Date __ / __ / __

Module 2

Linear Regression

Simple Linear Regression

- used when we want to predict a target value (dependent variable) using only one input feature (independent variable)

graph is a straight line

$y = mx + b$

y - dependent variable
x - independent variable
m - slope
b - intercept

Predicting the score based on study hours

score - dependent variable
study hours - independent variable

slope - how much y changes for a unit change in x
intercept - value of y when x is 0

eg: Predicting Pizza prices

1. Data collection
2. Calculations
3. Prediction
4. Visualization

Diameter(X) in Inches	Price (Y) in dollars	Mean (X)	Mean (Y)	Deviation (X) X−Mean	Deviation (Y) Y−Mean	Product of deviations	Sum of product of deviations	Square of deviation of X
8	10	30/3 = 10	13	10 − 8 = −2	−3	6	12	4
10	13			10 − 10 = 0	0	0		0
12	16			12 − 10 = 2	3	6		4

$Y = mX + b$

Least square method is used.

tries to minimize the error.

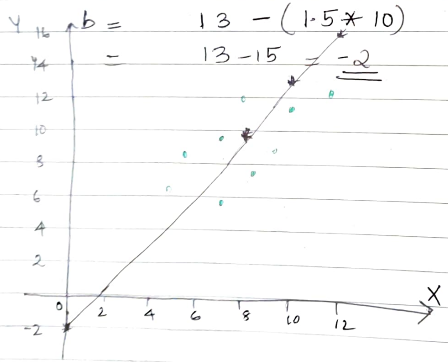
& Calculate $m = \text{sum of product of deviations} / \text{sum of square of deviation of } X$

$b = \text{Mean of } Y - (m * \text{Mean of } X)$

$m = 12/8 = \underline{\underline{1.5}}$

$b = 13 - (1.5 * 10)$

$= 13 - 15 = \underline{\underline{-2}}$



$Y = mX + b$

$Y = 1.5X + -2$

$Y = 1.5X - 2$

if $X = 20$

$1.5 \times 20 - 2$

$= 30 - 2 = \underline{\underline{18}}$

~~Note~~ In Graph,

note the green points

these are actual values

the line shows predicted values

the difference b/n actual pts & predicted pts = error

Slno	(X) Workers	(Y) Output	(X) Mean	(Y) Mean	X Deviation X-Mean	Y Deviation Y-Mean	Product of Devians	Sum of prod of devns	Sq. of deviation of X
1	2	40	30/5 = 6	330/5 = 66	-4	-26	104	280	16
2	4	50			-2	-16	32		4
3	6	65			0	-1	0		0
4	8	80			2	14	28		4
5	10	95			4	29	116		16

$Y = mX + b$

$$m = \frac{\text{sum of prod. of deviations}}{\text{sum of sq. of deviation of } X}$$

$$= \frac{280}{40} = 7$$

$b =$ Mean of Y − (m * Mean of X)

$66 - (7 * 6)$

$= 66 - 42$

$= 24$

$Y = 7X + 24$

7 workers ?

$Y = 7 \times 7 + 24 = 49 + 24$

$= 73$

Multiple Linear Regression

- multiple independent variables
- one dependent variable

$$Y = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + \dots b_n X_n + \varepsilon$$

size of house	Bed rooms	Price (in lakhs)
1	3	2
2	4	5
3	7	9

MLR → $Y = b_0 + b_1 X_1 + b_2 X_2 + \varepsilon$ ↓ error

SLR → $y = mx + b$

b_0 - intercept
b_1 - coefficient of X_1 - slope
b_2 - " " X_2 - "

$$\hat{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}$$

$$= ((X^T X)^{-1} X^T) Y$$

$$X = \begin{array}{ccc} b_0 & bX_1 & bX_2 \\ \end{array}$$

$$X = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 2 & 4 \\ 1 & 3 & 7 \end{bmatrix} \qquad Y = \begin{bmatrix} 2 \\ 5 \\ 9 \end{bmatrix}$$

$$X^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix}$$

$$X^T \cdot X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 3 & 4 & 7 \end{bmatrix} \begin{bmatrix} 1 & 1 & 3 \\ 1 & 2 & 4 \\ 1 & 3 & 7 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 6 & 14 \\ 6 & 14 & 32 \\ 14 & 32 & 74 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 3 & 1 & -1 \\ 1 & 6.5 & -3 \\ -1 & -3 & 1.5 \end{bmatrix}$$

inverse

$$(X^T X)^{-1} \cdot X^T = \begin{bmatrix} 1 & 1 & -1 \\ -1.5 & 2 & -0.5 \\ 0.5 & -1 & 0.5 \end{bmatrix}$$

$$((X^T X^{-1}) X^T) Y = \begin{bmatrix} 1 & 1 & -1 \\ -1.5 & 2 & -0.5 \\ 0.5 & -1 & 0.5 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 9 \end{bmatrix}$$

$$= \begin{bmatrix} -2 \\ 2.5 \\ 0.5 \end{bmatrix} \begin{matrix} b_1 \\ b_2 \\ b_3 \end{matrix}$$

$b_0 = -2$
$b_1 = 2.5$
$b_2 = 0.5$

If size = 4

bedrooms = 10

then what is the price?

$Y = b_0 + b_1 X_1 + b_2 X_2$

$= -2 + 4 \times 2.5 + 0.5 \times 10$

$= -2 + 10 + 5$

$= \underline{\underline{13}}$

Date ___/___/___

Simple Linear Regression

Python code

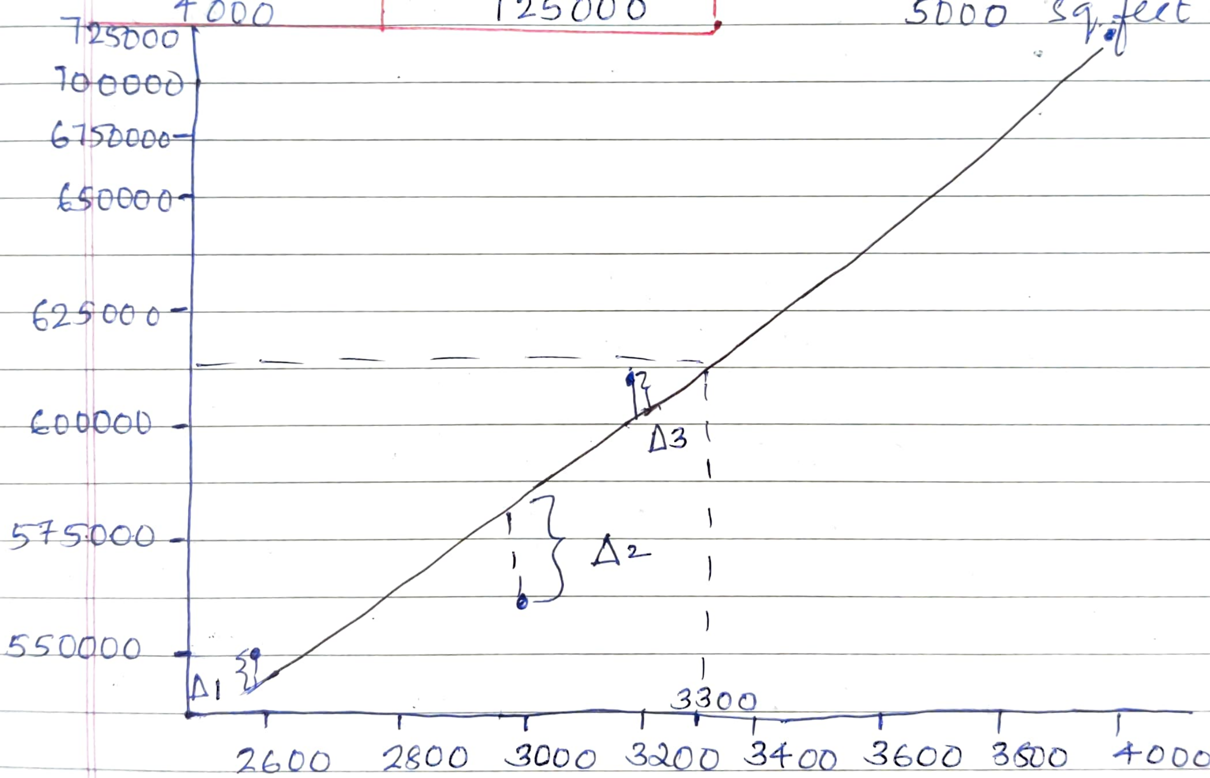
Dataset

Area	Price
2600	550000
3000	565000
3200	610000
4000	725000

Given these home prices find out prices of homes whose area is,

3300 sq feet

5000 sq feet.



we can draw diff lines.

black line is not going through all data pts.

we find out the difference Δ_1 Δ_2 Δ_3

~~an~~ will find out the line in which minimum error values are there.

homeprices.csv
↳ i/p file

Python code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model

df=pd.read_csv("homeprices.csv")
df

plt.xlabel('area')      plt.ylabel('price')
plt.scatter(df.area, df.price)

reg = ~~linear_model~~.LinearRegression()
reg.fit(df[['area']], df.price)
```
↳ double sq brackets needed for Area
fit fn expects the i/p feature(x)

~~reg.predict(3300)~~

reg.predict(np.array([[3300]])) to be in 2D array format.

reg.coef_   // return the coefficient

o/p: 131.73076923

reg.intercept_

190961.538461 5385

// Generate another (set of areas) csv file and find prediction of prices

Date

```
d = pd.read_csv ('area.csv')
d

p = reg.predict (d)
p

d ['price'] = p
d.to_csv ("prediction.csv")

plt.xlabel ('area')
plt.ylabel ('price')
plt.scatter (df.area, df.price, color='red')

plt.plot (df.area, reg.predict (df[['area']]), color=
'blue').
```

Multiple Linear Regression

area	bedrooms	age	price
2600	3	20	550000
3000	4	15	565000
3200		18	610000
3600	3	30	595000
4000	5	8	760000

Find out price of a home that has

3000 sqft area, 3 bedrooms, 40 yr old
2500 " " 4 " 5 " "

Ans

there is a linear rlnship b/n the ip var & target price ⇒ Linear regression
as the area and bedrooms increases price also increases.

price = m_1 * area + m_2 * bedrooms + m_3 * age + b_0

Python code

```
import pandas as pd
import numpy as np
from sklearn import linear_model.

df = pd.read_csv("homeprices.csv")
df
```

since one missing value is there we need to clean the data

Date

Data Preprocessing

```
import math   // inorder to use floor function
median_bedrooms = math.floor(df.bedrooms.
                  median())

median_bedrooms

df.bedrooms = df.bedrooms.fillna(median_bedrooms)
df

reg = linear_model.LinearRegression()
reg.fit(df[['area', 'bedrooms', 'age']],
        df.price)

reg.coef_          // m1, m2, m3

reg.intercept_     // b

reg.predict([[3000, 3, 40]])
   price =
    m1        area     y2 *
137.25 * 3000 +  26025 * 3 + 6825 * 40 +
          383724.9999583

reg.predict([[2500, 4, 5]])
```

Date / /

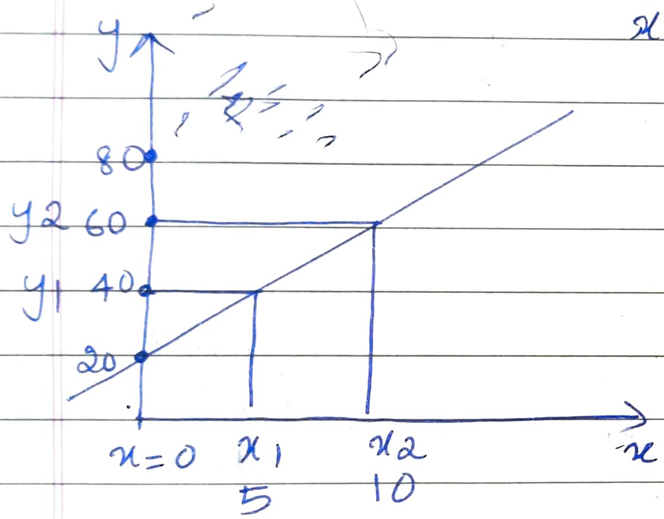
Cost Function

- to find best fit line
- finding appropriate values for 'm' and 'b'
- linear equation

$$\boxed{y = \underbrace{m}_{\text{slope}} * x + \underbrace{b}_{\text{intercept}}}$$

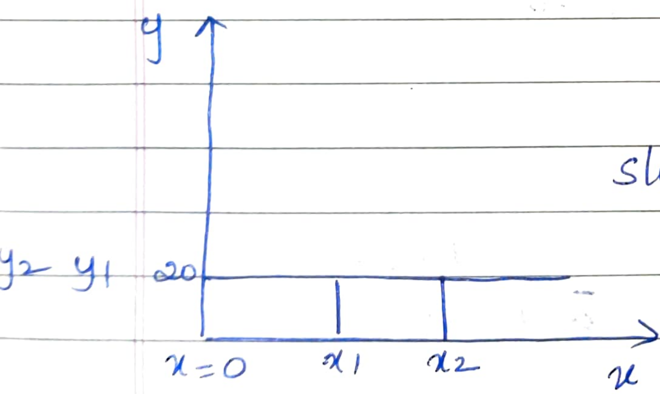
intercept ↳ value of y when $x = 0$.

slope ↳ $m = \dfrac{y_2 - y_1}{x_2 - x_1}$



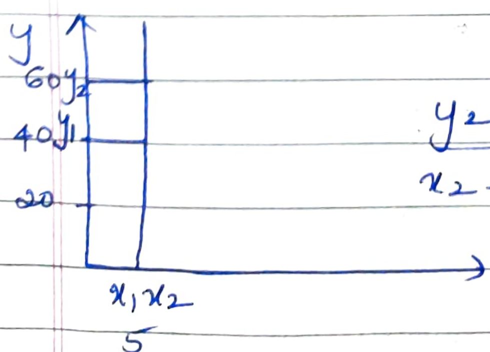
$m = \dfrac{60-40}{10-5} = \dfrac{20}{5} = \underline{4}$

$b = 20$



slope $= 0$

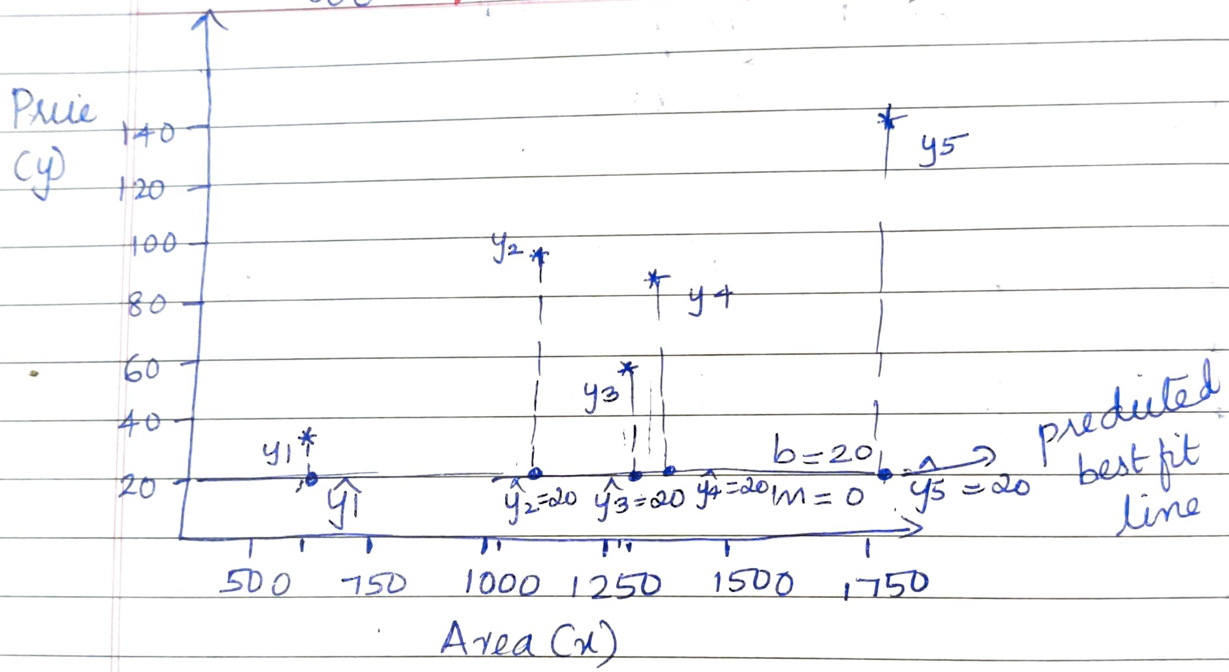
$\dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{20-20}{10-5} = \underline{0}$



$\dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{60-40}{5-5} =$ undefined.

Eg:

Area	Price (in lakh)
656	39
1260	83.2
1057	86.6
1259	59
1800	140



price = m * Area + b

Suppose $m=0$, $b=20$

error = ~~predicted~~ actual y value − ~~actual~~ predicted y value

pred. value of y_1 is $\hat{y}_1 = 20$

$y_1 = 39$

error $= 39 - 20 = \underline{19}$

$y_2 - \hat{y}_2 = 83.2 - 20$

$y_3 - \hat{y}_3 = 59 - 20$

$y_4 - \hat{y}_4 = 86.6 - 20$

$y_5 - \hat{y}_5 = 140 - 20$

} summation of all these becomes error

Date __/__/__

It can be represented as

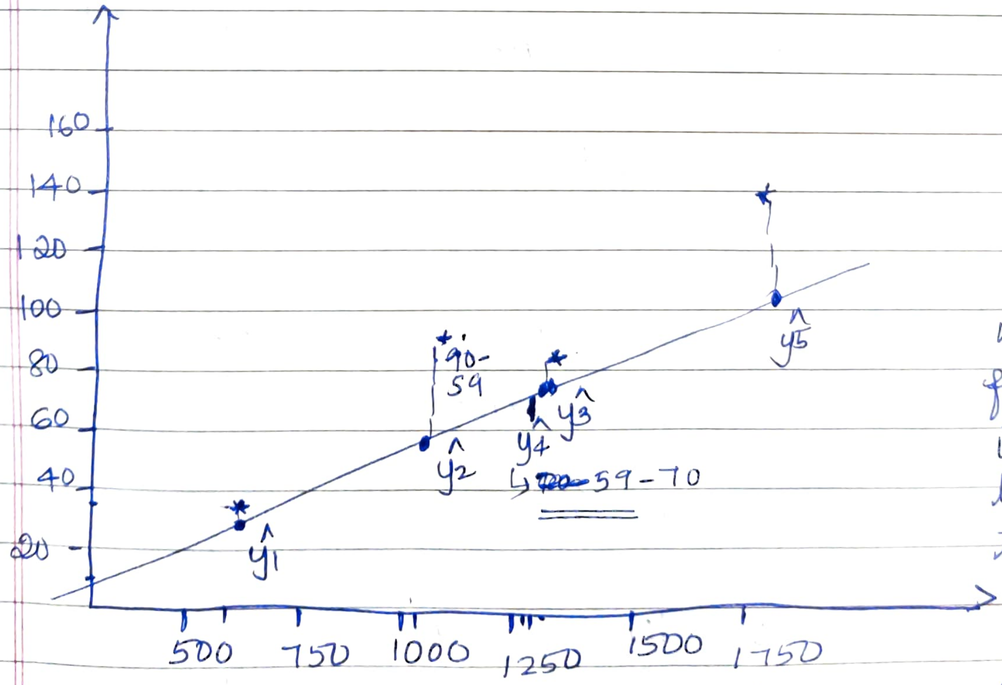
$$\sum_{i=1}^{n} y_i - \hat{y}_i$$

here $n=5$

$$\text{Total error} = \sum_{i=1}^{5} y_i - \hat{y}_i = \text{cost function}$$

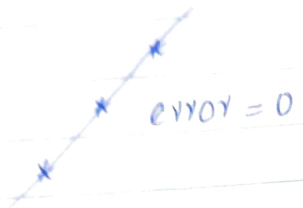
objective is to minimize the cost.
with random value $m=0$ $b=20$
we got certain loss or error
we now update value of m & b in a way that total loss reduces.
we will use trial & error no.

epoch no1 : $m=0$ $b=20$
epoch no2: ~~[illegible]~~ $b=10$. m → we obtained from slope.



now from fig, the error is somewhat less compared to previous trial

If negative values are also present there are chances that error = 0 but error cant be zero



there are chances that error = 0
but error $\neq$ 0

so it is advisable to take absolute value

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

to get absolute difference
$|-5| = 5$
instead of taking sum of all difference
lets take the average difference

ie Mean absolute error (MAE) $= \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$

We will continue iterations or finding epochs
until error is minimum

In Linear regression, we don't use MAE
but use Mean square error

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

If you are taking square of a negative
number, it will become positive
so that problem is solved

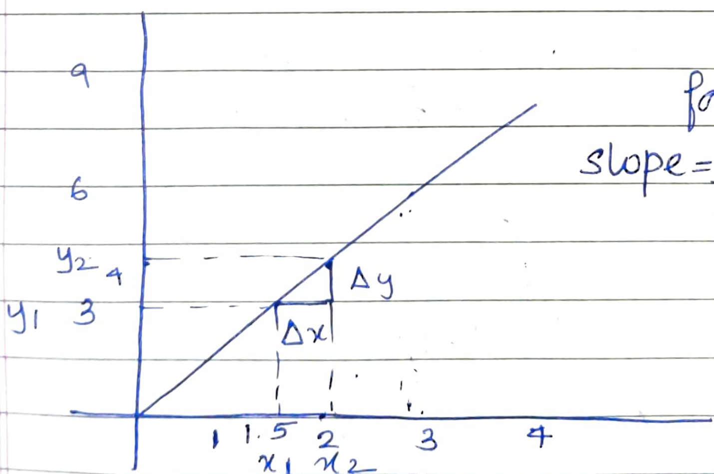
So the cost function used in Linear Regression is Mean Square Error

Cost function

MSE	MAE
It is the average of squared differences between predicted and actual observations. It effectively highlights larger errors.	The average of errors, disregarding their direction. Its the average of absolute differences between prediction and actual observation.

Derivatives & Partial Derivatives

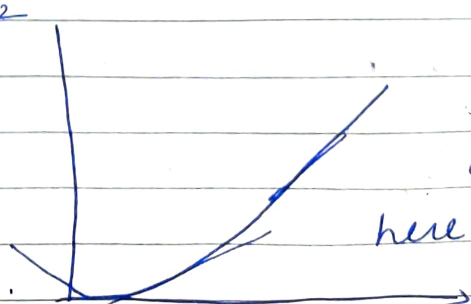
Slope - helps you understand how y (price) changes as per the change in x (area)



for a straight line, slope is constant

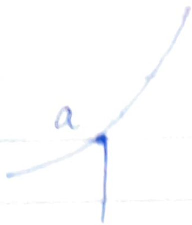
$$\text{slope} = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\frac{4-3}{2-1.5} = \frac{1}{0.5} = 0.5$$



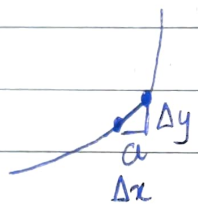
slope can be different at different points

here slope is a function not a constant



we want to measure slope at a specific pt.
say 'a'

if you are checking the point using a magnifier it will be like this



this segment look like a straight line
slop = $\Delta y / \Delta x$.
Δy & Δx are as small as possible.
most probably a zero value.

slope = $\frac{\text{change in } y}{\text{change in } x}$

for a given point

x changes from x to $x + \Delta$
y " " $f(x)$ to $f(x + \Delta x)$

fill in the slope formula

$$\frac{\Delta y}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

for e.g. if function $f(x) = x^2$

then $$\frac{\Delta y}{\Delta x} = \frac{(x + \Delta x)^2 - x^2}{\Delta x}$$

Date ___ / ___ / ___

$(a+b)^2 = a^2 + 2ab + b^2$

$$\frac{x^2 + 2x\Delta x + (\Delta x)^2 - x^2}{\Delta x}$$

$$= \frac{2x\Delta x + (\Delta x)^2}{\Delta x}$$

$$= 2x + \Delta x$$

Then as Δx heads towards 0 we get $2x$

the derivative of x^2 is $2x$.

Thus the slope at x is $2x$.
slope is a function not a constant.

dx can be used instead of Δx tends to 0.

eg: $\frac{d}{dx} x^3 = 3x^2$

$\frac{d}{dx} 7x^3 = 21x^2$

$\frac{d}{dx}(2x^3 + x^5) = 6x^2 + 5x^4$

Slope	Derivative
- used for linear equation	- used for nonlinear eqn.
- It is a constant	- It is a function

Partial Derivative

eg: a watch manufacturing expense depends upon material cost and labor cost.

eg: Let $cost(m, l) = 3m^2 + 2l^2 + 5ml + 23$

Let this be the function

now we want to find out the % of material cost in total expense.

It can be calculated using formula.

$$\underbrace{\frac{\partial cost}{\partial m}} = 6m + 5l$$

partial derivative of cost function with respect to material, labor become constant

$2l^2$ - constant $= 0$

$5 * m * l$.

derivative of m is 1 $= \underline{1}$

$5 * 1 \times l = 5l$

$23 - 0$

derivative of constant is 0.

$$\frac{\partial cost}{\partial l} = 4l + 5m$$

m-constant

eg: $f(x, y) = x^3 + y^2$

$\partial f / \partial x = 3x^2$

$\partial f / \partial y = 2y$

Date

The purpose of partial deerivative is to measure how a function changes as one of its variables is varied while keeping the other variables constant

Chain rule.

$f(x)=x^3$

valid notation for deerivative
$$\frac{d x^3}{dx} = 3x^2$$
$$f'(x) = 3x^2$$

composite function

$f(x)= (3x+1)^2$

$f(x) = g^2$

$g = 3x+1$

for finding deerivative of a composite function, chain rule is used.

Rule :- says that the composite function is divided into individual components & find the derivative

step 1: $f(x)= g^2$ $\quad g=(3x+1)$

$= 2g$

$= 2(3x+1)$

$f'(x) = \cancel{}\ 2 * (3x+1)$

step 2: deerivative of $3x+1$ wrt x

$3x^0 + 0$

3

$f'(x) = 2*(3x+1)*3$

$\cancel{f'(x) = 2*3 = 6}$

Deerivative

x^2	$2x$
$\sqrt{x}$	$\frac{1}{2}x^{-1/2}$
ax	a
x	1
c	0

eg: $\sqrt{4x+5}$

$f(x) = \sqrt{g}$ $\quad g=4x+5$

$f'(x) = \left(\frac{1}{2}\right)x^{-1/2} = \frac{1}{2\sqrt{x}}$

$f'(x) = \frac{1}{2\sqrt{g}}$

Date ___/___/___

$$f'(x) = \frac{1}{\cancel{2}\sqrt{4x+5}} \times \cdot \overset{2}{\cancel{4}}$$

$$= \frac{2}{\sqrt{4x+5}}$$

$4x+5$

$4 \times x^0$

$= 4$

Gradient :- slope of a function in multiple dimensions.

Gradient Descent

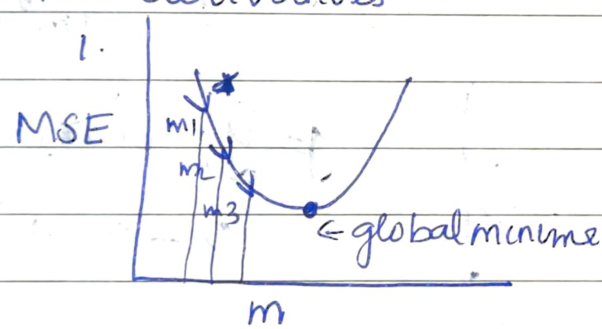
- is a trial and error method used to find optimal values for m and b such that you find the best fit line for your data points.

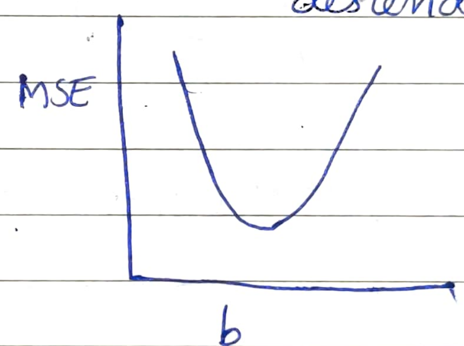
globalminima → lowest possible point of cost function across entire space

aim : how do you find out global minima on the mean square error. surface. (m & b in minimum no. of steps)
we need to find

Inorder to find out the global minima, we are using partial derivatives.

descending the gradient





Keeping b as constant finding partial derivative of MSE wrt m

keeping m as constant finding partial derivative of MSE wrt b

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

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b))^2$$

→ It starts with random initial weights
→ move step by step down the slope
→ at each step, it ~~redude~~ update weights

here we are using gradient, descending on the surface until you find the global minima

The purpose of a partial derivative is to measure how a function changes as one of its variables is varied while keeping the other variables constant

how MSE is changing wrt m & b.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\underbrace{y_i - (mx_i + b)}_{g})^2$$

whole function is a composite function

~~Let $g = mx_i + b$~~ Let $g = (y_i - (mx_i + b))$

$$= \frac{1}{n} \sum_{i=1}^{n} (g)^2$$

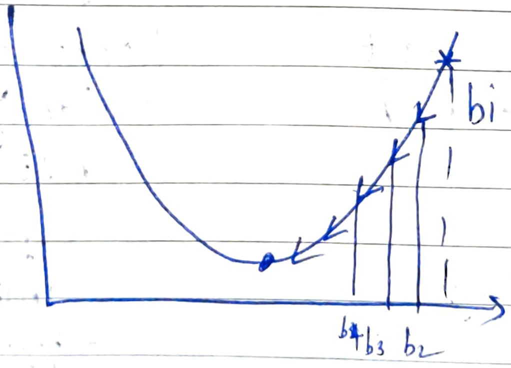
$$\frac{\partial MSE}{\partial m} = \frac{2}{n} \sum_{i=1}^{n} g$$

$$= \frac{2}{n} \sum_{i=1}^{n} (y_i - (mx_i + b)) * -x_i$$

$$\boxed{= \frac{2}{n} \sum_{i=1}^{n} -x_i (y_i - (mx_i + b))}$$

$$g = (y_i - (mx_i + b))$$

$$0 - x_i + 0$$



$$MSE = \frac{1}{n} \sum_{i=1}^{n} \underbrace{(y_i - (mx_i+b))}_{g}^2$$

$$\frac{\partial MSE}{\partial b} = \frac{2}{n} \sum_{i=1}^{n} (y_i-(mx_i+b)) * -1$$

$$\boxed{\frac{\partial MSE}{\partial b} = \frac{2}{n} \sum_{i=1}^{n} -(y_i-(mx_i+b))}$$

$$g=(y_i-(mx_i+b))$$

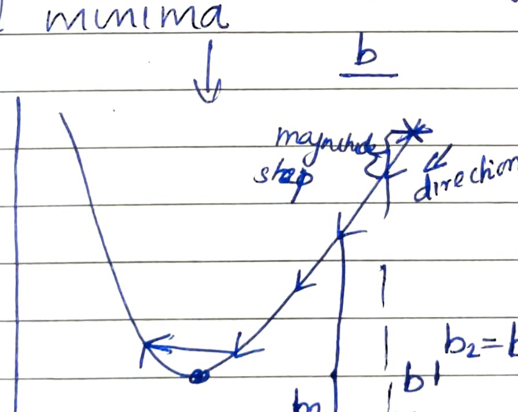
$$\frac{\partial g}{\partial b} = 0-0+1$$

$$= -1$$

$$b_2 = b_1 - \underbrace{\frac{\partial}{\partial b}}_{\text{direction}} * \underbrace{\text{learning rate}}$$

↳ rate ~~at which the~~ the number that tells the computer how big a step to take when its trying to learn something using gd.

If learning rate is bigger you might miss the global minima



initially started from b_1 then find slope in order to get direction $\frac{\partial}{\partial b}$

magnitude shows how much you want to go ahead (no. of steps) = lr

$b_2 = b_1 - \frac{\partial}{\partial b} * (lr)$ ↳ step is determined by lr.

If learning rate is small the no of trials of or epochs will be more, more effort is wasted.

A small learning rate = small baby steps — slow but careful.

A big learning rate = big steps — faster but might miss the bottom or fall off.

So the learning rate helps the model learn at the right speed — not too fast — not too low.

If small steps are taken, spending lot of effort; memory resources.

↳ no of trials will higher

If large steps, then you will miss the point.

so learning right must be optimal

x	y
1	5
2	7
3	9
4	4
5	13

$y = mx + b$