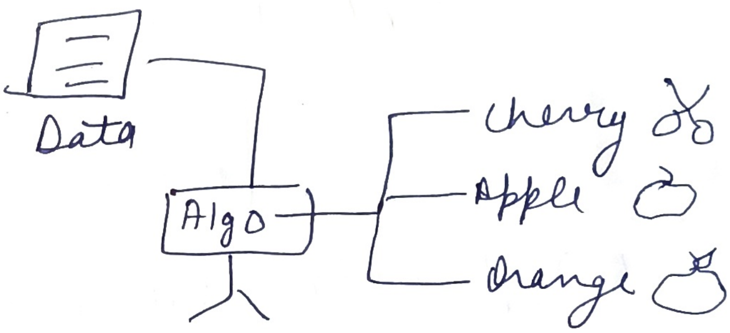
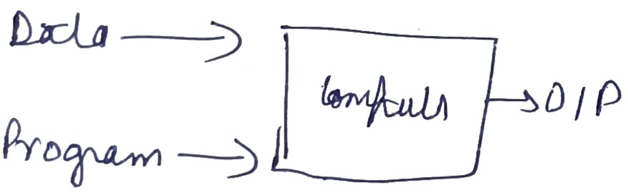
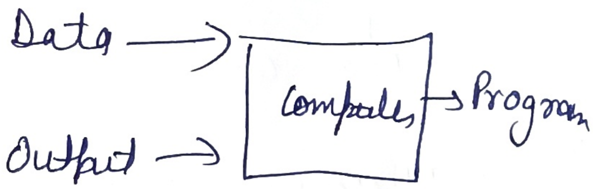
What is Machine Learning?

Machine learning is a subset of artificial intelligence (AI) which provides machines the ability to learn automatically & improve from experience without being explicitly programmed.

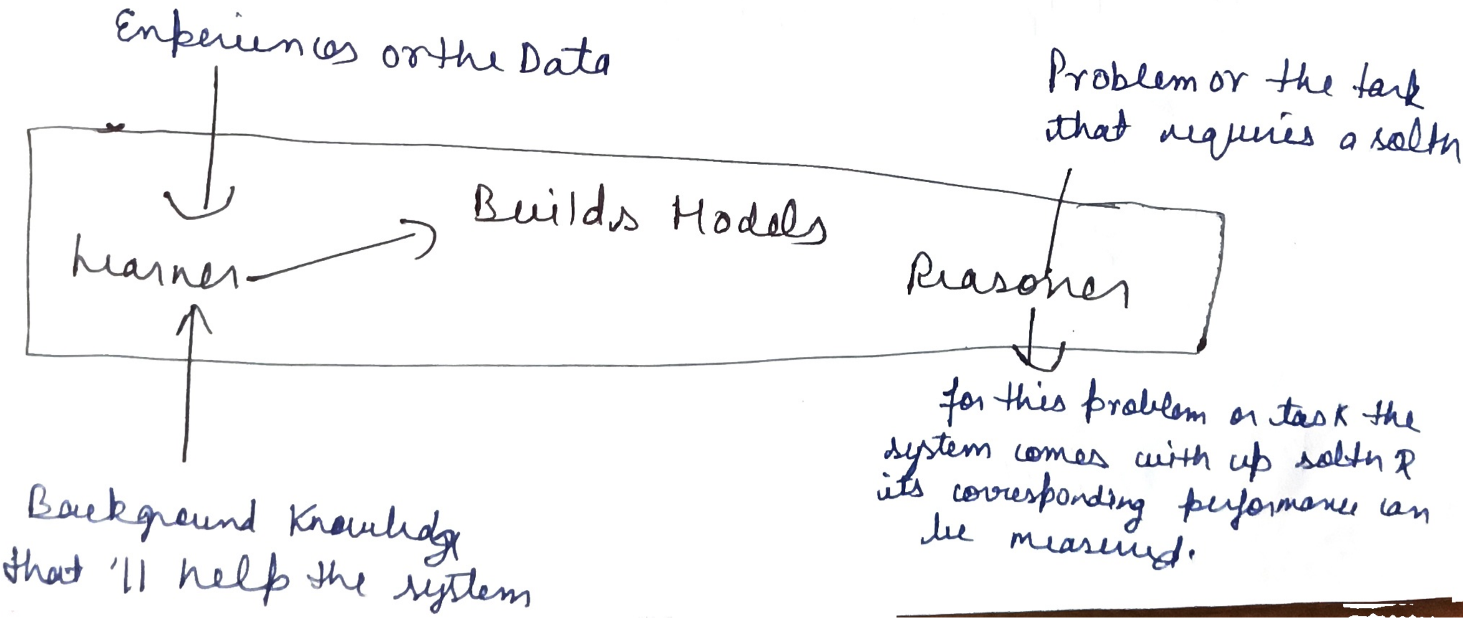


Traditional Learning Machine Learning





Learning System



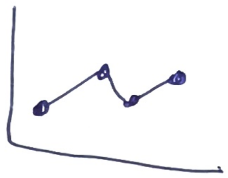
Types of learning

(1) Supervised learning

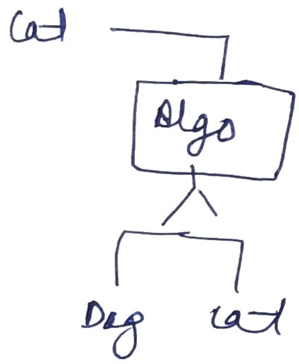
→ Train Me!

→ It is a method in which we teach the machine using labelled data

→ Problem Type: Regression, Classification



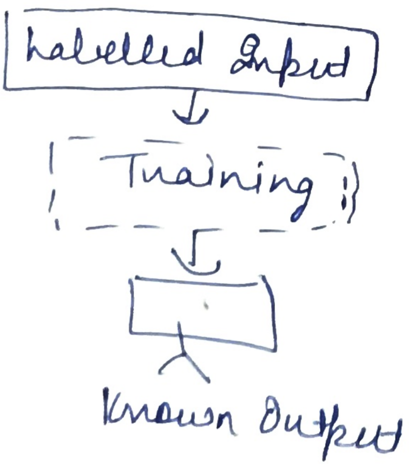
→ Types of Data: Labelled Data



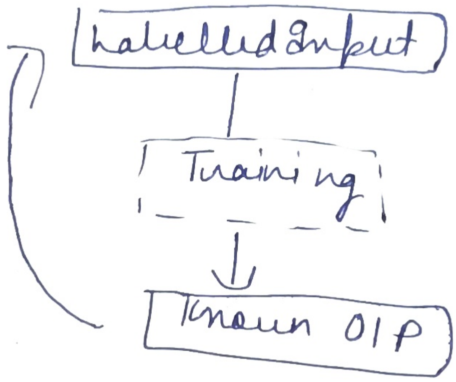
→ Training - External Supervision

→ Aim - Forecast outcomes

→ Approach - Map labelled input to known output



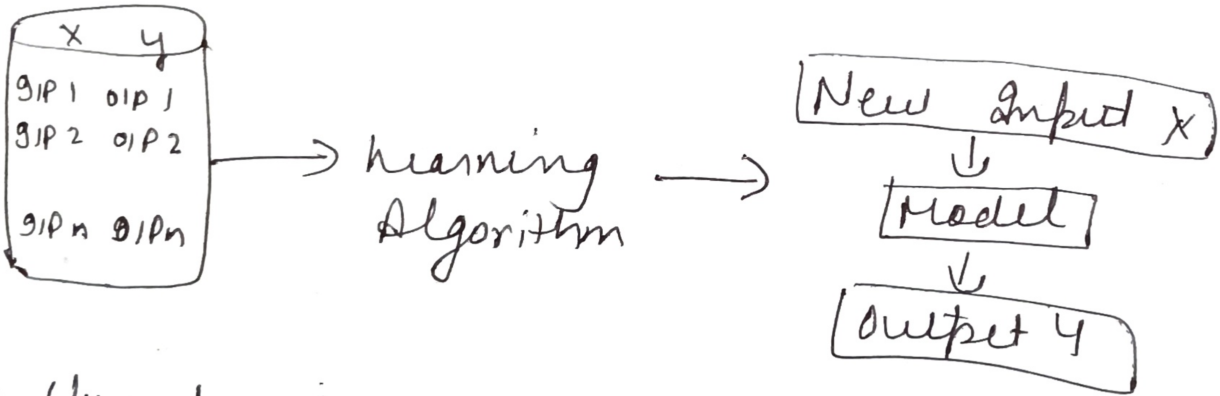
→ Output Feedback – Direct feedback



→ Popular Algorithms → Linear Regression, Logistic, SVM, K Nearest Neighbours, Random Forest.

→ Applications → Risk evaluation, Forecast sales

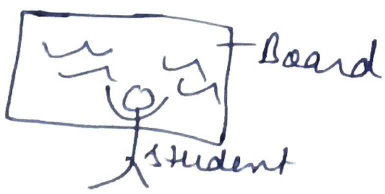
Block Diagram



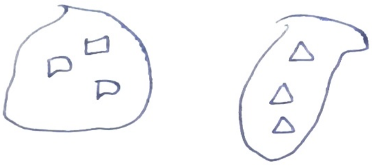
2. Unsupervised Learning

→ I am self sufficient in learning.

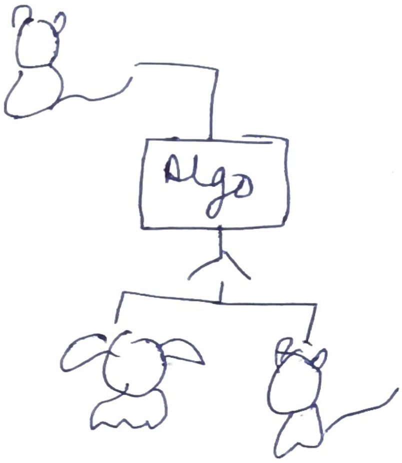
→ In U.S learning, the machine is trained on unlabelled data without any guidance



→ Problem Type: Association

→ Clustering 

→ Type of data – Unlabelled Data



→ Training – No supervision
→ Aim – Discover underlying patterns
→ Approach – Understand patterns & discover output

Unlabelled Input
↓
Explore patterns & trends
↓
Algo
↓
Output

→ Output Feedback → No Feedback

Unlabelled Input
|
Explore pattern & trends
|
O/P

→ Popular Algorithms — K-Means, Apriori, C-Means

→ Applications — Recommendations, Anomaly Detection

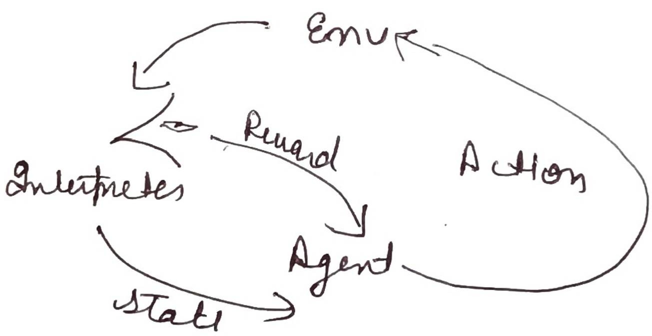
3. Reinforcement Learning

→ My Life My rules! (Hit & Trial)

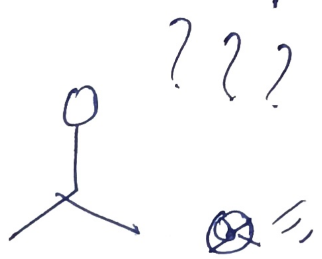
→ In R.L, an agents interact with its env by producing actions & discovers errors or rewards.

→ Problem Type — Maze Game

→ Type of Data — No Predefined Data



→ Training — No supervision

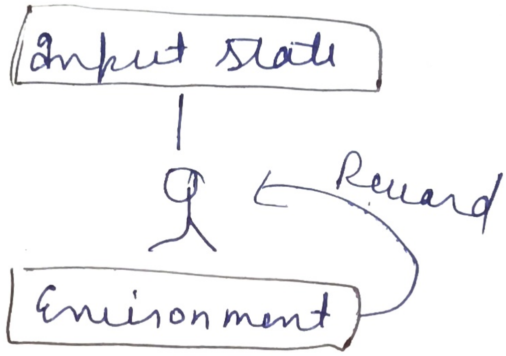


→ Aim — Learn series of Action

→ Approach — follow Trial & Error Method

— Maze Game

→ Output Feedback – Reward System



— Popular Algorithms – Q-learning, SARSA

– Applications – Self driving cars, Gaming

Well-Defined Learning Systems

→ A well defined learning problem is easy to define and model.

→ Main features :
- class of Task (T)
- Performance Measure (P)
- Training Experience (E)

The following are the examples of some well-defined learning problems.

1. Checkers (Chess) game learning problem.
2. Handwriting recognition " "
3. Robot car driving " "
4. Spoken words recognition " "
5. New astronomical structure " ".

Checkers Game Learning Problem

1. Class of Task (T) = Playing checker (How to play)
2. Performance Measure (P) = % of game won against problem.
3. Training Experience (E) = Playing practice games against itself.

Handwriting Recognition Learning Problem

For example, identify a handwritten character "6" using image machine learning & image processing as
- Task (T) = Recognizing & classifying handwritten words from given images.
- Performance Measure (P) = % of words correctly classified
- Training Experience (E) = A database of images of handwritten words.

eg: rupee note, name
VERNIKA

6 → $\boxed{\text{Image Recognition}}$ → Labels
[1
...
9]

Robot Car Driving Learning Problem

1) Task (T) = Driving on a 4-lane highway using vision sensors.

2) Performance Measure (P) = Average distance travelled before an error.

3) Training Experience (E) = A sequence of images & steering commands recorded while observing a human driver.

- GPS
- Wheel Encoder
- Ultrasonic sensors
- Radar
- Camera
- Lidar.

Step2: Choosing the target function:

What type of knowledge is learnt and how it is used by the Performance System.

Example:- Checkers game

→ While moving diagonally Set of all possible moves is called legal moves

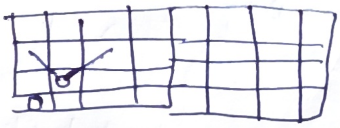
↓

One move

↓

target move

- Travel only in forward dir.
- Only one move per chance
- only in diagonal direction
- Jump over Opponent



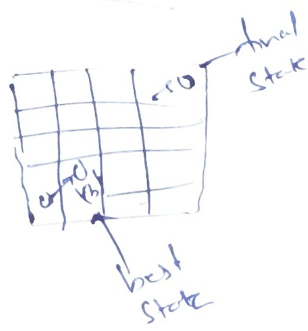
Target function ⇒ V(b)

Board state ⇒ b

Legal moves set ⇒ B

1. b is final board state that is won, then V(b) = 100
2. b is final board state that is lost, then V(b) = −100
3. b is final board state that is draw, then V(b) = 0
4. If b is not final State then V(b) = V(b')

b' → best final state



Step 3. Choosing a representation for target function

For any board state (b), we calculate function "c" as linear combination of following board features i.e. c(b)

→ features: $x_1 - x_6$

x_1 – No of black pieces on board

x_2 – No of red pieces on board

x_3 – No of black kings on board

x_4 – No of red kings on board.

x_5 – No of black pieces threatend by red (blacks which can be beaten by red)

x_6 – No of red pieces threatened by black.

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

or c(b) (representation of c(b))

$w_0 - w_6$) = numerical coefficients (or) weights of each feature,

↓

determine the importance / weightage of features

w_0 → additive constant.

Step 4: Choosing a learning algorithm for approximating the target function

To learn a target function (f) we need a set of training examples (describe a particular board state (b) and training value $V_{train}(b)$)

Ordered pair = (b, $V_{train}(b)$)
(training example representation)

Eg. black won the game (i.e. $x_2 = 0$, which means no red

x_1 = no of black pieces on board
x_2 = no of red pieces

$$V_{train}(b) = +100$$

$$b = (x1 = 3, x2 = 0, x3 = 1, x4 = 0, x5 = 0, x6 = 0)$$

~~We need to do 2 steps in this phase~~

$$< (x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0) +100 >$$

We need to do 2 steps in this phase

(1) Estimating training value

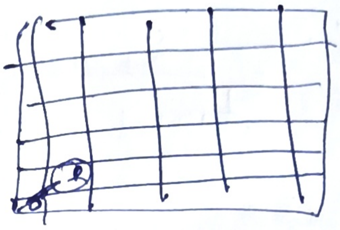
In every step, we consider successor (depending on the next step of opponent)

$$V_{train}(b) \leftarrow \hat{V}(Successor(b))$$

↓ represents the next board state

(estimating that this move will help / destroy opponent)

$\hat{V} \rightarrow$ represent approximation



② <u>Adjusting the weights:</u>

There are some alg. to find weights of linear functions, Here we are using LMS (least mean square)

(used to minimise the error)

$$Error = \left(V(train(b)) - \hat{V}(b)\right)^2$$

If error = 0, no need to change weights

error is positive, each weight is increased in proportion

error is negative each weight is decreased in proportion example: While playing chess with the opponent, when opponent will play then the machine learning algorithm will decide what be the number of possible legal moves taken to get success.

Step 3- Choosing Representation for Target function: When the machine algorithm will know all the possible legal moves the next step is to choose the optimized move using any representation i.e. using linear Equations, Hierarchical Graph Representation, Tabular form etc. The NextMove function will move the Target move like out of these moves which will provide more success rate. For Example: while playing chess machine have 4 possible moves, so the machine will choose that optimized move which will provide success to it.

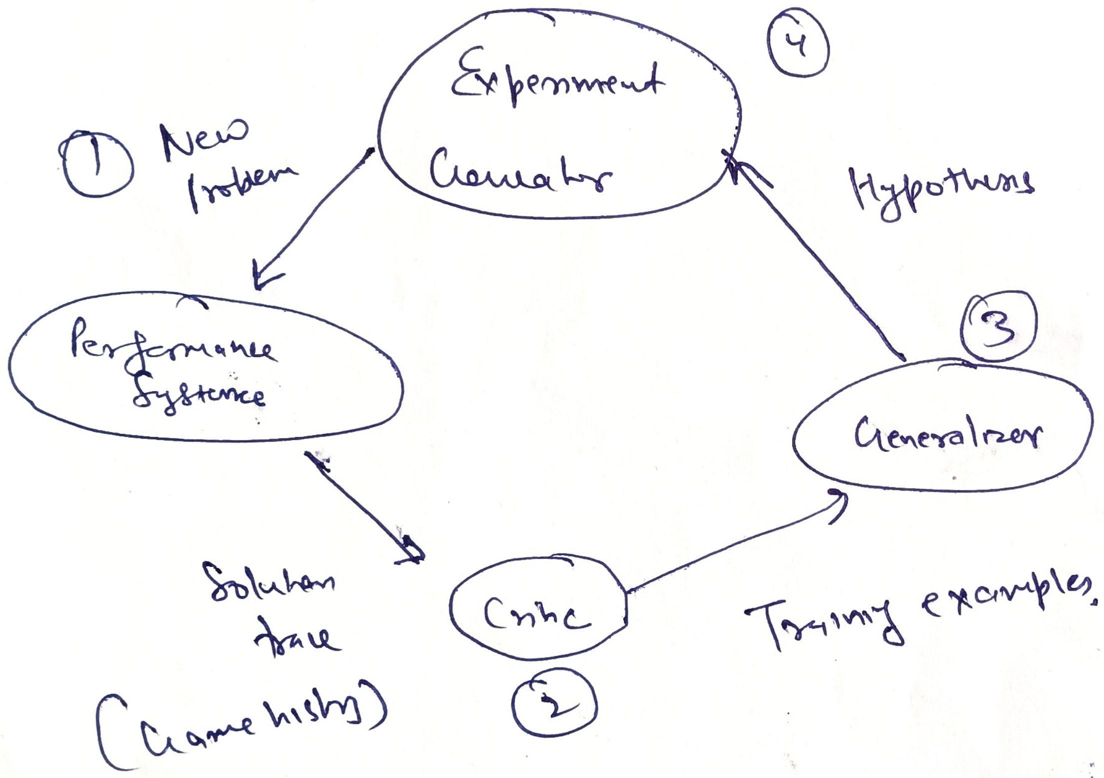
Step 4- Choosing Function Approximation Algorithm: An optimized move cannot be chosen just with the training data. The training data had to go through with set of examples and through these examples the training data will approximates which steps are chosen and after that machine will provide feedback on it. For Example: When a training data of Playing chess is fed to algorithm so at that time it is not machine algorithm will fail or get success and again from that failure or success it will measure while next move what step should be chosen and what is its success rate.

Step 5- Final Design: The final design is created at last when system goes from number of examples, failures and success, correct and incorrect decision and what will be the next step etc. Example: DeepBlue is an intelligent computer which is ML-based won chess game against the chess expert Garry Kasparov, and it became the first computer which had beaten a human chess expert.

FINAL DESIGN

— has 4 different moduls,

1. Performance System
2. Critic.
3. Generalizer
4. Experiment Generator



HISTORY OF MACHINE LEARNING 1950 — Alan Turing creates the "Turing Test" to determine if a computer has real intelligence. To pass the test, a computer must be able to fool a human into believing it is also human.

1952 — Arthur Samuel wrote the first computer learning program. The program was the game of checkers, and the IBM computer improved at the game the more it played, studying which moves made up winning strategies and incorporating those moves into its program.

1957 — Frank Rosenblatt designed the first neural network for computers (the perceptron), which simulate the thought processes of the human brain.

1967 — The "nearest neighbor" algorithm was written, allowing computers to begin using very basic pattern recognition. This could be used to map a route for traveling salesmen, starting at a random city but ensuring they visit all cities during a short tour.

1979 — Students at Stanford University invent the "Stanford Cart" which can navigate obstacles in a room on its own.

1981 — Gerald Dejong introduces the concept of Explanation Based Learning (EBL), in which a computer analyses training data and creates a general rule it can follow by discarding unimportant data.

Machine Learning (source: Shutterstock)

1985 — Terry Sejnowski invents NetTalk, which learns to pronounce words the same way a baby does.

1990s — Work on machine learning shifts from a knowledge-driven approach to a data-driven approach. Scientists begin creating programs for computers to analyze large amounts of data and draw conclusions — or "learn" — from the results.

1997 — IBM's Deep Blue beats the world champion at chess.

2006 — Geoffrey Hinton coins the term "deep learning" to explain new algorithms that let computers "see" and distinguish objects and text in images and videos.

2010 — The Microsoft Kinect can track 20 human features at a rate of 30 times per second, allowing people to interact with the computer via movements and gestures.

2011 — IBM's Watson beats its human competitors at Jeopardy.

2011 — Google Brain is developed, and its deep neural network can learn to discover and categorize objects much the way a cat does.

2012 — Google's X Lab develops a machine learning algorithm that is able to autonomously browse YouTube videos to identify the videos that contain cats.

2014 – Facebook develops DeepFace, a software algorithm that is able to recognize or verify individuals on photos to the same level as humans can.

2015 – Amazon launches its own machine learning platform.

2015 – Microsoft creates the Distributed Machine Learning Toolkit, which enables the efficient distribution of machine learning problems across multiple computers.

2015 – Over 3,000 AI and Robotics researchers, endorsed by Stephen Hawking, Elon Musk and Steve Wozniak (among many others), sign an open letter warning of the danger of autonomous weapons which select and engage targets without human intervention.

2016 – Google's artificial intelligence algorithm beats a professional player at the Chinese board game Go, which is considered the world's most complex board game and is many times harder than chess. The AlphaGo algorithm developed by Google DeepMind managed to win five games out of five in the Go competition.

Issues in Machine Learning and How to solve them

1.Lack of Quality Data:

The lack of adequate data is one of the most serious problems in Machine Learning. Algorithms often cause developers to spend the majority of their work on artificial intelligence while updating. For the algorithms to perform as intended, data quality is critical. The fundamental opponents of optimal ML are incomplete data, dirty data, and noisy data.

Noisy data, dirty data, and incomplete data are the quintessential enemies of ideal Machine Learning. The solution to this conundrum is to take the time to evaluate and scope data with meticulous data governance, data integration, and data exploration until you get clear data. You should do this before you start.

2.Implementation Problems

When companies opt to upgrade to machine learning, they frequently use examination engines to help them. It's a difficult task to combine newer machine learning algorithms with old operations. Maintaining proper documentation and interpretation will go a long way toward ensuring maximum utilization.

A few of the reasons or ways implementation problems can be caused are, lack of sufficient data, data security issues, and slow deployment.

3.Change in algorithm with growth in data

When being trained, ML algorithms will always demand a large amount of data. ML algorithms are frequently trained on a certain data index and then used to predict future data, a cycle that can only be expected with a large amount of work.

At a moment where the data arrangement changes, the prior "correct" model over the data set may no longer be regarded as accurate.

4. Wrong assumptions are being made

Missing data points are impossible for machine learning algorithms to handle. As a result, highlights that include a significant amount of missing data should be removed. On the other hand, rather than removing an element with a few missing attributes, we may fill those empty cells. The best way to cope with these challenges in Machine Learning is to guarantee that your data is free of gaps and can express a significant amount of information.

5.Lack of Skilled Resources

Another difficulty with Machine Learning is that deep analytics and machine learning in their current forms are still a relatively young technology.

Machine Learning professionals are necessary to maintain the process from the start coding to the maintenance and monitoring. The fields of artificial intelligence and machine learning are still relatively new to the market. It's also tough to find enough resources in the form of labor. As a result, there is a scarcity of capable representatives to design and handle scientific ingredients for ML. Data scientists frequently require a mix of spatial knowledge as well as a thorough understanding of mathematics, technology, and science.

5.Identifying Which Processes Should Be Automated

In today's world of Machine Learning, separating reality from fiction is getting increasingly challenging. You should analyze whatever challenges you're trying to tackle before deciding on which AI platform to utilize. The operations that are done manually every day with no variable output are the easiest to automate. Before automating complicated procedures, they must be thoroughly inspected. While Machine Learning may certainly aid in the automation of some processes, it is not required for all automation concerns.

6.Segmentation of User

Consider the data of a user's human behavior throughout a testing period, as well as any relevant prior habits. All things considered, an algorithm is required to distinguish between clients who will convert to a premium version of a product and those who will not.

Based on the user's catalog behavior, a model with this choice issue would allow the software to generate suitable recommendations for the user.

- Data collection
- less amt of training data
- Non representative " "
- Poor quality of data
- irrelevant & unwanted features
- ~~off~~ lack of skilled resources
- Complexity
- Maintenance
- High chance of error & wrong assumption

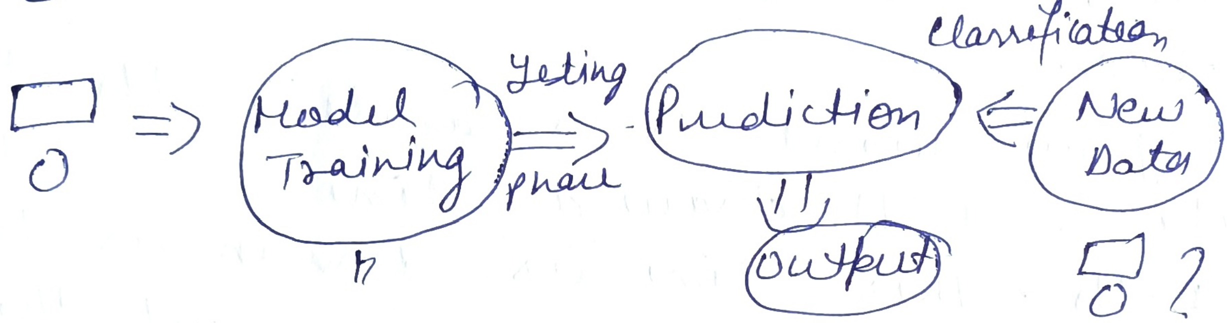
SUPPORT VECTOR MACHINE

→ SVM lies in the supervised category.

→ This is a kind of supervised learning used for classification and regression.

→ Here, we are going to use SVM for classification

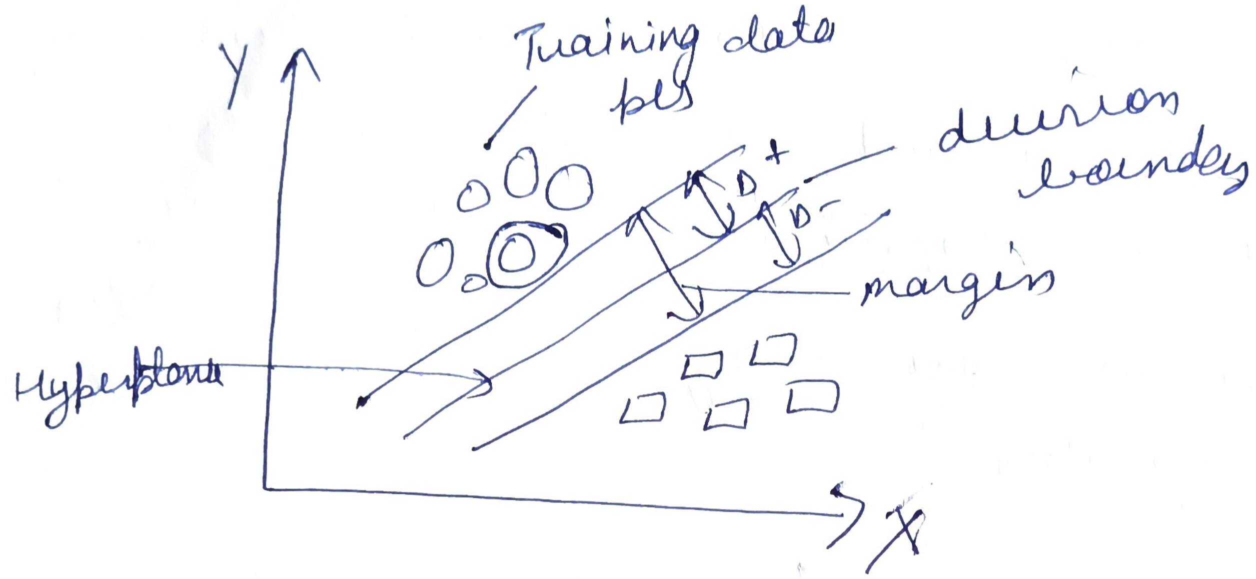
quad
circles } Training data > labelled



a) The data is given for model training & when the data is trained it is passed to predication phase for testing whether the model is trained right or wrong.

b) So, when the test or new data is passed to prediction phase it has to predict the correct data for en: if circle is passed to prediction phase & it classify as circle class so our model is trained right.

→ Now, let's see that in 2D :-



→ Here, we have 2 classes: circle & quad class.

→ If we want to separate both the classes I'll draw a line in between both. This line is known as 'decision boundary' as this decides whether the test data belongs to circle class or quad class

1 → The circle class is considered class & quad is opponent class.

2 → We will search a point in circle class which is nearer to opponent class.

3 → Draw a line parallel to hyperplane near closest pt of circle class.

4 → Repeat step 1, 2, 3 for quad class i.e. considered class now.

→ Now, due to 2 lines drawn we have 2 distance a) D^+ → distance to the +ve pt
b) D^- → " " -ve pt.

→ Sum of both the distance leads to one more distance called as 'margin'.

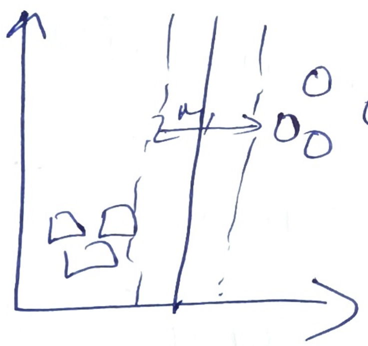
$$Margin = D^- + D^+$$

→ What is support vector?

Ans: The 2 pts which were considered to draw parallel lines to the decision boundary these are called as support vector.

Types of SVM

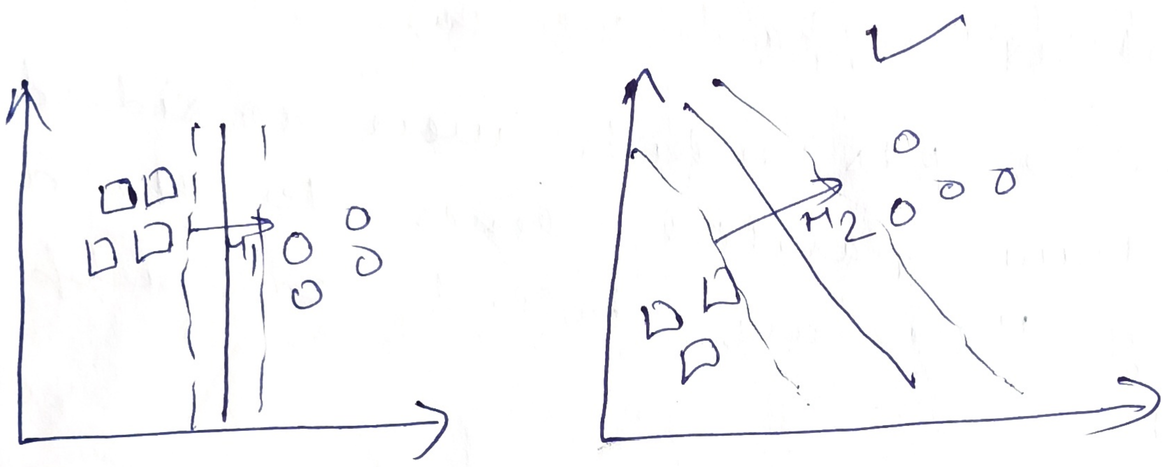
① Linear SVM



→ Just with a single line drawn btw 2 classes can separate the classes
This is the linear line.

2) Non Linear SVM

→ If we draw a single line we can't separate the data, accuracy will be 50%. This is misclassification.



→ $M_2 >> M_1$

→ Acc to standards, to classify any data choose the max. width.

→ Here, max width is M_2 so we will use this to classify data points

→ Max width = future prediction better

→ "Maximal Margin Hyperplane" should get selected

a) accuracy high

b) error rate low

Non - Linear SVM

→ when a given sample data can't be divided into 2 classes is known as non-linear data.

→ A single line can't separate the data

Here, we will use **Kernel function**.

LD low Dimensional feature space.

Input ⇓

Kernel takes i/p LD

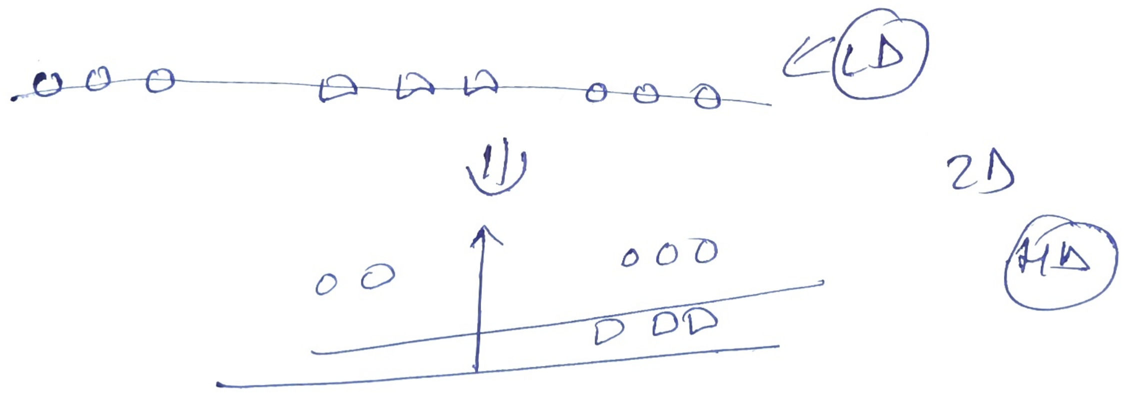
Output ⇓ gives

HD High Dim.

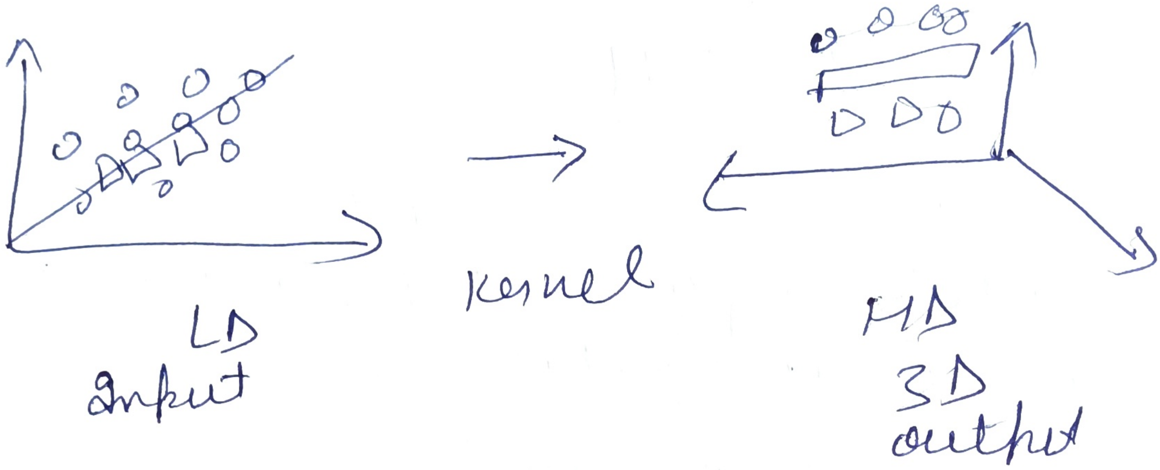
→ Kernel convert LD to HD.

→ Data is non-separable ~~in~~ in when input is given to Kernel.

→ HD is separable to classify.

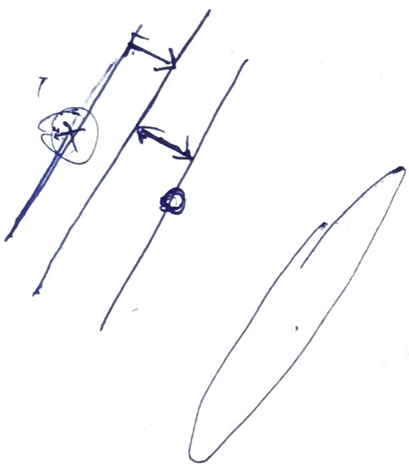


Eg:



How do you visualize Hyperplane?

→ It is Elliptical in real world.



usteing

→ This technique groups the unlabelled dataset

→ A way of grouping the data points into different clusters, consisting of similar data points.

→ The objects with possible similarities remain in a group that has less or no similarities with another group.

→ 

shopping centres

→ Eg: Amazon, netflix in its recommendation system to provide recommendation as per the past search of products, movies respectively.

Types of clustering techniques.

a) Partitioning

b) Density based

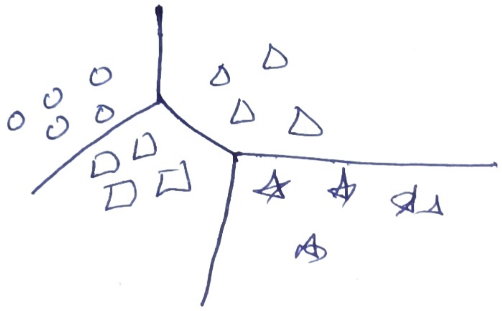
c) Distribution model based

d) Hierarchical

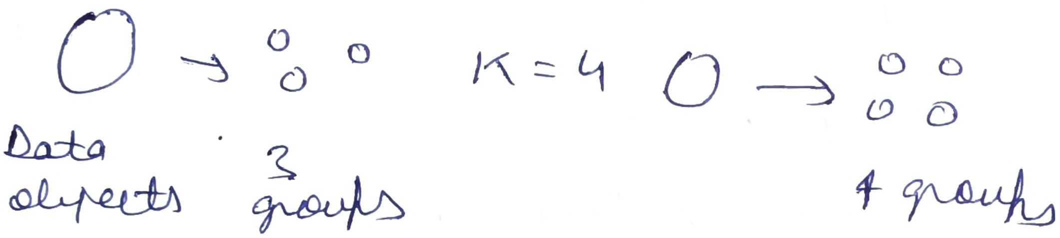
e) Fuzzy

(1) Partitioning clustering

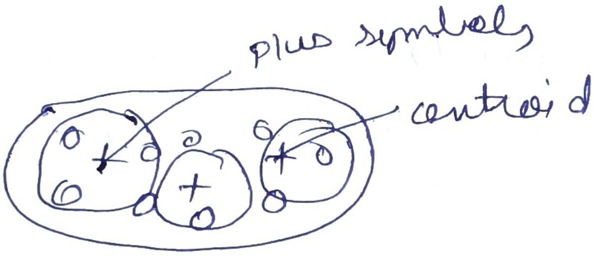
→ It divides the data into non-hierarchical groups. It is also known as centroid based method. · Dataset is divided into set of K groups
Eg: K-means Eg
& defin
no of predefined groups.



→ No of clusters is known as K = 3



→

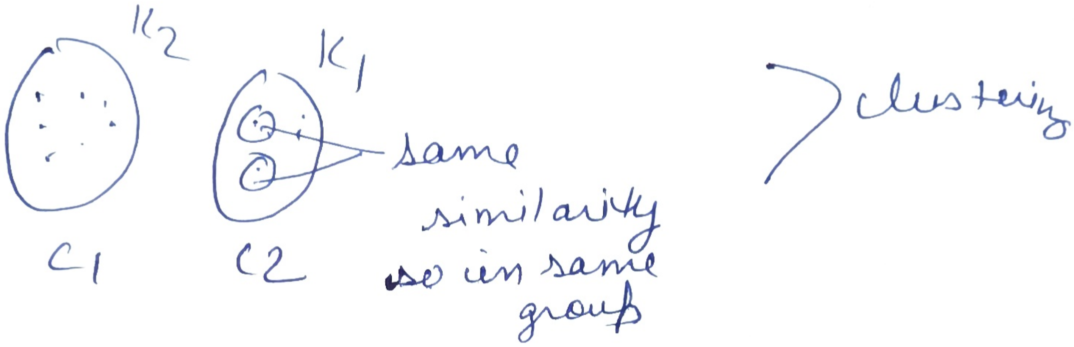


Around ~~Data~~ a centroid data will cluster as a single group or cluster.

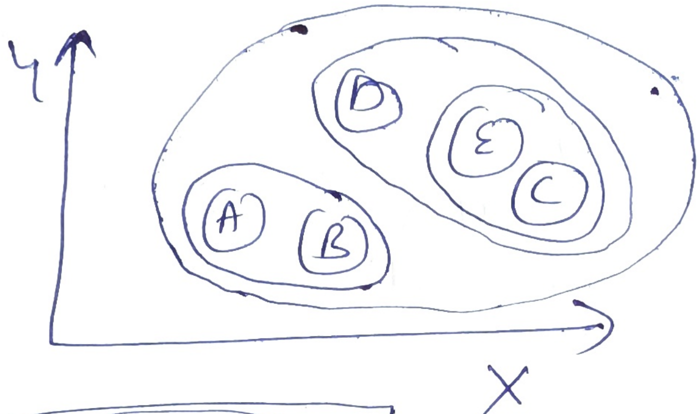
→ Hard clust. → D.P belong to only 1 group
→ Soft clustering " " another group

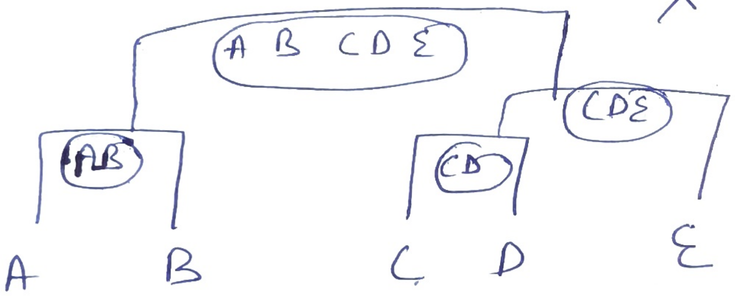
Fuzzy Clustering is a type of soft method in which a data object may belong to more than 1 group or clusters. Each dataset has a set of membership coefficient which depend on the degree of membership to be in cluster

hierarchial clustering



→ Data points: A B C D E





Hierarchy of clusters

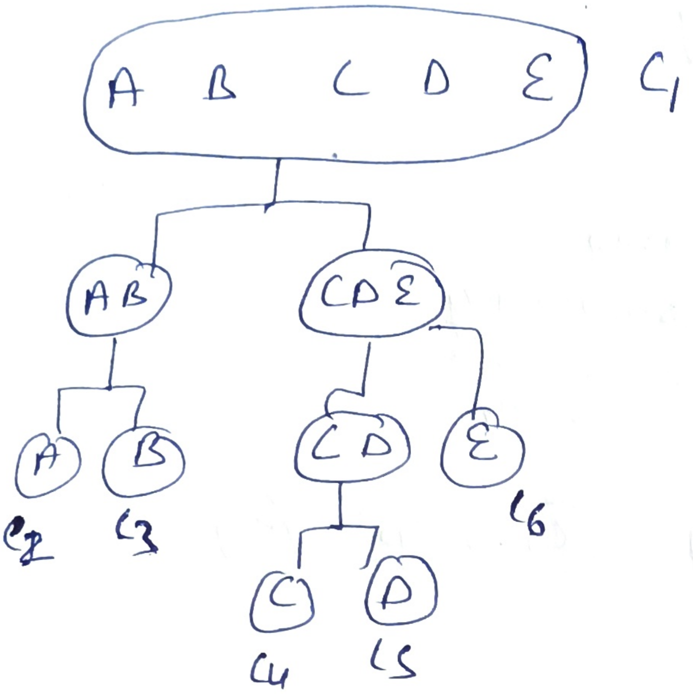
→ Agglomerative : Bottom up Approach
start with bottom

→ Dendrogram (Tree like structure) is used to represent this hierarchial clustering

→ Divise : Top down Approach

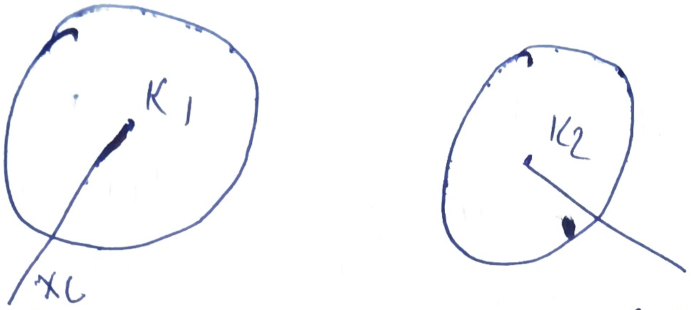
Divis ma

Divide into no clusters



K-means Clustering

	(X) Height	(Y) 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	70



(185, 72)
1st row

(170, 56)
2nd row

Euclidean Distance

$\sqrt{(x_o - x_c)^2 + (y_o - y_c)^2}$

3rd row $\;$ x_o (168, 60)

ED for (3) — $K_1 \rightarrow \sqrt{(168-185)^2 + (60-72)^2}$

$= 20.80$

$K_2 \rightarrow \sqrt{(168-170)^2 + (60-56)^2}$

→ 3rd row will go to cluster K_2 as $= 4.48$ its value is less than K_1

New centroid Calculation

$$\text{for } K_2 = \left(\frac{\overset{x_c + x_o}{170+168}}{2}, \frac{\overset{x_c + x_o}{60+56}}{2} \right) = (169, 58)$$

(185, 72) $\qquad$ (169, 58)

(179, 68)

E.D ~~for~~ $K_1 = \sqrt{(179-185)^2 + (68-72)^2} = (6.32)$

$K_2 = \sqrt{(179-169)^2 + (68-58)^2} = 14.14$

$K_1 = \{1, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

$K_2 = \{2, 3\}$

- we have to form 2 clusters K_1 & K_2
- to By using centroid concept → has cluster se ek centre value hogi jis centre value k basis pe we will group other atoms in the cluster.

- we have to use rest 10 rows as they belong to which cluster by using ED.
- ED is calculated btw observed value & centroid value
- The rest 10 value will be observed value & centroid value is K_1 & K_2 value
- * calculate new centroid after every row put in the cluster.
- K_2 ~~row~~ values was put in cluster K_2 so, will calculate new centroid for K_2

$$\text{New centroid } K_2 = \frac{\text{old centroid} + \text{New centroid}}{2}$$

$$= \frac{170 + \overset{3^{rd}\ \text{row}}{168}}{2} + \left(\frac{60+56}{2}\right)$$

Genetic Algorithm

Based on

Adaptive heuristic Algo

evolutionary Algo

Genetics & natural selection

To Generate High Quality soltn for optimization problem

Population & Individual

operator of GA

Encoding | Mutation | Selection | crossover

→ Adaptive means adapt the no of parameters given to the algo on change in the env.

→ Genetic → Genes, phenotype

→ natural selection → evolution, fitness value, score, offspring, survival of the fitted.

→ Individual is considered as a feasible solution for the problem

→ collection of Individual forms a population. Such a population of individual is maintained in search space.

→ selection → select the fittest individual.

crossover → reproduction for new offspring

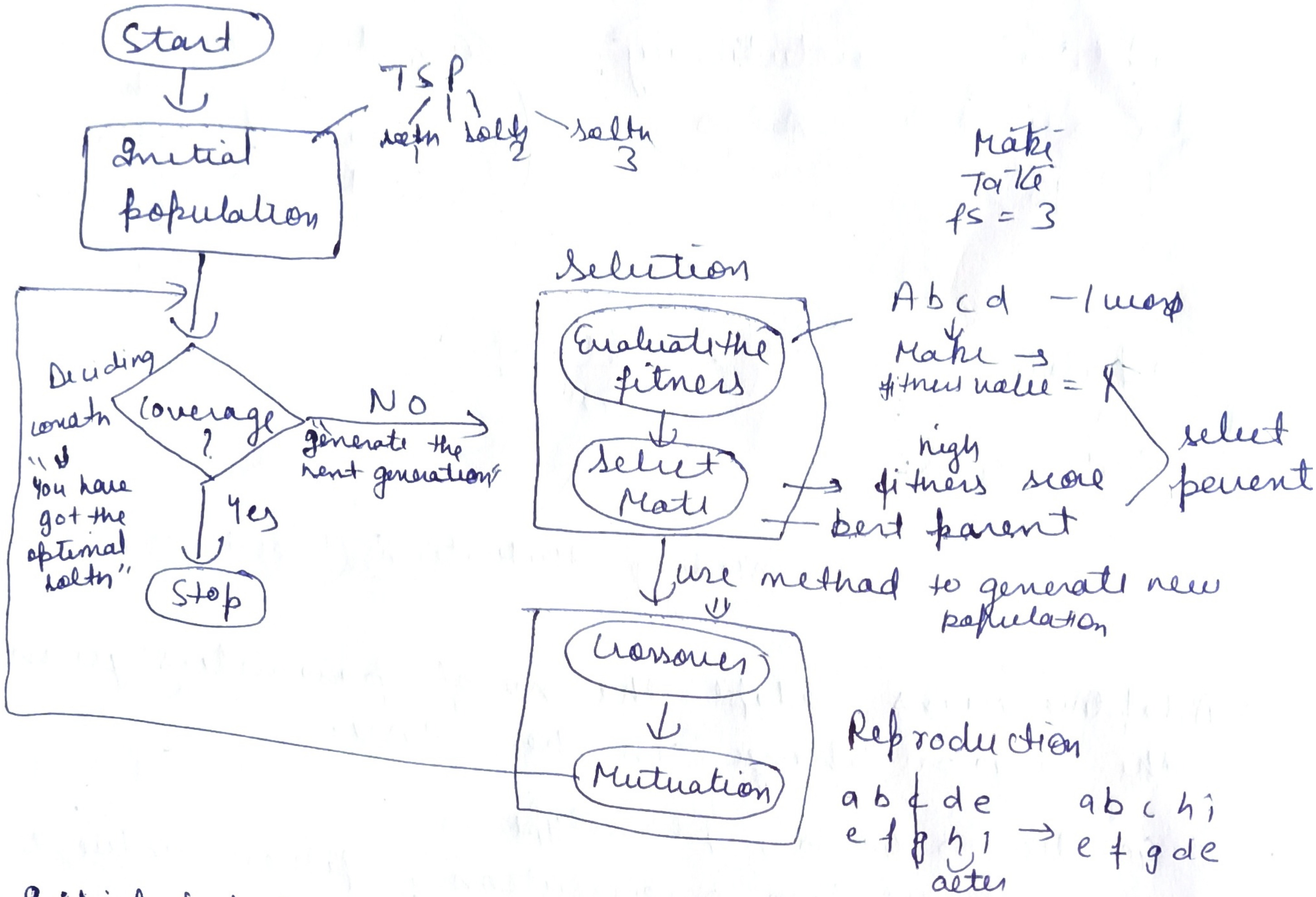
mutation → alter the genes

Encoding → used in initial population

OUTPUT → if we generated new population as @gh & old population was calc@ = so 1 match so, we have to generate O/P with high fitness score as earlier population was also 1 & 3.

→ a k m e →4
b a m e →4 stop

Flowchart of Genetic Algorithm



→ Initial population is current generation so that can produce new generation. (Diversity is there eg. no particular age group)

→ I.P can be a random solution.

→ Evaluate the fitness score of every individual.

→ Crossover → Use 2 parents to generate offspring.

→ Mutation → (Swap, 1bit flip) Genes of generated offspring are altered we get new population that transferred to the termination condtn & loop starts

Important points

→ John Holland gave the idea of G.A

→ Abstraction of Real biological evaluation

(Study of genes, chromosomes, 2 parents create a child & pass their best to newer generation, best genes travel to next to next generation)

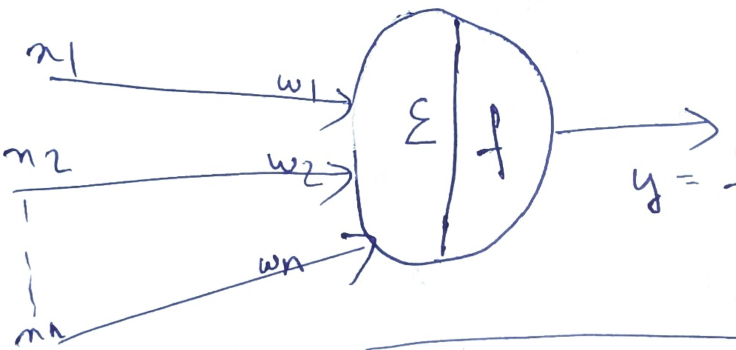
→ focus on optimization
(we have soltn already we have to find the best soltn from all of them)

→ population of possible solutions for a given problem
(we have complex problems here like NP hard, TSP to find the best soltn)

→ From a group of individual (chromsomes) the best will survive (most fit will survive)

→ phenotype $\xrightarrow{\text{encode}}$ Genotype
~~(soltn space in real life~~ raw soltn) $\xleftarrow{\text{Decode}}$ represent in a soltn space

Artificial Neural Networks



$y = f(\Sigma x_i w_i)$

$$\boxed{x_1 w_1 + x_2 w_2 + \cdots x_n w_n}$$

→ We have millions of neurons in our brain.

→ 'Neuron' is the basic unit of brain or smallest undividable unit.

→ Functionality of 'Neuron' is through our sensory organs like 'eyes, nose, ears' it takes the input in the brain using neurons → process it → interpret & then provide the appropriate solution(OUTPUT)

→ when we try to achieve this functionality artificially then it comes under the category of 'ANN'.

→ How does this node works?

$\Sigma \mid f$

node divided into 2 parts 'replica of neuron'
a) solution part
b) function part.

(when you touch something hot your skin receptors sense something is hot & heat level is going beyond threshold value. So, they think now we have to give this info. to brain & brain 'll take the action otherwise hand will get burn. So, sensory organs generate signals & send to brain. Brain collect input signals then process it & take appropriate action that put your hand away otherwise it 'll get burn (OUTPUT))

→ x_1 — signal
x_2 — signal — lot of signals — respective weight w_1
x_n — signal — w_2
— w_2

Σ → calculated weighted sum
$\boxed{x_1w_1 + x_2w_2 + \dots x_nw_n}$

send this to ↓

activation function → provide particular input to activation fn then through this fn you generate app. output or action. In a way, A.F generates particular output for a given node based on the input that is provided.

~~Naive Bayes~~

Decision Tree

Age	competition	Type	Profit
old	yes	S/W	Down
old	No	S/W	Down
old	No	H/W	Down
mid	yes	S/W	Down
mid	yes	H/W	Down
mid	No	H/W	up
mid	No	S/W	up
new	yes	S/W	Up
new	No	H/W	up
new	No	S/W	up

$$I.G = -\frac{P}{P+N} \log_2\left(\frac{P}{P+N}\right) - \frac{N}{P+N} \log_2\left(\frac{N}{P+N}\right)$$

$$E(A) = \sum_{i=1}^{v} \frac{P_i + N_i}{P+N} I(P_i N_i)$$

$$\text{gain} = IG - E(A)$$

$$\boxed{\log_2 x = \frac{\log_{10} x}{\log_{10} 2}}$$

Age:-

	Down	Up
old	3	0
mid	2	1
new	0	3

P = down = 5
N = up = ~~3~~
P + N = Total entries = 10

$I(old) = -[\frac{3}{3}\log_2(\frac{3}{3}) + \frac{0}{3}\log_2(\frac{0}{3})] = 0 \times 3/10 = 0$

respective I.G

$I(mid) = -[\frac{2}{4}\log_2(\frac{2}{4}) + \frac{2}{4}\log_2(\frac{2}{4})] = 1 \times 4/10 = 0.4$

$I(new) = -[\frac{0}{3}\log_2(\frac{0}{3}) + \frac{3}{3}\log_2(\frac{3}{3})] = 0 \times 3/10 = 0$

$E(Age) = 0.4$

$I.G = -[\frac{5}{10}\log_2(\frac{5}{10}) + \frac{5}{10}\log_2(\frac{5}{10})]$

('-' minus common lelenge '+' andar chla jaega)

$= -[0.5 \times \log_2 2^{-1} + 0.5 \log_2 2^{-1}]$

$= -[0.5 \times (-1\log_2 2) + 0.5 \times (-1 \log_2 2)]$

$= -[-0.5 - 0.5] = -[-1]$

$\log 0.5 = \lg 2^{-1}$
$\log_2 2 = 1$

I.G = 1
T.A

Gain = 1 − 0.4
= 0.6

→ Make a tree that will ~~help~~ support us to take some decision.

→ Dataset is given with 4 attributes & 10 entries (based on these entries make a decision tree)

Step 1: Choose one attribute that will be called as 'target attribute'.

→ Profit is T.A.

- Step 2: Calculate the information gain of T.A.

Step 3: Find the entropy of "Age, Comp-t, Type".
(one of these attributes will be the root of our decision tree).

Entropy of an attribute = I.G of that particular attribute into probability of that attribute.

Step 4: Find the Gain.
(~~Gain~~ Information of T.A − E of Age

- To calculate Entropy make a table first of that attribute

E(Age) → Possible outputs of T.A (column)

Down Up

→ possible outputs of undertaken attribute (Row)

old
mid
new

→

	Down	Up
old	3	0
mid	2	2
new	0	3

I.G =

I (old) = − [$\frac{3}{3}$ + $\frac{0}{3}$

3 — P = old kitni bar down hai
3 — Total count of old in dataset
0 — old 0 bar up hai

= 0 × $\frac{3}{10}$

0 ↓ I.G of old

3 — probability ' old occurred three times
10 — total (P) = 10

→ we will calculate Gain of Competition & Type.

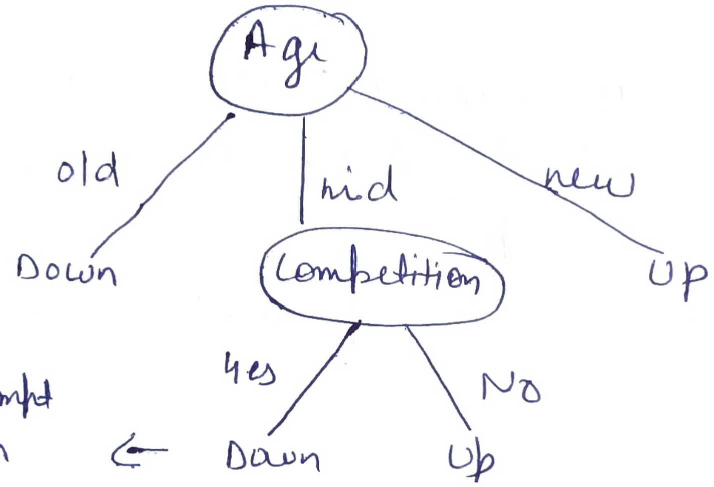
→ Gain (Age) = 0.60

Gain (Compt) = 0.124

Gain (Type) = 0.

→ $\boxed{I.G = 1}$ (Target Atribute of profit I.G = 1)

→ Highest gain will be the root node of D.T.



both yes in Compt
in mid are ~~Yes~~ down ←

→ Mid is compt. as some are down & some as Up.

→ So, we will add another node in Mid.

→ Take another node 'which has next highest Gain'

→ Gain of Type = 0

no need for Type node, as Yes D if Yes D then
Yes D Yes U

we might have taken Type node.

Important Points

① A classifier (Tree structured)
- Decision Node (Test)
- Leaf Node (classification value)
- Also used for classification & Regression)
- Test is performed on the feature / Attribute (value)

② D.T is a classifier in classification
(model)

Ⓓ $\xrightarrow{I/P}$ (Alg) $\xrightarrow{O/P}$ classifier generate / model (M/C) — model or This tree structured.

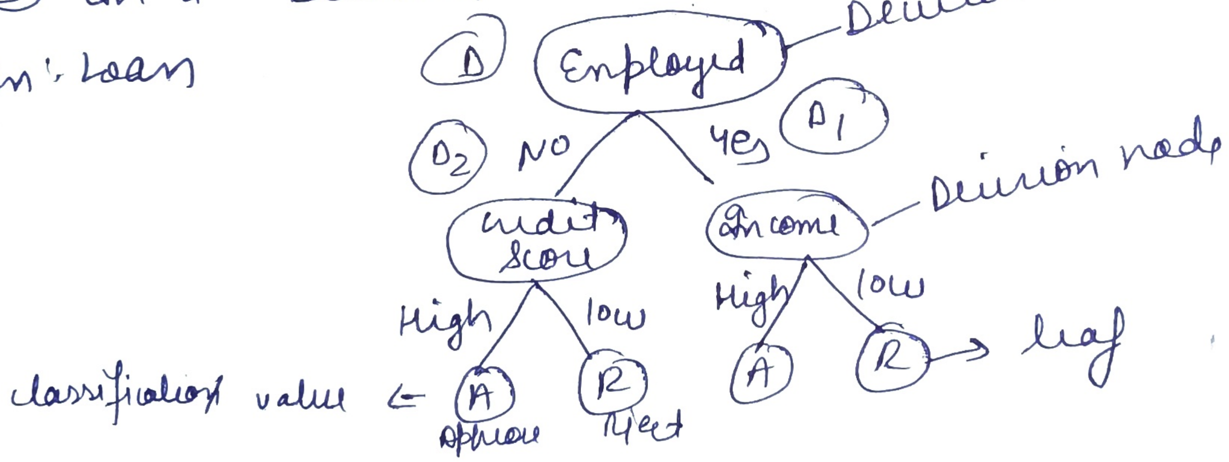
Training Dataset given to algo — classifier

What is the use of this classifier?

→ I/P sample unlabelled classifier (rules, mechanism) < it'll tell this sample belongs to which class.

③ In a Decision Tree: — Decision node are test

Ex: Loan



Significance of D.N & L.N

→ D.N are test, where test have 2 ans. Yes or No

→ Branches indicate the test (Yes / No)

→ Test karne k lie jo b lgne wala hai wo features hai.

Significance of L.N

→ L.N have values / classification (Accept / Reject) — class

Dataset

features considered

Employed	Credit score	Income
Y		
Y		
Y		
N		
N		
N		

$\}\ D_1$ Splitting

$\}\ D_2$

→ entire Dataset is given to root node.

→ Splitting action → Depending on the result of the test dataset is split.