Machine Learning !!

Definition !!

"Study of Computer Algorithms that allows the Computer Program to automatically improve through Experience"

[Training]

[Efficient - learning]

By Tom Mitchel (founder of ML Department)
School of Computer Science at Carnegie Mellon University.

* "ML is teaching the machine about something"

How !!

1. Collect and clean the data
2. Algorithm (model) — Select (Readymade) / Built.
3. Teach the model Essential pattern from data (Training).
4. Expect the model to give Helpful Answer.

Ex To Design a System that determine from MRI Scan, whether Tumor is present or not.

(1) Collect large No of MRI reports

Ex 10000 MRI report < 6000 has Tumor
4000 do not have Tumor } Labelled Data.

Test 30%. Train 70%.

(2) Build an efficient algorithm that detects presence or absence of Tumor in an MRI Scan.
[Expert Consultation – Radiologist].

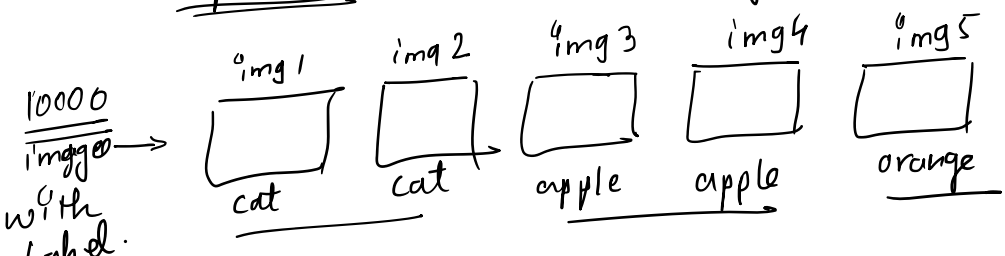
(3) To the Algorithm feed the 6000 MRI Scans and allow the model to learn (train).

(4) Use around 3000 Images (MRI Scans) for Testing

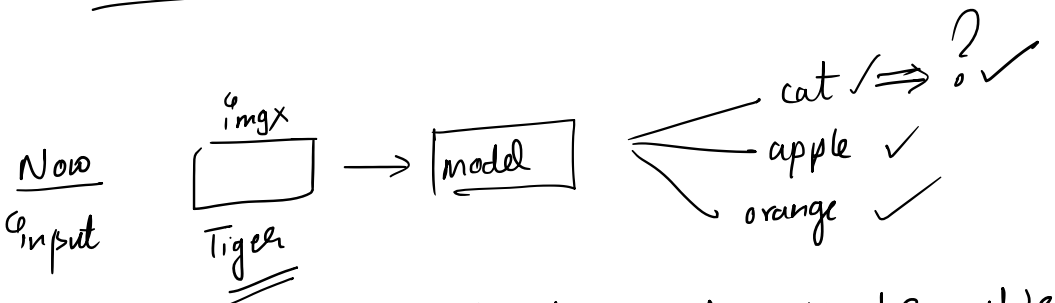
(5) Use this model to determine presence or absence of Tumor from a New Image (MRI Scan)

Types of machine learning Algorithms.

① Supervised machine learning Algorithms.



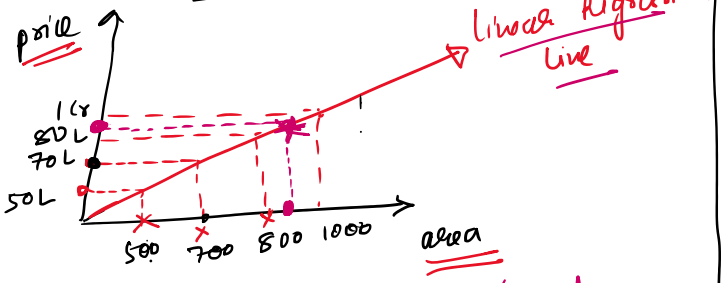
Input
→ Images ✓
→ Label ✓



* With lot of images as input, model will be able to identify pattern and will be able to predict.

Supervised.

Regression.



Ex for area = 875 (input)
Price = 75000 (output)

Here o/p is continous value.

Here area ⇒ Independent variable
price ⇒ Dependent variable

Classification.

Here o/p is not continous value but could be Boolean or some class / category as output.

* Spam detection → Spam Hai / Spam Nahi Hai

* MRI Scan For Tumor Detection → Tumor Hai / Tumor Nahi Hai

In above cases the o/p belongs to

Here area ⇒ Independent variable

price ⇒ Dependent variable

Here the dependent and Independent Variable can have Continous value.

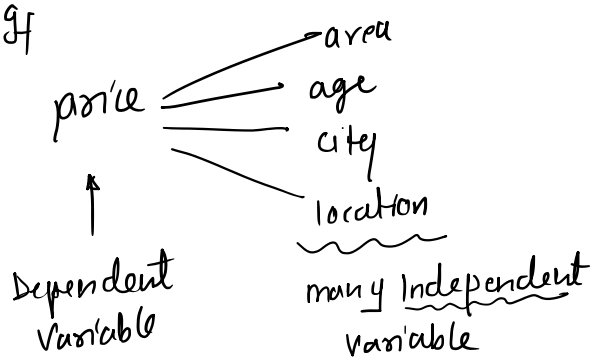
Types ⇒

if price → area.

Single Independent variable

→ Simple Regression

if

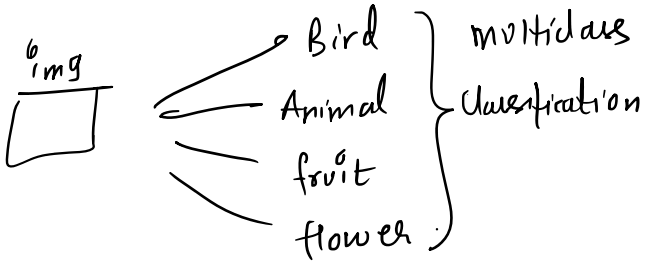


When we have more than one Independent variable ⇒ multiple Regression

In above cases the o/p belongs to one of the two dasses.

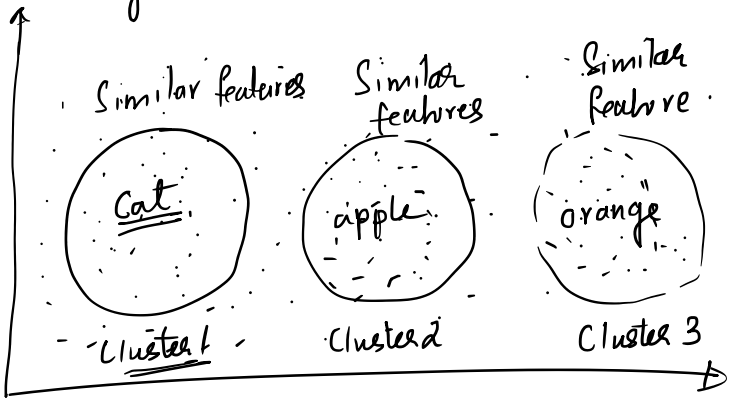
→ Binary classification.

if o/p can be one of many dasses (more than two) then it is called as multiclass classification.



Unsupervised Learning

Given 10000 images and no label are specified.



* Here the input is only data and no labels are associated with data.
* Not Sure about type of output
* Unsupervised Algo will work on 10000 images and will create cluster of images based on the similarities.
* It is "SELF LEARNING"

Unsupervised.

Clustering

(1) Model will identify the pattern in input data and will create cluster.

Ex Market Segmentation.

* Study the Spending / Income of population and create cluster

Association

(1) We find dependencies of one data item on another.

(2) This dependency will help to map the relation and will enhance prediction.

(3) It is like "If - Then"

Ex Used by Amazon known as

population and create clusters

(1) High Earning
(2) Medium Earning
(3) Low Earning

Ex Used by Amazon known as Market Basket Analysis.

"If a person purchases cellphone then the person has tendency to purchase screen guard & backcover"

Semi Supervised.

Text Document Classifier

Ex 1000000 articles. and need to classify them into
- News
- Literature
- Research paper
- medical Report

Label is not provided.

Manually labelling the 1000000 articles not possible.

→ we will label (10000) articles [Supervised Learning]

→ my model will be trained on these 10000 articles and will use the pattern identified to classify the remaining 990000 articles [UnSupervised]

* Uses small amount of labelled data
* and large amount of unlabelled data.
* Benefits of Both Labelled and unlabelled data.
* Overcome the challenge of finding large amount of Labelled data.

Reinforcement Learning [Experential Learning]

→ Here the agent/model learns how to behave in an Environment by performing action and experiencing the result.

Types

Episodic Learning

→ Here we have Start and End state, thus an episode is created.

→ Thus the feather action is based on feedback of result of earlier action.

Ex Fear of dog after being bitten is Episodic Learning.

Continous Task Learning

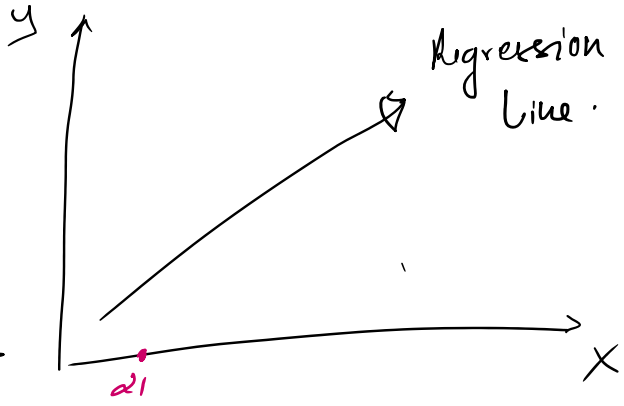
* There is no terminal state here
* Agent/model that does Automated Stock Trading goes for Continous Learning

	INDEPENDENT	DEPENDENT		
SR.NO	X	Y	x^2	xy
1	43	99	1849	4257
2	21	65	441	1365
3	25	79	625	1975
4	42	75	1764	3150
5	57	87	3249	4959
6	59	81	3481	4779
	$\Sigma x = 247$	$\Sigma y = 486$	11409	20485

↓ Actual value. $n = 6$

Now predict is $X = 55$, what is y ?

Σx^2 Σxy



→

$$y = a + bx + e$$

- y: Dependent variable
- a: Intercept
- b: decides impact of X on y.
- e: Error

a = intercept

b = slope (coefficient of Independent Variable)

e = Error

[if x↑ y↑ +ve Impact]

[if x↑ y↓ -ve impact].

Let us assume $e = 0$

$$y = a + bx \quad \text{— (1)}$$

To find a and b.

Take Σ on both side of 1

$$\Sigma y = \Sigma a + \Sigma bx$$

$$\Sigma y = a\Sigma 1 + b\Sigma x$$

$$\Sigma y = an + b\Sigma x \quad \text{— (2)}$$

multiply Eq 2 with Independent Variable X.

$$\Sigma xy = a\Sigma x + b\Sigma x^2 \quad \text{— (3)}$$

multiply Eq 2 with independent ...

$$\Sigma xy = a\Sigma x + b\Sigma x^2 \quad -(3)$$

Here $\frac{Eq\ 2}{Eq\ 3}$

$$486 = a6 + b247 \quad \checkmark$$

$$20485 = a247 + b11409 \quad \checkmark$$

$$a = 65.14$$

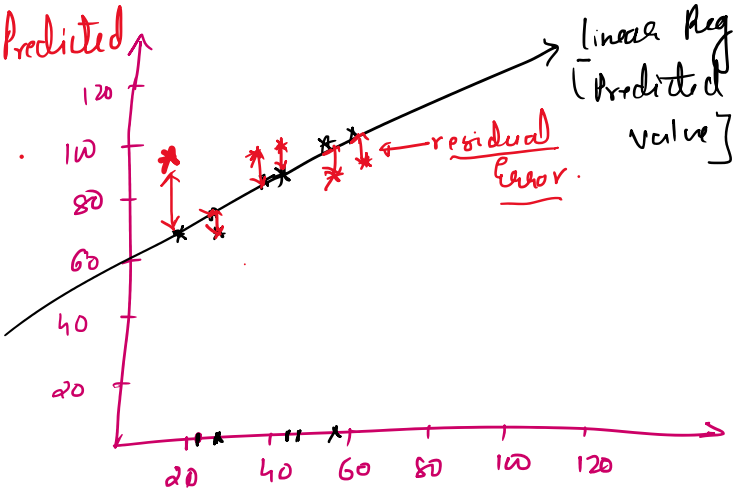
$$b = 0.385$$

$$y = 65.14 + 0.385x$$

Using this let us calculate Predicted Y for Every X

Actual − Predicted

SR.NO	INDEPENDENT X	DEPENDENT(ACTUAL) Y	PREDICTED Y	DIFFERENCE
1	43	99	81.695	-17.305
2	21	65	73.225	8.225
3	25	79	74.765	-4.235
4	42	75	81.31	6.31
5	57	87	87.085	0.085
6	59	81	87.855	6.855



Residual Error = Error between the Actual value and Predicted value.

Yaad Rakho

$$y = a + bx$$

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\ldots}$$

Quantitative Analysis

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Given X & Y pe^n $\boxed{y = a + bx}$

Find a, b.

If more than one Independent Variable ->

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 \cdots$$

b_1: impact of x_1 on y

b_2: impact of x_2 on y

b_3: impact of x_3 on y.

In Earlier Example we had one Dependent variable ie y and one Independent variable ie x

So it was Case of Simple Linear Regression.

Ex sqftarea → price $\Rightarrow$ $\hat{y} = \beta_0 + \beta_1 x$

But if

x_1 Size (feet)^2
x_2 No of Bedrooms
x_3 No of floor
x_4 location

→ Price y

← multiple linear Regression.

The linear Regression to express the above dependency is

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

General form of multiple linear Regression.

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Here $\beta_0, \beta_1, \beta_2 \ldots \beta_n$ < +ve OR -ve

$$h_\theta(x) = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

$x_0 = 1$

Here $x_0, x_1, x_2 \ldots x_n$ = features

$\beta_0, \beta_1, \beta_2 \ldots \beta_n$ ⇒ Parameter / coefficients.

let X = Feature Vector = $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$ $n+1 * 1$

n = no of features

θ = Parameter Vector = $\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$ $n+1 * 1$

$$\therefore h_\theta(x) = \theta^T \cdot X = \underbrace{\begin{bmatrix} \beta_0 & \beta_1 & \beta_2 & \cdots & \beta_n \end{bmatrix}}_{\theta^T} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} X$$

$$h_\theta(x) = \theta^T X = \beta_0 x_0 + \beta_1 x_1 + \cdots + \beta_n x_n.$$

x_0	X_1	X_2	y
1	7	2.6	78.5
1	1	2.9	74.3
1	11	5.6	104.3
1	11	3.1	87.6
1	7	5.2	95.9
1	11	5.5	109.2
1	3	7.1	102.7

3 2 2

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

To find β_0
β_1
β_2

Solve ->

Feature Vector = X = $\begin{bmatrix} 1 & 7 & 2.6 \\ 1 & 1 & 2.9 \\ 1 & 11 & 5.6 \\ 1 & 11 & 3.1 \\ 1 & 7 & 5.2 \\ 1 & 11 & 5.5 \\ 1 & 3 & 7.1 \end{bmatrix}$ 7×3 & $y = \begin{bmatrix} 78.5 \\ 74.3 \\ 104.3 \\ 87.6 \\ 95.9 \\ 109.2 \\ 102.7 \end{bmatrix}$ 7×1

$$C = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = (X^T . X)^{-1} . X^T . y$$

↑ coffiient Vector ↑ 3×1

Solution

$X = \begin{bmatrix} 1 & 7 & 2.6 \\ 1 & 1 & 2.9 \\ 1 & 11 & 5.6 \\ 1 & 11 & 3.1 \\ 1 & 7 & 5.2 \\ 1 & 11 & 5.5 \\ 1 & 3 & 7.1 \end{bmatrix}$

(1) find $X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 1 & 11 & 11 & 7 & 11 & 3 \\ 2.6 & 2.9 & 5.6 & 3.1 & 5.2 & 5.5 & 7.1 \end{bmatrix}$

(2) Find $X^T . X \Rightarrow$ 3×7 7×3 $\begin{bmatrix} 7 & 51 & 32 \\ 51 & 471 & 235 \\ 32 & 235 & 163.84 \end{bmatrix}$ 3×3

(3) Find $(X^T . X)^{-1}$

$= \begin{bmatrix} 1.79 & -0.06 & -0.25 \\ -0.06 & 0.01 & -0.0011 \end{bmatrix}$

$$= \begin{bmatrix} 1.79 & -0.06 & -0.25 \\ -0.06 & 0.01 & -0.0011 \\ -0.25 & -0.0011 & 0.0571 \end{bmatrix}$$

$$= (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

$$\theta = \begin{bmatrix} 51.6 \\ 1.5 \\ 6.72 \end{bmatrix}_{3\times 1} \quad \beta_0 = 51.6 \quad \beta_1 = 1.5 \quad \beta_2 = 6.72$$

$$y = 51.6 + 1.5x_1 + 6.72x_2$$

what is y when $x_1 = 3$ $x_2 = 2$

$$y = 51.6 + (1.5 \times 3) + 6.72 * 2 = 69.54$$

Consider Formulae that will be used, when only 2 Independent Variables specified.

[>2 features, use matrix Algebra].

Consider

X_1	X_2	y
3	8	-3.7
4	5	3.5
5	7	2.5
6	3	11.5
2	1	5.7
3	2	?

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$\theta_0 = ?$

$\theta_1 = ?$

$\theta_2 = ?$

$$\theta_0 = \bar{y} - \theta_1 \bar{x_1} - \theta_2 \bar{x_2}$$

$\bar{y} =$
$\bar{x_1} =$ } mean
$\bar{y_1} =$

$$\theta_1 = \frac{(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

$$\theta_2 = \frac{(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)}{(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2}$$

where

$$\Sigma x_1^2 = \Sigma x_1 x_2 - \frac{(\Sigma x_1)(\Sigma x_1)}{N}$$

$$\Sigma x_2^2 = \Sigma x_2 x_2 - \frac{(\Sigma x_2)(\Sigma x_2)}{N}$$

$$\Sigma x_2 = \Sigma x_2^2 - \frac{(\Sigma x_2)(\Sigma \quad)}{N}$$

$$\Sigma x_1 x_2 = \Sigma x_1 x_2 - \frac{(\Sigma x_1)(\Sigma x_2)}{N}$$

$$\Sigma x_1 y = \Sigma x_1 y - \frac{(\Sigma x_1)(\Sigma y)}{N}$$

$$\Sigma x_2 y = \Sigma x_2 y - \frac{(\Sigma x_2)(\Sigma y)}{N}$$

Sol^n = $\boxed{y = a_0 + a_1 x_1 - 1.67 x_2}$

HW

* To Evaluate Peerformance of M.L models
* To find how good the model fits on given data

(1) Karl Pearsons Coefficient of Correlation (γ)

→ To calculate Relationship bet two variable

$$\gamma = \frac{N\Sigma xy - \Sigma x \Sigma y}{\sqrt{\left(N*\Sigma x^2 - (\Sigma x)^2\right)\left(N*\Sigma y^2 - (\Sigma y)^2\right)}}$$

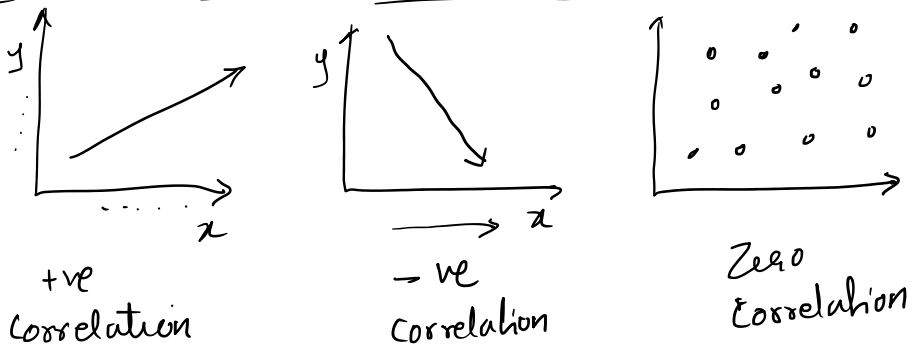
γ = quantifies the strength of relationship bet^n two variables.

The value of γ be between $+1$ and -1

If $\gamma = 1$ ⇒ Total +ve correlation ⇒ If $x\uparrow$ then $y\uparrow$

If $\gamma = -1$ ⇒ Total -ve correlation ⇒ If $x\downarrow$ then $y\uparrow$ or $x\uparrow$ then $y\downarrow$

→ It gives strength (degree) & direction of correlation.

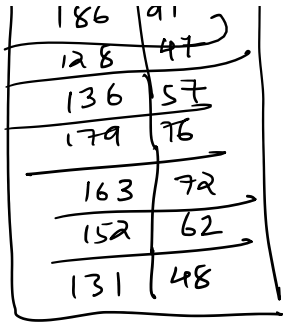


+ve Correlation — −ve Correlation — Zero Correlation.

Q. 11.10

x	y
151	63
174	81
138	56
186	91
128	47
136	57

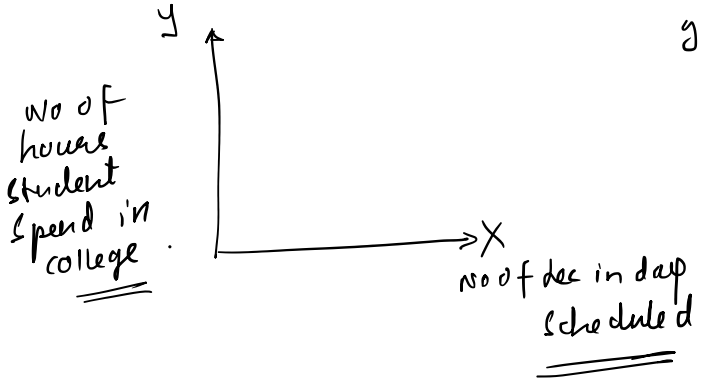
Find γ = ? 0.9884


186 91
128 47
136 57
179 76
163 72
152 62
131 48

② R^2 method (R Square)

↳ It gives information about good of fit feature of the model.

↳



If $r^2 = 0.85$

Variation in no of hours that students spend in college is 85% dependent on no of dec scheduled.

↳ Indicate percentage of variance in dependent and Independent variable pair

↳ value varies from 0 to 1

If $r^2 = 1$ ⇒ no diff bet^n actual & predicted value.

$r^2 = 0$ means the model does not learn any relationship bet^n variables.

① SST (Sum of squares of Total) = $\Sigma (y - \bar{y})^2$

actual y ↑ mean of y.

② SSR (Sum of Squares due to regression) = $\Sigma (\hat{y} - \bar{y})^2$

predicted y ↑ mean of y.

③ SSE (Sum of Squares of Error) = $\Sigma (y - \hat{y})^2$

actual y ↑ predicted y

$$R^2 = SSR/SST = \frac{\Sigma(\hat{y}-\bar{y})^2}{\Sigma(y-\bar{y})^2}$$

Q.

H.W

x	y	$\hat{y}$
151	63	
174	81	
138	56	
186	91	
128	47	
136	57	
179	76	
163	72	
152	62	
131	48	

To find R^2

To find Predicted i.e $\hat{y}$

$y = \beta_0 + \beta_1 x$

find $\beta_0 = -38.455$

$\beta_1 = 0.6746$

$$y = -38.455 + 0.6746x$$

Now $x = 151$ find $\hat{y} = 63.4$

$x = 174$ $\hat{y} = 78.92$

$x = 138$ $\hat{y} = 54.63$

Note $R^2 = \frac{\Sigma(\hat{y}-\bar{y})^2}{\Sigma(y-\bar{y})^2}$

if line fits properly

then $\hat{y} \approx y$

$\therefore R^2 \approx 1$

(3) Standard Error of Estimate →

* Measures the accuracy of Prediction.
* $\sigma_{est} = \sqrt{\frac{\Sigma(y-\hat{y})^2}{N}} = \sqrt{MSE} = \sqrt{\frac{SSE}{N}}$

(mean square error)

y = actual value
$\hat{y}$ = predicted value.

* It reflects how well the Regression model fits the dataset.
* Smaller the value better it is
* Larger the value worst it is

Q.

x	y
151	63
174	81
138	56
186	91
128	47
136	57
179	76
163	72
152	62
131	48

$\sigma_{est} = 2.909$

Classification →

Ex Mail → Spam (+ve) / Not Spam (-ve)

Online Transaction → Fraudulent (+ve) / Non Fradulent (-ve)

Tumor → Malignent (+ve) / Benign (-ve)

where $y \in \{0, 1\}$ — 0 ⇒ -ve class; 1 ⇒ +ve class.

Here y has Discrete Value.

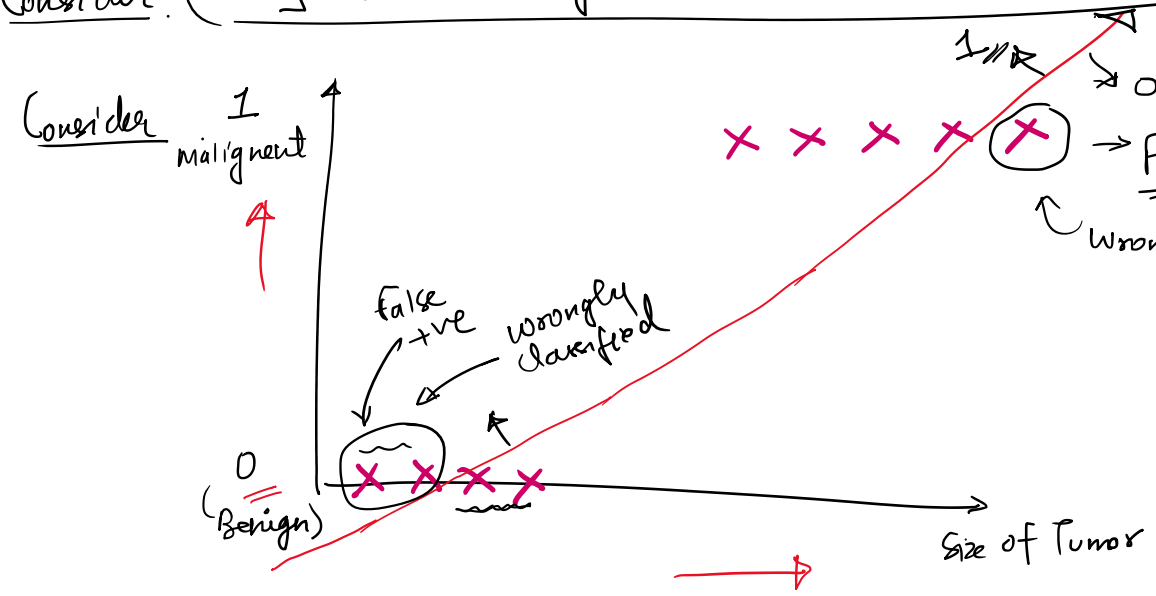
In above case o/p has only 2 class { 0, 1 } Binary classification.

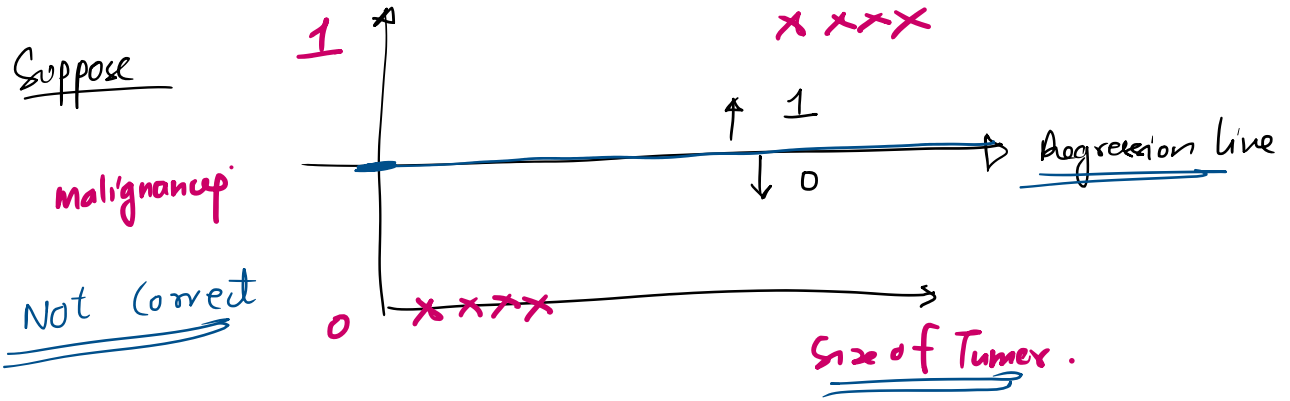
To check Weather.

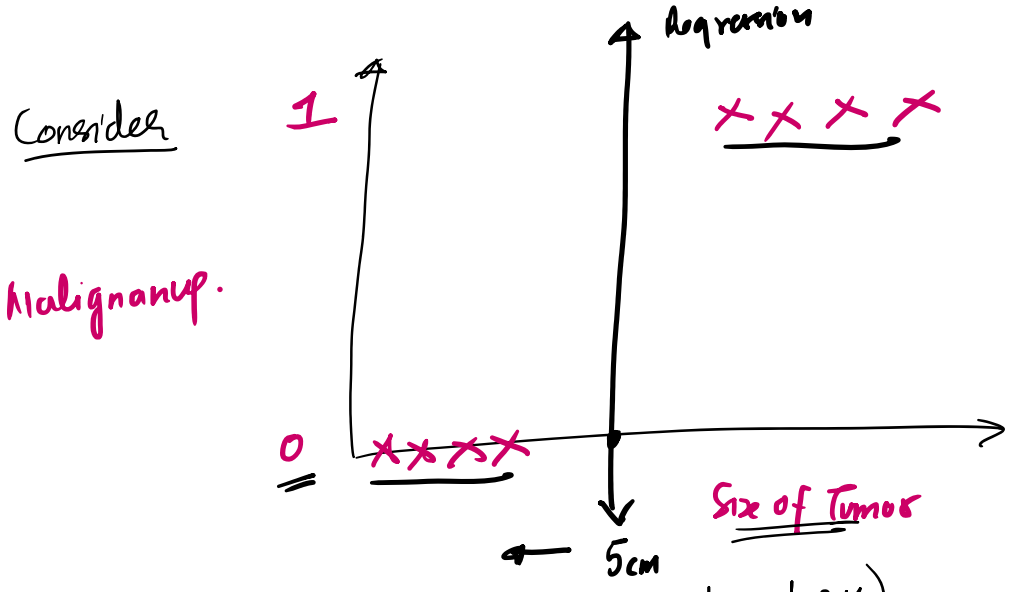
Possible o/p → Windy, Sunny, Cloudy, Rainy } o/p has more than one class [multi class classification]

* Classification bothers about label and not the Exact value.

Consider (Why linear Regression is not used for Classification)







Here we can observe for above Regression line

If tumor size $\leq$ 5cm → Yes (1) Malignant

If tumor size ≤ 5cm $\longrightarrow$ Yes (1) Malignant

tumor size > 5m $\longrightarrow$ No (0) Benign

We can say let 'P' denote Probability that $y = 1$ when $X = x$.

$$y = P_r(y = 1 \mid X = x) = [\beta_0 + \beta_1 x]$$

In linear Reg

P = probability lies bet^n 0 to 1

But linear function are unbounded.

and Expected o/p here is 0 or 1

So we cannot use Regression to build Classifier

∴ linear Regression is not suitable for Classification.

For Classification we will use Logistic Regression.

* In Logistic Regression we get Probability Score.
* It predicts the probability of occurance of event

Odd = (Represents chances that the event will occur) = $\frac{\text{No of time the Event happens}}{\text{No of time the Event will not happen}}$.

Ex ⇒ If odd of India winning against W.I is 4:1 = $\frac{\text{No of India win}}{\text{No of India not win}}$

$= \frac{4}{1}$

Best Case ⇒ The odd of India winning against W1 = ∞

If odd of W.I winning against India is 1:4

$= \frac{\text{No of W.I win}}{\text{No of W.I not win}} = \frac{1}{4}$

Worst Case ⇒ W.I is winning 0 match = Odd = 0

Range of value that odd can take = 0 to ∞

Relationship between Odd & Probability = Odd = $\frac{P}{1-P}$

$\therefore \quad odd = \frac{p}{1-p}$.

Here we know that odd has range 0 to ∞

→ There is no upper bound for odd.

→ But odd has lower bound.

+ To remove lower bound, to have symmetrical analysis

Ex odd = 1:6 1/6 = 0.167 } not symmetrical
odd = 6:1 6/1 = 6 }

But $\ln(1/6) = -1.79$ } Symmetrical
$\ln(6/1) = 1.79$ }

By taking log we have overcommed the lower bound also

log(odd) → will not have upper bound
→ will not have lower bound.

$\log(odd) = y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$

$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$.

Let $z = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$.

$$\therefore \log_e\left(\frac{p}{1-p}\right) = z$$

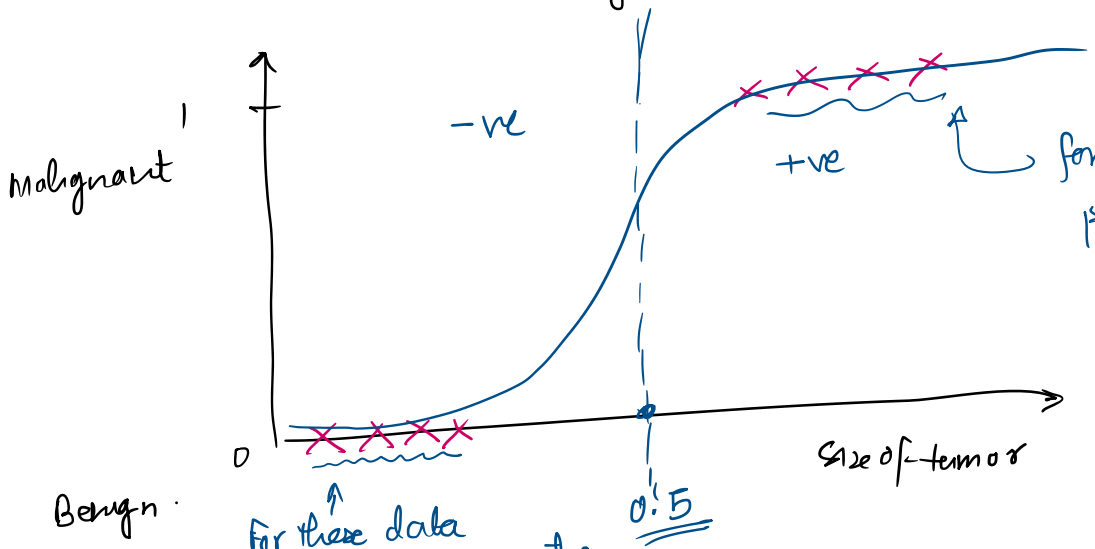
$$\frac{p}{1-p} = e^z$$

$$p = -e^z p + e^z$$

$$p(1+e^z) = e^z$$

$$p = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}}$$

Sigmoid fn.



Logistic Regression Model :

In linear reg $h_\theta(x) = \theta^T x$ → range $-\infty$ to $+\infty$

x ← feature vector.

θ ← parameter vector

$h_\theta(x)$ ← hypothesis (prediction).

In logistic Regression $0 \le h_\theta(x) \le 1$

$h_\theta(x)$ ← predicted value

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

g ← Sigmoid

Sigmoid

let $z = \theta^T x$

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

Estimated probability that $y=1$ on given input x.

prediction.

if $h_\theta(x) = 0.7$

this means there is 70% chance that tumor is malignant.

Since $h_\theta(x) = 0.7 > 0.5$

So $\boxed{y=1}$

if $h_\theta(x) = 0.3$

this means there is 30% chance of tumor being malignant

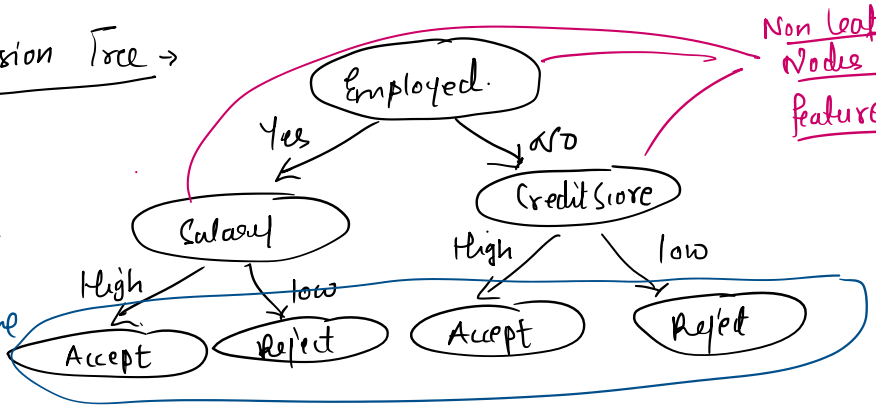
Since $h_\theta(x) = 0.3 < 0.5$

So $\boxed{y=0}$

Decision Tree :-

Consider → Bank → Details of Loan Applicant (Input) → Accept (1) / Reject (0)

A Sample Decision Tree →



Non leaf Nodes are features.

Here Independent Features → Employed → Salary → Credit Score

Outcomes are leaf Node

Outcome → Accept → Reject

Q) Create Decision Tree for following using Gini Index (Classification & Regression Tree) → CART ✓

Weekend	Weather	Parent	Money	(y) Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

p

Solution → Independent features: Weather, Parent, Money

Decision/Outcome → Cinema, Tennis, Stay In, Shopping.

Shopping.

Step 1>

We will calculate Gini Index for Overall collection of Outcomes of Training Examples.

There are 4 possible outcomes for decision
- Cinema — 6 instances
- Tennis — 2 instances
- Stay In — 1 instance
- Shopping — 1 instance.

$$Gini_{(decision)} = 1 - \left(\left(\frac{6}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^2\right)$$
$$= 1 - \left(\frac{42}{100}\right) = 0.58$$

Note -> In Machine Learning, Gini index/coefficient is utilized as an impurity measures in decision tree for classification.

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2$$ where P_i probability of outcome of specific class

Step 2: To find Gini Index for money

$Gini_{(money)}$ → a possible value → Rich, poor

(1) $Gini_{(money = Rich)}$ → 7 instance → possible decision
- Cinema → 3 time
- Tennis → 2 time
- Stay In → 1 time
- Shopping → 1 time.

$$Gini_{(money=Rich)} = 1 - \left(\left(\frac{3}{7}\right)^2 + \left(\frac{2}{7}\right)^2 + \left(\frac{1}{7}\right)^2 + \left(\frac{1}{7}\right)^2\right)$$
$$= 0.694$$

(2) Gini → ... → Cinema.

(2) Gini (money=poor) → 3 instance → Decision → Cinema.

$= 1 - \left((3/3)^2\right) = 0$ //

Weighted Average $Gini_{(money)} = \left(Gini_{(money=Rich)} * \text{Proportion of rich}\right) + \left(Gini_{(money=poor)} * \text{Proportion of poor}\right)$

$= (0.694 * 7/10) + (0 * 3/10)$

$Gini_{(money)} = 0.485$

Step 3 Gini Index on Parent

For Parent feature → Possible values → Yes / NO

$Gini_{(parent=Yes)}$ = 5 instances → Possible Decision → Cinema

$= 1 - \left((5/5)^2\right) = 0$ //

$Gini_{(parent=NO)}$ = 5 instance → possible decision → Tennis → 2 times; Stay In → 1 times; Shopping → 1 time; Cinema → 1 time.

$= 1 - \left((2/5)^2 + (1/5)^2 + (1/5)^2 + (1/5)^2\right)$

$= 1 - (7/25) = 0.72$

Weighted Average of $Gini_{(parent)} = (0 * 5/10) + (0.72 * 5/10) = 0.36$

$Gini_{(parent)} = 0.36$

$Gini_{(parent)} = 0.36$

Step 4) Gini Index for Weather.

Weather → Possible values
- Sunny = 3 instances
- Windy = 4 instances
- Rainy ⇒ 3 instances.

Gini (weather = sunny) ⇒ 3 instances → possible outcomes
- Cinema → 1 time
- Tennis ⇒ 2 time

$= 1 - \left((1/3)^2 + (2/3)^2 \right) =$ 0.444

Gini (weather = windy) → 4 instances → possible outcome
- 3 time Cinema
- 1 time Shopping

$= 1 - \left((3/4)^2 + (1/4)^2 \right) =$ 0.375

Gini (weather = Rainy) → 3 instance → possible outcomes
- 2 cinema
- 1 Stay In

$= 1 - \left((2/3)^2 + (1/3)^2 \right) =$ 0.444

Weighted Average $Gini_{(Weather)} = (0.44 * 3/10) + (0.375 * 4/10) + (0.44 * 3/10)$

$= 0.414$

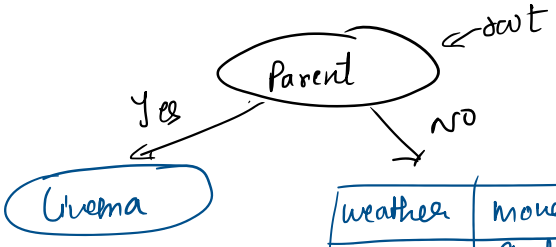
$= 0.486$

Gini (money) = 0.486
Gini (parent) = 0.36
Gini (weather) = 0.416

Minimum Gini ⟶ Minimum Impurity in Decision.

Here Minimum Gini Value = Gini (parent) = 0.36

So the root of Decision is Parent



weather	money	Decision
Sunny	Rich	Tennis
Rainy	Rich	Stayin
Windy	Poor	Cinema
Windy	Rich	Shopping
Sunny	Rich	Tennis

We need to find Gini (parent = No and weather)

Also Gini (parent = No and money)

Steps) Gini (parent = No and Weather) ⟶ possible value ⟶ Sunny – 2 time, Windy – 2 time, Rainy – 1 time

5 instances

Gini (parent = No and weather = Sunny) ⟶ 2 instance ⟶ Tennis = $1 - \left(\left(\frac{2}{2}\right)^2\right) = 0$ // possible Outcome

Gini (parent = No and weather = Windy) ⟶ 2 instances ⟶ Cinema, Shopping ⟶ $= 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 0.5$ possible Outcome

Gini (parent = No and Weather = Rainy) → 1 instance → Stayin = $1 - ((1/1)^2) = 0$ — possible outcome

Weighted Average Gini (parent = No and Weather) = $0.5 \times 2/5 = 0.2$.

Step 6) Gini (parent = NO and Money) — 5 instance

Possible values: Rich = 4 time; poor = 1 time

Gini (parent = NO and Money = Rich) → possible outcome

Tennis – 2

Stayin – 1

Shopping – 1

(4)

$= 1 - \left((2/4)^2 + (1/4)^2 + (1/4)^2 \right) = 0.625$

Gini (parent = NO and Money = Poor) → possible outcome

Cinema $= 1 - ((1/1)^2)$

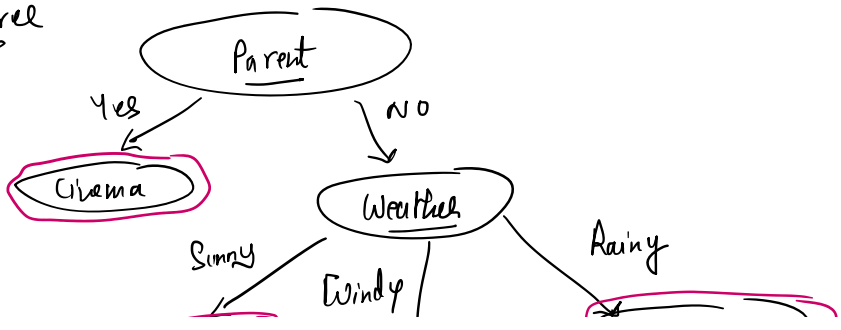
$= 0$

Weighted Average Gini (parent = No and Money) = $0.625 * 4/5 = 0.5$

Here Gini (parent = NO and Weather) has smallest value

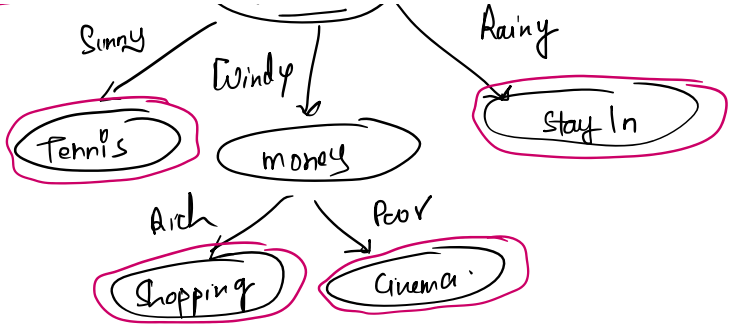
So now Next Node = Weather

* Updated Decision Tree



Ans.

Ans



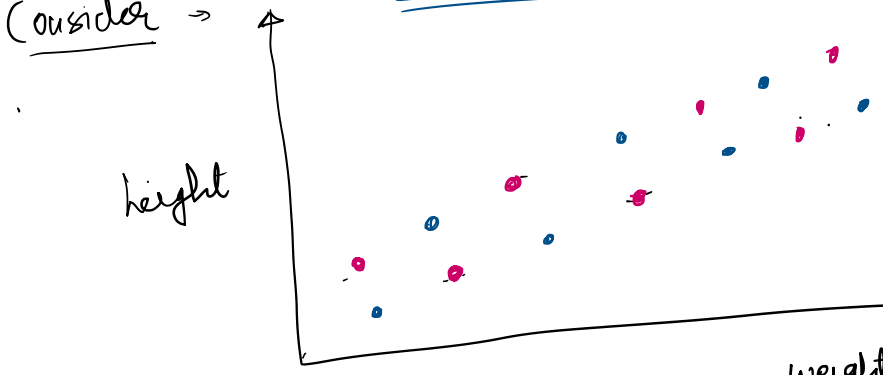
HW

Construct an optimal Decision Tree for following.

Outlook	Temperature	Humidity	Windy	Play (Decision)
Sunny	Hot	High	False	NO
Sunny	Hot	High	True	NO
Overlast	Hot	High	F	Y
Rainy	mild	High	F	Y
Rainy	Cool	Normal	F	Y
Rainy	Cool	Normal	T	N
Overlast	Cool	Normal	T	Y
Sunny	mild	High	F	N
Sunny	Cool	Normal	F	Y
Rainy	mild	Normal	F	Y
Sunny	mild	Normal	T	Y
Overlast	mild	High	T	Y
Overlast	Hot	Normal	F	Y
Rainy	mild	High	T	N

Total given Datapoints

Consider →

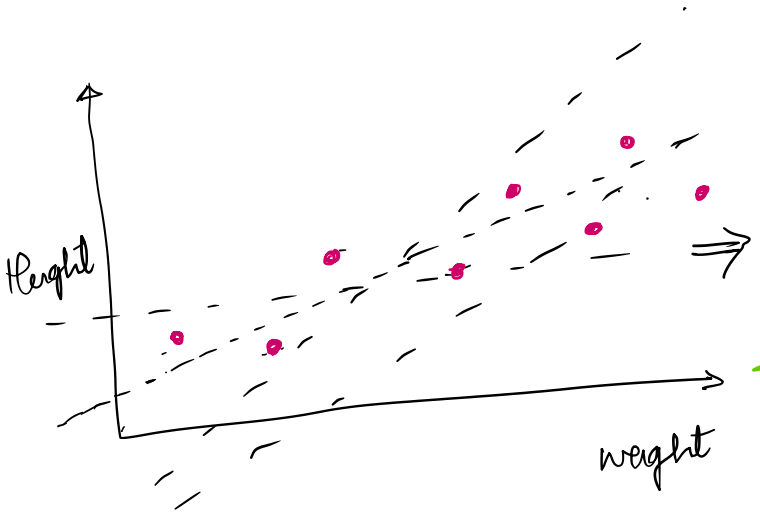


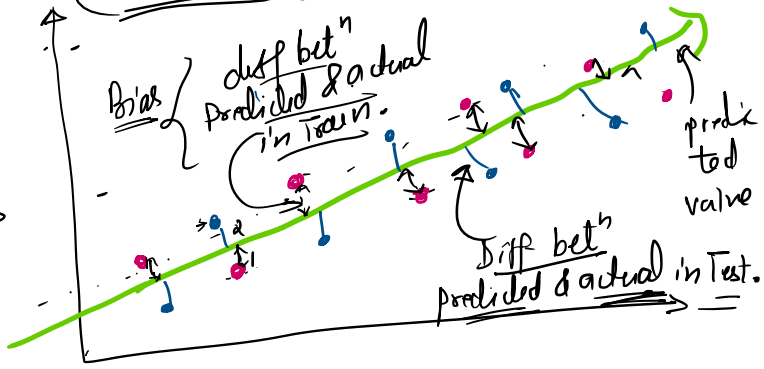
● → Test set
● → Train set

bias ↑ variance ↓

model be Consider Straight line

(Underfitting)





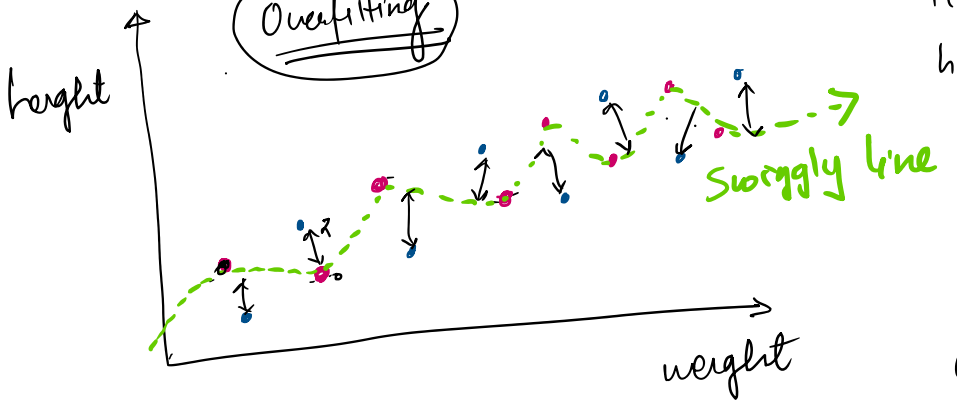
Train set

Bias → Inability of ML methods (Linear reg) to capture true relationship

Here any Simple Straight line cannot match all data points in Training Set

bias ↓ variance ↑

(Overfitting)



Here the squiggly line handles the true relationship bet^n weight & height

So Since the line almost passes through every training data points we say ⇒ Little/Zero

∴ Data bet^n

points we say ⇒ [illegible]

Zero

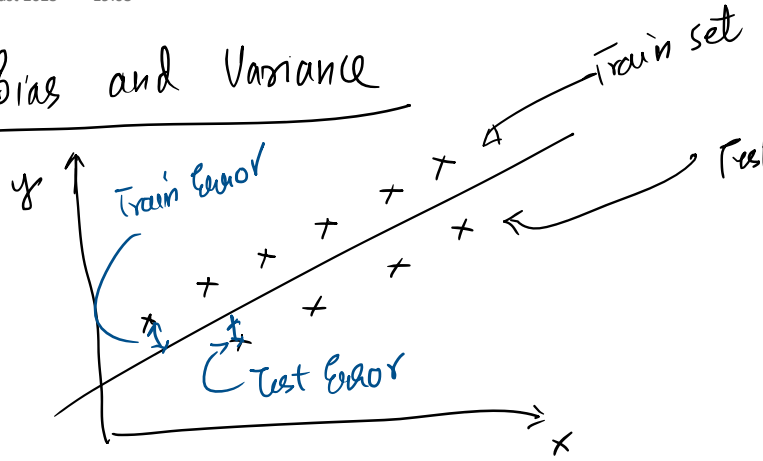
Bias.

Variance ⇒ Difference in Fits bet^n

the Datasets (Train & Test Data Sets).

09 August 2023 19:03

Bias and Variance



Straight line
↳ Polynomial degree = 1.

* Underfitting :- When the model is simple (lower degree polynomial), the model might not fit the train set data points.

So Here the difference bet^n Actual and predicted value for Train set is higher

This is bias.

So In Underfitting bias ↑

There will Error for predicted value and actual value for Test set

So the diff in Error for Test and Train set is low

This is Variance

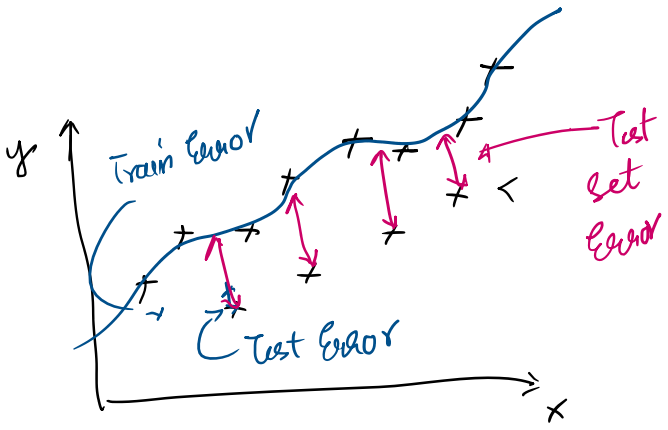
So In Underfitting Variance ↓

* Underfitting

- For Simple Model (lower order Polynomial)

Bias ↑ Variance ↓

Bias ↑ Variance ↓



Here model is complex
(higher order Polynomial)

→ So Error with Train set is very less

∴ Bias is ↓

Since the model perfectly fits the train set

It is case of Overfitting

and there is significant diff (Error) with Test set

∴ Variance is ↑

* Overfitting →

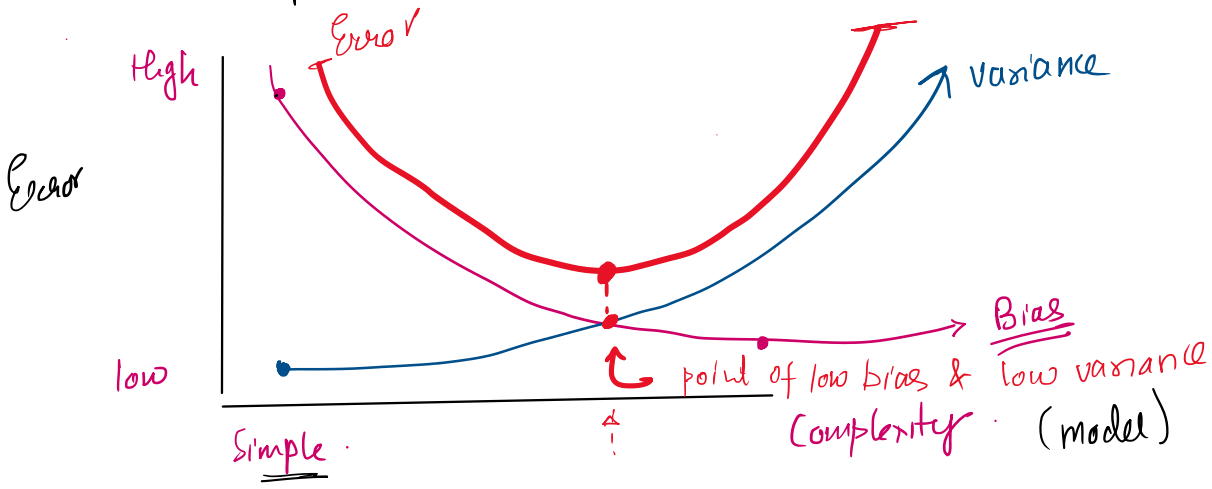
For Complex model (higher order polynomial)

bais ↓ variance ↑

Imp Bias - Variance Tradeoff

(1) If model is too simple and very few parameter → Underfitting
→ High bias
→ Low variance

(2) If model is complex and has large no of parameters → Overfitting
→ Low Bias
→ High Variance.



Bias Variance Tradeoff says

We need a model that gives

(1) low bias

(2) low variance.

ie we need to find point of low bias and low variance.

The methods to achieve this :-

* Regularization (Penalize 'Q' parameter)
* Boosting
* Bagging

* Used to Minimize the Total Error : $Bias^2$ + Variance + Irreducible Error.

We need to Minimize the Total Error: Bias + Variance + Irred Error.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$
$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

* Performance Metric :-

Consider a skewed Data set (One Sided).

Ex 1000 MRI Scan → 990 No cancer
→ 10 cancer

Here only 1 % Scan shows cancer

"it is very less -
Normally Allowed.
→ might be Ignored
→ But this is incorrect.

In Such Case to Evaluate Performance We may use

Confusion Matrix →

Predicted \ Actual	True	False
True	True +ve (TP)	False +ve
False	False -ve	True -ve

The above confusion matrix is OK for 2 classes.

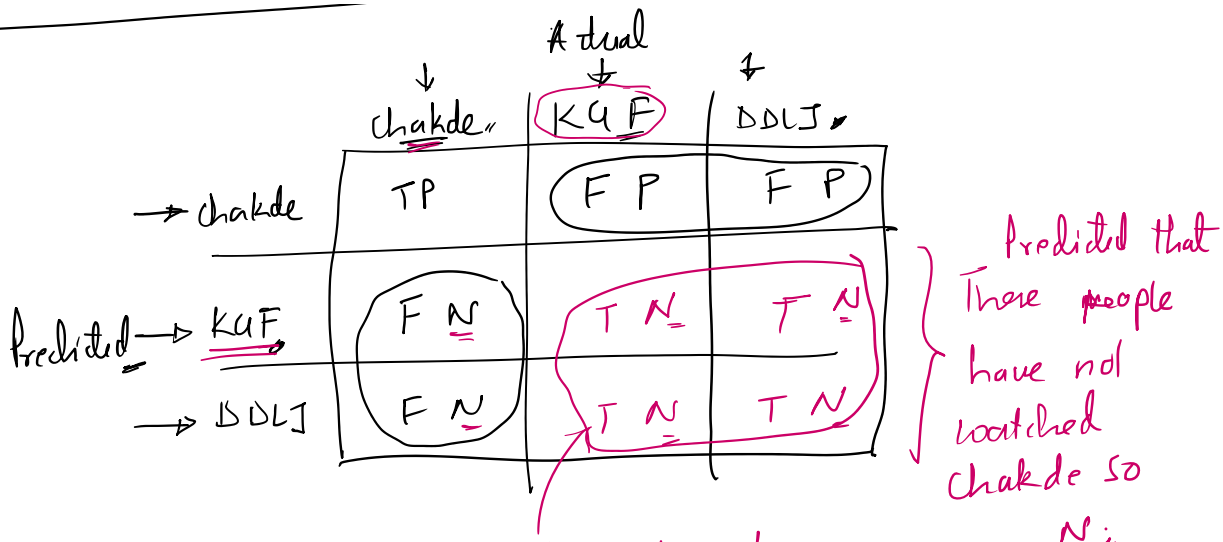
Question → Cancer ?

Predicted \ Actual	Has Cancer	Do not have cancer
(True) Has Cancer	TP	FP
Do not have cancer (-ve)	FN	TN

If we have More than Two classes.

Actual

Question → Person watched Chakde ?

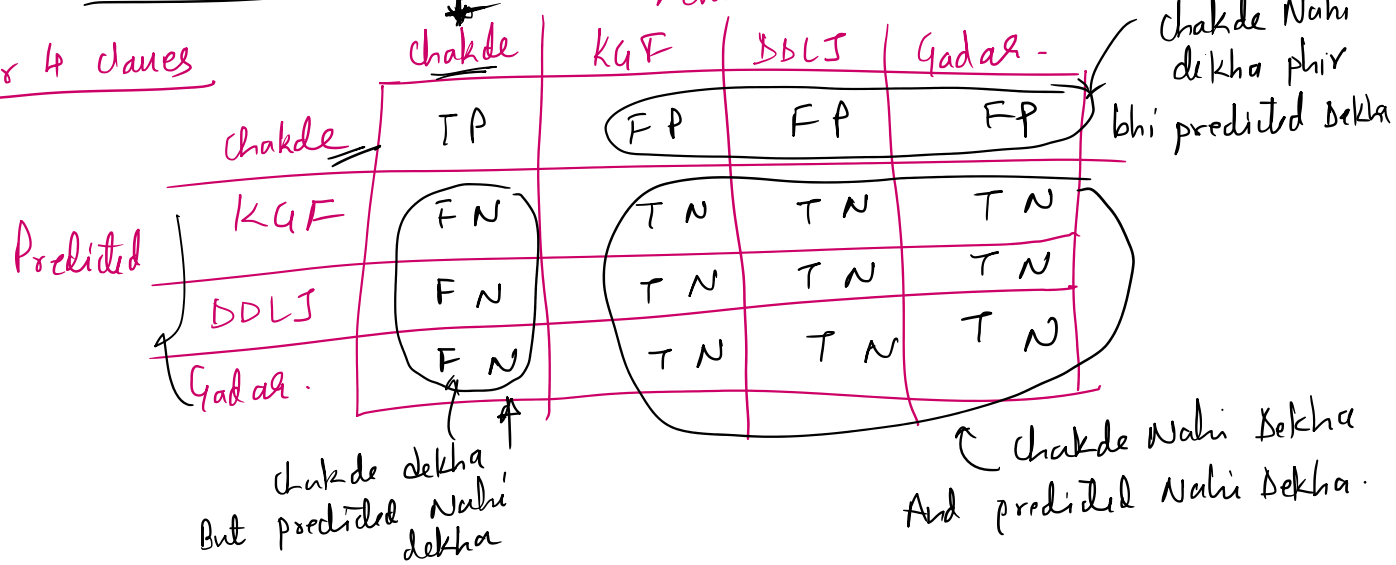


if person watched +ve Chakde

else -ve

Question → Watched Chakde? Yes +ve / No -ve.

For 4 classes



*

TP (True +ve) = Correctly Identified +ve (+ve Hai → +ve Predicted)

FP (False +ve) = Incorrectly Identified +ve (+ve Nahi Hai → +ve Predicted)

TN (True -ve) = Correctly Identified -ve ⇒ (-ve Hai → -ve Predicted)

FN (False -ve) ⇒ Incorrectly Identified -ve ⇒ (-ve Nahi Hai → -ve Predicted)

* 1) Accuracy = $\frac{\text{Total Correct Prediction}}{\text{Total Prediction}} = \frac{TP + TN}{TP + TN + FP + FN}$

2) Error Rate = 1 − Accuracy.

3) True +ve Rate (TPR) [Sensitivity / Recall] = $\frac{\text{Correctly Identified +ve}}{\text{Total Actual +ve}} = \frac{TP}{TP + FN}$

4) False Negative Rate (FNR). (Aisa +ve jisko -ve predict kiya) $\Rightarrow \frac{\text{Incorrectly Identified +ve}}{\text{Total Actual +ve}} = \frac{FN}{TP + FN}$

5) True Negative Rate TNR [Specificity] = $\frac{\text{Correctly Identified -ve}}{\text{Total Actual -ve}} = \frac{TN}{TN + FP}$

6) False Negative Rate [FNR] = $\frac{\text{Incorrect Identified -ve}}{\text{Total Actual -ve}} = \frac{FP}{TN + FP}$

[TPR = $\frac{\text{Correct +ve}}{\text{Total +ve}}$ (Jo h) ; TNR = $\frac{\text{Correct -ve}}{\text{Total -ve}}$ (Jo h)

FNR = $\frac{\text{Incorrect +ve}}{\text{Total +ve}}$ (Jo h) ; FPR = $\frac{\text{Incorrect -ve}}{\text{Total -ve}}$ (Jo h)]

Note To Identify Heart Disease

Sensitivity = TPR = $\frac{\text{Correct Identified +ve}}{\text{Total +ve}}$ = What percentage of patient with heart disease are correctly

Sensitivity = TPR = $\frac{\text{Correct Identified +ve}}{\text{Total +ve}}$ = What percentage of patients with heart disease are correctly Identified

Specificity = TNR = $\frac{\text{Correct Identified -ve}}{\text{Total -ve}}$ = What percentage of patients without heart disease are correctly Identified.

FPR = 1 − Specificity $= 1 - \frac{TN}{TN+FP} = \frac{TN + FP - TN}{TN+FP} = \frac{FP}{TN+FP}$

Training Error →

It is prediction Error that we get when we apply the model on Same data from where it is trained.

$$\mathcal{E}_{Train} = \frac{1}{n} \sum_{i=1}^{n} \text{Error}\left(f_0(x_i), y_i\right)$$

($f_0(x_i)$: prediction of x_i; y_i: actual value; sum for all the sample)

(2) Test Error: It is prediction Error we get when we apply model on altogether different data set (Test set) and not on the data on which it is trained.

$$\mathcal{E}_{Test} = \frac{1}{n} \sum_{i=1}^{n} \text{Error}\left(f_0(x_i), y_i\right)$$

for Test set //

(3) Generalization Error → also known as Out of Sample Error

→ Measure of how accurately an algorithm is able to predict outcome values for previously unseen data.

→ We want to know how the model will perform

→ We want to know how the model will perform on future data (we do not have today)

→ For Future we do not have X_i (input)
y_i (output)

$$\mathcal{E}_{gen} = \int_{\text{over all possible value of } x \& y} \text{Error}\left(f_0(x_i), y_i^0\right) \underbrace{P(y, x)}_{\text{How often we expect such } x \& y.} dx.$$

($f_0(x_i)$: Predicted, y_i^0: actual)

Usually

$$\mathcal{E}_{train} \leq \mathcal{E}_{gen}$$

as we do not have value of future $P(y, x)$

So we do not Compute generaliz^n Error,

we approximate it with Testing Error.

* To decide which one should we look for between Sensitivity & Specificity

* If identifying +ve is more important to us, then we will select Algo that has high sensitivity

* If correctly identifying -ve is more important then we will select algo that has high specificity

Precision & Recall :=>

→ Used for Information Retrieval.

→ Google Search Engine =>

→ query fired

→ Have millions of Related Records

→ From these top 10 - 100 records are returned.

$$\text{Precision} = \frac{\text{Correctly predicted +ve}}{\text{Total +ve Predicted}} = \frac{TP}{TP + FP}$$

Range (0 — 1)

$$\text{Recall} = \frac{\text{Correctly Identified +ve}}{\text{Total Actual +ve}} = \frac{TP}{TP + FN}$$

(TPR)
(range 0 - 1)

* Ideally we want precision to be high (i.e 1) for a good classifier

(range 0 - 1)

* Ideally we want precision to be high (i.e 1) for a good classifier

$$\text{Precision} = 1 = \frac{TP}{TP + FP} = 1 \Rightarrow \text{when } \boxed{FP = 0}$$

* Ideally we want Recall to be very high (i.e 1) for a good classifier

$$\therefore \text{Recall} = 1 = \frac{TP}{TP + FN} \Rightarrow \text{when } \boxed{FN = 0}$$

So Ideally a good classifier has High Precision & Recall

But in Reality there is trade-off

* When we tweak (update) our model to increase one, then the other decreases.

Q) Explain

F1-Score

* In reality we need a metric that takes into account both precision and recall.

* F1 score is a metric that takes into account both precision & recall.

* F1 score is harmonic mean of Precision & Recall.

$$\text{F1-Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Harmonic mean of two variables a & b

$$H = \frac{2}{\frac{1}{a} + \frac{1}{b}} = \frac{2ab}{a+b}$$

for n variables

$$H = \frac{n}{\frac{1}{n_1} + \frac{1}{n_2} + \cdots \frac{1}{n_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{n_i}}$$

If F1-Score = 1 ⇒ when Precision = 1, Recall = 1

* When Precision and Recall both are high then F1-Score is High

When to use F1-Score →

→ Accuracy is not a good metric to use when we have class imbalance.

Ex → Let say 99% of people visiting site are onlookers and not purchasing anything.

→ Suppose we have a model that predicts that 99% people visiting site are onlookers.

→ 1% Error is Acceptable

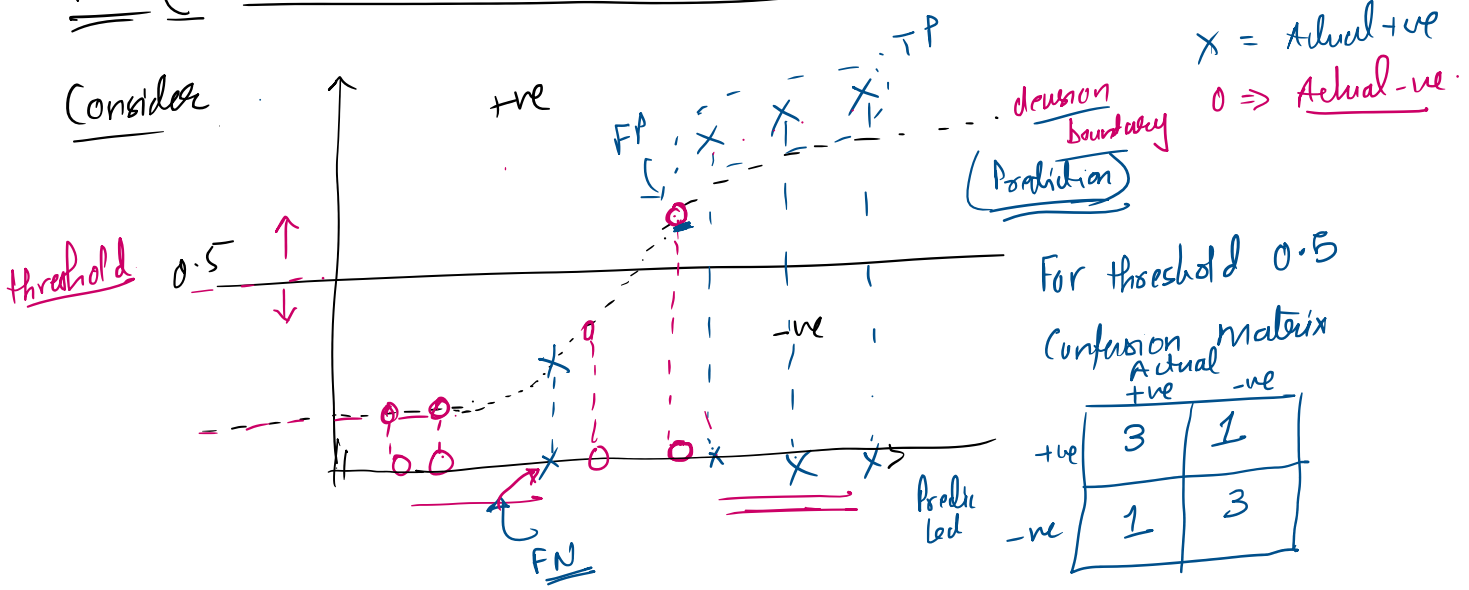
people visiting site are onlookers.

→ The model is 1% wrong, Generally 1% Error is Acceptable

→ But such model in this case is useless

→ In such case instead of accuracy, we will prefer F1-Score.

ROC [Reciever Operator characteristic]



For threshold 0.5

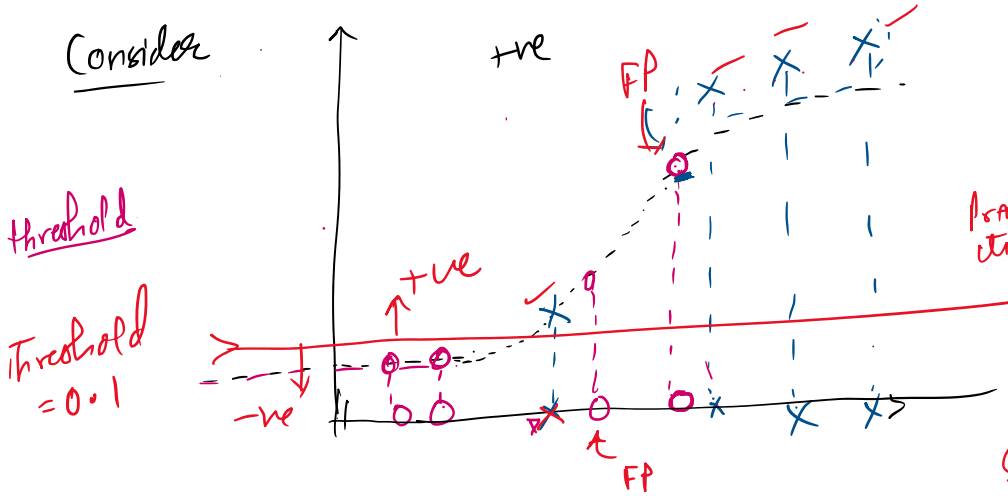
Confusion matrix

Predicted \ Actual	+ve	-ve
+ve	3	1
-ve	1	3

Sensitivity = $\frac{3}{4}$ TPR

Specificity = $\frac{3}{4}$ TNR

FPR = 1 − Specificity



Now threshold = 0.1

Confusion matrix

Predicted \ Actual	+ve	-ve
+ve	4	2
-ve	0	2

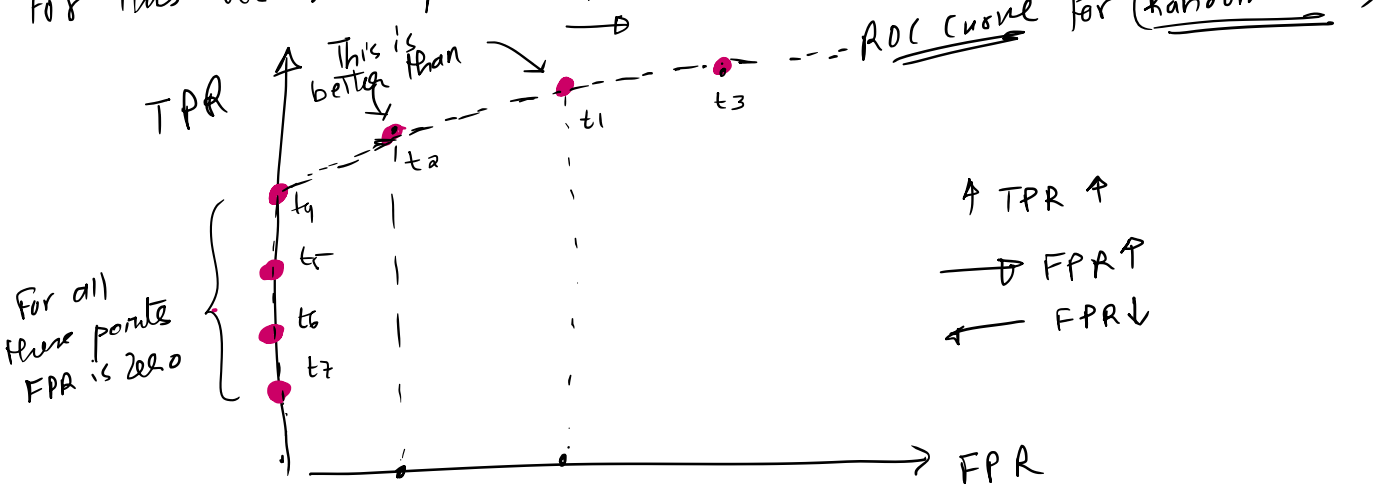
Sensitivity = $\frac{4}{4} = 1$

Specificity = $\frac{2}{4} = 1/2$

* Consider for logistic Regression, where we identify a threshold point and prepare confusion matrix and calculate Sensitivity & Specificity.

and calculate Sensitivity & Specificity

* If threshold changes then the confusion matrix and accordingly the Sensitivity and Specificity changes.
* We can have many such thresholds bet^n 0 → 1
* We want to analyze the performance at diff threshold and want to identify the best of it.
* For this we will plot TPR and FPR for diff threshold.



→ From the above ROC Curve for Random Forest model, it will help us to find the best threshold

Q) Explain

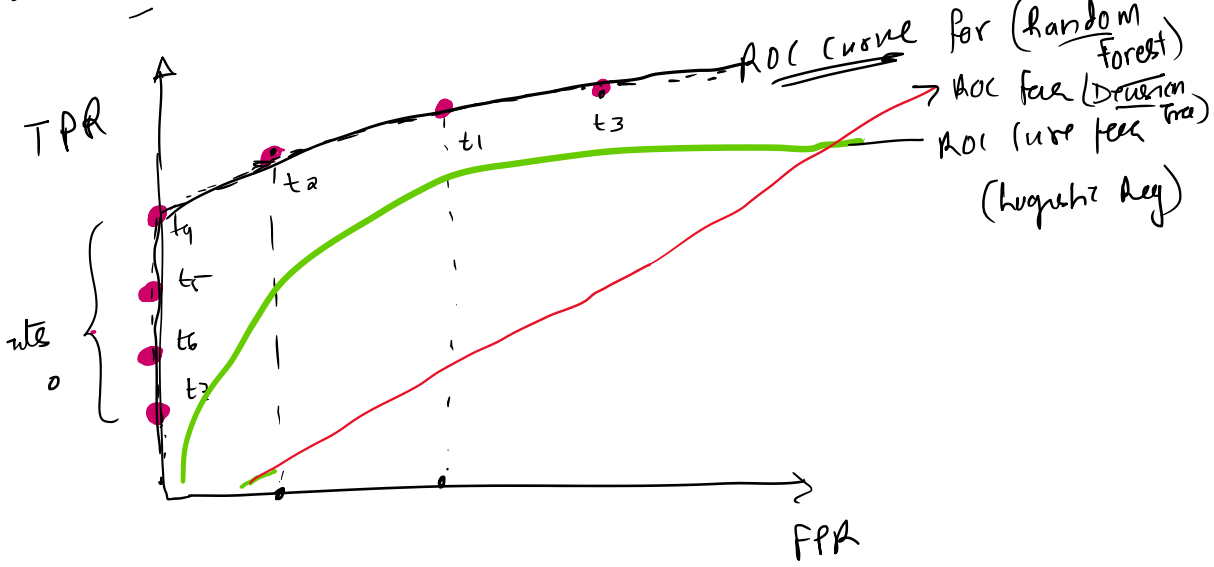
AUC [Area Under Curve] → It is a method to compare ROC for more than one method and will help to judge which one is better.

→ Here will find Area Under ROC for each method.

→ ROC that has maximum Area, the corresponding method will be best.

ROC Curve for (Random Forest)

→ ROC that has maximum Area, the corresponding



From above ROC Curves, we find that there is maximum area under ROC Curve for Random forest, hence the method Random forest will be the best method.

Q) Explain Kappa Statistic?

* Kappa statistic or Cohen's Kappa is statistical measure of inter-rater reliability for categorical variable.

* It is used when two/more raters apply a criteria based on a tool to assess whether or not some condition occur.

Ex Let say Two doctors rates whether or not each of 20 patients has diabetes based on symptoms

* If two raters uses same criteria on same target to evaluate and then their agreement is very high then we will have evidence of reliable rating

* If their agreement is not very high then →
  - → Either criterion tool is not useful
  - → or raters are not trained enough.

* Kappa statistics corred for chance agreement and not percent agreement

Ex

Evalor B \ Evalor A	Yes	NO
Yes	35	20
NO	15	40

⇒ 35 times both agreed – said Yes
⇒ 40 times both agreed → said NO
⇒ 20 time A said NO but B said Yes
⇒ 15 time A said Yes but B said NO.

⇒ 20 time A said NO but B said YES

⇒ 15 time A said Yes but B said NO.

Cohen Suggested following Statistics. ⇒

value ≤ 0	⇒	No agreement
0.01 → 0.20	⇒	as none to slight
0.21 → 0.40	⇒	as fair
0.41 → 0.60	⇒	as moderate
0.60 — 0.80	⇒	as Substantial
0.81 — 1.00	⇒	perfect agreement.

* Rather than Calculating the percentage of items, the raters agreed on Cohen's Kappa attempts to account the fact that raters may happen to agree on some items purely by chance

Ex Two Curators asked to rate 70 paintings.

Curator C_1 \ Curator C_2	Yes	NO
Yes	25	10
NO	15	20

Step (1) Calculate Relative Agreement bet^n Curators.

$$P_o = \frac{\text{Both Said Yes + Both Said NO}}{\text{Total}} = \frac{25+20}{70} = 0.6429$$

Step 2 > Calculate hypothetical probabilities of chance Agreement bet^n Curators.

$$P(\text{Yes}) = C_1(\text{Yes}) * C_2(\text{Yes}) = \frac{(25+10)}{70} * \frac{(25+15)}{70} = 0.2857$$

$$P(Yes) = \frac{C_1(Yes)}{\text{Total Rus}} * \frac{C_2(Yes)}{\text{Total Rus}} = \frac{(25+10)}{70} * \frac{(25+15)}{70} = 0.2857$$

$$P(No) = \frac{C_1(No)}{\text{Total Rus}} * \frac{C_2(No)}{\text{Total Rus}} = \frac{(15+20)}{70} * \frac{(10+20)}{70} = 0.214285$$

$$P_e = P(Yes)+P(No) = 0.2857 + 0.214285 \approx 0.5$$

Calculate Kohen's Kappa $= K = \frac{P_o - P_e}{1 - P_e} = \frac{(0.6429 - 0.5)}{1 - 0.5} = 0.2857$

It is in range $0.21 \longrightarrow 0.40$ so the agreement bet^n two Cuthor is fair

16 August 2023 11:30

Module - 3

Ensemble learning →.

3.1 Understand Ensembles

K-fold Cross Validation

→ Boosting

Stumping (AdaBoost)

XGBoost

3.2 → Bagging

Subagging

Random Forest

Comparison with Boosting

Different ways to Combine Classifiers

* Ensemble. ->

* In ML, ensemble is a model that combines the prediction from two or more models.

* The models that contributes to Ensemble are known as ensemble members.

* The members may or may not be trained on same training data and they may be of same type or different type.

* It is very powerful method to improve the performance of the model

* It is technique that uses group of weak learners in order to create a strong and aggregated learner.

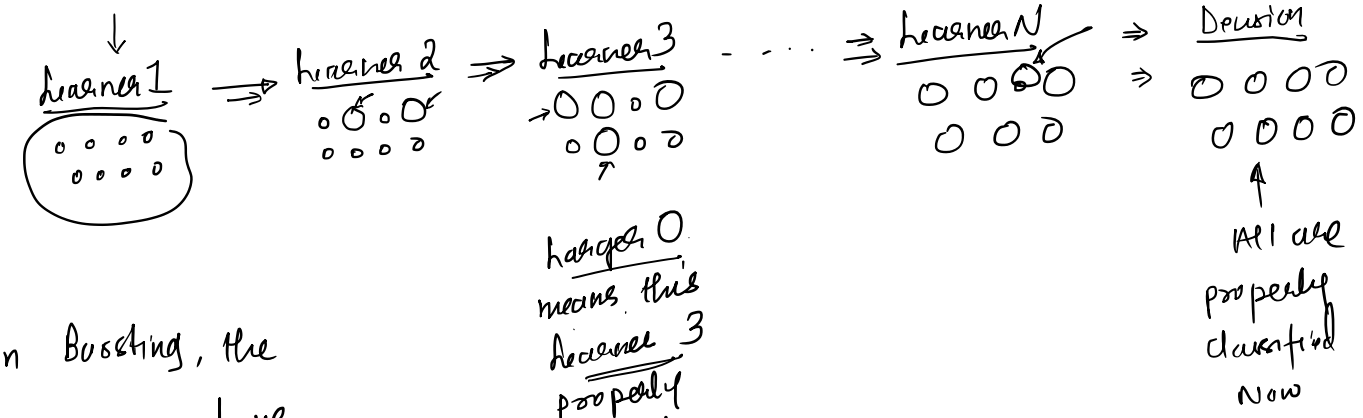
-> The Ensemble technique helps to reduce the Variance (By Bagging) and Bias (By Boosting) and thus helps in improving the predictions.

-> Boosting model :- * It falls inside family of Ensemble method.

* It consist of filtering or weighting the data that is used to train team of weak learners, so that the new learner can give more weight on sample that is poorly classified by previous learner.

-> In Boosting the learners are trained

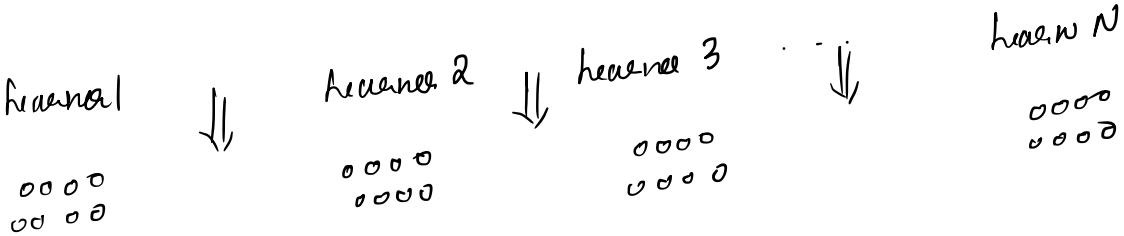
Sequentially.



Learner 0 means this Learner 3 properly classifieds the poorly classified result of learner 2

* In Boosting, the learners can have diff importance or weight (represented by diff sizes of circles).

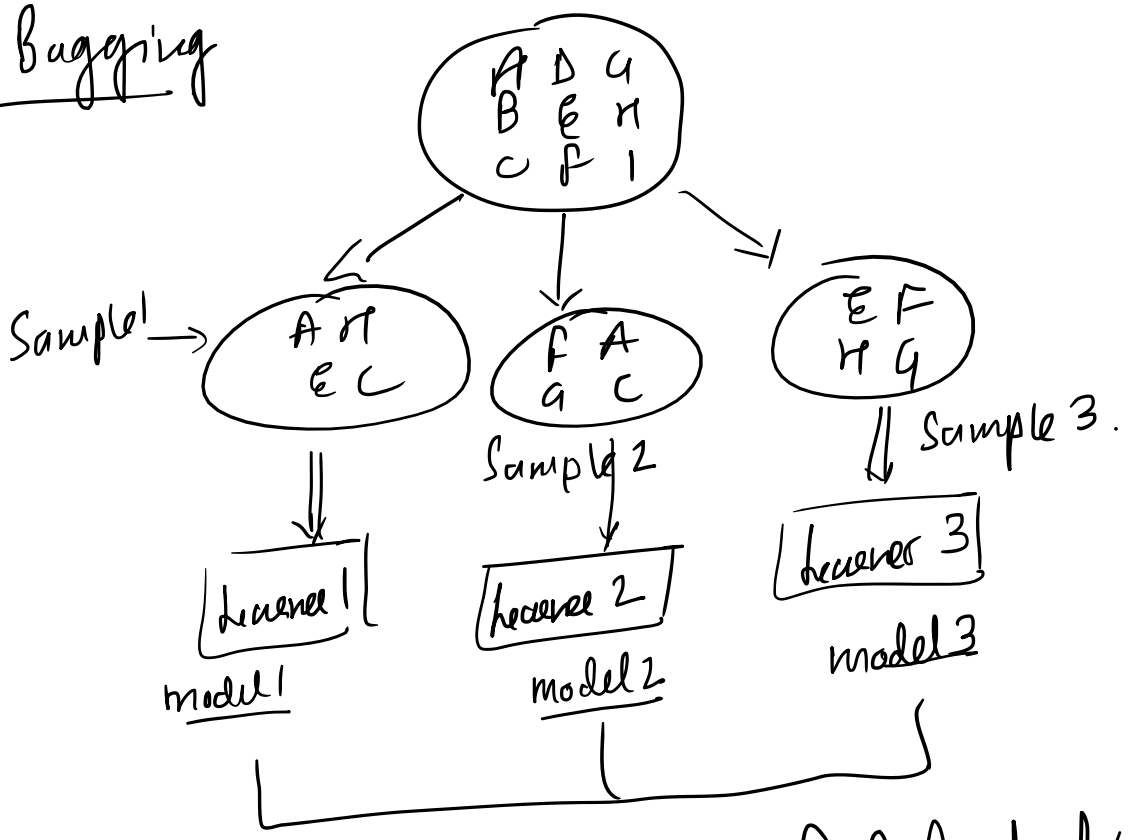
Bagging:



⇒ In Bagging the weak learners are trained in parallel using randomness

⇒ All learners have same weights.

Note Bagging



* Bagging helps in reducing variance
* Boosting helps in reducing Bias.

Cross Validation :→

→ In Supervised M.L

→ Train a model on a Dataset

→ Trained model is used to predict the target given new sample.

→ How to know if model we have trained will produce effective and accurate result on new input

Cross Validation → It is process that ensures the model will perform well on new Data.

Split the Dataset
- Training Set (60%)
- Validation Set (Hold-out set) (20%)
- Test Set (20%)

Training Set = part of data on which model is trained.
(This dataset will help to * build model)

***Validation Set** → * Evaluate the Model
→ Will help to chk if model overfits or Underfits.
→ Update the parameters and again train the model.
→ Repeat this until the model performs best on Validation set

→ typical [illegible]

Validation set

Test Set ⇒* Prediction.

→ The fully trained model after being evaluated on validation set can be used on test set to generate Estimation.

Q) Types of Cross Validation ⇒

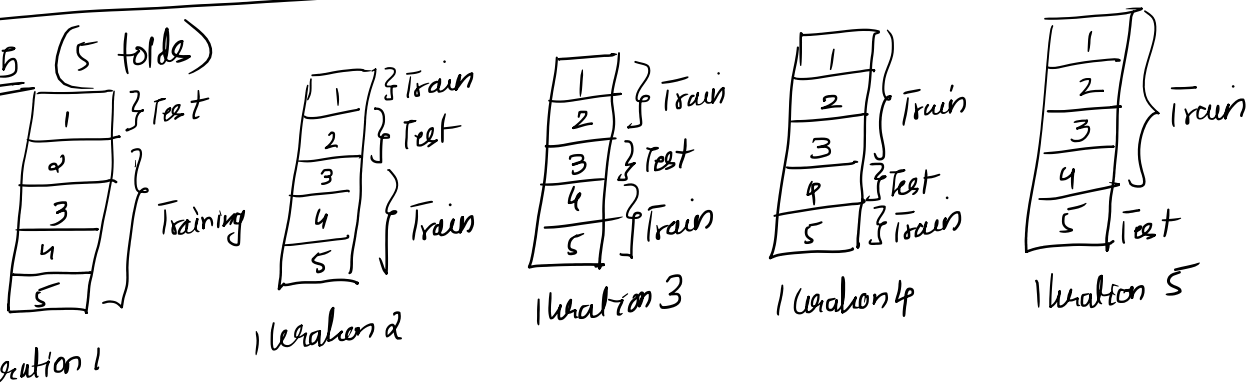
① The standard Validation Set Approach

② (LOOCV) Leave One Out Cross Validation.

③ K-fold Cross Validation.

K-fold Cross Validation ⇒

Ex K=5 (5 folds)

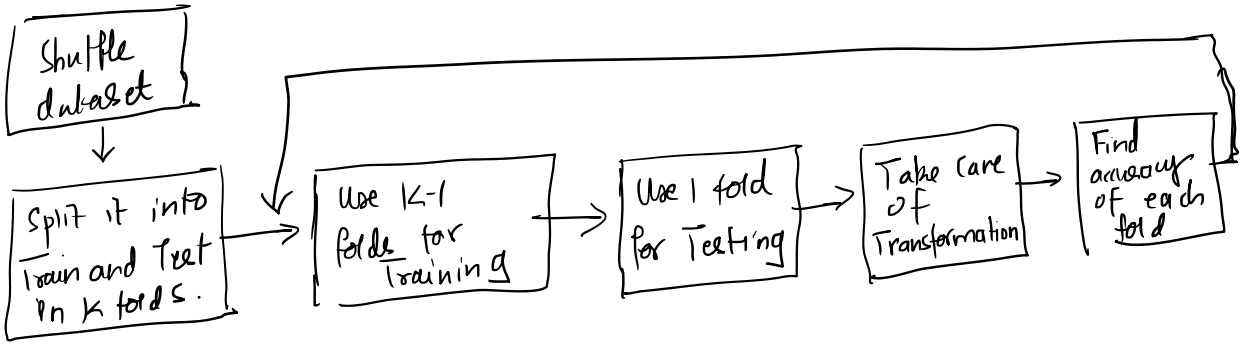


K fold ⇒ → K fold helps us to build the model in generalized form.

→ To achieve this K fold Cross Validation splits the dataset into Training Testing & Validation.

→ Here Test and Train data will support building the model.

→ Life cycle of K-fold Cross Validation.



* The No of iterations ideally is k time
* Finding mean of accuracy score of each iteration will give the consistency of the Trained model.

Rules

(1) $k \geq 2$

if k == 2 → just 2 iterations.

if $k = n >> 2$ ⇒ n-1 for Training
1 for Testing

(2) most commonly used value of $k = 10$

(3) If k is very large then the running time of process will increase.

(4) The value of k is inversely proportional to size of data i.e if dataset size is small then number of folds can increase.

23 August 2023 19:21

Bagging ⇒ Random Forest

* Random Forest is Example of Bagging.

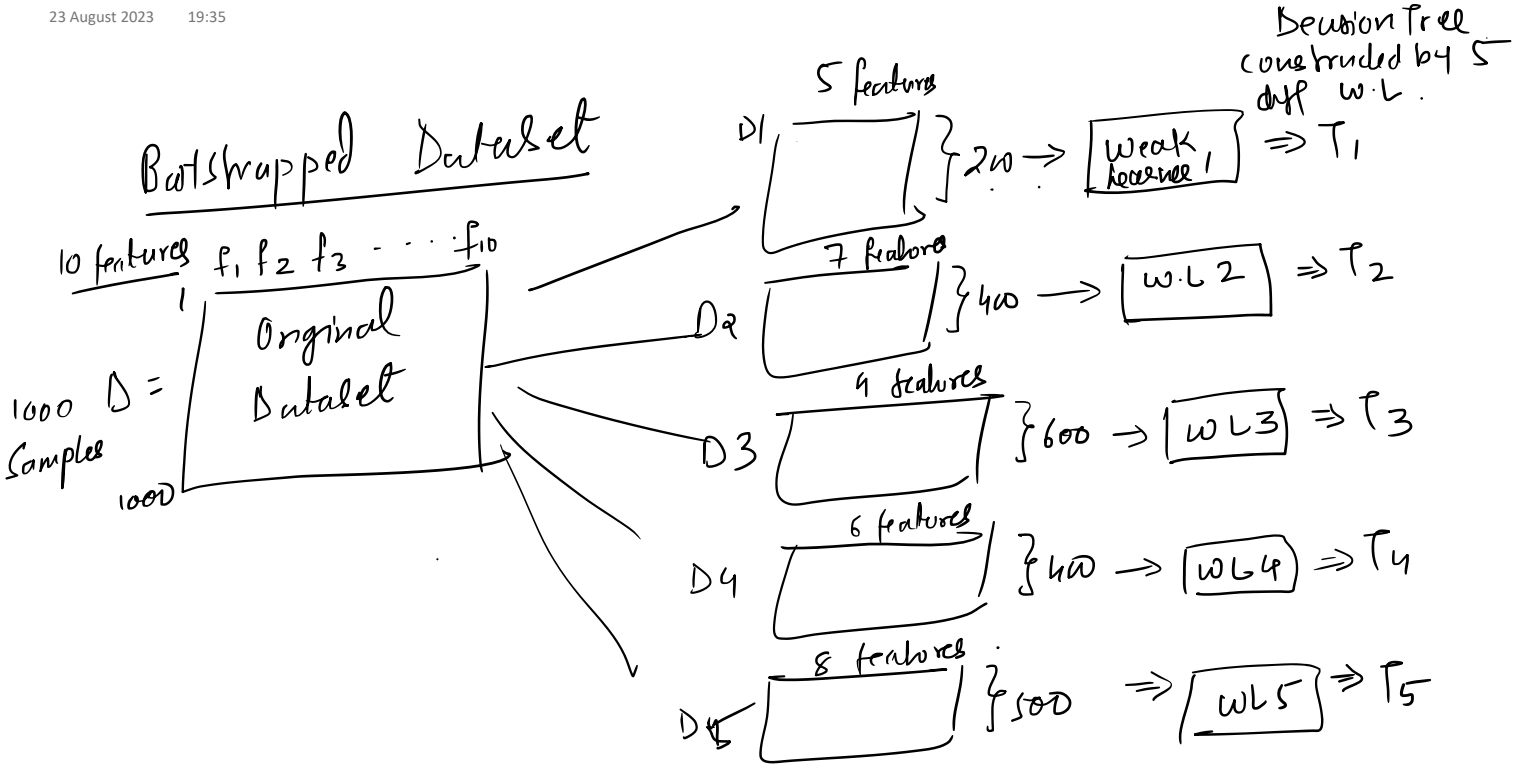
* You must Know Decision Tree Construction [Gini Index]

* Decision Tree
  - Used for Regression ⇒ Regression Tree
  - Used for Classification / Categorization ⇒ Classification Tree.

* Decision Tree is easy to Build and Interpret in practical.

* The Decision Tree is not very accurate on new test sample

* Random Forest Combines Simplicity of Decision Tree with flexibility resulting into vast Improvement in accuracy.



We have 5 decision Tree Constructed one each by a weak learner.

Now Test Sample = S_{test}

Now S_{test} is subjected to each of the 5 Decision Tree →

Suppose

$T_1(S_{test})$ = Yes
$T_2(S_{test})$ = NO
$T_3(S_{test})$ = Yes
$T_4(S_{test})$ = Yes
$T_5(S_{test})$ = NO

In 5 cases 3 cases are Yes
2 case are NO

Final Prediction Yes.

* Construct Random forest

Original Set

	Chest Pain	Good Blood Circulation	Blocked Arteries	Weight	Heart Disease
S1	NO	NO	NO	125	NO
S2	Yes	Yes	Yes	180	Yes
S3	Yes	Yes	NO	210	NO
S4	Yes	NO	Yes	167	Yes
	NO	Yes	Yes	170	?

Step1 => Create BootStrapped DataSet

1. could be / not be of same size
2. Samples are randomly Selected
3. Allowed to pick same sample more than once.

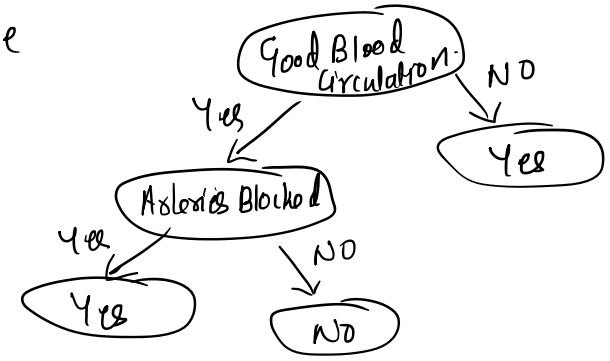
BootStrapped DataSet

Chest pain	Good Blood Circulation	Blocked Arteries	Weight	Heart Disease
Yes	Yes	Yes	170	Yes
NO	No	NO	125	NO
Yes	NO	Yes	167	Yes
Yes	NO	Yes	167	Yes

Step2 => Create a Decision Tree Using BootStrapped dataset but Use random subset of Variables (features/columns)

Let us Consider we use only 2 features
(Good Blood Circulation and Blocked Arteries)

Let the Tree be (Gini Index)



Step 3 => Go to step1 and Repeat

* Ideally we repeat it for 100 times
* So we have $T_1, T_2 \ldots T_{100}$ (large no of Trees)
* Each time the Tree Constructed is a weak learner.
  [As all the features and samples are not considered while Tree Construction].
* Now we have Random Forest of 100 Trees and will be more effective than Individual Decision Tree.

6) How To Use the Random Forest (Here 100 Trees) ->

Consider a Test Sample

Chest pain	Good Blood Circulation	Blocked Arteries	Weight	Heart Disease
Yes	NO	NO	168	?

Here we will Run it through each of 100 Trees and will note the Prediction

Let say of 100 Trees < 80 Trees Predicted Yes.
20 Trees Predicted NO

Answer = Yes

Answer = Yes

9) To find Random Forest is Effective or Not ?

* There might be some Sample not considered by any of Bootstrapped Dataset.
* We will create a New Dataset with such samples this is known as "Out of Bag" Dataset
* Now we will chk how many samples from out of Bag Dataset are predicted correctly
* Numbers of Incorrectly Predicted Out of Bag Samples = Out of Bag Error.

10) How to decide on Number of Columns to Use for Building Trees in Random Forest ?

(1) In our Case Dataset has 4 features

→ Create a Random Forest of Trees using 2 Features = F_1 (100 Trees)

→ Create a Random Forest of Trees using 3 features = F_2 (100 Trees)

(2) Now Chk the Accuracy of Out of Bag Data on each of the Random Forest

③ Use the one that gives More Accurate Answer.

Summarize

step 1. Create a Bootstrapped Data Set

step2 :- Create a Dession Tree using Subset of features

step 3 : Repeat step 1 and 2 approx 100 times.

step 4 : Use out of Bag Data Set to determine out of Bag Error

step 5 : If out of Bag Error is high then Repeat step 1 to step 4 until out of Bag Error is considerably low.

Note => To Compare the Peaformance for diff Random Forest Create Diff Random forest using different number of features

Use the one that gives Highest Accuracy.

ADABOOST (ADAPTIVE BOOST)

→ (Tree Stumping)

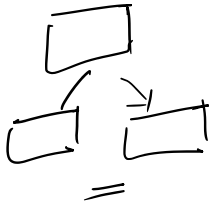
* Adaboost Creates forest of Trees from scratch and then it uses it to make classification.

* [In Random Forest Everytime we make a Tree with different depth (depending on number of features considered).

  * There is no predetermined maximum depth of Tree.]

  * In ADABOOST, the forest of Trees are usually just node and Two leaves

  * A Tree with one node and two leaves only is known as Tree Stump.



* In ADABOOST we have forest of stump instead of Trees.

* Stump alone is not great for classification as it takes only one parameter to make decision.

* Stumps are weak learners.

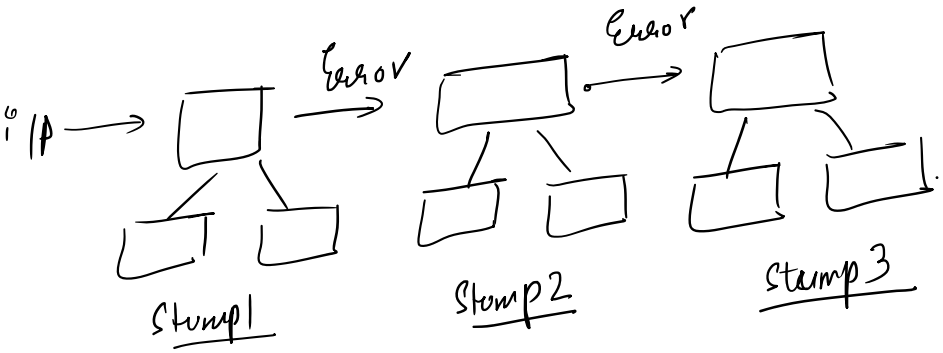
* ADABOOST Combines many stumps.

* In Random forest each tree contribute equally in final decision.

more can in final

Final decision:

* In ADABOOST, some stump may get more say in final decision than others.
* In Random Forest the decision made by Trees are Independent to other, so the order of tree generation is not important
* In ADABOOST, we have forest of stump and hence order is Important
* Here the Errors of first stump is Corrected by next stump and so on.



* Here Error of one stump influences the other stump so the stumps are generated in Sequences.

Idea Behind ADABOOST. (Boosting Technique → reduce Bias)

1. Adaboost combines lot of weak learners to make classification
2. The weak learners are almost always stump.
3. Some stumps will have more say in classification.

a the ...

3. Some stump will have more say in classification than other stump.

4. Each stump is made by taking previous stump mistakes into account

30 August 2023 18:39

6) Working of ADABOOST with Example.

Consider

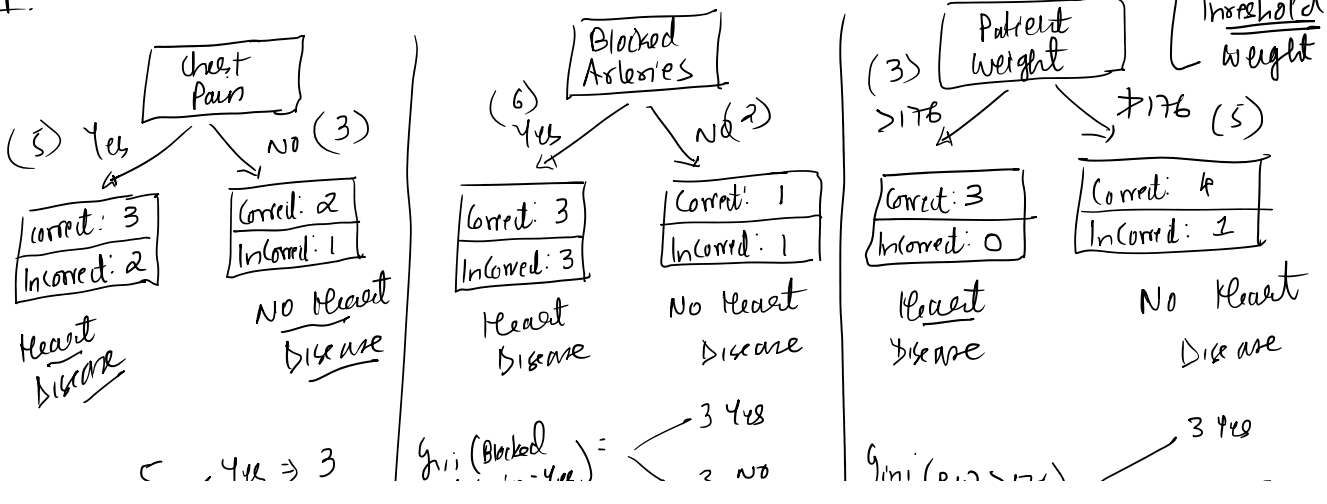
Chest Pain	Blocked Arteries	Patient weight	Heart Disease
Yes	Yes	205	Yes
NO	Yes	180	Yes
Yes	NO	210	Yes
Yes	Yes	167	Yes
NO	Yes	156	NO
NO	Yes	125	NO
Yes	NO	168	NO
Yes	Yes	172	NO

Step 1: Assign weight to every sample (Initial) $= \frac{1}{\text{Total No of Sample}} = \frac{1}{8}$

*(Initially Equal Weights are assigned to each Sample)

Chest Pain	Blocked Arteries	Patient weight	Heart Disease	Sample weight
Yes	Yes	205	Yes	1/8
NO	Yes	180	Yes	1/8
Yes	NO	210	Yes	1/8
Yes	Yes	167	Yes	1/8
NO	Yes	156	NO	1/8
NO	Yes	125	NO	1/8
Yes	NO	168	NO	1/8
Yes	Yes	172	NO	1/8

Step 2: Create Stump on Each Feature



Div:

Gini (Chest Pain = Yes) = 5 → Yes ⇒ 3, No ⇒ 2 (Heart Disease)

$= 1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right)$

$= 0.48$

Gini (Chest Pain = No) = 3 → Yes = 2, No = 1

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

$= 0.444$

Weighted Gini (Chest Pain)

$= 0.48 * \frac{5}{8} + 0.44 * \frac{3}{8}$

$= 0.4665 \approx 0.47$

Gini (Blocked Arteries = Yes) = 3 Yes, 3 No

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

$= 0.5$

Gini (Blocked arteries = No) = 1 Yes, 1 No

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

$= 0.5$

Gini (Blocked Arteries) = 0.5

Gini (P.W > 176) = 3 Yes, 0 No

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

$= 0$

Gini (P.W ≯ 176) = 4 Yes, 1 No

$= 1 - \left(\left(\frac{4}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right)$

$= 0.32$

Gini (weight) = 0.2

* The smallest Gini Index is 0.2 for weight > 176 feature

So first Stump will be weight > 176

Step 3 =>

To calculate Amount of Say of weight > 176 [feature].

$$\text{Amount of say} = \frac{1}{2} \ln\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$

Total Error: for a stump the Sum of weights associated with Incorrectly classified sample.

Here only one sample is Incorrectly classified.

Yes	Yes	167	Yes	1/8

∴ Total Error = 1/8

* Amt of Say = $\frac{1}{2} \ln\left(\frac{1-(1/8)}{(1/8)}\right) = \frac{0.97}{(\text{High})}$

[Note : If Amt of Say is less then the stump is not doing good job]

* Now we have → First stump (weight > 176)
→ Amt of say.

Now Initially Every Sample had Equal weight →

NOW Before Second stump we must modify weights of Samples

* Weights of Correctly classified Samples ⇒ must be decrease
" " Incorrectly classified Samples ⇒ must be increase
[So that next stump will focus more on Incorrectly identified Samples].

*

New Sample Weight (Correctly classified Sample) = Sample weight (old) * $e^{-\text{amount of say}}$

New Sample Weight (Incorrectly classified Sample) = Sample weight (old) * $e^{+\text{amount of say}}$

∴ for Incorrectly Classified Sample = $\frac{1}{8} * e^{+0.97} = 0.33$

For Correctly Classified Sample = $\frac{1}{8} * e^{-0.97} = 0.05$

Chest Pain	Blocked Arteries	Patient weight	Heart Disease	Sample weight	New Sample weight	Normalize Sample weight
Yes	Yes	205	Yes	1/8	0.05	0.05/0.68 = 0.07
NO	Yes	180	Yes	1/8	0.05	0.07
Yes	No	210	Yes	1/8	0.05	0.07
Yes	Yes	167	Yes	1/8	0.33	0.33/0.68 = 0.49

Yes.	No	210	Yes.	1/8	0.05	
Yes.	Yes	167	Yes	1/8	0.33	0.33/0.68 = 0.49
No	Yes	156	No	1/8	0.05	0.07
No.	Yes	125	No.	1/8	0.05	0.07
Yes.	No	168	No	1/8	0.05	0.07
Yes	Yes	172	No	1/8	0.05	0.07

Sum = 0.68

* Now we have New Normalized Sample weights.

* The Incorrectly classified Sample has Higher New Sample weight
  → which means Stump 2 will focus more on it.

* The Correctly classified Sample has comparatively lower Sample weight
  → which means stump2 will focus less on it.

* Similarly we can have Many such Stumps

* This is how Adaboost Creates and Uses Stumps.

* Let us chk how forest of stumps made by Adaboost does Classification →

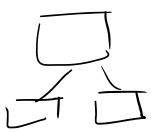
(1) Let us consider a test sample

(2) we will pass the test Sample through each stump

(3) Let there be few stumps that Classifies Sample as Yes (Has Heart Disease) and few stumps Classifies Sample as No (Do not have heart disease).

(4)

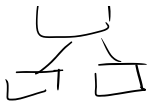
Has Heart Disease



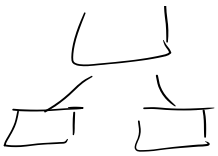
⇒ Amt of Say = 0.47

Has Not Heart Disease

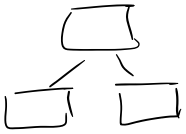
= Amt of Say = 0.41



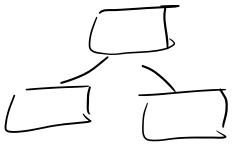
Say = 0.47



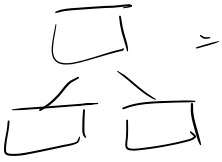
= Amt of Say = 0.82



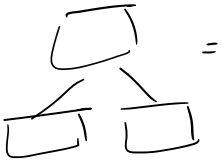
= Amt of Say = 0.32



Total Amt of Say (NO) = 1.23.



= Amt of Say = 0.78



= Amt of Say = 0.63.

Total amt of Say (Yes) = 2.7

Total Amt of Say for Stumps that classifies as Heart Disease (Yes) is greater

∴ The Test Sample is classified as Yes (Has Heart Disease).

Summary of Adaboost

1. Assign Sample weight (Initial Equal)
2. Create Stump for each feature.
3. Use Gini Index to Identify first stump.
4. For first stump.
   1. Calculate Total Error
   2. Amt of Say.

(4) For first stump.
   (1) Calculate Total Error
   (2) Amt of say.

(5) Update the sample weights for each sample

For Incorrectly classified = updated weight = old weight $* e^{+\text{Amt of say}}$.

For Correctly classified = updated weight = old weight $* e^{-\text{Amt of say}}$.

(6) Now Specify updated weight for each sample & Normalize the weight to generate New Sample weight.

(7) Identify New stump based on New Sample weight & Repeat step 3 to 7

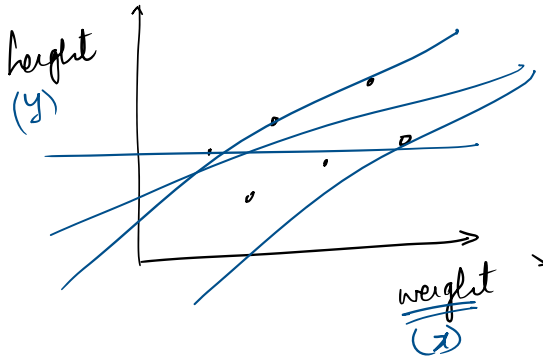
(8) For Classification (Testing)

8.1 ⇒ Run the test sample through all the stumps.

8.2 ⇒ Calculate Amt of say for sample classifying Yes & No

8.3 ⇒ Classify the test sample based on largest sum of Amount of say

Gradient Descent →



parameters

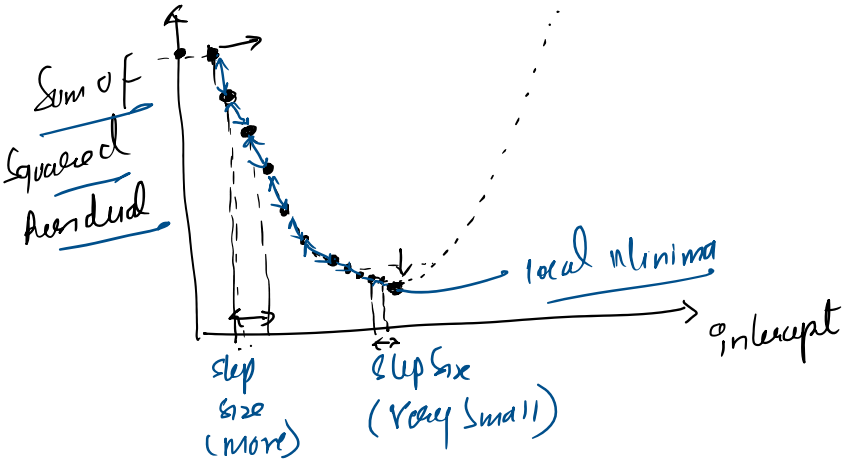
predicted height = intercept + slope * weight

$y = c + mx$

* Here we can have many line with different intercept and slope values

* We want a line with intercept and slope such that the Cost function should give Minimum value.
* Let the cost function be Sum of Squared Error (Residual Error)

* Let us Consider only intercept
* Lets plot a graph of Intercept V/s Sum of Squared Residual Error.



To find Minimum value for Sum of Squared Residual Error, we will start with very small value of intercept and will increase the value by very small amount.

Ex

x	y
0.5	1.4
2.3	1.9

Let the cost = Sum of Squared Error Fn be (Residual).

Ex

0.5	1.4
2.3	1.9
2.9	3.2

↑ ↑ actual

Let the cost = Sum of squared ... Fn be (Residual).

$$\text{Sum of Squared Residual Error} = (1.4 - (\text{intercept} + \text{slope} * 0.5))^2 + (1.9 - (\text{intercept} + \text{slope} * 2.3))^2 + (3.2 - (\text{intercept} + \text{slope} * 2.9))^2$$

Here we want to find value of intercept and slope that give minimum Sum of Squared Error.

Step 1 → Find $\frac{d}{d\,\text{intercept}}$ (Sum of Squared Residual Error)

$\frac{d}{d\,\text{slope}}$ (Sum of Squared Residual Error).

$$\text{Sum of Squared Residual Error (SSE)} = (1.4 - (\text{intercept} + \text{slope} * 0.5))^2 + (1.9 - (\text{intercept} + \text{slope} * 2.3))^2 + (3.2 - (\text{intercept} + \text{slope} * 2.9))^2$$

Using Chain Rule $\left[\frac{d}{dx} x^2 = 2x \frac{d}{dx} x\right]$

$$\frac{d(\text{Sum of Squared Error})}{d\,\text{intercept}} = 2 * (1.4 - (\text{intercept} + \text{slope} * 0.5)) * \frac{d}{d\,\text{intercept}}(1.4 - (\text{intercept} + \text{slope} * 0.5))$$

$\frac{d}{d\,\text{intercept}}(\text{intercept} + \text{slope} * 0.5) = (-1)$

$$+ 2 * (1.9 - (\text{intercept} + \text{slope} * 2.3)) \frac{d}{d\,\text{intercept}}(1.9 - (\text{intercept} + \text{slope} * 2.3))$$

$\frac{d}{d\,\text{intercept}}(\text{intercept} + \text{slope} * 2.3) = (-1)$

$$+ 2 * (3.2 - (\text{intercept} + \text{slope} * 2.9)) \frac{d}{d\,\text{intercept}}(3.2 - (\text{intercept} + \text{slope} * 2.9))$$

$\frac{d}{d\,\text{intercept}}(\text{intercept} + \text{slope} * 2.9) = (-1)$

$$\frac{d}{d\,intercept}\left(3.2 - (intercept + slope * 2.9)\right)$$

$$(-1)$$

$$\frac{d(SSE)}{d\,intercept} = (-2)(1.4 - (intercept + slope * 0.5)) + (-2)(1.9 - (intercept + slope * 2.3)) + (-2)(3.2 - (intercept + slope * 2.9)).$$

Eq 1.

Sum of Squared Residual Error (SSE)

$$= (1.4 - (intercept + slope * 0.5))^2 + (1.9 - (intercept + slope * 2.3))^2 + (3.2 - (intercept + slope * 2.9))^2$$

Now

$$\frac{d(SSE)}{d\,slope} = \frac{d}{d\,slope}(1.4 - (intercept + slope * 0.5))^2 + \frac{d}{d\,slope}(1.9 - (intercept + slope * 2.3))^2 + \frac{d}{d\,slope}(3.2 - (intercept + slope * 2.9))^2.$$

$$\frac{d(SSE)}{d\,slope} = (-2)(0.5)(1.4 - (intercept + slope * 0.5)) + (-2)(2.3)(1.9 - (intercept + slope * 2.3)) + (-2)(2.9)(3.2 - (intercept + slope * 2.9)).$$

Eq 2.

We got the above 2 equations.

Let start with Random values of intercept & slope

Let intercept = 0, slope = 1

intercept = 0 & slope = 1 in Eq 1 & Eq 2

Substitute $\varphi_{intercept} = 0$ & slope = 1 in Eq 1 & Eq 2

$\frac{d(SSE)}{d\,slope} = -0.8$ ⇒ slope of line wrt parameter slope

$\frac{d(SSE)}{d\,intercept} = -1.6$ ⇒ slope of line wrt parameter $\varphi_{intercept}$.

For step size

New step size = slope of line at point * learning Rate

* Learning Rate ⇒ takes smaller value [10^{-6} to 1.0]

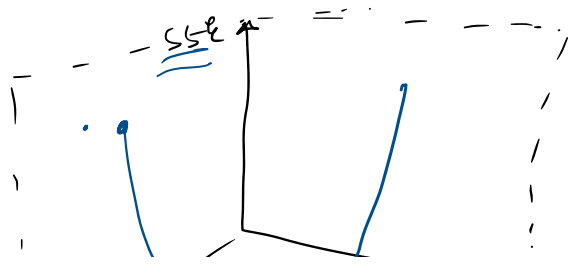
Let us take learning Rate = 0.01

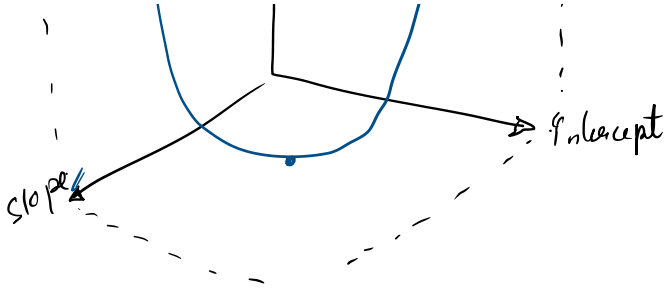
$\text{stepSize}_{(\varphi_{intercept})} = \frac{d(SSE)}{d\,intercept} * 0.01 = -1.6 * 0.01 = -0.016$

$\text{stepSize}_{(slope)} = \frac{d(SSE)}{d\,slope} * 0.01 = -0.8 * 0.01 = -0.008$

New Intercept = old $\varphi_{intercept}$ − $\text{stepSize}_{(intercept)}$ = 0 − (−0.016) = 0.016

New Slope = old slope − $\text{stepSize}_{(slope)}$ = 1 − (−0.008) = 1.008





[Substitute the new intercept and slope in $\frac{d(SSE)}{d\,intercept}$ & $\frac{d(SSE)}{d\,slope}$ and Calculate new step size for intercept & slope
And again Calculate New Intercept & New slope]

* We will repeat until the step size becomes very small
  ⇒ 0.001

  or Some maximum Number of steps is reached Ex: 1000

[For above Dataset Best fitting line will have
intercept = 0.95 & slope = 0.64]

[This is How Gradient Descent Optimizes Parameters]

Summary -> Gradient Descent

1. Get Eq^n of line [Identify Simple Linear or Polynomial]
2. Want intercept & slope value that Minimizes Cost f^n
3. Identify Cost (loss) function. [we took SSE]
4. Find Eq for $\frac{d(SSE)}{d\,intercept}$ & $\frac{d(SSE)}{d\,slope}$
5. Start with Some Initial value of slope & intercept.
6. Find the value of $\frac{d(SSE)}{d\,intercept}$ & $\frac{d(SSE)}{d\,slope}$

(6) Find the value of $\frac{d(SSE)}{d\,intercept} * \frac{d(SSE)}{d\,slope}$

(7) Calculate $\text{StepSize}_{(intercept)} = \frac{d(SSE)}{d\,intercept} * \text{learning rate}$

$\text{StepSize}_{(slope)} = \frac{d(SSE)}{d\,slope} * \text{Learning rate}$.

(8) Update intercept = old intercept - $\text{StepSize}_{(intercept)}$

Update slope = old slope - StepSize (slope)

(9) Repeat step 6 - 8 untill stepSize is very small

or Number of iterations $\leq$ 1000

Gradient Boost for Regression

* Gradient Boost start with Single Leaf instead of Tree/stump.

* Leaf represents Initial guess for the weights of the Samples.

Consider

Height	Favorite color	Gender	Weight (Actual)
1.6	Blue	male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	male	73
1.5	Green	male	77
1.4	Blue	Female	57

Step1 => First let make Initial Guess = Weight (Average) = 71.2

Step2 => Like Adaboost, Even Gradient Boost builds tree based on previous Tree

But unlike Adaboost, the Gradient Boost tree are larger than Stump

[However the Height is still Restricted [No of Leaf 8 → 32 in Each Tree]

* Gradient Boost will build Another Tree based on Error of previous Tree.

* [Gradient Boost continous to build Tree in this fashion untill it has made number of Trees you asked for or the Additional tree fails to improve the fit]

⇒ Step 3 : Initial Guess = weight (Average) = 71.2

Predicted Value = 71.2 + Tree1 + Tree2 . . .

71.2 ↑ Initial guess

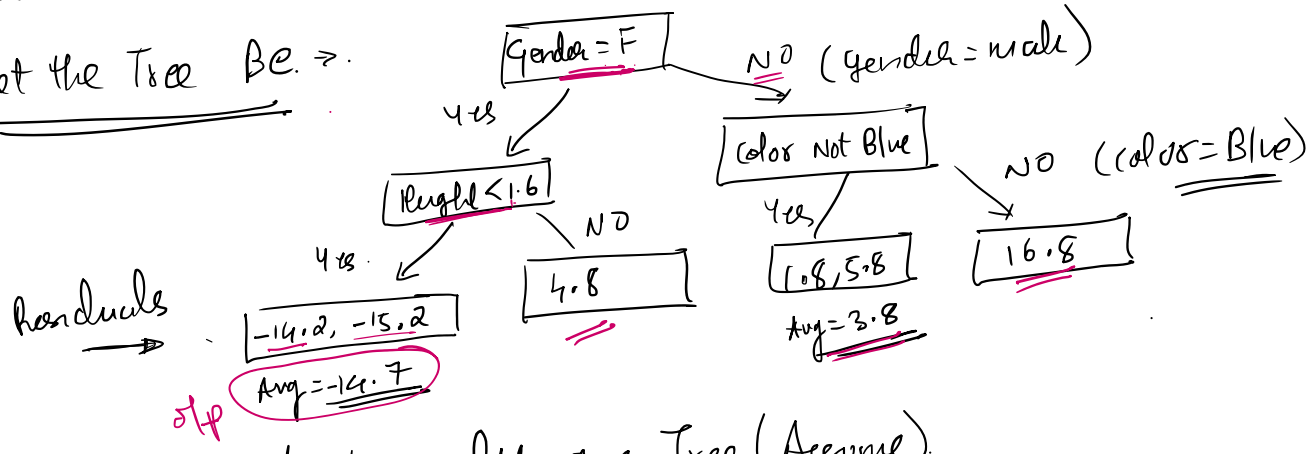
Tree1 + Tree2 . . . → Boosting

Height	Favorite Color	Gender	Weight	Residual (Observed - Average)
1.6	Blue	male	88	16.8
1.6	Green	Female	76	- 4.8
1.5	Blue	Female	56	-15.2
1.8	Red	male	73	1.8
1.5	Green	male	77	5.8
1.4	Blue	Female	57	-14.2

Initial 71.2

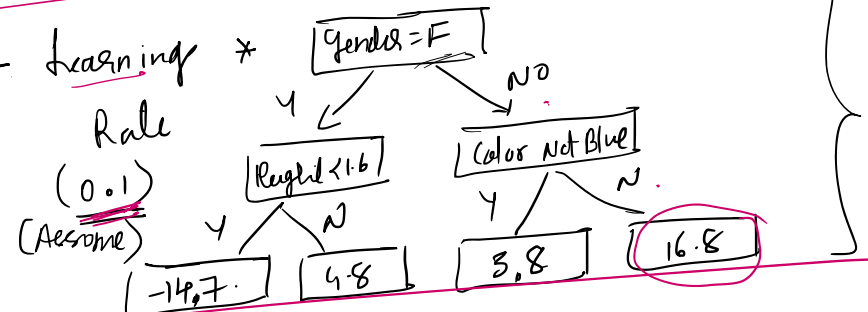
Let Build a Tree using Residual and not weight

Let the Tree Be. ⇒



If there will be only one Tree (Assume)

Then.

Predicted value = Initial Guess + Learning Rate (0.1) (Assume) * 

Now first Sample = | 1.6 | Blue | male | ? | weight

Initial - 71.2 + 0.1 * 16.8 = 72.88

Predicted value = $71.2 + 0.1 * 16.8 = 72.88$

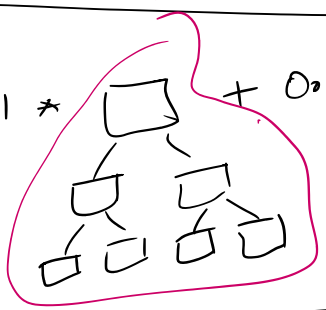
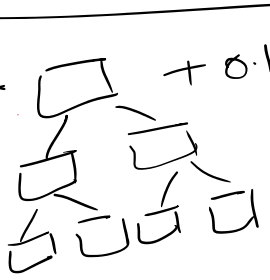
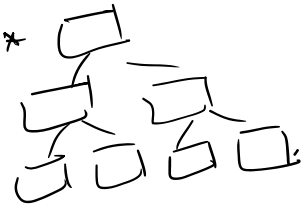
* Repeat this for Every Sample & Calculate Predicted value for Each Sample
  → Call it as New Prediction.

Height	Favorite Color	Gender	Weight (Actual)	Residual	New Prediction	New Residual
1.6	Blue	male	88	16.8 ✓	72.88	R1
1.6	Green	Female	76	-4.8	P1	R2
1.5	Blue	Female	56	-15.2	P2	R3
1.8	Red	male	73	1.8	P3	R4
1.5	Green	male	77	5.8	P4	R5
1.4	Blue	Female	57	-14.2	P5	R6

71.2

* Now find New Residual for New Prediction for all the Samples
* Using this New Residual Construct New Tree and so on Untill the New Tree does not make Significant Contribution to the New Prediction.

* Let us Say we have 3 Tree for above Dataset →

Final Prediction = Initial value + 0.1 *  + 0.1 *  + 0.1 * 

[Gradient Boost Principle → Taking lot of Small steps in Right Direction will result into better Predictions]

Gradient Boost for Classification →

Consider.

Likes Popcorn	Age	Favorate color	Loves Jawan movie. ↓ (Probability)
Yes	12	Blue	Yes (1)
Yes	87	Green	Yes (1)
NO	44	Blue	NO (0)
Yes	19	Red	NO (0)
NO	32	Green	Yes (1)
NO	14	Blue	Yes (1)

(Actual values)

Step1: Start with Initial Prediction →

[Note: When we use gradient boost for classification, the initial prediction for every individual sample :→ log(odd)]

$$odd = \frac{\text{No of Yes}}{\text{No of NO}} = \frac{4}{2} = 2$$

$$\log_e(odd) = \log_e(2) = 0.7$$

To use this we have to Convert it into Probability =

Probability of loving Jawan movie = $\frac{1}{1+e^{-\log(odd)}} = \frac{1}{1+e^{-0.7}} \approx 0.7$

So Initial Prediction for Probability of all the Samples for loving Jawan movie = 0.7

Initial Prediction 0.7

Likes Popcorn	Age	Favorate color	Loves Jawan movie. ↓	Residual Error
Yes	12	Blue	Yes (1)	0.3
Yes	87	Green	Yes (1)	0.3
		Blue	NO (0)	-0.7

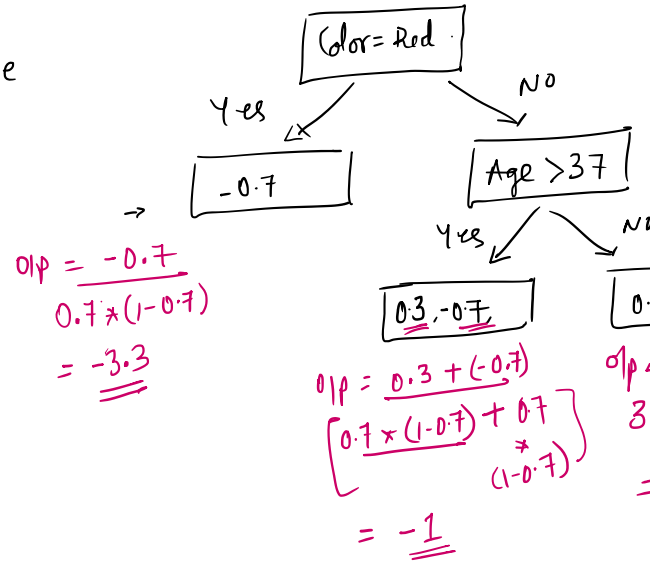
Yes	87	Green	Yes (1)	0.3
NO	44	Blue	NO (0)	-0.7
Yes	19	Red	NO (0)	-0.7
NO	32	Green	Yes (1)	0.3
NO	14	Blue	Yes (1)	0.3

Actual

(Initial Residual Error)

Now let us Build Tree Color And ~~Age~~

Let the Tree be



$o/p = \frac{-0.7}{0.7 * (1-0.7)} = -3.3$

$o/p = 0.3 + (-0.7)$ / $[0.7 \times (1-0.7) + 0.7 * (1-0.7)]$ $= -1$

$o/p = 0.3+0.3+0.3$ / $3 * (0.7) \times (1-0.7)$ $= 1.4$

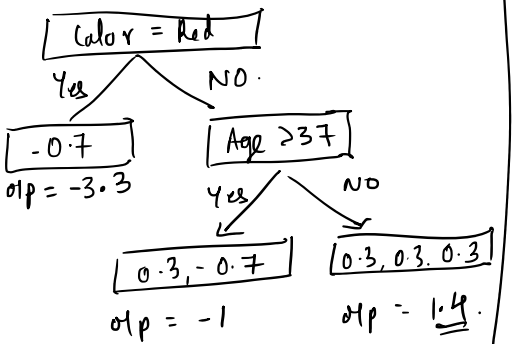
For Classification:

o/p value at every leaf

$= \frac{\sum \text{Residual}}{\sum (\text{Previous Prob} * (1 - \text{Previous Prob}))}$

Here this time Previous Prob = Initial Prediction

∴ New Prediction = Initial Prediction + 0.8 (learning Rate) *



Consider first Sample

Like Popcorn	Age	Color	Initial
Yes	12	Blue	0.7

New Prediction = $0.7 + 0.8 * 1.4 = 1.8$

Convert into Probability = $\frac{1}{1+e^{-1.8}} = 0.9$

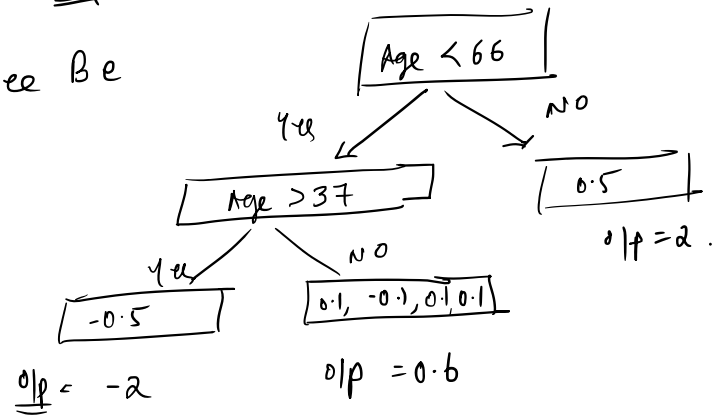
Similarly Calculate for all the other Samples -

Now the New Prediction be :→ Initial Prediction = 0.7

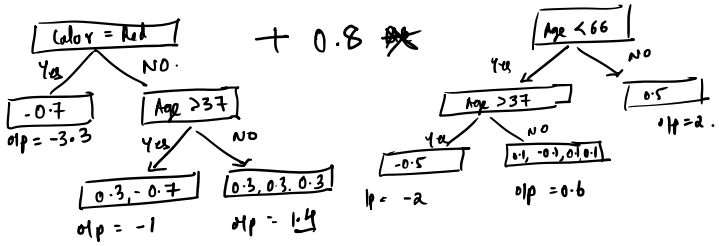
Likes Popcorn	Age	Favorate color	Loves Jawan movie (Actual values)	Residual Error	New Prediction	New Residual
Yes	12	Blue	Yes (1)	0.3	0.9	0.1
Yes	87	Green	Yes (1)	0.3	0.5	0.5
NO	44	Blue	NO (0)	-0.7	0.5	-0.5
Yes	19	Red	NO (0)	-0.7	0.1	-0.1
NO	32	Green	Yes (1)	0.3	0.9	0.1
NO	14	Blue	Yes (1)	0.3	0.9	0.1

* Here New prediction is not Same for all the Samples.
* Each Sample has its own prediction and calculated New residual for each Sample.
* Again on the basis of new Residual let us Build a Tree Using only Age only

Let the Tree Be



∴ New Prediction = Initial Prediction + 0.8 * (learning Rate)



* Apply each Sample in dataset on the above model and Calculate New Predictions

Apply ... simple ... number ...

New Predictions

* Again calculate New Residual and so on -
* This process repeats untill we have made the maximum specified number of Trees or the residual gets very very small.

Summary : Gradient Boost for classification.

(1) Consider the dataset
(2) Find log(odd) on the target
(3) Find Initial Probability $= \frac{1}{1+e^{-\log(odd)}}$.
(4) Now calculate Initial Residual based on Initial Probability
(5) Build a Tree using some of the features and Initial Residual (-Tree1)
(6) Calculate New Prediction for = Initial + learning * Tree1
Every sample Prediction Rate
(7) Using New Predicate Calculate New Residual.
(8) Construct a new Tree based on some feature and New Residual and so on.
(9) Repeat untill max no of Trees are constructed or the Residuals are insignificant.

Summary : Gradient Boost for Regression.

(1) Consider the dataset
(2) Find Initial Prediction =: Avg of observed value.
(3) Now calculate Initial Residual based on Initial Prediction
(4) Build a Tree using some of the features and Initial Residual (-Tree1)
(5) Calculate New Prediction for = Initial + learning * Tree1
Every sample Prediction Rate

(5) Calculate New Prediction for = Initial + learning rate * Every sample Prediction Rate

(6) Using New Predicate Calculate New Residual.

(7) Construct a new Tree based on some feature and New Residual and so on.

(8) Repeat until Max no of Trees are Constructed or the Residuals are insignificant.

13 September 2023 19:59

XG Boost (Xtreme Gradient Boost)

(1) XGBoost for Regression

Consider (x : Dosage, y : Effectiveness (Actual))

Dosage of Drug	Effectiveness of Drug
10	-10
20	7
25	8
35	-7

For Classification

Dosage	Effectiveness
10	0
20	1
25	1
35	0

Build Prediction Model Using XGBoost

Step1: Assume Initial threshold for Effectiveness = 0.5

Step 2 Calculate Residual.

Initial Prediction = 0.5

Dosage of Drug	Effectiveness of Drug	Residual
10	-10	-10.5
20	7	6.5
25	8	7.5
35	-7	-7.5

Let $\lambda = 0$

Step3: Consider all the Residuals in Root initially.

[-10.5, 6.5, 7.5, -7.5]

$$\text{Similarity} = \frac{(-10.5 + 6.5)^2}{4 + 0} = 4$$

Note

$$\text{Similarity Score} = \frac{(\text{Sum of Residual})^2}{\text{No of Residual} + \lambda}$$

[λ = Regularization Parameter ⇒ prevents Overfitting]

Gain = Similarity (left) + Similarity (Right) − Similarity (Root).

Now let us see if we can cluster the residual better on Similarity by split

Determine Dosage value for split ⇒.

Case1) Consider Average of first 2 dosage, i.e 10 & 20

Avg = 15

Similarly = 4

Case2) Consider dosage as Average of 20 & 25

Average = 22.5

Dosage < 22.5

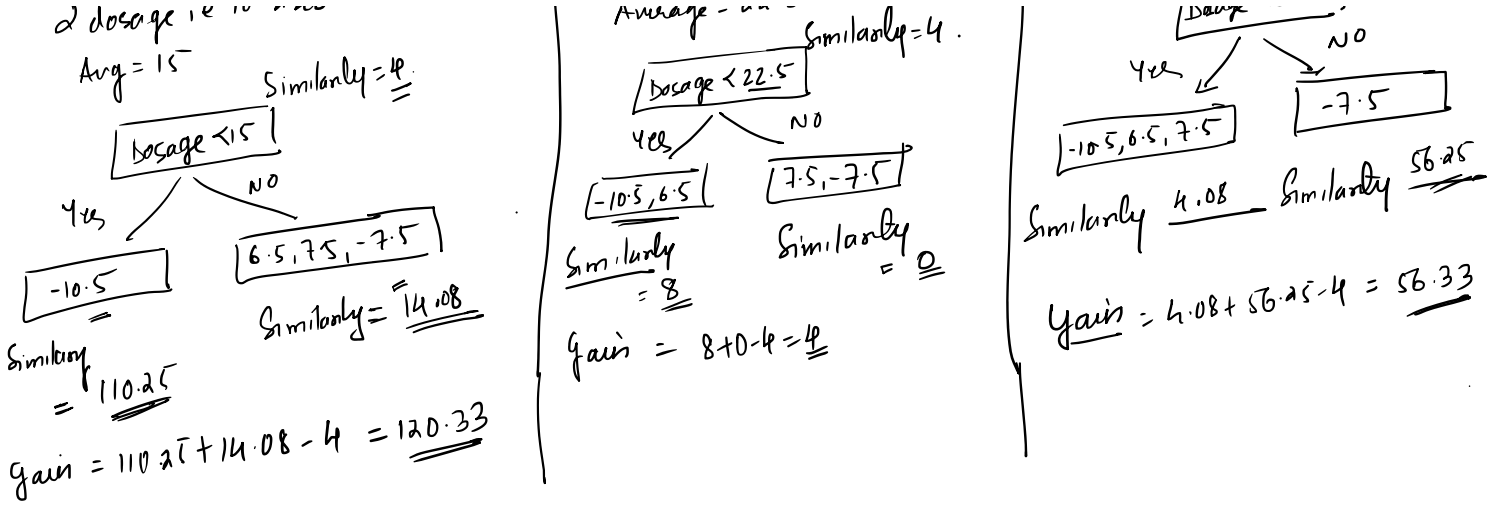
Similarly = 4.

Case3: Dosage as Average of 25 & 35 Avg = 30

Similarly = 4

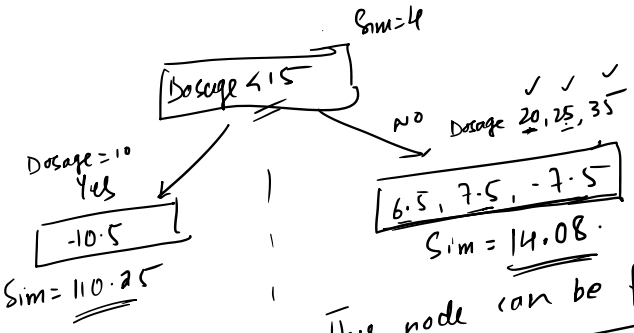
Dosage < 30

Yes / No

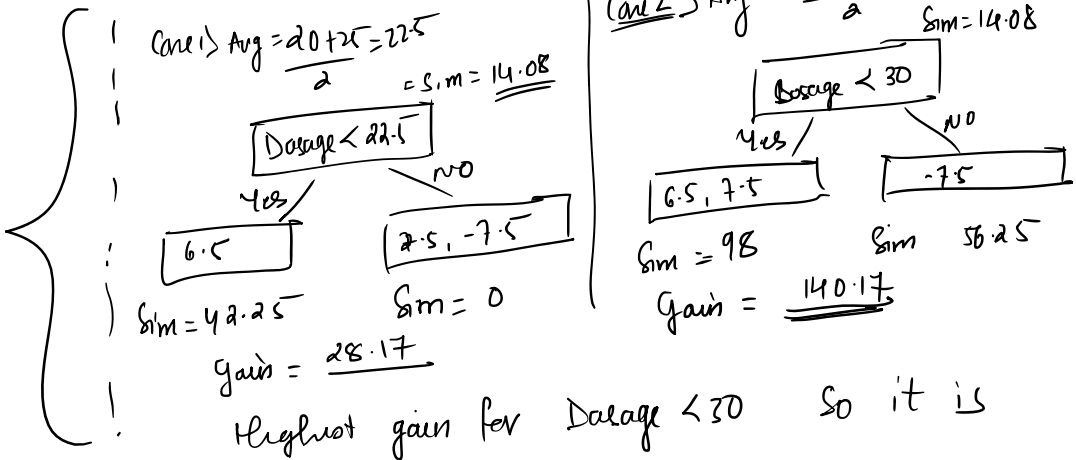


Since the Dosage < 15 gives highest gain = 120.33

The split factor = Dosage < 15

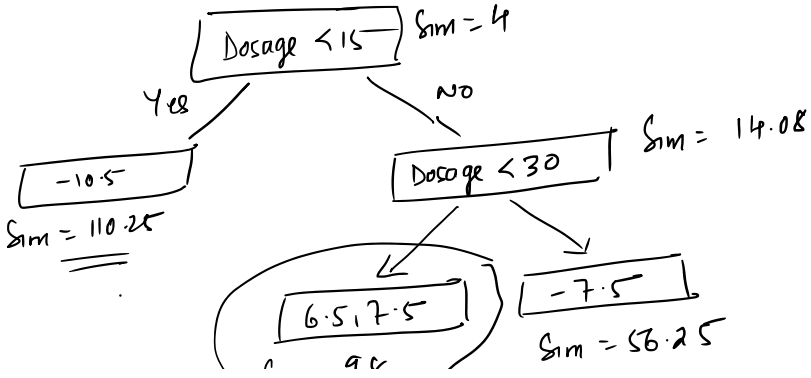


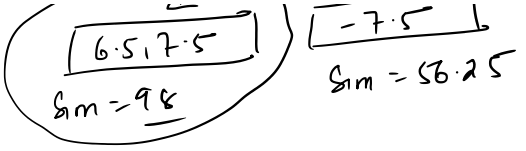
This node can be further splitted



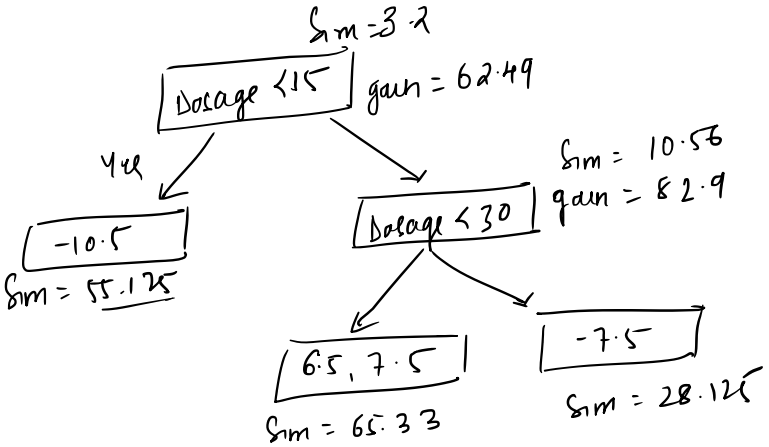
Highest gain for Dosage < 30 So it is the split condition

Now Tree $\lambda = 0$





Now for $\lambda = 1$



Note

When $\lambda > 0$ the Similarity Score decreases (↓)
Also Gain decreases (↓).

Pruning : Kyon chahiye

1. Height Restriction of Tree
2. Ignoring Unnecessary branches
3. Decrease computation
4. Less Time
5. Avoid Overfitting

Pruning of Trees in XGBoost is purely done on gain

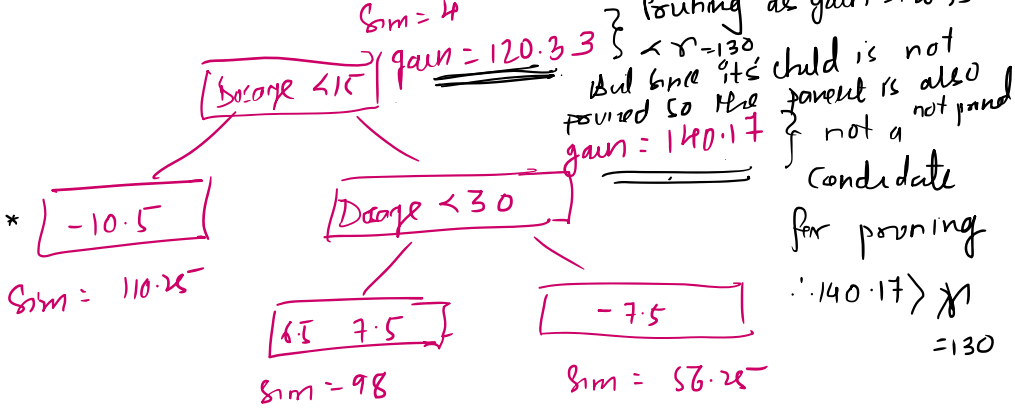
Let us Assume a Threshold gain = γ [gamma].

Rule for Pruning : * If at branch gain - γ < 0 then Prune and go above

* If gain - $\gamma \geq 0$ then do not Prune
* If child is not pruned then parent also will not be pruned

∴ Tree $\lambda = 0$

Let $\gamma = 130$



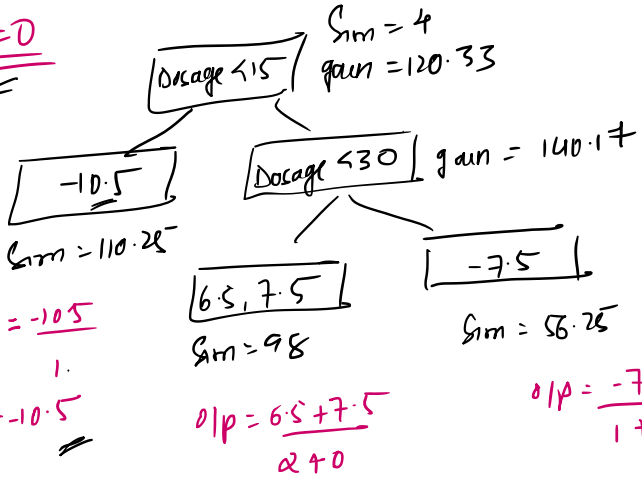
* Note
If γ is sufficiently large then it may result into Complete Tree pruning till root → Extreme Pruning

* Regardless of value of λ and γ let us assume the Tree as :->

* Regardless of value of λ and γ let us assume the tree as :->

* O/p of each leaf $\lambda = 0$

$$= \frac{\sum \text{Residual}}{\text{No of Residual} + \lambda}$$



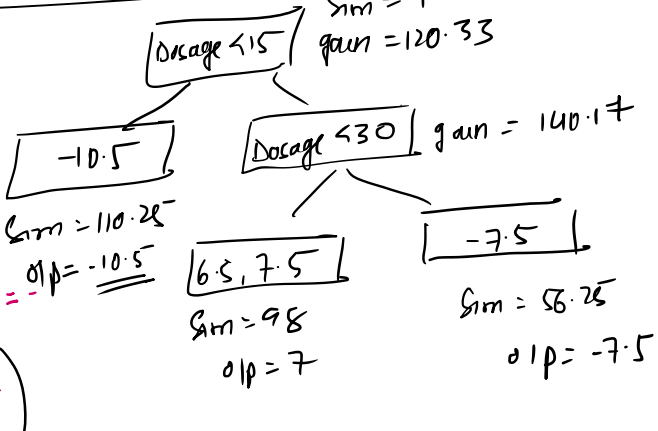
$o/p = \frac{-10.5}{1} = -10.5$

$o/p = \frac{6.5 + 7.5}{2 + 0} =$

$o/p = \frac{-7.5}{1 + 0} = -7.5$

New Prediction = $0.5 + 0.3 \times$

(In XG Boost the learning rate is known as eta (ϵ) and default value = 0.3)



Now Prediction for Dosage = 10

$= 0.5 + 0.3 * (-10.5) = -2.65$

for Dosage = 10 Initial Prediction = 0.5

New Prediction = -2.65

* Find New Prediction for all the other 3 Dosages.
* Using the New Residual Build a New Prediction that will give Smaller Residual
* Keep Building the Tree until the max No of Trees Constructed or Residual is Significantly Small.

Residual is significantly small.

Let Consider Four above Case there are 4 Tree (T_1, T_2, T_3, T_4)

* final Prediction = $0.5 + 0.3 * T_1 + 0.3 * T_2 + 0.3 * T_3 + 0.3 * T_4$.

XGBoost (Xtreme Gradient Boost)

* Implementation of Gradient Boost Decision Tree.
* Decision Tree is constructed in Sequential form.
* Weights plays Important Role in XGBoost
* Weights are assigned to all independent variables which are fed into Decision Tree that predict result.
* Weights of variables predicted wrong by previous Tree is increased and that variable are fed in Next Decision Tree.
* These Individual predictors/classifiers can ensemble to give Strong and precise model.
* XGBoost is faster than Gradient Boost
* There is stop criterion for Tree splitting in XGBoost
  - → XG Boost uses Max Depth parameter that it starts pruning the Tree backward.
  - → This pruning improves computational performance and helps to overcome problem of "Overfitting"

XG Boost for classification →

① Convert the observed value into Probabilities (1 & 0).

② Decide on initial prediction (Ex 0.5)

③ find Residual (observed − Initial Prediction).

④ Create Leaf with All Residuals.

⑤ Calculate Similarity = $\frac{\Sigma (\text{Residual})^2}{\Sigma \text{Prev Prob} \times (1 - \text{Previous Prob}) + \lambda}$ [$\lambda = 0$ or $\lambda = 1$]

(Initial prob)

⑥ Now to decide where to Split

[Take number of Averages cases and Calculate gain for each. Select the split with gain]

⑦ [XGB have threshold for Minimum No of Residuals in the leaf

* To find Minimum No of Residuals in each leaf

Cover is calculated.

Cover = Σ (Previous Prob) * (1 − Prev Prob)

[In Regression ⇒ Cover ⇒ 1 (default)]]

⑧ Construct a Tree.

⑨ Calculate o/p value at Each Leaf = $\frac{\Sigma (\text{Residual})}{\text{No of Residual} + \lambda}$.

⑩ Calculate New Predicted Value = Initial + Learning * Tree 1

(10) Calculate New Predicted Value = Initial Prob + Learning Rate * Tree 1

(11) Repeat the process until max No of Trees are reached or Residual becomes Insignificant.

(12) [Note > New Prediction will be log(odd).
To Convert into Probability = $\frac{1}{1+e^{-\log(odd)}}$.
=> Thus gives New Probability]

XGB for Classification → Example

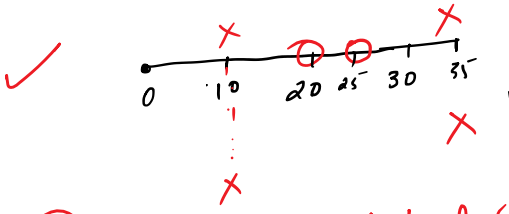
Q) XGBoost is used with Large Dataset

* Let us take a Sample for our Simplicity

X = Ineffective drug

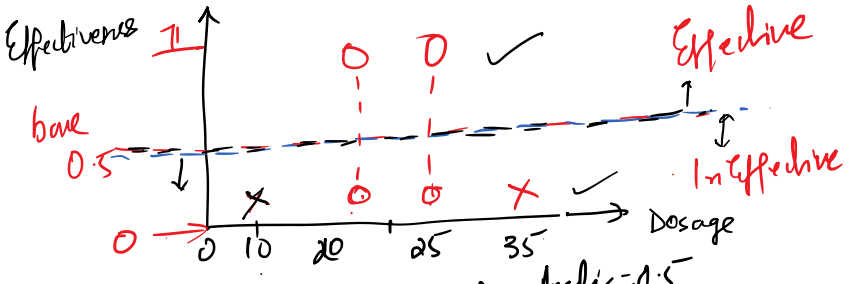
O = Effective drug

Dosage	Effective
10	-10
20	7
25	8
35	-7



① Step 1 → Make Initial (Base Prediction) = 0.5 [default value]

* Regardless of dosage there is 50% chance that drug is effective.



Dosage	Effective
10	0
20	1
25	1
35	0

Base Predic = 0.5

Dosage	Observ	Residual
10	0	-0.5
20	1	0.5
25	1	0.5
35	0	-0.5

$$\frac{(\Sigma \text{Residual})^2}{\Sigma \text{Previous Prob} * (1 - \text{Previous Prob}) + \lambda}$$

(formula)

Start the Tree with single leaf

[-0.5, 0.5, 0.5, -0.5] Similarity = 0

Here Prev Probability for all Samples = 0.5 (Initial base probability)

Consider Dosage < 15 | Sim = 0 | λ = 0

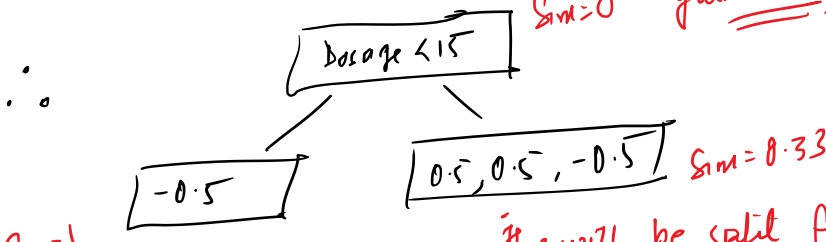
$$Sim = \frac{(-0.5)^2}{(0.5 \times 0.5)} = 1$$

$$Sim = \frac{(0.5 + 0.5 - 0.5)^2}{(0.5 \times 0.5) + (0.5 \times 0.5) + (0.5 \times 0.5)} = 0.33$$

gain = 1.33

$\frac{(-0.5)}{(0.5 \times 0.5)} = \frac{1}{4}$ (0.5×0.5) ...

Let us Assume that Dosage < 15 gives Best gain, then Dosage < 15 will be split factor.



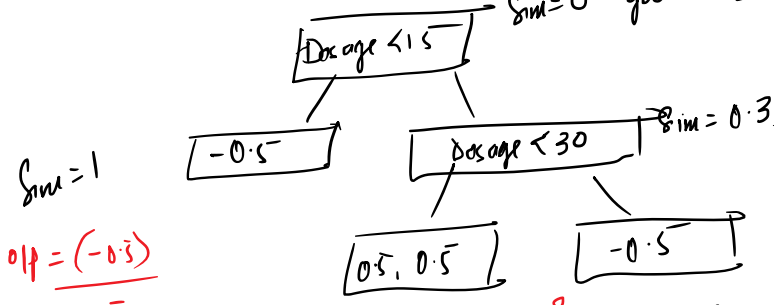
Sim = 0 gain = 1.33

Sim = 1

Sim = 0.33

∴ Thus will be split further.

Let Dosage < 30 is split factor that give highest gain



Sim = 0 gain 1.33

Sim = 0.33, gain = 2 + 1 − 0.33 = 2.67

Sim = 1

o/p = $\frac{(-0.5)}{0.5 \times 0.5} = -2$

Sim = $\frac{(1)^2}{0.5 \times 0.5 + 0.5 \times 0.5} = 2$

o/p = 2

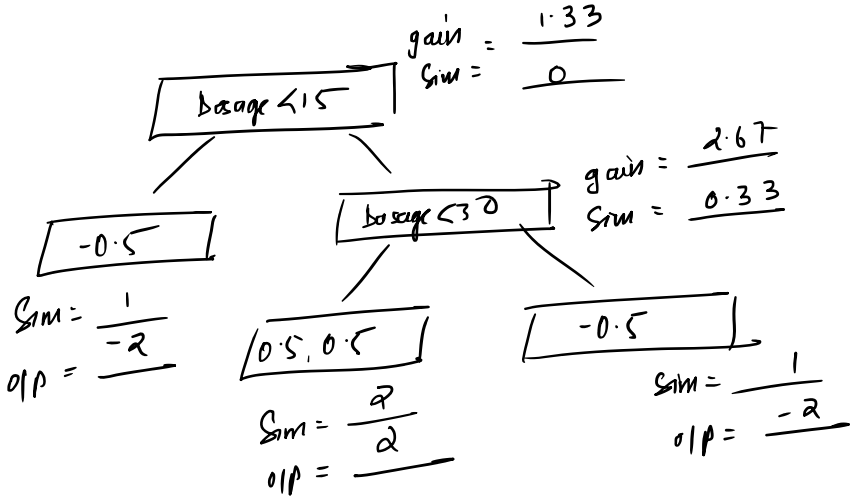
Sim = $\frac{(-0.5)^2}{0.5 \times 0.5} = 1$

o/p = $\frac{-0.5}{0.5 \times 0.5} = \frac{-1}{0.5} = -2$

o/p for each leaf = $\frac{\sum \text{Sum of Residual}}{\sum \text{Previous Prob} \times (1 - \text{Previous Prob}) + \lambda}$ = ____

for $\lambda = 0$.

New Prediction = $0.5 + 0.3 \times$ (↑ Learning Rate)



Sim = $\frac{1}{-2}$

o/p = ____

Sim = $\frac{2}{2}$

o/p = ____

Sim = $\frac{1}{-2}$

o/p = ____

for Dosage = 10 Prediction → $0.5 + 0.3 * (-2) = 0.5 - 0.6 = -0.1$

This is log(odd)

New Probab = $\frac{1}{1+e^{-0.1}} = 0.524$

For Dosage = 10 ✓

Original Predic
0.5

Residual = -0.5

New Prediction
0.524

Residual = -0.524

Smaller Residual than Before.

* Similarly do it for all the Samples.
* Find New ~~Pred~~ Probabilities & Residuals.
* Again Construct New Tree
* Continue until Max No of Trees reached or Residual generated becomes insignificant.

Vimp Bagging & Boosting

(1) Bagging (Bootstrapping Aggregate)

↳ Weak learners organized in Parallel.

↳ Independent weak learners

↳ Used to reduce Variance

Steps: (1) Multiple subsets are created from Original Dataset Selecting observations with replacements.

(2) A base model (weak) is created for each of subset

(3) The weak model runs in parallel and are Independent to each other.

(4) The final predictions are determined by combining the predictions from all the models.

Ex: Random Forest.

Boosting > Basic >

If a datapoint is incorrectly predicted by first model then the next model will correct the prediction thus giving better result for immediate next model.

Steps (1) A subset is created from Original Dataset

(2) Initially all the data points are given equal weights.

(3) A base model (Initial prediction) is created on dataset

(4) This base model is used to make prediction (Initial) on whole dataset

(5) [illegible] Calculated for each Sample.

whole dataset

(5) Residual (Initial) are calculated for each sample.

(6) Now the observation that are incorrectly classified are given higher weight

(7) Another model is created based on predictions of previous model.
[This model tries correct the errors from the previous model]

(8) Similarly we will have multiple models (each based on previous model)

(9) The final model (Strong learner) is weighted mean of all the model

Ex ADABOOST < Regression / Classification

XGBOOST < Regression / Classification

Q) Which one is Better?

Depends on
(1) Data
(2) Simulation
(3) Circumstances

* If Individual Single model has High Bias.
then Bagging will not improve Bias
However Boosting will improve the Bias.

* If Single model Overfits then Bagging is the Best option as Boosting will not help in Overfitting.

[Note
↑ Bias → Boosting is helpful
↑ Variance → Bagging is helpful.]

a) Similarities of Bagging & Boosting

(1) Both are Ensemble methods (we need weak learners)

(2) Both generates Several Training data sets by random Sampling.

(3) Both make final Desisions by averaging the N learners (Regression) or majority Voting (Classification).

(4) Both are good at Reducing Variance and providing Higher Stability.

b) Differences of Bagging & Boosting

Bagging	Boosting
(1) Bagging is Simplest way of Combining predictions that belongs to Same type	(1) Boosting is way of Combining predictions that belongs to diff types.
(2) Bagging primarily helps in reducing Variance	(2) Boosting primarily helps in reduuing of Bias
(3) Each model (Tree) recieves equal weights	(3) Boosting models are weighted according to their performance (Amount of say).
(4) Each model is built Independently.	(4) New model is built influenced by performance of previously built model.
(5) Different Training Dataset are randomly built with replacement from	(5) Every new subset contains Element that were misclassified

are randomly built with replacement from Entire Training Dataset

(6) It is better for overfitting.

(7) Ex: Random Forest

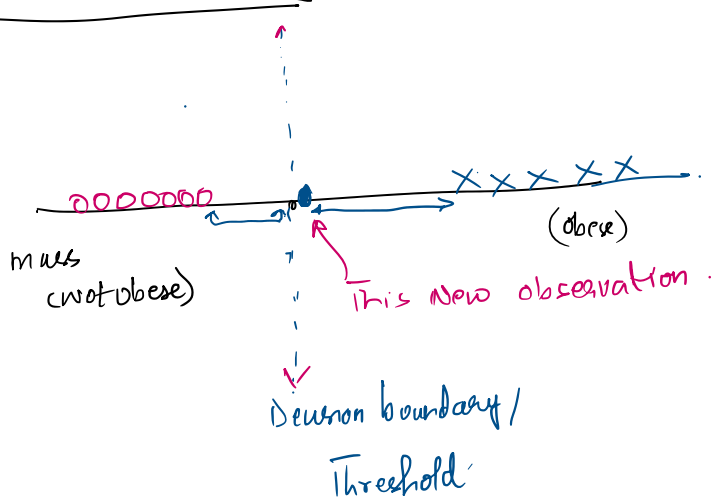
Element that were misclassified by previous model.

(6) It is better for Underfitting

(7) Ex. Gradient Boost
ADA Boost
XG Boost

SVM [Support Vector Machine]

Consider >



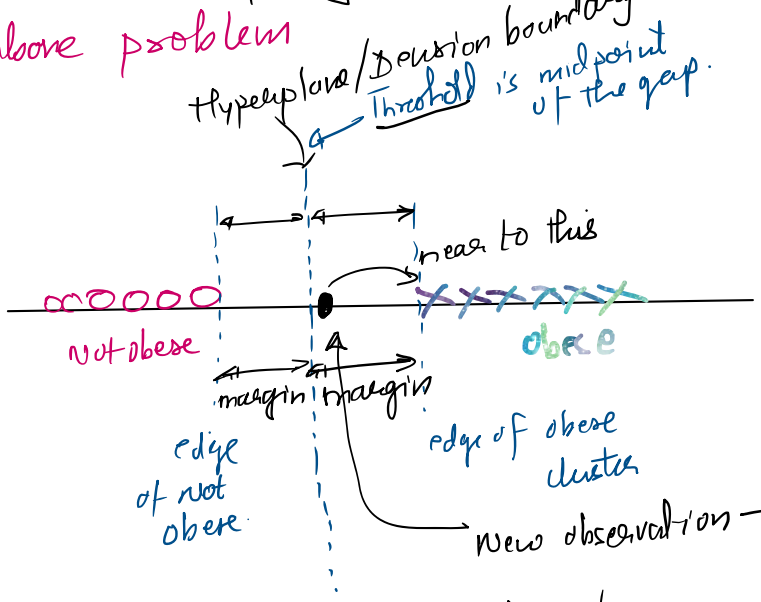
* Here the new observation will be classified as obese. Eventhough the new observation is more near to Not obese as compared to obese.

* The threshold here is pretty lame.

To Solve the above problem

Consider >

We will consider threshold as Midpoint of the gap bet^n the edges of Clusters



Here the margin > It is gap bet^n threshold (Decision Boundary) and the edge of the cluster

→ Here the margin is maximum.

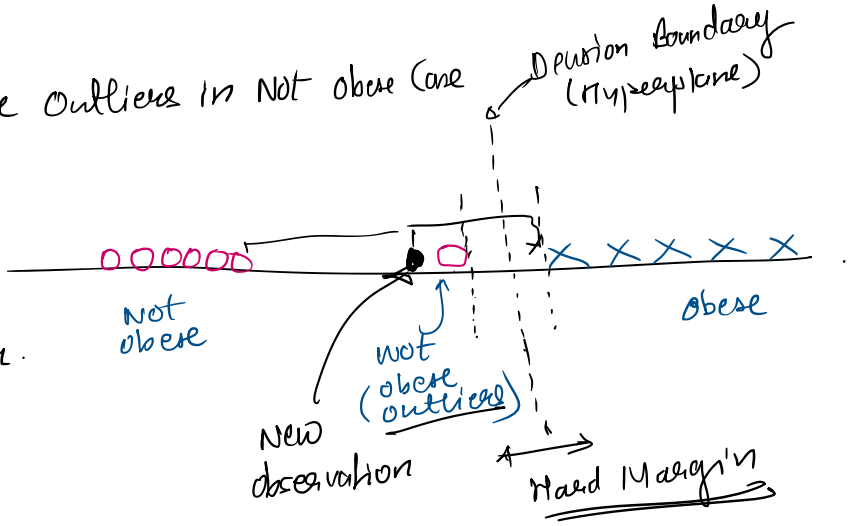
→ Known as Maximum Margin classifier →

→ Here in above Case there is No Misclassification

→ Also there are no Outliers

Consider → Here we have Outliers in Not obese Cone

* If we consider Margin as the edge of the cluster considering the Outlier.



* Here the new observation will be classified as Not obese, even though it is near to obese.

(Hard Margin)

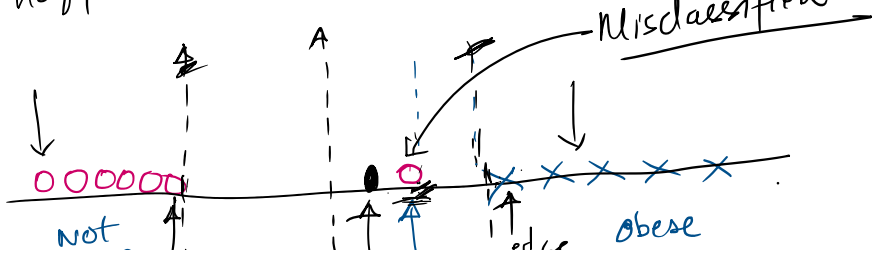
* Here the Maximum Margin Classifier is very sensitive to outlier.
* Here Even though we have outlier, however all the given data points are correctly classified → low Bias.
* But for the new observation it incorrectly classifies ⇒ High Variance

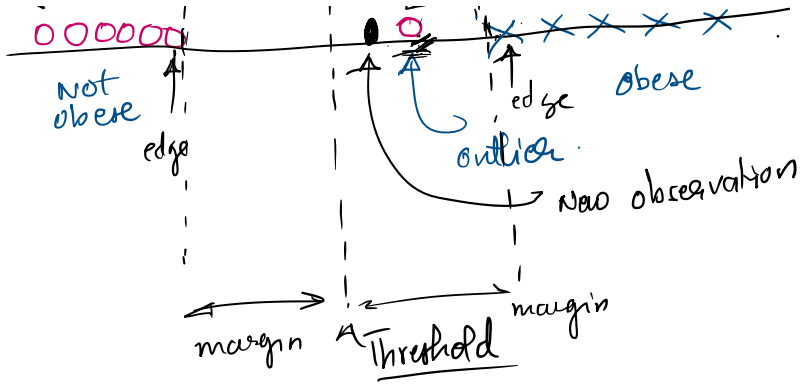
Note → With Hard Margin →

→ Misclassification Not Allowed
→ All Data points Correctly Classified ⇒ low Bias
→ New Observation Incorrectly Classified ⇒ High Variance

Now: We can do better (To make the threshold Insensitive to outlier)

→ For this to happen we must allow Misclassification.





Soft Margin.

* Here Since while considering edge of cluster, we have allowed Misclassification, so the margin now known as Soft Margin

Note * With Soft Margin →

→ Misclassification is Allowed

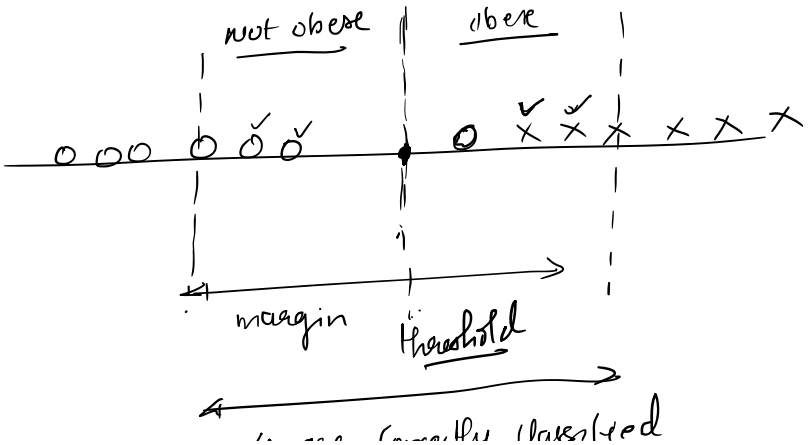
→ All given data points are not correctly classified (outliers). → High Bias.

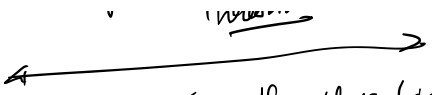
→ New observations are correctly classified ⇒ Low Variance

Soft Margin → When Misclassification is allowed then the distance bet^n threshold and given observation is known as soft margin.

* For above case we can have many soft margins.

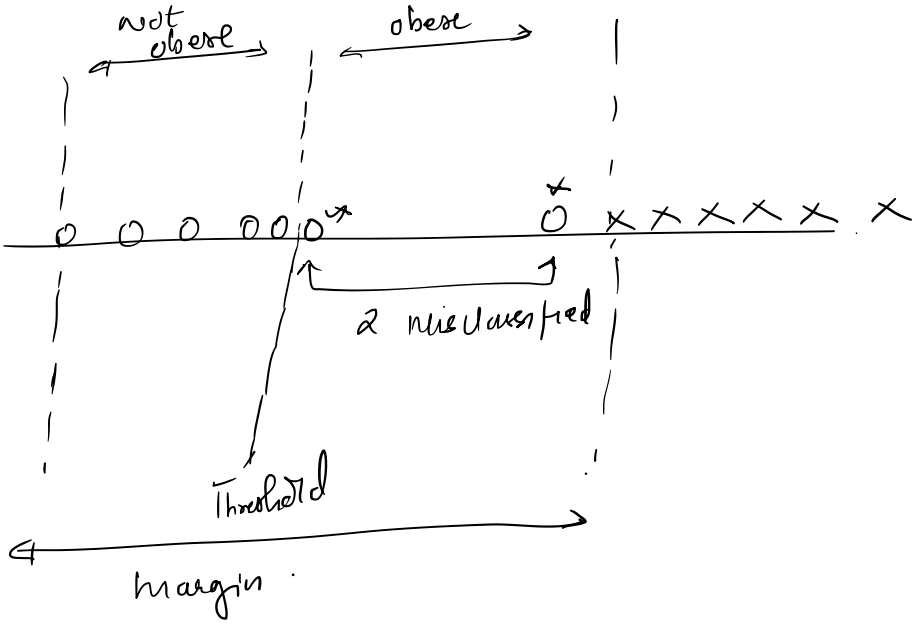
Ex Consider Case 1.





4 are Correctly classified
1 is Incorrectly classified.

Case 2)



* We will determine which soft margin is better by observing how many Misclassification each soft margin Allows.

VImP.

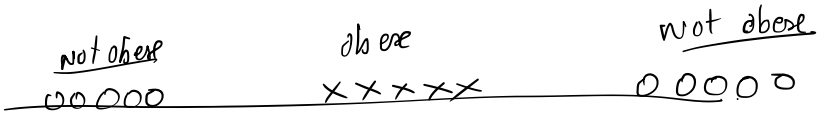
When we use Soft margin to determine the location of threshold then we are using "Soft margin classifier"
Also known as "Support Vector classifier"

* Support Vector Classifier → Main Aim is to create a Decision Boundary with allowed Misclassification
-> It allows small amount of Overlapping

But if the Dataset is as follows →

* Here there is very high level of overlapping

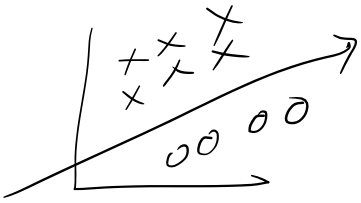
* Here there is very high level of overlap



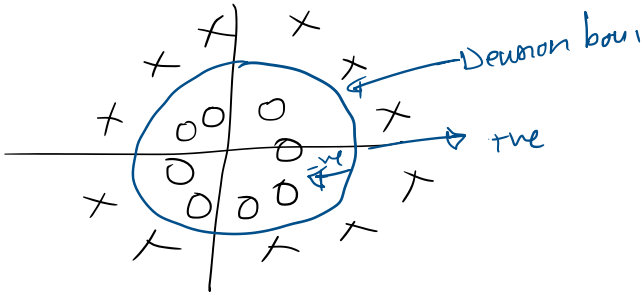
mass.

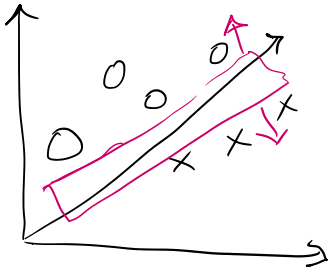
* Here we cannot have linear separator.

Also ⇒
- For 1-D data Decision Boundary is a point
- For 2-Data " " is a line
- For 3-D Data " " is a plane.



Also





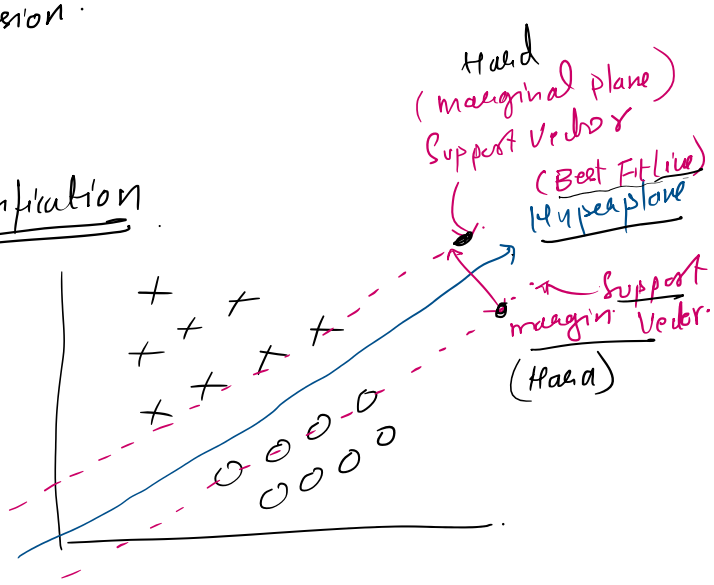
Here also Linear Decision Boundary Not Possible.

Note

SVM → ML Algorithm
→ Supervised
→ Task → Classification
→ Regression.

* Support Vector Classifier for Classification.

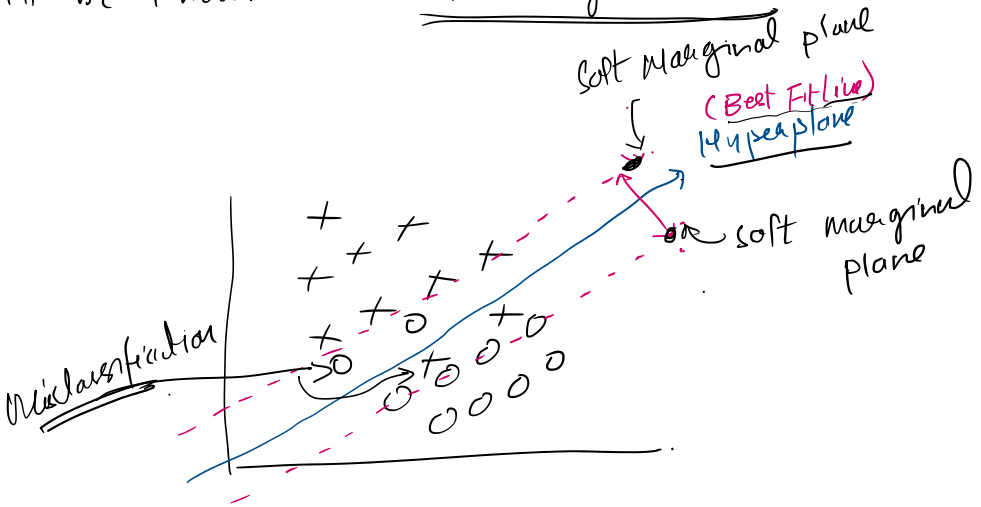
* Consider Binary Classification →



AIM →
Create a Hyperplane and Marginal plane such that the Margin is Maximum →.

* If we get perfect Marginal plane & Hyperplane [No Misclassification] then we can linearly separate the Data points.
This Margin will be known as **Hard Marginal Plane**

* If there are Misclassification allowed then the planes will be known as "**Soft Marginal Plane**"



Note Consider Hyperplane → Here 2-D Data
Hyperplane will be line.

Eg: $y = mx + c$

or $y = \theta_1 x_1 + \theta_0$

or $y = w_1 x_1 + b$

For multilinear Regression → Many features $x_1\ x_2\ x_3 \ldots x_n$
Each feature will have weight $w_1\ w_2\ w_3 \ldots w_n$

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots w_n x_n + b.$$

$$y = w^T X + b$$

y ↑ actual value

b ↑ predicted

$$w^T = [w_1\ w_2 \ldots w_n]$$

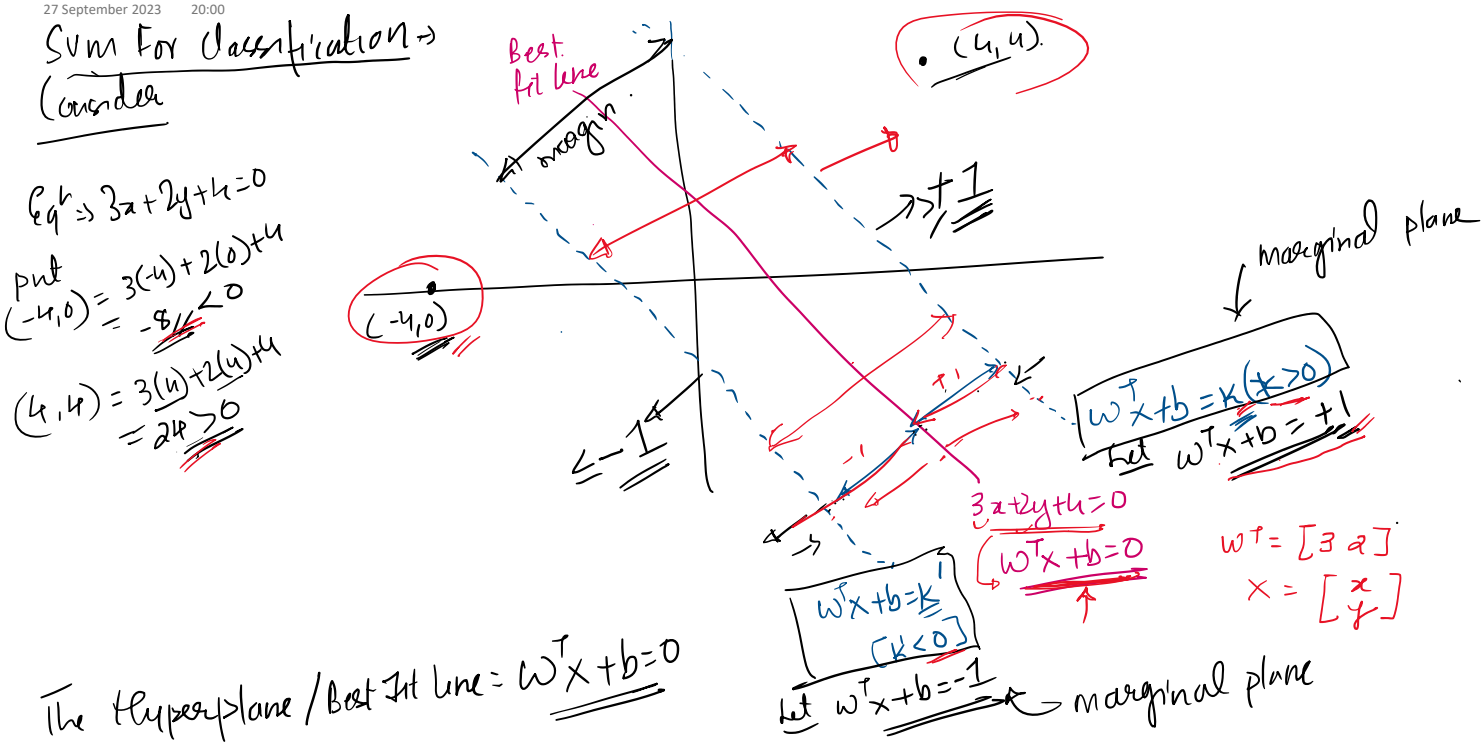
$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Consider →

27 September 2023 20:00

SVM For Classification →

Consider



The Hyperplane / Best Fit line : $w^T x + b = 0$

Let the Marginal plane be $w^T x + b = +1$ [on +ve side]

And $w^T x + b = -1$ [on -ve side]

Our Aim is to draw two Marginal planes (+ve & -ve side) and need to Ensure the distance (Margin) is Maximum.

* we want to find distance b/w the Marginal Planes.

Lets find difference :

$$\begin{aligned} w^T x_1 + b &= +1 \\ w^T x_2 + b &= -1 \\ \hline w^T (x_1 - x_2) &= 2 \end{aligned}$$

$\therefore$ $w^T (x_1 - x_2) = 2$.

Here w = slope (coefficient) < magnitude, direction } Vector.

To convert w^T into Vector ie $\vec{w}^T$

To Convert w^T into Vector ie $\vec{w}^T$

divide by $\|w\|$

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

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

$\frac{w^T}{\|w\|}(x_1 - x_2)$ ← difference (margin)

$\frac{2}{\|w\|}$ ← value

∴ To Maximize the Margin →

$$\text{Maximize}_{(w,b)} = \frac{2}{\|w\|}$$

Constraints →

y_i ← Actual/observed

$+1$ when $w^T x + b \geq 1$ (point lies outside of $w^T x + b = +1$ above) ← predicted

-1 when $w^T x + b \leq -1$ (point lies below of $w^T x + b = -1$)

The Constraints are for Correctly Classified Data point

∴ Final Constraints for S.V Classifier.

$$y_i * (w^T x + b) \geq 1$$

For Correctly Classified Data point

observed (actual) ← y_i ; Predicted value ← $w^T x + b$

$y_i = +1$

$w^T x + b \geq +1$

∴ To Maximize $= \frac{2}{\|w\|}$ Subjected to $y_i * (w^T x + b) \geq 1$

∴ To Maximize $= \frac{\alpha}{\|w\|}$ subjected to

Also can be written as

To Minimize $= \frac{\|w\|}{2}$ · Subjected to $y_i^o * (w^T x + b) \geq 1$

only for Correctly Classified Data points.

Here we have not considered for Misclassification

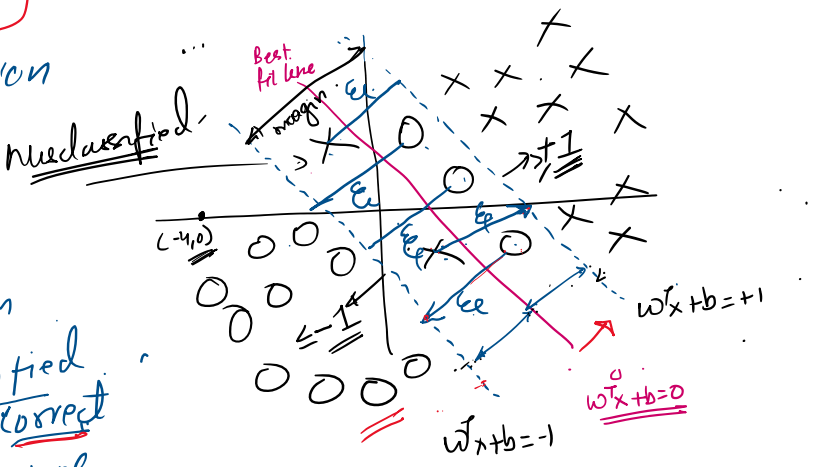
But in real world there will be always Misclassification.

∴ To Consider for Misclassification we have to use

"Hyper parameters"

We allow Misclassification

Now we will use two Hyperparameter



ξ_i ⇒ Eta ⇒ distance bet^n the Misclassified point and correct Marginal plane

C_i ⇒ No of Allowed Misclassified points

Now Final Cast F^n (For All data points) { Correctly & Incorrectly Classified }

* Objective ⇒

To Minimize (w, b) $\frac{\|w\|}{\alpha} + C_i \sum_{i=1}^{n} \xi_i$ Constraint $y_i^o * (w^T x + b) \geq 1$

C_i for Hyperparameters.

Correctly identified

for Misclassification.

∴ Label Cost F^n for SVC used for Classification

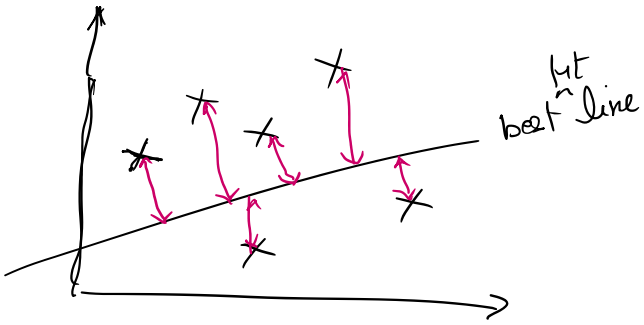
Misclassification.

The total cost F^n for SVG used for classification identified

[For Allowed Misclassification].

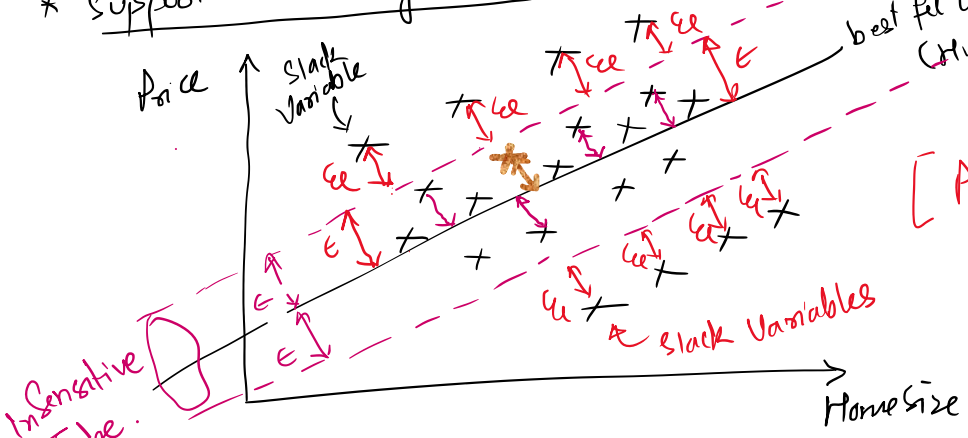
Support Vector for Regression :->

* In Linear Regression
  Objective -> Minimize MSE
  [Mean Squared Error]



Best Fit line -> g_t is the line that gives Minimum MSE

* Support Vector Regression :->



[Data Points inside ϵ Insensitive Tube -> Support Vector]

-> Here we are creating ϵ (Insensitive Tube) [Allowed Margin of Error]

* We can ignore the Error if difference between observed and predicted value is $\leq \epsilon$

* All the points inside ϵ Insensitive tube are not Considered for Calculating Error.

* The data points outside the ϵ Insensitive tube are known as Slack Variables.

* For a point x_i within Insensitive (ϵ) tube

* For a point x_i within Insensitive (ϵ) tube

$|y_i - w^T x_i| \leq \epsilon$

y_i = observed value

x_i = predicted value.

Constraint ⇒ if $|y_i - w^T x_i| \leq \epsilon$ (y_i ↑, $w^T x_i$ ↑) we can say that prediction is good and we will not consider the difference in Error Calculation.

* For Slack Variable [points outside ϵ Insensitive tube]

we need to calculate distance of Slack Variable from Hyperplane

= distance bet^n Slack Variable + ϵ & marginal plane

= $\xi_e + \epsilon$.

Hinge Loss.

Objective ⇒

Cost F^n ⇒ Minimize $\frac{\|w\|}{2} + C_i * \sum_{i=1}^{n} \xi_i$ over (w, b)

Hyperparameter

Constraints $|y_i - w^T x_i| \leq \epsilon + \xi_i$ → for points outside ϵ Insensitive tube

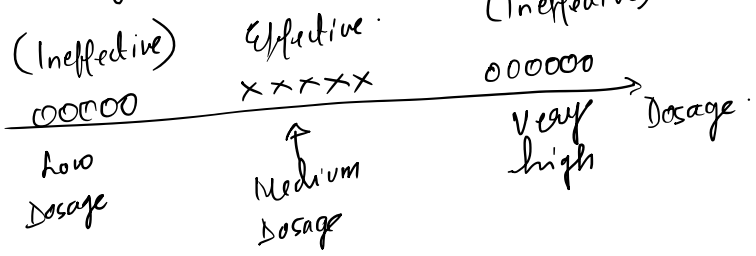
We have considered upto till Now Support Vector Classifier. (Soft Margin)

[We were interested in finding Max Margin classifier]

* If there are high amount of Overlapping than soft Margin classifier will not be efficient

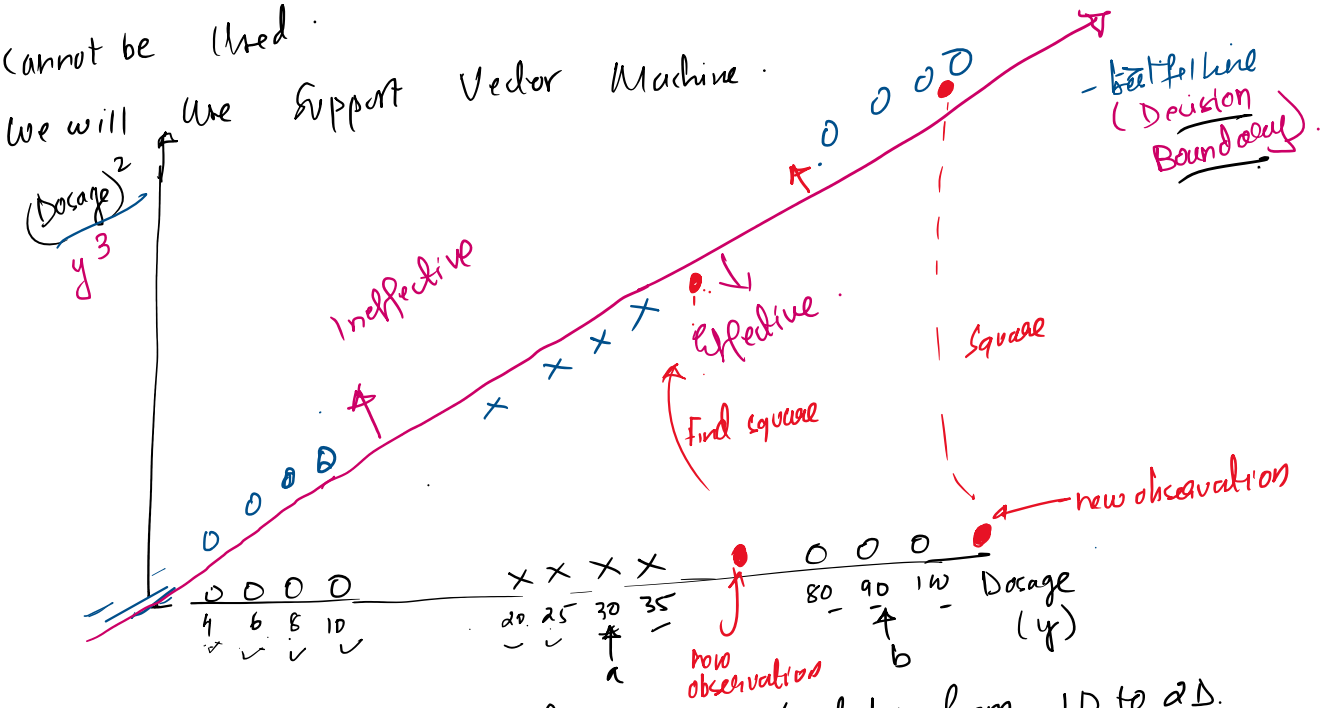
* In such case we will use Support Vector Machine.

Ex:- Consider the Dosage of Drug



→ Here we have lot of Overlapping so Maximum Margin classifier cannot be used.

↳ we will use Support Vector Machine.



* from above By increasing dimension of data from 1D to 2D. and by using Square (as function) we are able to separate the data

* from above [illegible] and by using Square (as function) we are able to separate the two points.

* We can also use it to predict for new observation
  → For new observation take square of the observation and chk if the square lies above (ineffective) or below (effective) the decision boundary.

SVM Main Idea →

* Start data with low dimension (Given).
* Now move data to Higher Dimension [Using some function]
* Find Support Vector Classifier to classify the data into different classes.

Question Arises -> Why Square, why not Cube or Some Other function?

→ How to decide, how to transform.

* The Function Used be known as Kernel

* Types of Kernel function in SVM.
  (1) Polynomial Kernel
  (2) Radial Basis Kernel.

Note 1)

If data points in 1D → Convert in 2D ⇒ line as Decision boundary
2D → Convert in 3D ⇒ Plane as Decision Boundary

2)

If Data points in 1D → Point is Decision Boundary
2D → line is Decision Boundary
3D → Plane is Decision Boundary.

Polynomial Kernel

* It calculates Higher Dimensional Relationship between observations (data points).

* The Kernel that transforms 1D to 2D is Polynomial Kernel.

* It may look like $(a * b + r)^d$

where a and b are two different observations in dataset (any two data points).

r = coefficient of Polynomial.

d = degree of Polynomial.

* Here we use SVM with Polynomial Kernel to Compute Relationship bet^n observations in Higher dimension and then find Good Classifier.

* Let a and b be two observations.
* Let $r = 1/2$ & $d = 2$ } Note ⇒ value of r and d is determined by Cross Validation.

$$\Rightarrow (a * b + r)^d = \left(a * b + 1/2\right)^2$$

$$= \left(a * b + 1/2\right) \cdot \left(a * b + 1/2\right)$$

$$= a^2 b^2 + \frac{1}{2} ab + \frac{1}{2} ab + 1/4$$

$$= ab + a^2 b^2 + \frac{1}{4} \Rightarrow \text{Polynomial.}$$

= can be written as dot product of

= can be written as dot product of

$= (a, a^2, \frac{1}{2}) \cdot (b, b^2, \frac{1}{2})$

For data point a — original value a — Higher dimension value of a — original value b — value of b in higher dimension

Dot product is sum of 1st term multiplied, and term multiplied and so on

Note: $\frac{1}{2}$ is third axis but value same so ignore.

So $(a + b \times \gamma)^d$ is used to get Higher dimension Rel^n between two data points a & b.

Let $a = 9$ $b = 14$

$\gamma = \frac{1}{2}$ $b = a$

$\Rightarrow$ Higher Dimension Relationship bet^n $a = 9$, $b = 14$

$\Rightarrow (9 + 14 + \frac{1}{2})^2 \Rightarrow 126.5^2 \Rightarrow 16002.25$

↑ Relationship Representation of $a = 9$, $b = 14$ in Higher Dimension

* We can find this Higher Dimension Relationship bet^n every pair of Data points.

VVImp → Kernel Trick

(1) Kernel function actually never does any transformation in Higher Dimension.

(2) Instead it calculates Relation bet^n observations and visualizes them in Higher Dimension.

visualizes them in Higher Dimension.

(3) Support Vector Classifier uses this visualization for classification.

(4) This is known as Kernel Trick.

$d=1$ ⇒ The Polynomial Kernel computes the relationship bet^n each pair of observation in 1D.

$d=2$ ⇒ " " " " " "

" " " " " " 2D.

Best value of d can be found by Cross Validation.

Note Since Polynomial Kernel is $(a*b+1/2)^2 = (a, a^2, 1/2)\cdot(b, b^2, 1/2)$

→ we actually do not have to Transform to understand higher dimension relationship.

→ All we need to do is calculate dot product bet^n each pair of point.

04 October 2023 20:05

[Not Important for Exam].

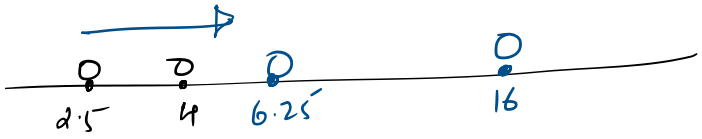
Let us Consider infinite Dimension Concept →.

Let a & b be two observations'. Let $r=0$, d $(a*b+0)^d = a^d \cdot b^d$

① If $d=2$ $(a*b+0)^2 = a^2 \cdot b^2$

$a = 2.5$ $(a*b+0)^2 = (2.5)^2 \cdot (4)^2$

$b = 4$ 6.25 16



Here dimension is not changed Instead the Kernel shifts the data down the orginal axis.

Summarize → Let a and b be two observation

$r=0$	$d=1$	$a^1 \cdot b^1$
$r=0$	$d=2$	$a^2 \cdot b^2$
$r=0$	$d=3$	$a^3 \cdot b^3$
$r=0$	$d=\infty$	$a^\infty \cdot b^\infty$

Let us add all

$= a^1b^1 + a^2b^2 + a^3 \cdot b^3 \ldots + a^\infty \cdot b^\infty$

$= (a, a^2, a^3 \ldots a^\infty) \cdot (b, b^2, b^3 \ldots b^\infty)$

This gives dot product with Coordinate for Infinite Number of dimension.

Radial Basis Kernel [Vimp]

* Works in Infinite Dimension.
* If there are lot of Overlapping in data points classification. then Support Vector Classifier cannot be used. to linearly Separate.
* One Way to deal with overlapping data is to use sum with Radial Kernel.
* Radial Kernel Uses Radial Basis function!

$$e^{-\gamma \cdot (a-b)^2}$$

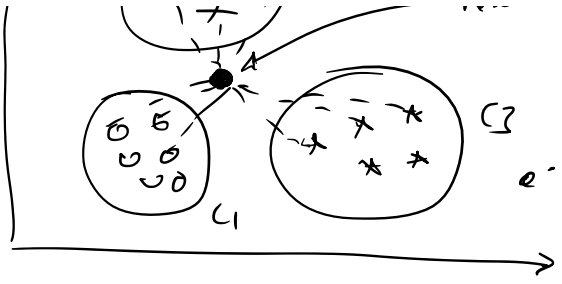
Here a and b are two observations.

$(a-b)^2 \Rightarrow$ diff b/w the measurement is squared. giving us the squared distance b/w two observation.

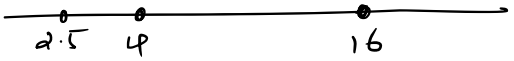
$\gamma \Rightarrow$ Scales the squared distance and thus Scales the influence.

* Since Radial Basis function finds Support Vector Classifier in Infinite Dimension it is not possible to visualise it.
* When applied to new observation, it behaves like weighted Nearest Model.
* Radial Kernel checks the influence of the Nearest

Influence of the Nearest Neighbour on New observation and accordingly makes classification.



* but see how radial Kernel determines how much influence each observation in training dataset has on classifying new observation.



but $a = 2.5$, $b = 4$ $\Rightarrow e^{-1(2.5-4)^2} \Rightarrow 0.11$ — (1)

$\gamma = 1$

$a = 2.5$, $b = 4$, $\gamma = 2$ $\Rightarrow e^{-2(2.5-4)^2} \Rightarrow 0.01$ } Thus γ scales the influence.

$a = 2.5$, $b = 16$, $\gamma = 1$ $\Rightarrow e^{-1(2.5-16)^2} \Rightarrow e^{-182.25} \approx$ close to zero

↳ (2)

From (1) & (2) if points are close enough then we have high influence.

and if points are far off then we have low influence.

* Thus to calculate influence b/t two data points, we can plug in the values in Radial Basis fn with appropriate γ.

Infinite Dimension.

* We get relationship in Infinite Dimension.

Not from Exam :- Let a and b are two observations.

Let $\gamma = 1/2$

$$e^{-\frac{1}{2}(a-b)^2} = e^{-\frac{1}{2}(a^2+b^2-2ab)} = e^{-\frac{1}{2}(a^2+b^2)} \cdot e^{ab} \quad \rightarrow (1)$$

Taylor Series Expansion $\Rightarrow$ [Allows any fn to split in infinite Sum]

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{\infty}(a)}{\infty!}(x-a)^{\infty}$$

Let $f(x) = e^x$.

$$e^x = e^a + \frac{e^a}{1!}(x-a) + \frac{e^a}{2!}(x-a)^2 + \cdots + \frac{e^a}{\infty!}(x-a)^{\infty}$$

Let $a = 0$

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \cdots \frac{x^{\infty}}{\infty!}.$$

$$\therefore e^{ab} = 1 + \frac{ab}{1!} + \frac{ab^2}{2!} + \cdots \frac{a^{\infty}b^{\infty}}{\infty!}$$

$$e^{ab} = \left(1, \frac{1}{\sqrt{1!}}a, \frac{1}{\sqrt{2!}}a^2 \cdots, \frac{1}{\sqrt{\infty!}}a^{\infty}\right) \cdot \left(1, \frac{1}{\sqrt{1!}}b, \frac{1}{\sqrt{2!}}b^2 \cdots, \frac{1}{\sqrt{\infty!}}b^{\infty}\right) \quad \rightarrow (2)$$

from (1) & (2)

$$= e^{-\frac{1}{2}(a^2+b^2)} \left(1, \frac{a}{\sqrt{1!}}, \frac{a^2}{\sqrt{2!}} \cdots \frac{a^{\infty}}{\sqrt{\infty!}}\right) \cdot \left(1, \frac{b}{\sqrt{1!}}, \frac{b^2}{\sqrt{2!}} \cdots, \frac{b^{\infty}}{\sqrt{\infty!}}\right)$$

Let $\varphi = e^{\frac{-1}{2}(a^2+b^2)}$

$$= \left(\varphi, 1, \frac{a}{\sqrt{1!}}, \frac{a^2}{\sqrt{2!}} \cdots \frac{a^\infty}{\sqrt{\infty!}}\right) \cdot \left(\varphi, 1, \frac{b}{\sqrt{1!}}, \frac{b^2}{\sqrt{2!}}, \cdots \frac{b^\infty}{\sqrt{\infty!}}\right)$$

We can see Radial Kernel is equal to Dot product that has coordinate for infinite No of Dimension.

11 October 2023 18:51

→ Dimensionality Reduction →

① PCA [Principal Component Analysis]

Consider

→

weight	Height	DBP	SBP	Health?

Diastolic BP

Systolic B.P.

Say Health of Person depends on 4 features

- weight ✓
- Height ✓
- DBP ✓
- SBP ✓

To represent this data ⇒ We need 4 Dimension

For more features ⟶ We need more Dimension.

* Visualizing data in More than 3 dimension is difficult.

* Computation on 4 features is also Complex

Possible Solution →

Height	weight	DBP	SBP	? Health.

BP [Blood Preserve]

[Body Mass Index] BMI

① look for Strong Correlated features: Here Height & weights } are Strongly correlated

Also DBP and SBP }

... Effective Correlation of Height & Weight ⇒ BMI

→ We can combine the effective correlation of Height & Weight ⇒ BMI

" " " " " " " DBP and SBP ⇒ BP.

Now Instead of 4 columns we have 2 columns.

BMI	BP.

→ This is 2-D data. ✓ Easy to Compute
✓ Easy to Visualize.

We have Reduced the Dimension of data from 4-D to 2-D.

Note → Consider 2 columns :-

Let → X_1 (DBP), X_2 (SBP)

DBP	SBP
78	126
80	128
81	127
82	130
84	130
86	132

$y = \alpha_1 X_1 + \alpha_2 X_2$

(BP)

α_1 & α_2 are weights of features

X_1 and X_2 are features

Ex

→ ① $BP = 0.8\,DBP + 0.6\,SBP$

We are giving more weights to DBP as compared to SBP.

Ex ② Let BP = mean of DBP & SBP.

$BP = \frac{DBP + SBP}{2} = 0.5\,DBP + 0.5\,SBP$

$\alpha_1 = 0.5$

$\alpha_2 = 0.5$

for mean

DBP	SBP	BP mean
78	126	102
80	128	104
81	127	104
82	130	106
84	130	107
86	132	109

Combine DBP and SBP.

~~82~~	150	
84	130	107
86	132	109

Ex (3) let BP = Sum of DBP and SBP.

$BP = DBP + SBP \Rightarrow \alpha_1 = 1$

$\alpha_2 = 1$

For Sum

DBP	SBP	Sum
78	126	204
80	128	208
81	127	208
~~82~~	130	212
84	130	214
86	132	218

Now PCA [⇒ Principal Component Analysis ✓

⇒ It is a method to find the Linear Combination that accounts for as much variability as possible in Combined Variable ↳(Maximum Variance)]

[To understand ⇒ $y = \alpha_1 x_1 + \alpha_2 x_2$

we want value of α_1 and α_2 such that the variance in the value of (y) should be Maximum

B'coz more the Variance → More is the information]

Note:⇒ for $y = \alpha_1 x_1 + \alpha_2 x_2$ (combined variable)

→ Here To maximize variance large value of α_1 and α_2 can be proposed

→ We will place restriction on value of weights i'e $(\alpha_1, \alpha_2 \dots)$ such that

$$\alpha_1^2 + \alpha_2^2 + \dots + \alpha_n^2 = 1$$

* If there are n features, we will need 'n' weights

In our Ex ⇒ we had 2 features ∴ we need α_1 and α_2

$$\therefore BP = \alpha_1 DBP + \alpha_2 SBP$$

constraints $\alpha_1^2 + \alpha_2^2 = 1$

but $\alpha_1 = 0.8$ $(0.8)^2 + (0.6)^2 = 0.64 + 0.36 = 1.00$ ✓

Let $\alpha_1 = 0.8$

$\alpha_2 = 0.6$

$(0.8)^2 + (0.6)^2 = 0.64 + 0.36 = 1.00$ ✓

Let Let $\alpha_1 = 0.8$ and $\alpha_2 = 0.6$

$BP = \alpha_1 DBP + \alpha_2 SBP$

DBP	SBP	BP
78	126	138
80	128	140.8
81	127	141.0
82	130	143.6
84	130	145.2
86	132	148.0
	Mean	142.8

$$\text{Variance} = \frac{1}{n-1} \sum_{i=1}^{n} (BP - \text{mean } BP)^2$$

$$= \frac{1}{5} \left((138 - 142.8)^2 + (140.8 - 142.8)^2 + (141.0 - 142.8)^2 + \dots \right)$$

$$= 12.74$$

Let for above 6 input Sample.

* we can take Many values of α_1 and α_2 Such that $\alpha_1^2 + \alpha_2^2 = 1$

⇒ and calculate variance for Each α_1 and α_2

	α_1	α_2	Var(y)
✓	0.8	0.6	12.74
✓	0.6	0.8	11.8
✓	0.98	0.2	10.4
✓	0.2	0.98	7.4

Of the above 4 Cases: we see for $\alpha_1 = 0.8$ and $\alpha_2 = 0.6$ we get Largest Variance = 12.74

Tarika 1 :- $BP = 0.8\,DBP + 0.6\,SBP$

To find α_1 and α_2 we got over α_1 and α_2

We got our α_1 and α_2

Taarika 2 →

Note → Consider again →

DBP	SBP
78	126
80	128
81	127
82	130
84	130
86	132
x	y

Construct Covariance Matrix $= \begin{bmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{bmatrix}$

$$cov(x,y) = \frac{1}{N-1} \sum (x_i - \bar{x})(y_i - \bar{y}) = cov(y,x)$$

For above Sample data set

Covariance matrix ⇒

	DBP	SBP
DBP	8.17	5.97
SBP	5.97	4.97

Eigen values = $\begin{bmatrix} -0.8 \\ -0.6 \end{bmatrix}$ ✓ ✓

from above Covariance Matrix.

$\therefore$ $BP = -0.8\,DBP - 0.6\,SBP$

$\therefore$ $BP = 0.8\,DBP + 0.6\,SBP$ ✓

Applications →

(1) Consider

DBP	SBP	weight	Height

BP BMI

$\Rightarrow$

BP	BMI

Reduced Dimension by Combining features.

BP BMI Reduced Dimension "1 Combining features.

(2)

DBP	SBP	Weight	Height	Pulse
78	126	82	5.1	90
80	128	74	5.4	80
81	127	64	5.11	70
82	130	82	6.2	90
84	130	92	6.3	95
86	132	85	5.10	94

Pat1
Pat2

Here if we have many no of features and we need to compare the Health chart/ parameters of samples, it is difficult

But if we combine the above 5 col in a col.

C1	C2
V1	V2
V3	V4
V5	V6

Now with less no of features comparing the Health chart/ parameters of samples is relatively Easy.

* PCA → It is Dimensionality Reduction Technique.

Consider: For House Price Prediction

Size	Location	YOC	OC Y/N	Builder	Color	Garden	Swimming	Metro Distance	Price

* Here Price of House depends on above features.
* Here Many No of features (Problem of Plenty).
* Issues:
  * Represent" Problem ✓
  * Overfitting Problem ✓
* We need to Reduce Dimension
* We need to Identify Important Components.

Dimensionality Reduction:-

* Reduces the dimension of feature space

  Ex if there are 100 features/col in dataset and you want to get only 10 features then with dimensionality reduction technique we can achieve this.

* It transforms dataset which is in n dimension space to n' dimension space where $n' < n$.

to n' dimension space where

Why Dimensionality Reduction ?

- Normally it is argued that many features gives More accurate result
- But after some point the model performance decreases (overfitting) with increase in number of features.
- This is Curse of Dimensionality
- So Dimension Reduction is Crucial.

* PCA enables us to identify the correlation and pattern in the dataset so that it can be transformed into new dataset with lower dimension without loss of important information.

Chapter 6

PCA (Principal Component Analysis)

LDA

SVD.

House Price Prediction

1	2	3	4	5	6	7	8	9	10
Size	Locality	YOC	OC Y/N	Builder	Color	Garden	Swimming	Distance from Metro	Price

" Problem of Plenty "

→ Representation problem

→ Overfitting problem

So we need to Reduce Dimension.

* we need to identify Important component.

* PCA is dimension Reduction Technique ✓

* ⇒ Dimensionality Reduction →

  * Reduces the dimension of feature space. [Reduces No of dimensions/features for Analysis.]

    Ex If there are 100 features/col^n in dataset and you want to get only 10 features then with dimensionality reduction technique we can achieve this.

  * It transforms dataset which is in [n dimension space] to n' dimension space where $n' < n$

Why Dimensionality Reduction →

↳ Normally it is argued that many features gives more accurate result.

↳ However after some point the performance of model decreases (Overfitting) with increase in no of features.

decreases (overfitting) with increase in no of features.

↳ This is known as "Curse of Dimensionality"

So Dimension Reduction is Crucial.

* PCA enables us to identify the correlation and pattern in a dataset so that it can be transformed into new dataset of significantly lower dimension without loss of important information.

Ex.

Height	Weight	DBP	SBP	Cholestrol
~~6 feet~~	~~70~~	90	110	?
~~Correlated~~		Highly Correlated		
cal				

BMI BP

⇒

BMI	BP
x_0^2	y_0^2

Steps in Performing PCA

step ① → get the data ✓

step ② → Substract the Mean and Produce New Dataset (Row Data Adjust)

step 3 → Calculate the Covariance Matrix.

step 4 → Calculate the Eigen Vectors and Eigen values of the Covariance matrix

step 5 → Choosing Components and forming a feature Vector.

Feature Vector = (eig1, eig2, eig3 ... eign)

step 6 → Derive the new data set

Final DataSet = Row Feature Vector * Row Data Adjust

* Row Feature Vector → is matrix with the Eigen Vectors in the column transposed so that Eigen Vectors are now in the rows with most significant Eigen Vectors on top.
* Final Datala Set is set with data items in col and dimensions along rows.

step 7 → Getting the old Data back.

step 7 ⇒ getting the old Data back.

Final Data = Row Feature Vector × Row Data Adjust.

Row Data Adjust = Row Feature Vector^{-1} * Final Data.

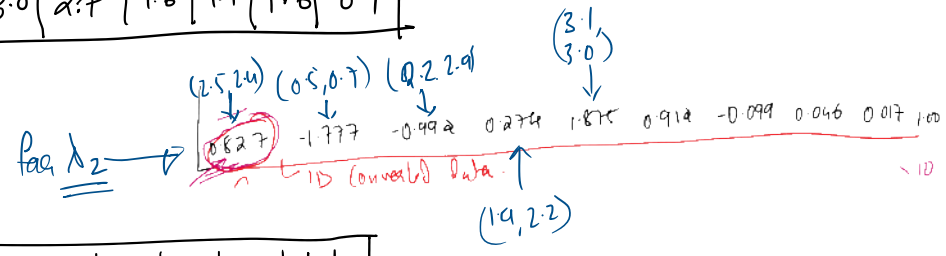
= Row Feature Vector^{T} * Final Data

Now Orginal Data = Row Data Adjust + Orginal Mean

Perform PCA on following Dataset

x	2.5	0.5	2.2	1.9	3.1	2.3	2.0	1.0	1.5	1.1
y	2.4	0.7	2.9	2.2	3.0	2.7	1.6	1.1	1.6	0.9

step1: given Dataset →



x	2.5	0.5	2.2	1.9	3.1	2.3	2.0	1.0	1.5	1.1
y	2.4	0.7	2.9	2.2	3.0	2.7	1.6	1.1	1.6	0.9

step2: calculate mean

$\bar{x} =$ 1.81 ✓

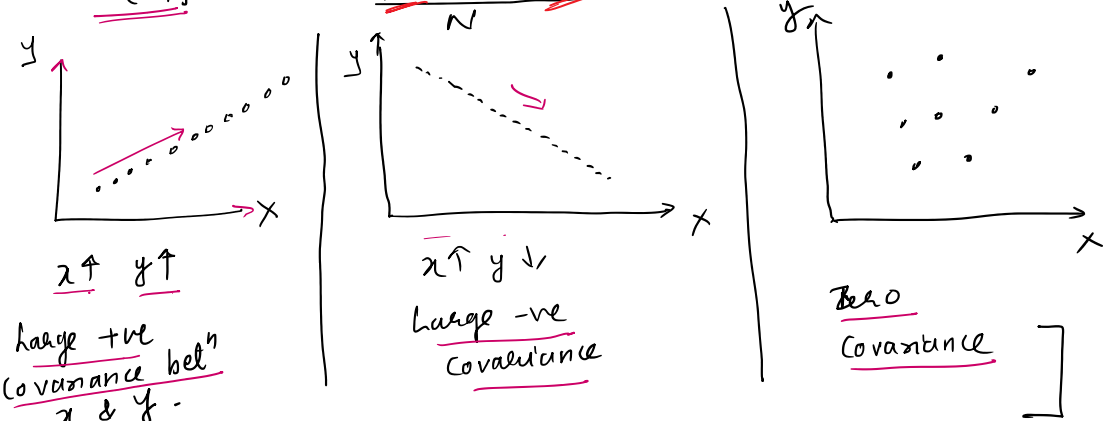
$\bar{y} =$ 1.91 ✓

step3 Adjusted Dataset (So the Dataset Mean is Zero).

$x = x - \bar{x}$	$x_i - \bar{x}$ 0.69	-1.31	0.39	0.09	1.29	0.49	0.19	-0.81	-0.31	-0.71
$y = y - \bar{y}$	$y_i - \bar{y}$ 0.49	-1.21	0.99	0.29	1.09	0.79	-0.31	-0.81	-0.31	-1.01

step4: [Note →
To find Covariance Matrix → it is measure of the extent to which the corresponding elements from two set of ordered data move in same direction.]

$$cov(x,y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{N}$$



Covariance Matrix → $C =$

$$\begin{bmatrix} cov(x,x) & cov(x,y) \\ 0.61655555 & 0.6154444 \\ 0.6154444 & 0.71655556 \end{bmatrix}$$

$\frac{1}{N}\sum (x_{ij} - \bar{x})(x_{ik} - \bar{x})$

Covariance Matrix → $C = \begin{bmatrix} & \\ 0.61544444 & 0.71655556 \end{bmatrix}$

$cov(x,x) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{ik} - \bar{x})(x_{jk} - \bar{y})$

$cov(x,x)$ $cov(y,x)$ $cov(y,y)$

Note $cov(x,y) = cov(y,x)$

Step 5: Compute Eigen Values & Vector.

$C - \lambda I = 0$

$$\begin{bmatrix} 0.61655555 - \lambda & 0.61544444 \\ 0.61544444 & 0.71655556 - \lambda \end{bmatrix} = 0$$

(x, y, cov(x,y), cov(y,x))

$\lambda^2 - 1.3332\lambda + 0.0630244 = 0$

$\lambda_1 = 0.04908340$

$\lambda_2 = 1.28462771 2$

Eigen values

* The largest Eigen value gives first PCA

Eigen Vector → Let $u = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ and $v = \begin{vmatrix} v_1 \\ v_2 \end{vmatrix}$ be Eigen Vectors for λ_1

$$\begin{bmatrix} 0.56747215 & 0.61544444 \\ 0.61544444 & 0.66747216 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$0.56767215 u_1 + 0.61544444 u_2 = 0$ — ①

$0.61544444 u_1 + 0.66747216 u_2 = 0$ — ②

Let $u_1 = 1$ in eq ①

$u_2 = -0.92205266$

$$\therefore u = \begin{bmatrix} 1 \\ -0.92205266 \end{bmatrix} \begin{matrix} u_1 \\ u_2 \end{matrix}$$

for λ_2

$$\begin{bmatrix} -0.66747216 & 0.61544444 \\ & \\ & -0.56747215 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} -0.667472 16 & 0.61544444 \\ 0.61544444 & -0.56747152 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

$$-0.667472 16 v_1 + 0.61544444 v_2 = 0 \quad -(3)$$

$$0.61544444 v_1 + -0.56747152 v_2 = 0 \quad -(4)$$

Consider Eq 3 put $v_2 = 1$

$$v_1 = 0.92220526 6$$

$$\therefore v = \begin{bmatrix} 0.92220526 6 \\ 1 \end{bmatrix} \quad \begin{matrix} v1 \\ v_2 \end{matrix}$$

(6) Normalize the Vector → to convert it into Vector of Unit length

How → divide it by length of Vector.

Length of Vector

$$u = \frac{1}{\sqrt{u_1^2 + u_2^2}} \begin{bmatrix} u1 \\ u_2 \end{bmatrix} = \frac{1}{1.360213596} \begin{bmatrix} 1 \\ -0.92220526 16 \end{bmatrix}$$

$$u = \begin{bmatrix} 0.735178 6533 \\ -0.67778734 \end{bmatrix}$$

$$v = \frac{1}{\sqrt{v_1^2 + v_2^2}} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{1}{1.360213596} \begin{bmatrix} 0.92220526 16 \\ 1 \end{bmatrix}$$

length of vector

$$v = \begin{bmatrix} 0.67778734 \\ 0.735778 6533 \end{bmatrix}$$

Since we got both the Eigen Vector, let us construct Feature Vector

$$\text{FeatureVector} = \begin{bmatrix} u & v \\ 0.735 778 6535 & 0.67778734008 \\ -0.67778734008 & 0.735778635 \end{bmatrix}$$

$$\text{Final Data} = \text{FeatureVector}^T \times \text{Data adjust}$$

$$= \begin{matrix} u \rightarrow \\ v \rightarrow \end{matrix} \begin{bmatrix} 0.735778 6535 & -0.67778734008 \\ 0.67778734008 & 0.735778635 \end{bmatrix} *$$

u = 0.735778635
v = 0.6778734008

$$\begin{bmatrix} -0.6678 \ldots \\ 0.735778635 \end{bmatrix}_{2\times 2} *$$

$$\begin{bmatrix} 0.69 & -1.31 & 0.39 & 0.09 & 1.29 & 0.49 & 0.19 & -0.81 & -0.31 & -0.71 \\ 0.49 & -1.21 & 0.99 & 0.29 & 1.09 & 0.79 & -0.31 & -0.81 & -0.31 & -0.71 \end{bmatrix}_{2\times 10}$$

depended on λ_1

$$\begin{bmatrix} 0.1751 \\ 0.827 \end{bmatrix} \quad \text{1D Converted dataset}$$

$$\begin{bmatrix} -0.1142 & -0.384 & -0.130 & 0.209 & -0.175 & 0.349 & -0.046 & -0.017 & -1.00 \\ -1.777 & -0.992 & 0.274 & 1.875 & 0.912 & -0.099 & 0.046 & 0.017 & 1.00 \end{bmatrix}_{2\times 10}$$

1D Converted Data.

From λ_2

Find Data of (2.5, 2.4) → 0.827 → will have contribution of both

Note Here only if λ_2 is used as it is greater & given first PCA

∴ Calculating feature Vector for λ_2 Let it be v as calculated above

$$v = \begin{bmatrix} 0.92205266 \\ 1 \end{bmatrix} \qquad \text{Feature Vector} = \begin{bmatrix} 0.92205266 \\ 1 \end{bmatrix}$$

only v

Final Dataset = Feature Vector^T × Adjusted Data.

$$= \begin{bmatrix} 0.92205266 & 1 \end{bmatrix}_{1\times 2} * \begin{bmatrix} \_\_\_\_\_ \end{bmatrix}_{2\times 10}$$

$$= \begin{bmatrix} \_\_\_\_\_ \end{bmatrix}_{1\times 10}$$

Single Row

Original Dataset = 2×10

Final Dataset = 1×10 ← Reduced Dimension from 2D to 1D.

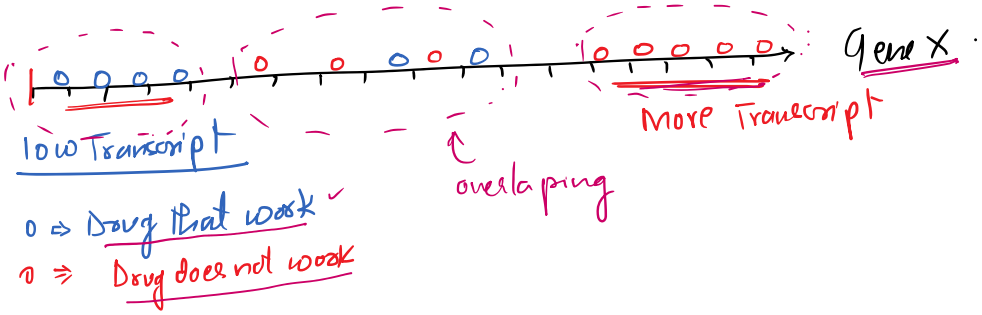
PCA
- LDA (Linear Discriminant Analysis)
- SVD (Singular Valued Decomposition)

LDA [Linear Discriminant Analysis].

→ Suppose ⇒ We got a Cancer drug ✓
↳ It works great for some people ✓
↳ But it make it worse for other people ✓

* How do we decide whom to give the drug ?
* May be gene study of patients will be of some help.

① Consider only one gene → Gene X. [1-D]

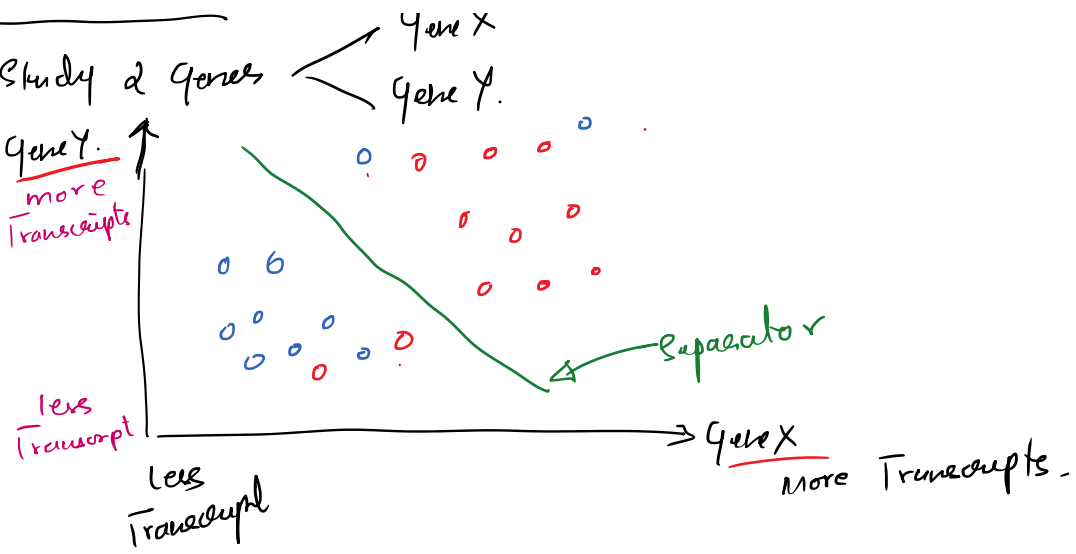


* From above we can see In cases with More Transcript drug is not working.
* Whereas In cases of low Transcript the drug tends to work
* How there are few exceptions (there are overlaps).
* Summary → Gene X does OK job but it has few Overlaps.

* Can we do Better ?

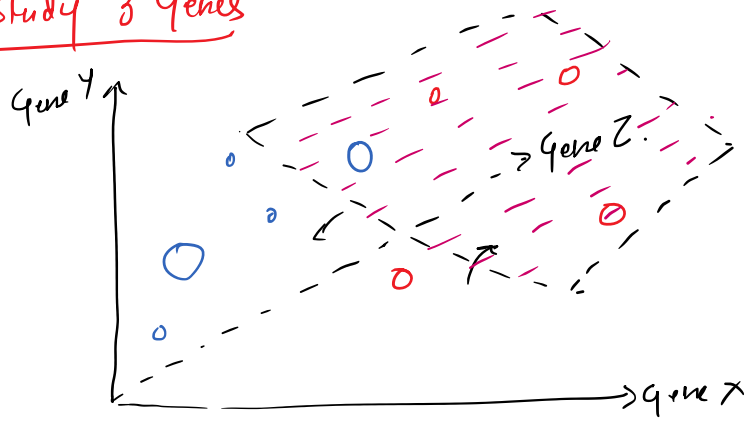
Let us study 2 genes < Gene X / Gene Y.

Let us study 2 Genes — Gene X, Gene Y.



It is better than studying only one gene but still there are overlapping.

Let us study 3 Genes



In case of 3 Genes we will need to represent the information in 3-D.

-> Here the separator is a plane.

Suppose we want to study 4 genes — Need 4-D Representn. -> Can't Draw 4D Graph.

* Here more the Dimension, more will be Complication in graph representation.

* PCA reduces dimensions by focussing on the genes with most Variance (more the variance, more the information).

* Normally PCA is useful in plotting data with lot of dimension (or genes) onto simple xy plane. →

* Here we are not interested much in identifying genes with most variance

Instead (LDA) we are interested in Maximizing Separability bet^n the two groups so we can make best classification decision.

Note

PCA → Help to identify feature with Max variation by reducing no of Component

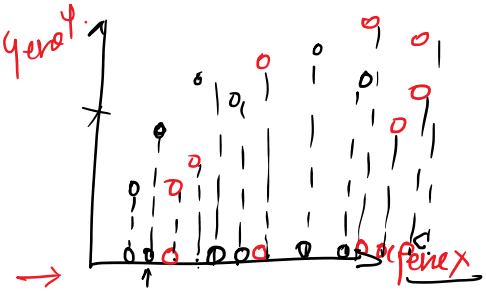
LDA → Helps to build better Separator/classifier by reducing Component

* Most possible ways to Reduce Dimension →

(1) We have study of 2 genes → Ignore gene Y.

So project all data point on X axis

→ This is not very good idea because it ignores the useful information that genes Y provides.



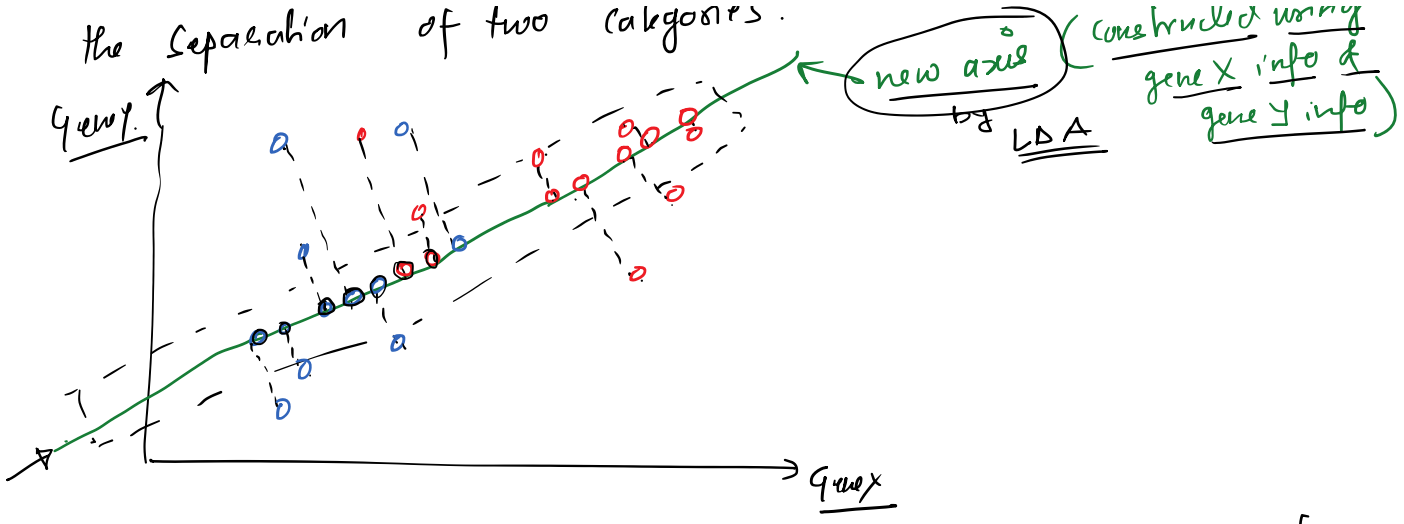
* LDA provides better way.

→ LDA reduces a 2-D graph to 1-D graph.

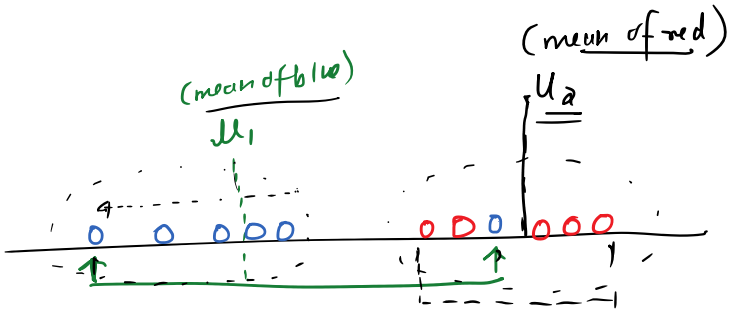
→ LDA uses both the genes to create a new axis and project the data on new axis in a way to maximize the Separation of two categories.

new axis (constructed using gene X info &

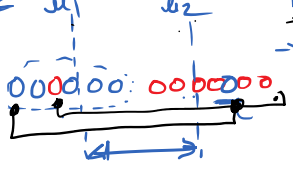
the Separation of two categories.



How LDA creates a New Axis → μ_1 and μ_2 are mean of respective class.



Case 1) Means of two classes are near.



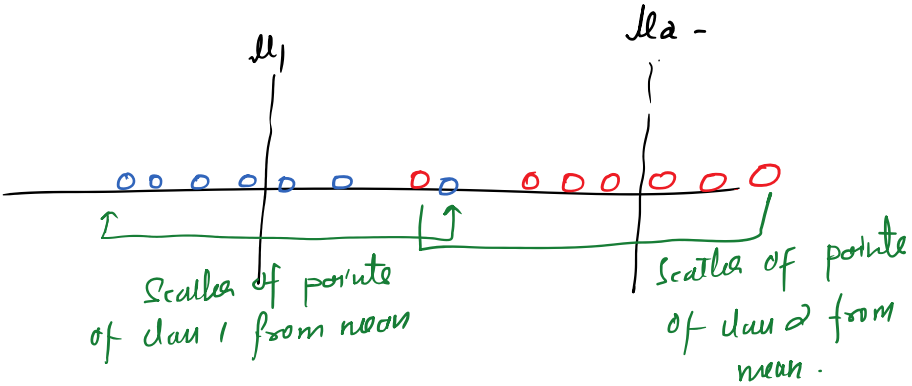
→ the points in each class are widely scattered

The new axis is created according to two criteria (considered simultaneously)

① Maximize the distance between mean of two classes)

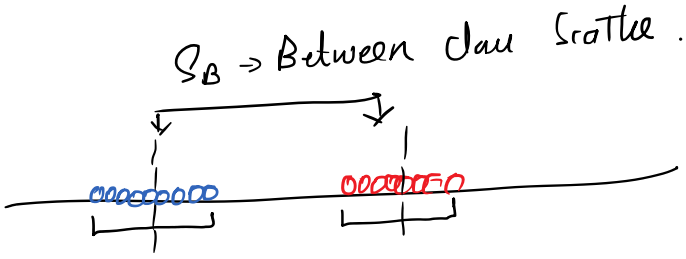
② Minimize the variation (LDA calls it scatter) and is represented by s^2) for each category.

Case 2) Scatter of points within class is large



S_B → Between class Scatter.

Case 3) Scatter of points within class is small.



S_w → within class scatter

Scatter of points is small within class.

Goal → S_B ⇒ Maximize
S_w ⇒ Minimize
} must be focussed on both simultaneously.

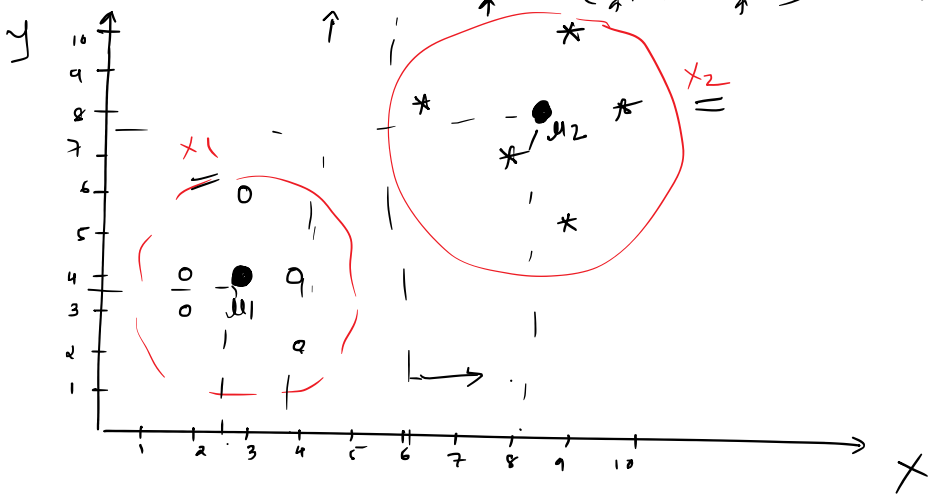
Mathematical Represent^n ⇒ $\frac{(\mu_1 - \mu_2)^2}{S_{w_1}^2 + S_{w_2}^2}$ = Square of diff bet^n two mean / Sum of scatter within each category/class.

Ideally $\frac{(\mu_1 - \mu_2)^2}{S_{w_1}^2 + S_{w_2}^2}$ ⇒ large ✓
⇒ small. ✓

Consider ⇒ Compute the (linear Discriminant projection) for the following two Dimensional Dataset.

Class1 → $X_1 = (x_1, y_1) = \{(4,2), (2,4), (2,3), (3,6), (4,4)\}$ ⇒ o

Class2 → $X_2 = (x_2, y_2) = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$ ⇒ *



Step 1) Find Class Means

$$\mu_1 = \frac{1}{N_1} \sum x \in X_1$$

$$= \frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix} + \begin{pmatrix}2\\4\end{pmatrix} + \begin{pmatrix}2\\3\end{pmatrix} + \begin{pmatrix}3\\6\end{pmatrix} + \begin{pmatrix}4\\4\end{pmatrix}\right] = \frac{1}{5}\begin{pmatrix}15\\19\end{pmatrix}$$

$$= \begin{pmatrix}3\\3.8\end{pmatrix}$$

$$\mu_2 = \frac{1}{N_2} \sum x \in X_2$$

$$= \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix} + \begin{pmatrix}6\\8\end{pmatrix} + \begin{pmatrix}9\\5\end{pmatrix} + \begin{pmatrix}8\\7\end{pmatrix} + \begin{pmatrix}10\\8\end{pmatrix}\right] = \frac{1}{5}\begin{pmatrix}42\\38\end{pmatrix}$$

$$= \begin{pmatrix}8.4\\7.6\end{pmatrix}$$

Step 2) Covariance Matrix of Both Class.

Covariance Matrix of Class 1. (X_1).

$$S_1 = \frac{1}{N-1} \sum_{x \in X_1} (x - \mu_1)(x - \mu_1)^T$$

$$S_1 = \frac{1}{N-1} \sum_{x \in X_1} (x - \mu_1)(x - \mu_1)^T$$

$$= \frac{1}{4} \left[\left[\begin{pmatrix} 4 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix} \right]^2 + \left[\begin{pmatrix} 2 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix} \right]^2 + \left[\begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix} \right]^2 \right.$$

(x_1, y_1) mean

$$\left. + \left[\begin{pmatrix} 3 \\ 6 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix} \right]^2 + \left[\begin{pmatrix} 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix} \right]^2 \right]$$

$$= \begin{bmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{bmatrix}$$

Covariance Matrix of Second class

$$S_2 = \frac{1}{N-1} \sum_{x \in X_2} (x - \mu_2)(x - \mu_2)^T$$

$$= \frac{1}{4} \left[\left[\begin{pmatrix} 9 \\ 10 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]^2 + \left[\begin{pmatrix} 6 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]^2 + \left[\begin{pmatrix} 9 \\ 5 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]^2 \right.$$

$$\left. + \left[\begin{pmatrix} 8 \\ 7 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]^2 + \left[\begin{pmatrix} 10 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]^2 \right]$$

$$= \begin{bmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{bmatrix}$$

Step 3 Within Class Scatter Matrix (Minimize)

$$S_W = S_1 + S_2$$

$$= \begin{bmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{bmatrix} + \begin{bmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{bmatrix}$$

$$= \begin{bmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{bmatrix}$$

Step 4 :> Between Class Scatter Matrix (Maximize).

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

$$= \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right] \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]^T$$

$$= \begin{bmatrix} -5.4 \\ -3.8 \end{bmatrix} \begin{bmatrix} -5.4 & -3.8 \end{bmatrix}$$

$$S_B = \begin{bmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{bmatrix}$$

Step 5 The LDA projection is then obtained as solution of the generalized Eigen Value Problem

$$S_w^{-1} S_B w = \lambda w.$$

$$\Rightarrow |S_w^{-1} S_B - \lambda I| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$= \left| \begin{pmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{pmatrix} \right| \quad \text{— I}$$

$$\Rightarrow (9.2213 - \lambda)(2.9794 - \lambda) - 6.489 \times 4.2339 = 0$$

$$\lambda^2 - 12.2007\lambda = 0 \quad \Rightarrow \quad \lambda(\lambda - 12.2007) = 0$$

$\lambda \Rightarrow$ $\lambda_1 = 0$, $\lambda_2 = 12.2007$ 2 Eigen Values.

Now By Substituting $\lambda = \lambda_1$ in I we get

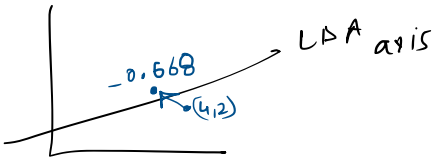
Eigen Vector for Eigen $= w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix}$

Eigen vector 1 = w_1 = for Eigen value λ_1 $\begin{pmatrix} 0.8178 \end{pmatrix}$

By Substituting $\lambda = \lambda_2$ in I we get

Eigen Vector 2 = w_2 = for Eigen value λ_2 $\begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix}$

Step 6 Find Data $= y = w^T x$ ← Input Data.
↑ Projection Vector



For Projection Vector w_1

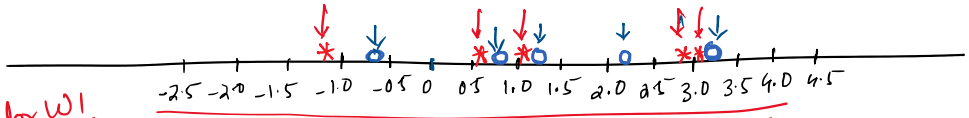
Find Data of $X_1 = w_1^T X_1 = [-0.5755 \ 0.8178] \begin{bmatrix} \begin{pmatrix} 4 \\ 2 \end{pmatrix} & \begin{pmatrix} 2 \\ 4 \end{pmatrix} & \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \begin{pmatrix} 3 \\ 6 \end{pmatrix} & \begin{pmatrix} 4 \\ 4 \end{pmatrix} \end{bmatrix}$
(1×2) (2×1) (2×1) (2×1) (2×1) (2×1)

$= [-0.668 \quad 2.12 \quad 1.3022 \quad 3.18 \quad 0.9688]$ (1×1)
Blue

Final Data of X_2 * $= w_1^T X_2 = [-0.5755 \ 0.8178] \begin{bmatrix} \begin{pmatrix} 9 \\ 10 \end{pmatrix} & \begin{pmatrix} 6 \\ 8 \end{pmatrix} & \begin{pmatrix} 9 \\ 5 \end{pmatrix} & \begin{pmatrix} 8 \\ 7 \end{pmatrix} & \begin{pmatrix} 10 \\ 8 \end{pmatrix} \end{bmatrix}$

$= [2.9985 \quad 3.0894 \quad -1.0905 \quad 1.1206 \quad 0.7874]$
Red

Class 1 & Class 2 on LDA axis Wrong Eigen Vector w_1



Here the projection vector w_1 constructed using Smaller Eigen value leads to bad Separability. ($\lambda_1 = 0$)

For Projection Vector w_2

Final Data $X_1 = w_2^T X_1$

$= [0.9088 \quad 0.4173] \left[\begin{pmatrix} 9 \\ 2 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} \begin{pmatrix} 3 \\ 6 \end{pmatrix} \begin{pmatrix} 4 \\ 4 \end{pmatrix} \right]$

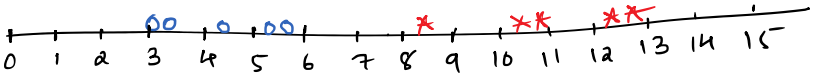
$= [4.4698 \quad 3.4868 \quad 3.0695 \quad 5.2202 \quad 5.3044]$

class 1 on LDA using w_2 vector ✓

Final Data $X_2 = w_2^T X_2$

$= [0.9088 \quad 0.4173] \left[\begin{pmatrix} 9 \\ 10 \end{pmatrix} \begin{pmatrix} 6 \\ 8 \end{pmatrix} \begin{pmatrix} 9 \\ 5 \end{pmatrix} \begin{pmatrix} 8 \\ 7 \end{pmatrix} \begin{pmatrix} 10 \\ 8 \end{pmatrix} \right]$

class 2 on LDA using w_2 vector $\longrightarrow = [12.3522 \quad 8.7912 \quad 10.2657 \quad 10.1957 \quad 12.4264]$



Here the projection vector (w_2) corresponding to larger Eigen value λ_2 leads to good separability.

Thus is LDA → when the 2D data points reduced to 1-D data points

→ and also using projection vector were able to predict a good separator of classes.

SVD [Singular Valued Decomposition]

* We normally use 2D matrix to represent Data Values. where column represents features and row represents samples data points
* Matrix Computation with all the values in matrix sometime become redundant or computationally expensive.
* we need to represent matrix in a form such that the most important part of matrix which is needed for further computation could be extracted easily.
* This can be done by SVD

SVD Theorem → A rectangular matrix A_{mn} can be decomposed it into product of 3 matrices.

$$A_{mn} = U_{m*m} \; \Sigma_{m \times n} \; V^T_{n \times n}$$

$m \times m \quad m \times n \quad n \times n$
$\hookrightarrow m \times n \quad \hookrightarrow m \times n$

where $U_{m \times m}$ ⇒ Orthogonal Matrix.
$\Sigma_{m \times n}$ ⇒ Diagonal matrix of Eigen Values.
$V^T_{n \times n}$ ⇒ Transpose of orthogonal matrix $V_{n \times n}$

Columns of U are the orthonormal Eigen Vectors of AA^T & Columns of V are the orthonormal Eigen Vectors of A^TA & Σ is diagonal matrix.

The elements of Σ are the square roots of Eigen values of

U & V in decreasing order.

Ex Find the SVD of $A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}_{2\times 3}$

Note we have to Decompose the matrix A into $U \Sigma V^T$.

To find U, Σ, V^T.

Solution → To find U := $AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$

For Eigen Values ⇒ $\begin{vmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{vmatrix} = 0$ ——— (I)

$\Rightarrow \lambda^2 - 22\lambda + 120 = 0$

$\boxed{\lambda_1 = 10}$ $\boxed{\lambda_2 = 12}$

For Eigen Vector → $\begin{bmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

(case 1) Substitute $\lambda = \lambda_1 = 10$

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\frac{x_1}{1} = \frac{-x_2}{1} = 1$

$\therefore x_1 = 1 \quad x_2 = -1$

Eigen Vector $X_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

(cont'd) Substitute $\lambda = \lambda_2 = 12$

$$\begin{bmatrix} 11-\lambda & 1 \\ 1 & 11-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Eigen Vector $X_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

* In U the Eigen Vector generated by largest Eigen value will be the first column.

$$\therefore U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Eigen vector of $\lambda = 12$ ↑ Eigen Vector of $\lambda = 10$

Now we need to Normalize the matrix ⇒ divide by length of respective Vector.

$$* \quad U = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} //$$

length of Vector $\lambda = 12$: $\sqrt{1^2+1^2} = \sqrt{2}$

length of Vector $\lambda = 10$: $\sqrt{(1)^2+(-1)^2} = \frac{1}{\sqrt{2}}$

Step 2 To find V

$$A^T A = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix} //$$

Here for Eigen values

$$\lambda^3 - 22\lambda^2 + 132\lambda = 0$$

$$\therefore \boxed{\begin{aligned} \lambda_1 &= 0 \\ \lambda_2 &= 10 \\ \lambda_3 &= 12 \end{aligned}}$$

We need to find Eigen Vectors for the 3 Eigen values.

Case 1) Eigen Vector for $\lambda_1 = 0$

$$\Rightarrow \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

By Cramers Rule

$$\frac{x_1}{\begin{vmatrix} 0 & 2 \\ 10 & 4 \end{vmatrix}} = \frac{x_2}{\begin{vmatrix} 10 & 2 \\ 0 & 4 \end{vmatrix}} = \frac{x_3}{\begin{vmatrix} 10 & 0 \\ 0 & 10 \end{vmatrix}}$$

$$= \frac{x_1}{-20} = -\frac{x_2}{40} = \frac{x_3}{100} = \frac{-1}{20}$$

$$x_1 = 1 \quad x_2 = 2 \quad x_3 = -5$$

Eigen Vector $X_1 = \begin{bmatrix} 1 \\ 2 \\ -5 \end{bmatrix}$

Case 2) Eigen Vector for $\lambda = 10$

$$\begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 4 \\ 2 & 4 & -8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

By Cramers Rule

$$\frac{x_1}{\begin{vmatrix} 0 & 4 \\ 4 & -8 \end{vmatrix}} = \frac{-x_2}{\begin{vmatrix} 0 & 4 \\ 2 & -8 \end{vmatrix}} = \frac{x_3}{\begin{vmatrix} 0 & 0 \\ 2 & 4 \end{vmatrix}} \Rightarrow \frac{x_1}{-16} = \frac{-x_2}{-8} = \frac{x_3}{0} = \frac{-1}{8}$$

$\therefore x_1 = 2 \quad x_2 = -1 \quad x_3 = 0$

Eigen Vector $= X_2 = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}$

Case 3) Eigen Vector for $\lambda_3 = 12$

$$\begin{bmatrix} -2 & 0 & 2 \\ 0 & -2 & 4 \\ 2 & 4 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

By Cramers Rule

$$\frac{x_1}{\begin{vmatrix} 0 & 2 \\ -2 & 4 \end{vmatrix}} = \frac{-x_2}{\begin{vmatrix} -2 & 2 \\ 0 & 4 \end{vmatrix}} = \frac{x_3}{\begin{vmatrix} -2 & 0 \\ 0 & -2 \end{vmatrix}} \Rightarrow \frac{x_1}{4} = \frac{-x_2}{-8} = \frac{x_3}{4} = \frac{1}{4}$$

$\therefore x_1 = 1 \quad x_2 = 2 \quad x_3 = 1$

Eigen Vector $X_3 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

Now $V = \begin{bmatrix} 1 & 2 & 1 \\ 2 & -1 & 2 \\ 1 & 0 & 5 \end{bmatrix}$

↑ Eigen Vector for $\lambda = 12$ ↑ Eigen Vector for $\lambda = 10$ ↑ Eigen Vector for $\lambda = 0$.

Now Normalizing the matrix → Divide by length of Vector.

$$V = \begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{5} & 1/\sqrt{30} \\ 2/\sqrt{6} & -1/\sqrt{5} & 2/\sqrt{30} \\ 1/\sqrt{6} & 0 & -5/\sqrt{30} \end{bmatrix}$$

↑ Length of Vector $\sqrt{6}$ ↑ Length of Vector $\sqrt{10}$ ↑ Length of Vector $\sqrt{30}$

$$\begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{5} & 1/\sqrt{30} \end{bmatrix}$$

$$V^T = \begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{30} & 2/\sqrt{30} & -5/\sqrt{30} \end{bmatrix}$$

Step 3 To find Σ (or D)

$$\Sigma = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \\ 0 & 0 & \sqrt{0} \end{bmatrix}_{3\times 3}$$

diag matrix ⇒ the diag elements are square root of Eigen values in Decreasing order.

$$\therefore A = U \Sigma V^T$$

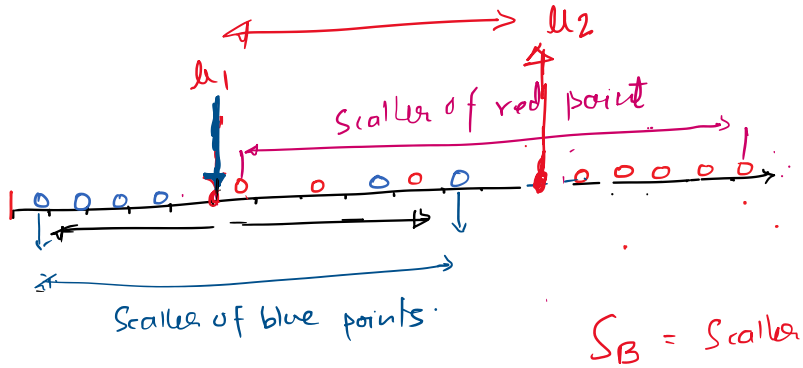
$$= \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{30} & 2/\sqrt{30} & -5/\sqrt{30} \end{bmatrix}$$

U Σ V^T

$=$

Consider 2 groups.

* Group of Blue points
* Group of Red points



μ_1 = Mean of blue point

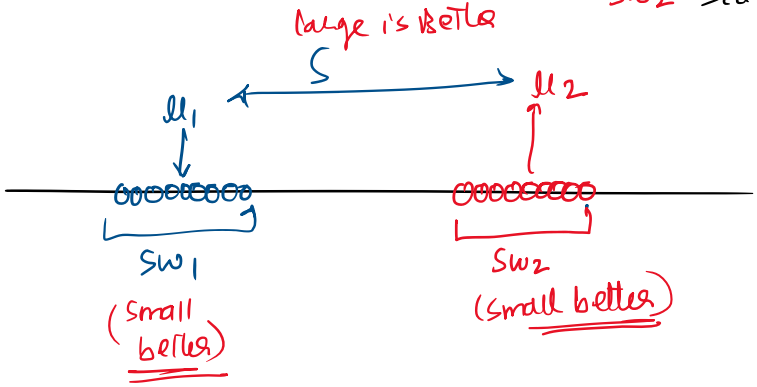
μ_2 = mean of red point

S_B = Scatter between two groups
= → Should be large

S_W ⇒ Scatter with a group.
= Should be

S_{W_1} = Scatter within group 1 (Blue points)

S_{W_2} = Scatter within group 2 (Red points)



Note In math Any Square Matrix A can be represented as (Decomposed)

3 component Matrix

$$A = U \Sigma U^T$$

- U: Eigen Vector
- U^T: Transpose of Eigen Vector
- Σ: Diagonal Matrix of Eigen Value.

This makes majorly of operations on matrix to be easy as operation on Diagonal Matrix is Easy.

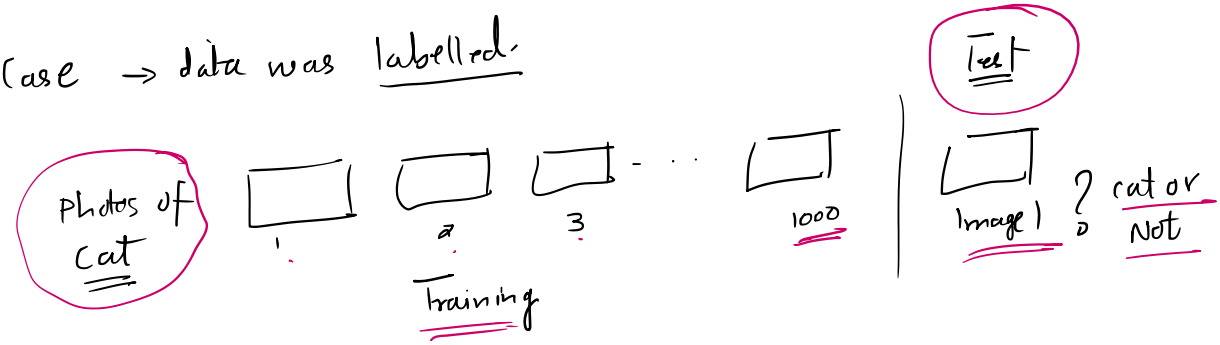
as operation on Diagonal matrix is Easy.

In Real life → Dataset → Matrix → Rectangular Matrix

Module 6

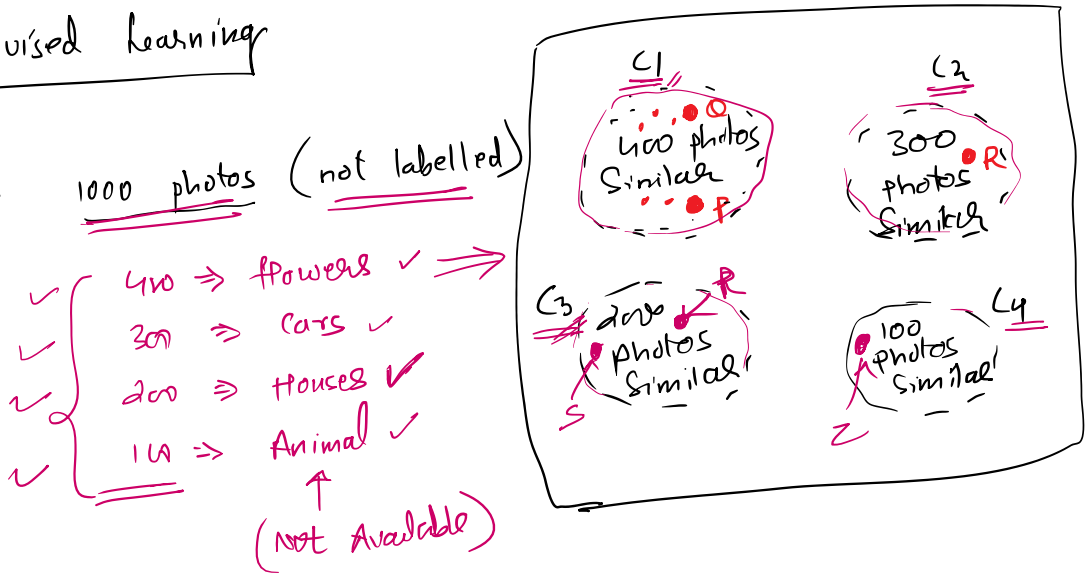
Clustering →

In earlier case → data was labelled.



Supervised learning

Now Consider → 1000 photos (not labelled)

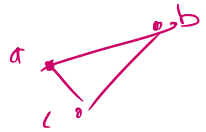


Clustering is the task of dividing the population or data points into no of groups such that the data points in the same group are more similar to other data points in same group than those in other group.

Clustering →
"Aim is to Segregate groups with Similar traits/features and assign them into clusters."

Distance Metric : It is a parameter that tells how close two points are

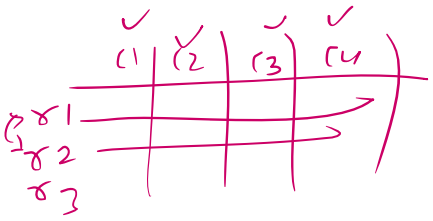
Given element a, b, c in a set, a distance metric is defined as a function with following properties:

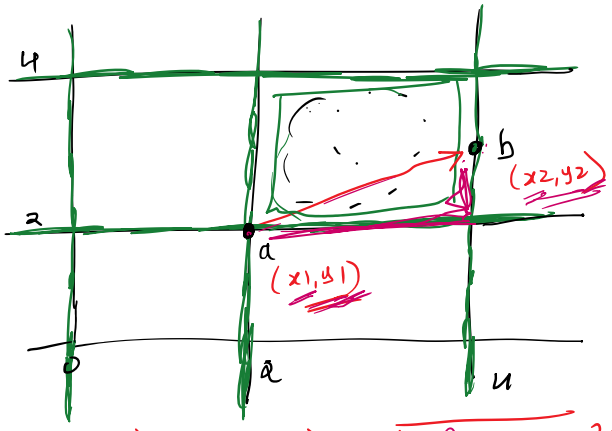


1. Non-negativity → $d(a,b) \geq 0$ +ve
2. Symmetry → $d(a,b) == d(b,a)$
3. Triangle Inequality $d(a,c) \leq d(a,b) + d(b,c)$. $\forall a,b,c \in S$

Different Distance Metric

1. Euclidean Distance → Most commonly used.
   → When data is dense or continous, this is the best proximity measure.





Euclidean Distance (vector) $= d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$

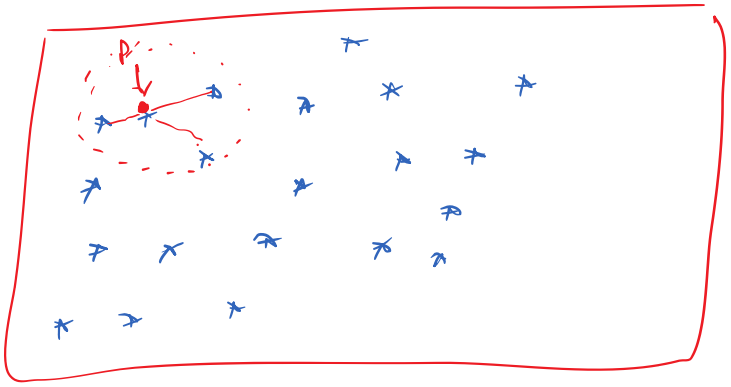
$= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

If for a point $P = (p_1, p_2 \ldots p_n)$ and $q = (q_1, q_2 \cdots q_n)$.

$$d(p,q) = d(q,p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

$= \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$ ✓

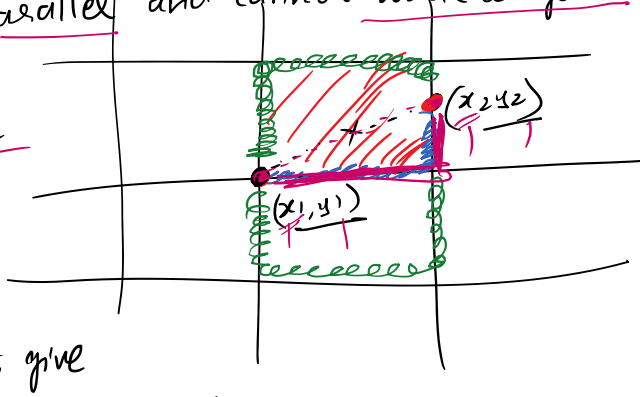


(2) Manhattan Distance

* If we are trying to find the distance bet^n two buildings that are separated by several blocks. We can only walk through sidewalk that are parallel and cannot walk diagonally through buildings

In such case we cannot use Euclidean distance.

Here Euclidean distance will not give realistic estimate for the distance



Manhattan distance.

$= |x_1 - x_2| + |y_1 - y_2|$ ✓

* Manhattan Distance is a metric in which the distance between two points is the sum of absolute difference of their Cartesian coordinates.

* Total sum of difference bet^n the x coordinates & y coordinates.

* Manhattan distance metric is also known as Manhattan length, Rectilinear distance, L1 distance, L1 Norm, City block distance taxi cab metric.,

$a = (a_1, a_2 \ldots a_n)$
$b = (b_1, b_2 \ldots b_n)$

has n features.

* If the data points has n features.

Manhattan Distance $d(a,b) = |a_1 - b_1| + |a_2 - b_2| + |a_3 - b_3| \cdots + |a_n - b_n|$

$$= \sum_{i=1}^{n} |a_i - b_i|$$

Syllabus

- Graph Based Clustering Using Minimum Spanning Tree
- Estimate Maximization Algo for Clustering
- DBSCAN

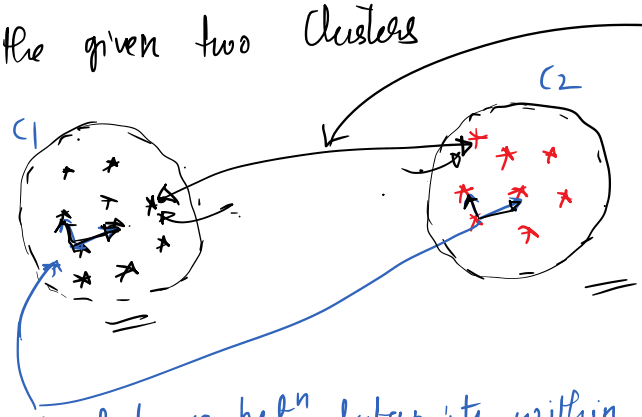
(1) Graph Based Clustering →

(1) First Consider How to Construct Graph of given data.

→ k Neighbourhood Graph. ✓

→ Epsilon Neighbourhood Graph. ✓

Consider the given two Clusters

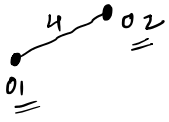


Distance betⁿ data points betⁿ diff clusters are Maximum.

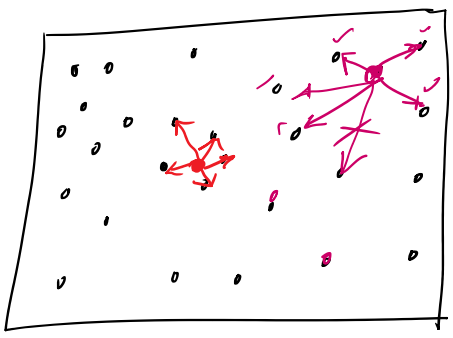
Here the distance betⁿ datapoints within the cluster is Minimum.

* In Graph Based Clustering the objects are represented as nodes in graph.

* The weight of edge / branch betⁿ the objects defines the distance betⁿ the objects



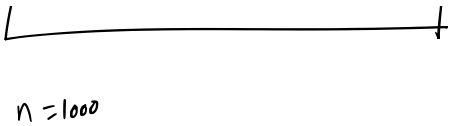
(1) k Neighbourhood Approach.



* To find distance of a point with all other points will be computationally Challenging.
* we will consider nearest k points

Let k=5. ✓

So we will calculate distance of nearest 5 data points and will be



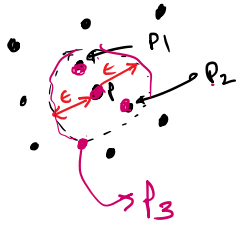
n = 1000

So we will [illegible]
nearest 5 data points are will be labelled as Nearest Neighbour of node

Challenge → Choosing K value is challenging task in graph Construction.

(2) Epsilon Neighbourhood Graph → Objects within a radius of epsilon from a given object are considered nearest neighbour.

Here P_1 & P_2, P_3 are neighbours of P



Challenge :- Selecting a proper epsilon value.

* Graph Based Clustering Using Minimum Spanning Tree →

→ Subgraph of given graph
→ Same No of vertex
→ and no cycle
→ Subgraph is connected

Algo

1. Determine MST of given graph. ✓
2. Delete (edge) branch iteratively. ✓
3. Each Connected Component is a cluster ✓

Different Strategies to Delete Branches.

(1) Delete Branch with max weight → ✓
(2) Delete Inconsistent Branch → ✓
(3) Delete by Analysis of weight. ✓

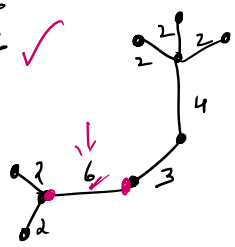
(1) Deletion of Branch with max weight

1. In each step create two clusters by deleting the branch with

1. In each step create two clusters by deleting the branch with max weight
2. Repeat until the desired no of clusters is reached

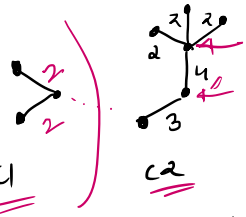
given no of cluster desired is 3

Consider MST

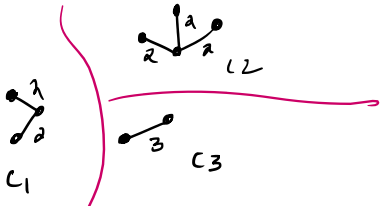


MST

Step 1) delete edge with weight 6



Step 2) delete edge with weight 4.

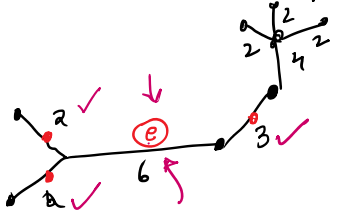


3 clusters

(2) Delete Inconsistent Branches.

* A branch is inconsistent if the corresponding weight (d_e) of branch is much larger than the reference value $\bar{d}_e$
* The reference value $\bar{d}_e$ can be defined by the average weight of all the branches adjacent to edge e (edges)

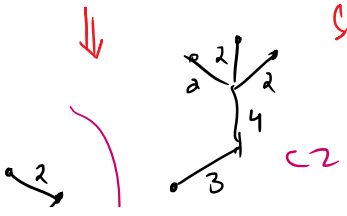
Consider MST:

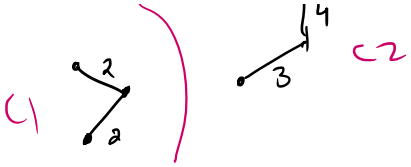


Here $\bar{d}_e = \frac{2+2+3}{3} = \frac{7}{3} = 2.3$

Here $d_e = 6 > \bar{d}_e$

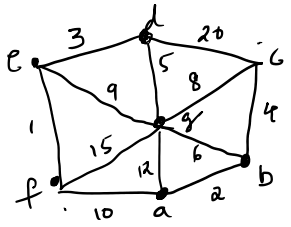
So delete the edge e"

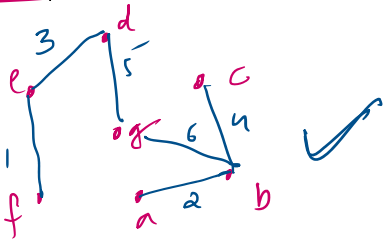




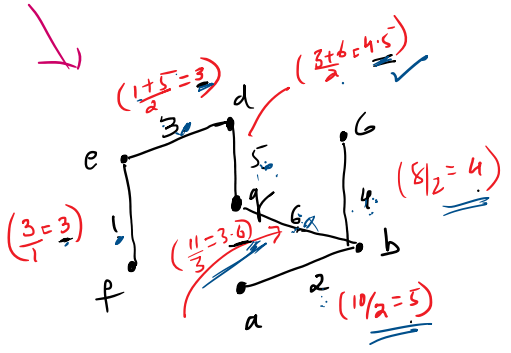
2 clusters

Q) Construct MST and provide Clustering of graph Using Inconsistent branches.





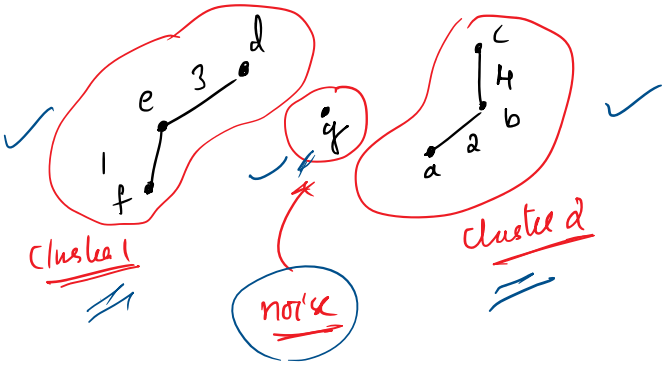
Step 1) MST of the Graph.



Step 2) For deletion we are using Inconsistent branch approach.

Here edge with 5 > reference value (4.5) so delete edge 5

Also edge with 6 > reference value (3.6) so delete edge 6.



(2) Estimation Maximization Algo

→ This Algo is for Maximum Likelihood Estimation in presence of latent variable (missing).

→ The most discussed application of EM Algorithm is used for Clustering with Mixture Model. (where data points have missing value)

* Estimation (E-step) → Estimate are expectations for machine & then classify the data into some classes.

* Maximization (M-step) → Whatever we estimated should now be maximized.

* Iteratively Repeat E-step and M step. until the values converges (no difference)

Imagine there are 2 coins A & B (Biased)
Here One is likely to get Head more no of times and other is likely to get tail more no of tail

* You pick one at random and toss it
Which one was it

Sol: Repeat this 5 time.
{
* pick a coin randomly (known whether A or B)
* Toss 10 times
* Read No of Head & Tails
}

Consider the Result

	Coin A	Coin B
First Round		5H, 5T
Second Round	9H, 1T	

Consider the

First Round		5H, 5T
Second Round	9H, 1T	
Third Round	8H, 2T	
Fourth Round		4H, 6T
Fifth Round	7H, 3T	
Total	24H, 6T	9H, 11T

Probability of getting Head with Coin A $= \frac{24}{30} = 0.80$

" " " " " Coin B $= \frac{9}{20} = 0.45$

Coin A yields head 80% of time and Coin B 45% of time.

* What if only results are given.
& (we need to guess the percentage of heads that each coin yields &) Part 1
Also (need to guess which coin was picked at each round of toss.) Part B

→ Coin Not given

Result

First Round		5H, 5T	?
Second Round	9H, 1T		?
Third Round	8H, 2T		?
Fourth Round		4H, 6T	?
Fifth Round	7H, 3T		?

Initial Assumption.

Assume → the probability of Head for

Coin A = 60% ⇒ $\theta_A = 0.6$
Coin B = 50% ⇒ $\theta_B = 0.5$

Round only

Initial

First Round ⇒ (5H, 5T.)

Compute the likelihood that it was Coin A & Coin B Using

Binomial Distribution i.e.

$$P(k) = \theta^k (1-\theta)^{n-k}$$

n = no of toss
k = successful (head)

θ => is probability on n trails with k success

Likelihood of A = $P_A(h)^k (1 - P_A(h))^{n-k}$

Here n=10, k=5

$= (0.6)^5 \times (1-0.6)^{10-5} = 0.000796824$

Tail

Likelihood of B = $P_B(h)^k (1 - P_B(h))^{n-k}$

n=10 k=5

$= (0.5)^5 \times (1-0.5)^{10-5} = 0.0009765625$

Normalize by Using $\frac{A}{A+B}$ [To Convert likelihood into Probability]

for coin A = $0.000796824 / (0.000796824 + 0.0009765625)$

$= 0.45$

coin B = $0.0009765625 / (0.000796824 + 0.0009765625)$

$= 0.55$

Estimating Likely no of head & tails for coin A & B for First Round.

for coin A = $0.45 * 5 = 2.2$ H
$= 0.45 \times 5 = 2.2$ T

for coin B = $0.55 \times 5 = 2.8$ H
$= 0.55 \times 5 = 2.8$ T

First Round

Consider Round 2 (9H, 1T)

Consider Round 2 (9H, 1T)

Likelihood of A = $0.6^9(1-0.6)^{10-9}$ = 0.004031078 4

likelihood of B = $0.5^9(1-0.5)^{10-9}$ = 0.000976562 5

Normalize A & B

P(A) = 0.80 ✓ A/A+B

P(B) = 0.20 ✓ B/A+B.

Estimate likely no of Head & Tail for Coin A & B in Round 2

No of H for Coin A = 0.80 × 9 = 7.2 H

T " " A = 0.8 × 1 = 0.8 T.

No of H for Coin B = 0.2 × 9 = 1.8 H ✓

T " B = 0.2 * 1 = 0.2 T.

After 5 Rounds

	Coin A	Coin B
Round 1	2.2H, 2.2 T	2.8 H , 2.8 T.
Round 2	7.2H, 0.8T	1.8 H , 0.2 T.
Round 3	5.9 H 1.5 T	2.1 H 0.5 T
Round 4	1.4 H 2.1 T	2.6 H 3.9 T.
Roun 1 5	4.5 H 1.9 T	2.5 H 1.1 T
Total	21.3 H 8.6 T	11.7 H 8.4 T.

Probability of getting Head

$\theta_A = \frac{21.3}{(21.3+8.6)} = 0.71$ ✓

$\theta_B = \frac{11.7}{(11.7+8.4)} = 0.58$.

After 1 step of E & M $\theta_A = 0.71$, $\theta_B = 0.58$ ✓

Similar.

Repeat until New Value of θ_A & θ_B is very Similar to Earlier.

In this Example

The Value Converge to

$\theta_A = 0.8$

$\theta_B = 0.52$

Part I Answered Q Percentage of Head Coin A yields = 80%.

B yields = 52%

Now Consider the Result →

$\theta_A = 0.8$

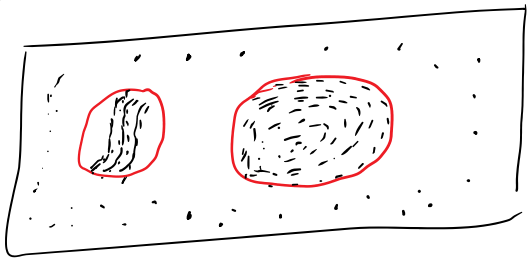
$\theta_B = 0.52$

	H	T	Which coin?
Round 1	5	5	$P(H) = \frac{5}{10} = 0.5$ = Coin B
Round 2	9	1	$P(H) = \frac{9}{10} = 0.9$ = Coin A
Round 3	8	2	$P(H) = \frac{8}{10} = 0.8$ Coin A
Round 4	4	6	$P(H) = \frac{4}{10} = 0.4$ Coin B
Round 5	7	3	$P(H) = \frac{7}{10} = 0.7$ Coin A

Soln to Part 2

DBSCAN (Density Based Spatial Clustering of Application With Noise)

* In Dense Region there is possibility of cluster than in Sparse Region
* Density based approach is better if the cluster is of arbitrary shape.

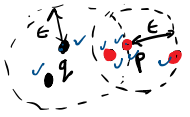


Key Features

To understand DBSCAN Algo we need to understand.

ϵ Maximum radius of neighbourhood.

MinPts: Minimum No of points in an ϵ neighbourhood of that point.



ϵ neighbourhood of q = {4}

ϵ ——— p = {3}

Density of q is high

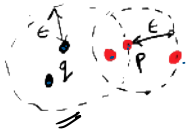
Density of p is low.

Terminologies → given ϵ and Minpts then categorize the objects into 3 group

(1) Core Point → if for a point the neighbourhood has the points ≥ Minpoint. then it is Core point.

i.e. Minpts = 4

q is Core point



No of points in neighbourhood of q = 4 →

So q is Core point

② Border Point → If for a point the neighbourhood has points < Minpt but it should be neighbour of core point

Minpt = 4

No of points in neighbourhood of p = 3 < Minpt (4)

& p is neighbour of q (core point)

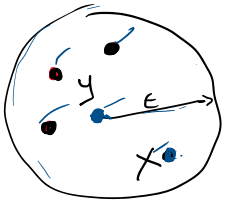
So 'p' is border point

③ Outlier → If a point is neither a core point nor a border point, it is called outlier.

"It is outlier"

* Directly Density Reachable → A point X is directly density reachable from point Y wrt epsilon & Minpt if

1. X belong to neighbourhood of Y (dist(X,Y) <= ε)
2. Y is core Point.



Minpt = 4

① as Y a core point →

no of point in neighbourhood of Y = 5 > Minpt

∴ Y is core point

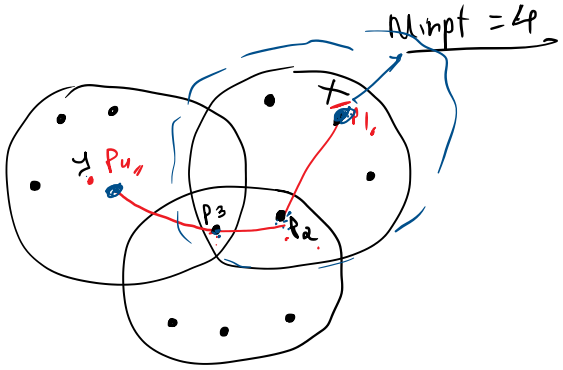
② X belong to neighbourhood of Y.

∴ X is Directly Density Reachable from Y.

② Density Reachable →

A point X is density reachable from Point Y wrt ε & Minpt if there is a chain of points $P_1, P_2 \dots P_n$ and $P_1 = X$

& $P_n = y$ such that P_{i+1} is directly density reachable from P_i



Here x & y are core points

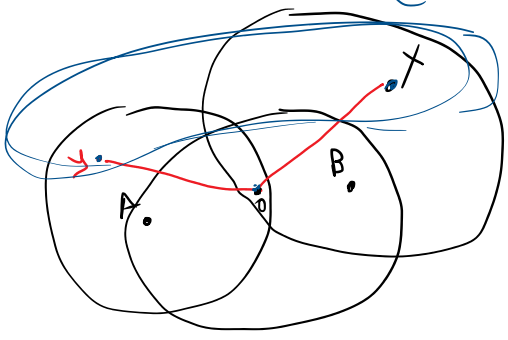
Here P_2 is directly Density Reachable from x

P_3 is directly Density Reachable from P_2

y is directly Density Reachable from P_3

∴ x is Density Reachable from y.

Density Connected → A point x is density Connected from point y wrt ϵ and Minpt if there exists a point O such that both x & y are density Reachable from O wrt ϵ & Minpt



y & O are direct Density Reachable

x & O are direct Density Reach

Thus x & y are density Connected

Direct → within ϵ → Direct Density Reachable.

Direct → Via Some chain → = = =

2 points are Density Reachable through Common pt → Density Connected

Algo

1. Arbitrary Select a Point P
2. Retrieve all points density Reachable from P wrt ϵ & Min Point
3. If P is Core Point then Cluster is formed.
4. If p is border point then DBSCAN visits next point of Dataset
5. Continue the process untill all the points are processed.

14 December 2023 15:33

Q1. a) Elaborate Bagging Ensemble Learning.