INDEX

NAME _____ SUBJECT _____

STD. _____ DIV. _____ ROLL NO. _____ SCHOOL _____

SR. NO.	PAGE NO.	TITLE	DATE	TEACHER'S SIGN / REMARKS

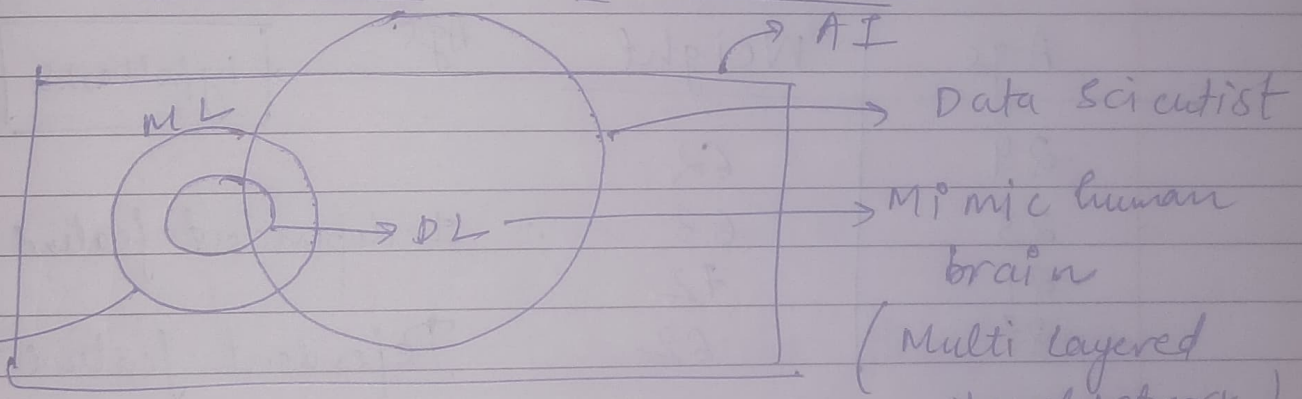
Sparsh Thariya

Machine Learning

Agenda

1. Introduction to ML (AI V/S ML V/S DL V/S DS)
2. Supervised ML and Unsupervised ML
3. Linear Regression (Maths & Geometric Intuition)
4. R^2 & Adjusted R^2
5. Ridge and Lasso Regression

AI V/S ML V/S DL V/S DS



AI application

AI application is able to do its own task without any human interventions.

Eg:- Netflix → Action → Recommendation
→ comedy → "

Amazon.in → Iphone → Headphones

Self driving Cars → Tesla

Machine & Deep learning → [Reinforcement]

- Supervised ML
  - Regression
  - Classification
- Unsupervised ML → CLUSTERING
  - → DIMENSIONALITY REDUCTION

Supervised ML

Age	Weight
24	62
25	63
21	72
27	62

Age → [hypothesis] → o/p weight

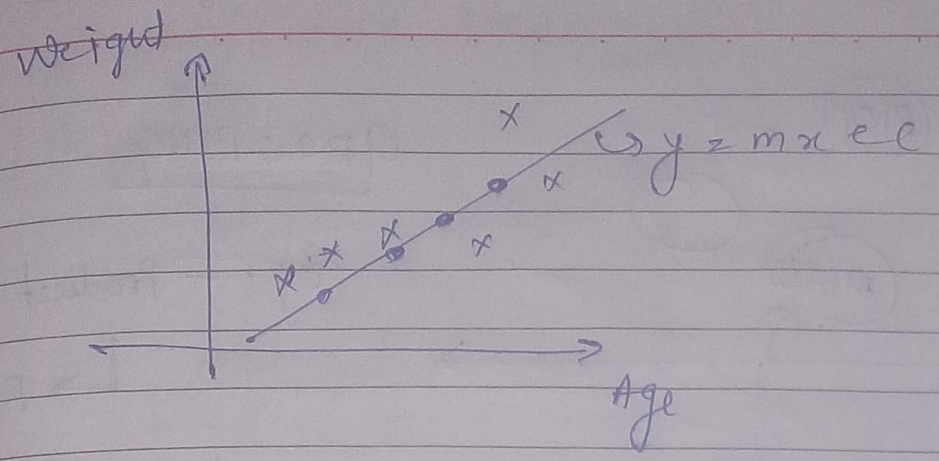
Independent feature → Age

Dependent feature → Weight

① Regression Problem

Age	Weight
24	72
23	71
25	71.5
-	-

o/p → continous variable ⇒ Regression Problem



② CLASSIFICATION

No. of hours	No. of play hours	No. of sleep	O/P Pass / Fail
—	—	—	P
—	—	—	P
—	—	—	F
—	—	—	P
—	—	—	F

Two O/P → Binary Classification

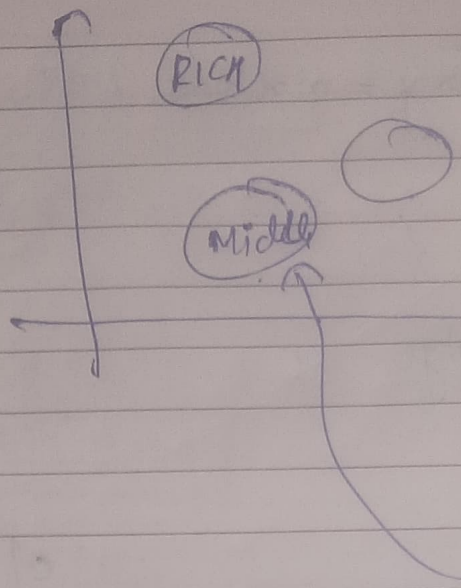
More than Two O/P → Multi class classification

Unsupervised ML → Clustering

→ Dimentionality Reduction

Salary | Age } No Depend Variable

Clustering → Customer segmentation



GROUPING

Eg:- Product 1
↳ Rich

Product 2
↳ Middle

Customer Segmentation

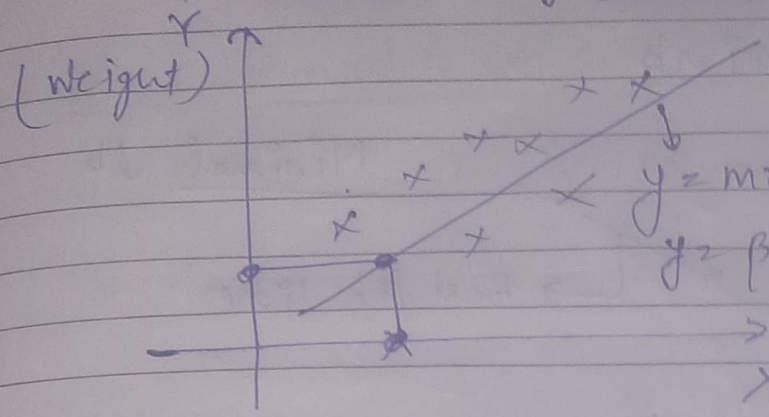
Dimentionality Reduction

Two → lower Dimension
↳ one

PCA, LDA

Supervised	Unsupervised
(1) Linear Regression	(1) K Means
(2) Ridge & Lasso	(2) DB Scan
(3) Logistic Reg	(3) Heirarchical
(4) Decision Tree	(4) K-Nearest Neighbor Classifier
(5) Adaboost	(5) PCA
(6) Random forest	(6) LDA
(7) Xg boost	
(8) KNN	
(9) Naive Bayes	
(10) SVM	

Linear Regression



$y = mx + c$

$y = \beta_0 + \beta_1 x$

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

y is alt. linear function of x

[TRAINING DATASET]

↓

[Model]

↓

New Age → [Hypothesis] → O/P weight

Equation of a straight line

$y = mx + c$

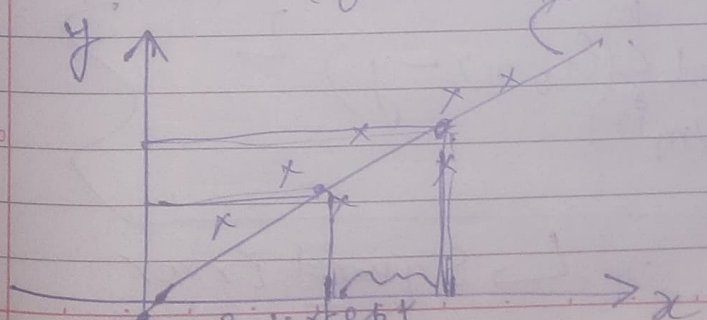
$y = \beta_0 + \beta_1 x_i$

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

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

When $x = 0$

$h_\theta(0) = \theta_0$

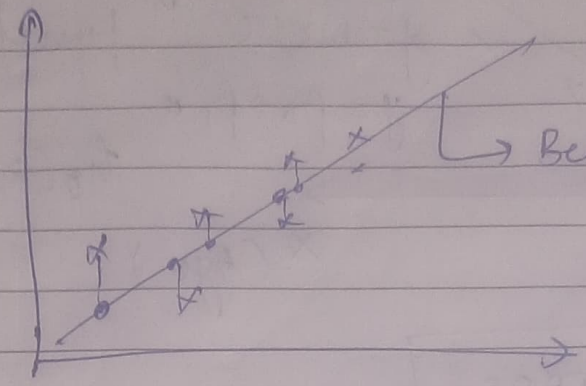


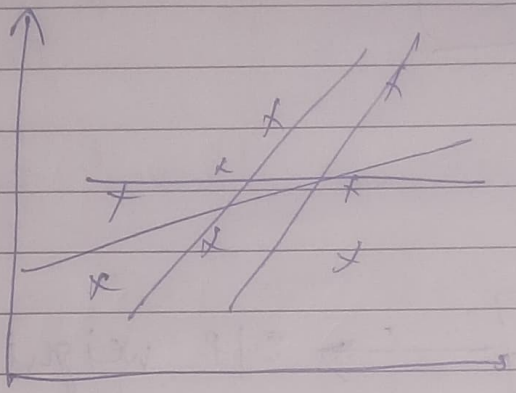
θ_0 = Intercept

θ_1 = slope or coefficient

x_i = data points

Minimal J(θ)





Start at point
↓
best fit line

Hypothesis

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

cost function

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

Cost fun^{cn} = $\frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x) - y)^2$

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

↳ Squared Error Function

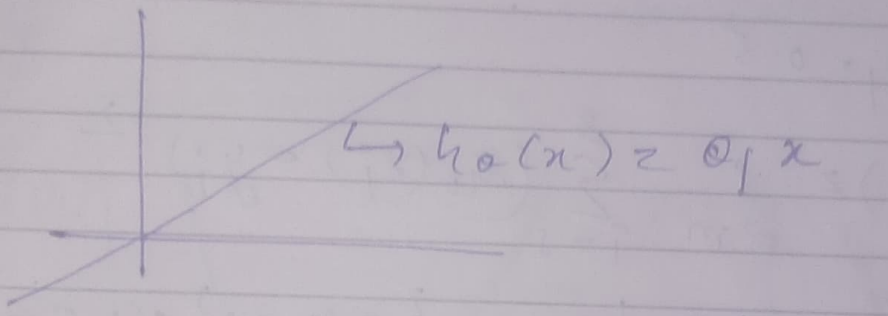
What we need to solve

$$\text{minimize}_{(\theta_0, \theta_1)} \quad \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

$$\Downarrow$$

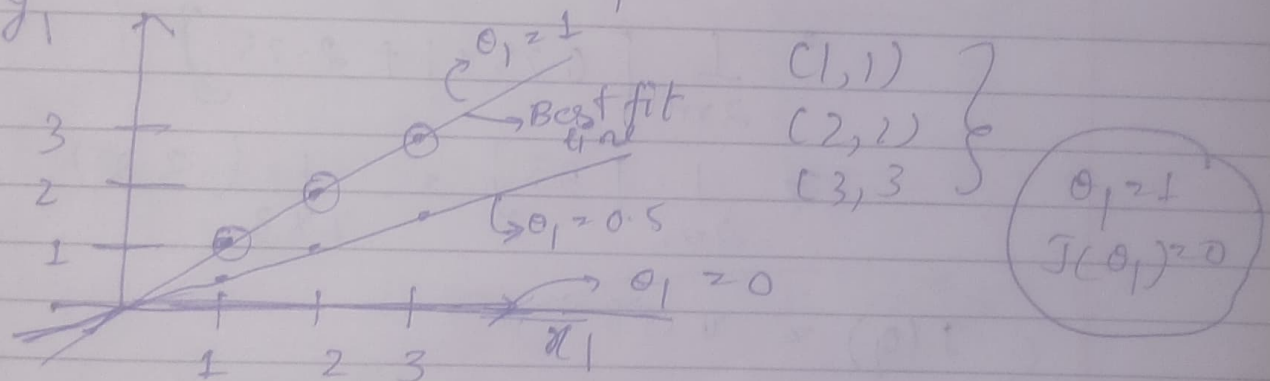
$$\text{minimize}_{\theta_0, \theta_1} \quad J(\theta_0, \theta_1)$$

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



$h_\theta(x) = \theta_1 x$ → slope

$\theta_1 = 1$

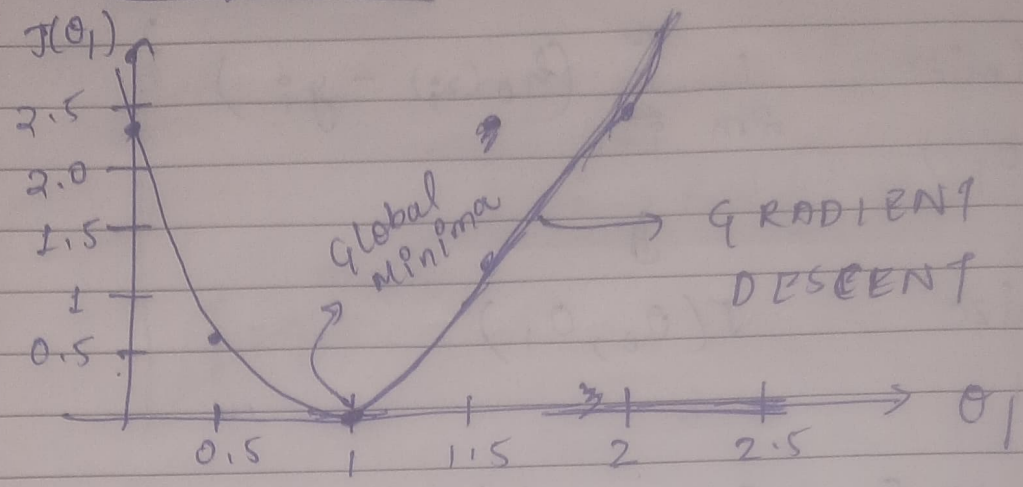


$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} (h_\theta(x_i) - y_i)^2$$

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

$$J(\theta_1) = 0$$

Cost function



let $\theta_1 = 0.5$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} (h_\theta(x_i) - y_i)^2$$

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

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

$$= \frac{1}{2m} \left[3.5 \right] = \frac{1}{2 \times 3} (3.5)$$

$$J(\theta_1) = 0.58$$

let $\theta_1 = 0$

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

$$= \frac{1}{6} \left[1 + 4 + 9 \right] = \frac{14}{6} = 2.3$$

Convergence Algorithm

Repeat until convergence {

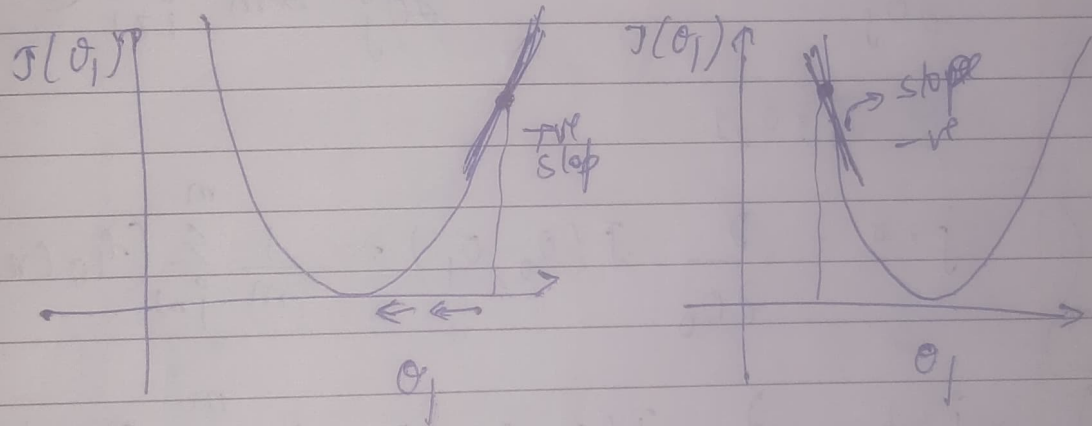
$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}$$

$\alpha \rightarrow$ learning rate

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

↓ derivative

}

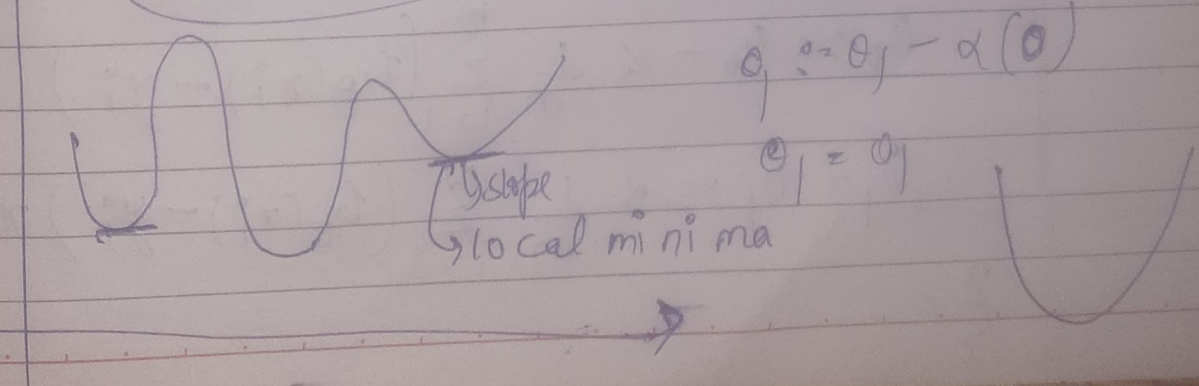


$\theta_1 := \theta_1 - \alpha(+ve)$

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

$\hookrightarrow \theta_1 + \alpha(+ve)$

$\alpha \rightarrow$ small



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

$\theta_1 = \theta_1$

GRADIENT DESCENT ALGORITHM

repeat until convergence

{

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

}

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$j = 0, 1$

Convergence algorithm

$j = 0$ $\quad \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$

$j = 1$ $\quad \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) x_i$

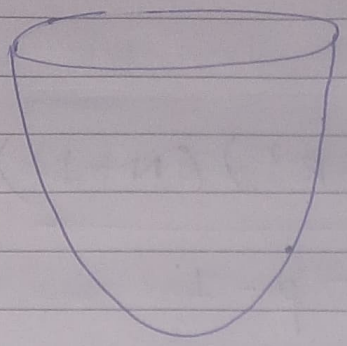
Repeat until convergence

{

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

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

}

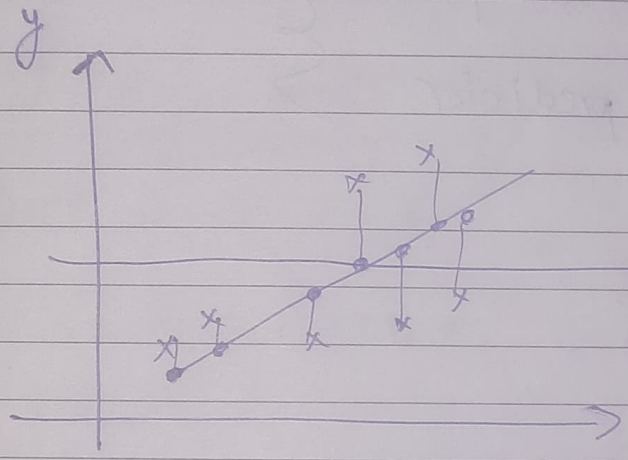


← multi variable

Performance Metrics

R^2 and Adjusted R^2

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



~~High~~

= 1 − [low / High]

small number

Gender — Bedrooms — Price — Location

No correlation

R^2 → 85%

R^2 = 90% ✓

R^2 = 91%

Adjusted R^2

$$R^2_{adjusted} = 1 - \left[\frac{(1-R^2)(N-1)}{N-p-1}\right]$$

p = features or predictors

$p = 2 \Rightarrow R^2 = 90\%$ $R^2_{adjusted} = 86\%$

$p = 3 \Rightarrow R^2 = 91\%$ $R^2_{adjusted} = 82\%$

N = No. of data points

p = No. of predictor

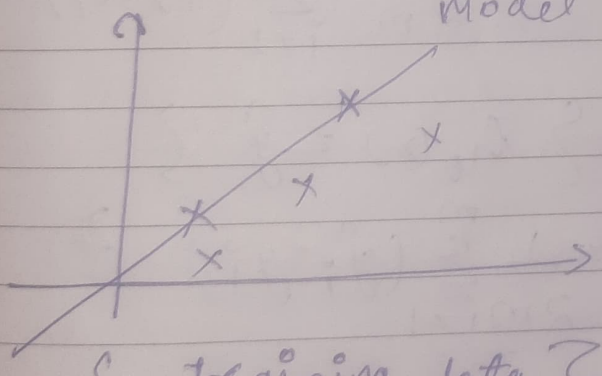
Agenda

1. Ridge and Lasso Regression
2. Assumption of Linear Regression
3. Logistic Regression
4. Confusion Matrix
5. ~~Practicals for Linear, Ridge, Lasso & Logistics~~ Practicals Implementation

Ridge And Lasso Regression

Cost function of Linear Regression $= \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$

$J(\theta_0, \theta_1)$



$\theta_0 = 0$

$J(\theta_0, \theta_1) = 0$ ↓↓

{ Training data }

Overfitting (Low Bias)

Model performs well ↓ training data

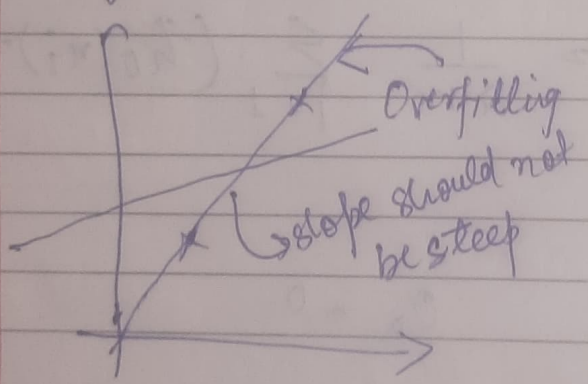
fails to perform well → Test data (High variance)

Underfitting { High bias, High variance }

① Model fails Accuracy is bad with Training data

② Model Accuracy is also bad with Test data

Model 1	Model 2	Model 3
Training Acc = 90%	Training Acc = 92%	Training Acc = 7%
Test Acc = 80%	Test Acc = 91%	Test Acc = 6.5%
⇓	⇓	⇓
Overfitting	Generalised Model	Underfitting
Low Bias	Low Bias	High Bias
High Variance	low Variance	High Variance



$$J(\theta_1) = 0$$

$$= \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

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

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

$$= 0$$

Ridge (L2 Regularization)

$$J(\theta_1) = (\hat{y}_i - y_i)^2 + \lambda (\text{slope})^2$$

Let $\lambda = 1 \quad \theta_1 = 2$

$$= 0 + 1(2)^2 = 4 \text{ J J}$$

$$J(\theta_1) = (\hat{y}_i - y_i)^2 + \lambda (\text{slope})^2$$

$$\downarrow$$

$$(\text{small value}) + 1 (1.5)^2$$

$$< 3$$

** { prevent overfitting }

$\lambda \rightarrow$ Hyperparameter

Lasso (L1 Regularization)

$$J(\theta_1) = (\hat{y}_i - y_i)^2 + \lambda |\text{slope}|$$

↳ feature selection

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

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

① Preventing Overfitting } → Lasso (L1 regularization)
② ~~Feature~~ Feature selection }

{ λ → found out through cross-validation }

Ridge Regression (L2 Norm)

Cost function = $(h_\theta(x_i) - y_i)^2 + \lambda(\text{slope})^2$

Purpose: Preventing overfitting

Lasso Regression (L1 Reg)

Cost function = $(h_\theta(x_i) - y_i)^2 + \lambda|\text{slope}|$

Purpose: 1) Prevent overfitting }
2) Feature selection }

Assumption of Linear Regression

① Normal / Gaussian Distribution → Model will get trained well

② ~~Standardization~~ Standardization } Scaling data { → z-score

$\mu = 0, \sigma = 1$

③ Linearity

④ Multi collinearity — 95% correlated

x_1 | x_2 x_3 | Y

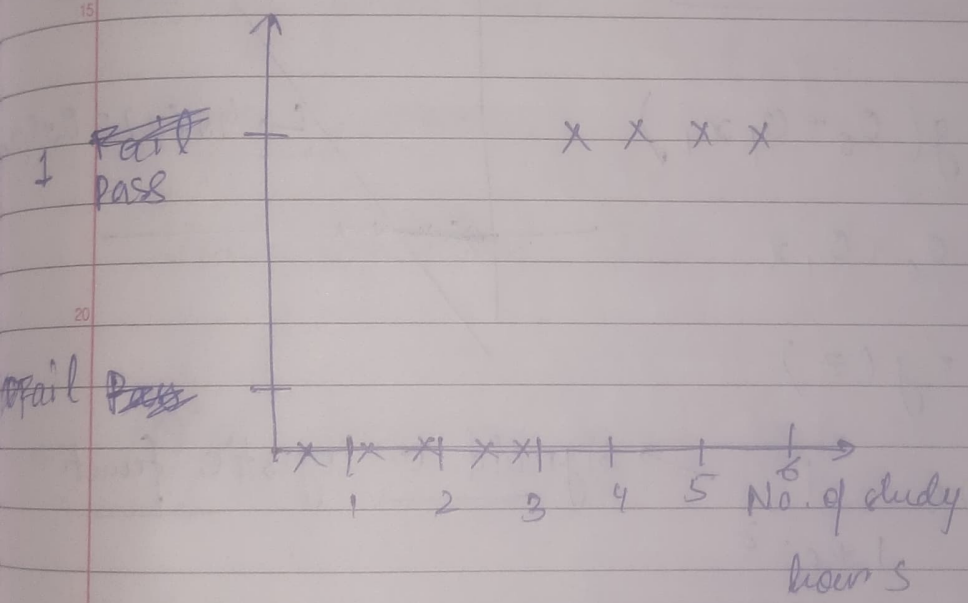
Variation Inflation factor?

Logistic Regression (Classification)

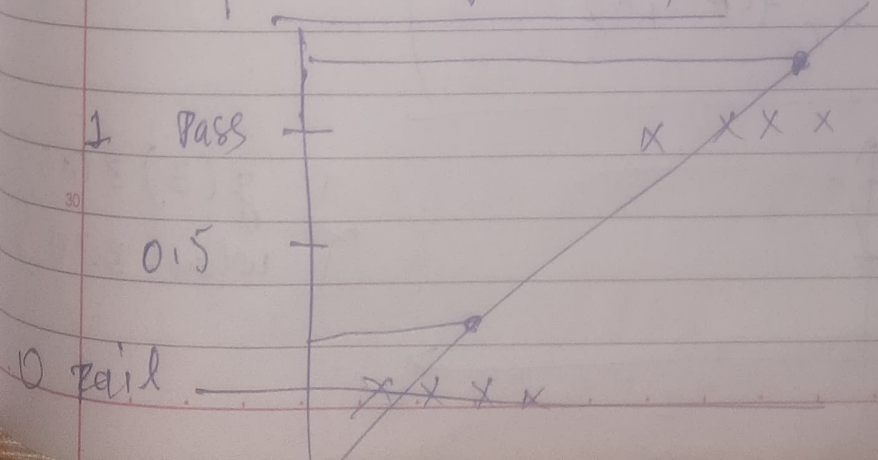
No. of study	No. of Play	P/F
—	—	P
—	—	F

→ Binary classification

→ Multi class classification



Linear Regression??



$h_\theta(x) \leq 0.5$

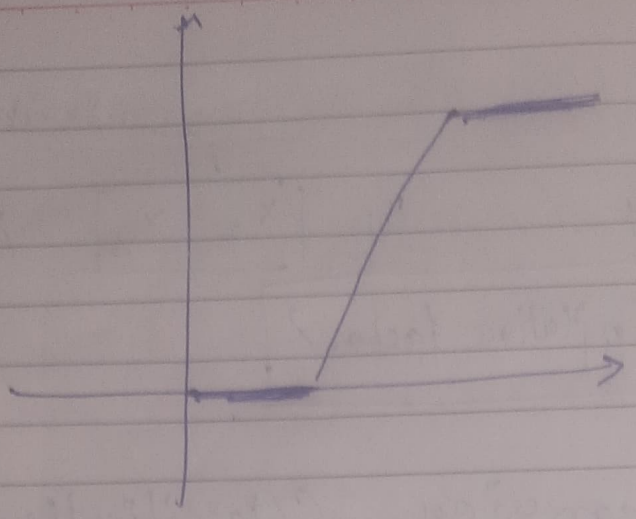
$\Downarrow$

0 → fail

$h_\theta(x) \geq 0.5 \Rightarrow 1$

$\Downarrow$

Pass

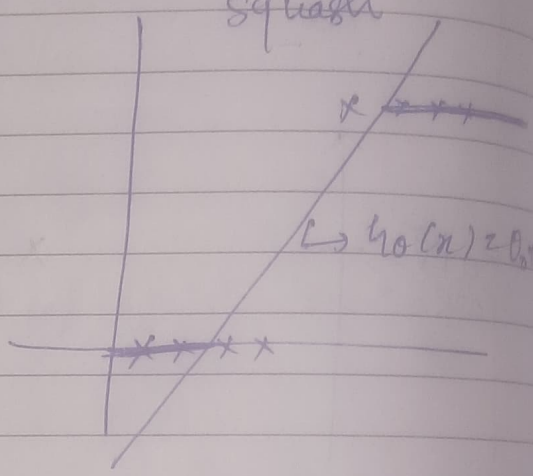


{ Sigmoid function
[0 to 1]

Decision Boundary Logistic Regression

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

$$\boxed{h_\theta(x) = \theta^T x}$$



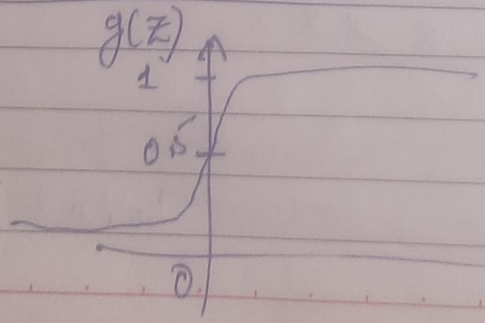
$h_\theta(x) = g(\theta_0 + \theta_1 x_1)$

Let $z = \theta_0 + \theta_1 x$

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

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

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



{ $g(z) \geq 0.5$ when $z \geq 0$ }

Training set

$\{(x_1, y_1), (x_2, y_2), (x_3, y_3) \cdots (x_n, y_n)\}$

$y \in \{0, 1\} \rightarrow$ 2 o/p

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

let $\theta_0 = 0$

$\rightarrow [z = \theta_1 x]$

Change parameter θ_1 ?

Cost function

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

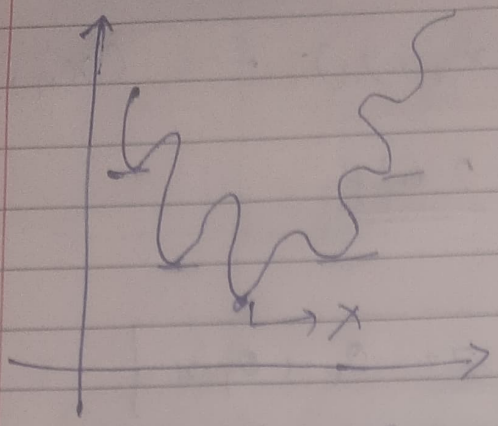
Logistic Regression $\quad [h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}]$

Logistic Regression Cost function $= \frac{1}{2} (h_\theta x_i - y_i)^2$ } we cannot use this cost function for logistic

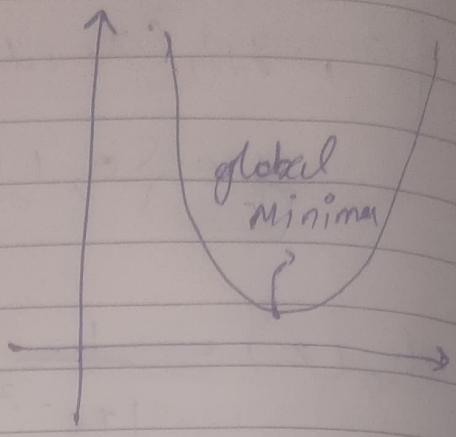
$h_\theta(x) = \frac{1}{1+e^{-\theta_1 x}}$

Non Convex function

Non Convex function



Convex function



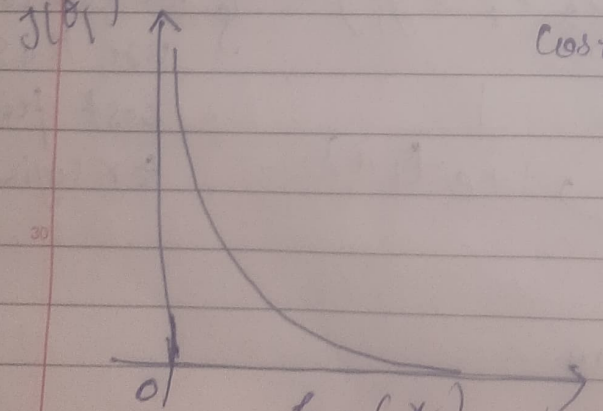
Logistic Regression cost function

~~Cost $(h_\theta(x^i), y^i)$~~

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

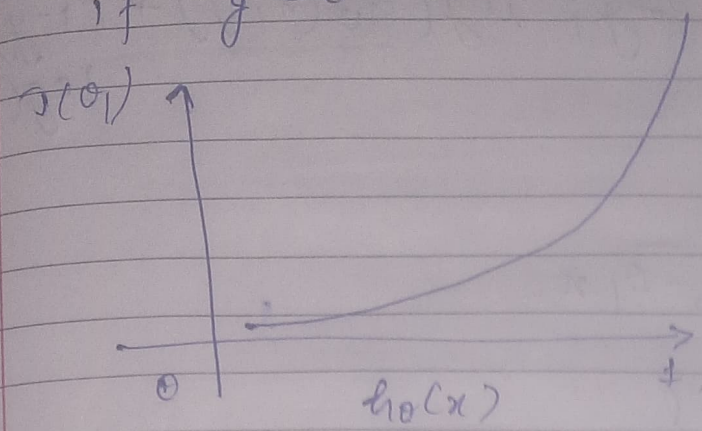
$$J(\theta_1) = \begin{cases} -\log(h_\theta(x^i)) & y = 1 \\ -\log(1 - h_\theta(x^i)) & y = 0 \end{cases}$$

if $y = 1$



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

if $y=0$



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

$$\boxed{\text{cost}(h_\theta(x_i), y) = -y \log(h_\theta(x_i)) - (1-y)\log(1-h_\theta(x))}$$

↓ cost function

if $y=1$

$$\text{cost}(h_\theta(x_i), y) = -\log(h_\theta(x_i))$$

if $y=0$

$$\text{cost}(h_\theta x_i, y) = -\log(1-h_\theta(x_i))$$

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

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

~~Conver~~ repeat until

{

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

}

Performance Metrics § Classification Problem §

x_1	x_2	y (Actual)	$\hat{y}$ (O/P)
-	-	0	1
-	-	1	1
-	-	0	0
-	-	1	1
-	-	1	1
-	-	0	1
-	-	1	0

§ Predictions §

Actual / Predicted

	1	0
1	2	2
0	1	1

Predicted / Actual

	1	0
1	TP	FP
0	FN	TN

§ Confusion Matrix §

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

$= \frac{3+1}{3+2+1+1} = \frac{4}{7} = 0.57$

57%

0 → 90%
1 → 10%
} Imbalanced data / Biased

0 → 60%
1 → 40%
} Balanced data

(1) Precision $\frac{TP}{TP+FP}$

(2) Recall (Sensitivity) $\frac{TP}{TP+FN}$

(3) F-Score

Pred \ Actual	1	0
1	TP	FP
0	FN	TN

{ Spam Classification } → Precision

{ HAS CANCER OR NOT } → Recall

{ Tomm. Stock Market is going to crash } → Recall / Precission

F-Beta → $(1+\beta^2)\frac{\text{Precision} \times \text{Recal}}{\beta^2 \times (\text{Precision} + \text{Recall})}$

$\beta = 1$: $(1+1)\frac{\text{Precision} \times \text{Recal}}{\text{Precision} + \text{Recall}}$

(F1-score)

$= 2\frac{(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$

Harmonic Mean = $\frac{2xy}{x+y}$

$\beta = 0.5$ $(1+(0.5)^2) \frac{P \times R}{(0.25)P + R}$

(F0.5-score)

$\beta = 2$ $(1+2^2)\frac{P \times R}{4P + R}$

(F2-score)

F1 score

Agenda

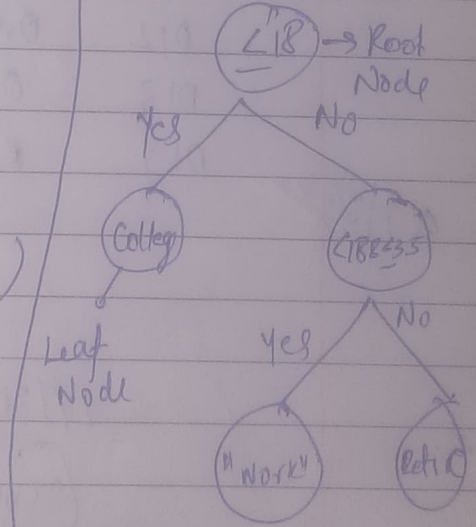
1. Decision Tree Classification
2. Decision Tree Regression
3. Practical Implementation
4. Ensemble Techniques

Decision Tree

- Regression
- Classification

```
if (age <= 18):
    print ("College")
elif (age <= 35 and age > 18)
    Print ("Work")
else
    Print ("Retire")
```

Decision Tree

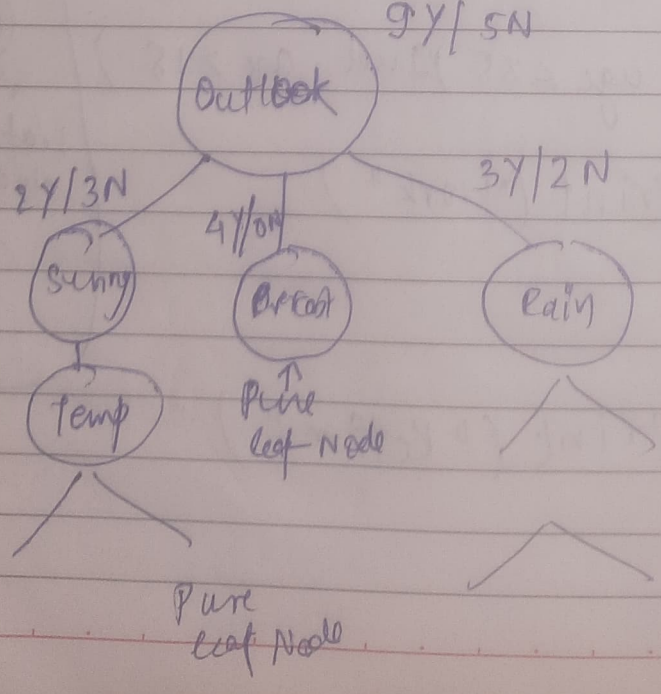


Regression & Classification ??

DECISION TREE

Nest if else → Decision tree

Day	Outlook	Temperature	Humidity	Wind	Play 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	Strong	No
D7	Overcast	Cool	Normal	Strong	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



Pure split ✓

Impure split ✓

① Purity → Pure split ??

→ Entropy

→ Gini coefficient / Impurity

② How the features are selected

→ Information Gain ?

Entropy

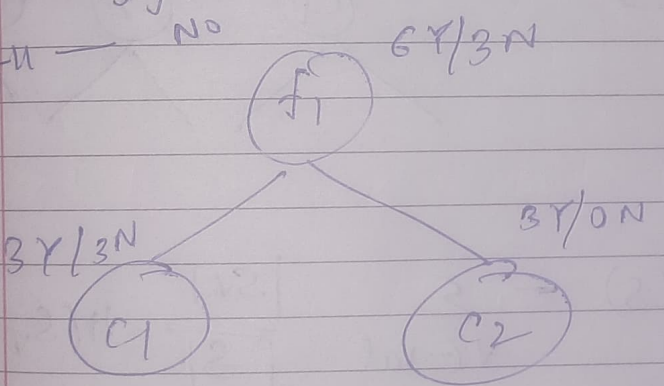
$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

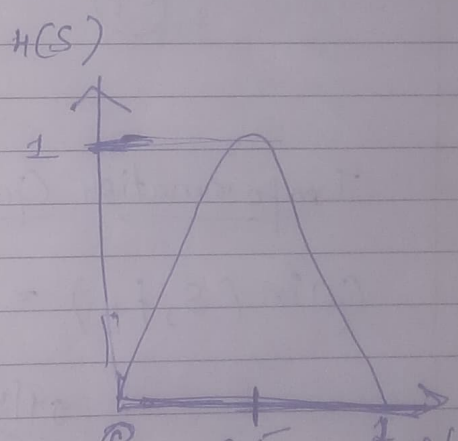
P_+ = probability of yes

P_- = probability of No

Gini Impurity

$$GI = 1 - \sum_{i=1}^{n} (P)^2$$





Entropy $H(S) = -\frac{3}{3} \log_2 \frac{3}{3} - \frac{0}{3} \log_2 \frac{0}{3}$ (C_2)

$H(S) = 0$

C_1 : $H(S) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}$

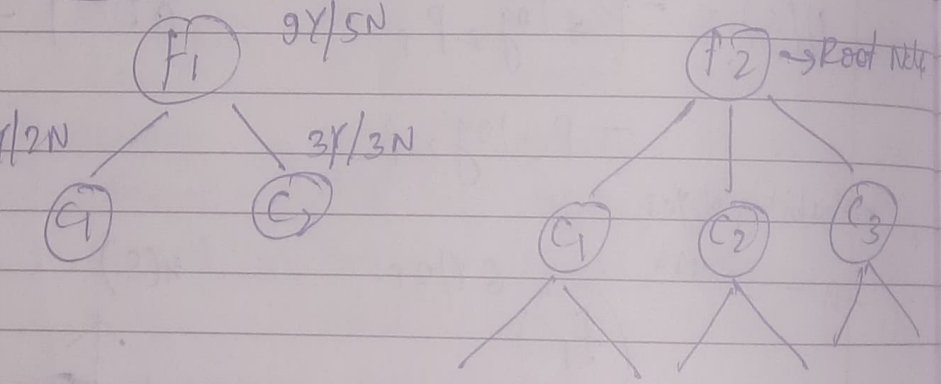
$$\boxed{H(S) = 1}$$

Purity Test → Entropy

$H(S) = 1$ Impure split → $H(S) \neq 0$

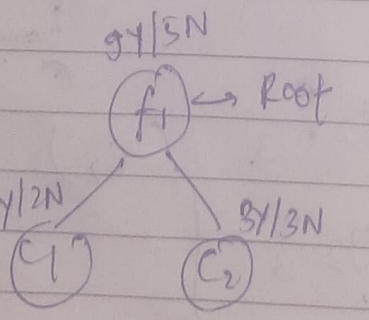
$H(S) = 0$ pure split

→ Which feature to take to split? f_1, f_2, f_3



Information Gain

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



$H(S) \rightarrow$ entropy

$$H(S) = -p_1 \log_2 p_1 - p - \log_2 (p-1)$$

$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log \frac{5}{14}$$

$$H(S) = 0.94$$

$$H(S_{V_1}) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8}$$

$$\boxed{H(S_{V_1}) = 0.8}$$

$$\boxed{H(S_{V_2}) = 1}$$

$$Gain(S, f_1) = 0.94 - \left[\frac{8}{14} \times 0.8 + \frac{6}{14} \times 1 \right]$$

$$\boxed{Gain(S, f_1) = 0.049}$$

$$Gain(S, f_2) = 0.05$$

using which feature should I start splitting first ?

f_2

as $Gain(S, f_2) > Gain(S, f_1)$

Gini Impurity

n = 2 output {Yes, No}

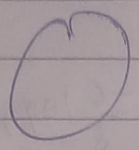
$$GI = 1 - \sum_{i=1}^{n} (p)^2$$

$$= 1 - \left[(p_1)^2 + (p_2)^2 \right]$$

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

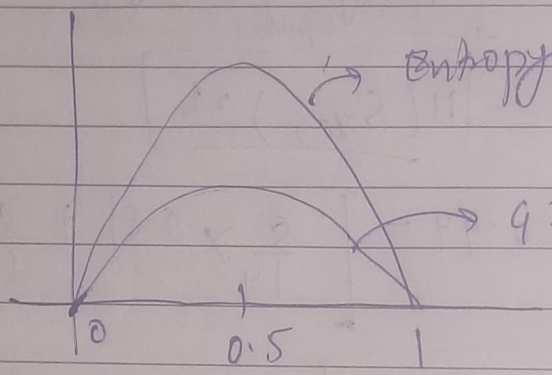
$= 1 - \left[\frac{1}{2}\right] = 0.5$

2Y/2N



→ Entropy = 1

GI = 0.5

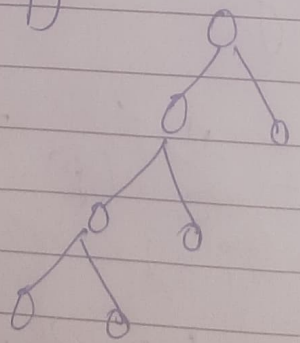


Entropy → log → More time

✓ GI → Simple maths → less time

Fast

Gini => Entropy

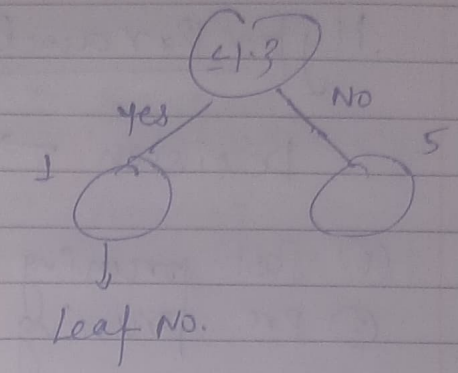


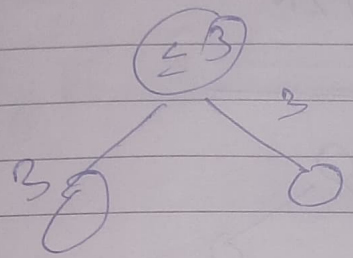
continous

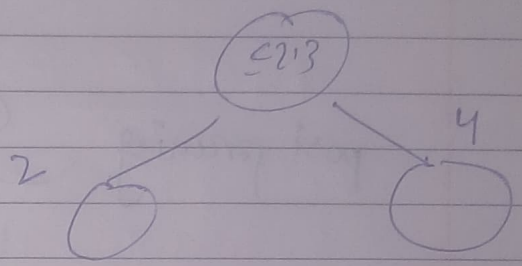
f_1	o/p
2.3	
1.3	
4	
5	
7	
3	

⇒

f_1
2
1.3
2.3
3
4
5
7



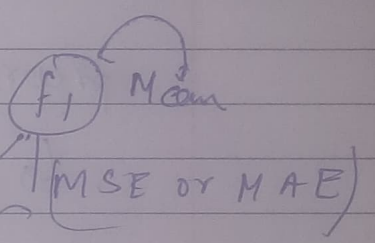




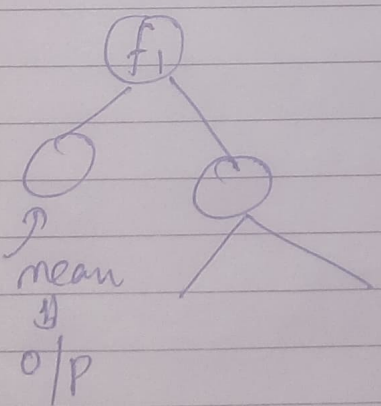
Decision tree Regressor

f_1 f_2 o/p

Continous o/p's → Mean



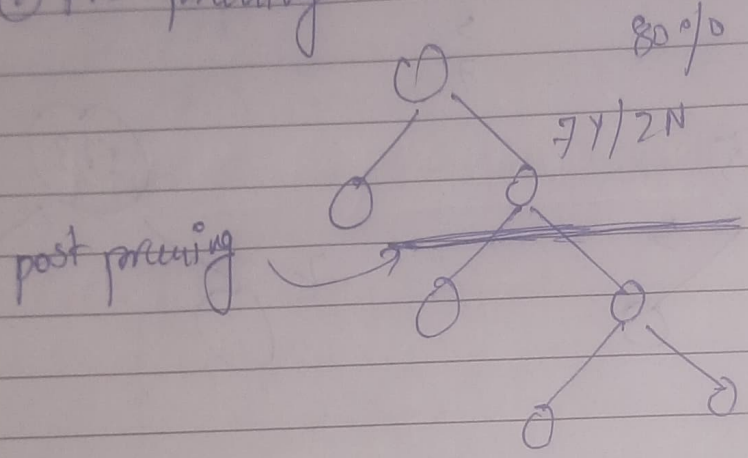
$$\frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_j)^2$$



Hyper Parameter

Decision → Overfitting

① Post pruning
② pre pruning



prepruning

↳ hyper parameter → Grid search - CV

max_depth, max_leaf

Agenda

1. Practicals (Simple Example)
2. Naive Baye's Intution
3. KNN algorithm

Previous Session

1. Linear Regression
2. Ridge & lasso
3. Logistic Regression

Naive Baye's Intution & Classification

⇓

BAYE'S THEOREM

Rolling a Dice

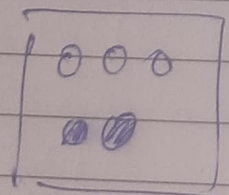
{1, 2, 3, 4, 5, 6}

$P(1) = \frac{1}{6}$ $P(3) = \frac{1}{6}$ {Independent Events}

$P(2) = \frac{1}{6}$

Dependent Event

0 - Red
0 - Green



First event

$P(R) = \frac{3}{5}$ → Red

Green Marble

$P(G) = \frac{2}{4} = \frac{1}{2}$

$P(R \text{ and } G) = P(R) * P(G/R)$

→ conditional probability

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

Camlin Page
Date / /

⇒ P(A and B) = P(B and A) ??? yes ✓

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

$$P\left(\frac{B}{A}\right) = \frac{P(B) * P(A/B)}{P(A')}$$

→ Baye's Theorem

Independent feature

$x_1\ x_2\ x_3\ x_4 \dots x_n$ — o/p y

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

$$= P(y) * P(x_1/y_1) * P(x_2/y_2) \dots P(x_n/y_n)$$

$$P(x_1) P(x_2) \dots P(x_n)$$

DATA SET

x_1	x_2	x_3	x_4	y
				yes
				No

$$P(y = yes / x_0) = P(yes) * P(x_1/yes) * P(x_2/yes) * P(x_3/yes) * P(x_4/yes)$$

$$P(x_1) * P(x_2) * P(x_3) * P(x_4)$$

$P(y=No|x_i^o) = P(No) * P(x_1/No) * P(x_2/No) * P(x_3/No) * P(x_4/No)$

$P(x_1) * P(x_2) * P(x_3) * P(x_4)$

x_i^o → Yes
→ No

$P(yes|x_i^o) = 0.13$ $P(No|x_i^o) = 0.05$

$\geq 0.5 \Rightarrow 1$

$< 0.5 \Rightarrow 0$

Normalization

$P(Yes|x_i^o) = \frac{0.13}{0.13+0.05} = 0.72$ 72%

$P(No|x_i^o) = 1 - 0.72 = 0.28$ 28%

DATASET

Day	outlook	Temp.	Humidity	Wind	Play
D1	Sunny	Hot	High		No
D2	Sunny				No
D3					yes
	Rain	cool			Yes
D14		Mild			

I/P
Outlook

O/P
Play Tennis

Sunny
Overcast
Rain

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

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

PLAY

		P(Yes)	P(No)
Yes	9	9/14	5/14
No	5		
	14		

→ Test (Sunny, Hot) → o/p

$P(Yes / (Sunny, Hot)) = P(Yes) * P\left(\frac{Sunny}{Yes}\right) * P\left(\frac{Hot}{Yes}\right)$

Constant → P(Sunny) & P(Hot)

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

$P(Yes / (Sun, Hot)) = 2/63 = 0.03$ (Normalize) → 0.27

$P(No / Sunny, Hot) = \frac{3}{35} = 0.085$ → 0.73

(Sunny, Hot) → Yes or No

Answer → No

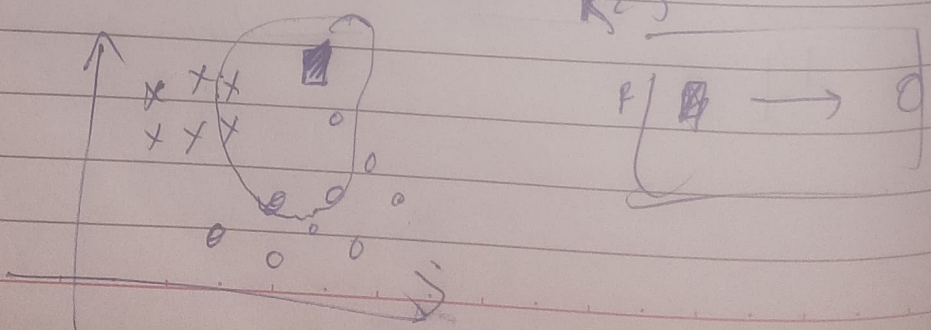
KNN Algorithm { K Nearest Neighbor }

Classification

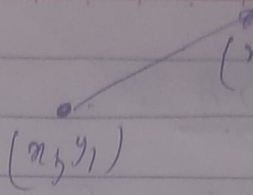
Regression

① Classification

K=5

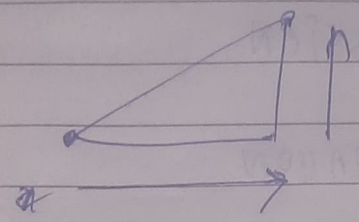


✓ Euclidean distance

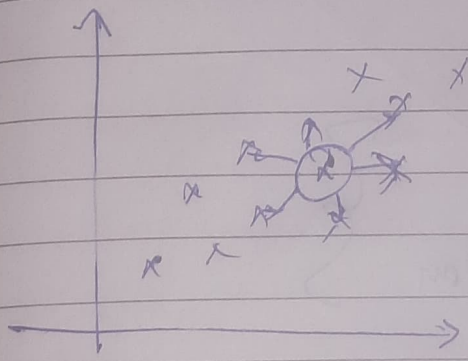


$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

✓ Manhatten distance

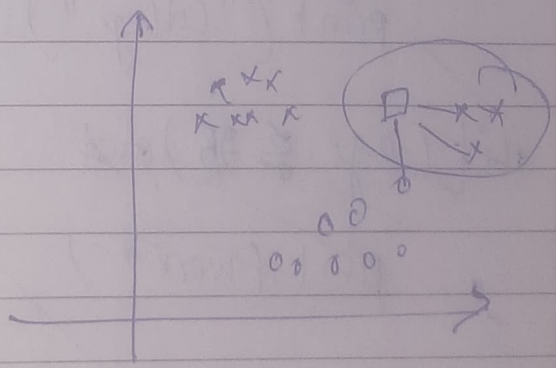


Regression



$k = 5$ → Hyperparameter

K - Nearest



(1) Outliers

(2) Imbalanced

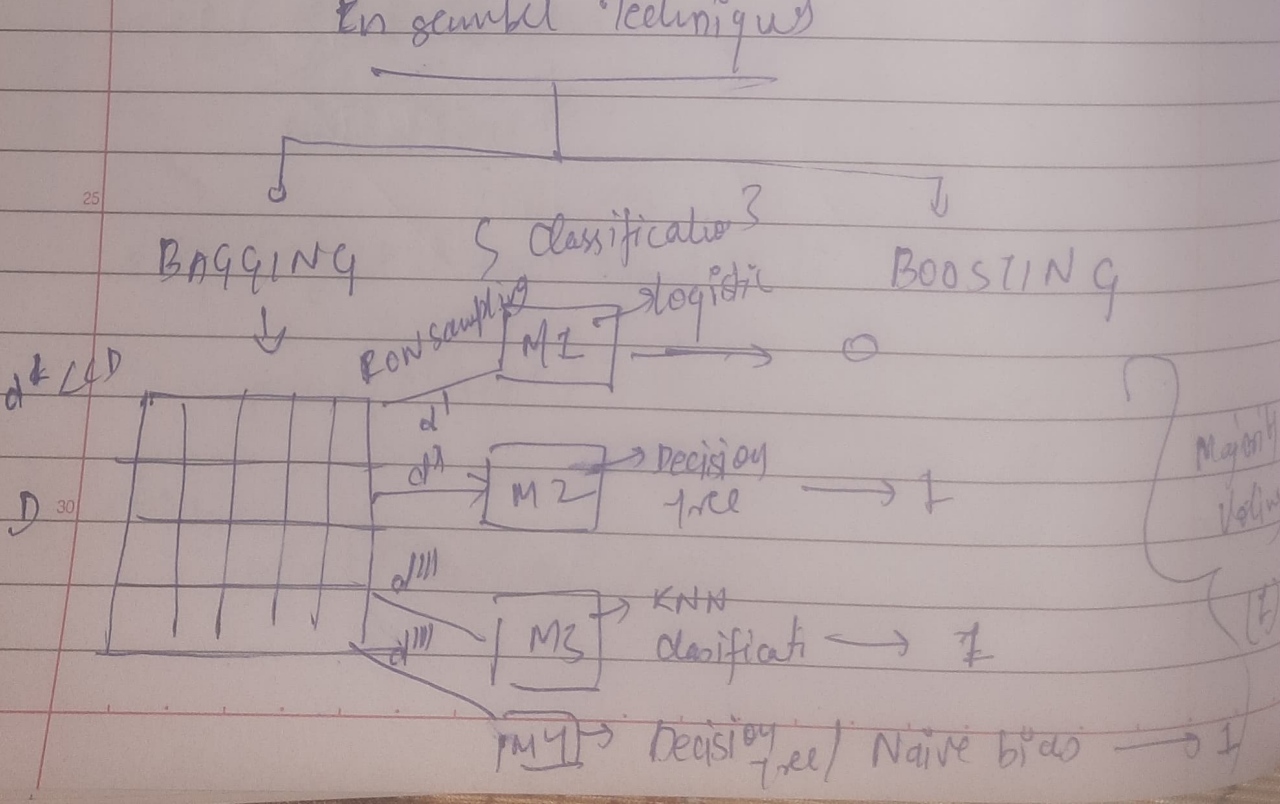
Agenda

1. Ensemble Techniques → Bagging, Boosting
2. Random Forest
3. Ada Boost
4. Xg Boost

Ensemble Techniques

1. Classification & Regression
   - 1 Algorithm → Class, Reg

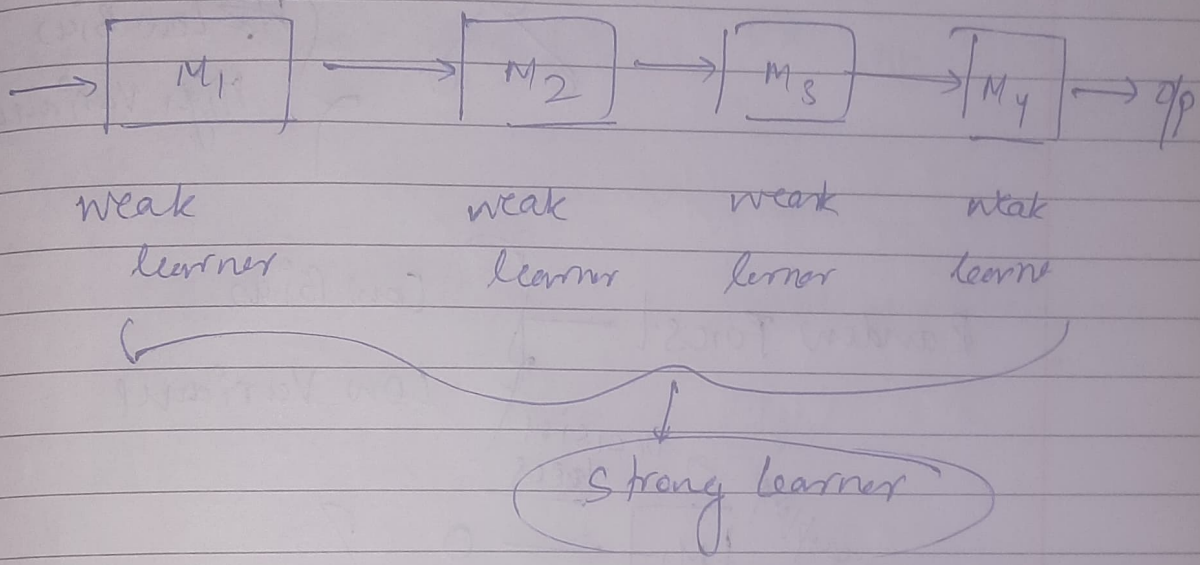
Multiple Algorithms to solve a problem ?

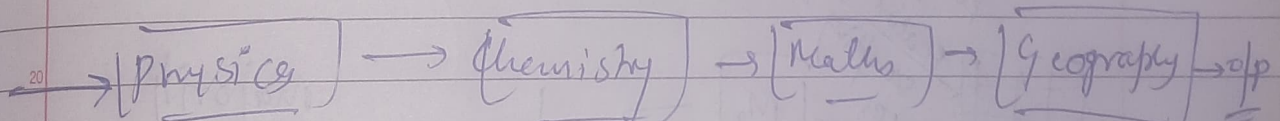


{ Bootstrap Aggregating }

Regression ? { Mean will be taken }

BOOSTING





BAGGING

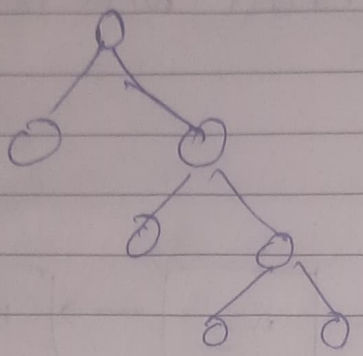
1. RANDOM FOREST CLASSIFIER
2. Random Forest Regressor

BOOSTING

1. Ada boost
2. gradient boost
3. xg boost

① Random Forest Classifier & Regressor

what is the main problem of DT?

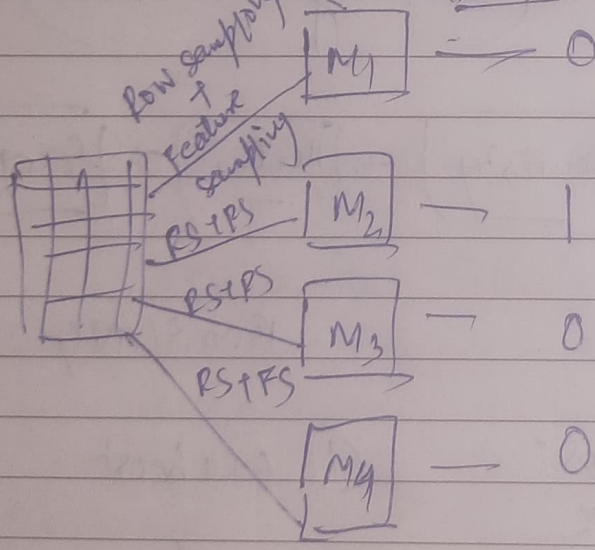


Overfitting?

- ~~Low~~ Low Bias
- High Variance

Random Forest
- Low Bias
- Low Variance

Decision Trees



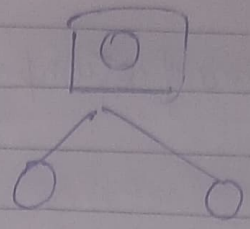
Majority Voting (o)

↳ Regression

↓

Mean

① Normalization?? NO



② KNN {Standardization}?? Yes

{Euclidean, Manhattan}

③ Random Forest → Outliers ?? No

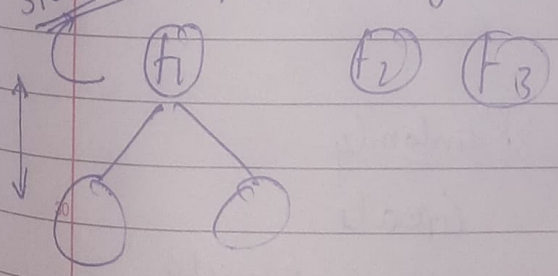
Boosting

1) Adaboost → Decision Tree

overall weight = 1

f_1	f_2	f_3	f_4	o/p	weight
–	–	–	–	Yes	1/7
–	–	–	–	NO	1/7
–	–	–	–	1	1/7
–	–	–	–		1/7
–	–	–	–		1/7
–	–	–	–		1/7
–	–	–	–	1	1/7

STUMPS Information gain, entropy



total error = 1/7

{Weak learner}

① Performance of stump $= \frac{1}{2}\log_e\left(\frac{1-TE}{TE}\right)$

$= \frac{1}{2}\log_e\left(\frac{1-1/7}{1/7}\right) \Rightarrow (0.895)$

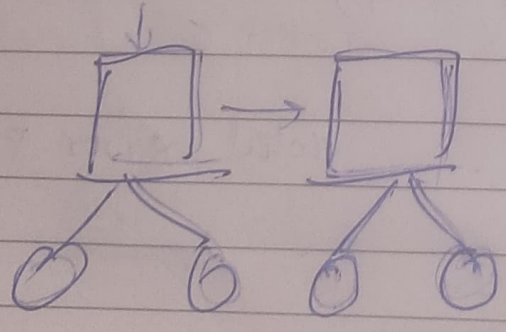
TE → total error

② New Sample Weight → Correct Records

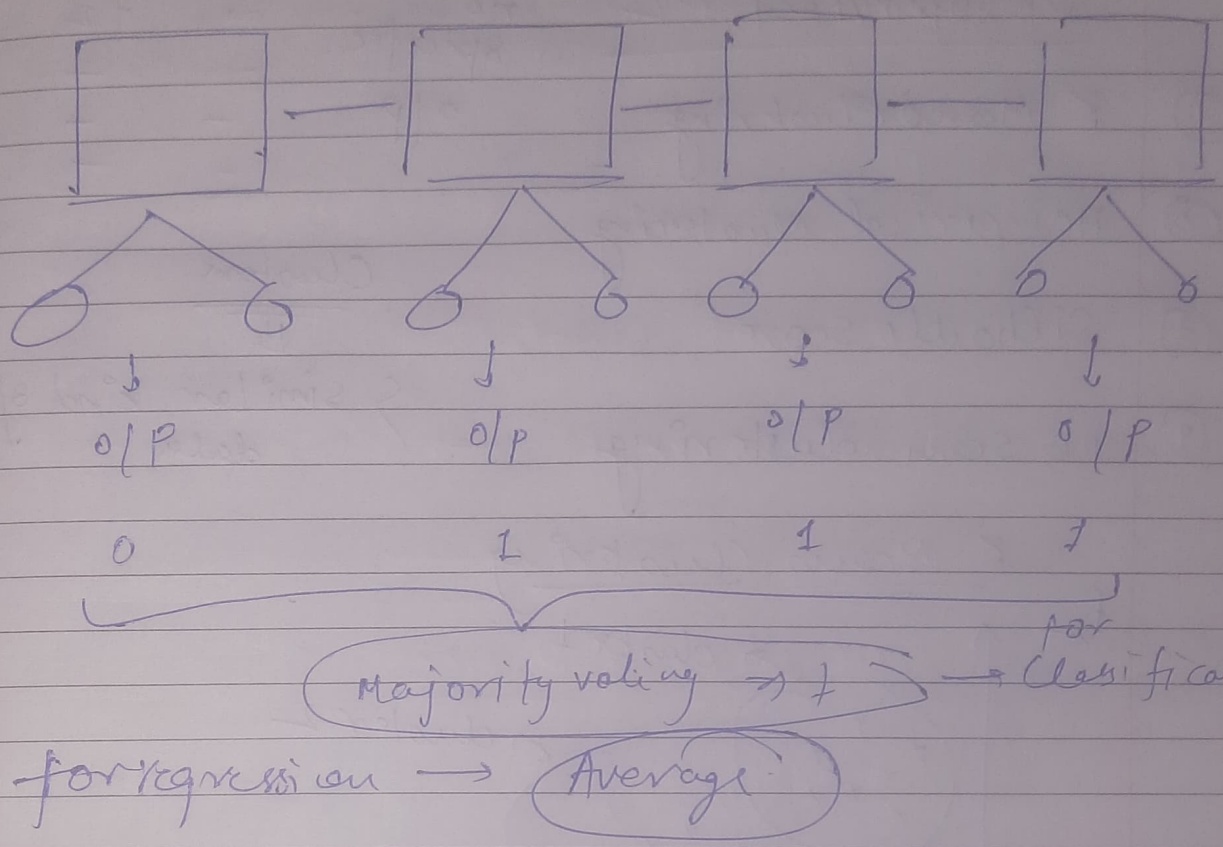
↳ Weight * $e^{-Ps} = \frac{1}{7} \times e^{-0.895}$

Incorrect record = Weight * $e^{Ps} = \frac{1}{7} \times e^{0.895} = 0.349$

New Weight	Is it ?? Normalized weight	Buckets
0.05	0.07	[0 - 0.07]
0.05	0.07	[0.07 - 0.14]
0.05	0.07	[0.14 - 0.21]
0.349	0.537	[0.21 - 0.747]
0.05	0.07	[0.747 - 0.817]
0.05	0.07	[!]
0.05	0.07	[!]
0.649	≃ 1	



Randomly Create some number b/w 0 - 1



Black box models V/S white box Models

Linear Regression → Whitbox ANN → Black Box

Random Forest → Blackbox

Decision Tree → whitebox

Unsupervised ML

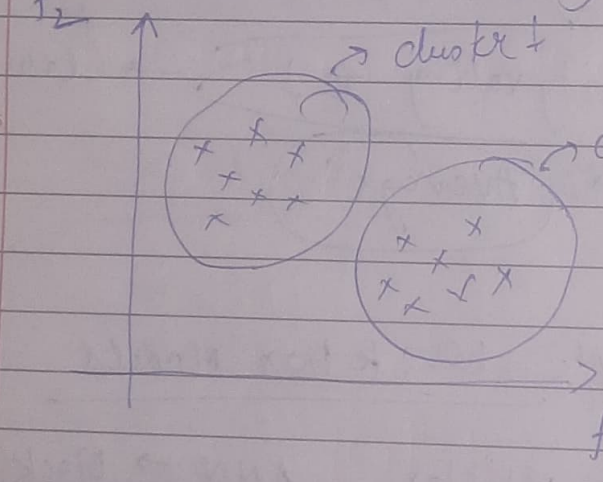
no specific o/p

f_1	f_2
–	–
–	–
–	–

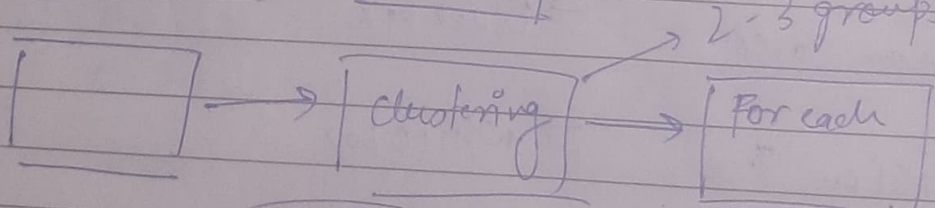
1. K Means Clustering
2. Heirarical clustering
3. Silhoutte score
4. DB scan clustering

Clusters ⇓ {similar kind of data}

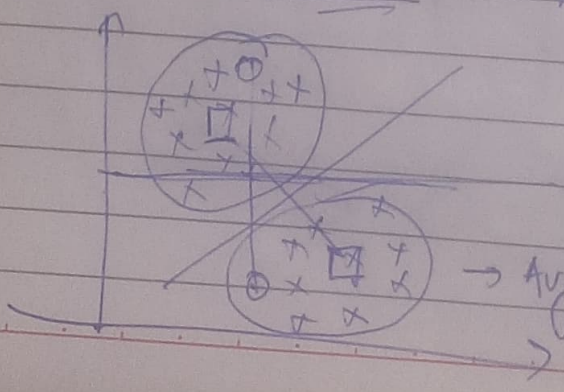
K Means Clustering



Custom Ensemble Technique



K Means → K = 2 centroids



K = 2

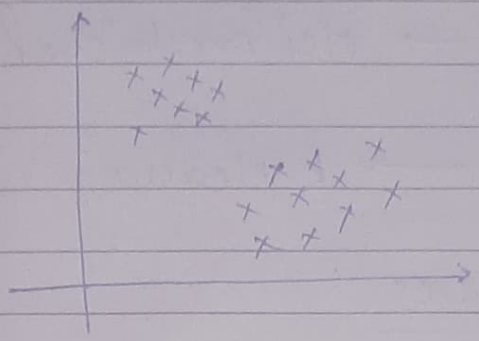
High dimension

① We try with different K values → suitable

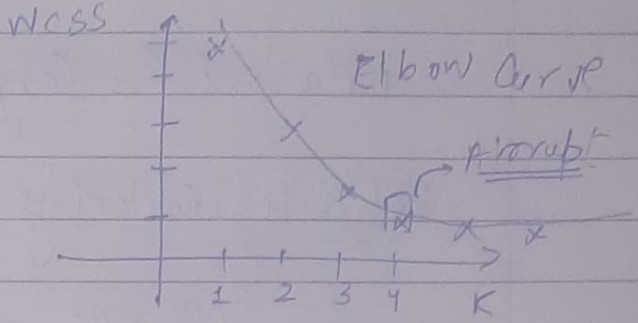
② Initialize K number of centroids

③ Compute the average to update centroids

Elbow method (K value??)

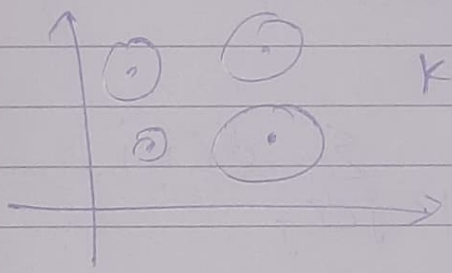


for i = 1, 10



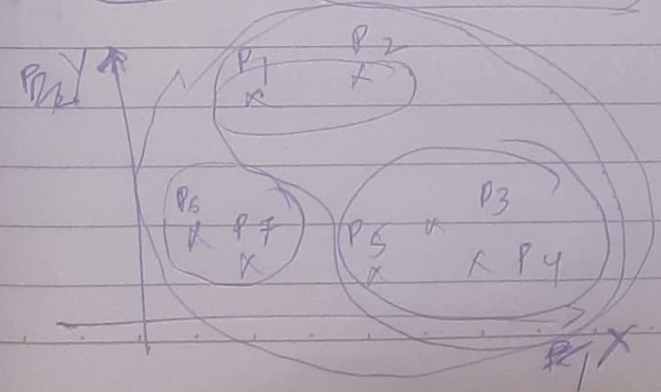
WCSS → Within cluster sum of squares

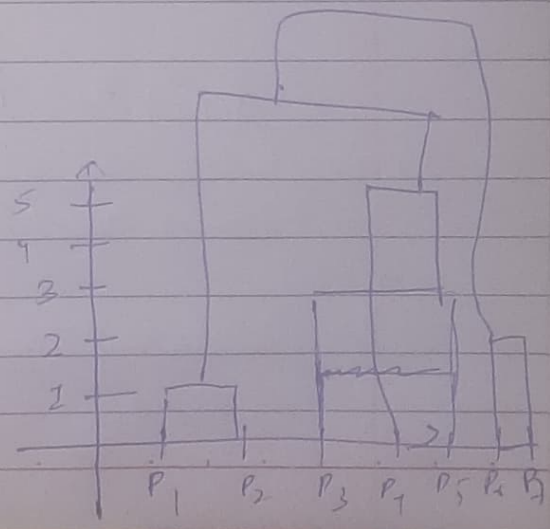
Validating → Silhouette score



Dendogram

Hierarchical Clustering





You need to find the longest vertical line that has no horizontal line passed through it.

Q. Max line is taken by K Means or Heirarchical clustering?

Ans: Heirarchical Clustering

Dataset is small → Heirarchical

Data set is large → K Means

Validate Clustering Models

→ Silhouette Clustering

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$ → Avg. distance

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

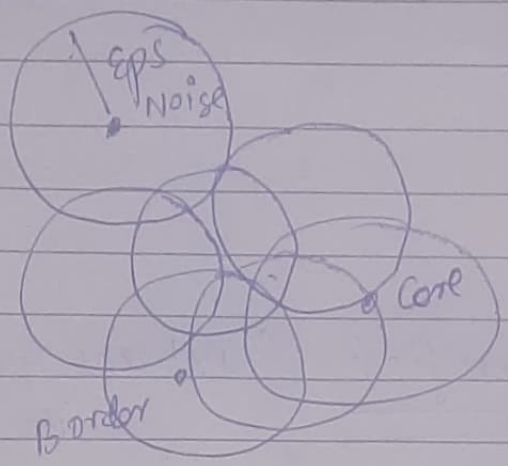
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

DB Scan Clustering

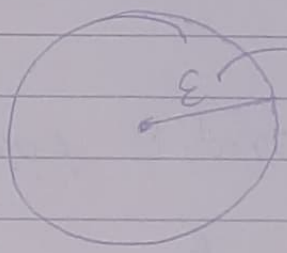
Density-Based Spatial Clustering of Application with Noise (DBSCAN)

C hyperparameter

MinPts = 4 → Eps



1. Min pts
2. Core points
3. Border points
4. Noise points



Agenda

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

Xg boost Classifier (Extreme Gradient Boosting)

Dataset

Salary	Credit	Approval	Residual
<=80	B	0	-0.5
<=50	G	1	0.5
<=50	G	0	0.5
>50	B	0	-0.5
>50	G	1	0.5
>50	N	1	0.5
<=50	N	0	-0.5

Base Model → pr = 0.5

1. Create a Binary Decision Tree using the feature
2. Calculate similarity weight (SW)

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

③ Information gain [-0.5, 0.5, 0.5, -0.5, 0.5, 0.05, -0.5

Binary DT

Salary → SW = 0.14

<50 → [-0.5, 0.5, 0.5]

SW = 0.33

>=50 → [-0.5, 0.5, 0.5, -0.5]

~~SW = [(-0.5)² + (0.5)~~

$SW = (-0.5 + 0.5 + 0.5 + 0.5)^2$

$0.5(1-0.5) + 0.5(1-0.5)$

$+ 0.5(1-0.5) + 0.5(1-0.5)$

SW = 0

Information gain = 0 + 0.33 - 0.14

= 0.19

credit

B → [-0.5]

SW = 1

G, N → [0.5, 0.5, -0.5]

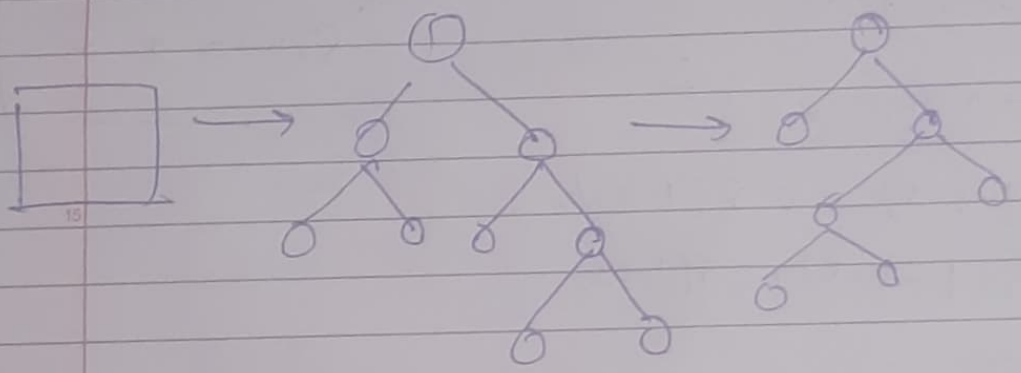
SW = 0.33

IG = 1 + 0.33 - 0 = 1.33

o/p → $\left[\sigma \left| 0 + \alpha_1(DT_1) + \alpha_2(DT_2) + \alpha_3(DT_3) + \cdots + \alpha_n(DT_n) \right. \right]$

Xg Boost → BLACK BOX Model

pre-pruning



$0 + \alpha_1(DT_1) + \alpha_2(DT_2) + \cdots \alpha_n(D$

↑ → Cross validation (Hyperparameter).

Xg Boost Regressor

Exp	Gap	Salary (O/P)	R1
2	Yes	40K	-11K
2.5	yes	41K	-10K
3	NO	42K	-9K
4	NO	60K	9K
4.5	yes	62K	11K

Base model → Avg of all value (51K)

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

Exp

SW = N/6

≤ 2 / > 2

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

Similarity weight = $\frac{\Sigma (\text{Residual})^2}{\text{No. of Residuals} + \lambda}$

SW = 60.55

SW = 28.8

Information Gain = 60.5 + 28.8 − 1/6 = 89.13

Exp

≤ 2.5 / > 2.5

[-11, 4] [1, 9, 11]

SW = 133.33

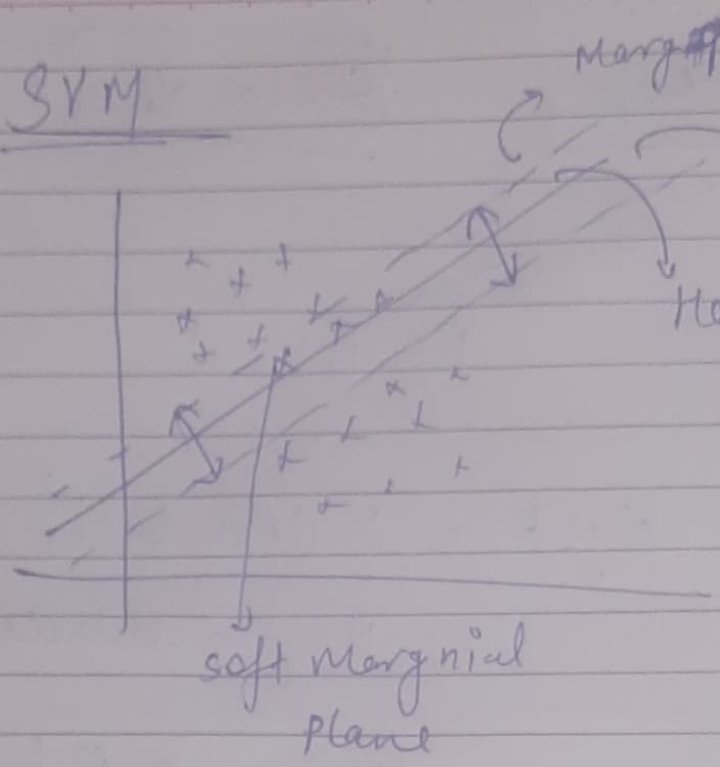
SW = 110.25

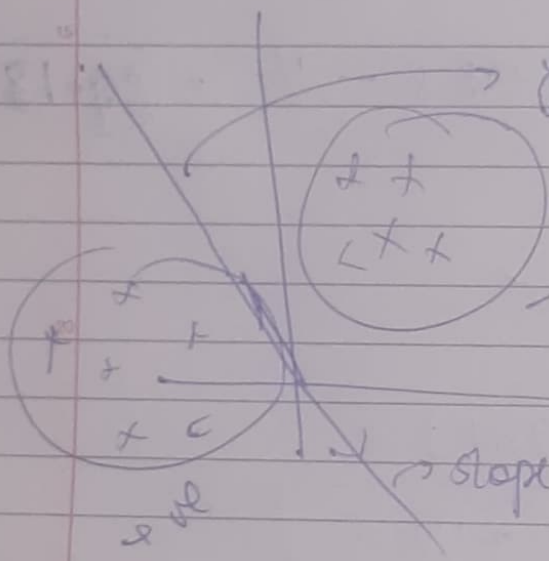
IG = 133 + 110.0 − 1/6

$\Sigma l \; \alpha_1(-10) + \alpha_2(D1_2) + \alpha_3(D1_3) - - -$

$\alpha_n(D1_n)$

SVM





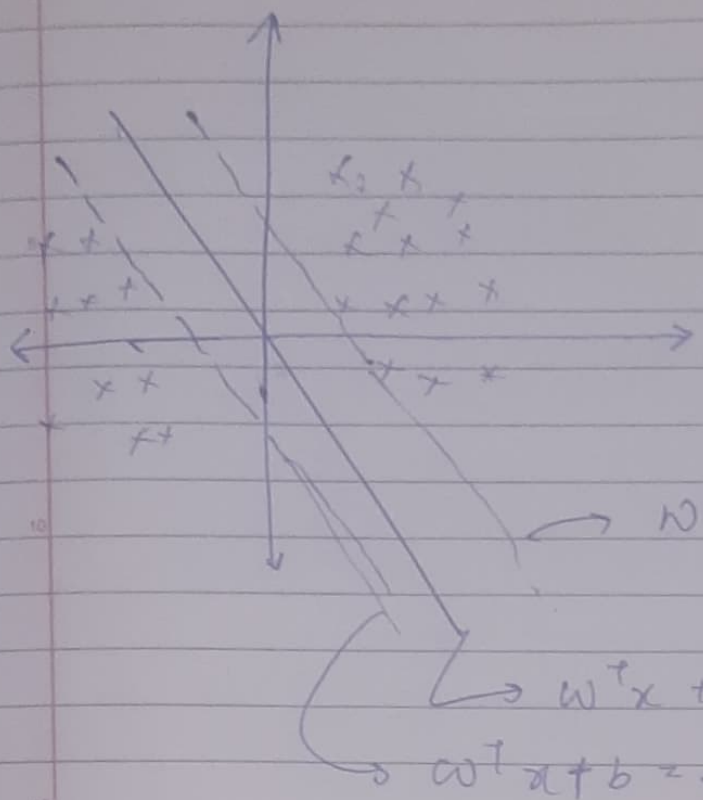
$ax + by + c = 0$ → Eq^n of a straight line

$$y = -\frac{c}{b} - \frac{ax}{b}$$

$$m = -\frac{a}{b} \qquad c = -\frac{c}{b}$$

$y = mx + c$

$y = w^T x + b$



$ax + by + c = 0$

$\Downarrow$

$w^T x + b = 0$

$w^T x + b = -1$

$w^T x + b = 0$

$w^T x + b = +1$

$w^T x_1 + b = +1$

$w^T x_2 + b = -1$

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

$\frac{}{|w|}$

~~Minimise~~ Maximise $\frac{2}{|w|}$

(w, b)

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