ID3 Decision Tree

Unsupervised Learning -

Clustering Algorithm -

Clustering
- Hierarchical
  - Aggmerative Clustering (Bottom up)
  - Divisive Clustering (Top down).
- Partitioning
  - k-means algo
  - k-medoids algo.

Clustering - Process of dividing data points in groups.

K-Means Algorithm -

k - no. of clusters

S - set of clusters, $S = \langle S_1, S_2, \ldots, S_k \rangle$

μ_i = Centre of cluster i.

minimize sum of squares error.

$$\sum_{i=1}^{k} \sum_{x \in i} ||x_i - \mu_i||^2 .$$ (Within cluster sum of square error).

Given k, given set of x (objects).

(1) Identify randomly k data points as initial clusters.

(2) Assign each data point to closest cluster centre.

... repetition we get same ...
... set.

Date ......

(3) Recompute the center.

(4) Repeat (2) till convergence criteria*

Measure of similarity -

(1) Euclidean distance

$$\sqrt{\sum_{s=1}^{n}(x_{is} - x_{js})^2}$$

$x_{js} \rightarrow$ centre.

$x_{is} \rightarrow$ any point in cluster.

(2) Manhatten distance.

$$\sum_{s=1}^{n}|x_{is} - x_{js}|$$

Minkowski family -

$$d(x_i, x_j) = \left(\sum_{s=1}^{n}|x_{is} - x_{js}|^p\right)^{1/p}.$$

p = 1 ; Manhatten distance

p = 2 ; Euclidean distance.

Example - (2,3) (5,6) (8,7) (1,4) (2,2) (6,7) (3,4) (8,6).

No. of clusters = 2.

$S_1 = (2,3)$
$S_2 = (5,6)$ } → assume 2 centres randomly.

Page No.

Date __/__/__

X	Y	S_1: $\sqrt{(x-2)^2+(y-3)^2}$	S_2: $\sqrt{(x-5)^2+(y-6)^2}$	Allocati
2	3	0	$3\sqrt{2}$	S_1
5	6	$3\sqrt{2}$	0	S_2
8	7	$\sqrt{52}$	$\sqrt{10}$.	S_2
1	4	$\sqrt{2}$	$\sqrt{20}$	S_1
2	2	1	5	S_1
6	7	$4\sqrt{2}$	$\sqrt{2}$	S_2
3	4	$\sqrt{2}$	$\sqrt{8}$	S_1
8	6	$\sqrt{45}$	3.	S_2

S_1; Cluster 1 – (2,3) (1,4) (2,2) (3,4).

S_2; Cluster 2 – (5,6) (8,7) (6,7) (8,6).

For S_1, (2, 1, 2, 3) $\Rightarrow$ $mean_x = 2$.

(3, 4, 2, 4) $\Rightarrow$ $mean_y = 13/4$.

(2, 3.25). (New center).

For S_2, $\frac{5+8+6+8}{4}$, $\frac{6+7+7+6}{4}$

$= \frac{27}{4}, \frac{26}{4}$

$= (6.75, 6.5)$ (New center).

X	Y	$\sqrt{(x-2)^2+(y-3.25)^2}$	$\sqrt{(x-6.75)^2+(y-6.5)^2}$	Allo.
2	3	1/4	$\sqrt{485}/4$	S_1
5	6	$\sqrt{265}/4$	$\sqrt{29}/4$	S_2
8.	7	$3\sqrt{89}/4$	$\sqrt{53}/4$	S_2
1	4	5/4	$\sqrt{541}/4$	S_1
2	2	5/4	$\sqrt{613}/4$	S_1
6	7	$\sqrt{481}/4$	$\sqrt{5}/4$	S_2
3	4	5/4	$\sqrt{269}/4$	S_1
8	6.	$\sqrt{697}/4$.	$\sqrt{53}/4$	S_2

no. of iterations — no. of clusters — no. of instances — no. of data points

Date ___/___/___

Reallocation S_1 (2, 3.25), S_2 (6.75, 6.5) $\rightarrow$ preserve it.

Q) (2, 3, 4, 10, 11, 12, 20, 25, 30)

S_1 (at 4), S_2 (at 12)

Assume

$\Rightarrow$ $S_1 = \{2, 3, 4\}$

$S_2 = \{10, 11, 12, 20, 25, 30\}$

Disadvantages-

1) Will need k's initially

2) hard assignment of clusters.

Mean $S_1 = 3$

$S_2 = 18$.

New, centre (3, 18)

$S_1 = \{2, 3, 4, 10\}$

$S_2 = \{11, 12, 20, 25, 30\}$

Mean $S_1 = 19/4 = 4.75$, $S_2 = 98/5 = 19.6$ $\rightarrow$ New centre.

$S_1 = \{2, 3, 4, 10, 11, 12\}$

$S_2 = \{\cancel{11}, \cancel{12}, 20, 25, 30\}$

Mean $= S_1 = \cancel{4.75}\ 7$, $S_2 = \cancel{19.6}\ 25$ $\rightarrow$ new centre.

$S_1 = \{2, 3, 4, 10, 11, 12\}$

$S_2 = \{20, 25, 30\}$.

So, centre = (7, 25).

Page No.

K-mediods Algorithm -

(1) Select k centres/mediods (any k data points from given values only)

(2) Distance betn rest of the points and mediods.
↓
Manhatten.

(3) Calculate cost.

(4) Choose some other mediods then resign and calculate cost again.

Q5 d1(2,6), d2(3,4), d3(3,8), d4(4,7), d5(6,2), d6(6,4)
d7(7,3), d8(7,4), d9(8,5), d10(7,6). (k=2)

D	x	y	$\lvert x-3\rvert+\lvert y-4\rvert$	$\lvert x-7\rvert+\lvert y-4\rvert$	Allo
d1	2	6	3	7	S1
d2	3	4	-	-	
d3	3	8	4	8	S1
d4	4	7	4	6	S1
d5	6	2	5	3	S2
d6	6	4	3	1	S2
d7	7	3	5	1	S2
d8	7	4	~~4~~	-	
d9	8	5	6	1	S2
d10	7	6	6	2	S2

For cost add accepted Manhatten distance values
Cost = 3+4+4+3+1+1+2+2
=20.

When (3,4) and (7, 3) is taken cost = 22
so perform the same 3-4 times.

Page No.

Date __ / __ / ____

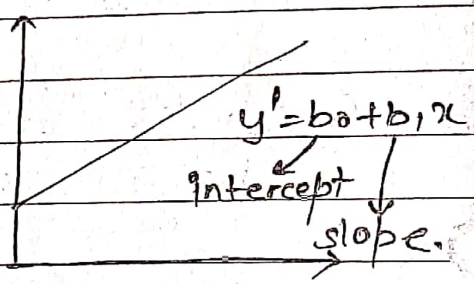
Regression Models

Regression Models.

- Simple
  - ↳ Linear
  - ↳ Non-linear
- Multiple
  - ↳ Linear
  - ↳ Non-linear

Linear Regression-

$$b_1 = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\Sigma(x-\bar{x})^2}.$$



calculate

y'	x	y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})^2$	$(x-\bar{x}).(y-\bar{y})$.
2.8	1	2	-2	-2	4	4
3.4	2	4	-1	0	1	0
4	3	5	0	1	0	0
4.6	4	4	1	0	1	0
5.2	5	5.	2	1	4	2

$\bar{x} = \frac{1+2+3+4+5}{5} = 3$

$\bar{y} = \frac{2+4+5+4+5}{5} = 4$

$b_1 = \frac{6}{10} = 0.6.$

$y = b_0 + 0.6x$

for (3,4)

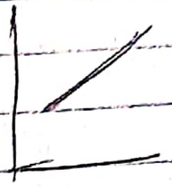
$4 = b_0 + 0.6(3)$

$b_0 = 4 - 1.8$

$= 2.2.$

$\Rightarrow y = 2.2 + 0.6x.$

Linear Regression (Supervised)
↳ continuous data



$\hat{y} = b_0 + b_1 x$

$$\boxed{b_1 = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\Sigma(x-\bar{x})^2}}$$ (Least square method)

The best line of fit always pass through mean.

standard error of estimate $= \sqrt{\dfrac{\Sigma(\hat{y}-y)^2}{n-2}}$ (SEE should be les than 1).

x	y	$x-\bar{x}$	$(x-\bar{x})^2$	$y-\bar{y}$	$(x-\bar{x})(y-\bar{y})$	$\hat{y}$	$\hat{y}-y$	$(\hat{y}-y)^2$
1	2	-2	4	-2	4	2.8	0.8	0.64
2	4	-1	1	0	0	3.4	0.6	0.36
3	5	0	0	1	0	4	-1	1
4	4	1	1	0	0	4.6	0.6	0.36
5	5	2	4	1	2	5.2	0.2	0.04
3	4							2.40

$b_1 = \frac{6}{10} = 0.6$

$\Rightarrow SEE = \sqrt{\frac{2.4}{5-2}}$

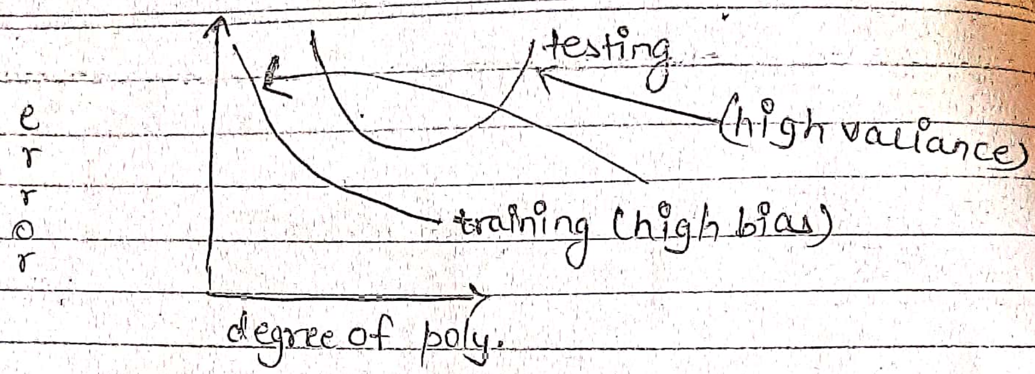
$\hat{y} = b_0 + 0.6x$
$4 = b_0 + 0.6(3)$
$b_0 = 2.2$
$\rightarrow \hat{y} = 2.2 + (0.6)x$

Bias and Variance -

Bias → Restrictions
→ Preferences.
↳ high training and testing errors.

Variance → due to dataset.
↳ high testing errors.

#



Underfitting (high bias)

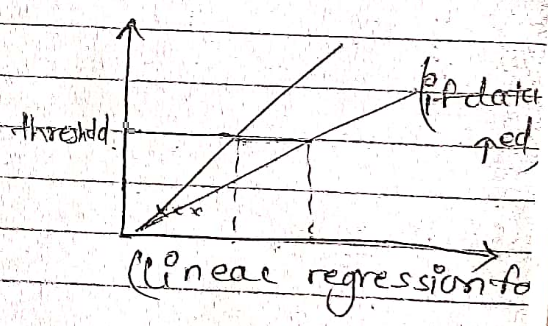
Overfitting (high variance)

Cross validation -

Logistic Regression (Method of classification) (Supervised).
(Discrete data).

↳ Data should be linearly separable

↳ use a logit/ logistic/ sigmoid f^n



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

$$P(class) = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2)}}$$ (Prediction).

$b = b + \alpha * (y - prediction) * prediction * (1 - prediction) * x.$

$\alpha \rightarrow$ learning rate (0.1 to 0.3)

Epoch - (until point of convergence).

Classification Method -

=> **Naive Bayes Classifier** - (Probabilistic ML algo).

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

A → class ; B → feature.
→ hypothesis ; → evidence

* $P(A/B_1, B_2) = \frac{P(B_1/A) \cdot P(B_2/A) \cdot P(A)}{P(B_1)\, P(B_2)}$

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

=> $P(Y/x_1, x_2, x_3, \dots, x_n) \propto P(Y) \prod_{i=1}^{n} P(x_i/Y)$.

Q- No. of fruits = 1000

Type	Lengthy	Sweet	Yellow	Total
Banana	400	350	450	500
Orange	0	150	300	300
Other	100	150	50	200
				1000.

Predict prob. of each class.

$P(\text{Banana}/\text{lengthy}, \text{sweet}, \text{yellow}) = \frac{P(\text{lengthy}/\text{Banana}) \cdot P(\text{sweet}/\text{Banana}) \cdot P(\text{yellow}/\text{Banana}) \cdot P(\text{Banana})}{P(\text{lengthy}) \cdot P(\text{sweet}) \cdot P(\text{yellow})}$

$$= \frac{\frac{400}{500} \times \frac{350}{500} \times \frac{450}{500} \times \frac{500}{1000}}{\frac{500}{1000} \times \frac{650}{1000} \times \frac{800}{1000}}$$

$\frac{400 \times 350 \times 450}{250 \times 325 \times 800} = 0.9692$

Class overlap.

Instance Space Learning (Lazy Learning)

K-Nearest Neighbour - (used for classification and regress

Training Phase → we just store the database
Testing Phase → prediction.

Prediction Phase (get test instance)
→ closest example.

Classification :→ most frequent.
(discrete data)

Regression :→ average.
(continuous data)

Weighted KNN algorithm-

→ Weights
→ how to measure closeness.
→ how to reache to the nearest one quickly.

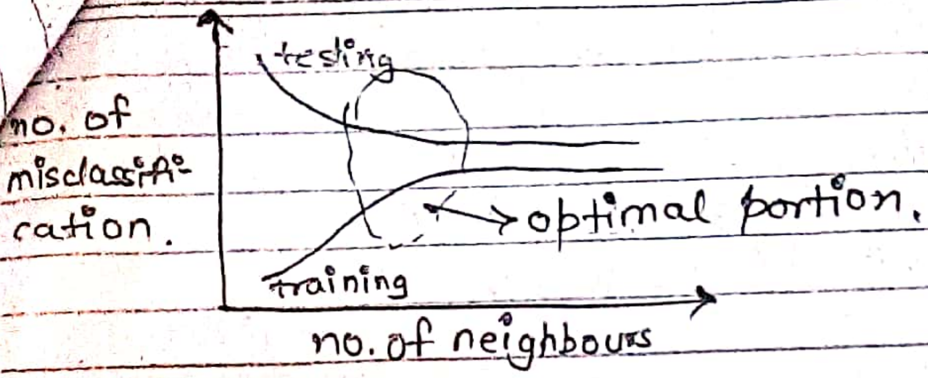
Standard distance fn - Euclidean Distance.

$$Dis\_Eu(x_i, x_j) = \sqrt{\sum_{i=1}^{n} (x_i - x_j)^2}$$

Scale of attribute } → should be same in prediction.
Variance in attribute }

small k → fine structure of problem.
large k → classes overlap.

generally k=(1-7) preferred.



Weighted difference function -

$$D(x_i, x_j) = \sqrt{\sum_{i=1}^{n} W_m (x_{im} - x_{jm})^2}$$

Example-

AIML	DAA	Result	D
4	3	Fail	$\sqrt{29}$
6	7	Pass	1 ✓
7	8	Pass	1 ✓
5	5	Fail	$\sqrt{10}$
8	8	Pass	2 ✓

X → AI=6 DAA=8.

feature selection

" Extn

↳ Dimensionality Redn

↳ PCA (unsupervised)

↳ LDA (supervised).

" Redn.

* PCA = Unsupervised (Principal Component Analysis)
* LDA = Supervised (Linear Discriminant Analysis)

* Difference b/w PCA and LDA.

LDA

→ within class scatter
→ scatter b/w classes.

S_1 = Scatter withing class 1
S_2 = " " " 2.

$$S_1 = \sum_{\forall x_i \in w_1} (x_i - \mu_1)(x_i - \mu_1)^t.$$

$S_W = S_1 + S_2$

For ex –

$W_1 = \begin{matrix} 1 & 3 & 4 & 5 & 7 \\ 2 & 5 & 3 & 6 & 5 \end{matrix} \Rightarrow \mu_1 = \begin{matrix} 4 \\ 4.2 \end{matrix}$

$W_2 = \begin{matrix} 6 & 9 & 10 & 12 & 13 \\ 2 & 4 & 1 & 13 & 6 \end{matrix} \Rightarrow \mu_2 = \begin{matrix} 10 \\ 3.2 \end{matrix}$

$\Rightarrow x_i - \mu_1 = \begin{matrix} -3 & -1 & 0 & 1 & 3 \\ -2.2 & 0.8 & -1.2 & 1.8 & 0.8 \end{matrix}$

$x_i - \mu_2 = \begin{matrix} -4 & -1 & 0 & 2 & 3 \\ -1.2 & 0.8 & -2.2 & -0.2 & 2.8 \end{matrix}$

$$\sum (x_i-\mu_1)\cdot(x_i-\mu_1)^t$$

$$S_1 = \forall x_i-\mu_1$$

$$\Rightarrow S_1 = \begin{pmatrix}-3\\-2.2\end{pmatrix}\begin{pmatrix}-3 & -2.2\end{pmatrix} + \begin{pmatrix}-1\\0.8\end{pmatrix}\begin{pmatrix}-1 & 0.8\end{pmatrix} + \begin{pmatrix}0\\-1.2\end{pmatrix}\begin{pmatrix}0 & -1.2\end{pmatrix}$$

$$+ \begin{pmatrix}1\\1.8\end{pmatrix}\begin{pmatrix}1 & 1.8\end{pmatrix} + \begin{pmatrix}3\\0.8\end{pmatrix}\begin{pmatrix}3 & 0.8\end{pmatrix}.$$

$$= \begin{pmatrix}9 & 6.6\\6.6 & 4.84\end{pmatrix} + \begin{pmatrix}1 & -0.8\\-0.8 & 0.64\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 1.44\end{pmatrix} +$$

$$\begin{pmatrix}1 & 1.8\\1.8 & 3.24\end{pmatrix} + \begin{pmatrix}9 & 2.4\\2.4 & 0.64\end{pmatrix}$$

$$= \begin{pmatrix}20 & 10.8\\10.8 & 10.8\end{pmatrix}$$

$$S_2 = \begin{pmatrix}-4\\-1.2\end{pmatrix}\begin{pmatrix}-4 & -1.2\end{pmatrix} + \begin{pmatrix}-1\\0.8\end{pmatrix}\begin{pmatrix}-1 & 0.8\end{pmatrix} + \begin{pmatrix}0\\-2.2\end{pmatrix}\begin{pmatrix}0 & -2.2\end{pmatrix}$$

$$\begin{pmatrix}2\\-0.2\end{pmatrix}\begin{pmatrix}2 & -0.2\end{pmatrix} + \begin{pmatrix}3\\2.8\end{pmatrix}\begin{pmatrix}3 & 2.8\end{pmatrix}$$

$$= \begin{pmatrix}16 & 4.8\\4.8 & 1.44\end{pmatrix} + \begin{pmatrix}1 & -0.8\\-0.8 & 0.64\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 4.84\end{pmatrix}$$

$$+ \begin{pmatrix}4 & -0.4\\-0.4 & 0.04\end{pmatrix} + \begin{pmatrix}9 & 8.4\\8.4 & 7.84\end{pmatrix}$$

$$= \begin{pmatrix}30 & 12\\12 & 14.8\end{pmatrix}$$

$$S = \begin{pmatrix}50 & 22\\22 & 25.6\end{pmatrix}$$

$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t$. Or $\vec{e} = S_\omega^{-1}(\mu_1 - \mu_2)$.

$S_\omega^{-1} S_B \omega = \lambda \omega$.

+2)

$$S_W = \begin{pmatrix} 50 & 22 \\ 22 & 25.6 \end{pmatrix}$$

$$|S_W| = 50 \times 25.6 - 22 \times 22$$
$$= 796.$$

$$adj(S_W) = \begin{pmatrix} 25.6 & -22 \\ -22 & 50 \end{pmatrix}^T$$

$$= \begin{pmatrix} 25.6 & -22 \\ -22 & 50 \end{pmatrix}$$

-2.2)

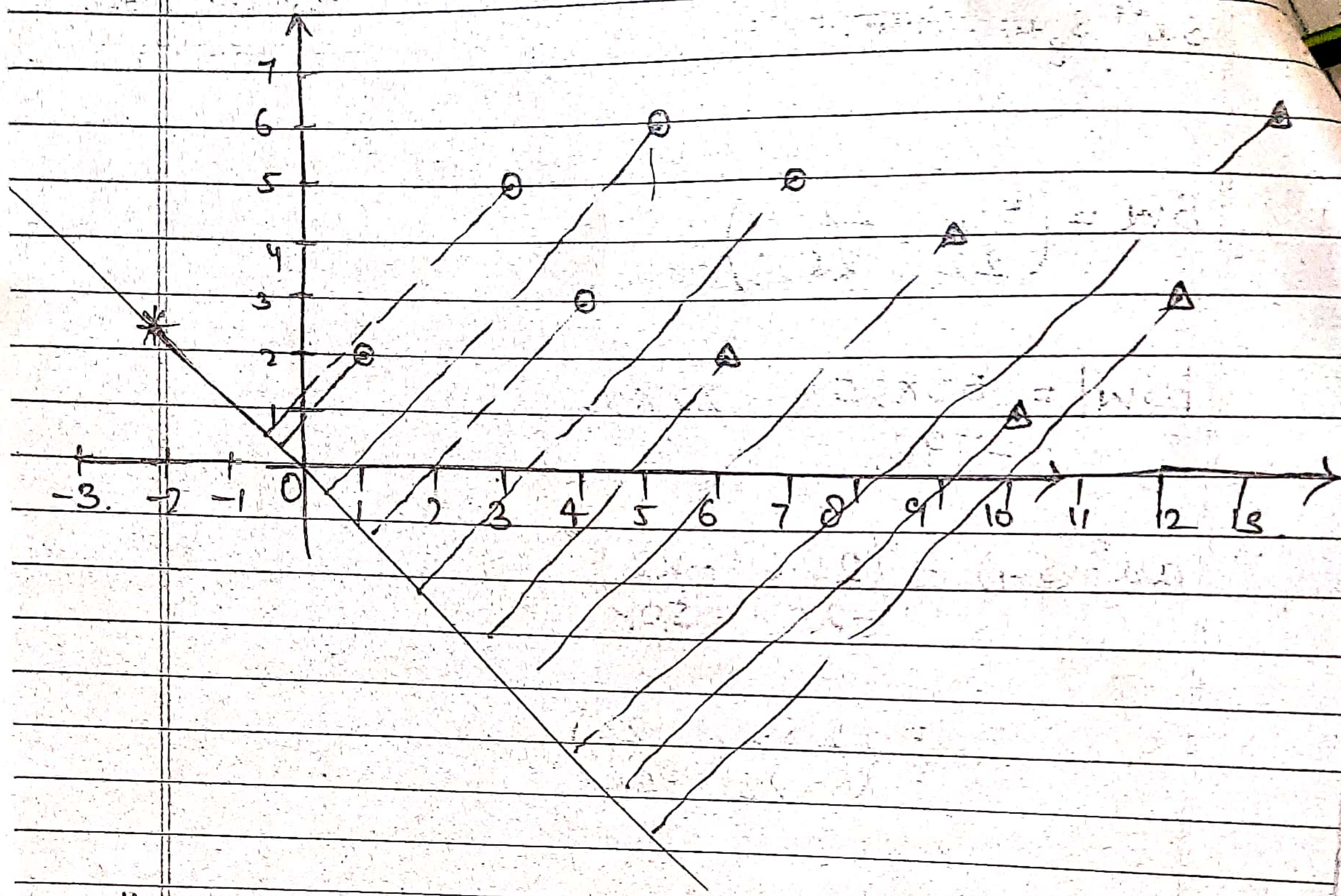
$$\Rightarrow S_W^{-1} = \frac{adj(S_W)}{|S_W|} = \frac{1}{796}\begin{pmatrix} 25.6 & -22 \\ -22 & 50 \end{pmatrix} = \begin{pmatrix} 0.032 & -0.03 \\ -0.03 & 0.06 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} 4 \\ 4.2 \end{pmatrix} \qquad \mu_2 = \begin{pmatrix} 10 \\ 3.2 \end{pmatrix}$$

$$\mu_1 - \mu_2 = \begin{pmatrix} -6 \\ 1 \end{pmatrix}$$

$$\vec{e} = \begin{pmatrix} 0.032 & -0.03 \\ -0.03 & 0.06 \end{pmatrix}\begin{pmatrix} -6 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} -0.222 \\ 0.24 \end{pmatrix} \times 10 \rightarrow \begin{pmatrix} -2.2 \\ 2.4 \end{pmatrix}$$

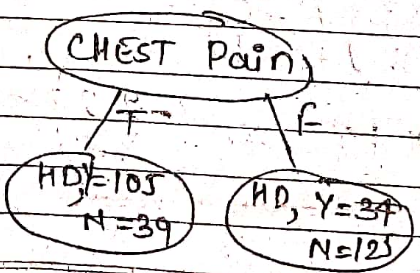


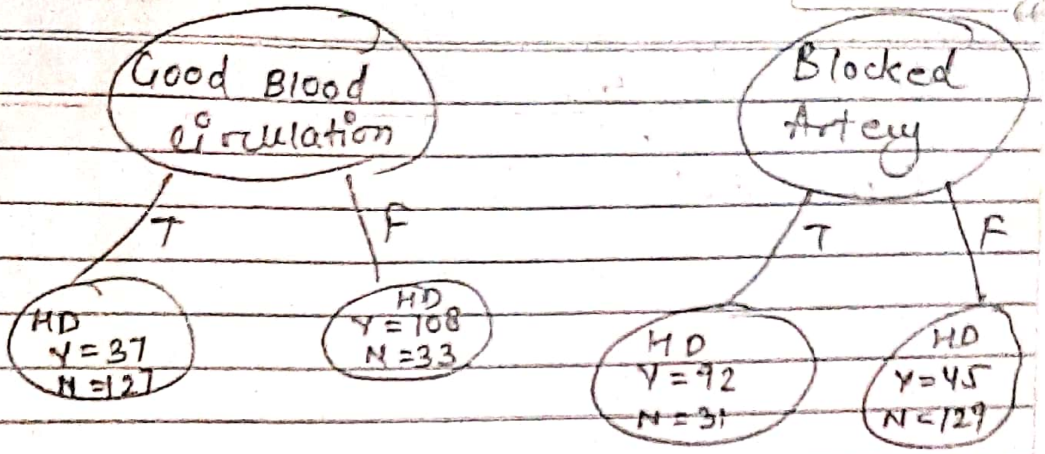
CART (Classification And Regression Tree) Model-

Gini index = 1-(Probability of YES)^2 - (Probability of NO)^2

For ex_ HD → Heart Disease.

Chest Pain	Good Blood Circulation	Blocked Arteries	HD
Y	Y	Y	Y
Y	Y	Y	N
Y	N	N	Y





For chest pain,

$$G.I. = 1 - (P(Y))^2 - (P(N))^2$$

(LC) Left child = $GI = 1 - \left(\frac{105}{105+39}\right)^2 - \left(\frac{39}{105+39}\right)^2$

$= 0.395.$

(RC) Right child, $GI = 1 - \left(\frac{34}{34+125}\right)^2 - \left(\frac{125}{34+125}\right)^2$

$= 0.336.$

$\Rightarrow$ Total Gini Impurity (TGI) $= \left(\frac{144}{144+159}\right)(0.395) + \left(\frac{159}{144+159}\right)(0.336)$

$= 0.364$ (for chestpain).

For blood circulation,

$L.C. = 1 - \left(\frac{37}{37+127}\right)^2 - \left(\frac{127}{37+127}\right)^2 =$

$T.G.I = 0.36.$

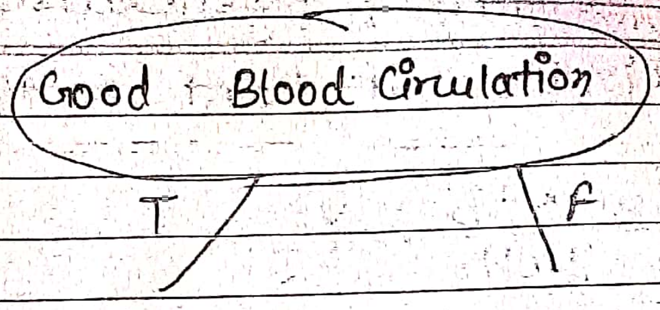
for blocked artery -

TGI = 0.381

⇒ G.I. CP = 0.364

BC = 0.36 ⟵⟶ minimum ⇒ root

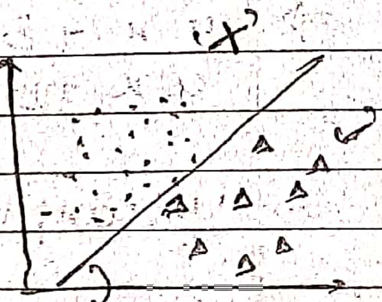
B.A = 0.381



Support Vector Machine-

→ Supervised Learning Algorithm.

→ Classifier



→ +ve side hyperplane:

⇒ $g(x) = \omega^{t}x + b > 0$

→ -ve side hyperplane:

⇒ $g(x) = \omega^{t}x + b < 0$

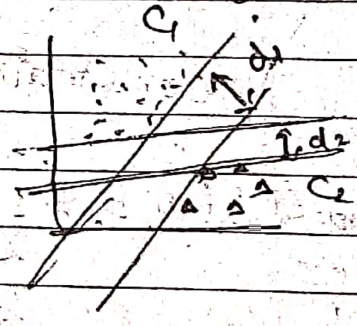
$= 0$.

in mid posn b/w 2 classes.

→ find dirn in such a way that gives maxm distance b/w classes.

$d_1 > d_2$

so d_1 is suggested.



Suppose $x \in C_1$ (unknown)

for C_1 $\quad \omega^{T}x + b > 0$.

$y_i = \pm 1$ (for 2 classes)

for C_1 $y_i = 1$

$\boxed{y_i(\omega^{T}x_i + b) > 0}$ (always)

$+ \quad +$

$- \quad -$

$y_i(\omega x_i + b) > \gamma$ (generalised)

Distance of point x from hyperplane

$$\frac{\omega \cdot x + b}{||\omega||} \geq \gamma$$

vector ω, $\vec{\omega}$ = tells orientation of hyperplane.

$\Rightarrow \omega \cdot x + b \geq \gamma ||\omega||$

By scaling, $\gamma ||\omega|| = 1$

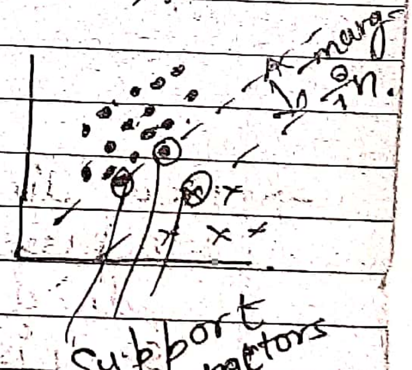
$\Rightarrow \omega x + b \geq 1$ (if a point belongs to C_1)

$\omega x + b < 1$ (if a point belongs to C_2).

$\Rightarrow y_i(\omega x_i + b) \geq 1$.

Now, $y_i(\omega x_i + b) = 1$ (if x_i is support vector)

→ we need to maximise the margin.



$\omega = \phi(\omega) = \frac{1}{2} \omega \cdot \omega = \omega^t \omega$.

constraint (used to minimise ω).

Lagrange's eq^n $L(\omega, b) = \frac{1}{2}(\omega, \omega) - \sum \alpha_i [y_i[\omega \cdot x_i + b] - 1]$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i$$

m = no of feature vectors.

$\frac{\partial L}{\partial \omega} = \sum_{i=1}^{m} \alpha_i y_i x_i$ → feature vector.

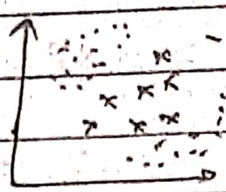
scalar vectors

$$\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \alpha_i \alpha_j y_i y_j x_i x_j$$

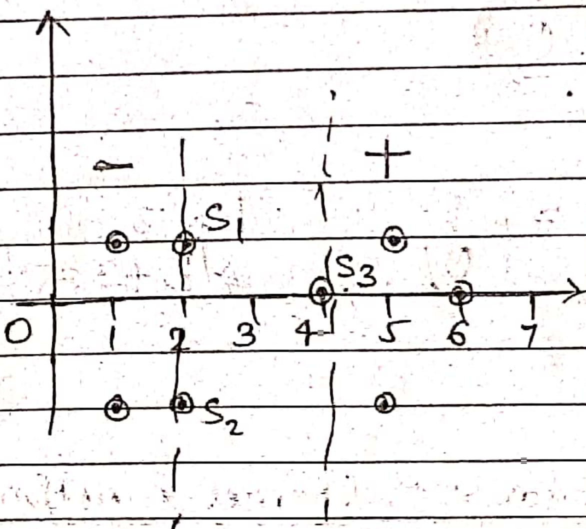
$$\alpha_i \geq 0 \; ; \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

$$\Rightarrow D(z) = \text{sign}\left(\sum_{j=1}^{m} \alpha_j y_j x_j \cdot z + b \right).$$

Assume- Classes are linearly separable.

Not linearly separable ∈ 

For ex- (1,1) (2,1) (1,-1) (2,-1) (5,1)
(5,-1), (6,0) (4,0)



$S_1, S_2, S_3 \Rightarrow$ support vectors

$S_1(2,1)$ $S_2(2,-1)$ $S_3(4,0)$.

$$S_1' = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad S_2' = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \quad S_3' = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

Bias.

$\sum \alpha_i$

$\alpha_1 S_1 S_1 + \alpha_2 S_1' S_2 + \alpha_3 S_1 S_3 = -1$

$\alpha_1' S_2 S_2 + \alpha_2 S_2' S_2 + \alpha_3 S_2 S_3 = -1$

$\alpha_1' S_1 S_3 + \alpha_2 S_2 S_3 + \alpha_3 S_3 S_3 = 1.$

$\Rightarrow \alpha_1 = -3.25$

$\alpha_2 = -3.25$

$\alpha_3 = 3.5.$

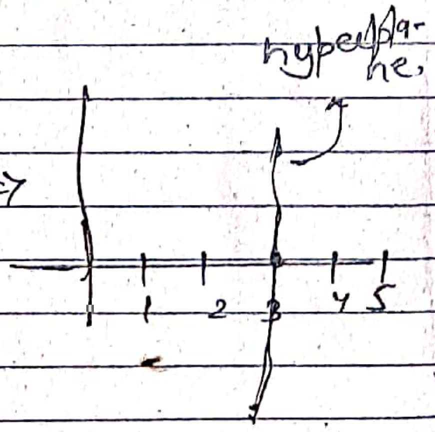
$\bar{\omega} = \sum \alpha_i S_i$

$$= -3.25\begin{pmatrix}2\\1\\1\end{pmatrix} + -3.25\begin{pmatrix}2\\-1\\1\end{pmatrix} + 3.5\begin{pmatrix}4\\0\\1\end{pmatrix}$$

$$\omega = \begin{pmatrix}1\\0\\-3\end{pmatrix}$$

$\Rightarrow \omega = \begin{pmatrix}1\\0\end{pmatrix}$ $b = -3$

$\Rightarrow \boxed{b+3=0}$ $\Rightarrow$



Non-linear SVM (Kernel functions) -

$x \longrightarrow \phi(x)$

Original feature, Transformed feature.

Computational Cost -

SVM $\rightarrow \sum \alpha_i \alpha_j y_i y_j x_i x_j$.

d dimensions $\rightarrow$ $O(d^2)$ (Computational Cost)

$$x_a \qquad\qquad x_b$$
$$\phi(x_a) \qquad\qquad \phi(x_b)$$
$$K(x_a, x_b) = \phi(x_a) \cdot \phi(x_b)$$

$$g(x) = w^T \phi(x) + b.$$

$$= \sum_{i \in SV} \alpha_i \underbrace{\phi(x_i)^T \phi(x)}_{\text{computed as kernel function.}} + b.$$

Kernel fn is defined as fn that corresponds to a dot product of 2 feature vectors in som expanded feature space.

$$\bar{x} = (x_1, x_2).$$

let $K(x_i, x_j) = (1 + x_i \cdot x_j)^2.$

$x_{i1} \quad x_{i2}$

$$= 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$x_{j1} \quad x_{j2}$ $= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i}$

$$\cdot\, [1 \;\; x_{j1} \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x$$

$$= \phi(x_i) \cdot \phi(x_j)$$

Scanned by CamScanner

Date
Page

4. Kernels -

(1) Linear Kernel -

$$K(x_i\ x_j) = x_i . x_j$$

(2) Polynomial of power p -

$$K(x_i\ x_j) = (1 + x_i x_j)^P$$

(3) Gaussian Kernel / Radial Basis fn -

$$K(x_i\ x_j) = \exp\left(- \frac{\|x_i - x_j\|^2}{-2\sigma^2} \right)$$

(4) Sigmoid

$$K(x_i, x_j) = \tanh\left(\beta_0 x_i x_j + \beta_1 \right)$$

⇒ Kernel functions are the functions that satisfy Mercer's condition.

$$\sum_{i,j} k(x_i, x_j)\ c_i c_j \geq 0.$$

Mercer's condition - If you find similarities b/w the points as a matrix and this matrix is symmetric positive semidefinite then kernel function exist.