Machine learning

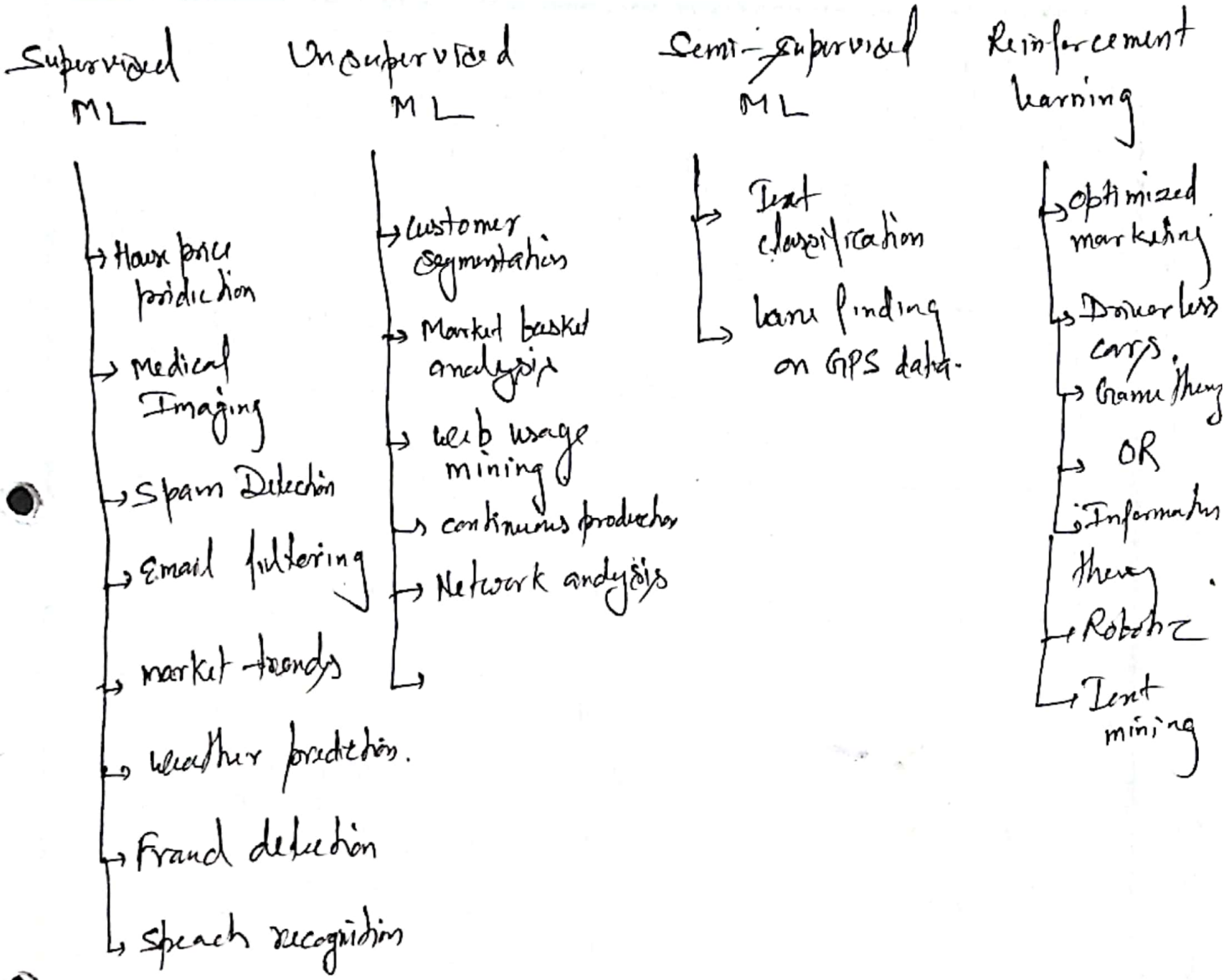
A computer program is said to learn from Experience E with respect to some class of task T and performance P, if its performance in task T, as measured by P, improves with experience E.

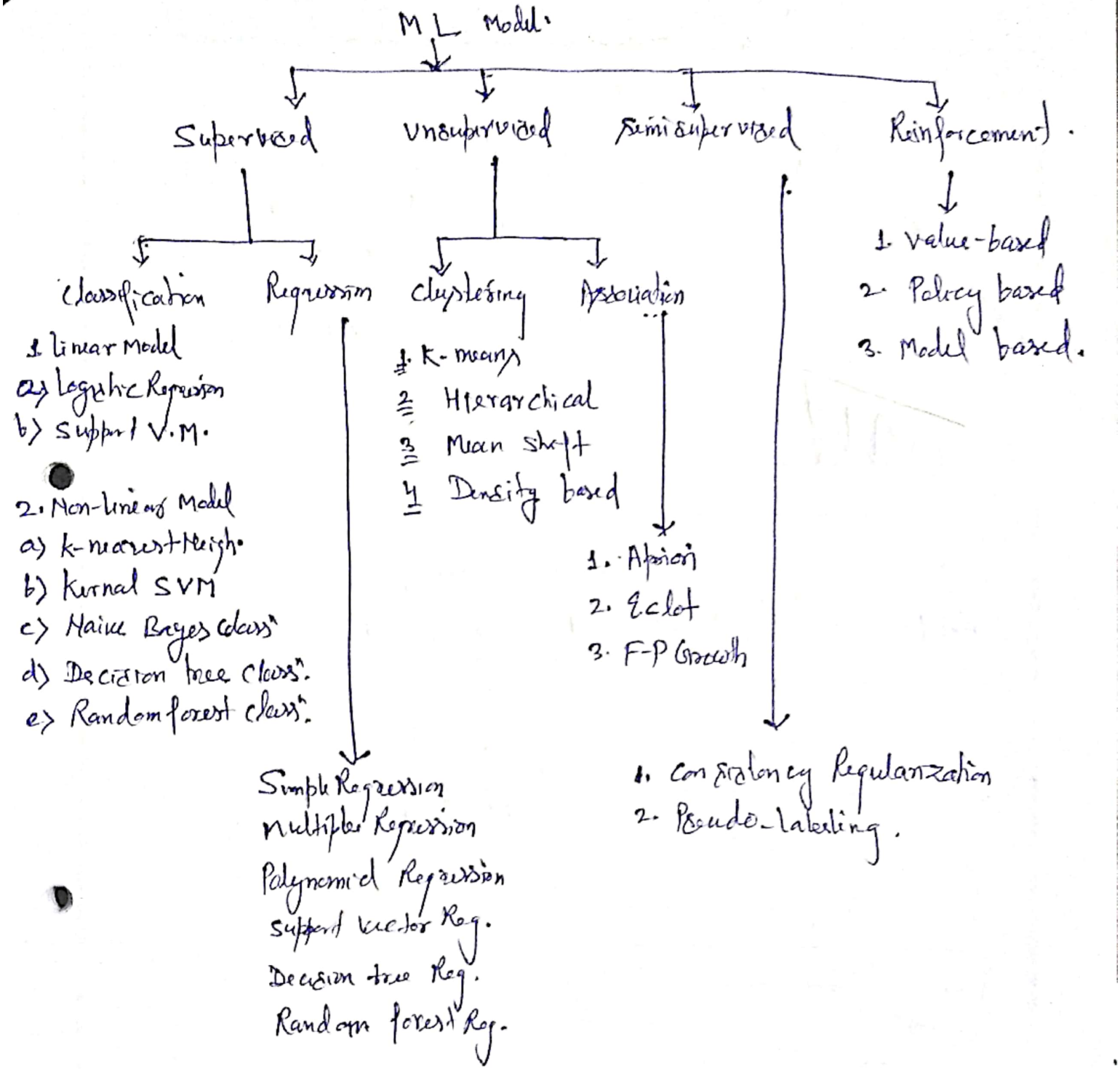
Type of Machine learning

Based of the methods and way of learning, Machine learning is divided into four type

1. Supervised ML
2. Unsupervised ML
3. Semi-supervised ML
4. Reinforcement learning.

ML

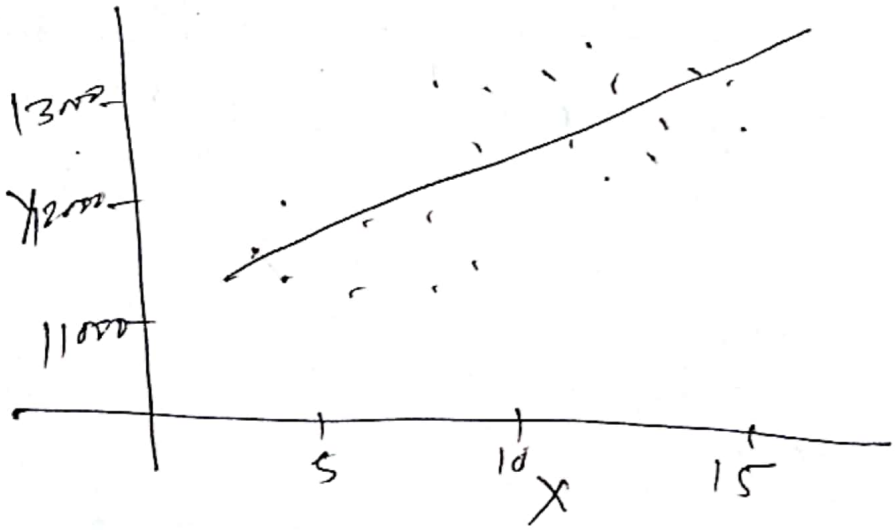

Supervised ML
Unsupervised ML
Semi-supervised ML
Reinforcement learning
House price prediction
Medical Imaging
Spam Detection
Email filtering
market trends
weather prediction.
Fraud detection
Speach recognition
customer segmentation
Market basket analysis
web usage mining
continuous production
Network analysis
Text classification
Land finding on GPS data.
optimized marketing
Driverless cars.
Game theory
OR
Information theory
Robotz
Text mining



Linear models for regression: A Regression model is used for finding out the relationship b/w variables and forecasting. Different regression models differ based on

- the kind of relationship b/w dependent and independent variables they are considering
- The number of independent variables getting used.

The linear regression performs the task to predict dependent variable values (y) based on independent variable (x).



In figure X (input) is the work experience and (output) is the salary of a person. Here regression line is the best fit line for our model.

Linear Regression : least square method

$x = (x_1, \ldots x_n)$ $\quad \bar{x} = \frac{\sum x}{n}$

$y = (y_1 \ldots y_n)$ $\quad \bar{y} = \frac{\sum y}{n}$

Fit a linear line of the form.

$$y = mx + c \quad \text{—— (1)}$$

$$\sum y = m \sum x + c \cdot \sum 1$$

$$\sum y = m \sum x + cn \quad \text{—— (2)}$$

$$xy = mx^2 + cx$$

$$\sum xy = m \sum x^2 + c \sum x \quad \text{— (3)}$$

from (2) & (3)

$$m = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

$$c = \bar{y} - m\bar{x}$$

$\Rightarrow \tilde{y} = mx + c$ & Error $= y - \tilde{y}$

Ex.1

x	2	6	5	7
y	3	10	4	13

Ex.2

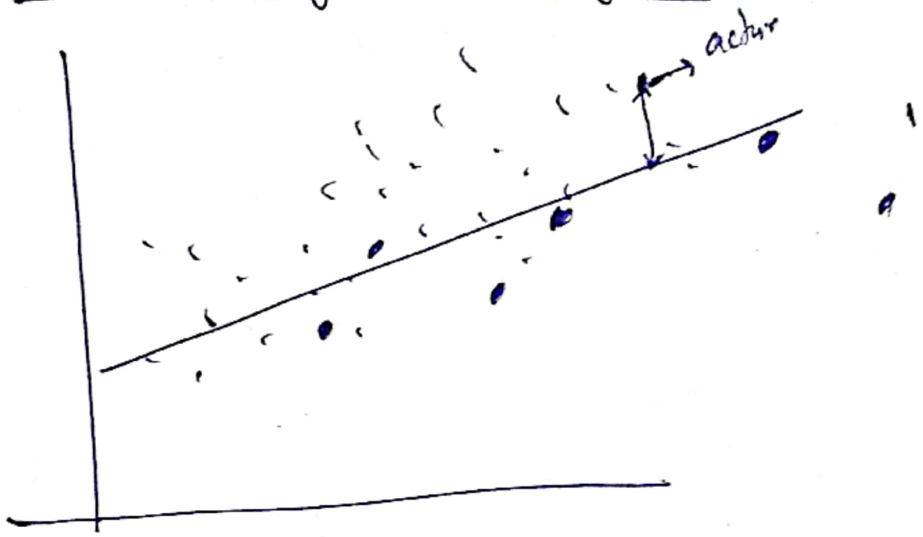
x	0	1	2	3	4	5	6	7	8	9
y	1	3	x	5	7	8	8	9	10	12

Ex.1

x	y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})(y - \bar{y})$
2	3	-3	-4.5	13.5
6.	10.	1	2.5	2.5
5	4	0	-3.5	0
7	13	2	5.5	11.0
$\Sigma x = 20$	$\Sigma y = 30$	0	1	26.0

$\bar{x} = \frac{20}{4} = 5, \quad \bar{y} = \frac{30}{4} = 7.5, \quad m = \frac{26}{8}$

Efficiency Checking.



Error: measure of how far the data is from the fitted regression line.

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

(ii) Mean squared error (MSE) $= \frac{1}{n}\sum(y_i - \hat{y}_i)^2$

(iii) Root mean squared error (RMSE) $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

(iv) ~~Root~~ relative absolute error (RAE) $= \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\sum_{i=1}^{n}|y_i - \bar{y}|}$

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

(vi) R^2 ~~error~~. $R^2 = 1 - RSE$

(vii) R^2 ~~square~~ adjusted $= 1 - \frac{(1-R^2)(N-1)}{N-p-1}$

$N \rightarrow$ Sample size, p - No of features.

Gradient Decent Approach

Hypothesis fun^n for linear Regression.

$$y = \theta_0 + \theta_1 \cdot x$$

where

x : input training data | θ_1 : intercept
y : labels to data. | θ_2 : coefficient of x.

when training the model - it fits the best line to predict the value of y for a given value of x. The model get best regression fit line by find the best θ_1 and θ_2 values.

Finding ~~best fit~~ values of θ_1 & θ_2 for best fit line

cost function: The model aims to predict y value s.t. the error diff. b/w predicted value and true value is minimum. so it is important to update the θ_1 & θ_2 to reach the best value that minimize the error b/w predicted value(pred) and true value (y)

i.e minimize $\frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$

i.e $J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$

Cost fun^n of L.R is RMSE b/w predicted & true value.

* cost fun^n of L.R. is RMSE b/w predicted & true value.

Gradient Descent: · cost fun^n

$$J(\theta_1, \theta_2) = \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x_i) - y_i \right]^2$$

Gradient Descent:

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

Now

$$\frac{\partial}{\partial \theta} J_\theta = \frac{\partial}{\partial \theta} \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x_i) - y \right]^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y \right) \cdot \frac{\partial}{\partial \theta_j} (\theta_{x_i} - y)$$

$$\frac{\partial}{\partial \theta} \cdot J_\theta = \frac{1}{m} \left(h_\theta(x_i) - y \right) x_j$$

New

$$\theta_j = \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} \left[\left(h_\theta(x_i) - y \right) x_j \right]$$

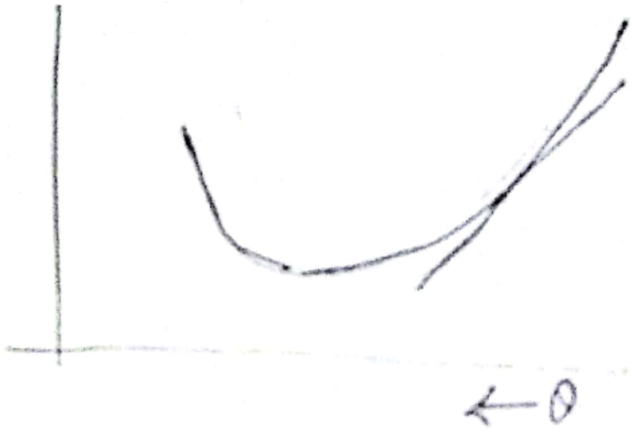
θ_j : weight of hypothesis.

$h_\theta(x_i)$: predicted value of i^{th} input.

j : feature index

α : learning rate. where $x_j = \begin{cases} 1 & \text{for } j = 0 \\ x_i & \text{for } j = 1 \end{cases}$

(i) If slope is +ve θ_j decreases



$\leftarrow \theta$

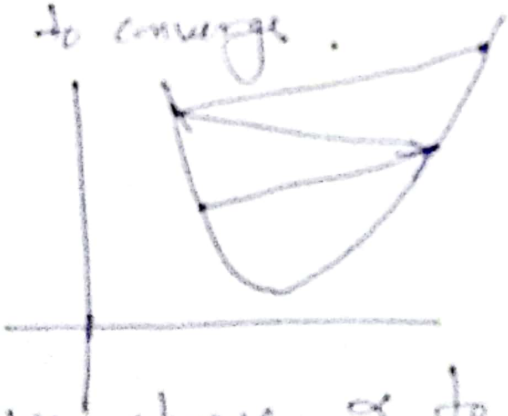
Note (i) for linear regression cost the fun^n graph is always convex shaped.

(ii) i is feature index.

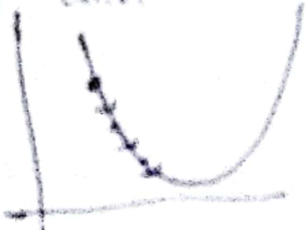
(ii) If slope is -ve : θ_j increases.

$\rightarrow \theta$

(iii) If we choose α to be very large, Gradient Decent can overshoot the minimum. It may fail to converge.



(iv) If we choose α to be very small, Gradient decent will take small steps to reach local minimum and will take longer time to reach mini.



Eg

X	Y
2	3
6	10
5	4
7	13

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

$= [\theta_0 \ \theta_1] \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \end{bmatrix}$

Iteration 1 $\quad \theta_0 = 0, \quad \theta_1 = 0.$

$$h_\theta = [\theta_0 \ \theta_1] \begin{bmatrix} x_0 & x_0 & x_0 & x_0 \\ x_1 & x_2 & x_3 & x_4 \end{bmatrix}$$

$$= [0 \ 0] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 6 & 5 & 7 \end{bmatrix}$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x_i) - y_i \right]^2$$

$$= 16.75$$

Here $j = 0$

$$\theta_j = \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$= 0 - \frac{0.01}{4} \left[(0-3) + (0-10) + (0-4) + (0-13) \right]$$

$$= 0.045$$

$j=1$

$$\theta_j = \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta(x_i)-y_i\right)x_i$$

$$= 0 - \frac{0.001}{4}\left[(0-3)\times 2 + (0-10)*6 + (0-4)*5 + (0-13)*7\right]$$

$$= -\frac{0.001}{4}\left[-6-60-20-91\right] = \cancel{0.02657}\ 0.04425$$

$\Rightarrow$ we have

$\theta_0 = \cancel{0.005}\ 0.0075$ & $\theta_1 = 0\cancel{.02657}\ 0.04425$

Iteration 2 :

Predicted values after iteration 1 with linear regression hypothesis.

$$h_\theta = \begin{bmatrix}\theta_0 & \theta_1\end{bmatrix}\begin{bmatrix} x_0 & x_0 & x_0 & x_0 \\ x_1 & x_2 & x_3 & x_4\end{bmatrix}$$

$$= \begin{bmatrix}\cancel{0.005}\ 0.0075 & \cancel{0.026}\ 0.04425\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1 \\ 2 & 6 & 5 & 7\end{bmatrix}$$

$$= \cancel{\begin{bmatrix}0.057 & 0.161 & 0.135 & 0.187\end{bmatrix}}$$

$$= \begin{bmatrix}0.096 & 0.273 & 0.22875 & 0.31725\end{bmatrix}$$

Now, find θ_0 & θ_1. (same as in previous step).

x	0	1	2	3	4	5	6	7	8	9
y	1	3	5	7	9	11	13	15	17	19

Multiple Linear Regression: multiple linear Regression used to model data or more features and response by fitting a linear line to observed data.

Now, consider a data set with p features (Independent variables) and one response (dependent variable).

Also the dataset (training dataset) contains n observations. Then

- X (feature matrix) = matrix $[x_{ij}]_{n \times p}$

where x_{ij} denotes the value of jth feature for ith observation.

- y (response vector) = $[y_i]_n$

y_i is response of ith observation.

⇒ The regression line for p features is represented as

$$h(x_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

$$h(x_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}$$

Therefore, we can write

$$y_i = h(x_i) + e_i \Rightarrow e_i = y_i - h(x_i)$$

In matrix form model can written as

$$y = X\beta + \epsilon$$

where $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$, $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$, $\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}, \quad x_i = \begin{bmatrix} 1 \\ x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{bmatrix}$$

for which residual error ϵ is minimum.

Here we determine β using least square method.

And we get

$$\hat{\beta} = (X'X)^{-1} X'y$$

and Response vector is estimated as

$$\boxed{\hat{y} = X\hat{\beta}}$$

Polynomial Regression:

General model for polynomial Regression.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_m x^m .$$

Classification:

Classification is a supervised learning approach which can be thought used for categorizing or classifying some unknown items into a discrete set of classes. Classification attempts to learn the relationship between a set of feature variables and a target variable of interest.

The target attribute in classification is a categorical variable with discrete values.

Type of attributes:

1. Binary: Possesses only two values ex. True or False
   - * Symmetric: Both values are equally important in all aspects.
   - * Asymmetric: when both values may not be important.
2. Nominal: when more than two outcomes are possible.
   ex Classification for different colors like green, red, black
   - * Ordinal: values that must have some meaningful order.
     ex grades: A, B, C, D.
   - * Continuous: may have infinite no of values.
     ex weight
   - * Discrete: finite No of values:
     ex colors:

Type of classification algorithms:

Classification algorithms are categorized into two types:

1. Discriminative: It is a very basic classification algorithm used to determine one class row of data.
   Ex Logistic Regression

2. Generative: It models the distribution of individual classes and tries to learn the model that generate the data behind the scenes by estimating assumptions and distributing of the model.
   Ex Naive Bays classifier.

Various classification models

1. Logistic Regression
2. Linear model
3. SVM
4. Non linear model
5. K-nearest Neighbour
6. Naive Bays classifier.

Types of classification algorithms:

Classification algorithms are categorized into two types:

1. Discriminative: It is a very basic classification algorithm used to determine one class row of data.

Ex Logistic Regression

2. Generative: It models the distribution of individual classes and tries to learn the model that generate the data behind the scenes by estimating assumptions and distribution of the model.

Ex Naive Bays classifier.

Various classification models

1. logistic Regression
2. linear model
3. SVM
4. Non linear model
5. K-nearest Neighbour
6. Naive Bays classifier.