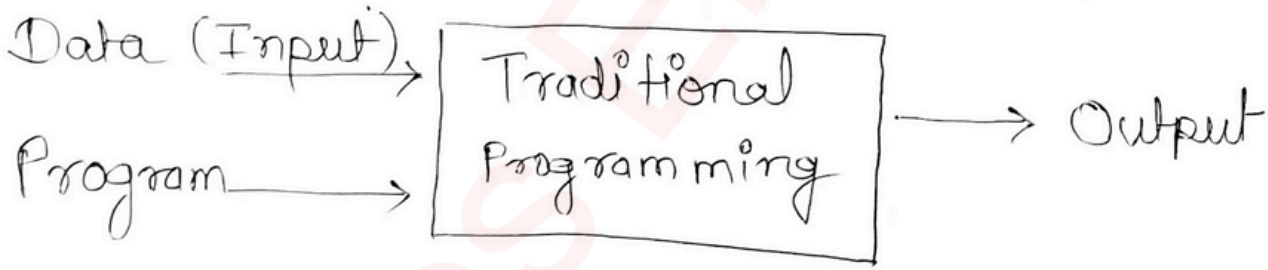
ML

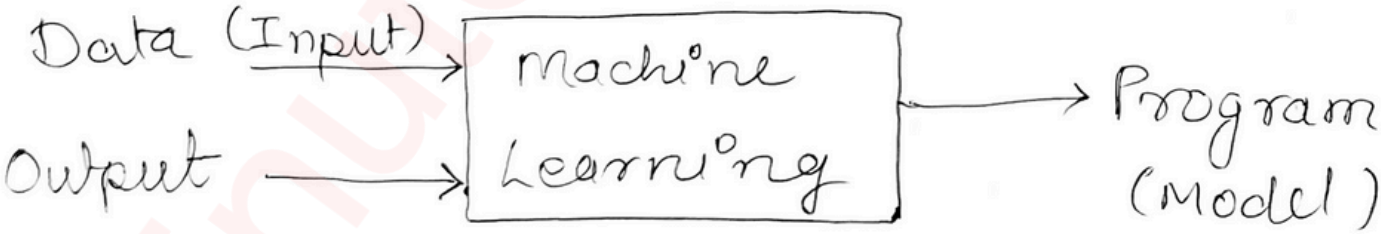
↓ Machine ↓ Learning

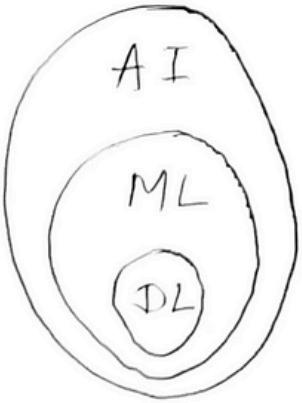
- Capability to learn without being explicitly programmed.

↓

(without being actually programmed)

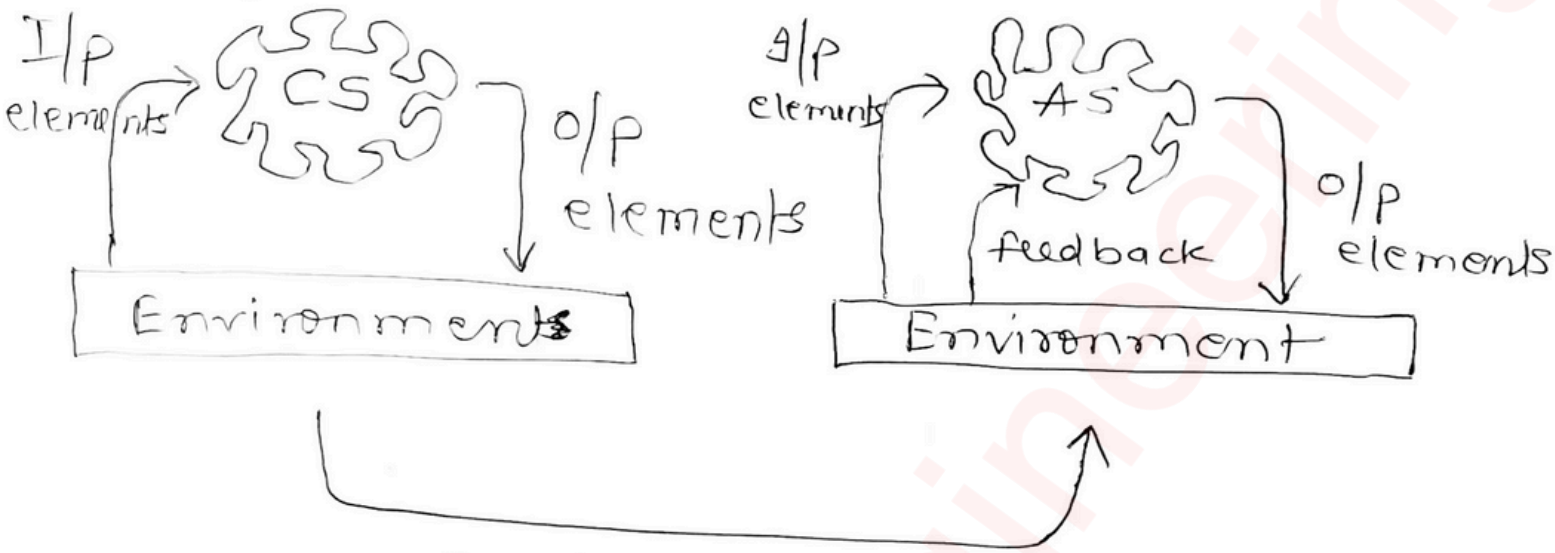






5 Minutes Engineering

- Classic & Adaptive machines



Simple to Smart machine.

- Machine Learning cycle:

1. Problem Understanding
2. Data collection
3. Data preparation
4. Model selection
5. Model building
6. Model Evaluation
7. Model Tuning
8. Deployment
9. Monitoring

5Minutes Engineering

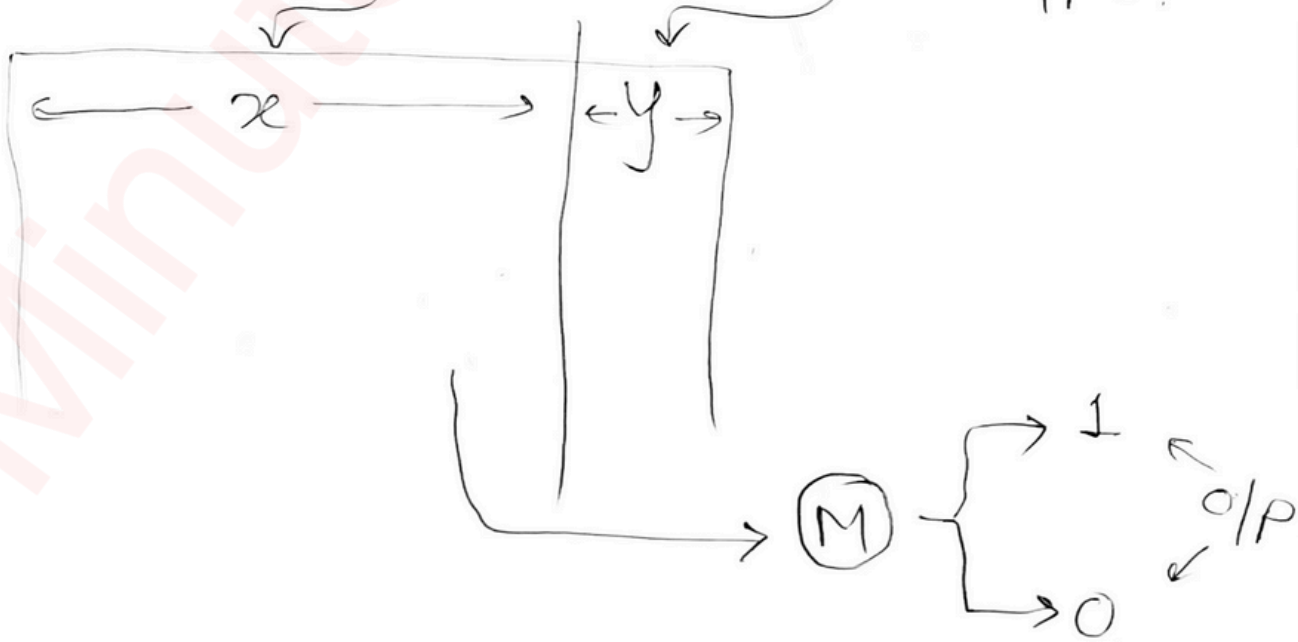
- **Data** :- facts, figs, statistics, observations.
- **Information** : meaningful, useful, relevant data.
- **Labelled data**: I/p + O/p

* **Types of Learning**:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning.

① **Supervised Learning**:

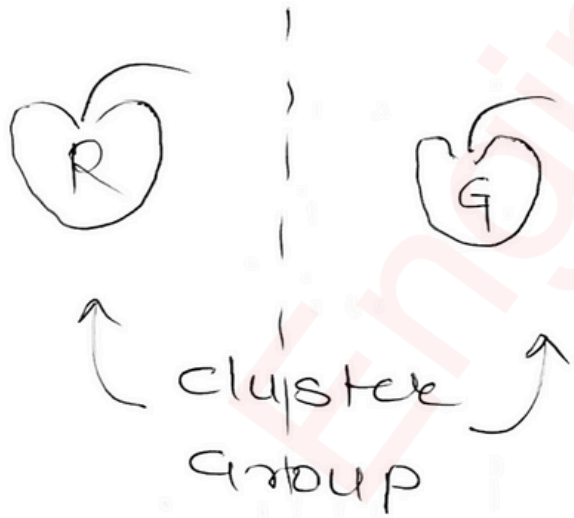
- → I/p + O/p (labelled data).
- → Question & Answer type.



5 Minutes Engineering

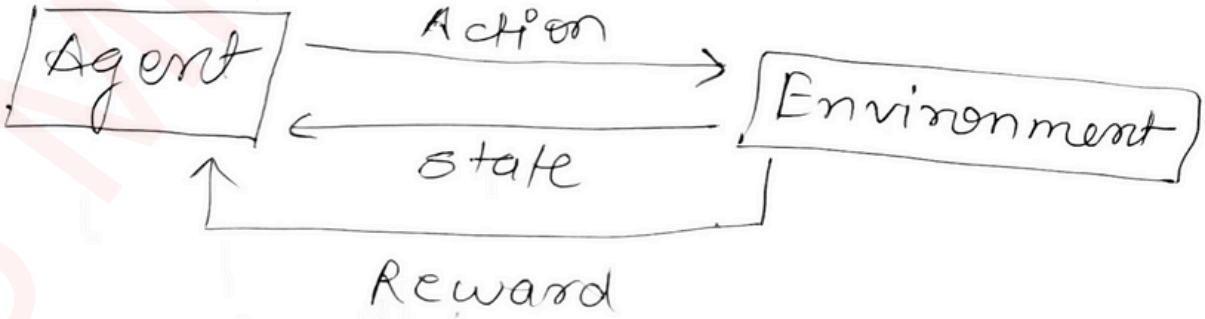
② Unsupervised Learning:

- Only I/p 'NO' output
- Only Question type
- Unlabelled data
- Look out for similarity.



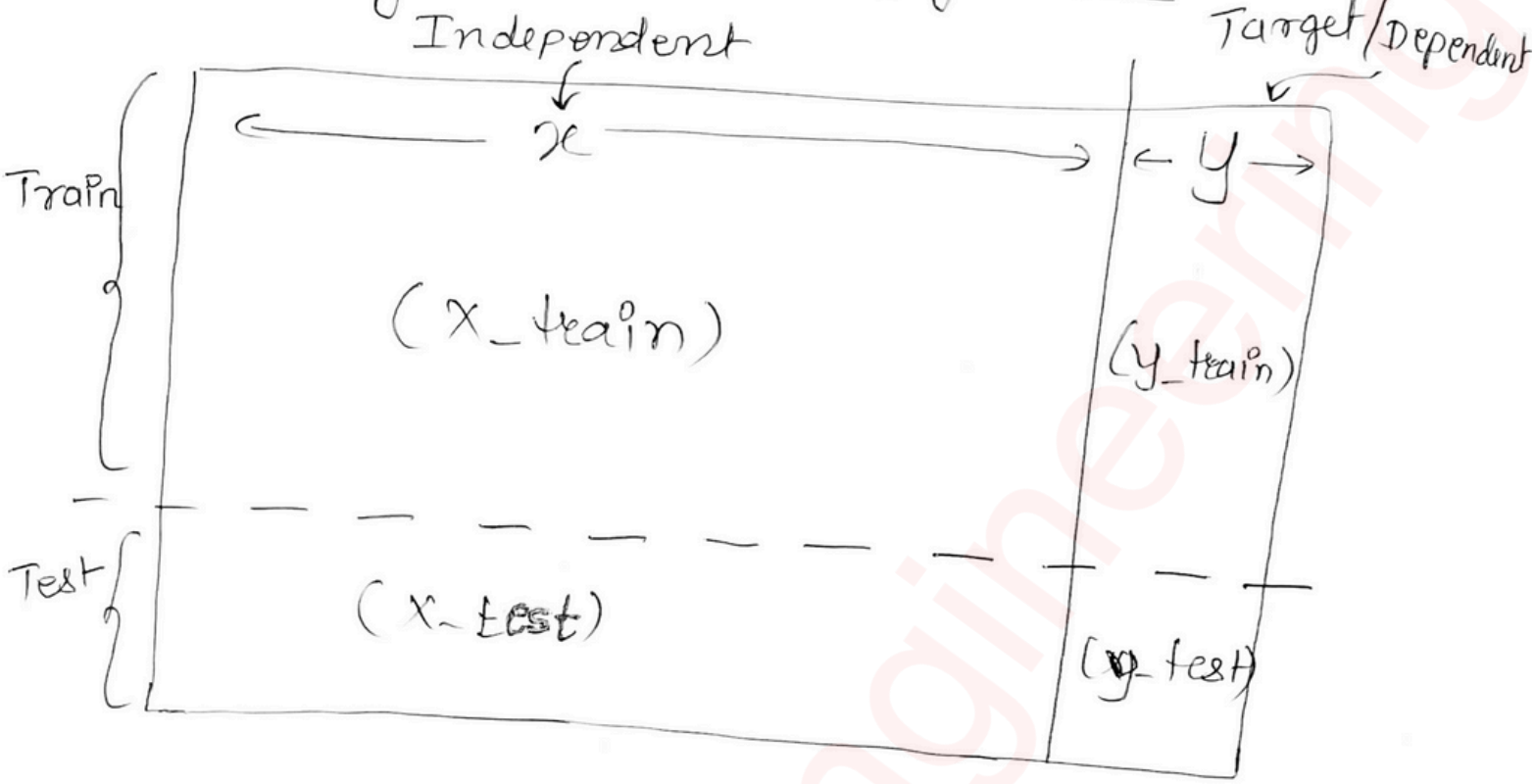
③ Reinforcement Learning:

- Reward / penalty based learng.



5 Minutes Engineering

* Testing & Training Data



(x_test)

$\begin{pmatrix} x_train \\ y_train \end{pmatrix} \longrightarrow$ TM $\longrightarrow$ Trained Model

↓

y_predicted

y_test (actual) $\longleftrightarrow$ y_predicted (predicted)

↑

Error

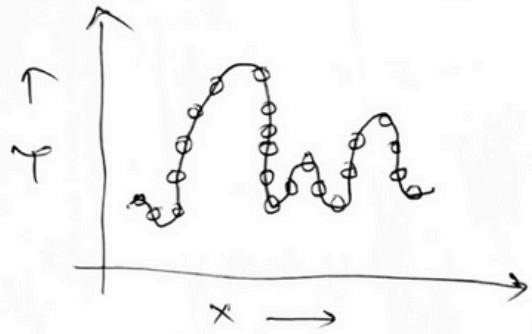
↓

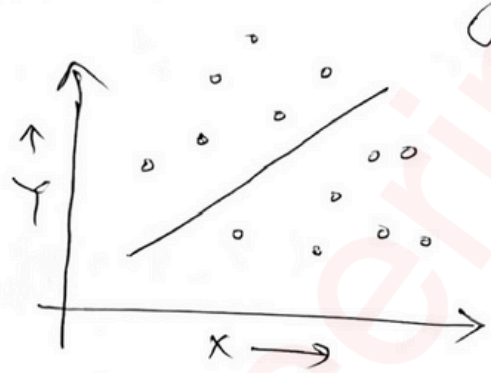
Cost function.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - y_p)^2$$

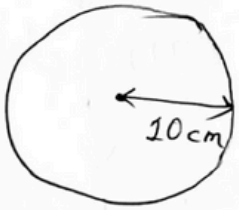
5 Minutes Engineering

- Overfitting & Underfitting





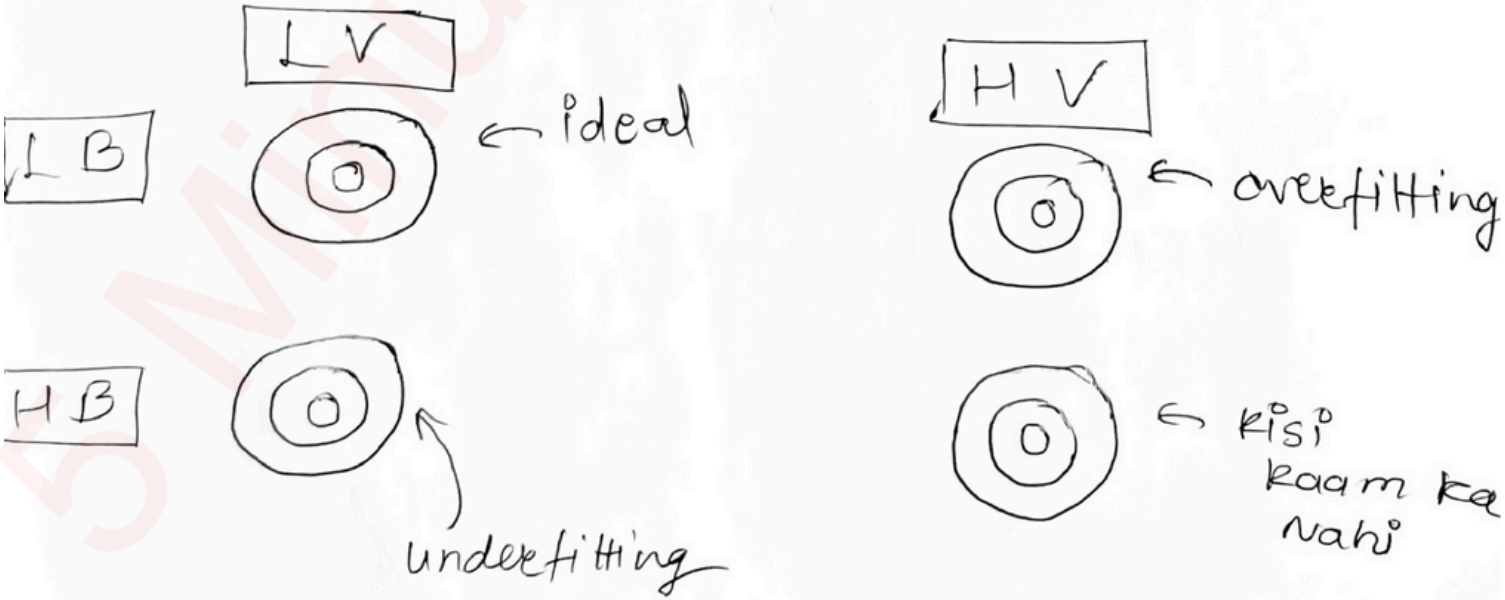
- Sphere
- Play
- Eat
- Radius = 5cm



→ Ball

Sphere

- Bias & variance



5 Minutes Engineering

- **Confusion Matrix:**

		Predicted		
	165	NO	Yes	
Actual	NO	50 [TN]	10 [FP]	60
	Yes	5 [FN]	100 [TP]	105
		55	110	

- **Accuracy** $= \dfrac{TP + TN}{TN + TP + FN + FP} = \dfrac{100+50}{165}$

$= 0.91$

- **Error:** $1 - \text{Accuracy}$

or

$\dfrac{FP + FN}{\text{Total}} = 0.09$

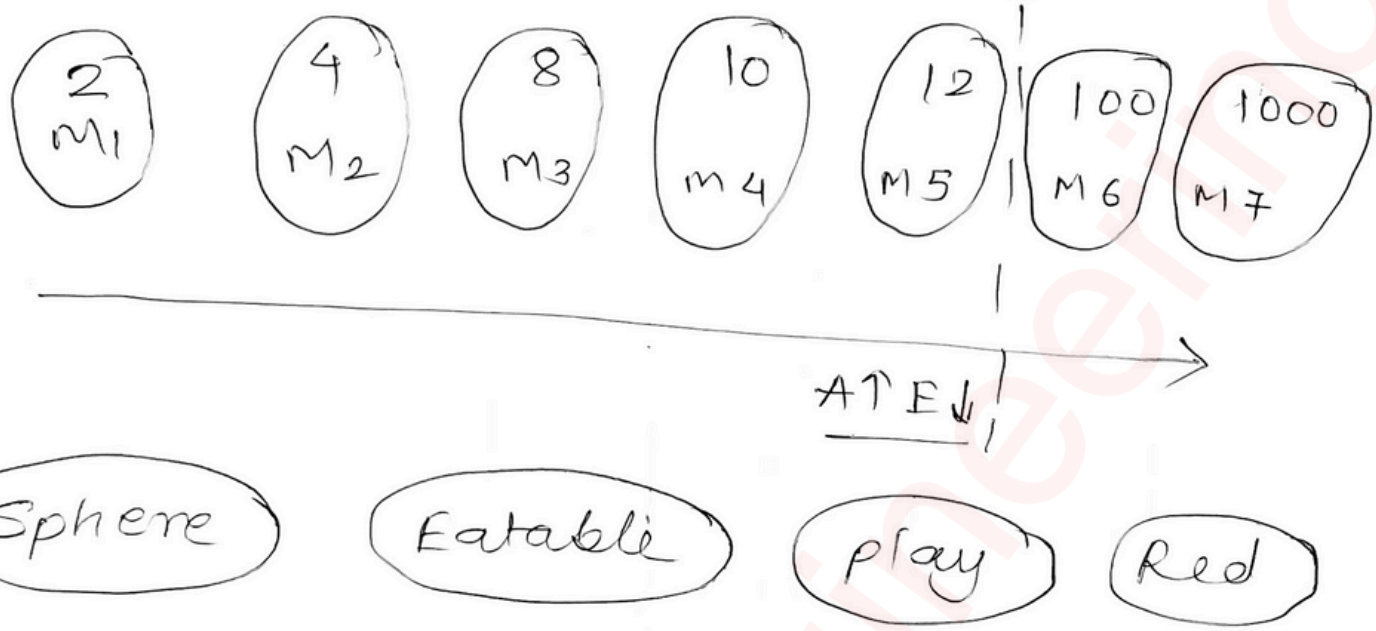
- **Precision:** $\dfrac{TP}{TP + FP} = \dfrac{100}{110} = 0.909$

- **Recall:** $\dfrac{TP}{\text{Actual Yes}} = \dfrac{TP}{TP + FN} = \dfrac{100}{105}$

$= 0.95$

5 Minutes Engineering

- Curse of Dimensionality



①

②

- F1 score :

TP = 50
FP = 10
TN = 10
FN = 30
P = 0.83
R = 0.62

(A)

(P)

	+	−
+	TP	FP
−	FN	TN

$$P = \frac{TP}{TP+FP}$$

$$R = \frac{TP}{TP+FN}$$

$\boxed{1, 2, 3, 4, 5}$

$$\rightarrow HM = \frac{5}{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}}$$

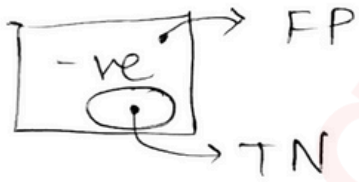
$$\rightarrow f1\,score = \frac{2}{\frac{1}{P}+\frac{1}{R}} = \frac{2 \times P \times R}{P+R}$$

$$= TP / TP + \tfrac{1}{2}(FP+FN)$$

$\rightarrow \boxed{0.70}$

5 Minutes Engineering

- Specificity $= \dfrac{TN}{TN + FP}$

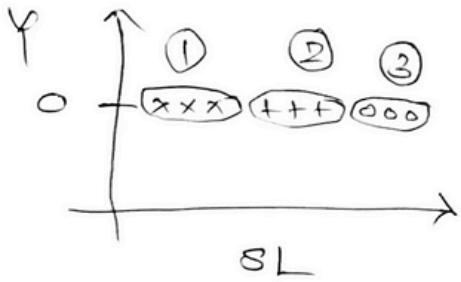


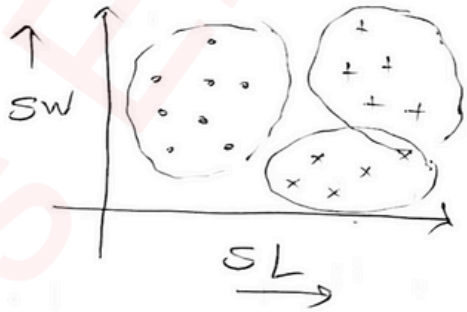
- Sensitivity = Recall $= \dfrac{TP}{TP + FN}$

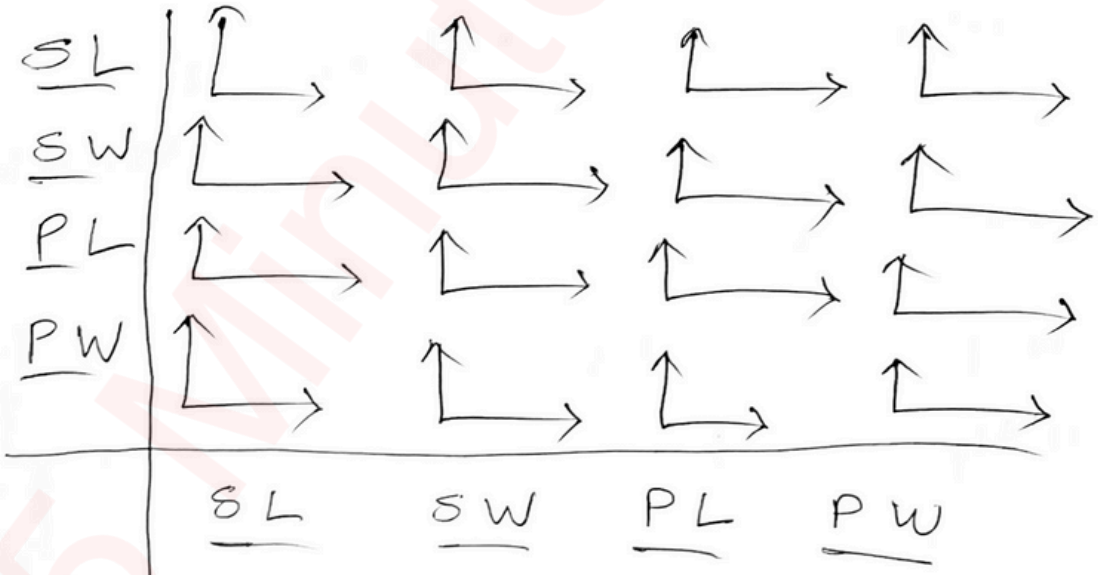
* Uni, Bi, Multi Variate Analysis

SL SW PL PW

- setosa
- versicolor
- virginica







5 Minutes Engineering

- Preprocessing

① Normalization [MinMax Scaler]

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \{0 - 1\}$$

② Standardization [standard scaler]

$(\mu \to 0 \ \& \ \sigma \to 1)$

$$x' = \frac{x - \mu}{\sigma}$$

③ Robust Scaler

↳ IQR (Interquartile Range)

⇒ 10, 20, 30, 30, 45, 50, 55, 60, 70, 80

1st Quartile = 25% P = 30

2nd Quartile = 50% P = 47.5

3rd Quartile = 75% P = 58.75

4rd Quartile = 100% P = 80

$$x' = \frac{x - x_{med}}{Q_3 - Q_1}$$

5 Minutes Engineering

④ Label Encoding

eg:-

f
Pune → 0
Goa → 1
Mumbai → 2
Pune → 0
Punjab → 3
Goa → 2

Pune → 0
Goa → 1
Mumbai → 2
Punjab → 3

⑤ One hot encoding

eg:

f	Pune	Goa	Mumbai	Punjab
Pune →	1	0	0	0
Goa →	0	1	0	0
Mumbai →	0	0	1	0
Pune →	1	0	0	0
Punjab →	0	0	0	1
Goa →	0	1	0	0

5 Minutes Engineering

ML

- **Supervised**
  - **Classification**
    - Decision Tree
    - Logistic Reg.
    - Naive Bayes
    - SVM
    - Random forest
    - KNN
  - **Regression**
    - Linear Reg.
    - Polynomial
    - Ridge & Lasso
    - Decision Tree
    - Random forest
    - SVM Reg.
- **Unsupervised**
  - **Clustering**
    - kmeans
    - Hierarchical
    - DBSCAN
- **Reinforcement**
  - **Decision Making**
    - Q Learning

* Regression:

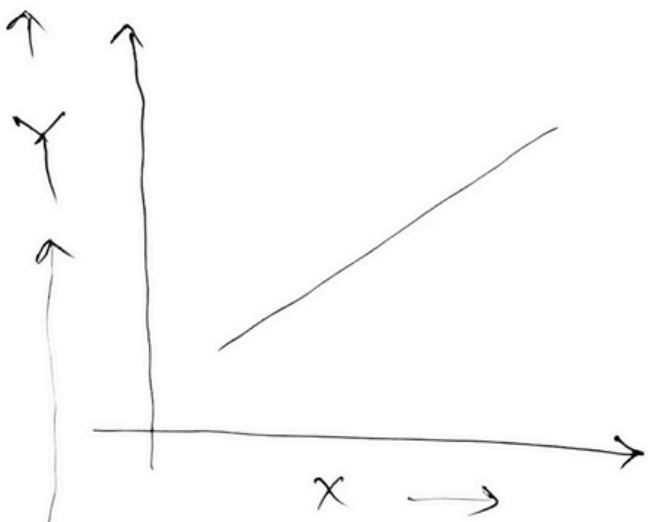
- Continuous data
- Relationship betn Independent & dependent variables

① Linear regression

- Simple (1 IV)
- Multiple (>1 IV)

5 Minutes Engineering

Simple Linear Regression :



$\Rightarrow \quad y = mx + c$

$\downarrow$

$y_p = b_0 + b_1 X$

1 IV

eg:- Marks = $b_0 + b_1$(Hours)

Multiple Linear Regression

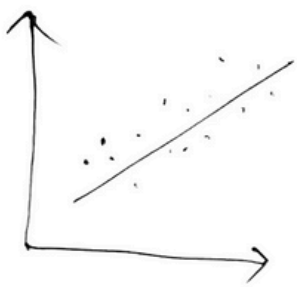
x1	x2	x3	y

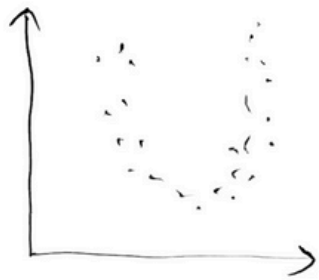
Hours	Attend.	Assig.	Marks

Marks = $b_0 + b_1$(Hours) + b_2(Attendance) + b_3(Assignment)

5 Minutes Engineering

Polynomial Regression





$$y = b_0 + b_1 x_1$$

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4$$

$$y = \alpha_0 + \sum_{i=1}^{m} b_i x_i$$

0 degree poly: $y =$ constant

1 degree poly: $y = mx^1 + c$

2 degree poly: $y = ax^2 + bx + c$

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_n x^n$$

$$\boxed{y = b_0 + \sum_{i=1}^{m} b_i x_i + \boxed{F_p}}$$

polynomial feature

5 Minutes Engineering

- Ridge Regression

Lasso Regression

Regularization

(can be zero)

$RR = Loss + \alpha \|w\|^2$

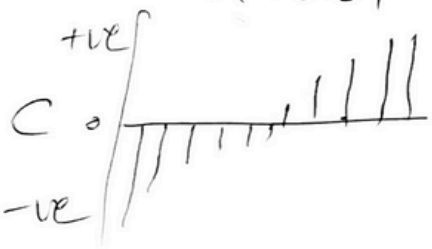
(penalty)

$LR = Loss + \alpha \|w\|$

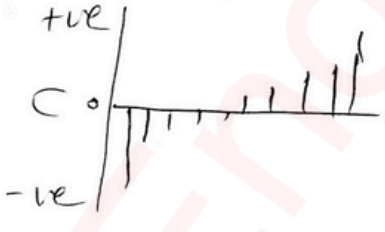
absolute value.

$\|w\|^2 = w_1^2 + w_2^2 + w_3^2 + w_4^2 + \cdots w_n^2$

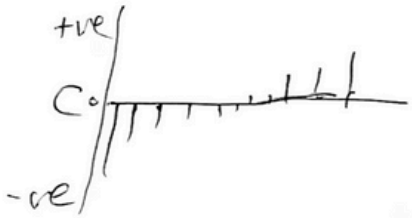
$\alpha = 0.01$



$\alpha = 1$



$\alpha = 2$



$y = 0.9 + 1.2x_1 + 20x_2 + 39x_3$

$\Downarrow$

$y = 0.9 + 0.7x_1 + 2x_2 + 5x_3$

- ElasticNet Regression (Hybrid)

$RR = Loss + \alpha \|w\|^2$

$LR = Loss + \alpha \|w\|$

$ER = Loss + \alpha_1 \|w\|^2 + \alpha_2 \|w\|$

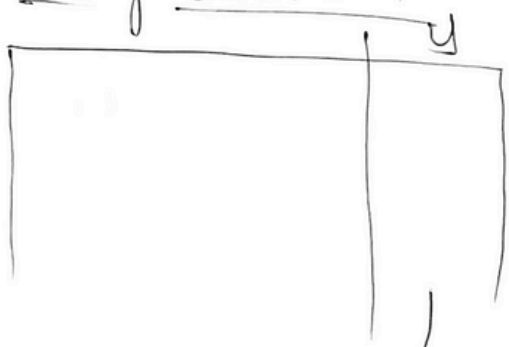
(multicollinearity) RR

(feature selection) LR

5 Minutes Engineering

- Decision Tree, SVM, KNN, Random forest

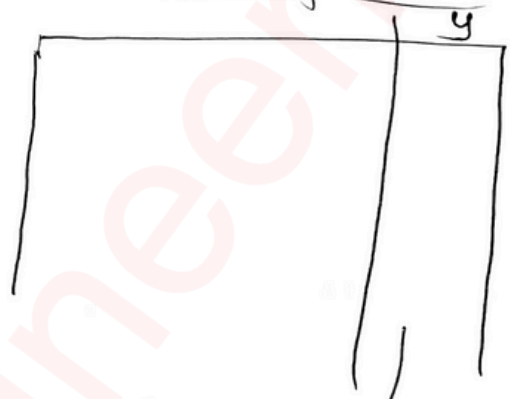
Regression



Continuous

(Mean/Average)

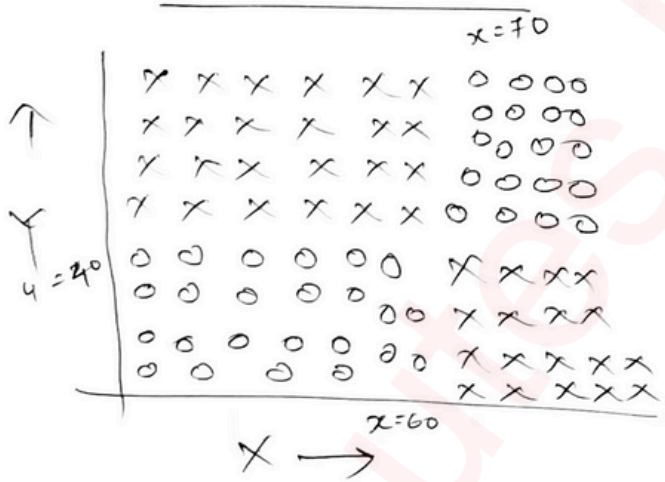
Classification

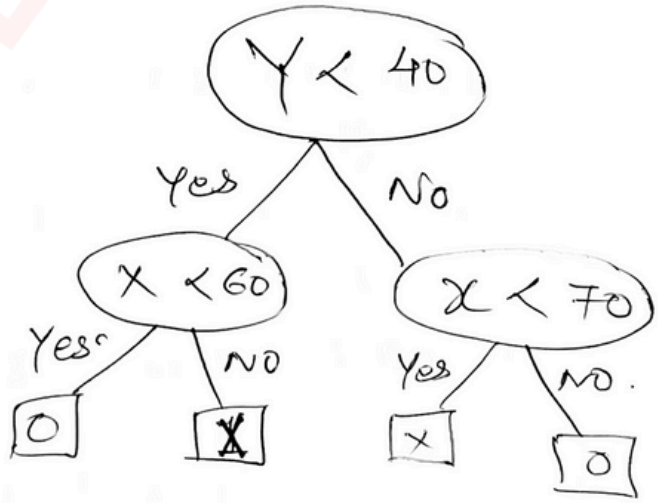


Categorical

(Majority)*

* Decision Tree





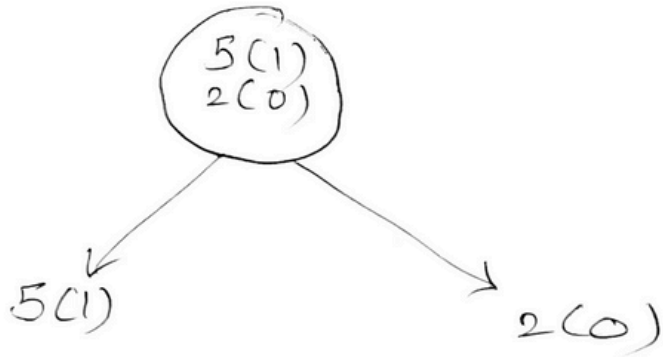
Entropy (E) = $-P(+)\log_2 P(+) - P(-)\log_2 P(-)$

Probability of +ve class

Probability of -ve class.

5 Minutes Engineering

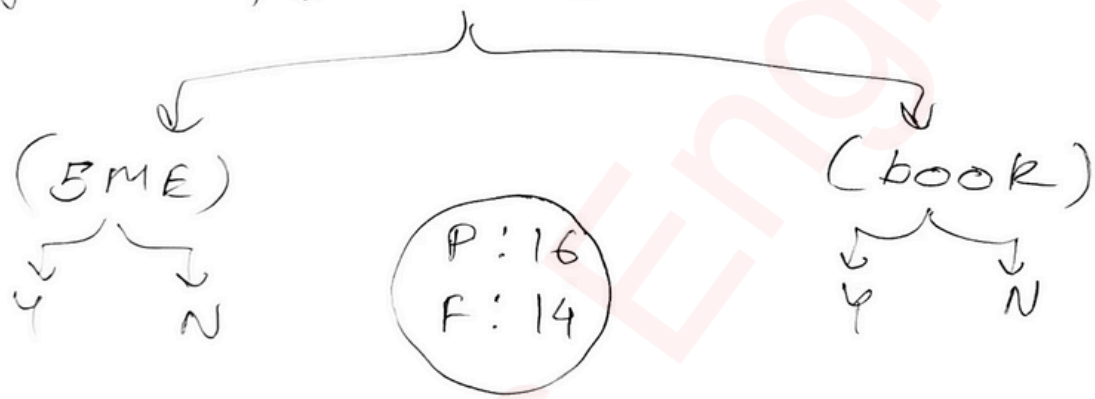
eg:



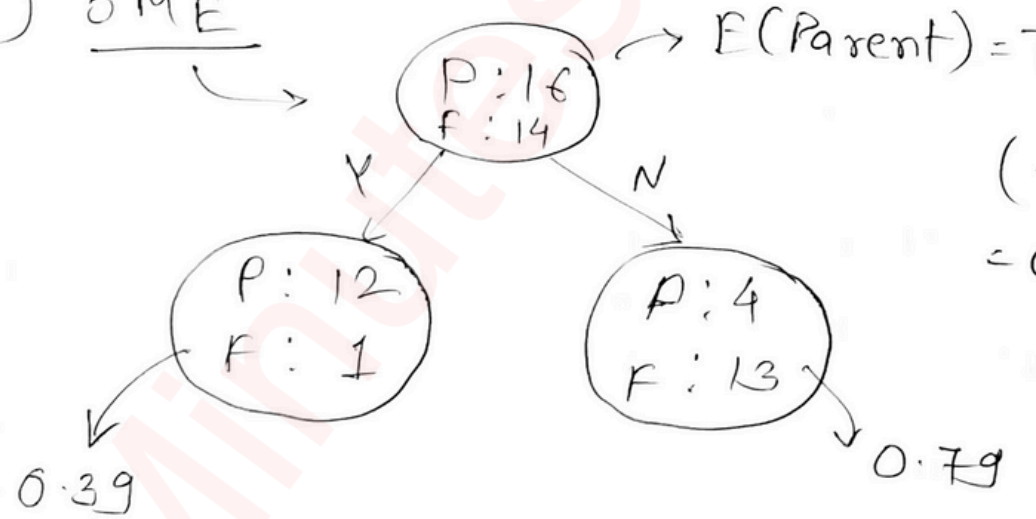
$$\Rightarrow -\left(\frac{5}{7}\right)\log_2\left(\frac{5}{7}\right) - \left(\frac{2}{7}\right)\log_2\left(\frac{2}{7}\right)$$

$$\Rightarrow 0.85$$

eg:- <u>2 features</u>



① <u>SME</u>



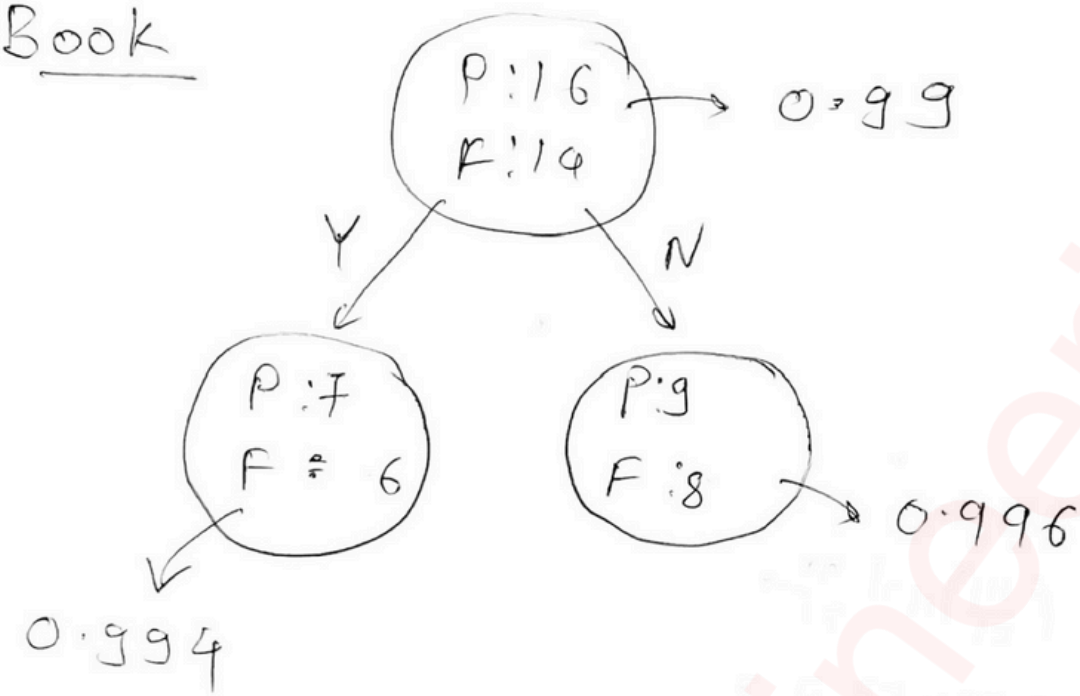
$$E(\text{Parent}) = -\left(\frac{16}{30}\right)\log_2\left(\frac{16}{30}\right) - \left(\frac{14}{30}\right)\log_2\left(\frac{14}{30}\right) = 0.99$$

$$\Rightarrow \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79 = 0.62$$

$$\therefore\ 0.99 - 0.62 = \underline{0.37} \rightarrow IG.$$

5 Minutes Engineering

② Book



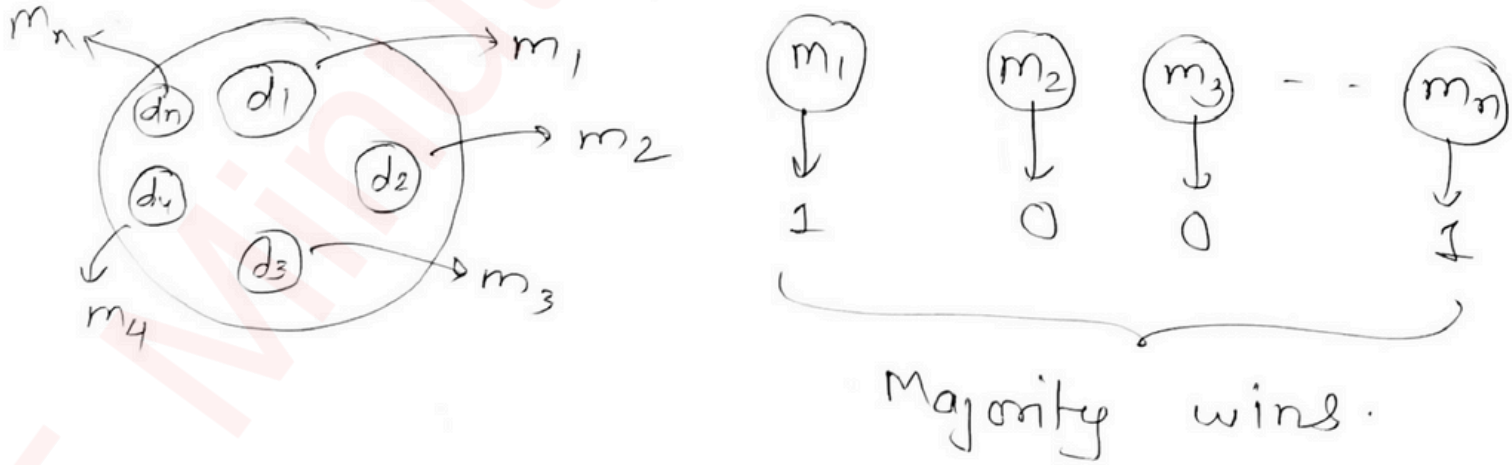
$$\Rightarrow \frac{13}{30} \times 0.994 + \frac{17}{30} \times 0.996$$

$$\Rightarrow 0.99$$

$$\longrightarrow \quad 0.99 - 0.99 = 0.00 \rightarrow IG$$

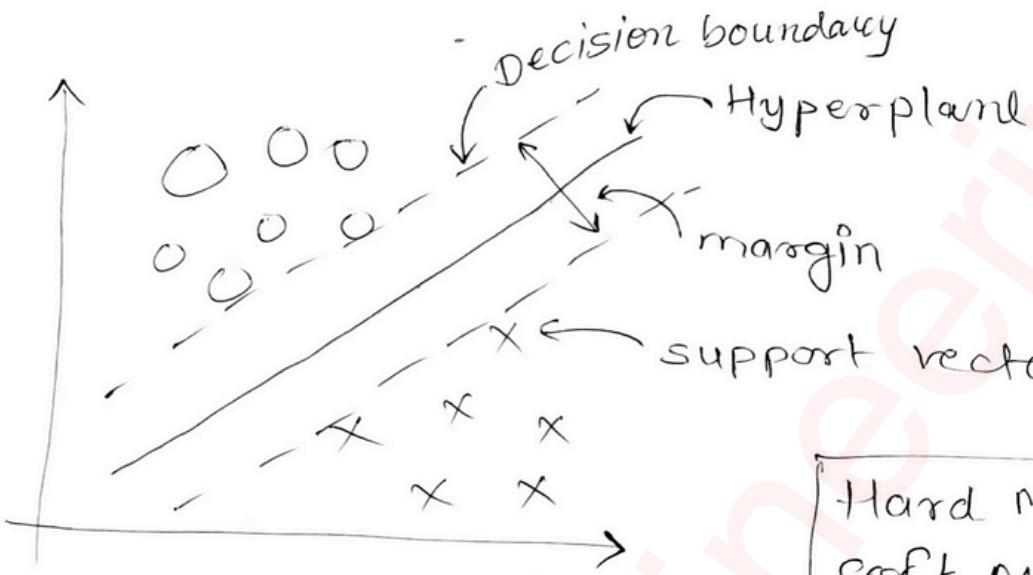
$$\boxed{0.37 > 0.00}$$

→ Random forest



5 Minutes Engineering

◦ SVM : Support Vector Machine



Hard Margin
Soft Margin

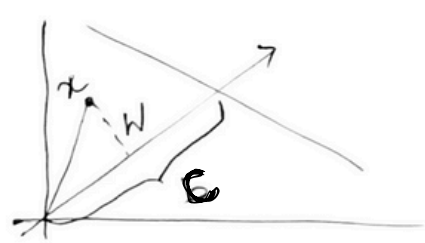
⇒ Binary classification

$+1$ $\quad$ -1

Hyperplane eq^n (H) = $w^T x + b = 0$

Normal vector to hyperplane.
(direction $\perp$)

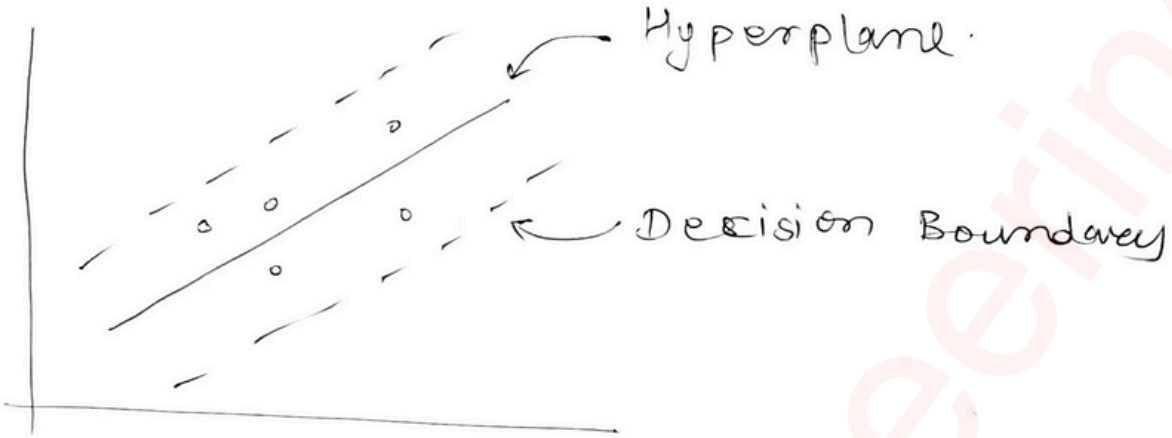
Linear SVM : $y_p = \begin{cases} +1 & : w^T x + b \geq 0 \\ -1 & : w^T x + b < 0 \end{cases}$



$\vec{x} \cdot \vec{w} = c$ (on H)
$\vec{x} \cdot \vec{w} > c$ (+ve)
$\vec{x} \cdot \vec{w} < c$ (−ve)

5 Minutes Engineering

SVR (Regression)



→ Best hyper plane that has max number of points.

eqn of (H)

↳ $Y = wx + b$

eqn of DB

↳ $wx + b = +a$

$wx + b = -a$

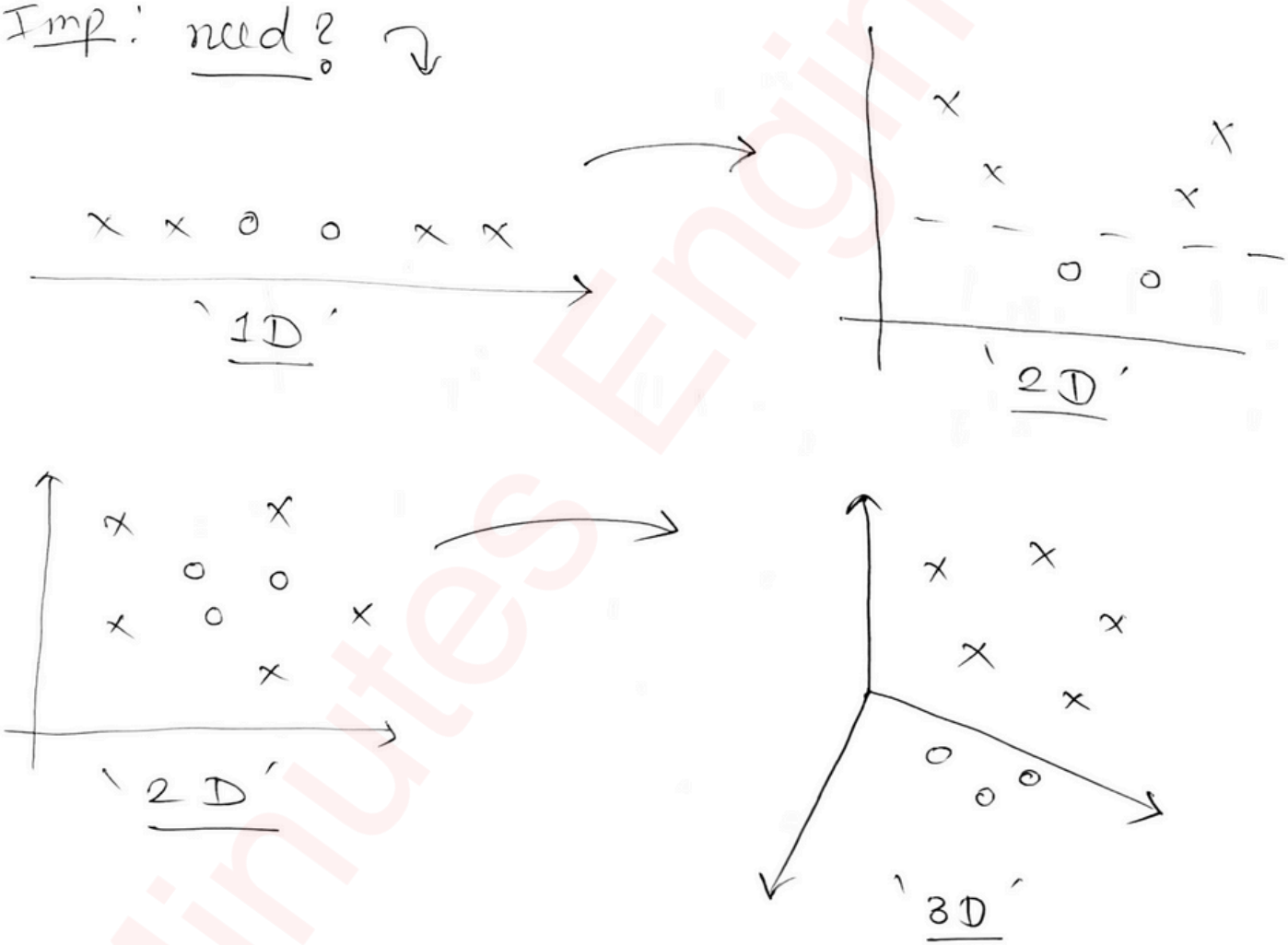
Range: → $-a < Y - wx + b < +a$

5 Minutes Engineering

- SVM kernel function:
  - → Higher 'D' ← To — Lower 'D'

→ Linear
→ polynomial
→ RBF

Imp: need? ↓



Linear kernel
→ $k(x, x') = x\, x'$ $\quad [\, f(x) = x \,]$

Polynomial kernel
→ $k(x, x') = (1 + x^T x')^D$

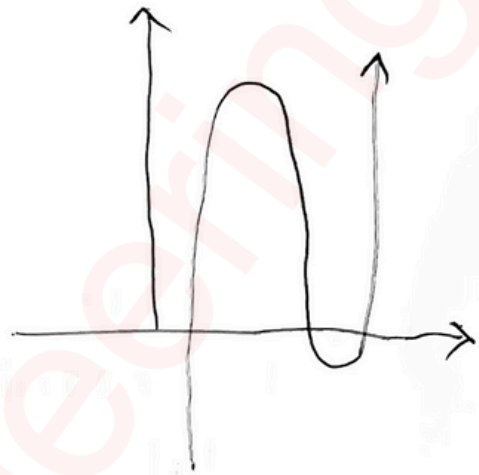
5 Minutes Engineering

$$f(x_1, x_2) = (1 + x_1^T \cdot x_2)^{D=2}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} x_1^2 & x_1 \cdot x_2 \\ x_1 \cdot x_2 & x_2^2 \end{bmatrix}$$



$$\Rightarrow \quad x_1, \; x_2, \; x_1^2, \; x_2^2, \; x_1 \cdot x_2$$

<u>RBF kernel</u>

$$K(x, x') = e^{-\gamma \| x - x' \|^2}$$

"$(x_2 | y)$"

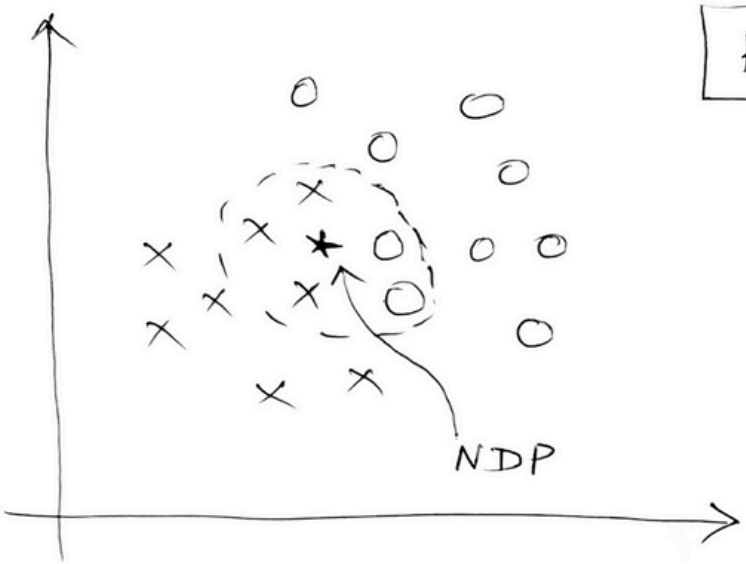
Euclidean (d) distance betn (x_1, x_2) or (x, x').

$$\gamma = \frac{1}{2\sigma^2}$$

$$= e^{-d\gamma}$$

5 Minutes Engineering

KNN (k-nearest neighbors).



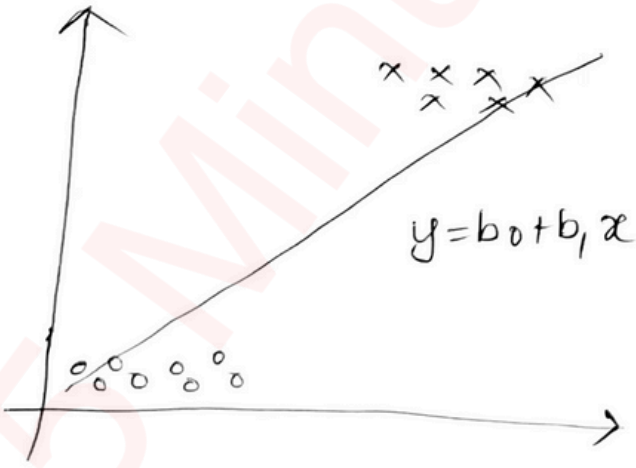
$\boxed{k = 5}$

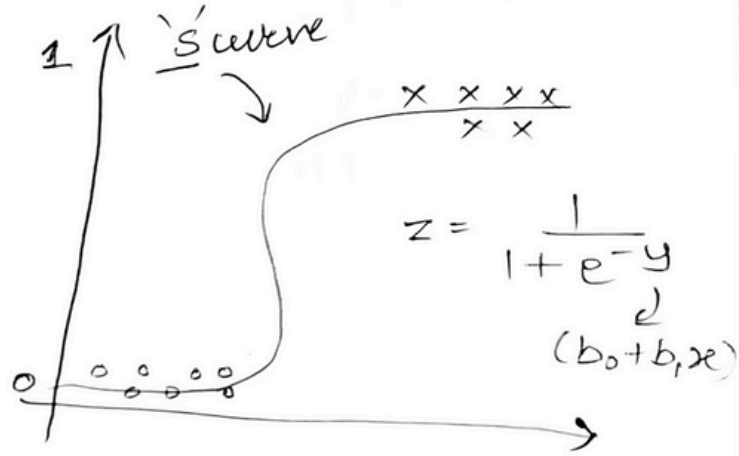
Classification (Majority)

Regression (Average/mean).

$$\frac{\overset{\times}{y_1} + \overset{\times}{y_2} + \overset{\times}{y_3} + \overset{\circ}{y_4} + \overset{\circ}{y_5}}{k = 5} = y_{avg} \checkmark$$

Logistic Regression



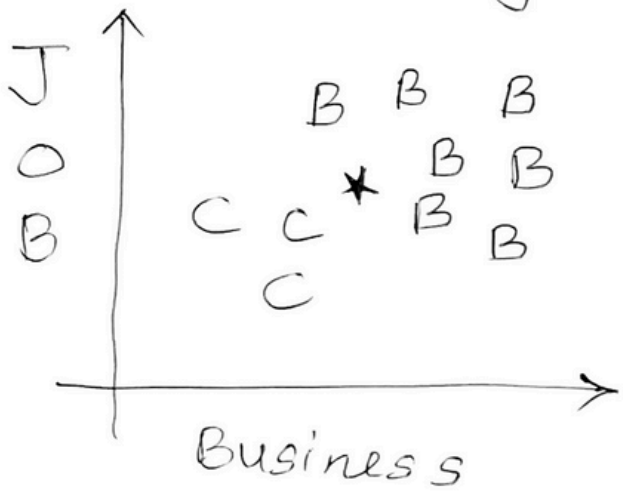


$$z = \frac{1}{1+e^{-y}}$$

$(b_0 + b_1 x)$

5 Minutes Engineering

Naive Bayes



B : Bike
C : Car

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$ ← Bayes theorem.

$$P(y/x) = \frac{P(x/y) \cdot P(y)}{P(x)}$$

$$P(y/x_1, x_2, x_3 \cdots x_n) = \frac{P(x_1/y) \times P(x_2/y) \times \cdots P(x_n/y) \times P(y)}{P(x_1) \times P(x_2) \times \cdots P(x_n)}.$$

x_1 → Yes, No

x_2 → Yes, No

y → Bike, Car.

5 Minutes Engineering

$$P(\text{Car}) = \frac{3}{10} = 0.3$$

$$P(\text{Bike}) = \frac{7}{10} = 0.7$$

$$P(\text{Car} / J, B) = \frac{P(J/\text{Car}) \times P(B/\text{Car}) \times P(\text{Car})}{P(J) \times P(B)}$$

$$= 1/3 \times 2/3 \times 3/10$$

$$= 2/30 = 0.066$$

$$P(\text{Bike} / J, B) = \frac{P(J/\text{Bike}) \times P(B/\text{Bike}) \times P(\text{Bike})}{P(J) \times P(B)}$$

$$= 2/7 \times 5/7 \times 7/10$$

$$= 0.142$$

- <u>Gaussian NB</u>

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

5ME
- → Yes [80, 70, 90, 75, 70, 86, 96, 80, 65]
- → NO [70, 90, 91, 95, 85]

$\mu(Y) \rightarrow 79.1$ $\sigma(Y) \rightarrow 10.2$

$\mu(N) \rightarrow 86.2$ $\sigma(N) \rightarrow 9.7$

5 Minutes Engineering

$P(\text{Marks} = 74 \mid 5ME = Y)$

$$= \frac{1}{\sqrt{2\pi}\,(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}}$$

$$= \underline{0.034}$$

$P(\text{Marks} = 74 \mid 5ME = N)$

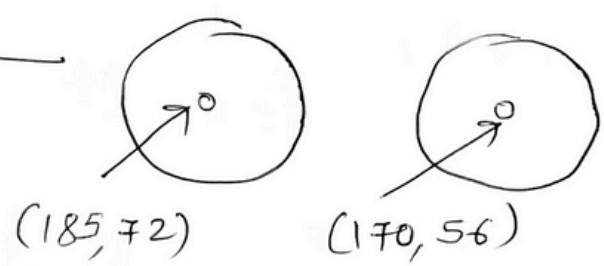
$$= \frac{1}{\sqrt{2\pi}\,(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}}$$

$$= 0.018$$

* Unsupervised Learning:

(i) Kmeans:

	Height	Weight
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77
7	180	71
8	180	70
9	183	84
10	180	88
11	180	67
12	177	76



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

5 Minutes Engineering

ED for ③ :-

$k1 \rightarrow \sqrt{(168-185)^2 + (60-72)^2}$

$d_1 \rightarrow 20.80$

$k2 \rightarrow \sqrt{(168-170)^2 + (60-56)^2}$

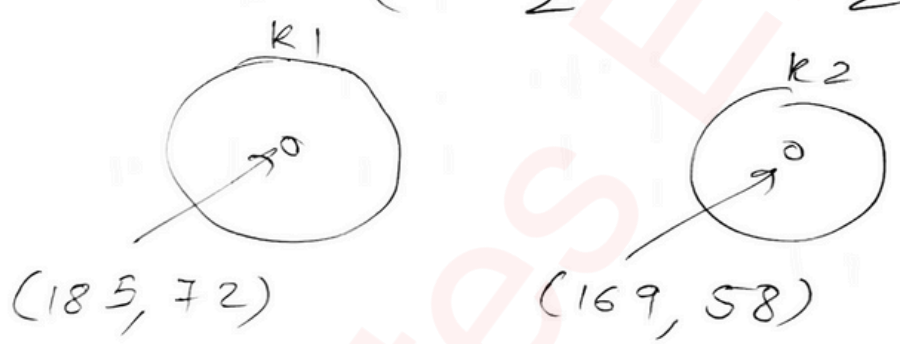
$d_2 \rightarrow 4.48$

$d_2 < d_1$

③ $\longrightarrow$ k_2 ✓

New centroid calculation:

for $k_2 = \left(\frac{170+168}{2}, \frac{60+56}{2}\right) = (169, 58)$



ED for k1 $\rightarrow \sqrt{(179-185)^2 + (68-72)^2}$

$d_1 \rightarrow 7.21$

for k2 $\rightarrow \sqrt{(179-169)^2 + (68-58)^2}$

$d_2 \rightarrow 14.14$

Ans:- $k1 \rightarrow \{1, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

$k2 \rightarrow \{2, 3\}$

5 Minutes Engineering

② k medoids Clustering

	X	Y	K1	K2	Cost	K1	K2	Cost
①	8	2	0	6	0	0	2	2
②	3	5	8	2	2	8	6	6
③	4	7	9	3	3	9	7	7
④	8	4	2	4	2	2	0	2
⑤	5	5	6	0	0	6	4	4
					7			21

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

$= |5-8| + |5-2|$

$= 3 + 3 = 6$

$\Rightarrow |3-8| + |5-2|$

$= 5 + 3 = 8$

$\Rightarrow |3-5| + |5-5|$

$= 2$

$\Rightarrow |4-8| + |7-2|$

$= 4 + 5 = 9$

$\Rightarrow |4-5| + |7-5|$

$\Rightarrow 1 + 2 = 3$

$\Rightarrow |8-8| + |4-2| = 2$

$\Rightarrow |8-5| + |4-5| = 4$

$\rightarrow |8-8| + |4-2| = 2$

$\rightarrow |3-8| + |5-2| = 8$

$\rightarrow |3-8| + |5-4| = 6$

$\rightarrow |4-8| + |4-7| = 7$

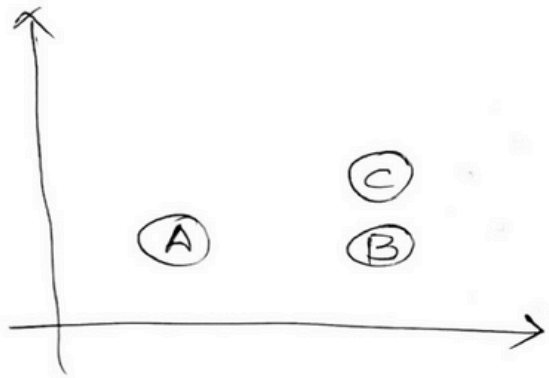
$\rightarrow |5-8| + |4-5| = 4$

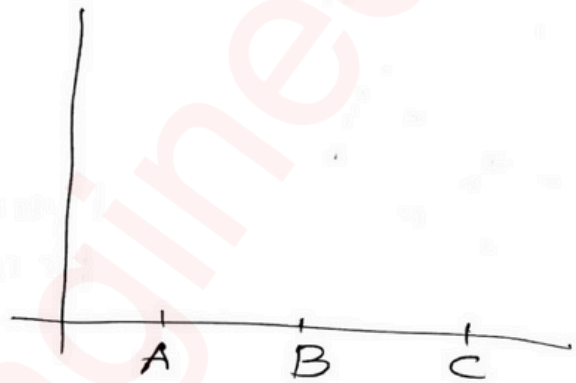
5 Minutes Engineering

③ Hierarchical clustering:

- → Agglomerative
- → Divisive

* Dendrogram.



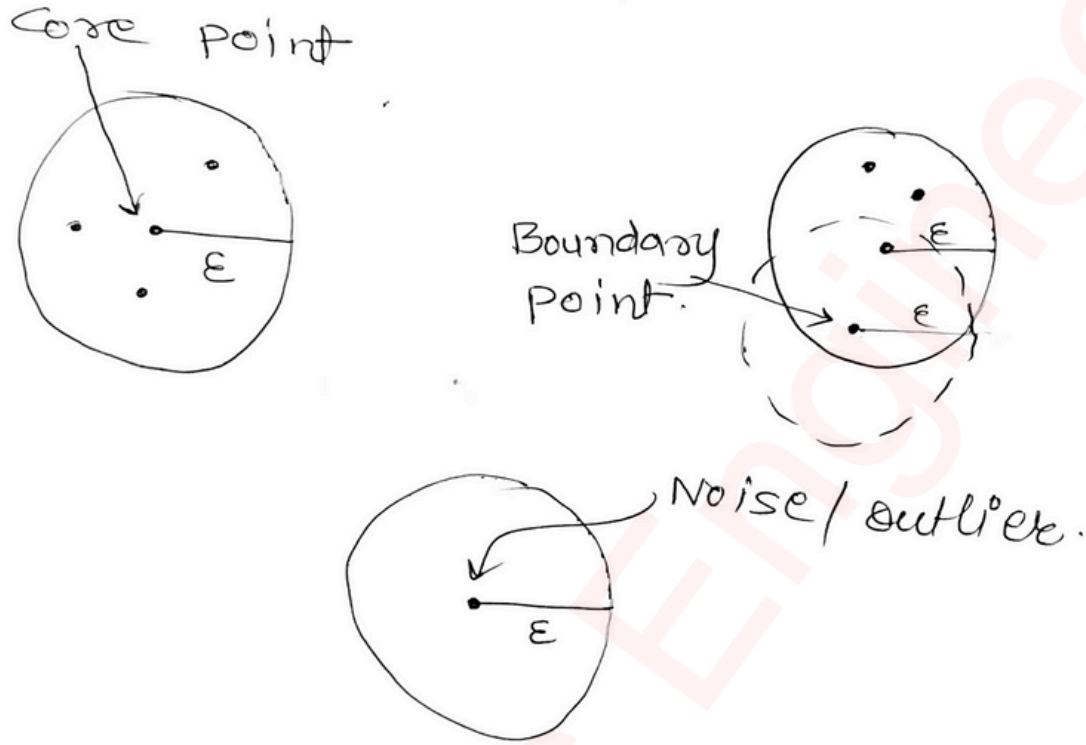


5 Minutes Engineering

④ DBSCAN

↳ Density Based spatial clustering of Applications with Noise.

'ε' & 'minpoint'



⑤ PCA (Principal Component Analysis)

↳ Dimension Reduction Techniques

Steps:

① D + 1 ⟶ D

② Compute μ for every D.

③ Calculate Covariance matrix.

④ Calculate eigen value & eigen vector.

⑤ Top k - eigen values.

⑥ (feature vector)^T × (Original Dataset)^T = New data.

5 Minutes Engineering

Eg:

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9
$\bar{X} = 1.81$	$\bar{Y} = 1.91$

$$Cov(x, y) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{x})}{n-1}$$

$$C = \begin{bmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{bmatrix}$$

$\Rightarrow$

X	$X - \bar{X}$	$(X-\bar{X})(X-\bar{X})$
2.5	0.6	0.476
0.5	-1.31	1.716

Sum = 5.5490

$\Rightarrow$

Y	$Y - \bar{Y}$	$(Y-\bar{Y})(Y-\bar{Y})$
2.4	0.49	0.2401
0.7	-1.21	1.4641

Sum = 6.449

$\Rightarrow$

X	Y	$X - \bar{X}$	$Y - \bar{Y}$	$(X-\bar{X})(Y-\bar{Y})$
2.5	2.4	0.69	0.49	0.3381
0.5	0.7	-1.31	-1.21	1.5851
				sum = 5.5390

5 Minutes Engineering

$$C = \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix}$$

$$\Rightarrow \quad C - \lambda I = 0$$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{bmatrix}$$

$$\Rightarrow \quad \lambda^2 - 1.333\lambda + 0.0630 = 0$$

$$\lambda_1 = 0.0490$$

$$\lambda_2 = 1.2840$$

$$\Rightarrow \quad C\vec{v} = \lambda \vec{v}$$

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7164 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = 0.0490 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\Rightarrow 0.6165\, x_1 + 0.6154\, y_1 = 0.0490\, x_1$$

$$\Rightarrow 0.6154\, x_1 + 0.7165\, y_1 = 0.0490\, y_1$$

$$\Rightarrow \quad 0.5674\, x_1 = -0.6154\, y_1$$

$$\Rightarrow \quad 0.6154\, x_1 = -0.6674\, y_1$$

$$\rightarrow \quad x_1 = -1.0845\, y_1$$

$$\begin{bmatrix} -1.0845 \\ 1 \end{bmatrix} = \frac{1.17614 + 1}{\sqrt{2.17614}} = 1.47517$$

5 Minutes Engineering

$$\Rightarrow \begin{bmatrix} -0.7351 \\ 0.6778 \end{bmatrix} \leftarrow \gamma_1$$

<u>then</u>, $x_2 = 0.92194\, k_2$

$$\rightarrow \begin{bmatrix} 0.92194 \\ 1 \end{bmatrix} = 0.8499 + 1$$

$$= \sqrt{1.8499}$$

$$= 1.3601$$

$$\Rightarrow \begin{bmatrix} 0.6778 \\ 0.7351 \end{bmatrix} \leftarrow \gamma_2$$

$\Rightarrow$ Sum of λ_1 & λ_2 $= 1.2840 + 0.0490$

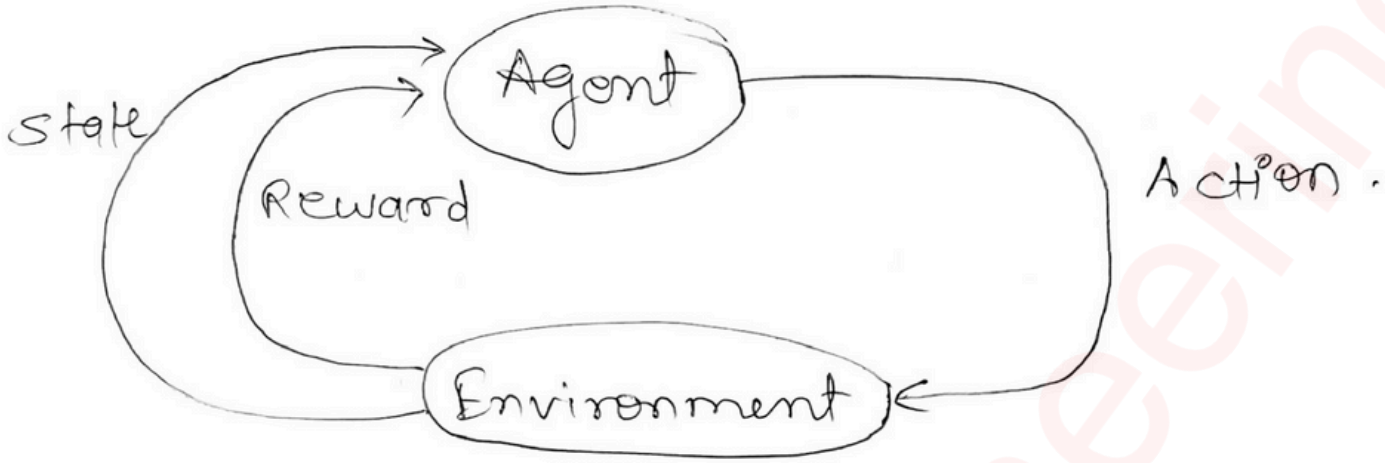
$= 1.33$

$\lambda_1 > \lambda_2$

So, $\gamma_1 \rightarrow$ is one principal component.

5 Minutes Engineering

* Reinforcement Learning:



Elements of RL

- → Policy
- → Reward f^n
- → value f^n

① **Policy**: Defines agents behavious for a given state / time.

state $\xrightarrow{\text{mapping}}$ Action

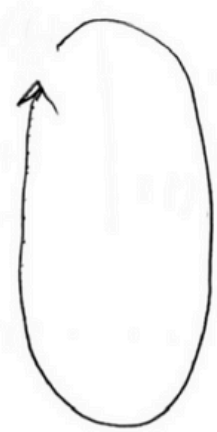
② **Reward f^n**: It provides a numerical score based on state of environment

5 Minutes Engineering

③ <u>value</u> f^n: value of state is total amount of reward an agent can expect to gain over the future starting from that particular state.

① Q <u>Learning</u>

- → Initialize Q-table
- → Choose an action
- → Perform action
- → Measure reward
- → Update Q-table



5 Minutes Engineering

① Initialize Q table

↳ It's a simple lookup table where we calculate max. expected future rewards for action at each state.

Actions: ↑ → ↓ ←

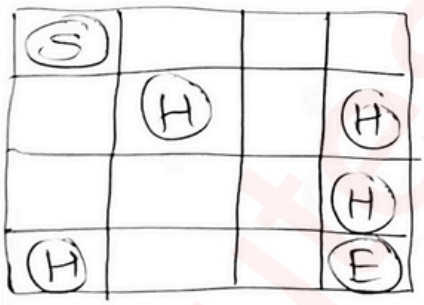
S	0	0	0	0
B	0	0	0	0
H	0	0	0	0
E	0	0	0	0

Rows :- no. of states

Column: no. of actions

Start, Blank, Hole, End

② choose an action:



→ '→' or '↓'

③ Perform '↓'

Rewards → Reaching goal = 1

Hole = 0

Blank = 0

5 Minutes Engineering

- Update Q table
  - ↳ using Q function
    - ↳ uses Bellman eqn.

$$\Downarrow$$

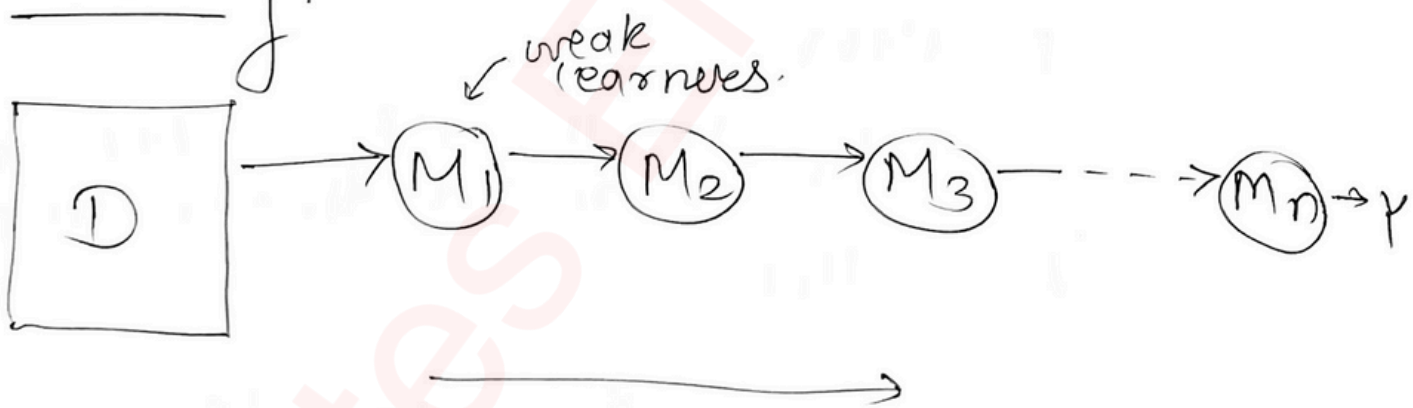
$$\underbrace{NQ(s,a)}_{\text{New Q value}} = \underbrace{Q(s,a)}_{\text{Current Q value}} + \underbrace{\alpha}_{\text{Learning Rate}} \Big[\underbrace{R(s,a)}_{\text{Reward for taking 'a' at 's'}} + \underbrace{\gamma}_{\text{Discount Rate.}} \underbrace{\max Q'(s',a')}_{\text{Max future reward.}} - Q(s,a) \Big]$$

5 Minutes Engineering

* Ensemble Learning

- → Bagging (Random forest)
- → Boosting
  - → AdaBoost
  - → Gradient boosting.

◦ Boosting:



Sequence
not parallel.

$$\alpha_1 M_1 + \alpha_2 M_2 + \alpha_3 M_3 - - - + \alpha_n M_n$$

$\alpha \rightarrow$ weights / Importance / Influence.

5 Minutes Engineering

① Dataset:

①	1/5
②	1/5
③	1/5
④	1/5
⑤	1/5

equal sample weight.

② Get the best model.

→ DT (Highest IG)

③ Calculate the 'α'

→ $\frac{1}{2} \log \frac{1-TE}{TE}$ ← Miss-classified.

$$\text{Performance} = \frac{1}{2} \log \frac{1-TE}{TE}$$

$$\alpha = \frac{1}{2} \log \frac{1-1/5}{1/5}$$

$$\boxed{\alpha = 0.69}$$

④

	New SW	NSW	NNSW
①	1/5	0.10	0.126
②	1/5	0.10	0.126
③	1/5	0.39	0.493
④	1/5	0.10	0.126
⑤	1/5	0.10	0.126
		0.79	→ 0.997

5 Minutes Engineering

$$NSW = OW * e^{\pm\alpha}$$

'−' → correct

'+' → wrong

$$NSW_{(correct)} = \frac{1}{5} * e^{-0.69} = 0.1004$$

$$NSW_{(wrong)} = \frac{1}{5} * e^{+0.69} = 0.3988$$

⑤ Bins / Bucket

	NNSW	Bins
①	0.126	0 — 0.126
②	0.126	0.126 — 0.252
③	0.493	0.252 — 0.745
④	0.126	0.745 — 0.871
⑤	0.126	0.871 — 0.997

⑥ Random selection (0 − 1)

→ 0.011 → ①

→ 0.33 → ③

→ 0.45 → ③ } 3 times.

→ 0.60 → ③

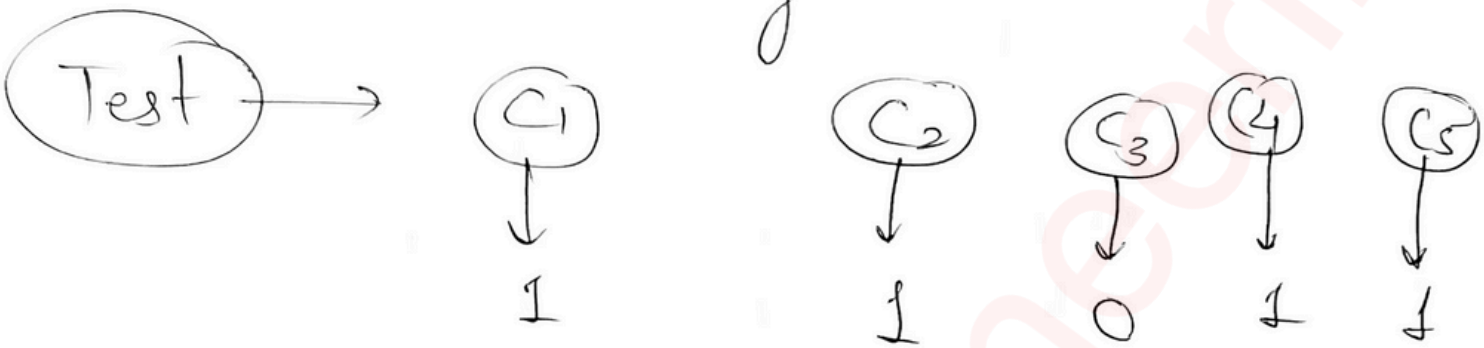
→ 0.90 → ⑤

①
③
⑤

New dataset → M_2

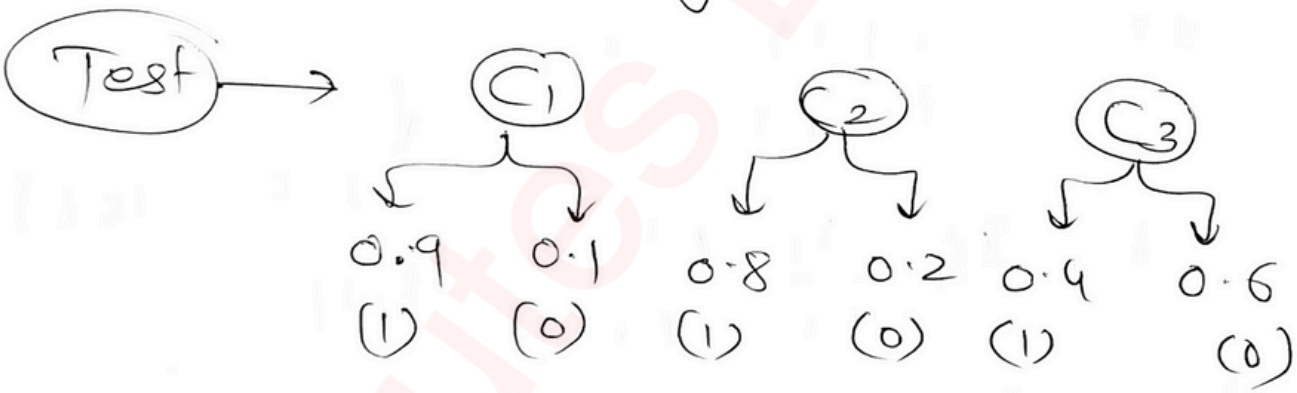
5 Minutes Engineering

- Voting classifier
  - → Hard voting



Prediction → '1' (Majority.

→ soft voting



$$(1) = \frac{0.9 + 0.8 + 0.4}{3} = 0.7 \checkmark$$

$$(0) = \frac{0.1 + 0.2 + 0.6}{3} = 0.3$$

5 Minutes Engineering

Gradient Boosting

Age	Income	P1	E1	P2	E2
20	10k	20k	-10k	19.9k	-9.9k
30	30k	20k	10k	20.1k	9.9k
35	40k	20k	20k	20.2k	19.8k
25	15k	20k	-5k	19.95k	-4.95k
50	5k	20k	-15k	19.85k	-14.85k

$\mu = 20k$

① Base Model Prediction = 20k.

② Train DT on (Age, E1)

I/p → BM → DT1 → O/P

(x, E1)

for I/p Age 20, E1 -10k → = 20k + (-10k) = 10k

③ Introducing LR.

⇒ BMP + LR × DT1

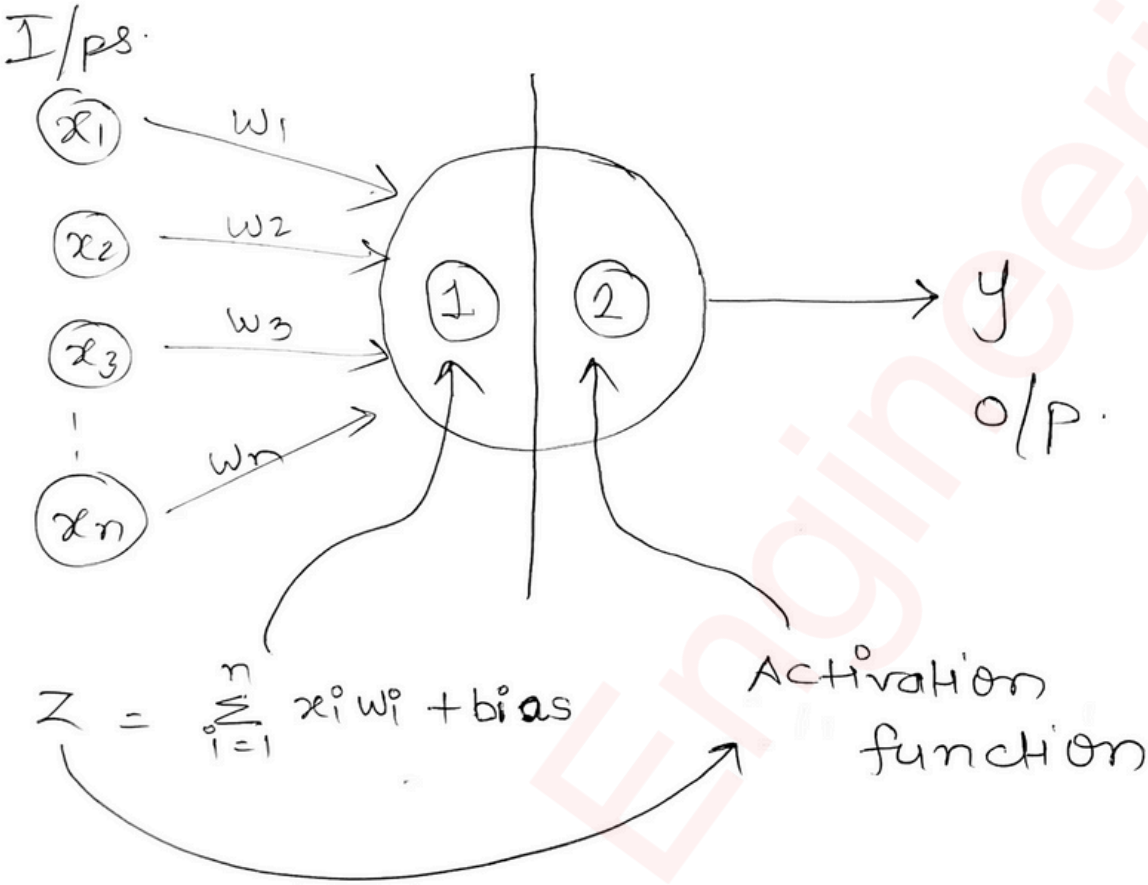
⇒ 20k + (0.01) × (-10k) = 19.9k

Age	E1	P2
30	10k	20.1k
35	20k	20.2k
25	-5k	19.95k
50	-15k	19.85k

5Minutes Engineering

- Neural Network :

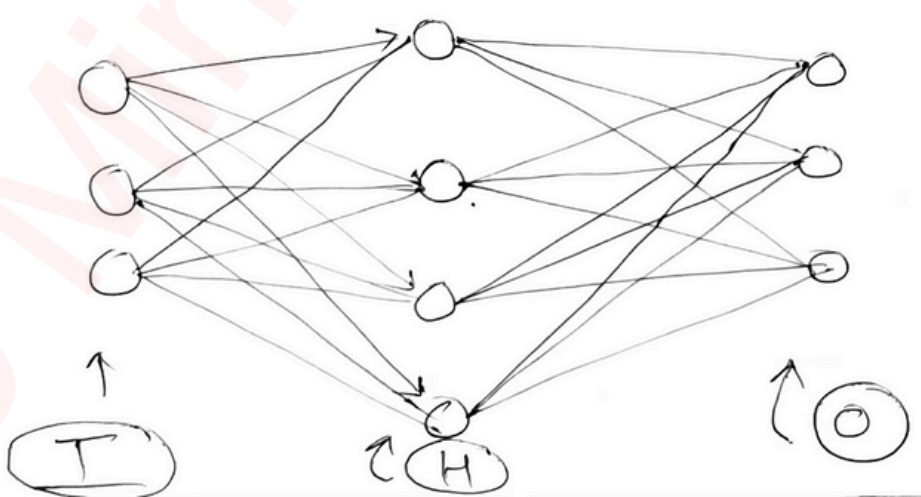
↳ mimic human brain



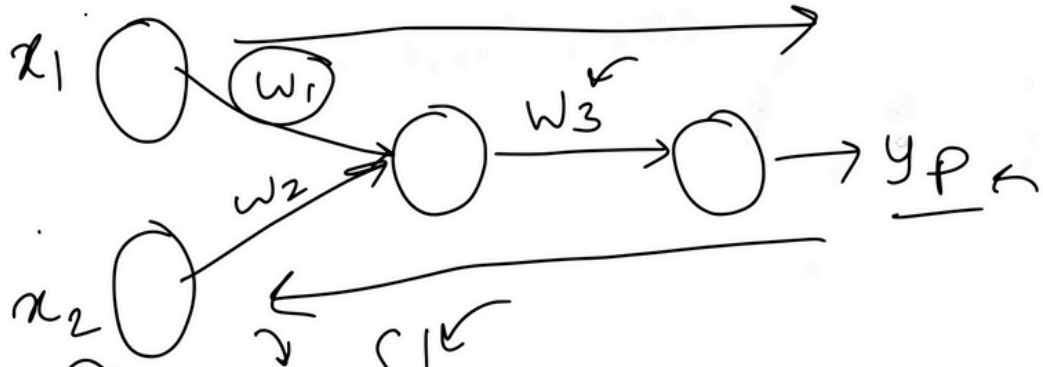
$$Z = \sum_{i=1}^{n} x_i w_i + \text{bias}$$

Layers :

- → Input
- → Hidden
- → Output



5 Minutes Engineering



$$L = (y - y_p)^2$$

$$W_{3\,new} = W_{3\,old} - LR \times \frac{\delta L}{\delta W_3}$$

(0.01)

Types of NN

- ↳ FFN
- ↳ CNN
- ↳ RNN
- ↳ LSTM

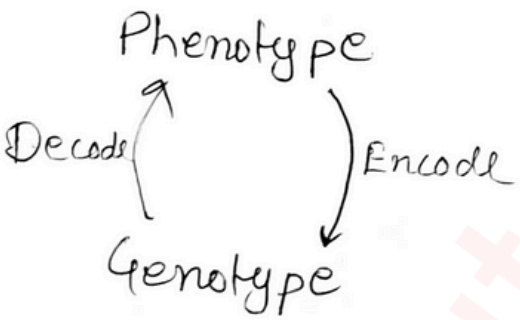
5 Minutes Engineering

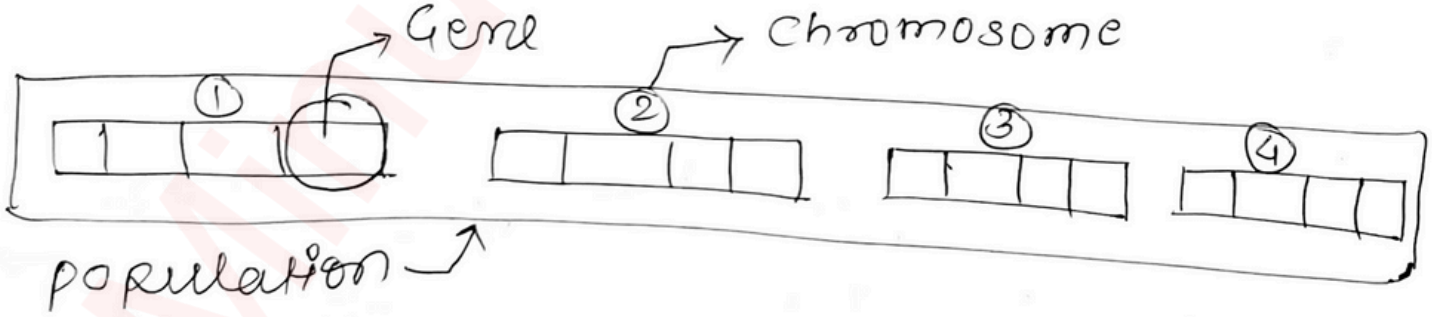
Genetic Algorithm

- → Adaptive heuristic search algo. inspired by "DTOE". (Genetics & natural selection)
- → To generate high quality soln for optimization problem.

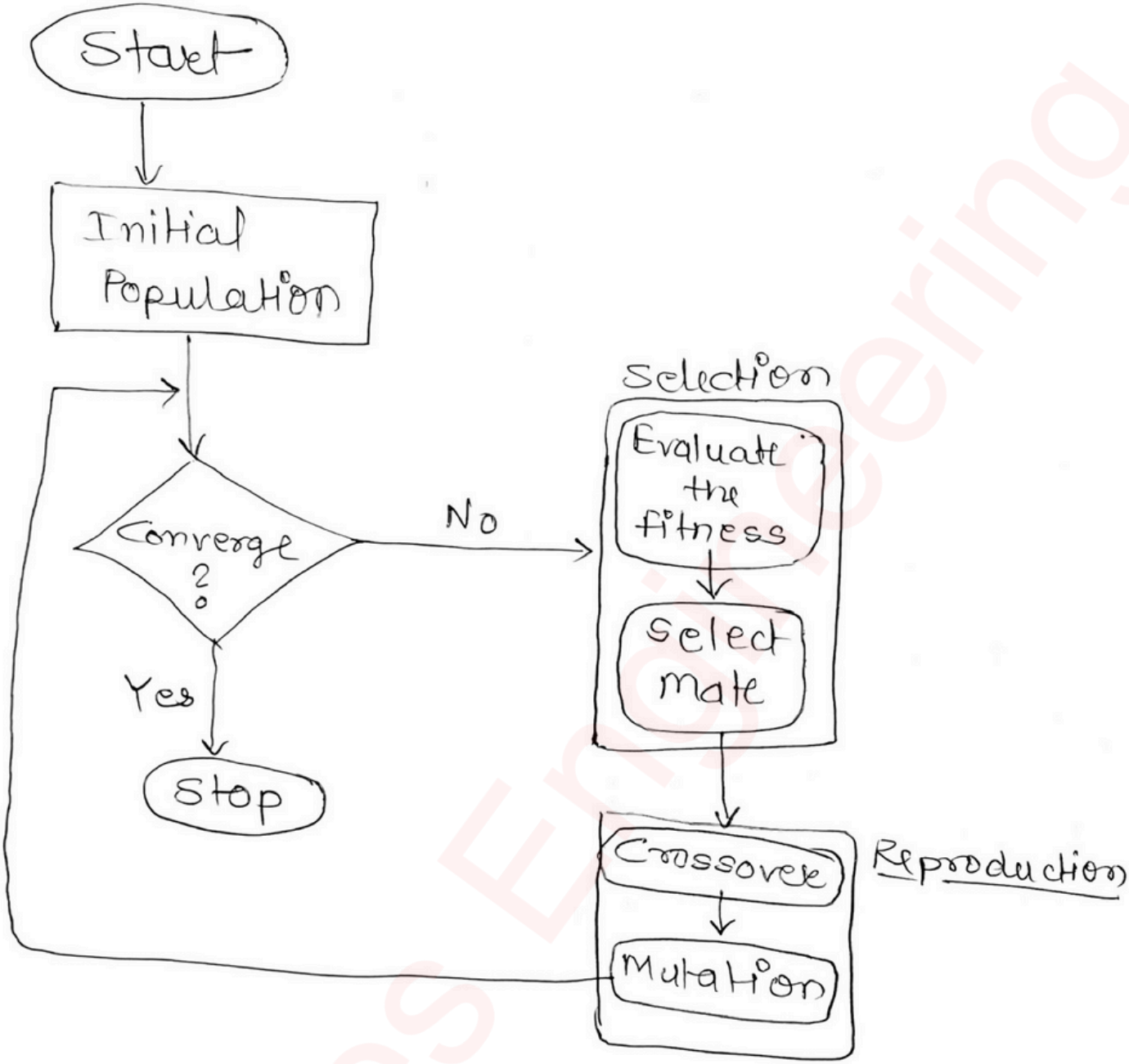
→ Terms

- → Population
- → Chromosomes
- → Gene
- → Allele
- → fitness function
- → Operator of GA
  - → selection
  - → crossover
  - → mutation



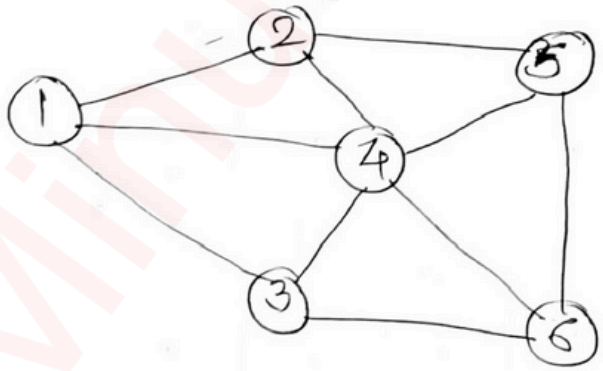


5 Minutes Engineering



fitness function :

eg:-



P1 : 1 2 5 6 4 3 2 1 → 18

P2 : 1 2 5 4 6 3 1 → 20

P3 : 1 2 4 5 6 3 1 → 12

P4 : 1 2 5 6 3 4 1 → 17

5 Minutes Engineering

- Crossover

P1 : | 0 | 1 | 1 | 0 | 1 | 0 |

P2 : | 1 | 1 | 0 | 1 | 0 | 0 |

- single point

O1 : 0 1 0 1 0 0

O2 : 1 1 1 0 1 0

- 2 point

O1 : 0 1 0 1 1 0

O2 : 1 1 1 0 0 0

- Multipoint

O1 : 1 1 0 1 1 0

O2 : 0 1 1 0 0 0

- Mutation

offspring : | 0 | 1 | 1 | 0 | 0 | 1 |

MP (Probability : | 0 | 1 | 0 | 0 | 0 | 1 |

MO : | 0 | 0 | 1 | 0 | 0 | 0 |

or Interchanging or Exchange ⇒ 0 1 1 0 1

0 0 1 0 1 1

5 Minutes Engineering