Lecture - 01

→ Introduction to ML (AI vs ML vs DL vs DS)

→ Supervised ML and Unsupervised ML.

→ Linear Regression (Maths and Geometric Intuitions)

→ R^2 and Adjusted R^2.

→ Ridge and Lasso Regression.

Introduction to ML:

AI vs ML vs DL vs DS

Artificial Intelligence:

It is the Simulation of human intelligence processes by machines, especially Computer Systems. Specific applications of AI include Expert System, NLP, Speech Recognization and Computer Vision.

Machine Learning:

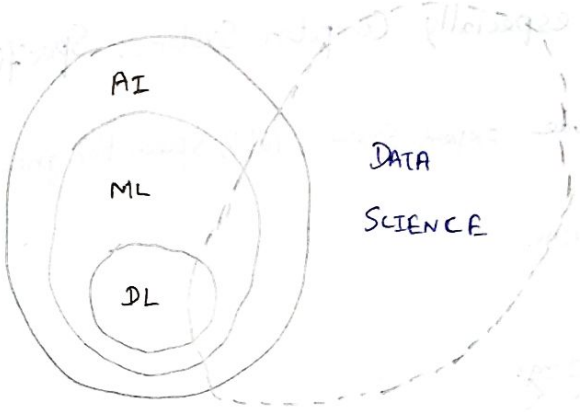
It is the branch of artificial intelligence (AI) and Computer Science which focuses on the use of data and algorithm to imitate the way human learn to Machines.

Deep Learning:

Deep learning is a method in artificial intelligence (AI) that teaches Computers to process data in a way that is inspired by human brains. [Using Neural network]

Data Science:

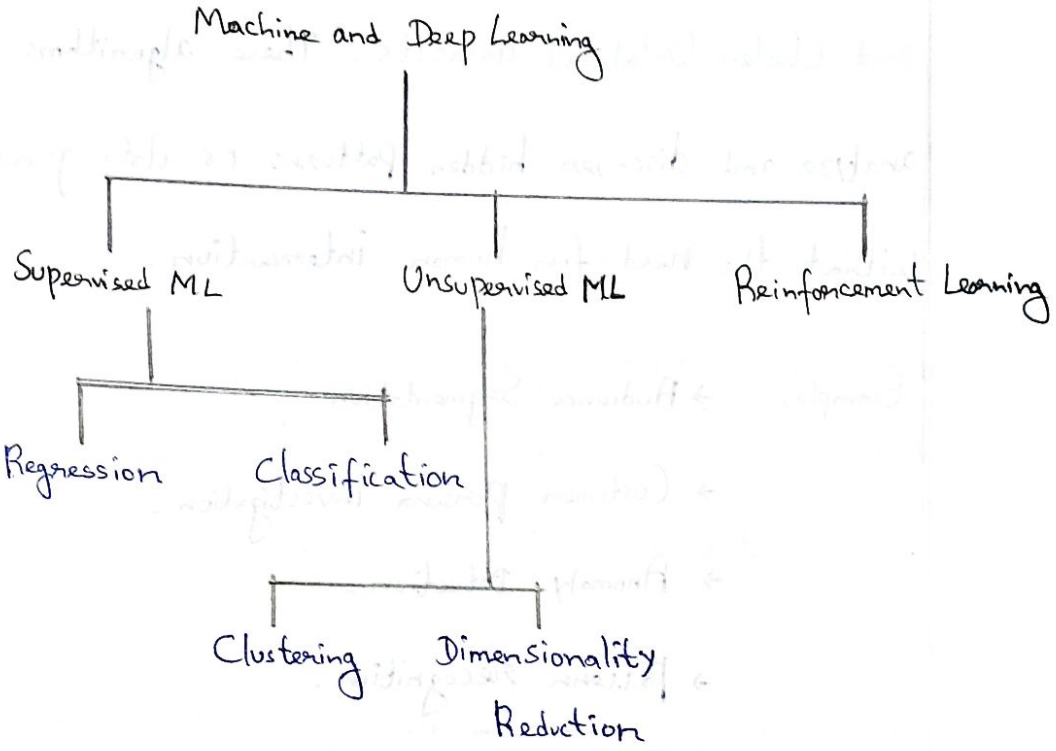
Data Science is the study of data to extract Meaningful insights for business. It is multidisciplinary approach that combines principles and practices from the fields of Mathematics, statistics, artificial intelligence and Computer Engineering to analyze large amount of data.



Examples,

Self driving Cars, Predictive modelling, Analyzing, Netflix Movie Recommendations etc.

Supervised and Unsupervised Learning:



Supervised Learning:

It is the subcategory of Machine learning and artificial Intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data (or) predict outcomes accurately.

Example,
- → Text categorization.
- → Face Detection.
- → Spam detection.
- → Weather forecasting etc.

Unsupervised Learning:

It uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms to analyze and discover hidden patterns or data groupings without the need for human intervention.

Example,
- → Audience Segmentation.
- → Customer persona investigation.
- → Anomaly Detection.
- → Pattern recognition.
- → Inventory Management etc.

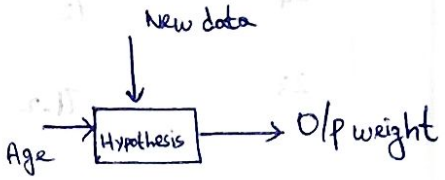
Reinforcement Learning:

It is a machine learning training method based on rewarding desired behaviours and/or punishing undesired ones. In General, a Reinforcement learning agent is able to perceive and interpret its environment, take actions and learn through trail and Error.

Example,
- → Playing a games [chess, Go]
- → Humanoid Robots.
- → Self driving Cars etc.

Supervised ML:

Suppose we have a dataset,



Age	Weight
24	62
25	63
21	72
27	62

We want to predict the weight respect to age. Suppose New input of age, we want to predict weight.

Independent Variable → Age and other features in the dataset.

dependent Variable → Weight [predicted feature]

Regression problem:

It is the type of problem in Supervised learning, this type of problem cannot classify. Instead, this problem Output as a Continuous Variable. These type of numerical output problem is called as regression.

Example,

Age	weight
24	72
23	71.2
25	71.5

→ o/p

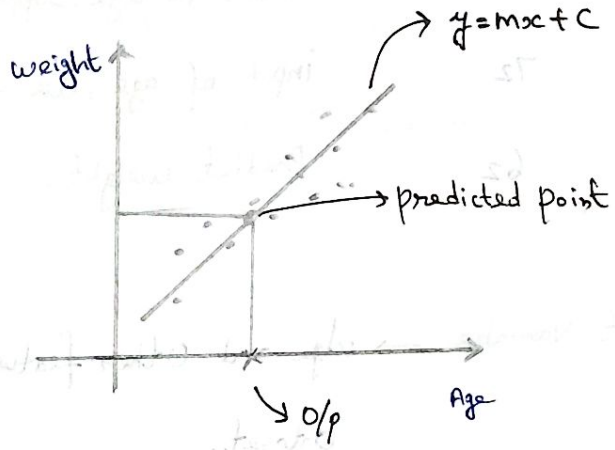
↓

Continuous Variable

↓

Regression problem

If we are plot this, look like



Classification Problem:

In this problem, we can Classify the output. The output dependent variable is based on all or some independent variable in the dataset. The classification problem is for only Categorical variable Output.

Example,

No of hours study	No of hours play	No of hour sleep	Pass / Fail
6	4	3	P
5	2	8	F
-	-	-	P
-	-	-	F

Binary Classification

→ o/p Categorical variable.

Unsupervised ML:

The Unsupervised ML has no labeled one.

Suppose we have a dataset,

Person	Grender	Height
AA	Male	72
BB	female	72.5
Cc	Male	73

The above dataset, There are only independent variable. There is no dependent variable (or) any labeled feature. So, this non-labeled one and this is called as the Unsupervised ML.

It is mainly focus on categorizing of data and predict nearest point.

There are two category are the,

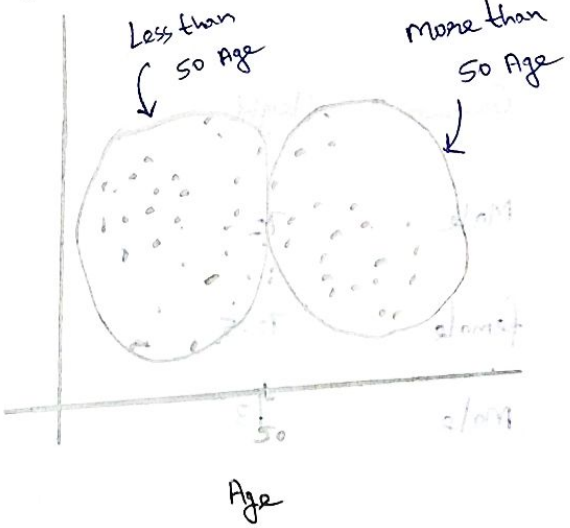
→ Clustering.

→ Dimensionality Reduction.

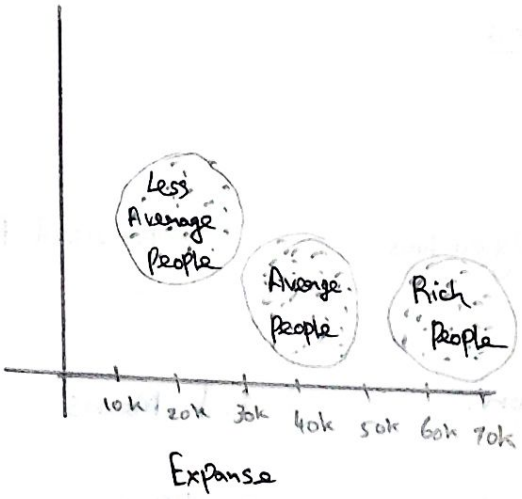
Clustering:

→ Clustering is nothing but grouping of the particular feature on the dataset.

Example,



Customer Segmentation is one of the perfect example for clustering.



Based on this, The rich people have more expanse in products. So again analyze products which products are highly purchased from rich people. Based on this, The such product productivity is high.

Dimensionality Reduction:

The dimensionality reduction is nothing but the feature reduction.

Suppose we have 500 features, we reduce into 50 (or) less feature.

Feature Reduction on dimension of dataset. (100 features)

↓

Using various algorithm [PCA, LDA] etc.

↓

Reduce features

Discussing Algorithms:

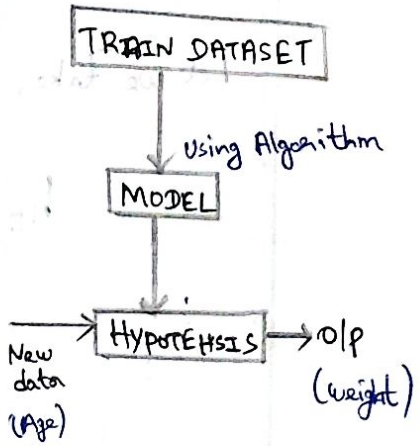
Supervised Algorithms	Unsupervised Algorithms
1. Linear Regression.	1. k Means.
2. Ridge and Lasso Regression.	2. DB Scan.
3. Logistic Regression.	3. Hierachical Clustering.
4. Decision Tree.	4. k Nearest Neighbour Clustering.
5. AdaBoost.	5. PCA.
6. Random forest.	6. LDA.
7. Gradient Boosting.	
8. Xg boost.	
9. Naive Bayes.	
10. SVM, KNN	

The above algorithm are the most important algorithm in machine learning and use it various applications.

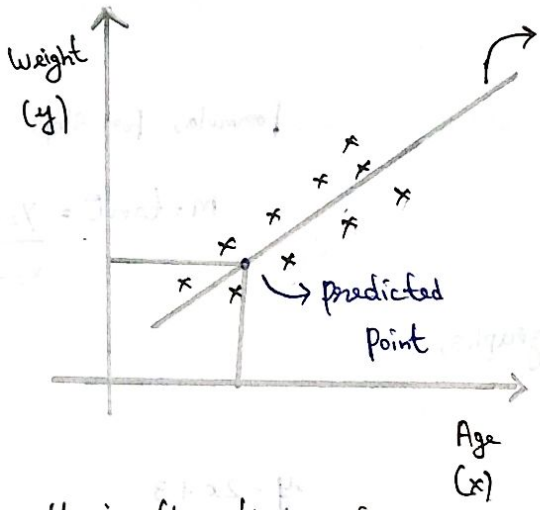
Linear Regression:

Linear Regression analysis is Used to predict the value of a Variable based on the value of another Variable Using Constructing the Straight line.

In normally,



Model is evaluated by Performance metrices.



Equation of straight line,

$$y = mx + c$$

Also written as,

$$y = \beta_0 + \beta_1 x$$

(or)

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

y is the linear function of X.

Equation of Straight line:

$$y = mx^{(i)} + c$$

$$y = \beta_0 + \beta_1 x^{(i)}$$

$$h_\theta(x)^{(i)} = \theta_0 + \theta_1 x^{(i)}$$

(i) → Indicates the independent Variable.

Equation of Straight line:

Let we take,

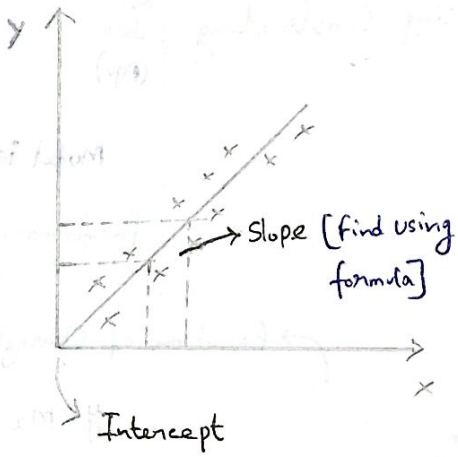
$$h_\theta(x) = \theta_0 + \theta_1 x$$

When $x = 0$

$\theta_0 \rightarrow$ Intercept

$\theta_1 \rightarrow$ Slope (or) Coefficient

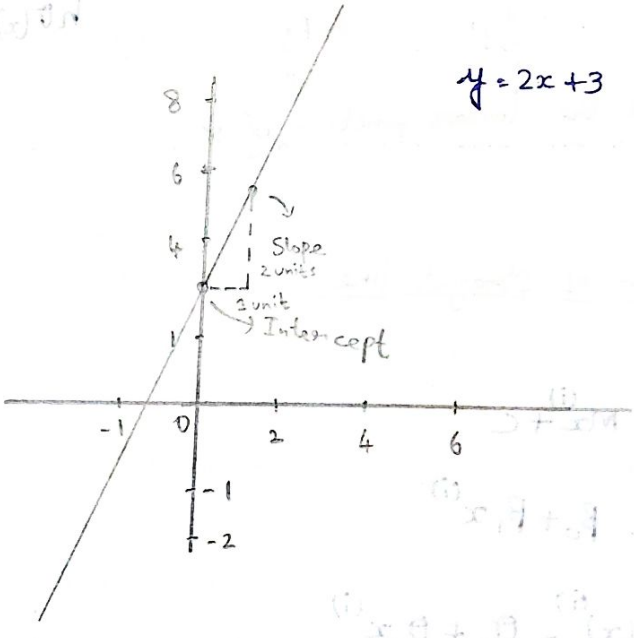
$x_i \rightarrow$ data points.



Formula, for slope

$$m = \tan\theta = \frac{y_2 - y_1}{x_2 - x_1}$$

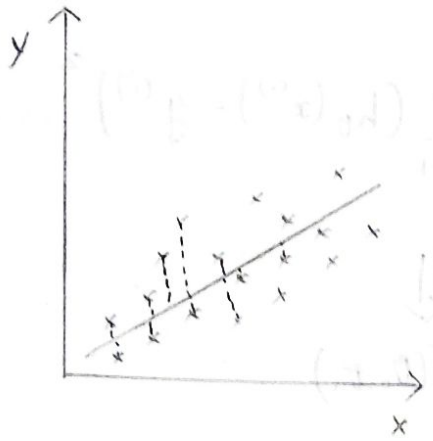
Equation for lines on graphs,



$y = 2x + 3$

The goal of linear regression is to find best fit line across the datapoints.

How to find best fit line ?

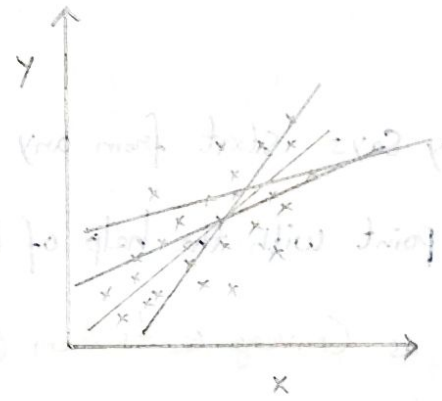


The datapoints from the line (distance) should be Minimal.

The Calculation is done by Cost function.

Actually, the Calculation is done every datapoints and Summation itself.

Samelike, we can Create multiple linear line.



Then Conclude all the line, which line Contains minimal distance from the datapoints. That line is the best fit line.

Cost function : [Distance Formula]

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

Cost function,

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

↳ Cost function ↓ Entire Equation called as Squared Error function

$h_\theta(x)$ → line Equation

y → data points (Actual)

$1/M$ → Average of all Points.

Σ → Summation of all points.

$(\,)^2$ → Avoid neg value

$1/2$ → For Simple derivative Purpose.

Suppose,

$\frac{d(x^2)}{dx} = 2x \Rightarrow \frac{2x \times 1}{2}$ [this type of purpose]

What we need to solve ?

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

$$\Downarrow$$

$$J(\theta_0, \theta_1)$$

We need to minimize [Cost function]

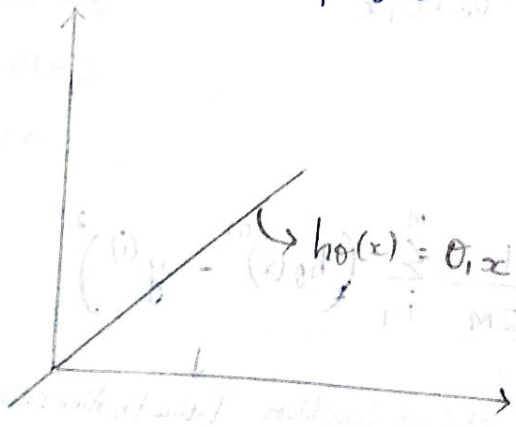
Consider Some Cases :

Before that take $\theta_0 = 0$ for linear regression because the intercept meets at 0.

$\theta_0 = 0$ [It actually says start from any fixed point and find out minimal point with the help of θ_1]. It is Something applied Called as Convergence theorem (or) Algorithm.

Case 1:

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



After fixing $\theta_0 = 0$ and keep changing θ_1 value to find Out best fit line with the help of gradient descent.

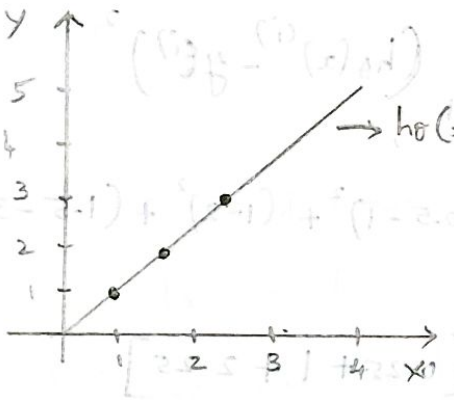
Case 2:

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

Suppose, we have the datapoints (1,1), (2,2), (3,3).

Consider the slope value, [case 2-1]

$$\boxed{\theta_1 = 1}$$



Try to find Minimal with respect to Cost function,

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} \left(h_\theta(x)^{(i)} - y^{(i)}\right)^2$$ ∴ Apply datapoints value.

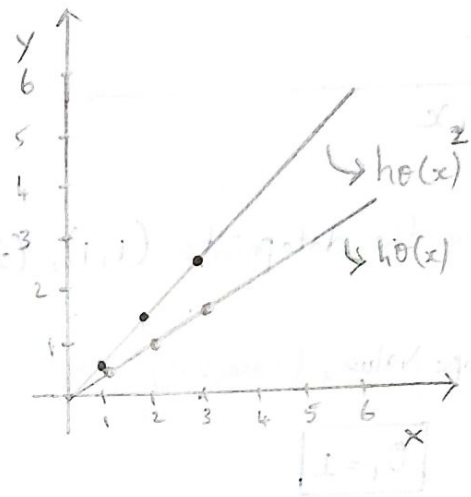
$$= \frac{1}{2\times 3} \left[(1-1)^2 + (2-2)^2 + (3-3)^2\right]$$

$$= \frac{1}{6} [0+0+0] \Rightarrow 0$$

$$\boxed{J(\theta_1) = 0}$$

[Case 2.2]

Consider the slope value, $\theta_1 = 0.5$



$\theta_1 = 0.5$

$h\theta(x^{(1)}) = 0.5 * 1$

$\boxed{h\theta(x^{(1)}) = 0.5}$

$h\theta(x^{(2)}) = 0.5 * 2$

$\boxed{h\theta(x^{(2)}) = 1}$

$h\theta(x^{(3)}) = 0.5 * 3$

$\boxed{h\theta(x^{(3)}) = 1.5}$

Let's find $J(\theta_1)$,

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} \left(h\theta(x)^{(i)} - y^{(i)}\right)^2$$

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

$$= \frac{1}{2\times 3} \left[0.25 + 1 + 2.25\right]$$

$$= 0.58$$

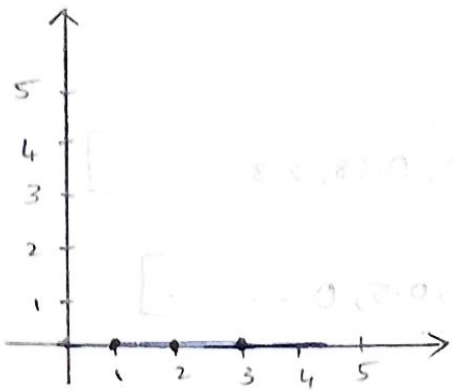
$$\boxed{J(\theta_1) = 0.58}$$

[Case 2.3]

Consider the slope value, $\theta_1 = 0$

So,

$$h\theta(x^{(1)}) = 0, \quad h\theta(x^{(2)}) = 0, \quad h\theta(x^{(3)}) = 0$$



Let's find $J(\theta_1)$,

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

$$= \frac{1}{6}[1+4+9] = 2.3$$

$$\boxed{J(\theta_1) = 2.3}$$

To try this number of times and get different different $J(\theta_1)$ values. Based on this value, to find which one is minimal distance with the help of Gradient descent.

Real time:

We are just considering θ_0 as fixed and try different $J(\theta_1)$ value. The actual datapoints are also vary based on $h_\theta(x)$. [It means Actual datapoints x and y vary].

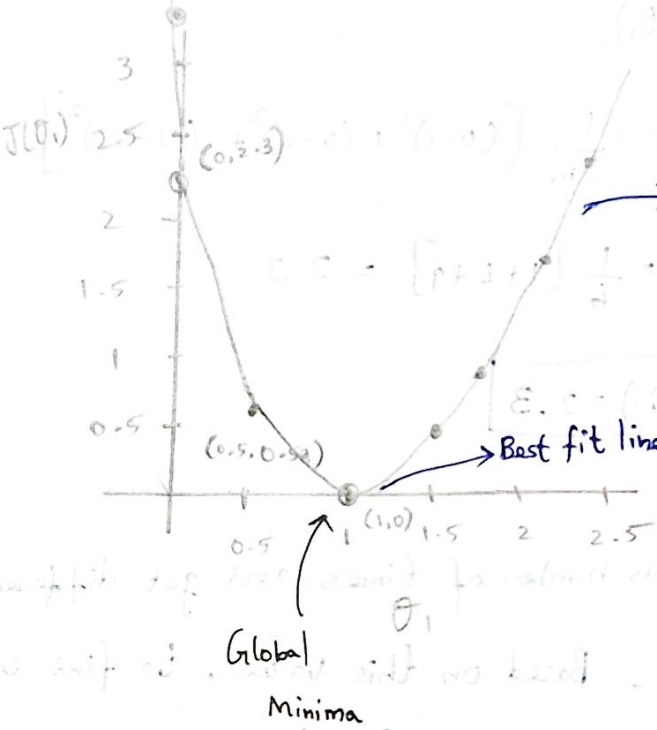
In real time, we just θ_0 as fixed based on the graph and try different $J(\theta_1)$ value. Finally find the minimal one Using Gradient descent.

Case 3:

$J(\theta_1) = [0, 0.58, 2.3 \ldots\ldots]$

$\theta_1 = [1, 0.5, 0 \ldots\ldots]$

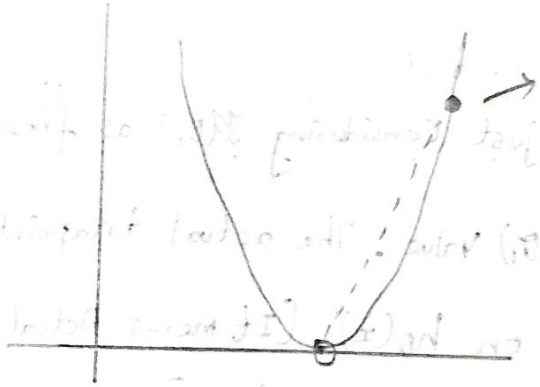
[Considering Slope value]



→ Gradient descent.

→ Best fit line. [The distance between predicted and real Points are very very less]

Global Minima



→ Suppose this is the point we need to come minimum

Convergence Algorithm:

It is the algorithm for gradient desent. We can Say [Repeat Until Convergence]

Algorithm,

Repeat until Convergence

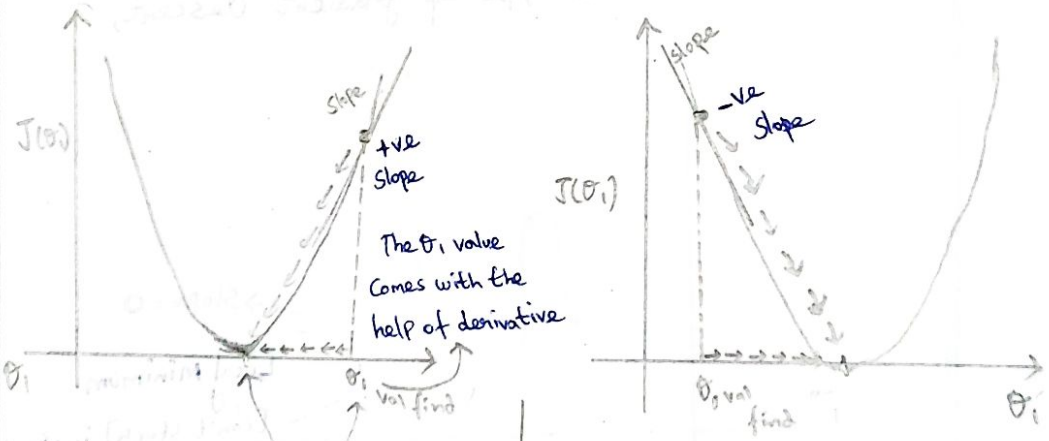
{

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}$$

↗ Derivative of Slope

}

Suppose we have two cost function,



When get + slope,

$$\theta_1 := \theta_1 - \alpha(+ve)$$

↳ This formula has various Summation and iteration to reach global minimum.

α → Learning rate

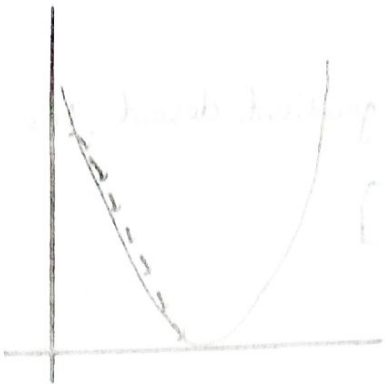
When get - slope,

$$\theta_1 := \theta_1 - \alpha(-ve)$$

$$\theta_1 := \theta_1 + \alpha(ve)$$

α → Learning rate.

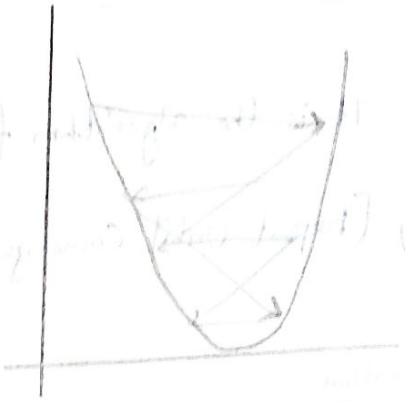
How much Speed the Solution to find Global Minima is called as Learning rate.



If α is too small, gradient descent can be slow.

$\alpha = 0.01$

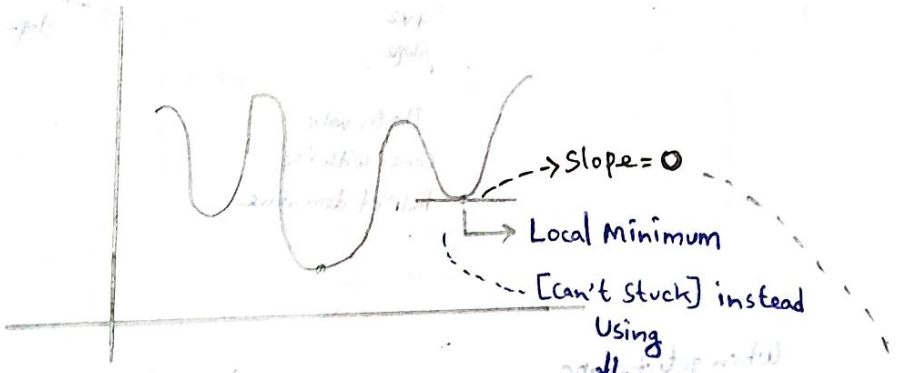
[But it takes time to reach]



If α is too large, gradient descent can overshoot the minimum, It may to converge or even diverge.

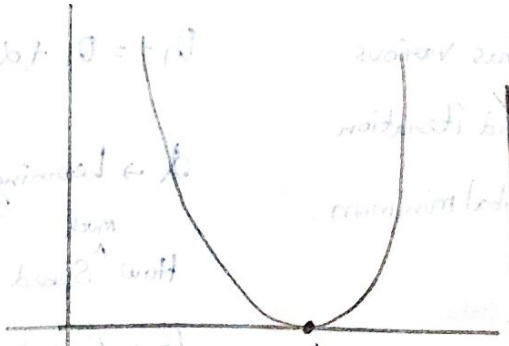
Maintain Middle α → Learning rate is better.

Suppose, we have this type of gradient descent,



⇓ with help of deep learning algorithm

Adam optimizer, Rms prop



$$\theta_1 := \theta_1 - \alpha(0)$$

$$\theta_1 := \theta_1$$

↳ Global Minimum

Interview Question:

Does any local minima is present in linear regression ?

In Machine learning problems Moreover see the Global minimum. In Some Cases, we are Seen in local Minimum are found large Convergence. At that time, Using deep learning algorithm (or) optimizers to Solve local Minimum to reach Global Minimum.

Convergence algorithm is nothing but a Gradient descent.

Algorithm,

Repeat until Convergence

{

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

Find Out the $\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$,

When $j = 0$ and 1 [Because θ_0 and θ_1] i.e $h_\theta(x) = \theta_0 + \theta_1 x$

Rewrite us,

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

Partial derivative applied

$j=0$,

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

$\frac{1}{2m} x^2$

$2 \times \frac{1}{2m} = \frac{1}{m}$

$j=1$,

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

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

Repeat until Convergence

{

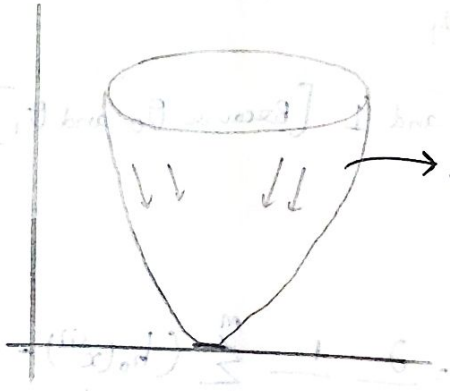
$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

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

}

This is known as the linear regression.

Suppose we have multiple number of features $x_1, x_2, x_3, x_4 \ldots x_n$. The Gradient descent look like,



3D Gradient Descent.

Performance Metrics:

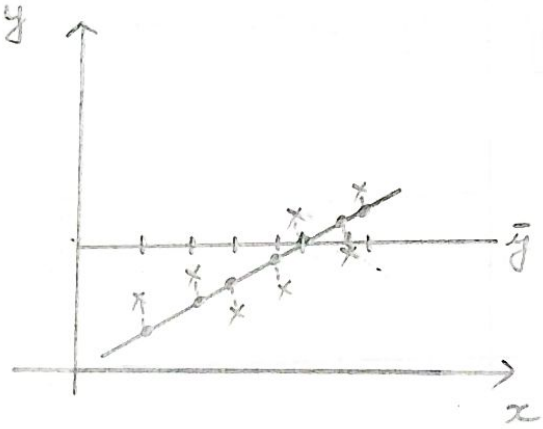
R^2 and Adjusted R^2.

Performance metrices says that the how much this model can be efficient.

R^2:

$$R^2 = 1 - \frac{SS_{Residal}}{SS_{total}} \qquad \hat{y}_i = h_\theta(x).$$

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



$x \rightarrow y_i$ [datapoints]

$\bullet \rightarrow h_\theta(x)$ [predicted point]

$1 \rightarrow$ Mean point

$\bar{y} \rightarrow$ Mean of y.

$1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$ [This distance value is low / This distance value is high]

$1 - \frac{\text{low}}{\text{high}}$ → Small number with 0 to 1.

$= 1 - \frac{\text{Low}}{\text{high}}$

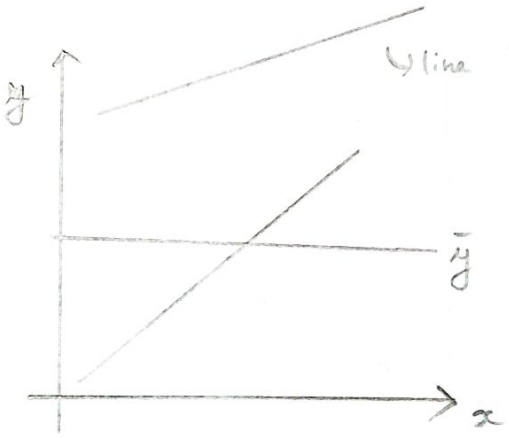
$= 1 - \text{Small number} = 0.\text{Something}$ [Positive number]

Suppose, $1 - \text{Small number} = 0.9$

90 % of accuracy.

Suppose, $1 - \text{big number} =$ -Ve number.

This case is probaly occuring in some Situations, At that time. The graph look likes,



In this case, Very less number of Chances are having.

Suppose we have dataset,

Dataset features:

Bedrooms price $[R^2 = 85\%]$

Bedrooms location price $[R^2 = 90\% \uparrow\uparrow]$ increases because 'location' feature correlated to price.

Suppose we have one or two uncorrelated feature,

Gender Bedrooms Location price $[R^2 = 91\%]$

R^2 increases, But Gender is no correlated to price.

So, R^2 take 91% as efficiency. But this is wrong efficiency.

Therefore, we move to the adjusted R^2.

Adjusted R^2:

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

P = features (or) predictors.

N = No of datapoints.

if $P=2$ $= R^2 = 90\%$ but R^2 adjusted $= 86\%$.

↓ Increase ↓ decrease

if $P=3$ $= R^2 = 91\%$ but R^2 adjusted $= 82\%$

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

(1−R²)(N−1) → Big number

N−P−1 → Small number.

When $P>3$ (or) any feature the whole $N-P-1$ keeps on decreasing.

So, The R^2 is little bit increasing when the Uncorrelated feature not huge. The adjusted R^2 Solve that particular problem.