Introduction To Machine Learning

Agenda

1. Machine Learing Introduction
2. AI VS ML VS DL VS DS
3. Simple Linear Regression → Mathmatical Intution

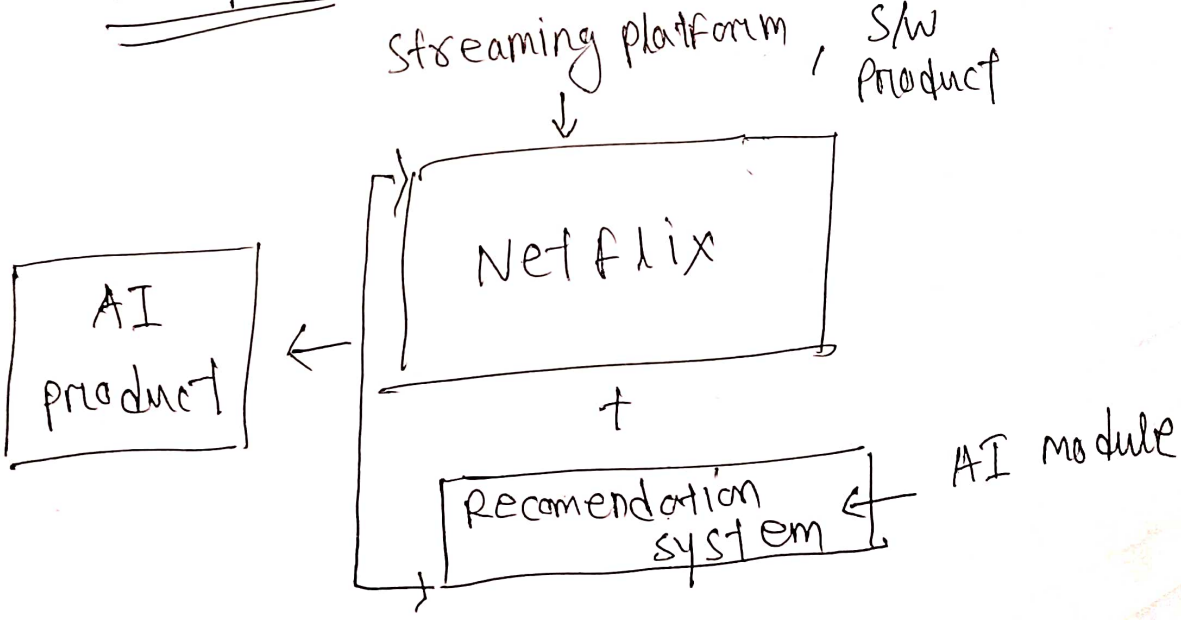
AI Product

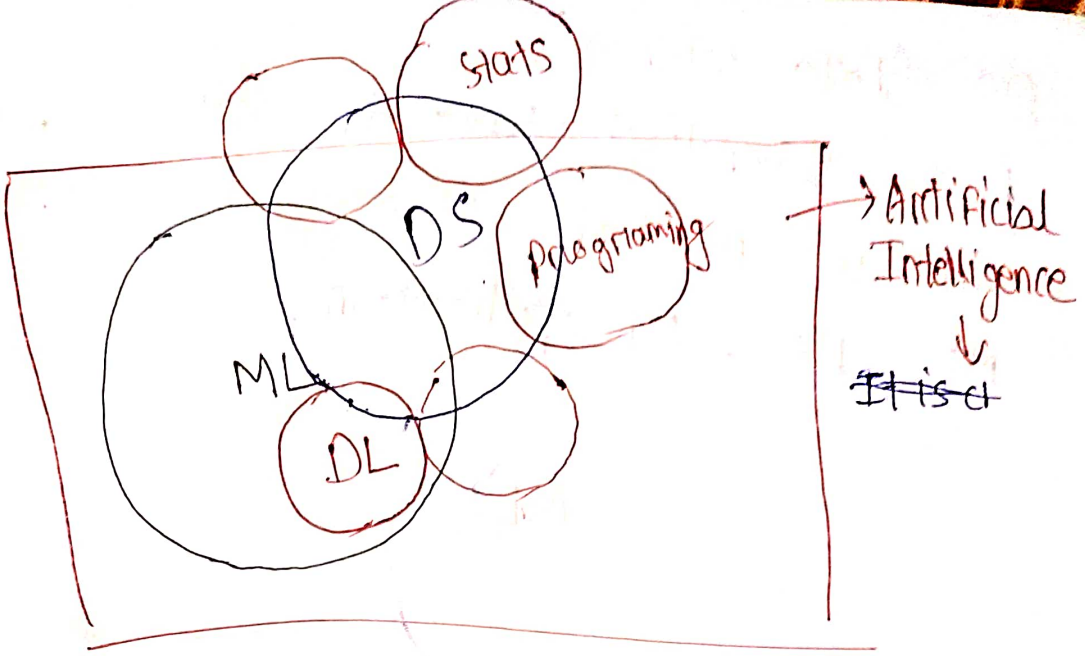
ML engineer
AI engineer
Vison Developer
NLP engineer

End → Goal

AI application

example





AI (Artificial Intelligence)

It is creating an application where it performs all its task without any human intervention.

[Human try to make this product better and better]

examples

AI chatbots, Alexa, self-driving car, Youtube

ML (Machine Learning)

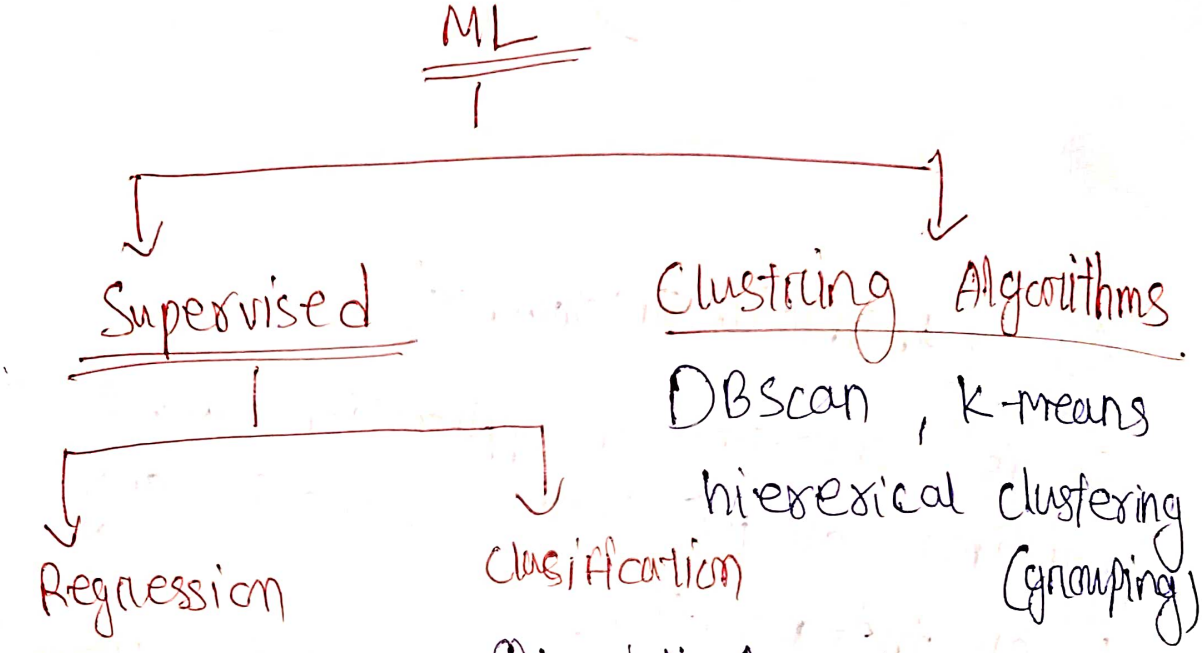
ML provides stats tools to analyze, visualize perform predictions and other task with the help of Data.

DL (Deep Learning)

Around 1950's Researcher's thought's that we make machine can learn, how we are able to learn.

DS (Data science)

Data scientist has to deliver, what is needed. it may be related to ML, DL, AI application.



(1) Linear regression
(2) Polynomial "
(3) SVR
(4) Decision tree regressor
(5) Random forest Regressor
(6) Xgboost
(7) Naive Bays

(1) Logistic Regression
(2) SVM
(3) Decision Tree Classifier
(4) Random forest Classfier

Supervised

O/P will be given

↓

make classification, Regression

Unsupervised

O/P not given

↓

Cluster (groups)

Supervised

Degree	exp	(O/P) Salary
B.E	7	50K
PHD	2	70K
-	-	-
-	-	-
-	-	-
-	-	-

Continuous feature

→ Regression problem

↳ Dependent feature.

Input or Independent features

Number of study (hours)	No. of Play hours
9	1
7	3
2	5
1	6

O/P

Person pass/fail
pass (1)
1
0 (fail)
0

Classification problem

→ Flight price prediction (Regression)

→ Tomorrow going to Rain or not (classification)

→ Air quality index (prediction or Regression)

Simple Linear Regression

one independent → one dependent feature

examples

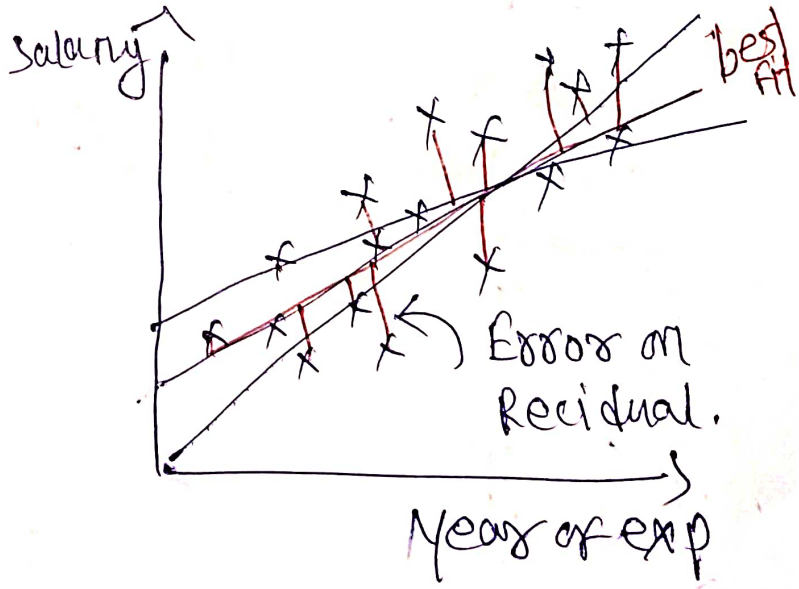
No. of rooms → price

Model ← Years of experience & Salary

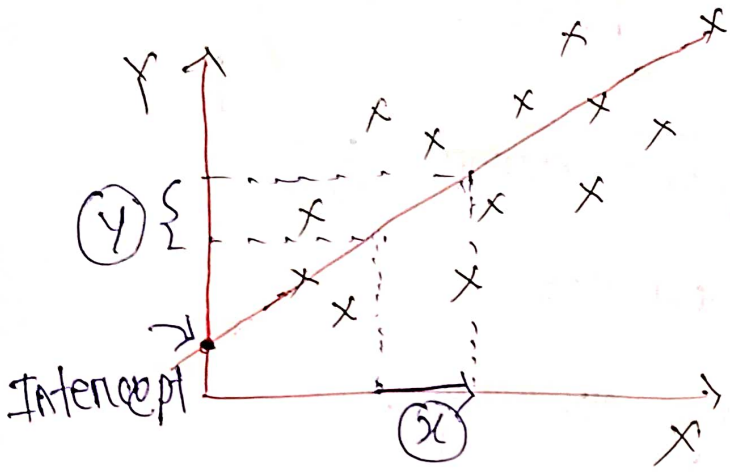
↓

Predict → Salary Based on I/p year.

Years of Exp	salary
—	30K
—	30K
—	100K
—	—
—	—
—	—
—	—
—	—
—	—



In Simple linear regression we try to find the best fit line such that error or distance between all points and prediction (actual) is minimum.



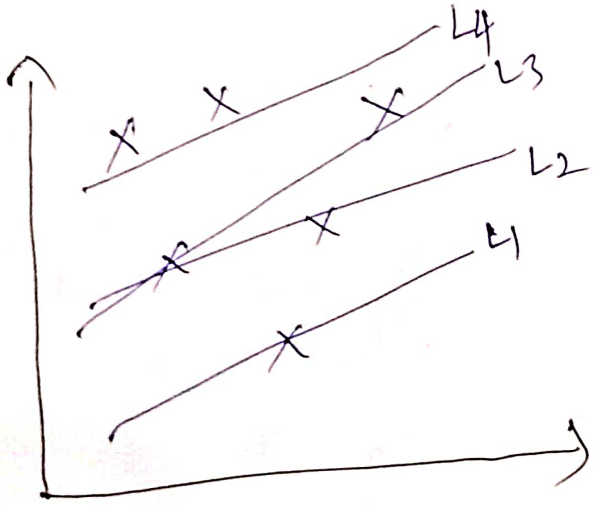
Equation of straight line

$Y = mx + c$

$y = B_0 + B_1 x_1$ — slope (unit movement in Y axis, when unit movement in x axis)

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

θ_0 → Intercept ($Y = 2$ when $x = 0$)



→ Training of the model.

→ To find the bestfit line we change in θ_0 and θ_1 and predict the best fit Line.

→ **Cost function**

Predict output, Actual o/p

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

mean square error ↓↓↓

$J(\theta_0, \theta_1)$ → cost function

$\frac{1}{m}$ → for finding average.

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

Let $\theta_0 = 0$

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

when

$\theta_1 = 1$

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

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

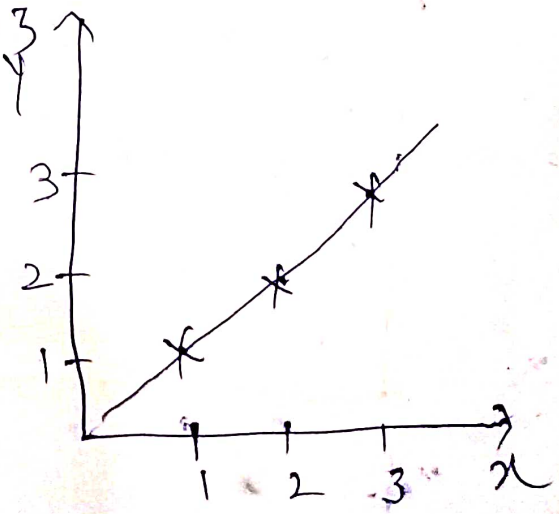
$h_\theta(3) = 3 \quad x = 3$

Let

Training Data

X	Y
1	1
2	2
3	3

→ Both actual and predicted points are equal.



error

cost

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

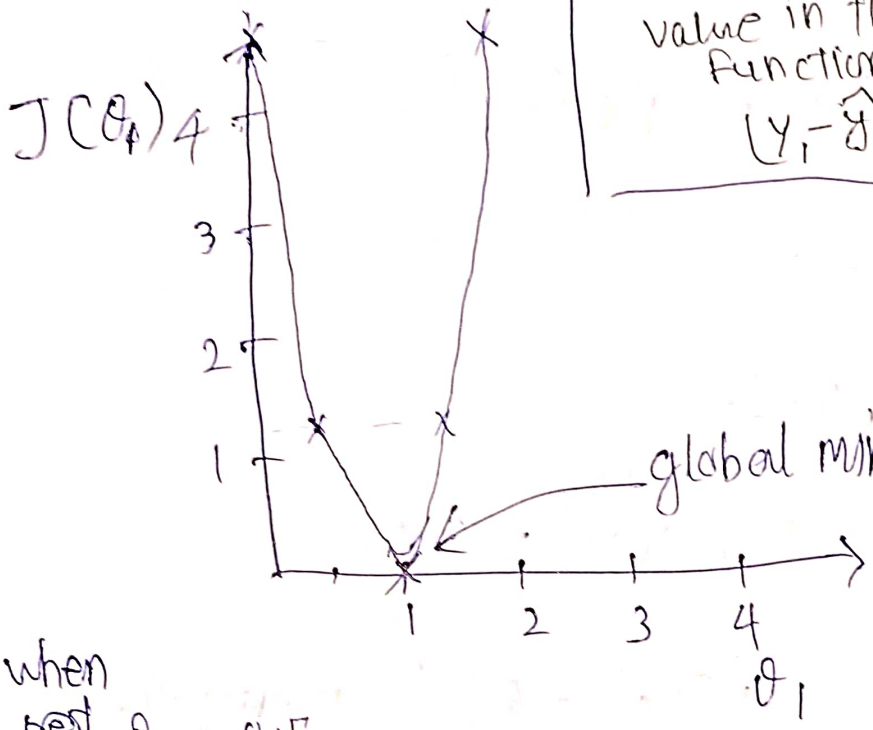
$$= \frac{1}{3}\left((1-1)^2 + (2-2)^2 + (3-3)^2\right)$$

$$= 0$$

Note

Loss is a single value in the cost function

$$(Y_i - \hat{Y}_i)^2$$



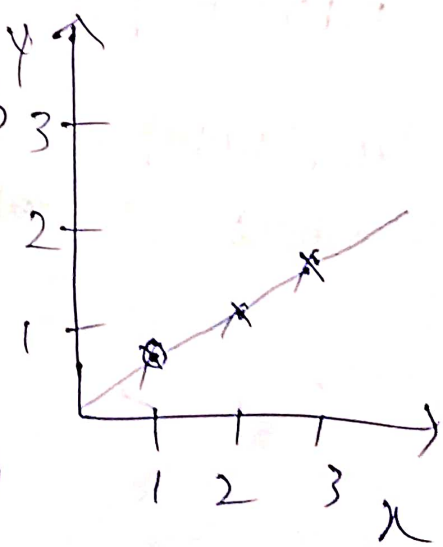
when

$\theta_1 = 0.5$

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

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

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



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

$$= \frac{1}{3}(0.25 + 1 + 2.25)$$

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

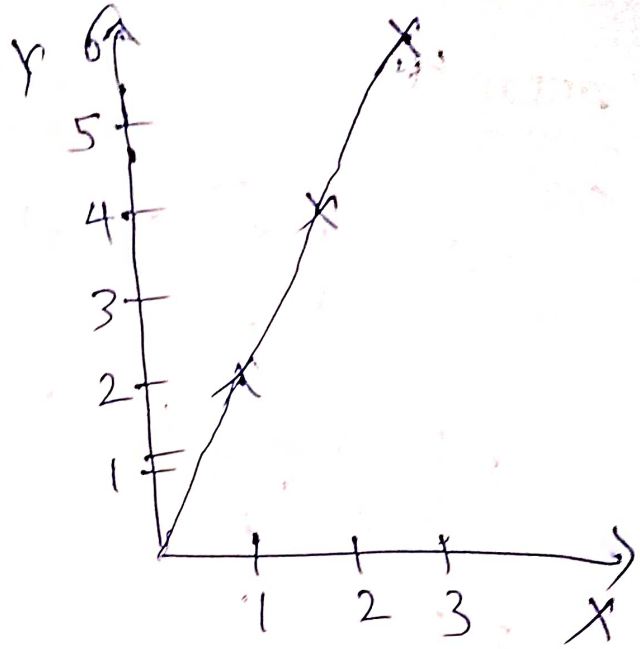
when

$\theta_1 = 2$

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

$h_\theta(x) = 4 \quad , x = 2$

$h_\theta(x) = 6 \quad , x = 3$



$J(\theta_1)$

$= \frac{1}{3}\left((2-1)^2 + (4-2)^2 + (6-3)^2\right)$

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

$= \frac{14}{3} \approx 4.66$

similarly.

when $\theta_1 = 0$, $J(\theta_1) \approx 4.66$

At $\theta_1 = 1$ $J(\theta_1) = 0$ ↓↓ minimum.

Like that we have to reach the minimum point called <u>global minima</u>.

→ The technique for reaching global minima is called convergence algorithm.

convergence Algorithm (optimize the changes of θ_1 value)

Algorithm

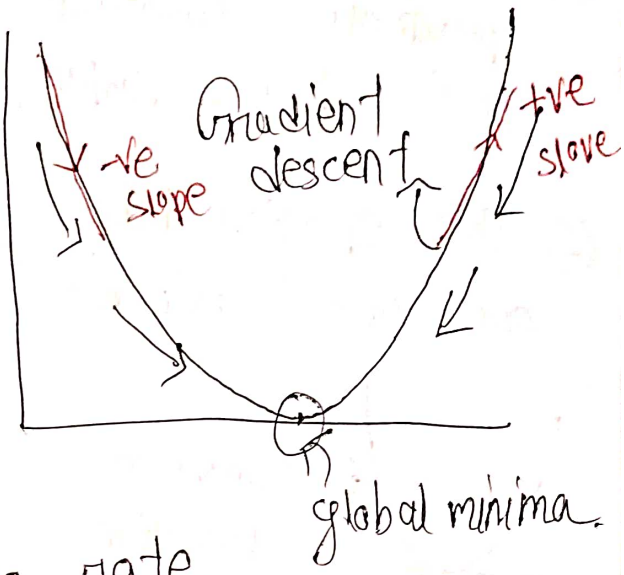
Repeat until convergence

{

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

}

α = Learning rate



α / Learning rate decides the speed to reach the global minima. It should not be to high it moves in zig-zag speed and not able to reach global minima. It should not be too low also. generally $\alpha = 0.001$

Cost functions

① MSE ② MAE ③ RSME

① MSE (Mean square error)

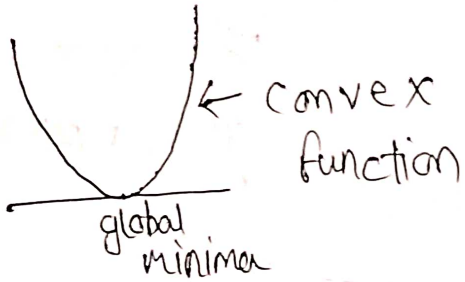
$$J(\theta) = \frac{1}{m} \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

or

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

Advantages of MSE

1. As MSE is a quadratic equation if is differentiable.
2. This equation has one global minima.



Disadvantage of MSE

1. This is not robust to outliers.
2. Due to quadratic equation unit of output feature changes on penalizing error

MAE (Mean absolute Error)

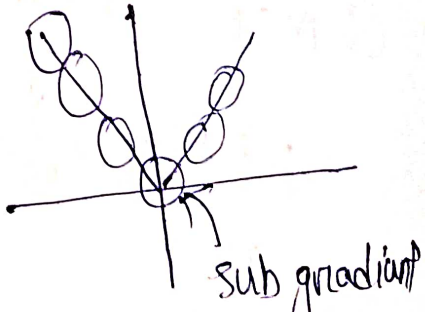
$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y^{(i)} - \hat{y}^{(i)}|$$

Advantages

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

Disadvantage

1. convergence usually takes more time. optimization is a complex task



because of sharp zero value.

RSME (Root Mean square Error)

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

Advantages

① Unit will be same due to square root.

② It is differentiable

Disadvantages

① This is not robust to outliers.

Performance Matrix

① R Squared ② Adjusted R squared.

① R squared

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

SS_{Res} = Sum of square Residual.

SS_{Total} = Sum of square Average.

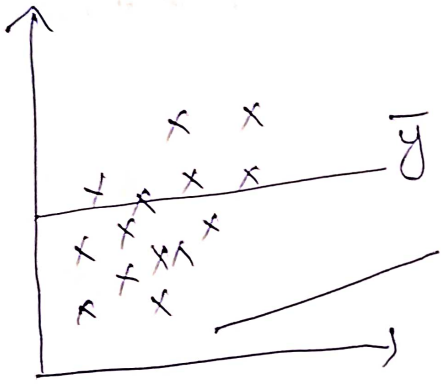
$$R\ squared = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$\bar{y}$ = Average of Y

$R\text{squared} = 1 - \left(\frac{\text{small num}}{\text{large num}}\right)$ small numbers

→ R square measures performance of the model.

Q. Can R^2 squared value -ve?



$R^2 = -ve$

model is very waste.

$R^2 = 1 - \left(\frac{\text{large num.}}{\text{small num}}\right)$ - large (>1)

$= (-ve)$ ✓

② Adjested R squared

I/P

size of house	city location	No. of bedrooms	Gender

O/P

Price

R^2 65% 75% 88% 90%

Adjusted R^2 65% 75% 88% 85%

By adding a valuable columns accuracy (R^2) generally increases. with adding unnecessary column it increases slightly (R^2)

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

But in case of adjusted R^2 it decreases slightly.

Overfitting and Underfitting

(Bias and Variance)

Dataset

1000 data points

→ Training (700) → Model training

→ Test (300) → model Evalution

Training Dataset

→ Train 500

→ validation 200 ← used for hyper parameter tuning.

Traing Data → Very Good Acc. (90%) (Low Bias)

Testing Data → Very Good Acc. (85%) [Low variance]

↓

Generalize model.

Training — Very Good Acc. [90%] [Low Bias]

Testing — Very Bad Acc. [50%] [High variance]

↓

overfitting model.

↓

Hyper parameter tuning / increase Data

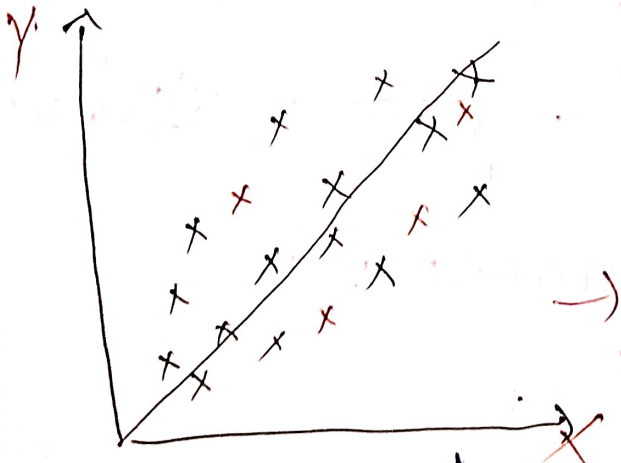
Training — Accuracy Low [High Bias]

Testing — Accuracy Low/High [Low/High Variance]

↓

Underfitting Model.

[Model not able to learn the pattern]

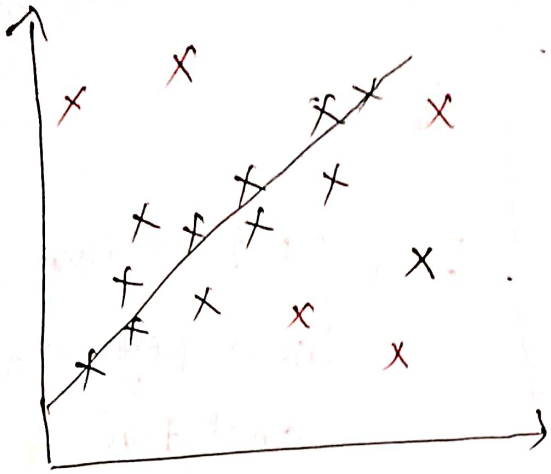


x → training points

x → Testing points

→ Generalized model.

↳ Best Fit model.



X → Training points

X → Testing points

→ Overfitting model.

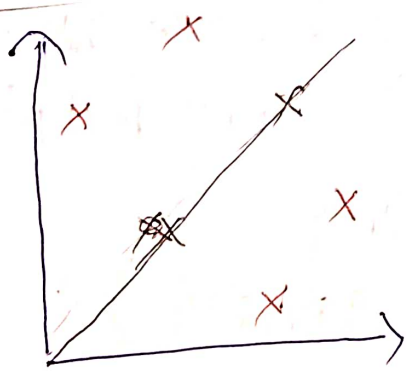
X → Traing points

X → Testing points.

→ Under fitting model

Ridge Regression (L_2 Regularization)

L_2 Norm



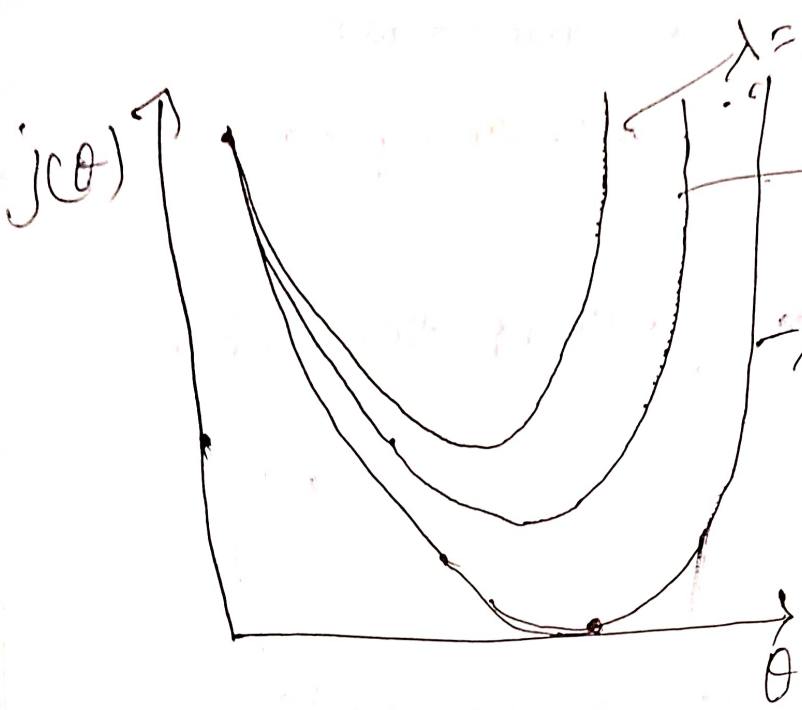
overfitting

Training accuracy = 95%

Test accuracy = 50%

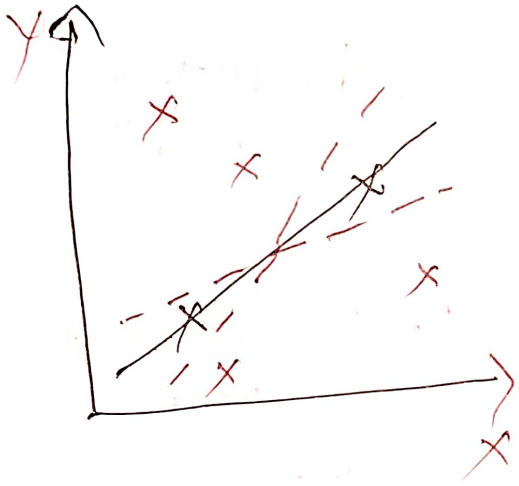
$$\text{Cost fn (Linear Reg.)} = \frac{1}{m}\sum_{i=1}^{n}\left(h_\theta(x)^i - y^i\right)^2 + \lambda\left[\theta_1^2 + \theta_2^2 + \dots\right] \quad \left(\lambda \sum_{i=1}^{n} \text{slope}\right)^2$$

λ = hyperparameter



(Behave like linear Regression cost fun^n)

$$\lambda \propto \frac{1}{\theta}$$



cost fun^n = 0 + λ(slope)^2

= +Ve value

to decrease this we need to modify best fit line.

→ Reduce overfitting

(Never overfit)

Note

θ ≠ 0

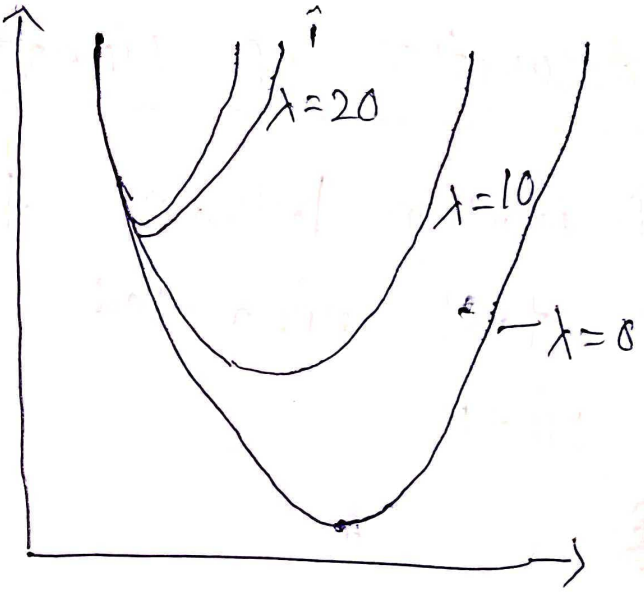
Lasso Regression (L_1 Regularization / L_1 Norm)

↓

Used for feature selection.

$$\text{cost fun} = \frac{1}{m}\sum_{i=1}^{n}\left(h_\theta(x)^{(i)} - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{m}|\text{slope}|$$

As λ increases
slopes decreases
so some features
deleted (both important
features)



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

$$= \theta_0 + 0.54 x_1 + 0.82 x_2 + 0.10 x_3$$

↓

0 feature removed.

* Elastic Net [Combination of L_1 and L_2 Norm]

Ridge

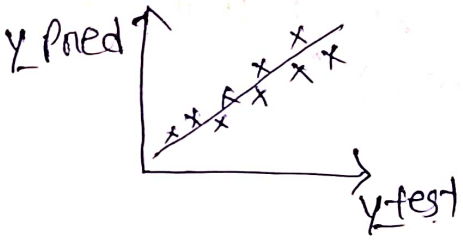
$$\text{Cost fun}^n = \frac{1}{m}\sum_{i=1}^{n}(h_\theta(x)^i - y^i)^2 + \lambda\sum(\text{slope})^2$$

$$+\lambda_2 \sum_{i=1}^{n} |\text{slope}|$$

↑ Lasso.

Assumption of Linear Regression

① Relation between test thouth and test predicted data should a linear.



② Recidual = Y test − Y preds

→ Recidual should follow approximate gaussian distribution.

③ Plot between prediction and recidual follow uniform distribution.