Machine Learning

7-days Machine Learning Live Sessions

Agenda :-

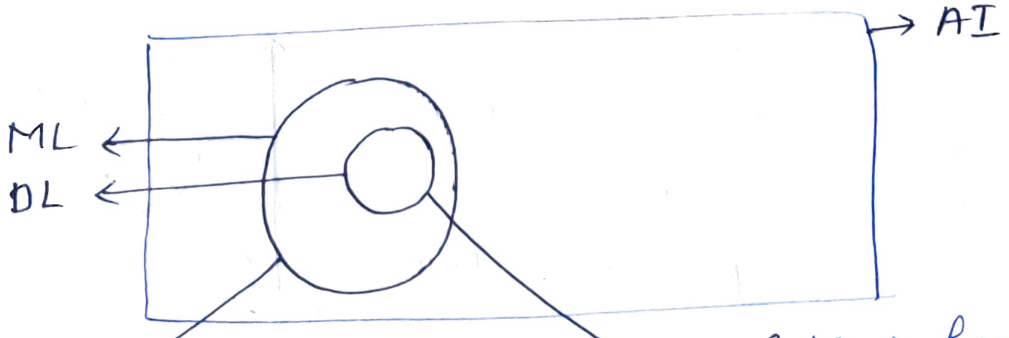
1. Introduction to ML (AI vs ML vs DL vs DS)
2. Supervised ML vs Unsupervised ML
3. Linear Regression (Math's and Geometric intution)
4. R^2 and Adjusted R^2
5. Ridge and Lasso Regression.

(1) AI vs ML vs DL vs DS

→ AI application is able to do it own task without any human intervention

Eg :- Netflix App → Recommendation ← Action, Comedy, Romance

Amazon.in → Recommendation



ML { Stats to analyze, visualize, prediction & forecast

DL { Mimic human brain, Multi layered Neural Network

Machine & Deep Learning

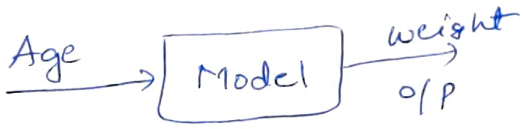
- Supervised ML
  - Regression
  - classification
- unsupervised ML
  - clustering
  - Dimensional Reduction
- Reinforcement Learning

Supervised ML :—

* labeled data
* Independent and dependent features

i/p ↑ Age | weight ↑ o/p

Age	weight
24	62
25	63
21	72
27	62



Independent features : Age

Dependent features : weight

(1) Regression problem :—

Age	weight
24	62
25	63
21	71
27	62

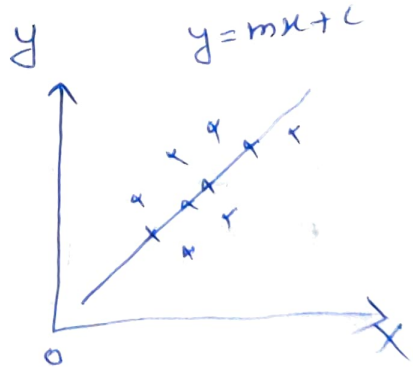
→ o/p

⇓⇓

Continous variable

↓

Regression problem



(ii) classification problem :-

* output variable i/p Discrete (categorical)

No. of hours	No. of play hours	No. of hours Sleep	Pass or Fail ↗ o/p
—	—	—	p
—	—	—	f
—	—	—	f
—	—	—	p

only 2 outputs ⇒ Binary classification

More than 2 outputs ⇒ Multiclass classification.

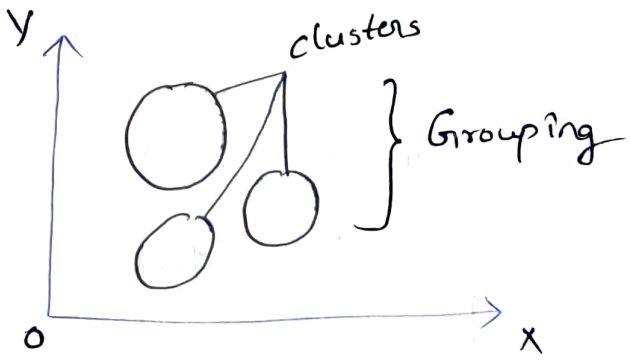
Unsupervised ML :-

* No output or dependent variable

(i) clustering

Salary	Age
20k	20
30k	25
40k	35
50k	42

clustering → customer segmentation



(ii) Dimensionality Reduction :-

1000 features → Convert to → lower dimension

Eg (100 features)

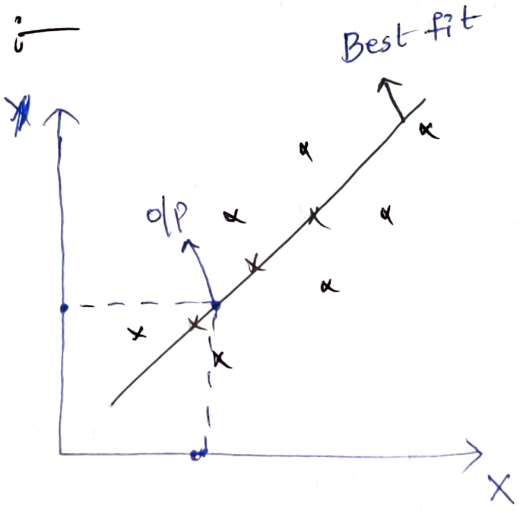
Supervised ML

- Linear Regression
- Ridge & Lasso
- Logistic Regression
- Decision Tree
- AdaBoost
- Random Forest
- Gradient Boosting
- XGBoost
- Naive Bayes
- SVM
- KNN

unsupervised ML

- k-Means
- DBSCAN
- k Nearest Neighbour cluster
- PCA
- LDA

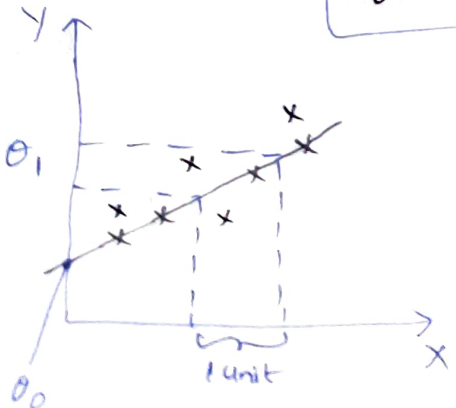
Linear Regression :-



y is a linear function of X

Suppose equation of a straight line

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



θ_0 = Intercept { where best line meet at y-axis }

θ_1 = Scope or Coefficient

⇓

for 1 unit in x-axis vary in y-axis

Best fit line :- The distance between actual values to line is minimum.

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

called as squared error function

$h_\theta(x^{(i)})$ – Predicted

$y^{(i)}$ – Actual.

∵ $\frac{1}{2m}$ used to derivation purpose.

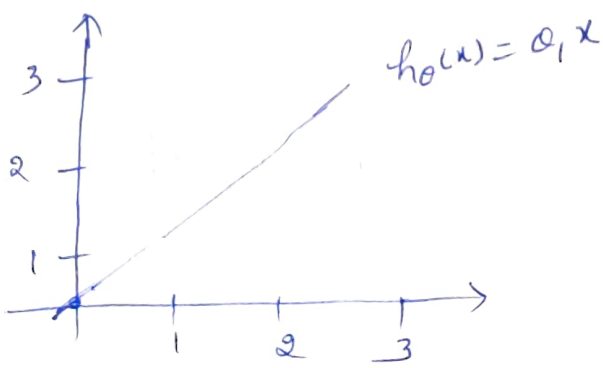
what we need to solve

Minimize $J(\theta_0, \theta_1)$
θ_0, θ_1

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

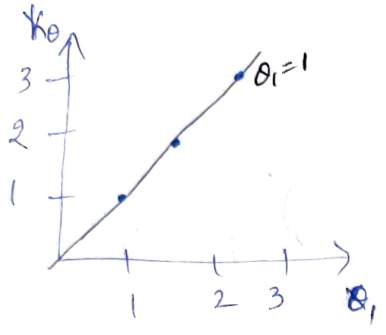
∵ $h_\theta(x) = \theta_0 + \theta_1 x$

if $\theta_0 = 0 \implies h_\theta(x) = \theta_1 x$

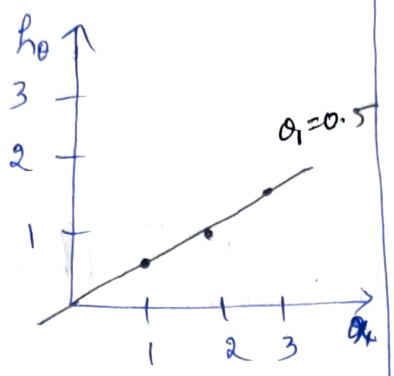


Suppose data points (1,1) (2,2) (3,3)

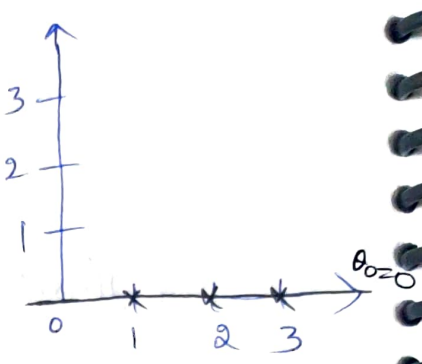
if $\theta_1 = 1$



if $\theta_1 = 0.5$



if $\theta_1 = 0$



if $\theta_1 = 1$

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

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

$$= \frac{0+0+0}{2m}$$

$$= 0$$

if $\theta_1 = 1 \Rightarrow J(\theta_1) = 0$

if $\theta_1 = 0.5$

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

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

$$= 0.58$$

if $\theta_1 = 0.5 \Rightarrow J(\theta_1) = 0.58$

if $\theta_1 = 0$

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

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

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

$$= 2.3$$

if $\theta_1 = 0 \Rightarrow J(\theta_1) = 2.3$

if $\theta_1 = 2$

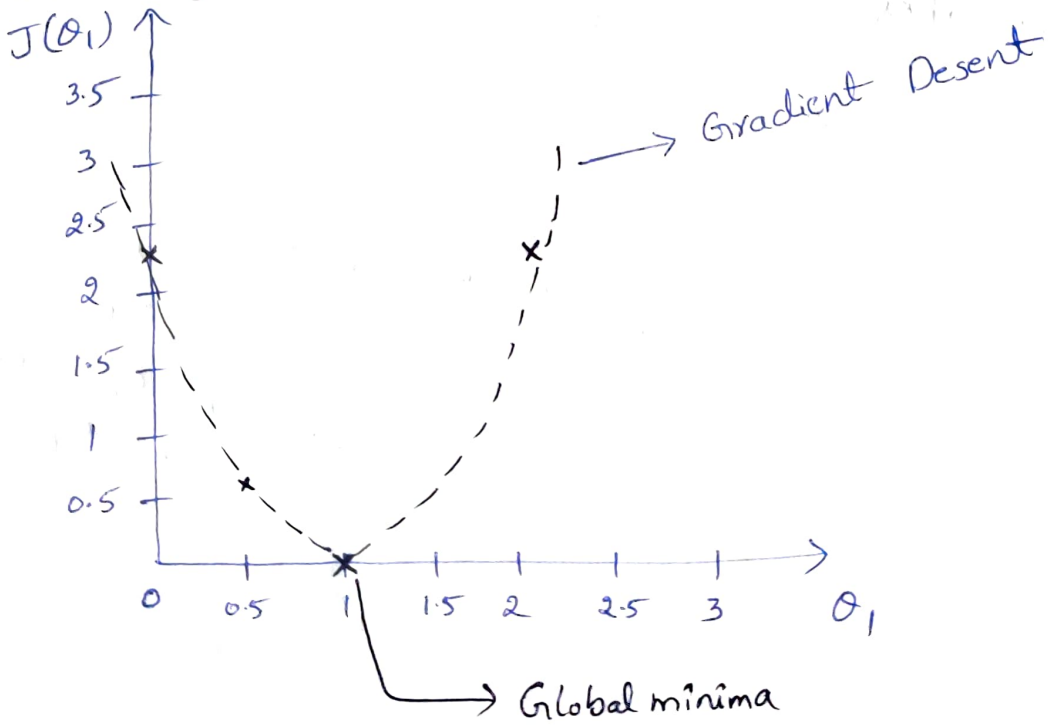
$$J(\theta_1) = \frac{1}{2 \times 3} \left[(2-1)^2 + (4-2)^2 + (6-3)^2 \right]$$

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

$$= 2.3$$

if $\theta_1 = 2 \Rightarrow J(\theta_1) = 2.3$

Cost function graph



Convergence algorithm :-

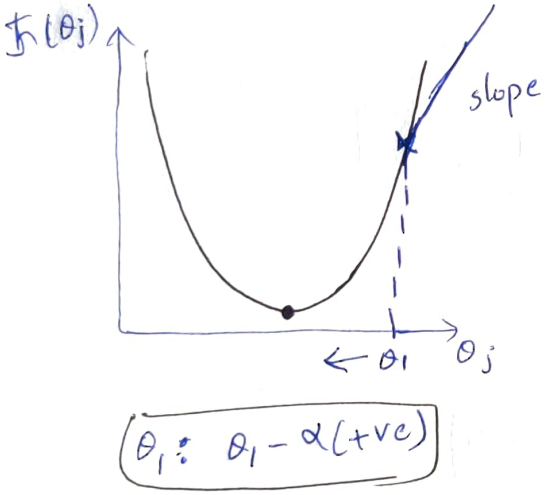
Repeat until Convergence

{

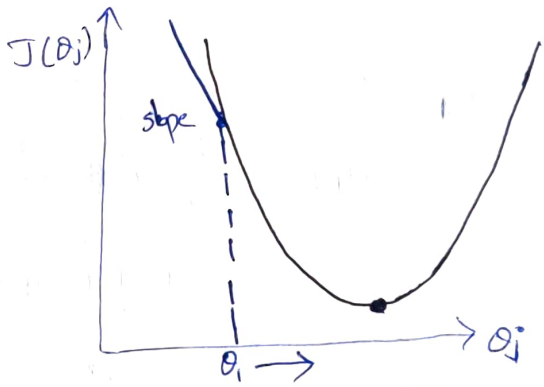
$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$$

}

α = Learning Rate $\Rightarrow$ how speed the value θ_j changes

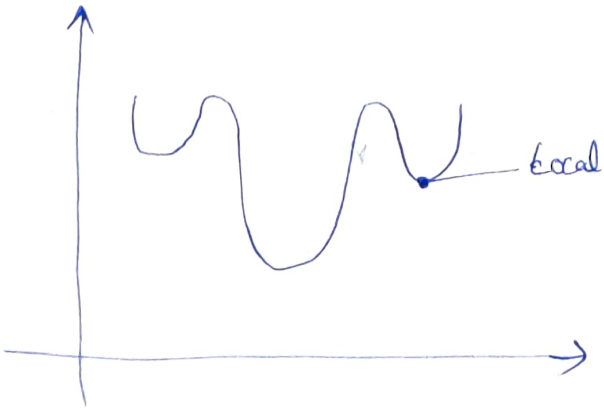


$\theta_1 : \theta_1 - \alpha(+ve)$



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

if graph likes



GRADINT DESCNT Algorithm G—

{

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

}

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

with .r.to θ_0

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

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

w.r.t θ_1

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

Repeat until Convergence

{

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

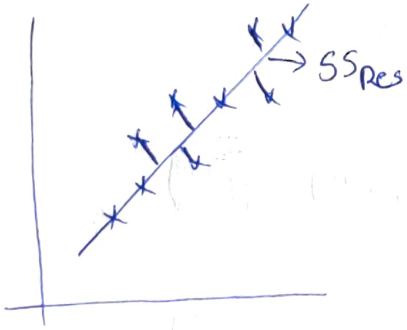
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

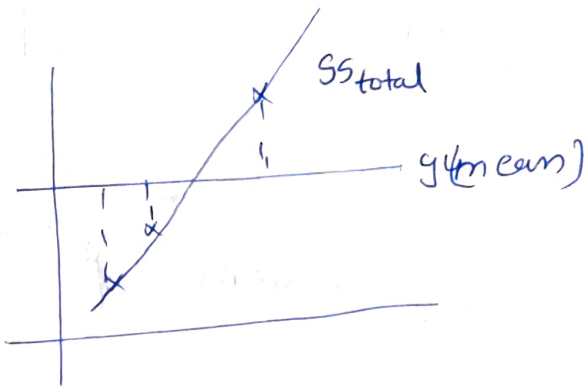
}

R^2 and Adjusted R^2 :—

* R^2 is performance matrics

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





if non-Corelated features added, then R^2 increased. This should be avoided., so using Adjusted R^2

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

N = No. of samples

P = features or predictors

Eg:—

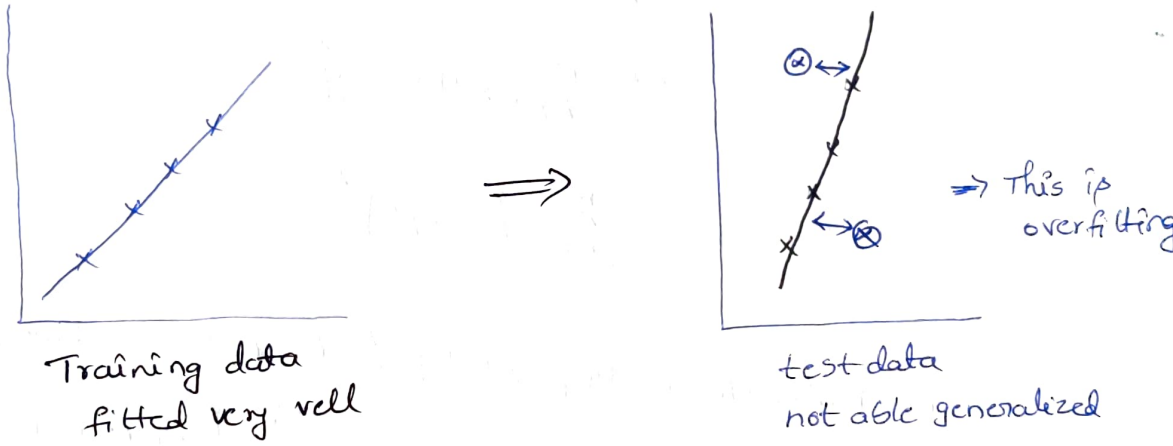
- P = 2 $\Rightarrow$ R^2 = 90 %. Adjusted R^2 = 86 %.
- P = 3 $\Rightarrow$ R^2 = 91 %. Adjusted = 82 %.

Day-2 :—

① Ridge & Lasso Regression
② Assumption of Linear Regression
③ Logistic Regression
④ Confusion matrix
⑤ practical implementation.

Ridge & Lasso Regression :—

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



Training data fitted very well

test data not able generalized

overfitting :—

Low Bias
⇑
Model performs well on — Training data
fail to Perform well on — Test data
⇓
High Variance

underfitting :—

Model accuracy bad with training data ⇒ High Bias

Model accuary also bad with test data ⇒ High Variance

Model 1

Train Accuracy = 90%

Test Accuracy = 80%.

⇓

overfitting

⇓

Low Bias, high variance

Model 2

Train Accuracy = 92%.

Test Accuracy = 91%.

⇓

Generalized model

⇓

Low Bais, Low variance

Model 3

Train Accuracy = 70%.

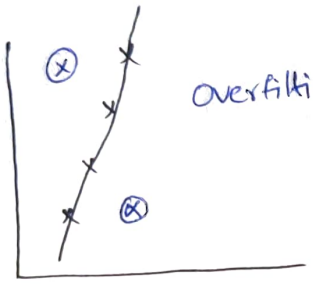
Test Accuracy = 65%.

⇓

underfitting

⇓

High Bias, High Variance



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

$\because h_\theta(x^{(i)}) = \hat{y}$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y} - y^{i}\right)^2$$

Now Introduce Ridge by adding $\lambda(\text{slope})^2$ to Cost function.

λ - hyperparamter

iteration → how many times θ value need to change

Ridge [L2 Regularization] :—

$$\text{Cost function} + \lambda(\text{slope})^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y} - y^{i}\right)^2 + \lambda(\text{slope})^2$$

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

⊗ By using L2 regularization, we can prevent overfitting and generate generalized model.

Lasso [L1 Regularization]

|slope| => helps to feature selection.

Cost function + λ |slope|

$$= \frac{1}{2m} \sum_{i=}^{m} (\hat{y} - y^i) + \lambda |\text{slope}|$$

Suppose

$$h_\theta(x) = \hat{y} = \theta_1 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

$$|\text{slope}| = |\theta_1 + \theta_2 + \theta_3 + \cdots + \theta_n|$$

(i) preventing overfitting

(ii) helps to feature selection (Remove unwanted features)

Ridge Regression (L2 Regularization)

$$\text{Cost function} = \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda (\text{slope})^2$$

Purpose : Preventing overfitting

Lasso Regression (L1 Regularization)

$$\text{Cost function} = \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda |\text{slope}|$$

Pupose : (i) prevent overfitting

(ii) feature selection.

Assumptions of Linear Regression

(i) Gaussian/Normal Distribution → Model will get trained well

(ii) Standardization (scaling data) z-score [$\mu=0, \sigma=1$]

(iii) Linearity

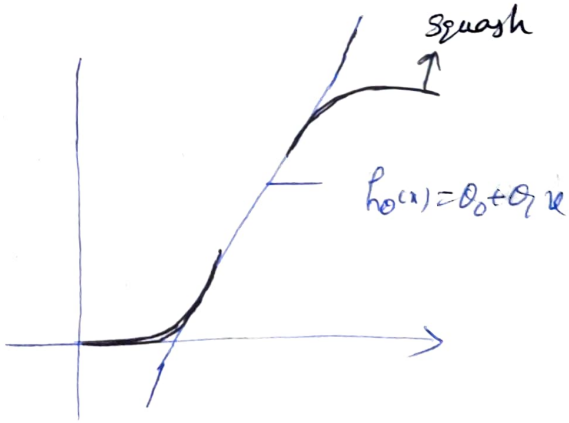
(iv) Multi Colinearity.

Logistic Regression :— [classification]

Decision boundary Logistic Regression

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

$$h_\theta(x) = \theta^T x$$



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

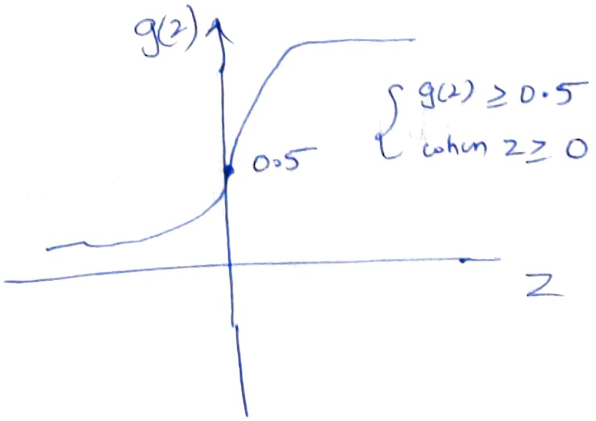
$$h_\theta(x) = g(\theta_0 + \theta_1 x)$$

$$h_\theta(x) = g(z)$$

$$h_\theta(x) = \frac{1}{1+e^{-z}} \rightarrow \text{Sigmoid or logistic function}$$

$$\Downarrow$$

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1)x}}}$$



Training set :—

$\{(x^1, y^1), (x^2, y^2), (x^3, y^3), \dots, (x^n, y^n)\}$

$y \in \{0, 1\}$ → 2 o/ps ⇒ Binary classification.

$h_\theta(z) = \frac{1}{1+e^{-z}}$ ∵ $z = \theta_0 + \theta_1 x$

if assume

$\theta_0 = 0$

⇓

$z = \theta_1 x$

(*) Change parameter θ_1 to get best fit line.

Cost function

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

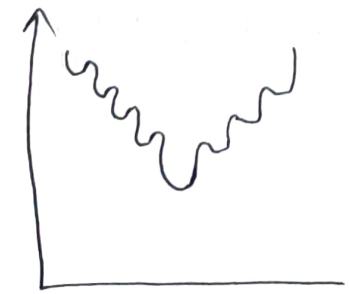
Logistic Regresion = Replace $h_\theta(x) = \frac{1}{1+e^{-(\theta_0 + \theta_1 x)}}$ → this cannot be used as cost function

→ This has local minima problem.

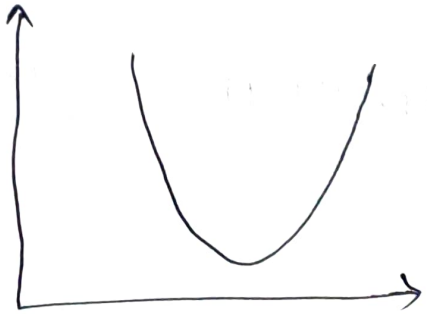
$\frac{1}{1+e^{-z}}$ is a non-convex function.

↓ lot of local minimas, difficult to get global minima

Non-Convex



Convex

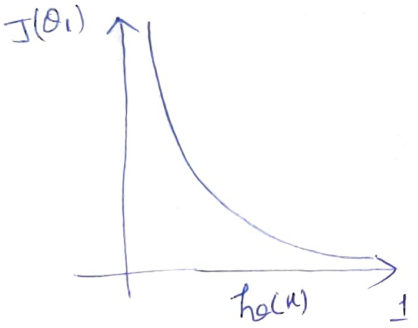


In order to solve local minima problem, the logistic cost function as

$$J(\theta_1) = \begin{cases} -\log(h_\theta(x)) & y=1 \\ -\log(1-h_\theta(x)) & y=0 \end{cases}$$

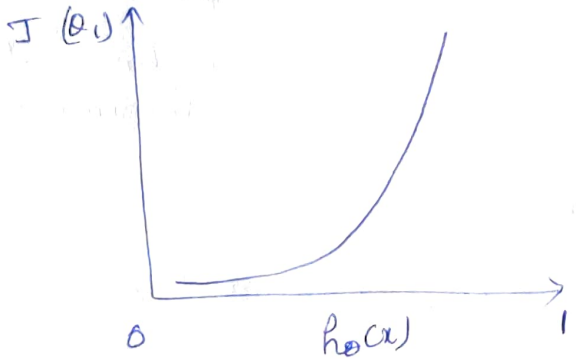
$\because h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$

if $y=1$



$\Rightarrow$ Cost $=0$, if $y=1$, $h_\theta(x)=1$

if $y=0$



$\Rightarrow$ $\begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$

$$\text{Cost}(h_\theta(x^{(i)}), y) = -y\log(h_\theta(x^{(i)})) - (1-y)\log(1-h_\theta(x))$$

Substitute $y=0, y=1$, you will get before one.

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

$$h_\theta(x^i) = \frac{1}{1+e^{-\theta_1 x}}$$

Repeat until Convergence

{

$$\theta_j := \theta_j - \alpha \frac{d}{d\theta_j}\left(J(\theta_1)\right)$$

}

Performance Matrix { classification problem }

Predicted \ Actual	1	0
1	TP	FP
0	FN	TN

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad \{\text{Spam}\}$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad \{\text{Cancer}\}$$

$$\text{F-Score} = (1+\beta^2)\left(\frac{P * Re}{P+Re}\right)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\text{Recall or TPR or Sensitivity} = \frac{TP}{TP+FP}$$

$$\text{Precision} = \frac{TP}{TP+FN}$$

$$\text{F-score} = (1+\beta^2)\ \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

if $\beta = 1 \Rightarrow \frac{2xy}{x+y} \Rightarrow$ F1 score

if $\beta = 0.5 \Rightarrow (1+0.5^2)\left(\frac{P \times R}{P+R}\right)$, FP >> FN

if $\beta = 2 \Rightarrow$ FN >> FP $\Rightarrow$ F2 Score.

Day-36

1. practical Implementation
2. Naive Baye's
3. KNN

Naive Baye's :-

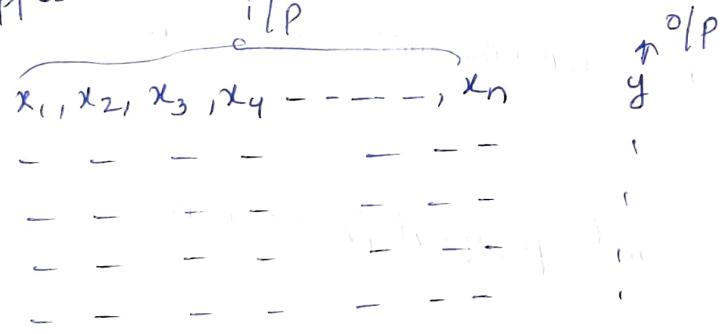
* used for classification problem
* Bayes theorem

$$P(A \text{ and } B) = P(B \text{ and } A)$$

$$P(A) * P(B/A) = P(B) * P(A/B)$$

$$\boxed{P(B/A) = \frac{P(B) * P(A/B)}{P(A)}} \Longrightarrow \text{Baye's theorem}$$

Suppose



$$P(y/x_1, x_2, \dots x_n) = \frac{P(y) * P(x_1, x_2, x_3, \dots, x_n / y_i)}{P(x_1, x_2, x_3, \dots, x_n)}$$

$$= \frac{P(y)\, P(x_1/y_1)\, P(x_2/y_2)\, P(x_3/y_3) \dots P(x_n/y_n)}{P(x_1)\, P(x_2)\, P(x_3) \dots P(x_n)}$$

Assume 5

x_1	x_2	x_3	x_4	y
				Yes/No
				Yes/No

$$P(y=yes/x_i) = \frac{P(yes) * P(x_1/yes) * P(x_2/yes) * P(x_3/yes) * P(x_4/yes)}{P(x_1)\ P(x_2)\ P(x_3)\ P(x_4)}$$

$$P(y=No/x_i) = \frac{P(No) * P(x_1/No) * P(x_2/No) * P(x_3/No) * P(x_4/No)}{P(x_1)\ P(x_2)\ P(x_3)\ P(x_4)}$$

Here $P(x_1)\ P(x_2)\ P(x_3)\ P(x_4)$ → is fixed constant, so ignore.

if $P(y=yes/x_i) = 0.13$ and $P(y=No/x_i) = 0.05$

$\geq 0.5 \Rightarrow 1$

$< 0.5 \Rightarrow 0$

Now do Normalization

$$P(yes/x_i) = \frac{0.13}{0.13+0.05} = 0.72 = 72\%.$$

$$P(No/x_i) = 1 - 0.72 = 28\%.$$

} 72% is best

(Q) Dataset

Day	outlook	Temparature	Humidity	wind	Tennis
D1	Sunny	Hot	High	weak	No
D2	Sunny	Hot	High	strong	No
D3	overcast	Hot	High	weak	Yes
D4	Rain	Mild	High	weak	Yes
D5	Rain	Cool	Normal	weak	yes
D6	Rain	Cool	Normal	Stronge	No
D7	overcast	cool	Normal	stronge	Yes
D8	Sunny	Mild	High	weak	No
D9	Sunny	Cool	Normal	weak	yes
D10	Rain	Mild	Normal	weak	yes
D11	Sunny	Mild	Normal	strong	yes
D12	overcast	Mild	High	Strong	yes
D13	overcast	Hot	Normal	weak	yes
D14	Rain	Mild	High	strong	No.

Sol:

outlook

	Yes	No	P(Y)	P(N)
Sunny	2	3	2/9	3/5
overcast	4	0	4/9	4/5
Rain	3	2	3/9	2/5
Total	9	5		

$$P(yes) = \frac{9}{14}$$

$$P(No) = \frac{5}{14}$$

Tempearature

	Yes	No	P(Y)	P(N)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
cool	3	1	3/9	1/5
	9	5		

$$P(yes) = \frac{9}{14}$$

$$P(No) = \frac{5}{14}$$

$$P(\text{yes}/(\text{sunny, Hot})) = \frac{P(\text{yes}) * P(\text{Sunny}/\text{yes}) * P(\text{Hot}/\text{yes})}{P(\text{sunny})\ P(\text{Hot})}$$ Ignore

$$= \frac{9}{14} * \frac{2}{9} * \frac{2}{9}$$

$$= \frac{\cancel{3}}{\cancel{5}} \times \frac{4}{14 \times 9}$$

$$= \frac{2}{7 \times 9}$$

$$= 2/63 = 0.031$$

$$P(\text{No}/(\text{sunny, Hot})) = \frac{P(\text{No}) * P(\text{sunny}/\text{No}) * P(\text{Hot}/\text{No})}{P(\text{Hot})\ P(\text{sunny})}$$ Ignore

$$= \frac{5}{14} * \frac{3}{5} * \frac{2}{5}$$

$$= \frac{3}{7 \times 5} = \frac{3}{35} = 0.085$$

No, do Normalization

$$P(\text{No}) = \frac{0.085}{0.085 + 0.031} = 73\%$$

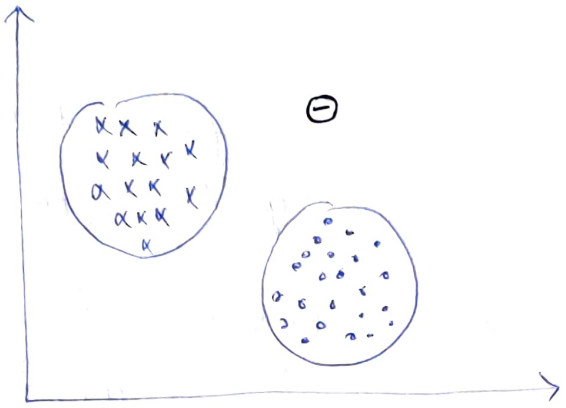
$$P(\text{yes}) = 1 - 73\% = 27\%$$

73% $\Rightarrow$ Hence answer is No

KNN :-

used to solve → Classification
→ Regression

① Classification



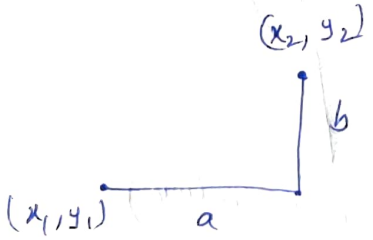
Assume = 5
K = 5

More number of points near to which group that is answer.

Eculedian distance

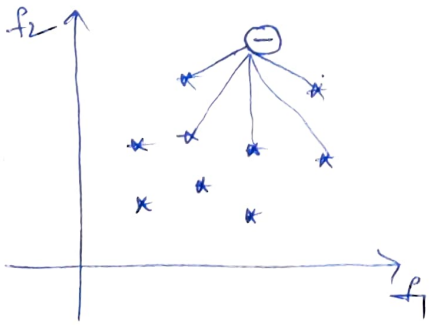
$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Manhat distance



$$\Rightarrow \{ |x_2 - x_1| + |y_2 - y_1| \}$$

Regression :-



Take the avg of points and that is answer

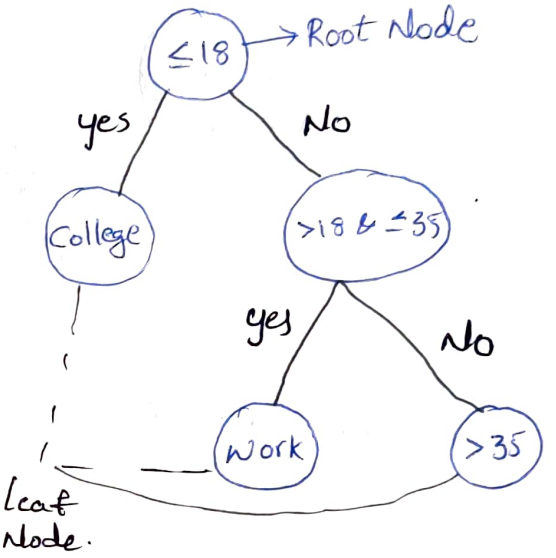
KNN works very bad on
(i) outlier
(ii) Imbalanced dataset

Day-4 :—

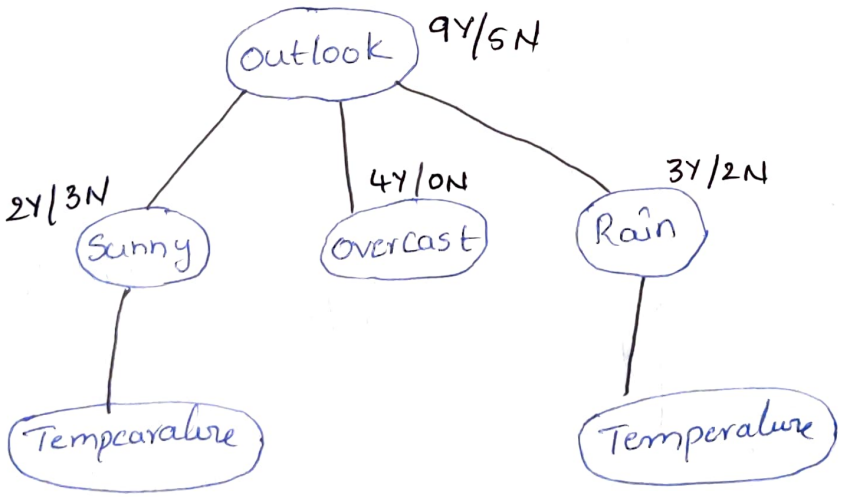
1. Decision Tree classification
2. Decision Tree Regression
3. Practical Implementation
4. Ensemble Techniques

Decision Tree :—

```
if age ≤18 :
    Print("college")
elif (age >18 && age ≤35) :
    Print("work")
else :
    Print("Relative")
```



Dataset



Pure Split
Impure Split

① Purity ⟹ Pure split ?

- Entropy
- Gini Impurity

Entropy $\in \{0, 1\}$

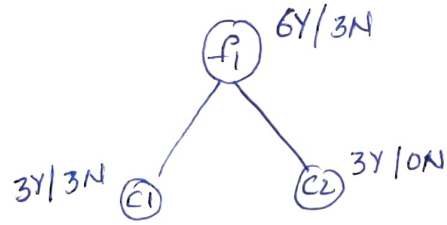
② How the features are selected ? ⟹ Information Gain

p^+ = probability of YES

p^- = probability of NO

Entropy

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$



C2

$$= -\frac{3}{3} \log_2 \frac{3}{3} - \frac{0}{3} \log_2 \frac{0}{3}$$

$$= -1 \log_2 1 - 0$$

$$= 0 - 0$$

$$= 0 \Rightarrow \text{pure split}$$

C1

$$= -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6}$$

$$= 1 \Rightarrow \text{Impure split}$$

Gini Impurity

$$G.I = 1 - \sum_{i=1}^{n} (P_i)^2$$

$$= 1 - [(P_+)^2 + (P_-)^2]$$

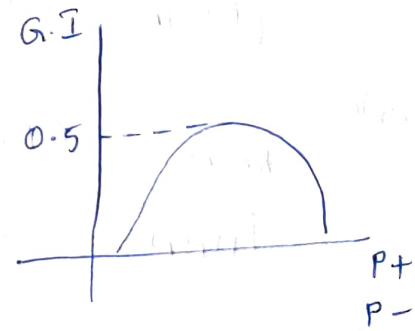
f1 2Y/2N

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

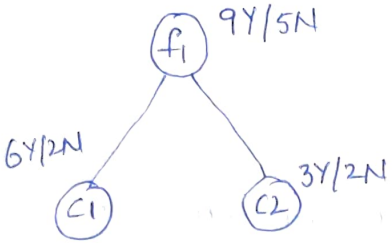
$$= 1 - \left(\frac{1}{4} + \frac{1}{4}\right)$$

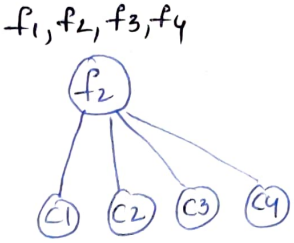
$$= 1 - 1/2$$

$$= 0.5$$



⊛ How to select which feature to split ?

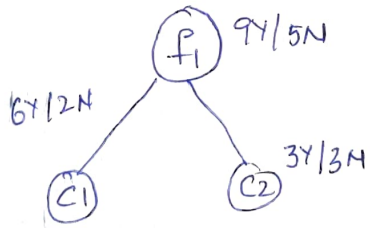




Information Gain

$$Gain(S, f_1) = H(S) - \sum_{v \in Val} \frac{|S_v|}{S} H(S_v)$$

$H(S)$ = Entropy of feature
$|S_v|$ = Sample at category
S = Total Sample
$H(S_v)$ = Entrope of Category



{f1}

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$
$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14}$$

$H(S_{f1}) = 0.94$

{C1}

$$H(C_1) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8}$$

$H(C_1) = 0.81$

{C2}

$$H(C_2) = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6}$$

$H(C_2) = 1$

$$Gain(S, f_1) = H(S_{f_1}) - \sum_{v \in Val} \frac{|S_v|}{S} H(S_v)$$

$$= 0.94 - \left[\frac{8}{14}(0.81) + \frac{6}{14}(1) \right]$$

$Gain(S, f_1) = 0.041$

if $Gain(S, f_2) = 0.051$

} which feature should start first

(*) Node which has more gain starts first.

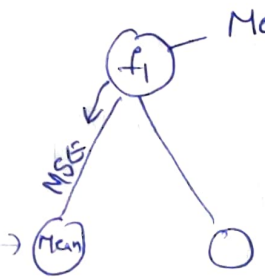
→ Entropy takes more time to calculate ondgorithm

→ Gini Impurity takes less time than entropy.

Decision Tree Regressor ↓

f_1	f_2	o/p
		=
		=
		=
		=

Mean



Mean = MSE

$$\Downarrow$$

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

Hyperparameters ↓

(*) Decision tree leads to overfit

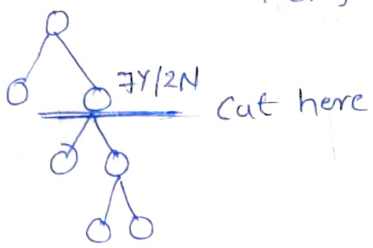
To avoid overfitting

(i) Post pruning

(ii) pre pruning

Post pruning :—

Here, for yes = 70, so cut tree here



Pre-pruning :—

max_depth, max_leaf ⇒ hyper parameters

Day 5 :—

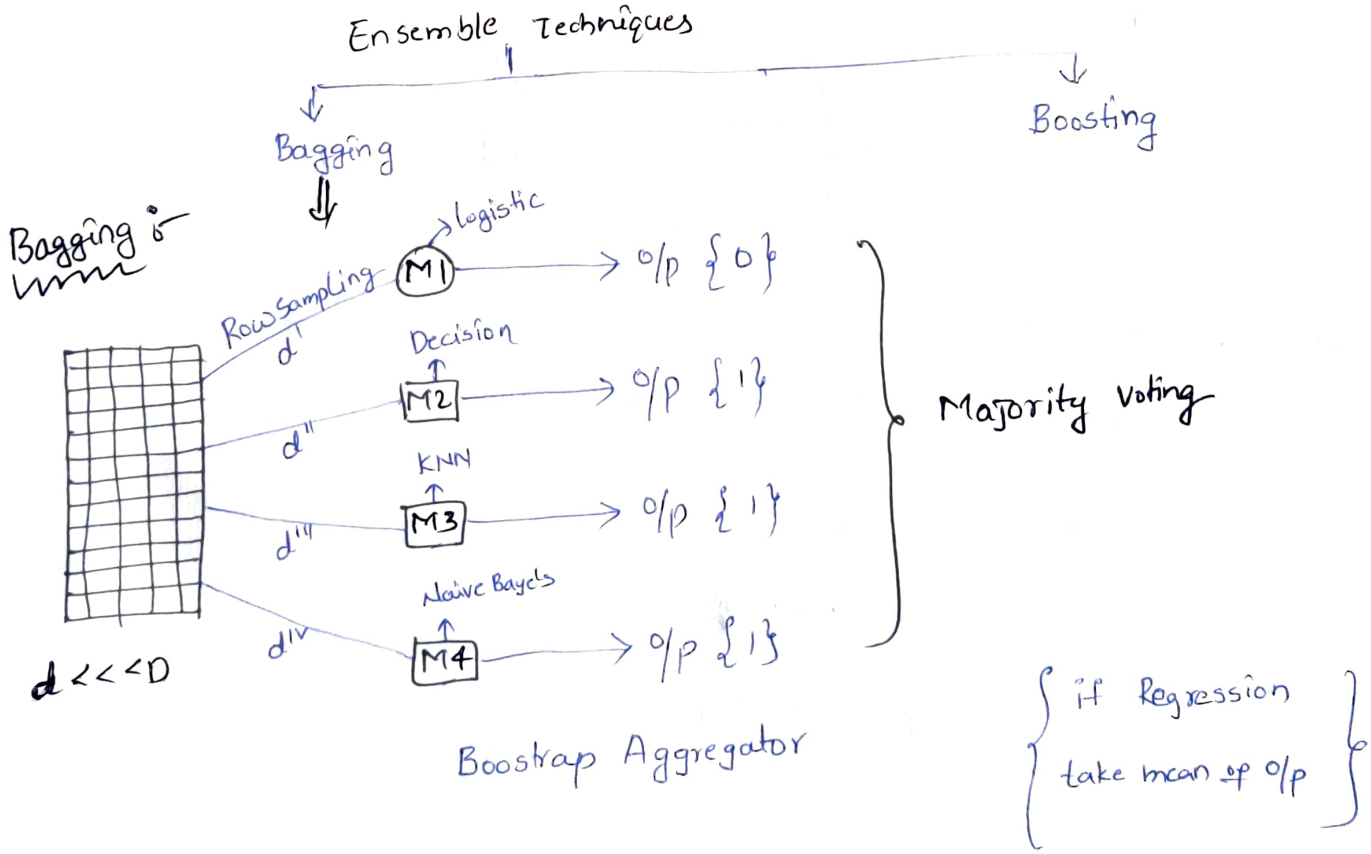
① Ensemble Techniques — Bagging, Boosting

② Random Forest

③ AdaBoost

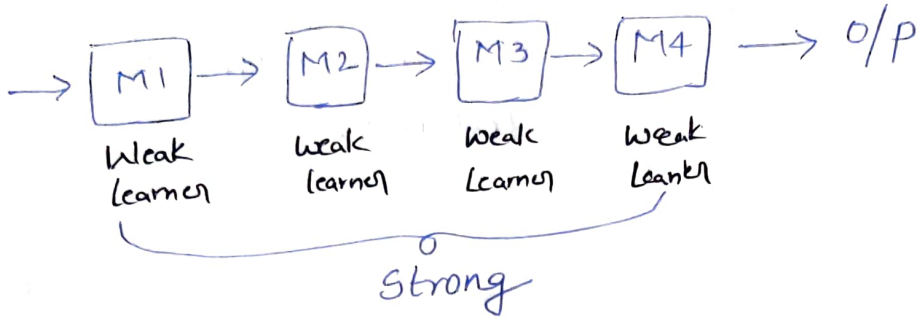
④ XGBoost.

Ensemble Techniques :—



Boosting :—

Series of models



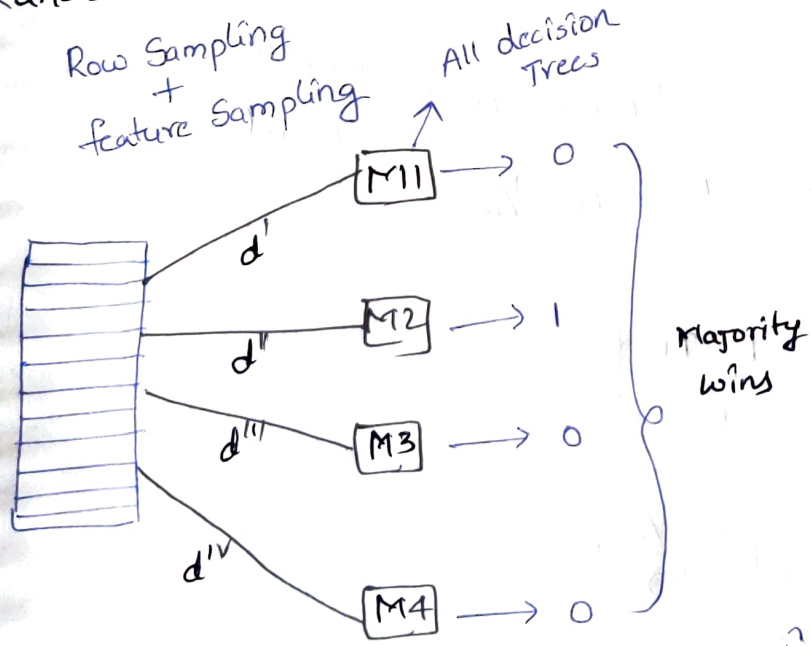
BAGGING

- Random Forest Classifier
- Random Forest Regressor

BOOSTING

- AdBoost
- Gradient Boost
- XGBoost

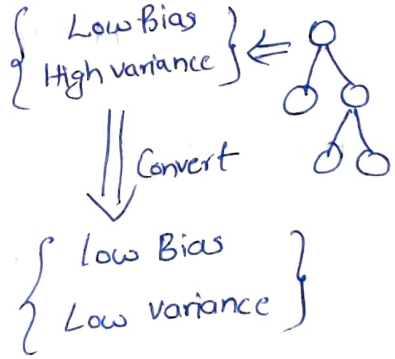
Random Forest classifier & Regression



What is problem of DT?

⇓

overfitting



(i) is Normalization required?

⇓

No in decision Tree.

is Normalization required in DT ? ⟹ No

is standardization required in KNN ? ⟹ Yes

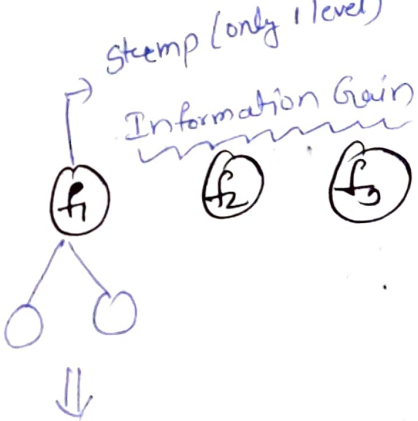
is random forest effect by outlier ? ⟹ No

↳ It gives equal opportunity to select features.

Boosting :

AdaBoost ← Sum of weight = 1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	O/P	weight
Total = 7								Y	1/7
								N	1/7
								N	1/7
								Y	1/7
								Y	1/7
								N	1/7



Pass all the data and find how many wrongly predicted

$$\text{Total Error} = TE = \frac{\text{wrongly}}{\text{Total}}$$

if suppose wrong = 1 ⟹ $TE = 1/7$

Performance of STUMP :

$$\frac{1}{2}\log_e\frac{1-TE}{TE}$$

$$P_S = \frac{1}{2}\log_e\left(\frac{1-1/7}{1/7}\right)$$

$$= 0.895$$

New sample weight

Correct = weight * e^{-P_S} ⟹ $\frac{1}{7} \times e^{-0.895} = 0.05$

Incorrect = weight * e^{P_S} ⟹ $\frac{1}{7} \times e^{0.895} = 0.349$

New weight

0.05
0.05
0.05
0.349
0.05
0.05
0.05
0.649

⇓

Here, sum of weight ≠ 1

So, Normalize by divide 0.649

Normalized weight

0.07
0.07
0.07
0.537
0.07
0.07
0.07

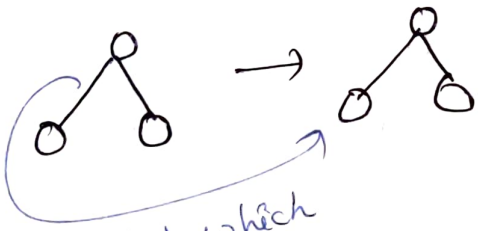
Buckets

[0 - 0.07]
[0.07 - 0.14]
[0.14 - 0.21]
[0.21 - 0.747]
[0.747 - 0.751]
[-]
[-]

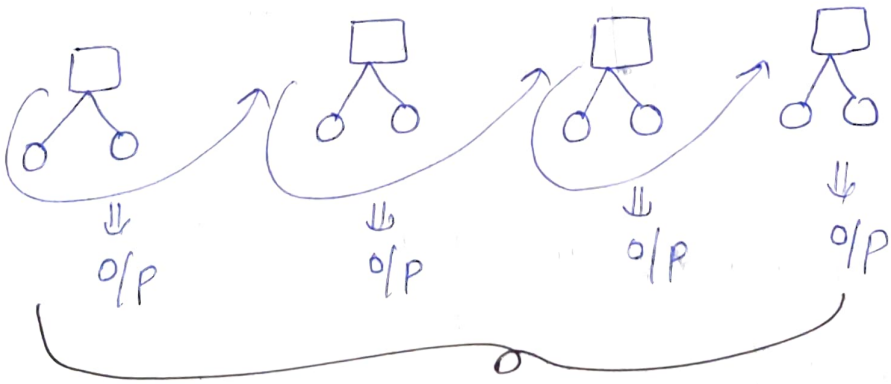
This having high wrong values

⇓

Passed to next node



Bucket which has more no. of wrongly predictions



Majority → classification

Mean → Regression

Linear Regression → White Box Model

Random Forest → Black Box model

Decision Tree → White Box model

ANN → Black Box model

Day-6 :-

Unsupervised ML

1. KMeans clustering
2. Hicrarical clustering
3. Silhoutte score
4. DBSCAN clustering

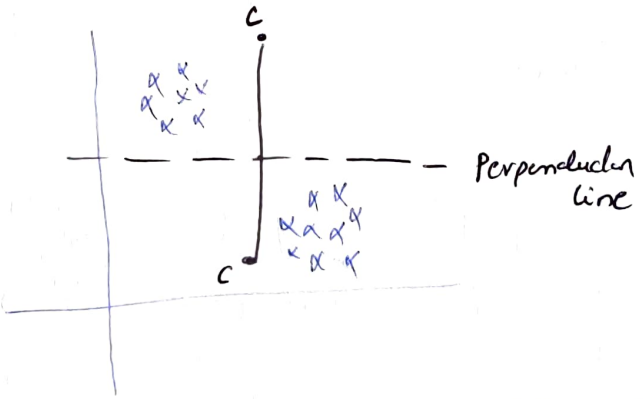
K-Means clustering :-

k-centroids

(i) we try with different k-values

(ii) Initialize k number of centroids

(iii) Compare avg to update centroids

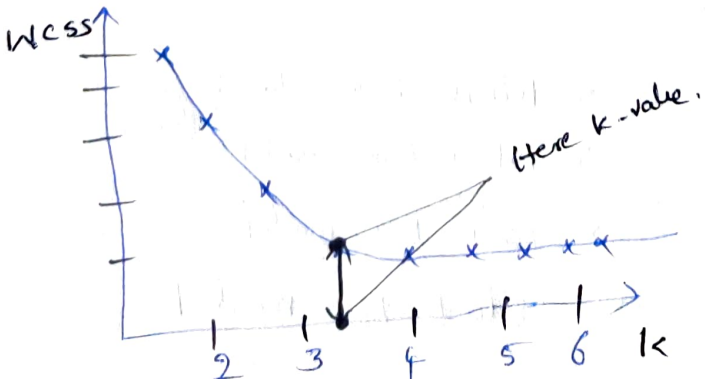


Note :- Initilize centrod very far ⟹ K-Means ++

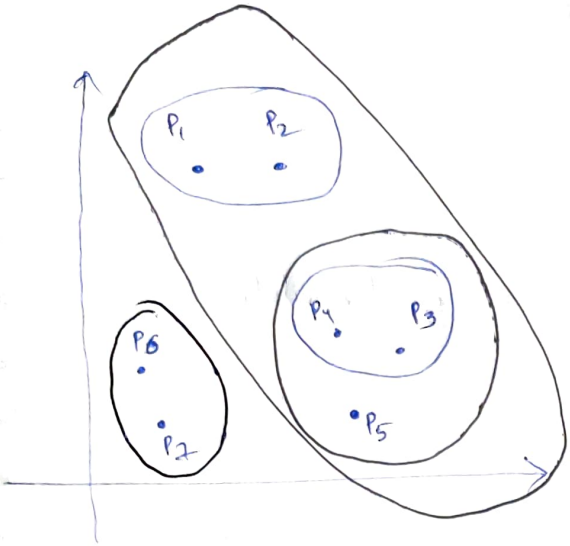
How to decide k-value ?

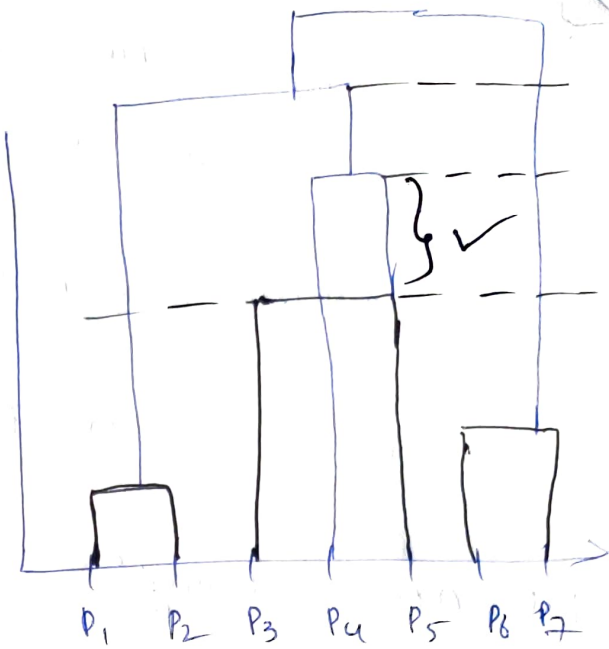
Elbow method

WCSS - Within cluster sum of squares



② Hierarical clustering :—





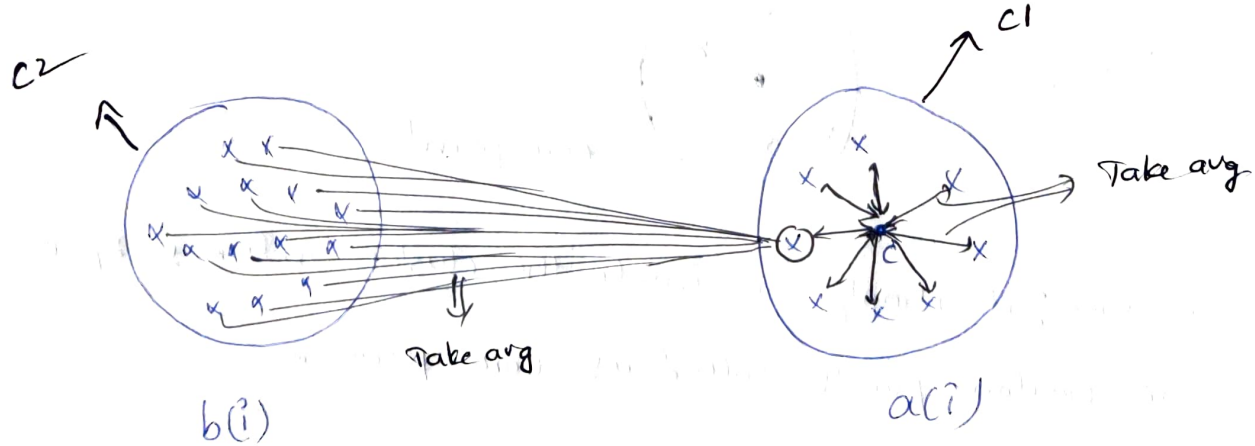
Dendogram

⊛ Find the largest vertical line that has no horizontal line pass through.

→ Maximum time taken by k-Means or Hierarical clustering ?

Sol⊳ Hierarical clustering

Silhouette Score(clustering) :—



$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad , \{if\ |ci| > 1\}$$

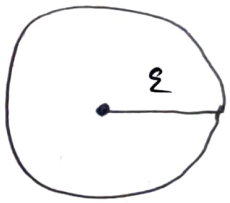
$$Values = \{-1 \text{ to } 1\}$$

if S(i) is towards +1 then it is more good model.

DBSCAN clustering :-

Density Based Spatial Clustering of Applications with Noise

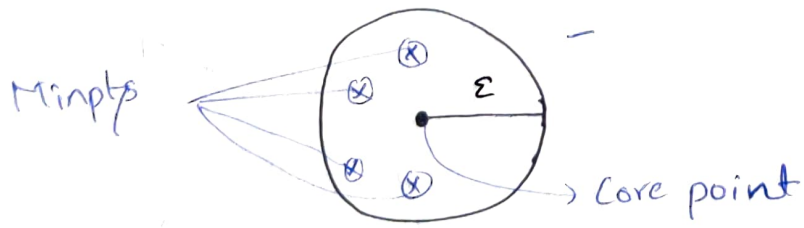
(i) Min points
(ii) Core point
(iii) Border point
iv) Noise point
v) Epsilon (ε)



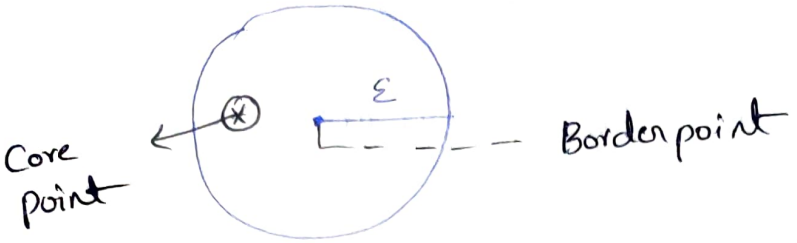
* DBSCAN used to avoid outlier or Noise point.

* Minpts is hyperparameter.

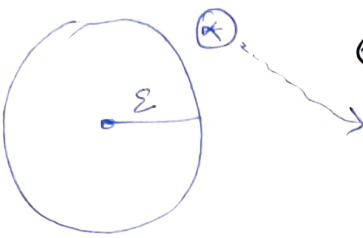
if Minpts = 4



Condition: if Minpts are inside the cluster radius of Epsilon the epsilon points called as Core point.



*) within only one core point, then it becomes border point.



⊛ No points inside

⊛ This is out side point, neglect

→ It is Noise point

Bias :-

It is phnomenon that skews the result of an algorithm in favour or against an idea.

Variance :-

The variance refers to the changes to then model when using different portions of training or test data.

Day - 7 :-

1. Xg boost classifier
2. Xg boost Regressor
3. SVM
4. SVR

Xgboost classifier :— Extreme Gradient Boosting

Salary	Credit	Approval	{Residual} (1-Pr)
<=50	B	0	-0.5
<=50	G	1	0.5
<=50	G	1	0.5
<=50	G	0	-0.5
~~<50~~	G	1	0.5
>50	N	1	0.5
>50k	N	0	0.5

Base Model ⟹ pr(0.5)

① Create a binary decision tree using feature

② Calculate Similarity weight λ = hyperparameter

$$= \frac{\Sigma(\text{Residual})^2}{\Sigma(Pr(1-pr)) + \lambda}$$

③ Information Gain

Solⁿ

[-0.5, 0.5, 0.5, -0.5, 0.5, 0.5, 0.5]

Salary

<=50k

[-0.5, 0.5, 0.5, -0.5]

Assume λ = 0

$$Sim(\omega) = \frac{(-0.5+0.5+0.5-0.5)^2}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0} = \frac{0}{Any}$$

Sim(ω) = 0

>50

[-0.5, 0.5, 0.5]

$$Sim(\omega) = \frac{(-0.5+0.5+0.5)^2}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0}$$

$$= \frac{0.25}{0.75} = 0.33$$

Sim(ω) = 0.33

$$\frac{-0.5+0.5+0.5-0.5+0.5+0.5+0.5}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)+0}$$

$$= \frac{0.25}{1.75} = 0.142$$

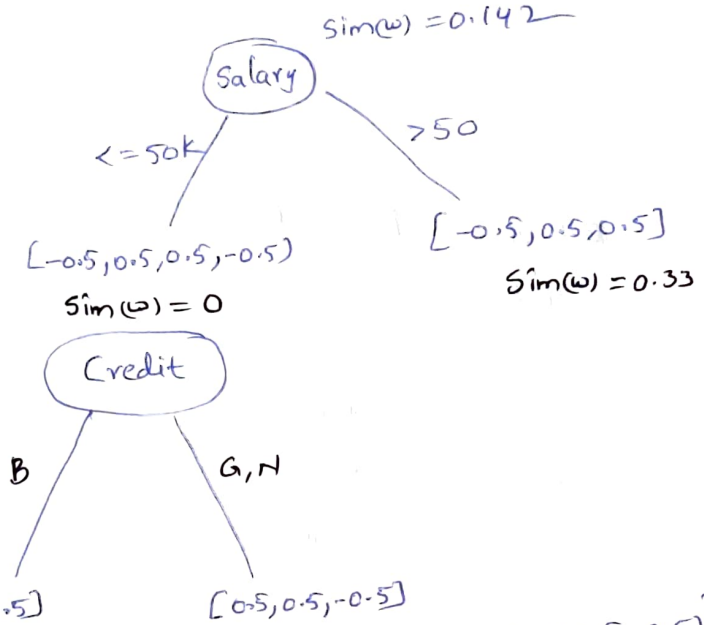
Sim(ω) = 0.142

Information Gain = (Sum of sim(ω) of branch nodes) − sim(ω) of Root Node

= 0 + 0.33 − 0.14

= 0.19

Divide again above DT



$$Sim(\omega) = \frac{(-0.5)^2}{0.5(1-0.5)+0} = \frac{0.25}{0.25} = 1$$

Sim(ω) = 1

$$Sim(\omega) = \frac{(0.5+0.5-0.5)^2}{0.5(1-0.5)+0.5(1-0.5)+0.5(1-0.5)} = \frac{0.25}{0.25+0.25+0.25} = \frac{0.25}{0.75}$$

Sim(ω) = 0.33

G.I = 1 + 0.33 − 0

G.I = 2.33

for base model $\log\left(\frac{p}{p-p}\right) = \log\frac{0.5}{1-0.5} = 0$

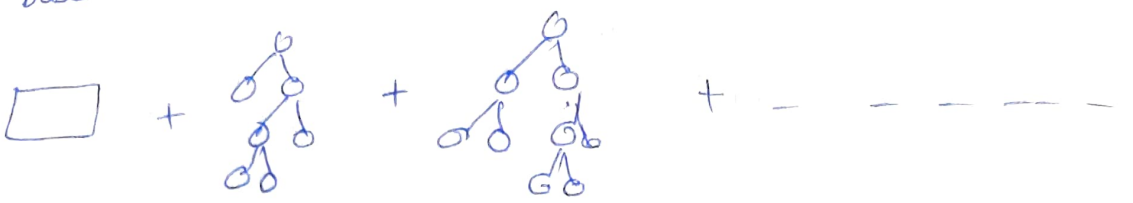
To find real probability

$$\sigma\left[0 + \alpha(\text{sim}(\omega) \text{ of leaf Node})\right]$$

$$\sigma\left[0 + \alpha_1(DT_1) + \alpha_2(DT_2) + \alpha_3(DT_3) + \dots + \alpha_n(DT_n)\right]$$

* Xgboost is also blackbox model.

Base + $\alpha_1(DT_1)$ + $\alpha_2(DT_2)$ + - - - - + $\alpha_n DT_n$



XG Regressor :—

Exp	Gap	Salary	Residual
2	Yes	40K	-11K
2.5	Yes	42K	-9K
3	No	52K	1K
4	No	60K	9K
4.5	Yes	62K	11K

⇓

Base → avg of o/p = 51K

$$Sim(w) = \frac{\sum(Residuals)^2}{No.\ of\ residuals + \lambda}$$

Exp

[-11K, 9K, 1, 9, 11]

≤ 2

[-11]

$Sim(w) = \frac{121}{1+0}$

$Sim(w) = 121$

> 2

[-9, 1, 9, 11]

$Sim(w) = \frac{(-9+1+9+11)^2}{4+0}$

$= \frac{144}{4}$

$Sim(w) = 36$

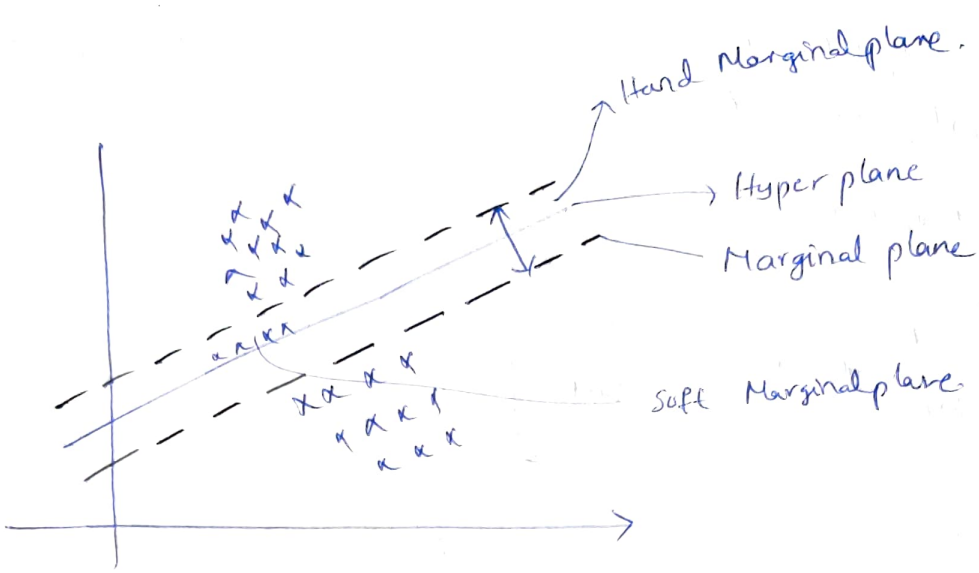
$Sim(w) = \frac{(-11-9+1+9+11)^2}{5+0}$

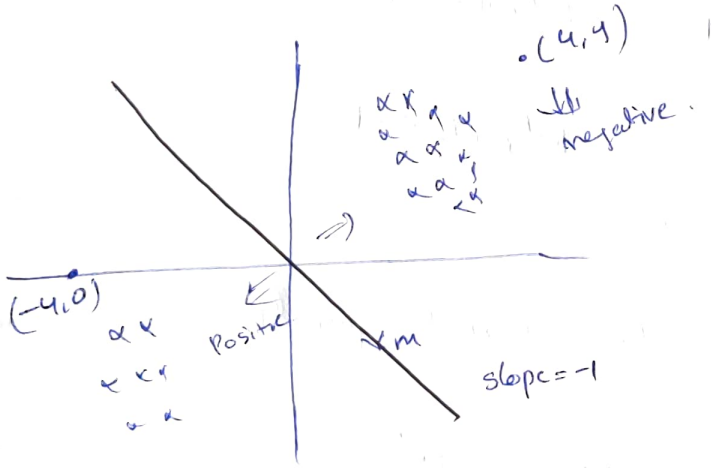
$Sim(w) = \frac{1}{5} = 0.2$

Information Gain = 121 + 36 − 0.2

= 156.8

SVM :-





$y = mx + c$

$y = w_1 x + w_2 x_2 + \dots + b$

$y = w^T x + b$

$(-4,0)$

$$y = \begin{bmatrix} -1 \\ 0 \end{bmatrix} [-4, 0]$$

$$= 4$$

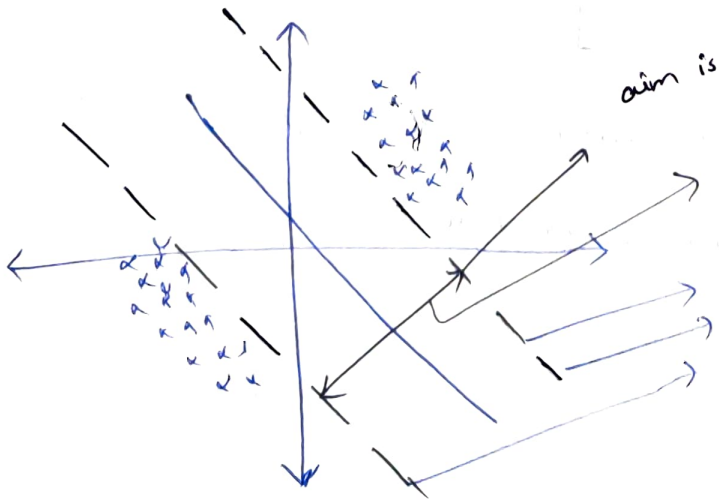
$\Rightarrow$ +ve values

$(4,4)$

$$y = \begin{bmatrix} -1 \\ 0 \end{bmatrix} [4, 4]$$

$$= -4$$

-ve value



aim is to increase or maximize distance

$$w^T(x_1 - x_2) = 2$$

$$w^T x + b = -1$$
$$w^T x + b = 0$$
$$w^T x + b = 0$$

$w^T x_1 + b = 1$

$w^T x_2 + b = -1$

$w^T (x_1 - x_2) = 2$ ⟶ divide by $|w|$ to remove magnitude.

$$\frac{w^T}{|w|}(x_1 - x_2) = \frac{2}{|w|}$$

Aim is to maximize $\frac{2}{|w|}$

Such that

$$y_i \begin{cases} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{cases}$$

Major aim

$y_i * (w^T x_i + b) \geq 1$ for Correct points

$$\underset{(w,b)}{\text{Maximize}} \frac{2}{|w|} \iff \underset{(w,b)}{\text{Min}} \frac{|w|}{2}$$

$$\underset{(w,b)}{\text{Min}} \frac{(|w|)}{2} + C_i \sum_{i=1}^{m} \xi_i$$

ξ_i = Summation of the distance of the wrong data points

C_i = How many errors we have