Introduction to machine learning

Agenda

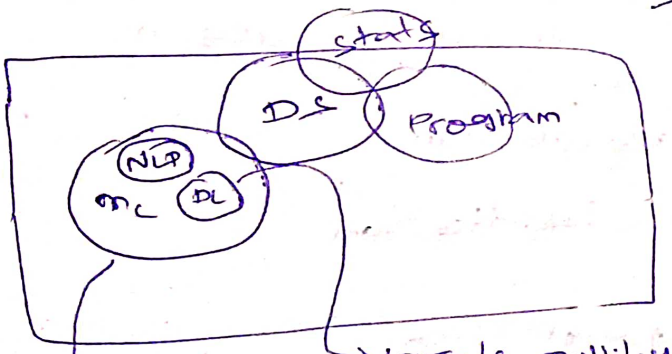
1. machine learning introduction
2. AI VS ML VS DL VS DS
3. Simple linear Regression — mathematical Introduction

AI product

Netflix :- Now a day we all are seeing Netflix when we see some movies it will record some movies is called AI product

streaming platform
↓
Netflix ← Recommendation
↳ AI module

AI VS ML VS DL VS DS



→ Artificial Intelegence

It is creating an application where it performs all its task without any Human interfaces

chatbots → AI chatbot

→ 1950's multilayer Network

AI → subset is ML → subset is DL

ML provides starts tools to analysis, visualizes perform prediction and other task with the help of data

Supervised ML | Unsupervised ML

Supervised ML

Regression

1. linear regr
2. Polynomial
3. SVR
4. DecissionTree R
5. Random Forest R
6. Xgboost R
7. KNN

Classification

1. logistic regression
2. SVM
3. Decission Tree
4. Random Forest
5. KNN

Unsupervised ML

Clustering Algorithm

- DBScan
- kmeans
- kmeans++

↓ cluster (groups)

Suppervised ML:

Independent feature: No.of playing hrs, No.of Study hrs

Dependent feature: Pass/fail

No.of playing hrs	No.of Study hrs	Pass/fail
9	1	0
7	2	0
3	5	1

0 or 1 } classification

Degree	Exp	salary
BTech	1	50K
PhD	2	100K

} Regression

Examples:

Flight Price Prediction → Regression

Algerian Fire Forest → Classification

Air Quality Index → Regression

Tom (Run/NOT) → Classification

Buy Day of the person → classifict

T, S, M, → Day

Unsupervised ML

Age	Salary	Spending score (1-10)
24	70K	1
26	100K	9
21	20K	9
25	120K	2

Customer segmentation

* we divid groups based on spending score to recomand advertisement

* offer 10% or 20%.

↑ Sales increase 20%.

Simple linear Regression :- (contains 1 Independ, 1 Dependent) feature

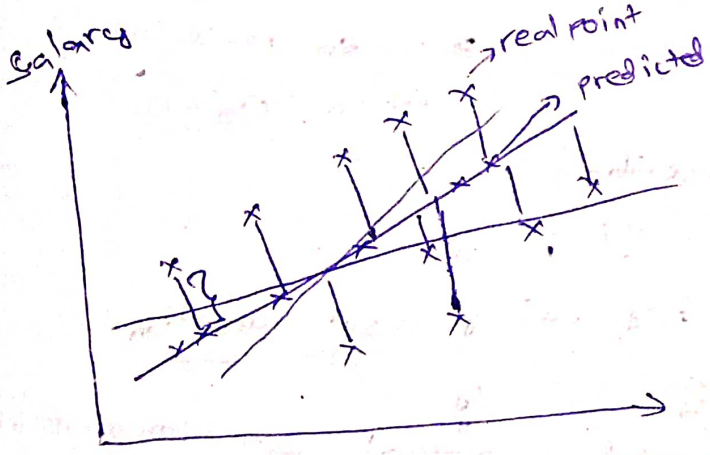
* we find best fit line in two variables ie linear relation ship.

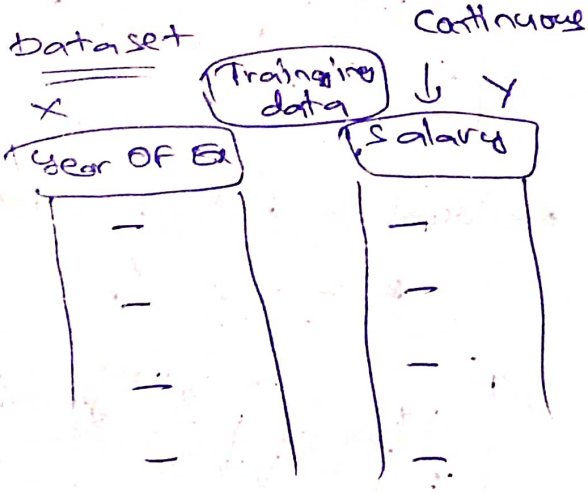
Aim → To create a model

model → Year of Experiance & Salary ⇒ input

Predict → Salary Based on I/p year

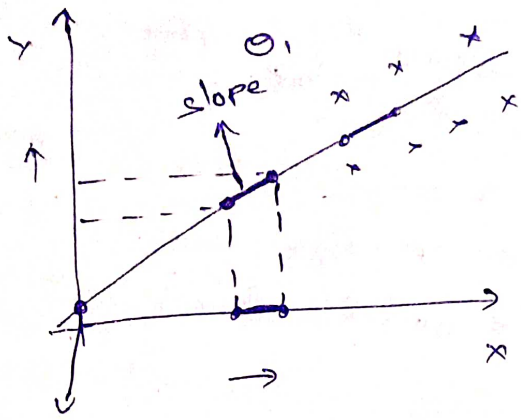
Take 1 Independent & 1 dependent





* minimal main aim find best fit line should be minimal
* difference b/w actual and predicted (Error, Residuals)
* minimal distance model is trained well
* summation should be minimal (less distance)

Equation of straight line :-



Intercept θ_0

* best fit line is equation of straight line

$y = mx + c$

or

$y = \beta_0 + \beta_1 x_1$

or

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

intercept (c) — slope (mx)

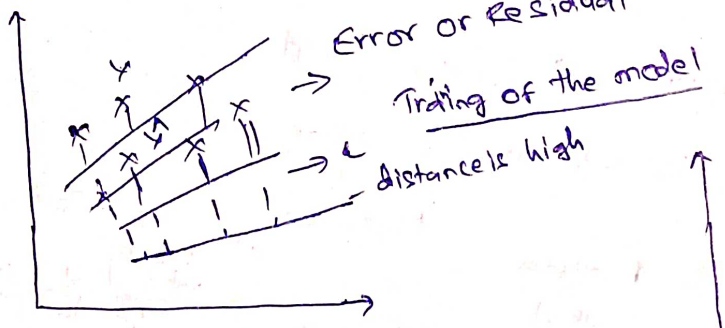
* Intercept and slope change for best fit line

unit movement in x axis what unit movement in y axis is

Example :- freegher

Gap	Salary
0	3.25 lakh

$\theta_0 \theta_1 \rightarrow$ Value

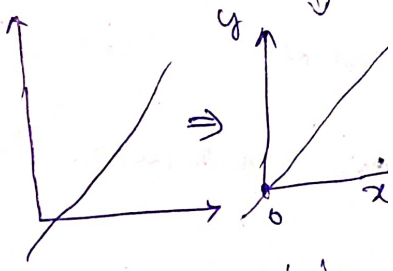


Residual = distance between Y and $\hat{Y}$

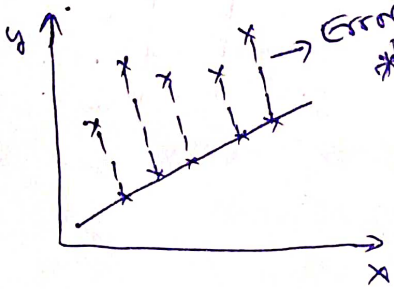
$R = Y - \hat{Y}$

Intercept = 0

$\theta_0 = 0$



* when pass origin then $\theta_0 = 0$



*minimize the Error ??

by using cost function

Cost function :-

$\hat{y}$ ⇓ Predicted — y ⇓ Actual → minimize error ⇓ reduce distance ⇓ decrease

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

m → all data ⇓ ; $y^{(i)}$ ⇓ ; 2 ⇓ ⇒ mean square Error

Called ⇒

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

$\hat{Y}_i$ = predicted value

MSE = mean square Error

n = no. of data points

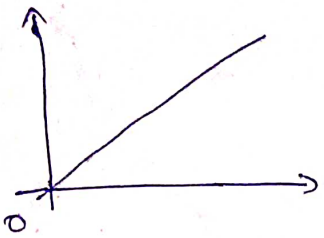
Y_i = Observed values

Final Aim :- (Intercept, slope) (mse)

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

↓ change θ_0 and θ_1 to minimize mse

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



lets Consider

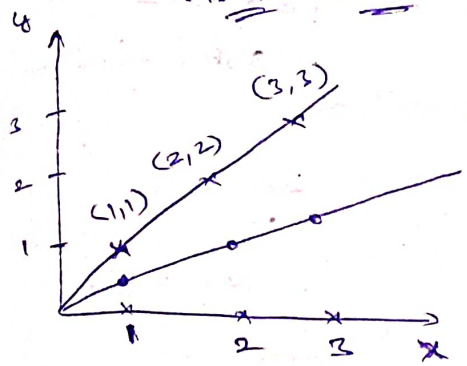
$\theta_0 = 0$ $h_\theta(x) = \theta_0 + \theta_1 x$

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

Train

x	y
1	1
2	2
3	2

Train Graph



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

$\theta_1 = 1$

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

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

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

$\theta_1 = 0.5$

$m = 3$ m = total no. of elements value

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

$h_\theta(x) = 0.1 \quad x = 2$

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

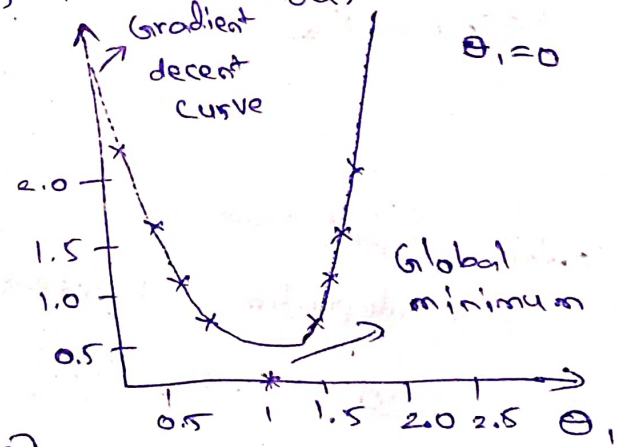
$J(\theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x)^i - y^{(i)})^2$

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

$\Rightarrow \frac{1}{3} [0+0+0]$

$\Rightarrow 0$ it is zero so decrease ↓↓↓

$\theta_1 = 0$



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

$= \frac{1}{3} [0.25 + 1 + 2.25]$

$= \frac{3.5}{3} = 1.16$

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

$J(\theta_1) \simeq 4.66$

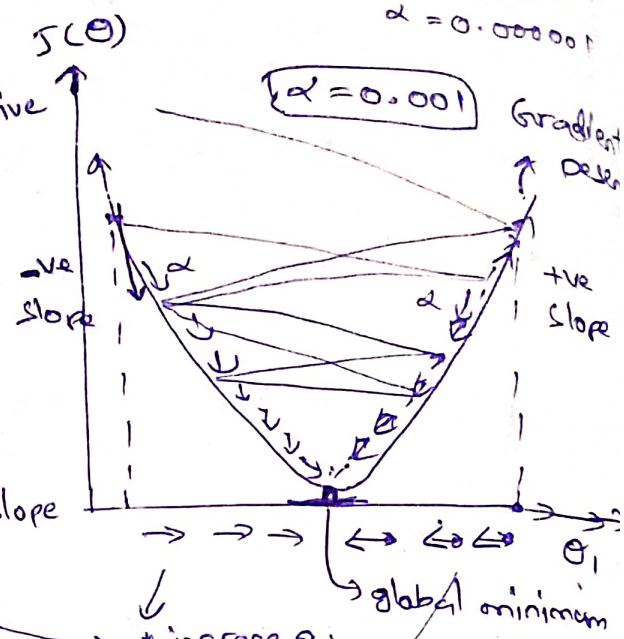
Convergence Algorithm ⇒ Optimize the change of θ_1 value?

Repeat until Convergence

{

$\theta_j = \theta_j - \alpha \left[\frac{\partial}{\partial \theta_j} J(\theta_j) \right]$ — α: learning rate; derivative ⇒ slope

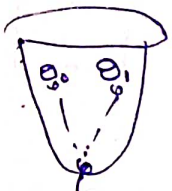
}



θ_0, θ_1

$\theta_j = \theta_j - \alpha(-ve)$ slope

$= \theta_j + \alpha$ → * increase θ_j



global minimum

$\theta_j = \theta_j - \alpha$ (+ve slope) → decrease θ_j

α → learning ⇒ speed of slope

$\alpha = 0.0000001$

$\alpha = 0.001$

MAE

RMSE

mse ⇒ V

↳ is Cost function

x_1	x_2	x_3	y
No. of rooms	City	Room size	Price

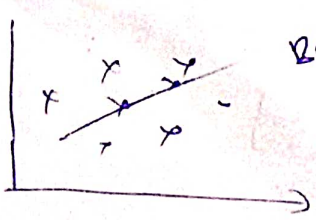
Cohen simple linear

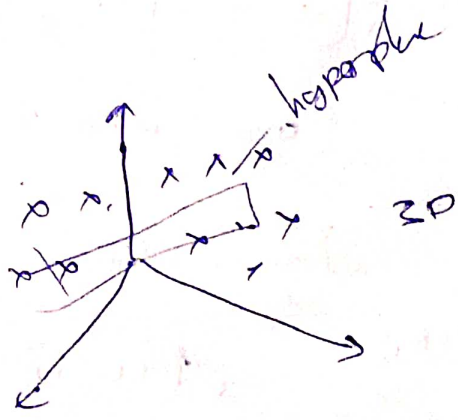
1 independent 1 dependent

$h_\theta(x) = \theta_0 + \theta_1 x$ ⇒ ~~simple class~~

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ ⇒

multiple features





higher plane.

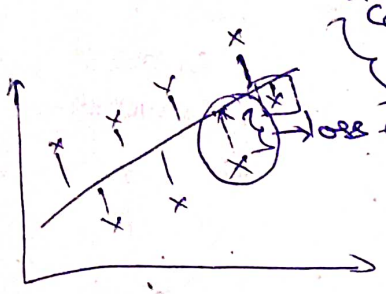
simple linear regression

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

multiple linear regression

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$

Convergence algorithm



m = no. of data points

mse is Cost function :- all points apply every point we calc. difference summation

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

↓ Predicted ↓ Actual (Truth)

loss function :- * single point * each point * every point

$$(h_\theta(x)^{(i)} - y^{(i)})^2 \Rightarrow (\hat{y}_{(i)}, y_{(i)})$$

↓ Predicted ↓ Actual

$j = 1$ to m

** repeat until Convergence

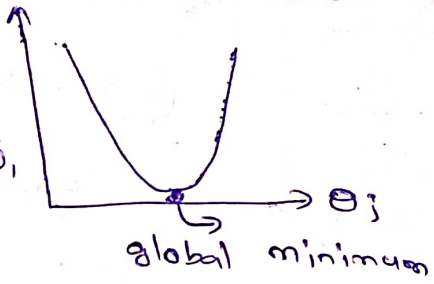
{

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

slope

* update θ_0 & θ_1

}



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

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

$j = 0$:-

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

$$= \frac{2}{m}\sum_{i=1}^{m}\left(h_\theta(x) - y^{(i)}\right)\boxed{\frac{\partial}{\partial \theta_0}(\theta_0 + \theta_1 x_1)}$$

$(1+0)$

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

Example

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

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

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

$\times (1+0)$

$\Rightarrow 2(x+1)$

$j=1$:

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

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

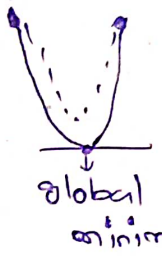
α = learning rate = Speed of Convergence

Repeat until Convergence :

updated θ_0, θ_1

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

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



Cost function & Types

(1) MSE (2) MAE (3) RMSE ⇒ linear regression

(1) MSE {mean Square Error} :

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

$y = mx + c$

$ax^2 + by + c = 0$

$\hat{y} = \theta_0 + \theta_1 x$ ↳ predicted value

$$MSE = \sum_{i=1}^{n}\frac{(y-\hat{y})^2}{n}$$ ↳ (quadratic equation)

θ_0 & θ_1

$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}\left((y - (\theta_0 + \theta_i x)\right)^2$$

Advantages :

(1) this equation is differentiable

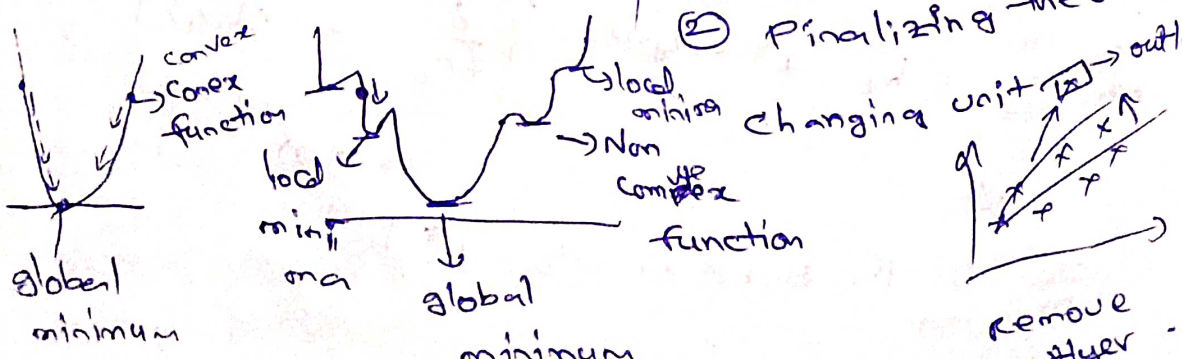
(2) this equation also has one global minima

Disadvantages

(1) this is not robust to outlyers

cost fun ↑↑ Error

(2) Finalizing the Error changing unit



Exp → Independent feature

Salary → dependent feature

(lakh INR) $(y-\hat{y})^2$ (lakh)^2

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

⇒ Cost function ↓↓↓

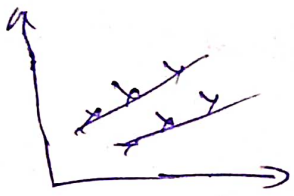
reduce cost function

② Mean Absolute Error (MAE)

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

$|1-5| = |-2|$

$= 2$



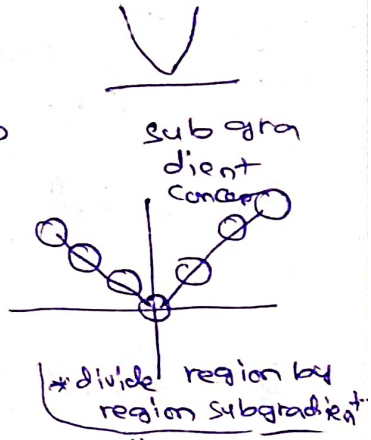
Advantage

① Robust to outliers

② It will also be in the same unit

Disadvantage

* Convergence usually take more time of optimization is a complex task



* Time Consuming → Unit

Outliers

Difference table

⇒ Combination (MSE & MAE)

① Huber loss

② RMSE

↓ is

$\sqrt{MSE}$

① Harber loss :-

* Harber loss you want to use the huber loss any time you feel that you need to balance between giving outliers some weight, but not too much

For case where outliers very important to you use MSE you don't care about outliers.

* RMSE

* RMSE has the benefit of penalizing large errors more so can be more appropriate in some cases.

Huber loss

Performance metrices :-

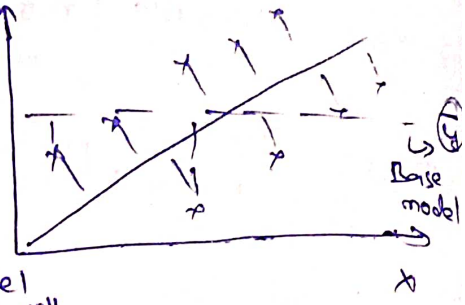
(1) R Squared (2) Adjusted R Squared

(1) R Squared :-

SSR = Sum of Square residuals

SS Total = Sum of Square Average

$$R\,Squared = 1 - \frac{SS_{Res}}{SS_{Total}}$$



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

→ low value { if model is fit well

→ high value

$\bar{y}$ = Average of y

$$R\ Squared = 1 - \left\{ \frac{Small\ number}{Bigger\ number} \right\}$$ Small number

≤ 1

⇒ 0.85 ⇒ 85% Accuracy

⇒ 0.75%

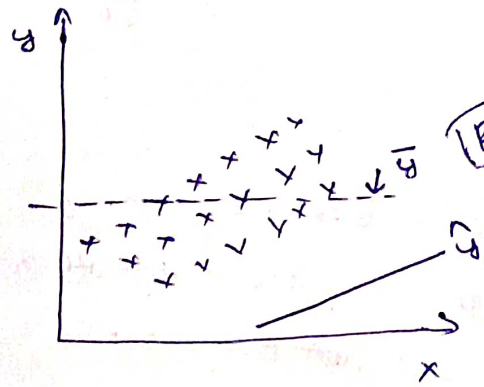
⇒ 75% Accuracy

⇒ R^2 squared is -ve

measured ↓ model is very bad

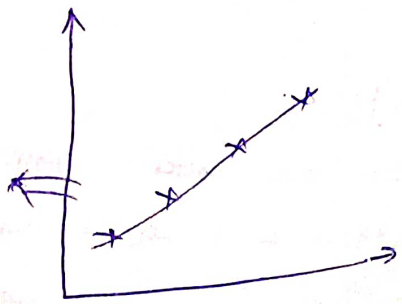
$1 - \{ \frac{Biggg}{Small} \}$ ⇒ badmodel

Performance of the model that you have created



$R^2 = -ve$

$R^2 = 1$



* Adjusted R Squared :-

add feature

* add city location R^2 increase again
* add No. of rooms R^2 increase again

① Size of House ② city location ③ No. of bed rooms ④ Gender ⑤ Price

① R^2 65% Adj R^2 = 63% P=

② R^2 75% Adj R^2 = 73% P=

③ R^2 88% } ⇒ No

④ R^2 90%

Adj R^2 85%

* R^2 is increased in slightely
* adj R^2 is ...

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

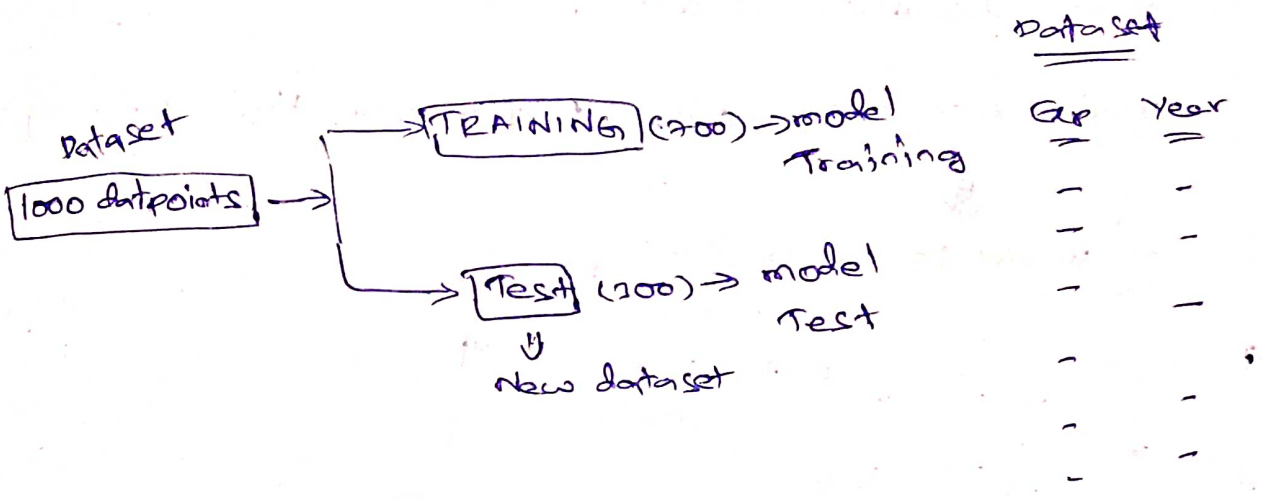
N = No. of data points

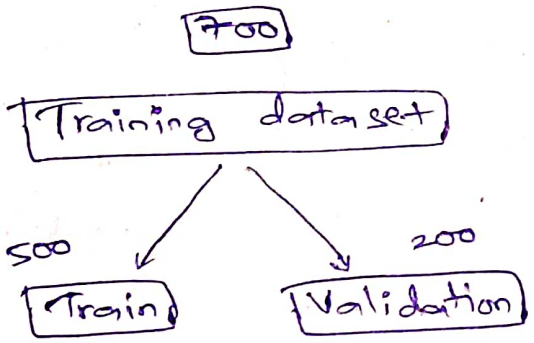
P = No. of Independent feature

* add unnecessary column R^2 increase but adj R^2 is decreased

* (adj R^2 best)

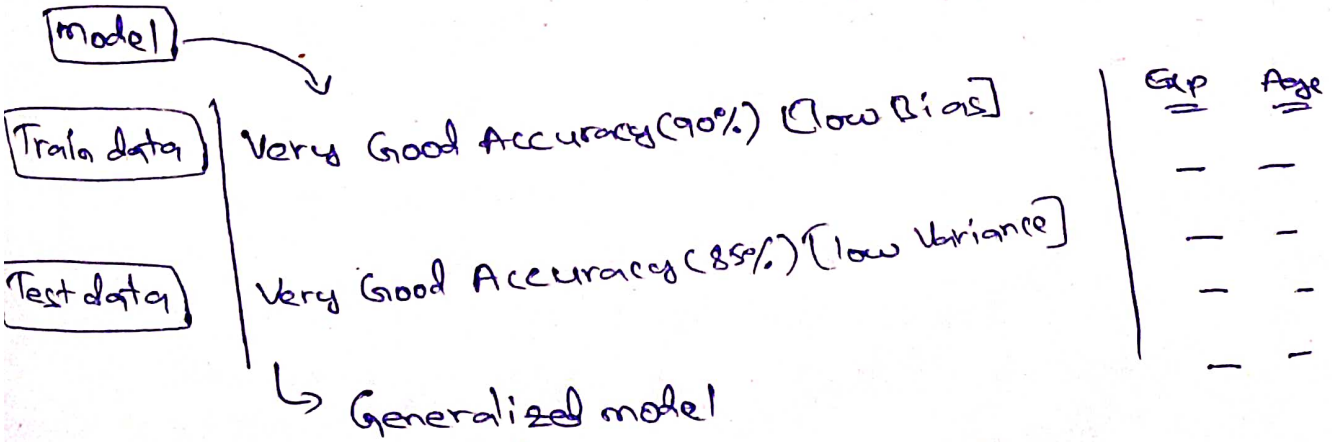
Overfitting And Underfitting (Bias And Variance)





Train the model

Hypothesis Tuning the model



↳ Generalized model

Train Very Good Accuracy [90%] → [low Bias]

Test Bad Accuracy [50%] [High Variance]

⇓

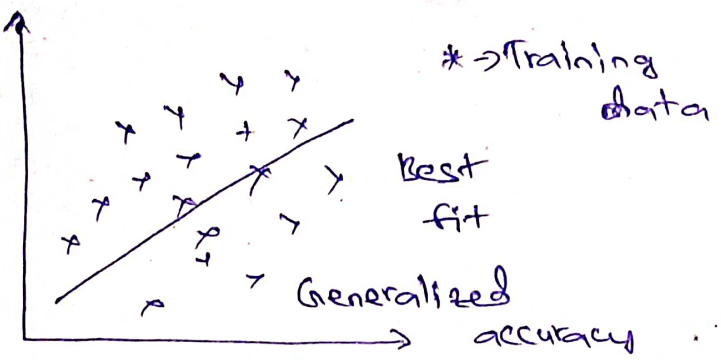
Overfitting

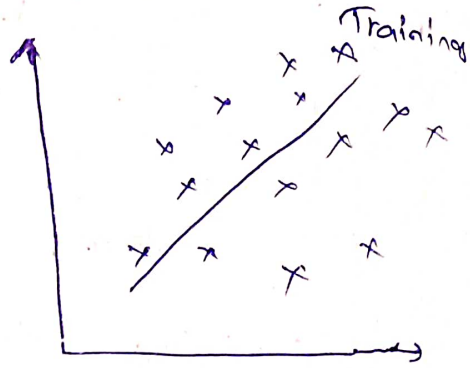
model Accuracy is low [High Bias]

model Accuracy is low [low or High Variance]

⇓

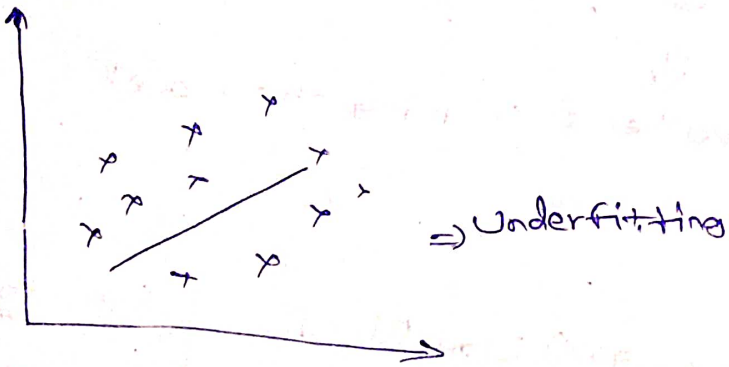
model is underfitting





⇓

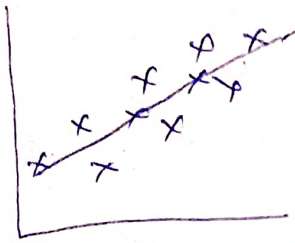
Overfitting



⇒ Underfitting

Ridge Regression , lasso Regression , Elasticnet regression

linear regression :-



$h\theta(x) = \theta_0 + \theta_1 x$ → simple linear

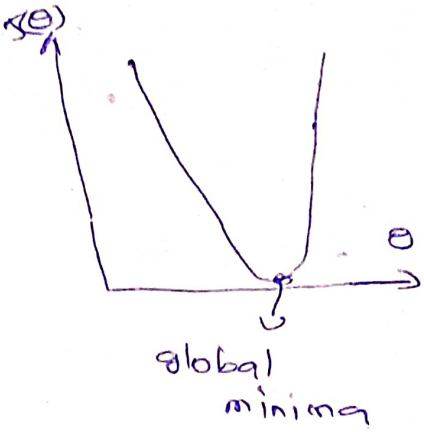
→ multi linear

$h\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$

$$\text{Cost function} = \frac{1}{m} \sum_{i=1}^{m} (h\theta(x)^{(i)} - y^{(i)})^2$$

↓

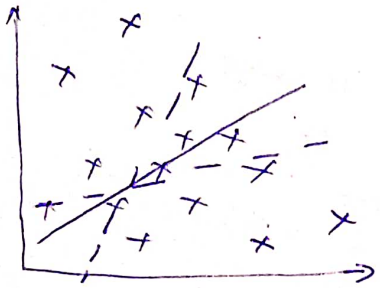
mean square error



① Ridge Regression :- (L_2 regularization)

(L_2 Norm)

* used to reduce Overfitting * Cost function = 0



Overfitting

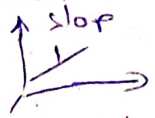
(low bias)

Training Data → Accuracy ↑↑ = 95%

Test data → Accura ↓↓ ⇒ 60%

(High variance)

hyper Parameter



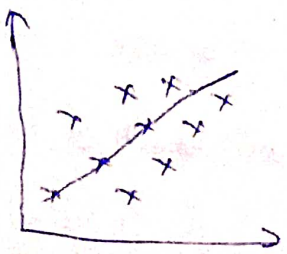
$$\text{Cost fun}^{n} \frac{1}{m} \sum_{i=1}^{m} (h\theta(x)^{i} - y_{\theta}^{i})^2 + \boxed{\lambda \sum_{i=1}^{n} (\text{slope})^2}$$

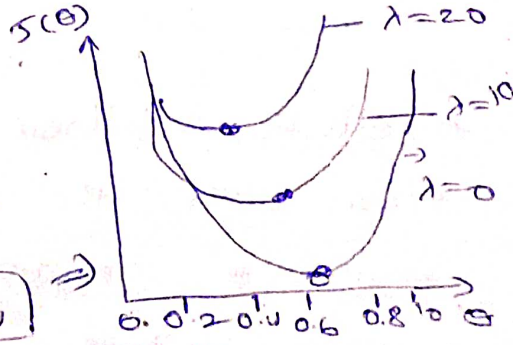
↳ $\{\theta_1^2 + \theta_2^2 + \theta_3^2 + \dots \theta_n^2\}$

$h\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

$= \theta_0 + 0.95 x_1 + 0.82 x_2 + 1.5 x_3$

$= \theta_0 + 0.85 x_1 + 0.70 x_2 + 0.77 x_3$





$\lambda \uparrow \quad \theta \downarrow$

Cost fn = 0 + λ(slope)²

= +ve ↓↓

② Lasso Regression :- [L_1 Norm]

L_1 regularization

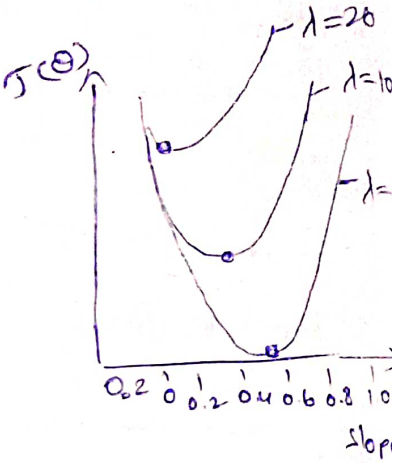
* used to reduce the features
* feature selection

Cost fn : $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda \sum_{i=1}^{m} |slope|$

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_3$

$= \theta_0 + \theta_1\, 0.54 x_1 + 0.27 x_2 + 0.10 x_2$

removed feature



③ Elastic [L_1 and L_2 Norm] :-

Ridge

Cost fn = $\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda_1 \sum_{i=1}^{m} (slope)^2 + \lambda_2 \sum_{i=1}^{m} |slope|$

MAE

RMSE

lasso

* Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models.
* both methods by learning from their shortcomings to improve the regularization of statistical models