0.01$ ⇐

α :) Learning Rate :) 1.00

⇓

Speed of Convergence.

learning rate = alpha is also +ve.
learning rate is speed of convergence or step size.
if learning rate if large then it may overshoot global minima, if too small then will take too much time to converge. So it should be taken accordingly. In most of the ml model it is 0.01

if right end of tangent is facing upward then +ve slope and in case faces downward it's -ve slope.

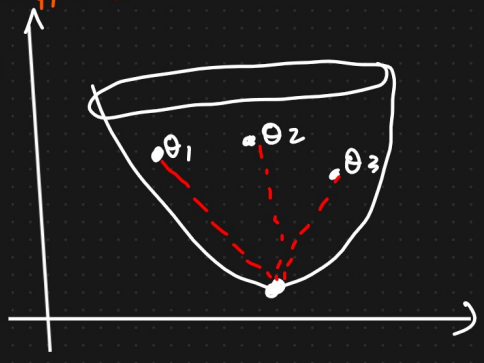
f_1 f_2 f_3 y

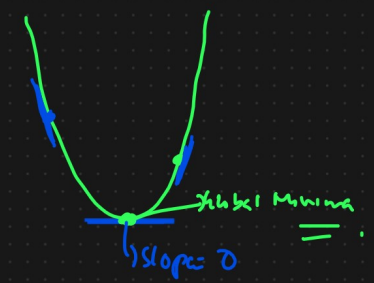
In case of multiple regression we will have more than 1 coefficient of x. So, in such a case 3d inverted hill type plane (not curve since curve is formed in 2 d plane) will be formed.

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

$\theta_1, \theta_2, \theta_3$:) Coefficients

θ_0 :) Intercepts





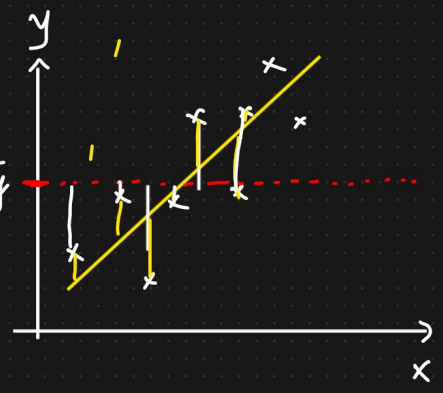
$\theta_{new} = \theta_{old}$

Performance Metrics

Performance metrices are used in order to obtain how good our ml model is.

① R squared ② Adjusted R squared

① R Squared



$$R_{squared} = 1 - \frac{SS_{Res}}{SS_{Total}}$$

{Best fit line} {Average of y}

SS_{Res} = Sum of Square Residuals or Errors

SS_{Total} = Sum of Squared Total

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

simple y is actual value, y cap is predicted value and y bar is average value.

$\Rightarrow$ Small value / Big value $= 0.7$

$\Downarrow$

70% Accuracy

If R squared is 0.7 then that means model is 70% accurate.

In case of regression SS res will be always small as compared to SS total. Therefore R squared will be always less than 1 but greater than 0

R squared = 70%

75%

76%

When we talk about correlation then we are basically checking whether independent variable can put dependency or act as a deciding factor for dependent variable.

Adjusted R Squared

i/p

Size of house | No of Rooms | Location | Gender | → Price (o/p)

Gender - not important feature as it cannot decide the price

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

Adjusted R square < R squared Adjusted R square will always be less than R square.

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

N = no. of data points

p = No. of independent predictors

R2 = R squared

Adjusted R square penalizes additional feature if it is not important.

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

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

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

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

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

⇓

Feature is not important

We can observe above that as we increase independent variables, r square increases (Even after adding a non important feature). Whereas, adjusted r square on the other hand decreases on adding features mainly that are not important. Due to this it becomes important to not just check r square and also check for adjusted r square. Hence, to conclude on adding features R squared will always increase whereas adjusted R square may increase or decrease.

Doubt : How R square for more than 1 independent variable is determined? Formula?