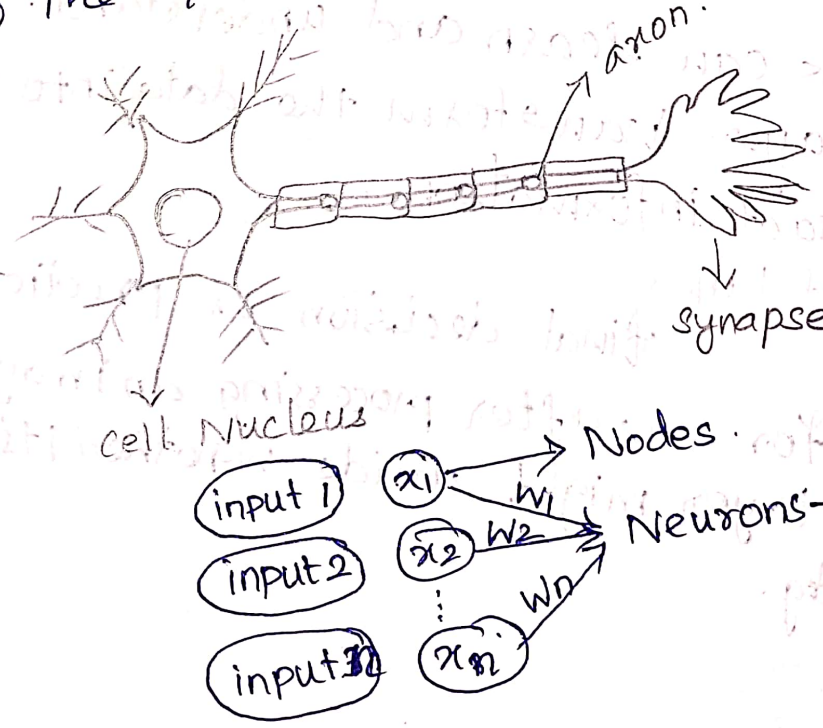
UNIT - 5

NEURAL NETWORKS.

Perceptron - Multilayer perceptron, activation function, n/w training - gradient descent optimization - stochastic gradient descent, error back propagation, from shallow n/w to deep n/w - unit saturation (vanishing gradient problem) - ReLU, hyperparameter tuning, batch normalization, regularization, dropout.

Neural Networks :-

The term "Artificial Neural Network" is derived from Biological Neural N/w that develop the structure of human brain. Similar to the human brain that has neurons interconnected to one another, artificial Neural n/w also have neurons that are inter connected to one another in various layer of the n/w. These neuron are known as nodes.



BNN	ANN.
Dendrities	Inputs.
cell nucleus.	Nodes.
Synapse	weights.
Axon	output.

Artificial Neural Network:-

ANN are the computing system that is designed to simulate the way the human brain analyzes and process the information.

Key Components of an ANN:-

i) input layer:-

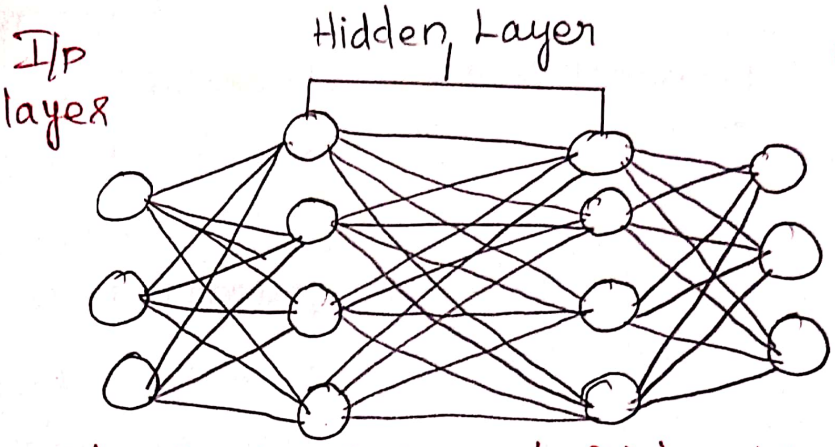
This is where the n/w receive information ex: In an image recognition task, the i/p could be an image.

ii) Hidden Layer:- These layer process the data received from the input layer. The more hidden layers.

These are more complex patterns the network can learn and understand. Each hidden layer transform the data into more abstract information.

iii) Output layers:-

final decision or prediction is made. for ex: after processing an image, the output layer might decide whether its Cat or dg.

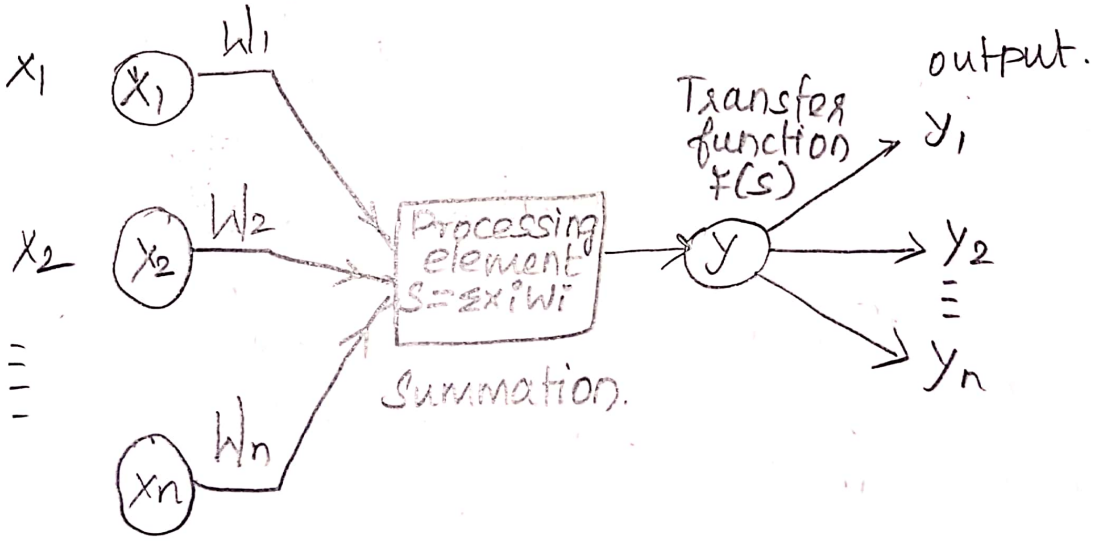


Working of Artificial Neural Network:-

Instead of directly getting into the working of Artificial Neural Networks, lets breakdown and try to understand Neural Networks basic unit, which is called perceptron.

Perceptron can be defined as a neural network with a single layer that classifies the linear data. It further constitutes of 4 major components.

i) Inputs (ii) Weights and Bias.
iii) Summation function (iv) Activation or transformation function.



Perceptron:-

Perceptron is Machine Learning algorithm for Supervised learning of various binary classification tasks.

Perceptron is also understood as a artificial neuron or neural n/w unit that helps to detect certain i/p data computation in business intelligence.

Perceptron model is also treated as one of the best and simplest types of ANN. It is a supervised learning algorithm of binary classifiers. Hence we can consider it as a single layer neural n/w with four main parameters.

* i/p values, * weight / Bias * Netsum
* An Activation function.

Binary classifier:-

In ML Binary classifier are defined as the function that helps in deciding whether i/p data can be represented as vector of numbers and belongs to some specific class.

Binary classifier can be considered as linear classifier. In simple word, we can understand it as a classification algorithm that can Predict linear predictor function in terms of weight and feature vectors.

Perceptron function :-

Perceptron function $f(x)$ can be achieved as output by multiplying the input x with the learned weight co-efficient 'w'

$f(x) = 1$; if $w \cdot x + b > 0$.

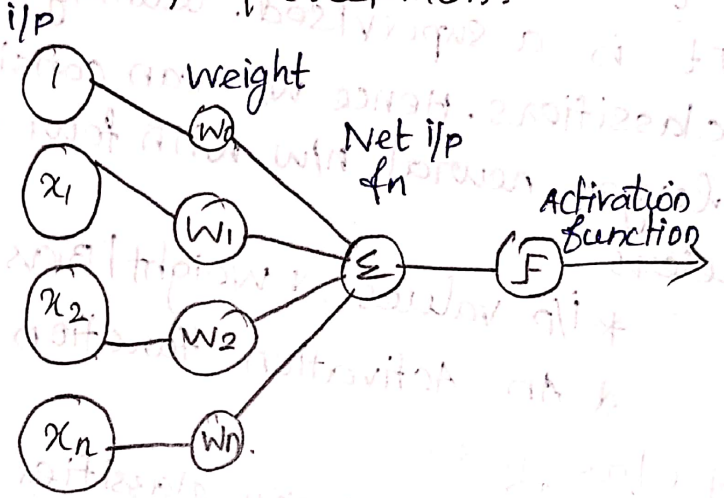
otherwise $f(x) = 0$.

w represent real valued weights vector.

b represent the bias.

x represent a vector of i/p x values.

Components of perceptron :-



Input nodes or Input layers :-

This is primary component of perceptron which accepts the initial data into the s/m for further processing. Each i/p node contains a real numerical value.

Weight and Bias :-

weight parameter represent the strength of the connection b/w units. This is another most important parameter of Perceptron. Components. weight is directly proportional to the strength of the associated input neuron is

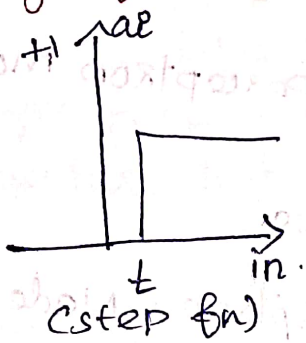
deciding the output. Further, Bias can be considered as the line of intercept in a linear equation.

Activation function:-

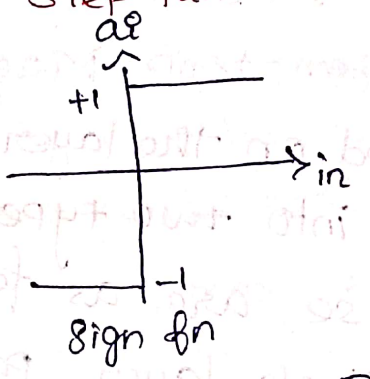
These are final & important components that help to determine whether the neuron will fire or not. Activation function can be considered primarily as a step function.

Types of Activation function:

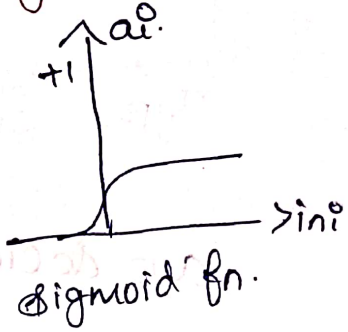
Sign function



Step function

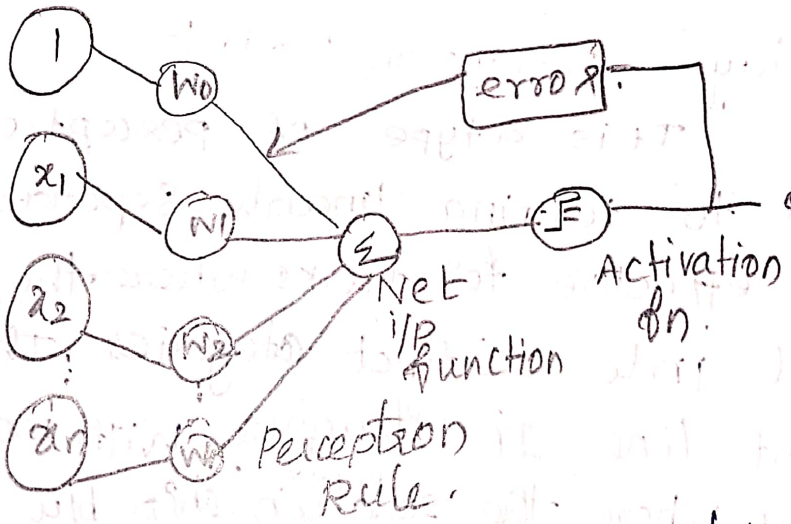


Sigmoid function.



Relu - overcom vanishing gradient problem to avoid.

Working of Perceptron:-



Perceptron model works in two important.

Steps:-

Step1:- In 1st step multiply all i/p values with corresponding weight values and then add them to determine the weighted sum. Calculates the weighted sum as follows.

$\sum w_i * x_i = x_1 * w_1 + x_2 * w_2 + \dots w_n * x_n$

Add a special term called bias 'b' to this weighted sum to improve the models performance.

$\sum w_p * x_i + b$

Step 2:-

In the second step, an activation functions is applied with above-mentioned weighted sum, which gives us output either in binary form or a continuous value as follows.

$$y = f\left(\sum w_i * x_i + b\right)$$

Types of perceptron Models:-

Based on the layer, perceptron models are decided into two types.

These are as follows.

i) Single layer perception Model

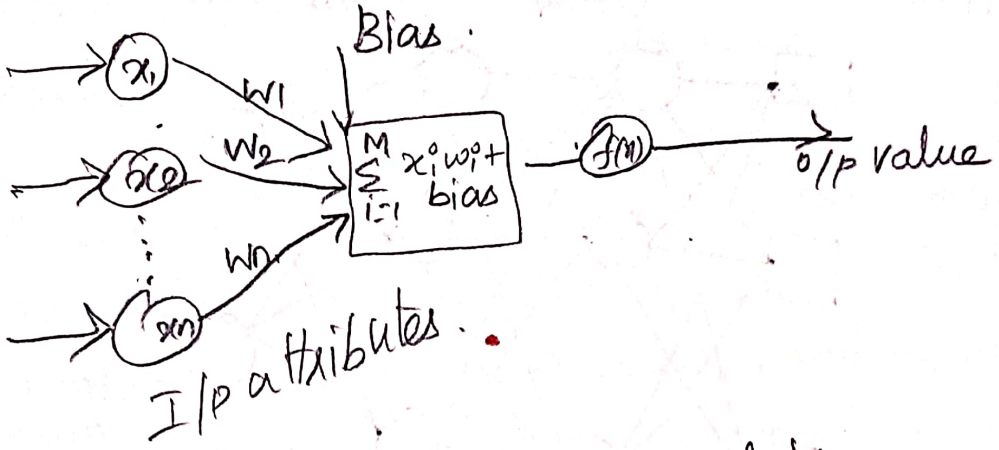
ii) Multi layer Perceptron Model.

Single layer perceptron Model:-

It is a type of perceptron is limited to learning linearly seperable patterns. It is effective for tasks where the data can be divided into distinct categories through a straight line. It struggles with more complex problems where the relationship b/w i/p & o/p is non linear.

A single layered perceptron model consist feed forward network and also include a threshold transfer function inside the model.

The main objective of the single layer Perceptron model is to analyze the linearly seperable object with binary outcomes.



Multi layered Perceptron Model:-

Like a single layer perceptron model a multi layer perceptron model also has the same model structure but has a greater number of hidden layer.

Multilayer perceptron define the most sample architecture of artificial neural network.

The multi layer perceptron model is also known as the back propagation algorithm, which executes in two stages.

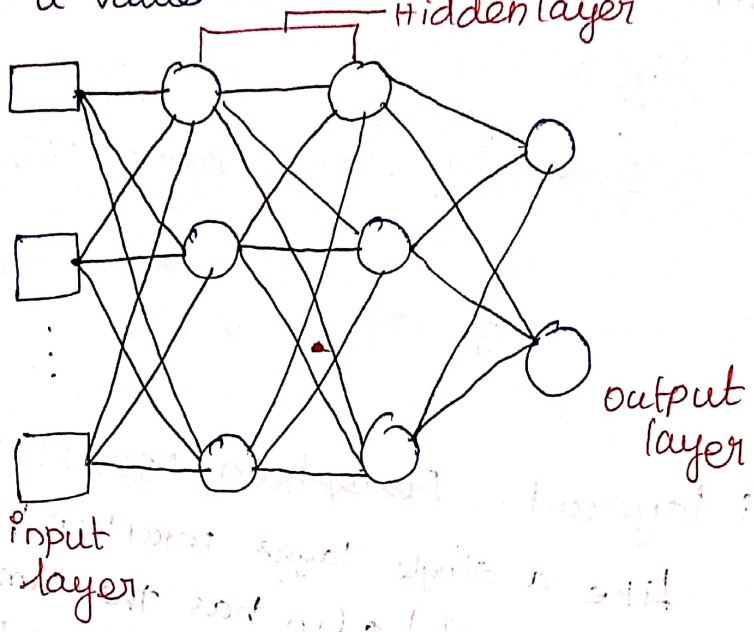
Forward stage:-

Activation function start from the input layers in the forward stage and terminate on the output layer.

Backward stage:-

In the backward stage, weight and bias values are modified as per the models requirement. In this stage the error b/w actual output and demanded originated backward on the output layer and ended on the input layer.

The i/ps are pushed through the ML
the dot product of the input with the we
exist b/w input and the hidden layer. This d
yields a value at the hidden layer.



The hidden layer neither directly receive inputs nor send output to the external environment

The final layer is the OP layer which outputs a single value or a vector of values.

MLP model has considered as multiple AMV having various layers in which activation function does not remain linear, similar to the SLP model.

Instead of linear, activation function can be executed as sigmoid, TanH, Relu etc... for development.

A Multilayer perceptron model has greater processing power and can process linear and non linear patterns. it can also implement logic gates such as: AND, OR, XOR, NAND, NOT XNOR, NOR.

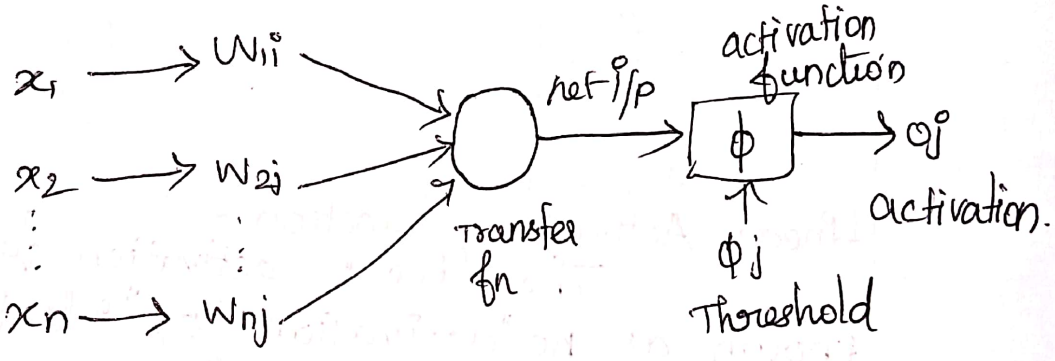
Adv
- Versatility
- Non-linearity
- Parallel Computation

DisAdv
- computationally expensive
- overfitting
- Sensitivity to data scaling.

Activation function:-

The NN Activation fn's are the most significant component of deep learning, they are fundamentally used for determining the output of deep learning models. its accuracy, and Performance efficiency of the training model that can design or divide a huge scale Neural Network.

Activation function also helps to normalize the o/p of any i/p in range b/w 1 to -1 Activation function must be efficient and it should reduce the computation time because the neural network sometimes trained on data points.



The neuron is basically a weighted average of i/p, then this sum is passed through an activation function to get an output.

$$y = \Sigma (\text{weights} * \text{i/p} + \text{bias})$$

These y can be anything for a narrow b/w range - infinity to + infinity, so we have to bound our o/p to get the desired prediction or

generalized results.

y = Activation function $\leq$ (weights * input + bias)

So we pass that neuron to activation function to bound output values.

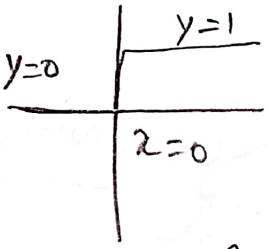
Types of Activation function :-

i) Binary step function
ii) Linear activation function
iii) Non-linear Activation function

Binary step function :-

A binary step function is generally used in the perceptron linear classifier. It threshold the input values to 1 and 0. if they are greater or less than zero respectively.

The step function is mainly used in binary classification and it can't classify the multiclass problems.



$f(x) = 1$; if $x > 0$

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

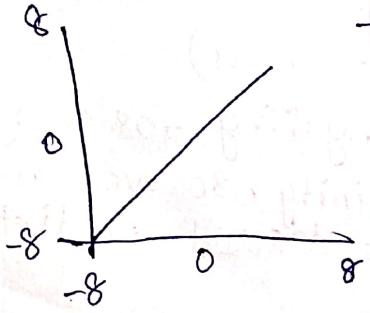
Linear Activation function :-

The linear activation function, also known as no "activation" or "identity function" (multiplied $x_1.0$) is where the activation is proportional to the input.

The equation for linear activation function

$f(x) = a.x.$

When $a=1$ then $f(x) = x$ and this is a special case known as identity.



Linear Activation Function:-

Modern neural networks models use non-linear activation functions. They allow the model to create complex mappings b/w the n/w i/p & o/p such as images video, audio and data set that are non-linear or have high dimensionality.

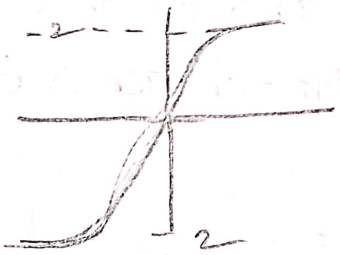
Three types of non linear Activation function

1) Sigmoid Activation Function
2) ReLu (Rectified Linear units)
3) Complex Non-linear Activation function.

Sigmoid Activation function:-

It is characterized by s shape. This function takes by any real value as i/p & o/p values in the range of 0 to 1 hence useful for binary classification.

The function exhibits a steep gradient when x values are b/w -2 and 2.



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

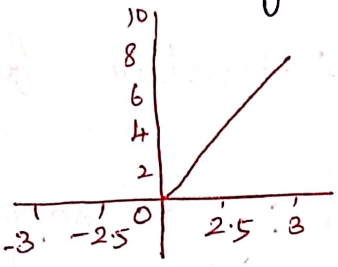
ReLu:- Rectified Linear unit function:-

ReLu activation fn means that if the i/p x is positive, ReLu returns x, if the i/p is negative, it returns 0.

Value range [0, ∞] meaning the function only o/p non-negative values.

Nature:- It is a non-linear activation function, allowing neural n/w to learn complex patterns and making backpropagation more efficient.

Relu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the n/w sparse making it efficient and easy for computation.



$$f(x) = \max(0, x).$$

Tanh function (Hyperbolic Tangent)

Tanh function is very similar to the sigmoid and logistic activation function and even has the same S-shape with the difference in output range of -1 to 1.

Tanh fn is a shifted version of the sigmoid, allowing it to stretch across the y-axis It is defined as.

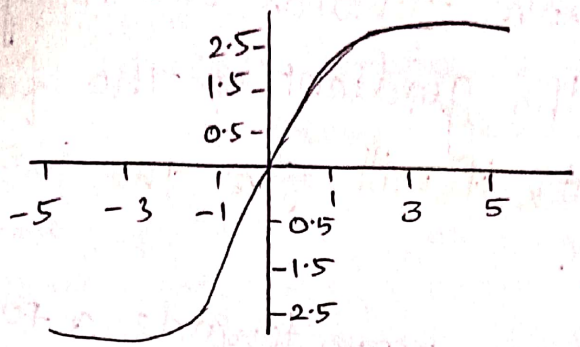
$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

It can be expressed using the sigmoid function.

$$\tanh(x) = 2 \times \text{sigmoid}(2x) - 1$$

Value Range:- outputs values from -1 to +1

Non linear :- Enable modeling of complex data patterns.



Network Training:-

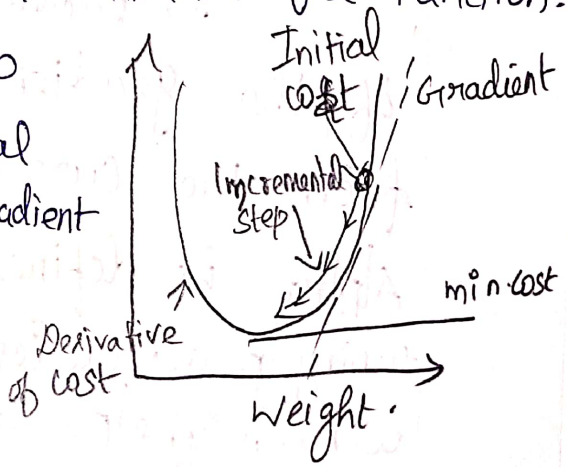
Gradient Descent optimization.

Gradient Descent is known as one of the most commonly used optimization algorithm to train machine learning models by means of minimizing. error blw actual and expected results. Further, gradient descent is also used to train Neural Network.

The main objective of gradient descent is to minimize the convex function using iteration. of parameter updates. once this ML models are optimized, these models can be used as powerful tools for AI and various computer science application

Gradient Descent is defined as one of the most commonly used iterative optimization. algorithms of ML to train the ML and DL models. It helps in finding the local minimum of a function.

The best way to define the local minimum or local maximum of a function using gradient descent is as follows.



If we move towards a -ve or away from the gradient of the current point, it will give the local minimum of that function.

Whenever we move towards a positive gradient or towards the gradient of the function at the current point, we will get the local maximum of that function.

Gradient Ascent :-

This entire procedure is known as gradient Ascent, which is also known as steepest descent. The main objective of using a gradient descent algm is to minimize the cost function using iteration.

To achieve this goal, it performs two steps :-

→ Calculate the first-order derivative of the function to compare the gradient or slope of that function.

→ Move away from the direction of the gradient which means slope increased from the current point by alpha times, where Alpha is defined as learning rate. It is a tuning parameter in the optimization process which helps to decide the length of the steps.

Cost function:-

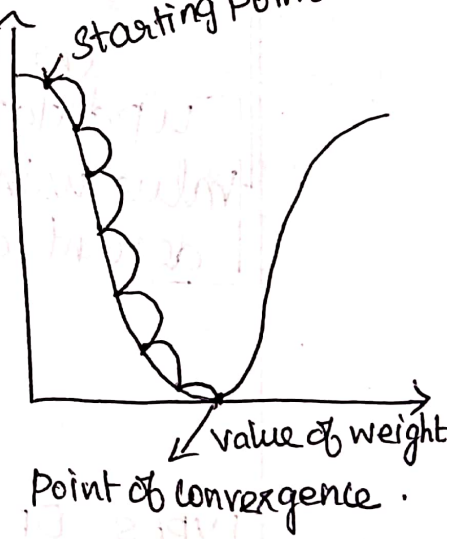
The cost function is defined as measurement of difference or error between actual values and expected values at the current position and present in the form of a single real number.

Working of Gradient Descent:-

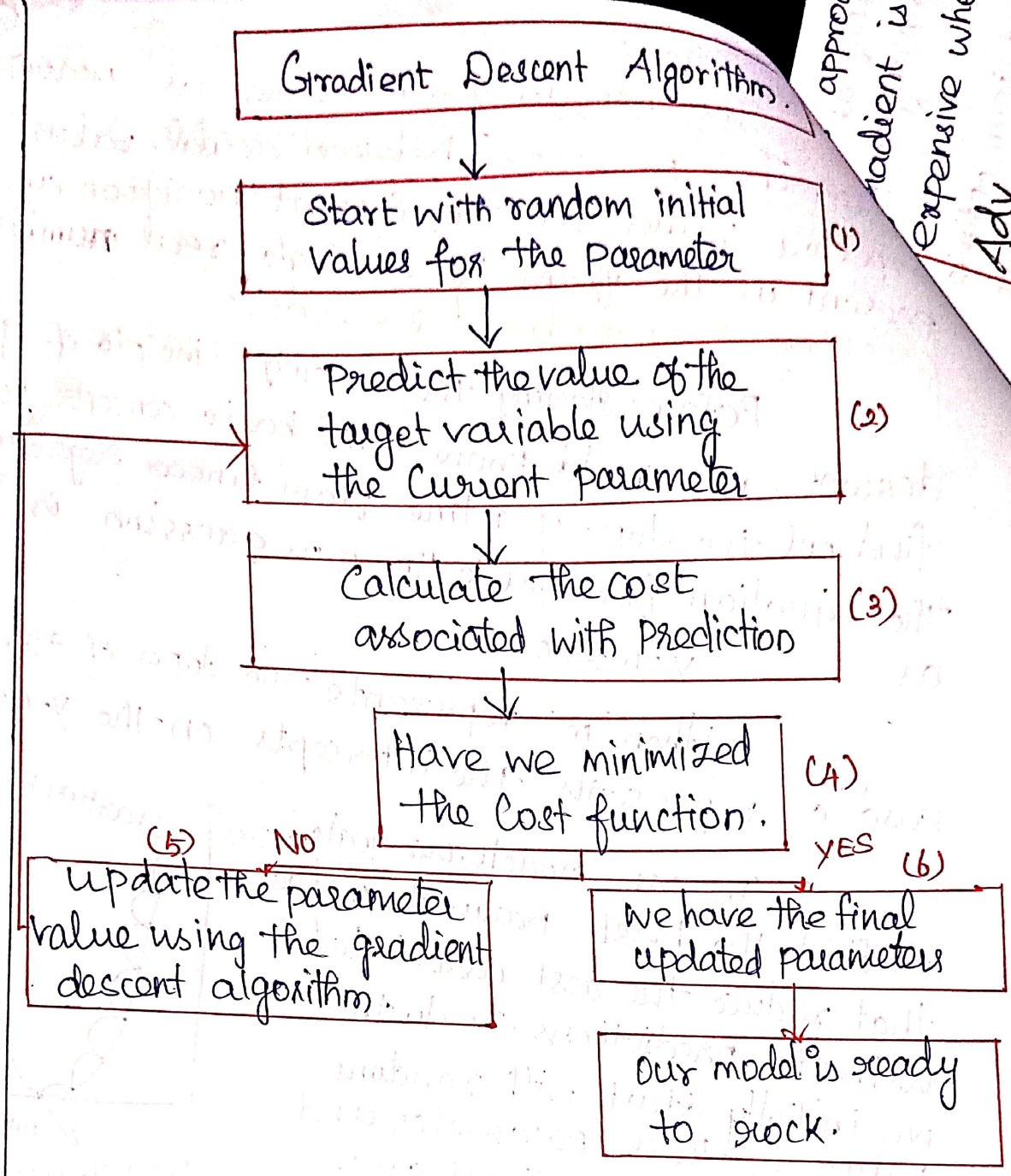
Before starting the working principle of gradient descent, we should know some basic concepts to find out the slope of a line from linear regression. The equation for simple linear regression is given as $y = mx + c$

When 'm' represents the slope of the line, and 'c' represents the intercepts on the y-axis.

In ML models, our goals is to find the best parameter values that reduce the cost associated with the predictions. To do this, we initially start with random values of these parameter and try to find the optimal ones. To find optimal values.



We use the gradient descent algorithm.



TYPES OF GRADIENT DESCENT:-

Based on the error in various training models, the Gradient descent learning algorithm can be divided into Batch gradient descent Stochastic Gradient descent, and mini-batch gradient descent.

1) Batch Gradient Descent:-

Batch gradient descent computes the descent gradient of the cost function using the entire training dataset for each iteration.

is approach ensures that the computed gradient is precise but it can be computationally expensive when dealing with very large datasets.

Adv
- → Accurate Gradient estimates.
- → Good for smooth error surfaces.

DisAdv:
- Slow convergence
- High memory usage
- Inefficient for large datasets.

2) Mini Batch Gradient Descent:-

Mini Batch Gradient Descent combines concepts from both batch gradient descent and Stochastic gradient descent. It splits the training dataset into small batch sizes and perform updates on each of these batches. This approach strikes a balance btw the computational efficiency of batch gradient descent and the speed of Stochastic gradient descent.

Adv
- Faster Convergence
- Reduced memory usage
- Smoother gradient estimation
- Parallelization & speed.

Dis adv.
- choice of Batch size
- Requires more epochs
- Complexity.

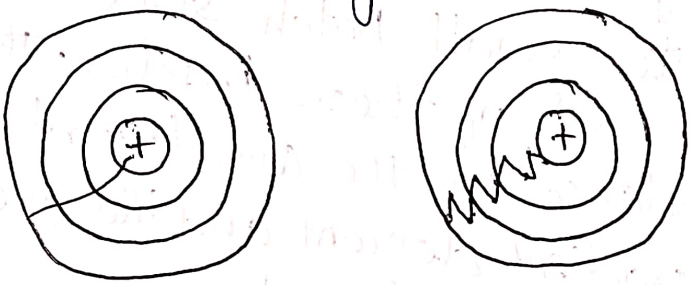
Stochastic Gradient Descent:-

Stochastic gradient descent is an optimization algorithm in machine learning, particularly when dealing with large datasets. It is a variant of the traditional gradient descent algorithm but offers

Several advantages interms of efficiency making it the go-to method for many tasks.

Stochastic Gradient Descent is an optimization algorithm that finds the set of input variables for a target function that results in a minimum value of the target function, Called the minimum of the function.

Stochastic Gradient descent : extension of the gradient descent optimization algorithm for minimizing a loss function of a particular model on a training data set.



Gradient descent Stochastic Gradient Descent.

The word stochastic means random.

In SGD, we pick a few data points randomly (instead of using all data) for each step.

It takes one training example at a time to update the model.

It's faster and needs less memory.

Because it updates often, it may help escape local minimum and reach the global one.

APP of SGD :-

Deep Learning NLP computer vision Reinforcement Learning.

Adv.	DisAdv.
Faster convergence	Noisy Gradient estimates
Lower memory requirements	Convergence issues.
Escape local minima.	Requires Stuffing.

Error Back propagation:-

Back propagation is one of the important concepts of a neural network, our task is to classify our data set. For this, we have to update the weights of parameter and bias. but how can we do that in deep neural network. In the linear regression model, we use gradient descent to optimize the parameter similarly here we also use gradient descent algorithm using Backpropagation

The main feature of Back propagation are the iterative, recursive and efficient method through which it calculates the updated weights to improve the network until it is not able to perform the task for which it is being trained. Derivative of the activation function to be known at network design time is required to Back propagation.

The Backpropagation algorithm look for the minimum value of the error function in weight space using a technique called the data rule or gradient descent. The weights that minimize the error function. is then considered to be a solution to the learning problem. The training algorithm of Back propagation involves four stages which one as follows.

Initialization of weights:-

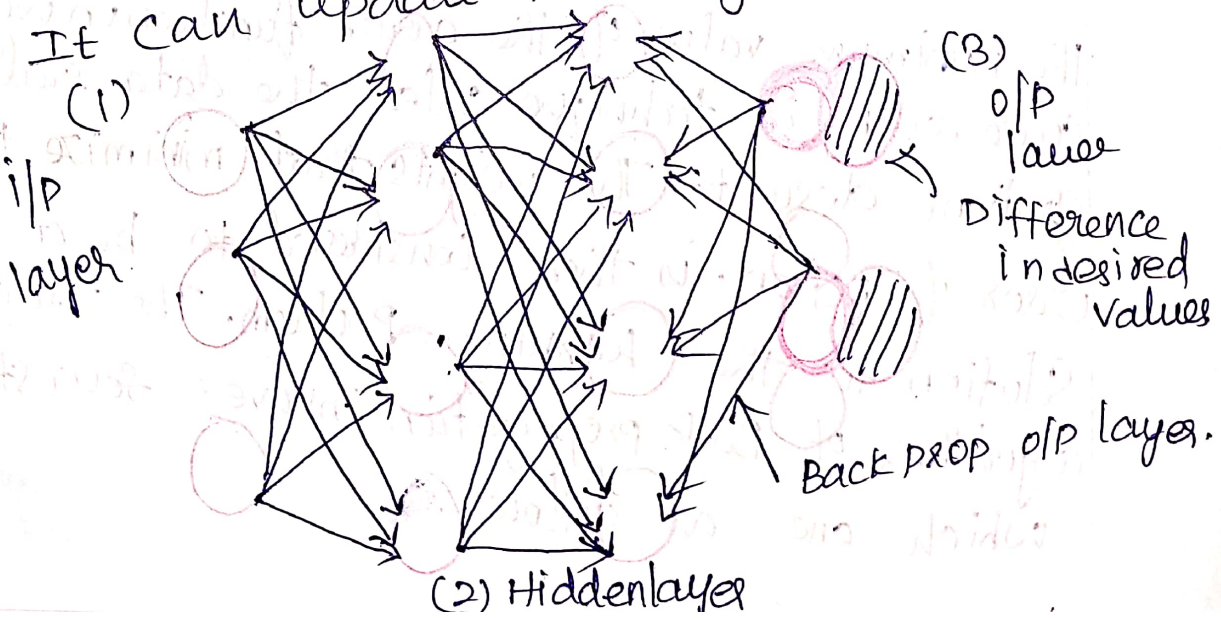
There are some small random are assigned.

Feed forward: Each unit X receives an i/p and transmit this signal to each of the hidden unit $z_1, z_2, \cdots z_n$. Each hidden unit calculate the activation function and sends its signal z_i to each output unit. The o/p unit calculates the activation function to form the response of the give input pattern.

3) Back propagation of errors:-

Each o/p unit, compare activation y_k with the target value T_k to determine the associated error for that unit. It is based on the error, the factor δ_k ($k=1 \cdots m$) is computed and is used to distribute the error at the output unit y_k back to all units in the previous layer similarly the factor δ_j ($j=1 \cdots p$) is computed for each hidden unit z_j.

4) It can update the weight & biases.



Consider the back propagation neural network example diagram to understand.

types of Backpropagation:-

(i) Static Back propagation.

(ii) Recurrent Back propagation.

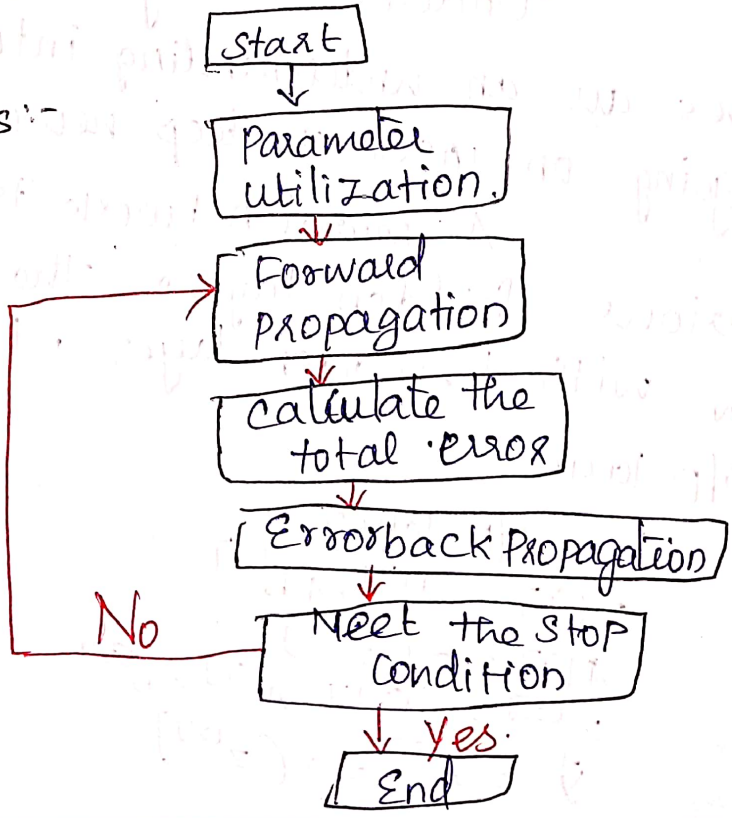
Static Back Propagation:-

In this type of backpropagation the Static output is created because of the mapping of Static input. It is used to resolve Static classification problems like optical character recognition.

Recurrent Back propagation:-

The Recurrent propagation is directed forward or directed until a specific determined value or threshold value is acquired. After the certain values, the error is evaluated and propagated backward.

How it works:-



1) i/p is modeled using real weights w. The weights are usually randomly selected.

2) Calculate the output for every neuron from the input layer, to the hidden layer, to the o/p layer.

(3) Calculate the errors in the outputs.

Error B = Actual o/p - Desired o/p.

(4) Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

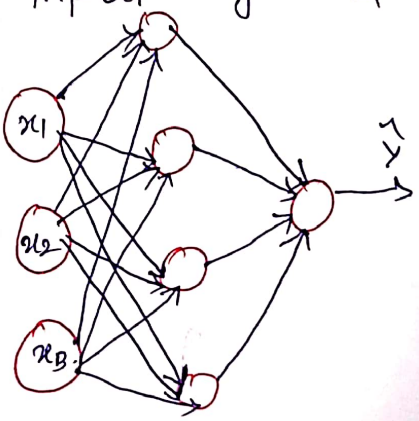
(5) Keep repeating the process until the desired o/p is achieved.

From Shallow Network to Deep Networks :-

Shallow neural networks give us basic idea about deep neural network which consist of only 1 & 2 hidden layer.

Understanding a shallow Neural Network gives us an understanding into what exactly is going on inside a deep neural network.

A neural network is built using various hidden layers. The shallow neural n/w with 1 hidden layer, 1 input layer and 1 o/p layer.



1 - o/p layer

$$z^{[1]} = w^{[1]T} x + b^{[1]}$$

$$A^{[1]} = \sigma(z^{[1]})$$

$$z^{[2]} = w^{[2]T} A^{[1]} + b^{[2]}$$

$$\hat{y} = A^{[2]} = \sigma(z^{[2]})$$

1) The 1st eqn calculates the intermediate output Z[1] of the first hidden layer.

2) The 2nd eqn calculates the final o/p A[1] of the first hidden layer

3) The third equation calculates the intermediate output Z[2] of the output layer.

4) The fourth equation calculates the final output A[2] of the o/p layer which is also the final output of the whole neural network.

Unit Saturation (i.e. the vanishing gradient problem):

The vanishing gradient problem is an issue that sometimes arises when training machine learning algorithm through gradient descent. This most often occurs in neural networks that have several neural layer such as in a deep learning system, but also occurs in recurrent neural networks.

The key point is that the calculated partial derivatives used to compute the gradient as one goes deeper into the network. Since the gradient control how much the network learns during training the gradient are very small or zero, then little to no training can take place, leading to poor predictive performance.

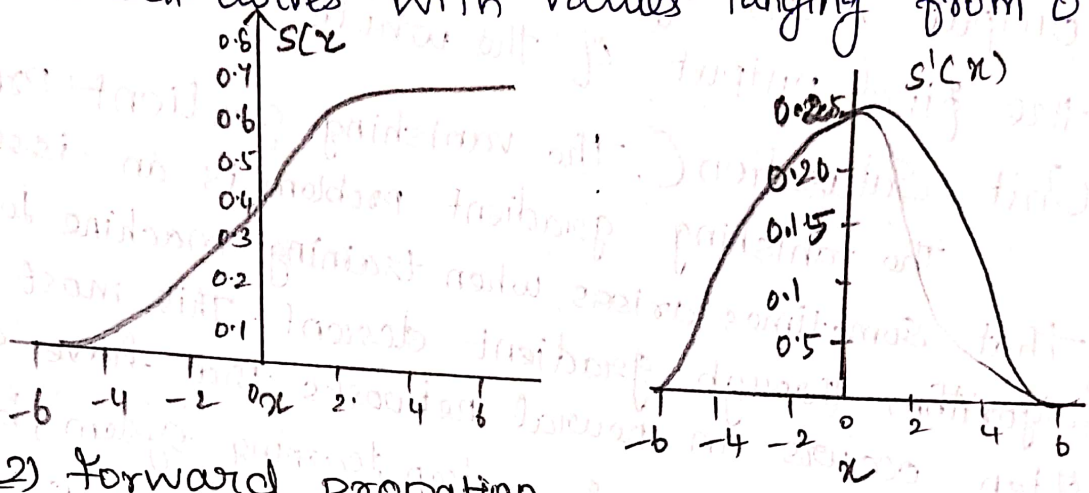
Sigmoid function:-

sigmoid function are used frequently in neural networks to activate neurons. It is a logarithmic function with a characteristics shaps. The o/p value of the functi

is b/w 0 and 1. The sigmoid function for activating the o/p layer is binary classification problems.

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

from the below graph, you can see a comparision b/w the sigmoid function itself and its derivative. First derivatives of sigmoid function are bell curves with values ranging from 0 to 25.



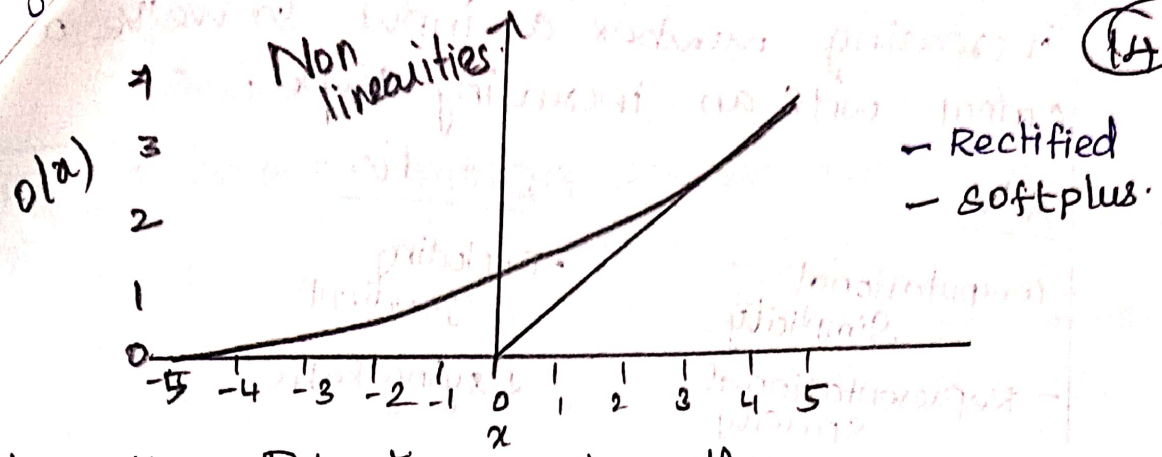
(2) forward propation
(3) Back Propagation

RELU :-

The rectified linear activation unit or ReLu. is one of the few landmarks in the deeplearning revolution. It's simple, yet its far superior to previous activation function like sigmoid or tan h.

$$f(x) = \max(0, x)$$

The diagram below with the blue line is the representation of Rectified linear unit where as the black line is a variant of ReLu includes Leakly ReLu exponential linear unit (ELU) and sigmoid linear unit (SiLU) etc..



Implementing Relu function in python.

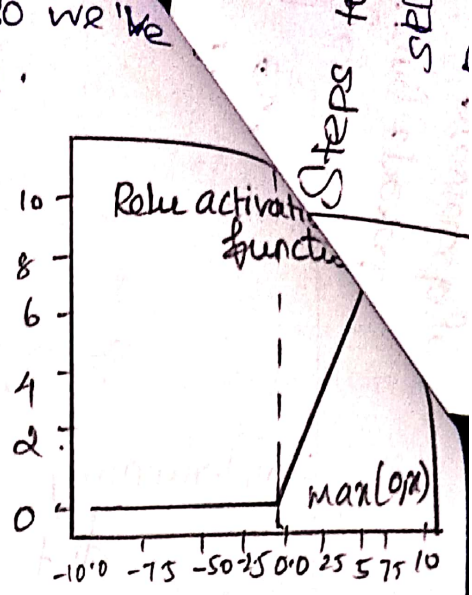
```
def relu(x)
    return max(0.0, x)
```

To test the function, lets run it on a few inputs.

```
x = 1.0
Print ('Applying Relu on (%.1f) gives %.1f' %
                                (x, relu(x)))
x = -10.0
Print ('Applying Relu on (%.1f) give %.1f'
                                % (x, relu(x)))
x = 0.0
Print ('Applying Relu on (%.1f) gives %.1f' %
                                (x, relu(x)))
x = 15.0
Print ('Applying Relu on (%.1f) gives %.1f' %
                                (x, relu(x)))
x = -20.0
Print ('Applying relu on (%.1f) gives %.1f' %
                                (x, relu(x)))
```

We see from the plot that all the -ve values have been set to zero. and +ve values are returned as it, Note that we have given a set of consecutively

increasing number as input so we've output with an increasing slope.



Adv

- Computational Simplicity
- Representational Sparsity
- Linear behaviour
- Mitigation of Vanishing Gradient
- Faster training.

Disadv

- Exploding gradient
- Dying Relu

Hyperparameter Tuning :-

Hyper parameter in ML are those parameter that are explicitly defined by the user to control the learning process. These hyperparameter are used to improve the learning of the model. and their values are set before starting the learning process of the model.

Hyper parameters are defined as the parameter that are explicitly defined by the user to control the learning process.

Some examples of model hyperparameter include.

- The penalty in logistic regression classifier ie: L1 or L2, regularization.
- The learning rate for training a neural network.
- The C and sigma hyperparameter for support vector machine.
- The k in k-nearest neighbors.

Models can have many hyperparameter and finding the best combination of parameter can be treated as a search problem.

activation function

Steps to perform Hyper parameter tuning:-

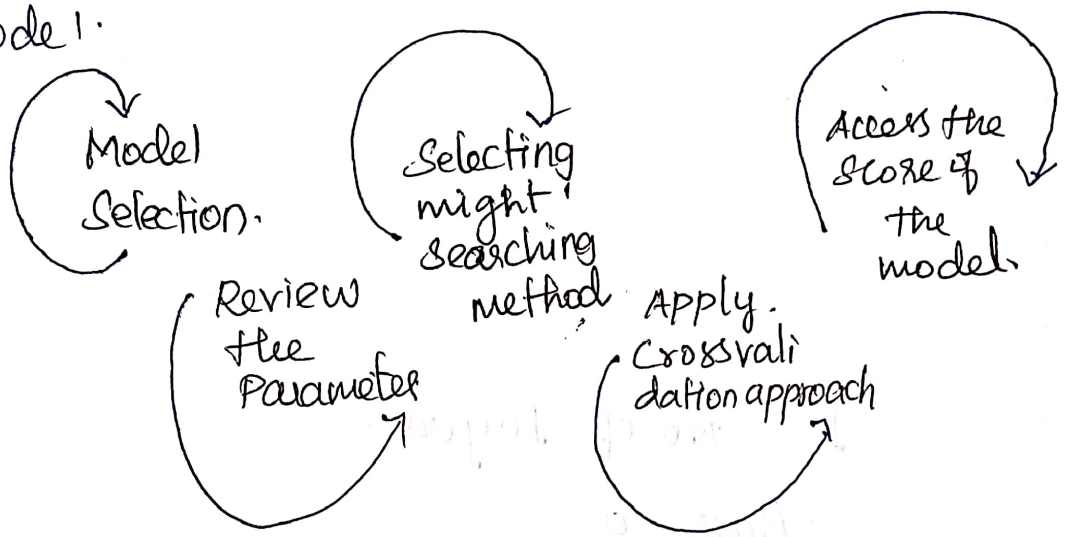
select the right type of model:

Review the list of parameters of the model and build the HP space.

Finding the methods for searching the hyperparameters space.

Applying the cross validation scheme approach.

Access the model score to evaluate the model.



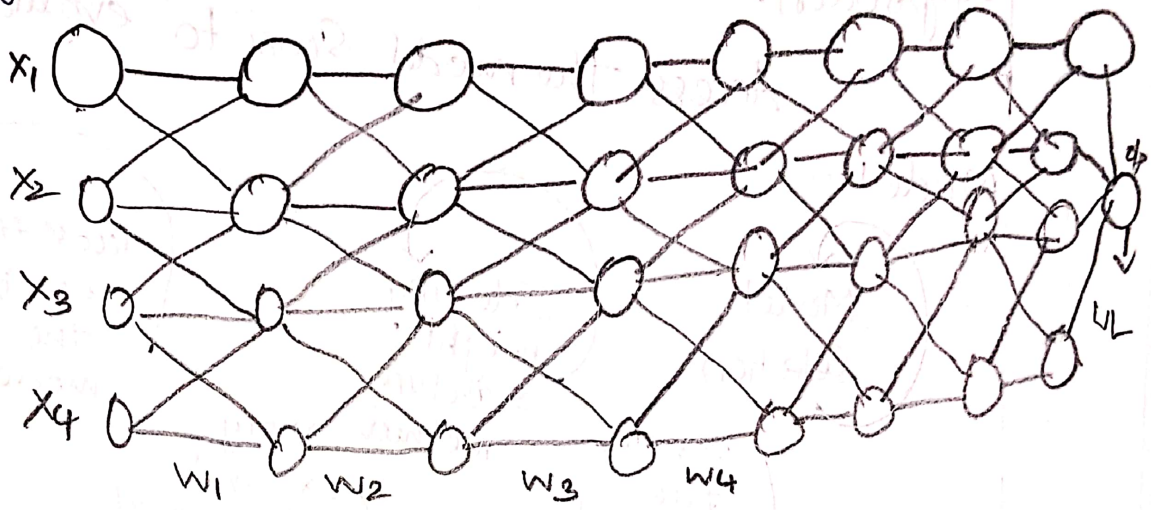
Batch Normalization:-

Normalization is a data preprocessing tool used to bring the numerical data to a common scale without distorting its shape.

Generally when we input data to a machine or deep learning algorithm we tend to change the values to a balanced scale. The reason we normalize parts to ensure that our model can generalize appropriately.

It is a process to makes neural ... faster and more stable through adding ... in a deep neural network. The new layer perform the standardizing and normalizing oper... on the input of a layer coming from a previous la...

A typical Neural N/w is trained using a collected set of i/p data called batch. Similarly the normalizing process in batch normalization takes place in batches, not as a single input.

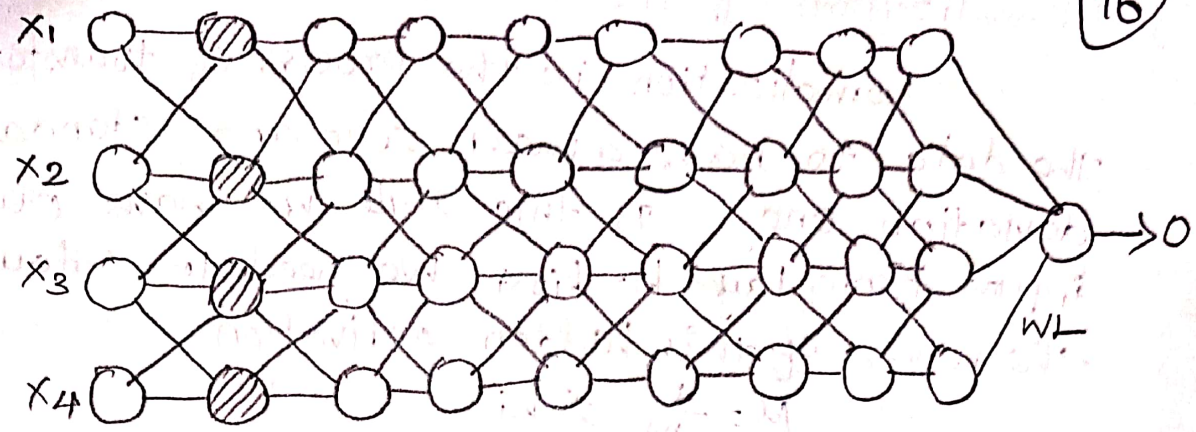


L - No of layers.

Bias - 0

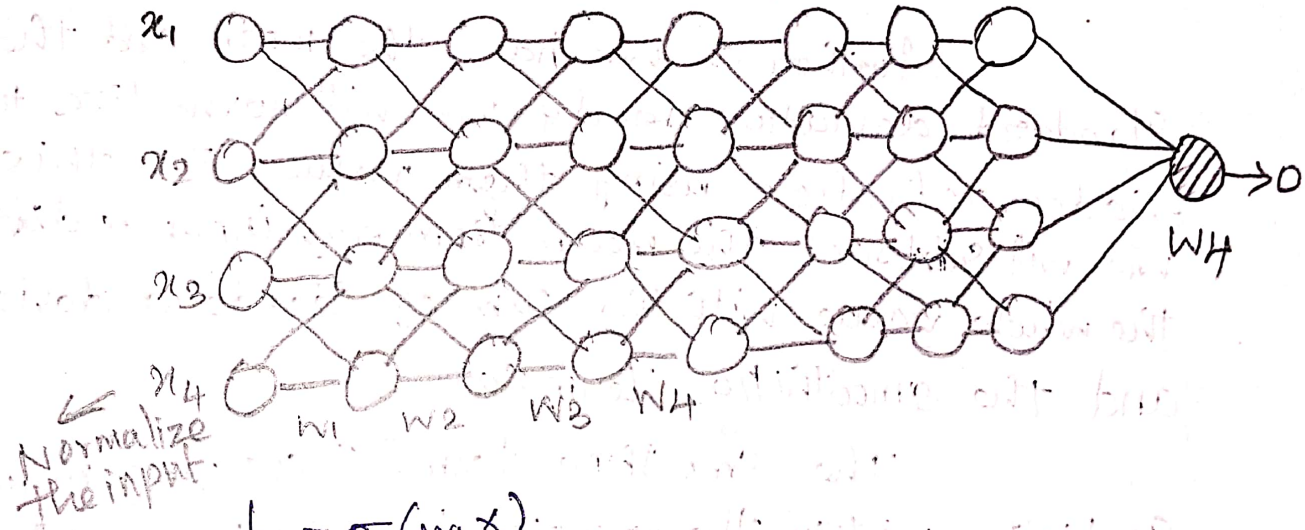
Activation function: Sigmoid.

Initially our i/p, $x_1, x_2, x_3 x_4$ are in normalized form as they are coming from the preprocessing stage, when the i/p passes through the first layer, it transform as a sigmoid function applied over the dot product of i/p x and the weight matrix w.



$$h_1 = \sigma(w_1 x)$$

Similarly, this transformation will take place for the second layer and go till the last layer L as shown in image.



$$h_1 = \sigma(w_1 x)$$

$$h_2 = \sigma(w_2 h_L) = \sigma(w_2 \sigma(w_1 x))$$

$$O = \sigma(w_L h_{L-1})$$

Working of Batch Normalization:-

Since by now we have a clear idea of why we need Batch normalization, Lets understand how it works. It is a two step process. First the input is normalized and later rescaling and offsetting is performed.

Normalization of the input:-

Normalization is the process of transforming the data to have a mean zero and standard deviation one. In this style we have our batch input from layer h. First we need to calculate the mean of this hidden activation.

$$\mu = \frac{1}{m} \sum hi$$

Here m is the no of neuron at layer h. once we have meant at our end, the next step is to calculate the standard deviation of hidden activation.

$$\mu = \sqrt{1/m \sum (hi - \mu)^2}$$

Further as we have the mean and the Standard deviation ready we will normalize the hidden activation using these value. For this we will subtract the mean from each input & divide the whole value with the sum of Standard deviation and the smoothing term (ε)

The Smoothing term (ε) assure numerical Stability within the operation by stopping a division by a zero value.

$$h(i(num)) = \frac{(hi - \mu)}{\sigma + \varepsilon}$$

Adv of Batch Normalization:-

*Speed up the training:-

By Normalizing the hidden layer activation the Batch normalization Speed up the training Process.

* Handle internal covariate shift
* internal covariate shift
* Smoothens the loss function.

Regularization:-

Regularisation refers to techniques that are used to calibrate ML models in order to minimize the adjusted loss function and prevent overfitting or underfitting.

over fitting. Good fitting.

Regularization techniques:-

The commonly used regularization techniques are:

1) L1 regularization
2) L2 regularization
3) Dropout regularization.

A regression model which uses L1 regularization technique is called LASSO regression.

A regression model that uses L2 regularization technique is called Ridge regression.

Lasso regression adds absolute value of magnitude of coefficient as penalty term to the loss function (L).

Working of Regularization

Regularization works by adding a penalty or complexity term of the complex model.

Let's consider the simple linear regression equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots \beta_n x_n + b$$

In the above eqn, 'y' represents the value to be Predicted.

$x_1, x_2 \cdots x_n$ are features for y.

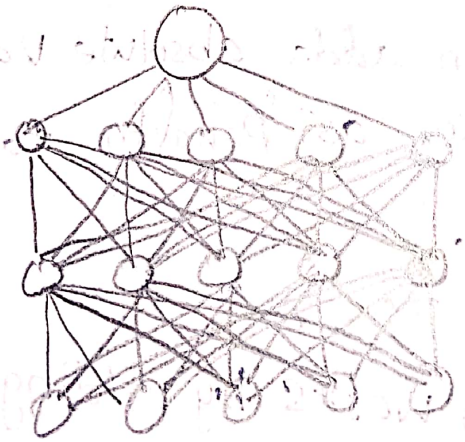
$\beta_0, \beta_1 \cdots \beta_n$ are weights or magnitude attached to the features, respectively.

The Here represents to the bias of model, and b represent the interrupt.

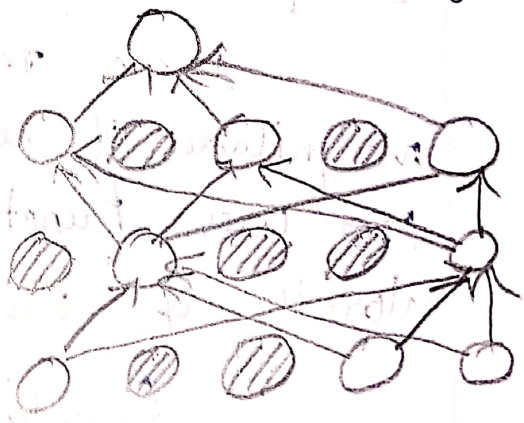
Dropout:-

Dropout in ML refer to the process of randomly ignoring certain nodes in a layer during training.

In the figure below, the neural network on the left represents a typical neural n/w ~~on the left repres~~ where all units are activated. on the right, the red units have been dropped out of the model - The values of their weights and bias are not considered during training.



Standard neural n/w



After applying dropout.

Dropout is used as a regularization technique. it presents overfitting by ensuring that no units are co dependent.

Other Common Regularization Methods:-

1) Early stopping.

Stop training automatically when a specific performance measure (eg: validation loss, accuracy) stops improving.

2) weight decay:-

The n/w to use smaller weights by adding a penalty to the loss function (this ensure that the norms of weights are relatively evenly distributed among it all the weights from heavily influencing n/w output).

Unit - IV Ensemble Techniques & Unsupervised learning (1)

Combining multiple learners: Model Combination Schemes, Voting, Ensemble Learning- bagging, boosting, stacking, Unsupervised Learning: k-means, Instance based learning: KNN, Gaussian mixture models & Expectation maximization.

1) Combining multiple learners:

Model Combination schemes:

→ Multiple base-learners are combined to generate the final output.

→ Two main approaches.

(i) multi expert combination.

(ii) multi stage combination.

i) multi expert combination model:-

It have base-learners that work is parallel.

Two types

i) Global approach (Learner Fusion).

→ All base learner generate an output and uses all the output for final decision.

Ex: Voting, Stacking.

ii) Local Approach (Learner Selection)

It uses a gating model to select specific learners.

Ex: Mixture of Expert.

Voting:-

In a voting-based ensemble, each model makes an independent prediction, and the final decision is based on majority rule or probability averaging.

Stacking:-

Stacking involves combining multiple base learners using a meta-Learner (blender model) that learns to best combine their prediction.

Steps in mixture of Experts (MoE):-

1) Input data is fed into multiple experts models.

2) The gating network assigns a probability score to each expert.

3) The experts produce outputs, weighted by the gating scores.

4) A final decision is made using a weighted combination of expert outputs.

c) Multistage Combination model :-

→ Base - learners are working is serial, the next learner is trained only if the precision one is not accurate, so increasing complexity in learners

Ex: Cascading

Mathematical representation,

→ Let L base - learner exist

→ Each base - learner M_j gives a prediction $d_j(x)$

→ Final prediction $y = f(d_1, d_2 \ldots d_L | \phi)$

where ϕ is the Combining function.

Decision making in classification,

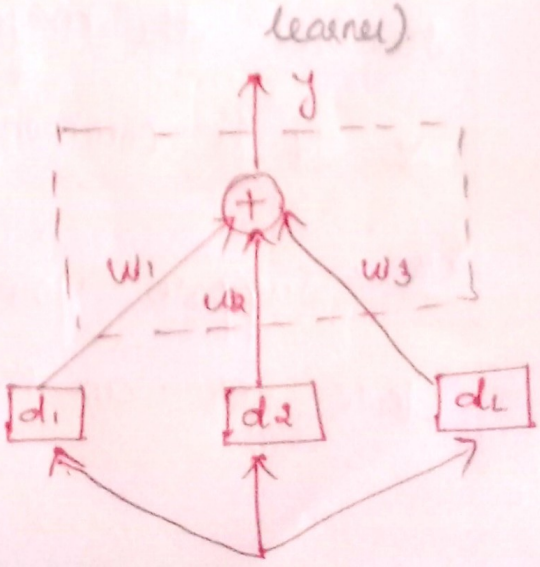
→ when k output exist for each learner,

→ we compute y_i values,

→ The class with maximum y_i is chosen.

→ Chose C_i if $y_i = \max_{k=1}^{k} y_k$ (k rep o/p of each learner)

Here base - learners are d_j, their o/p are combined using $f(\cdot)$. This is for single o/p, in case of classification, each base learner has k o/p that used to calculate y_i



Voting :-

→ Voting is an ensemble method that combines the performance of multiple models to make predictions.

→ In this technique, the 1st step is to create multiple classification models using a training dataset.

→ When the Voting is applied to regression problems, The prediction is made with the average of multiple of the regression models.

Two types of Voting :-

i) Hard Voting (majority Voting)

ii) Soft Voting (weighted Voting)

i) Hard Voting :-

* In hard Voting, each base model predicts a class label, and the final O/P is the class that receives the majority of the votes.

* It is commonly used in classification problem.

Ex: Suppose you have 3 models predicting the class for an instance.

Model	Prediction
M_1	class A
M_2	Class B
M_3	Class A

Result by hard Voting: class A, because it receive 2 out of 3. Votes.

Advantages:-

→ Simple to implement

→ works well when base models are diverse

Disadvantage:-

→ Ignore the confidence or probability of prediction.

→ All models have equal weight, which may not be ideal.

ii) Soft Voting:-

* In soft voting, each model output are based on probability scores for each class.

* The final prediction is made by averaging the predicted probabilities and choosing the class with the highest average probability.

Ex:- You have 3 classifier and their class probabilities for a binary classification problem:

Model	Class A	Class B
M_1	0.3	0.7
M_2	0.4	0.6
M_3	0.2	0.8

Average probability.

* class A: $(0.3 + 0.4 + 0.2) / 3 = 0.9/3 = 0.3$
* class B: $(0.7 + 0.6 + 0.8) / 3 = 0.21/3 = 0.7$

result by soft voting: Class B.

Advantage:

→ Takes into account the confidence of each model.

→ Typically performs better than hard voting.

Disadvantage:

→ Require models that can produce probability estimate (eg, not decision trees by default unless modified)

→ Slightly more complex to implement.

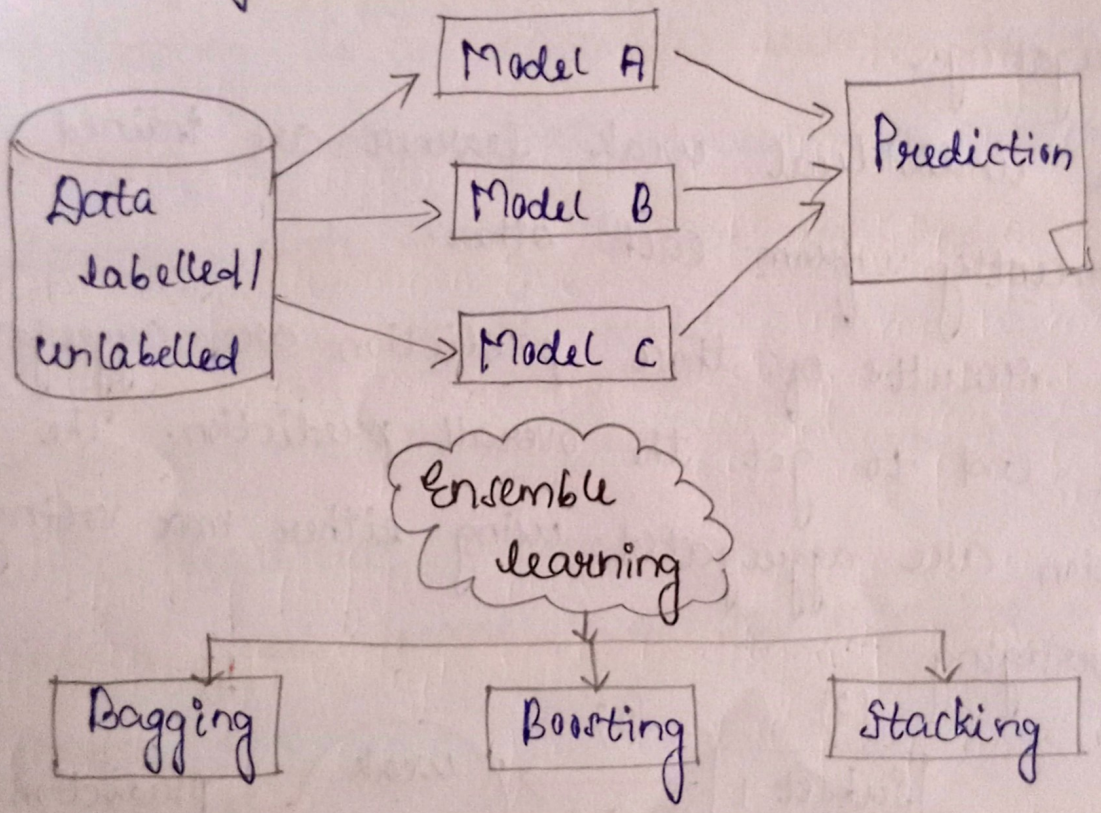
When to use Voting:

* Individual models are diverse in their architecture or learning mechanisms.
* You want to reduce variance (hard voting) or reduce bias (soft voting).
* There is no single model that clearly out performs others.

Ensemble Learning:-

→ Ensemble learning is a powerful machine learning tech that combines the predictions of multiple model to improve overall performances.

→ This technique is used to enhance accuracy, minimizing Variance and removing overfitting.



Bagging:-

Bagging is also known as bootstrap aggregating, it consists of two steps bootstrapping and aggregation.

i) Bootstrapping:-

* It involves resampling subset of data with replacement from an initial dataset. In other words, subset of data are taken from the initial dataset.

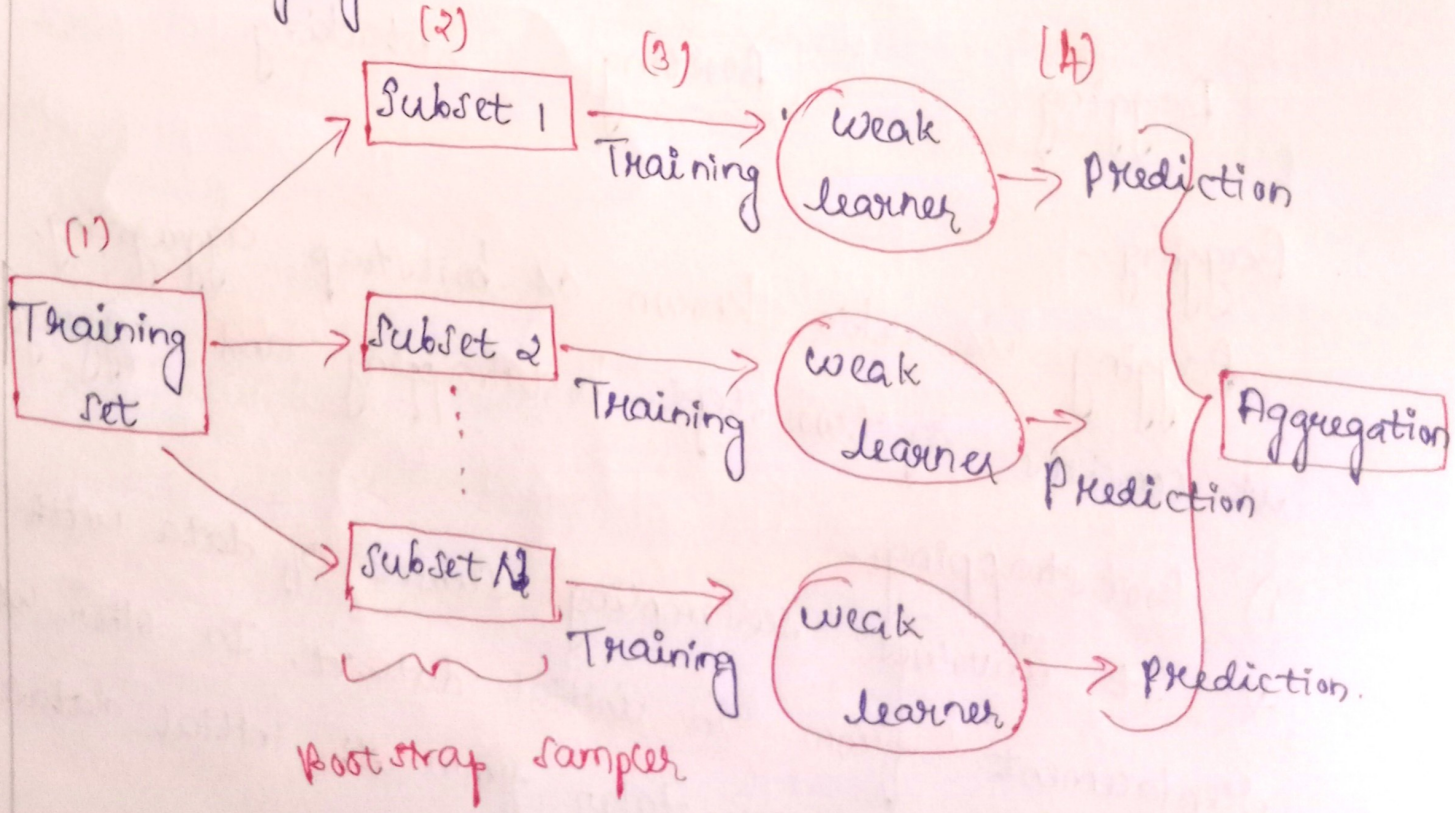
* These subsets of data are called bootstrapped dataset or simply, bootstraps.

* Resampled "with replacement" means are an individual data point can be sampled multiple times. Each bootstrap dataset is used to track a weak learner.

ii) Aggregating:-

* The individual weak learners are trained independently from each other.

* The results of those prediction are aggregated at the end to get the overall prediction. The prediction are aggregated using either max voting or averaging.



Challenges of bagging:-

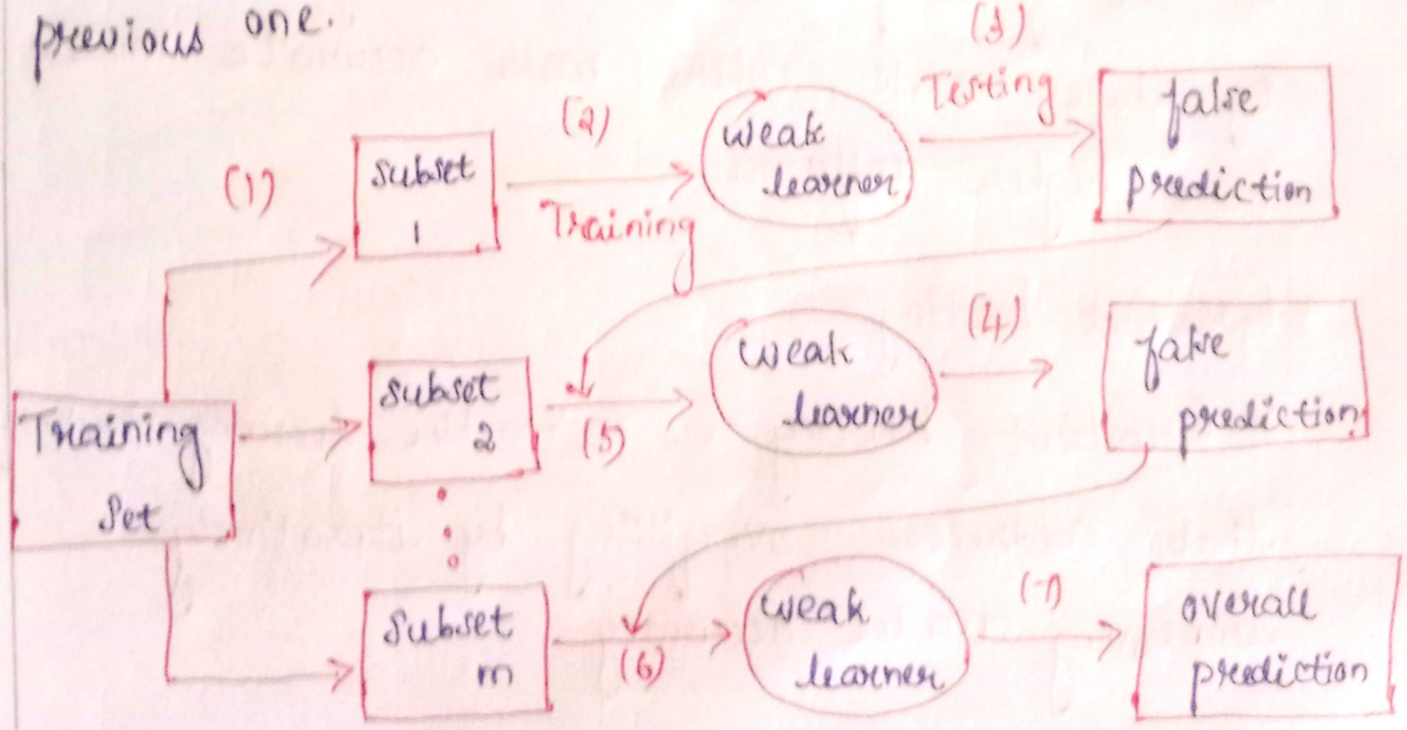
* Loss of interpretability
* Computationally expensive
* Less flexible.

Applications:

* Healthcare
* IT
* Environment
* Finance

Boosting:-

* Boosting is an ensemble learning techniques in machine learning that focuses on converting weak learner into strong learner in order to increase the accuracy of the model.
* Boosting works by combining several weak models in a sequential manner, where;
* each model learns from the mistakes of the previous one.



* The final model is a weighted sum of weak learners.
* focus is on reducing bias & variance.

Types of boosting :-

* Adaboost boosting
* Gradient boosting * XG Boost

i) Adaboost boosting :-

→ Adaboost boosting is a machine learning techniques that focuses on improving the performance of weak learner.

→ It works by sequentially training models, giving more weight to misclassified instances in each iteration.

* The final prediction is a weighted combination of these models, where more accurate models have higher influence.

Gradient boosting :-

* Gradient boosting is a machine learning technique that addresses overfitting by iteratively improving model accuracy.

* It constructs an ensemble of weak learners, typically decision trees, in a sequential manner.
* Each new learner focuses on correcting the errors made by the previous ones.
* The final model is a weighted sum of these learner's predictions.
* Gradient boosting effectively reduces bias & variance, producing a strong predictive model that generalize well to new data.

XG Boost :-

* XG Boost is a machine learning algorithm known for its speed, accuracy and efficiency in handling structured, tabular data.
* It falls under the category of gradient boosting algorithm, which work by combining multiple weak learners (usually decision trees) sequentially to create a strong predictive model.

Ans: Differences b/w bagging & Boosting.

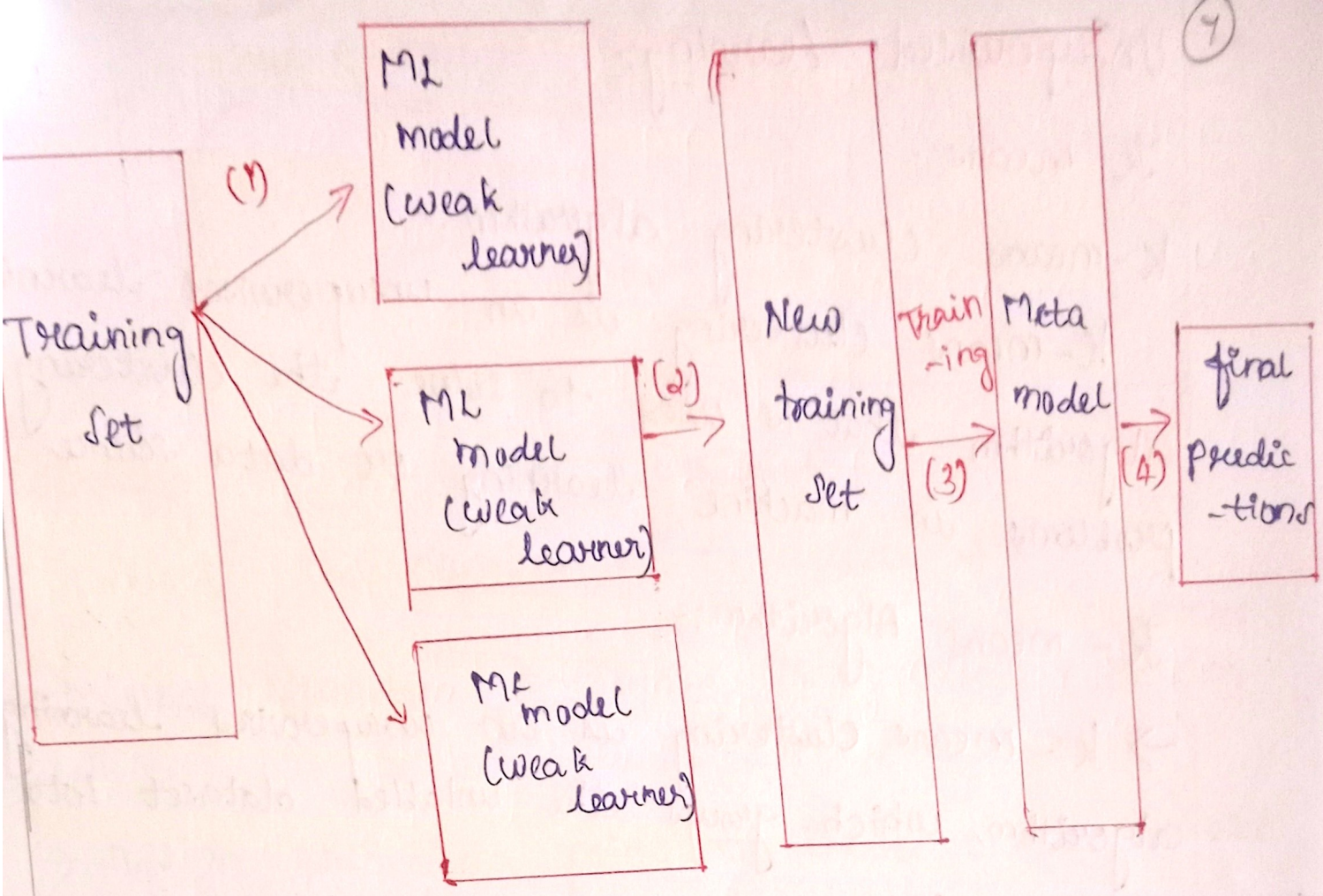
Stacking:-

* Stacking is one of the popular ensemble modeling techniques in machine learning, Various weak learners are ensembled in a parallel manner in such a way that by combining them with meta learners. we can predict better predictions for the future.

* This method is a combination of multiple regression or classifier techniques with a meta - regressor or meta - classifier.

* Stacking is different from bagging and boosting. Bagging and Boosting models work mainly on homogeneous weak learners and don't consider heterogeneous learner, where the stacking works mainly on heterogeneous weak learners and consists of different algorithm together.

* The bagging and boosting techniques combine weak learners with the help of deterministic algorithms, where the stacking method combines the weak base learners with the help of a meta- model.



	Bagging	Boosting	Stacking
Purpose	(Test data) reduce variance	(Training data) reduce bias	Improve accuracy
Base learner types	Homogeneous	Homogeneous	Heterogeneous
Base learner training	Parallel	Sequential	meta model
Aggregation	Max Voting averaging	weighted averaging	weighted averaging

Unsupervised Learning:-

K means:-

K-means clustering algorithm:-

K-means clustering is an unsupervised learn algorithm that is used to solve the clustering problems in machine learning or data science

K- means Algorithm:-

→ k-means clustering is an unsupervised learning algorithm, which groups the unlabled dataset into different cluster.

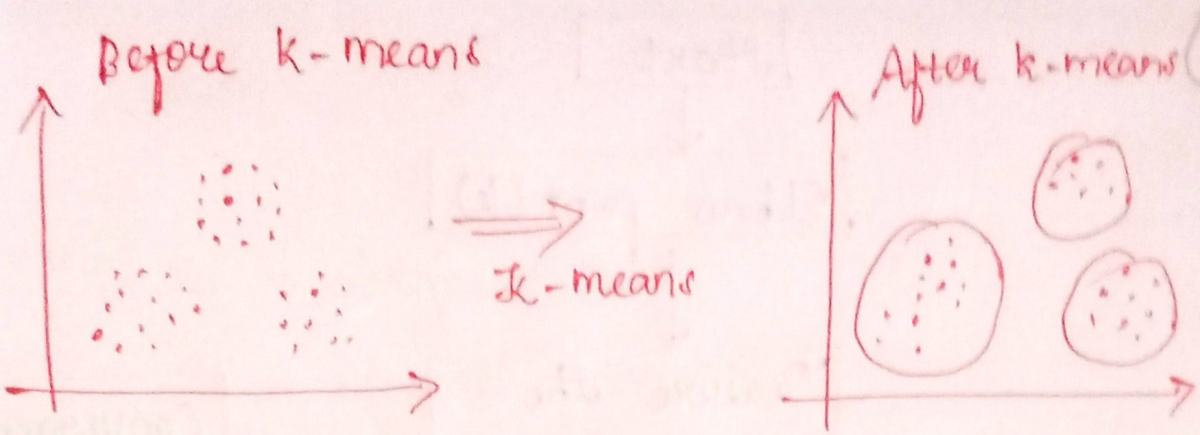
→ Here k defines the no of pre-defined clusters that need to be created in the process.

→ If k=2, there will be two clusters, &
k=3, there will be three clusters.

K means clustering algorithm perforns two tasks:-

→ Determine the best value for k center points or centroids by an iterative process.

→ Assigns each data points to its closest k-center Those data points which are near to the particular k-center, create a cluster.



Working of k-means algorithm:-

→ Select the number k to decide the no of clusters.

→ Select random k-points or centroids (It can be other from the i/p dataset).

→ assign each data point to their closet centroid which will form the predefined K-cluster.

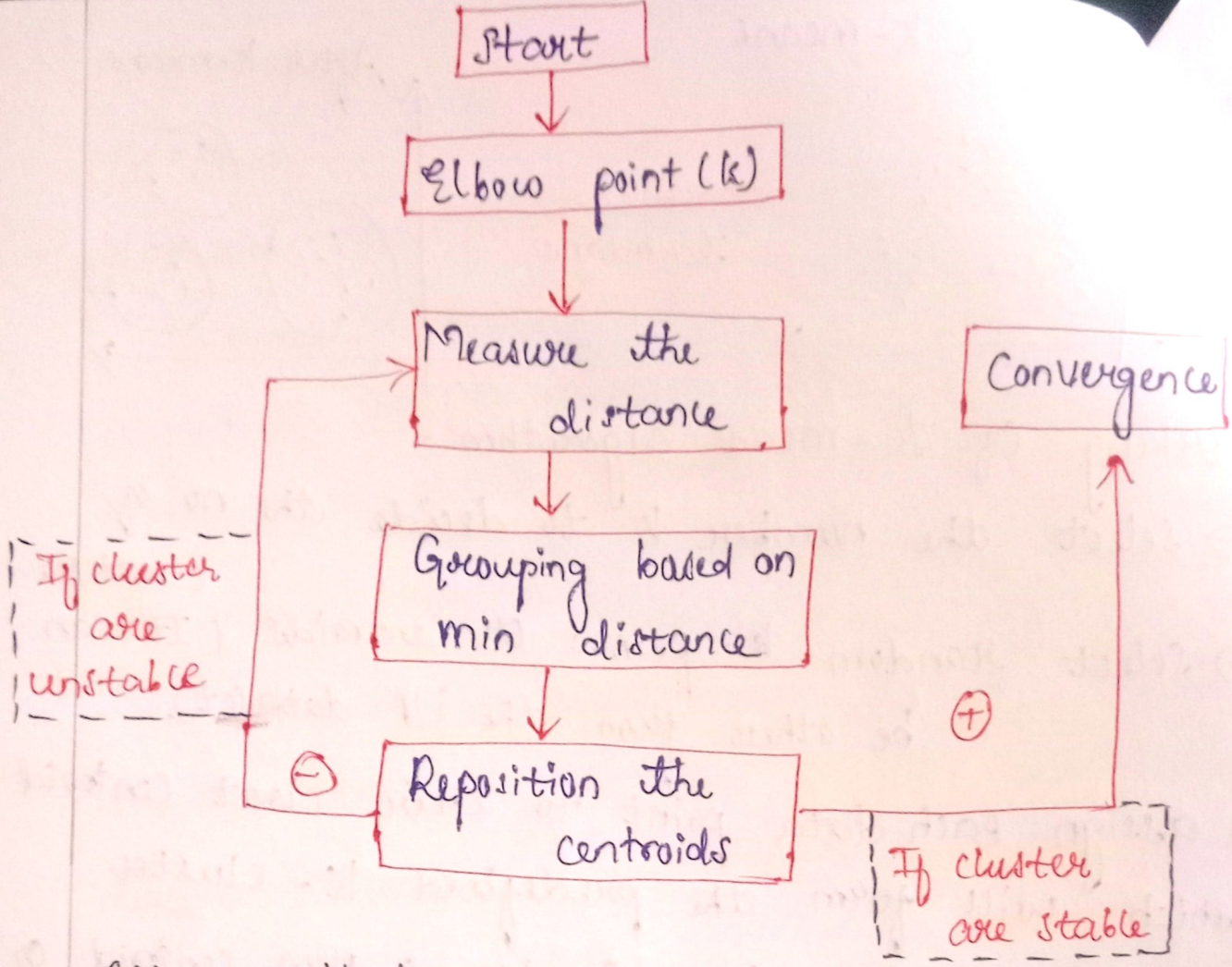
→ Calculate the variance & place a new centroid of each cluster.

→ Repeat the 3rd steps, which means reassign each data point to the new closest centroid of each cluster.

→ If any reassignment occurs, then go to step 4. else go to FINISH.

→ The model is ready.

How to choose the value of "K Number of Cluster" in k-means clustering.



Elbow method:-

→ The Elbow method is the best way to find the no of clusters, the elbow method consitutes running k-means clustering on the dataset.

→ This method uses the concept of WCSS value. within cluster sum of squares as a measure to find the optimum no of clusters and the total Variations within a cluster

→ WSS is defined as the sum of the squared distance b/w each member of the cluster and its Centroid.

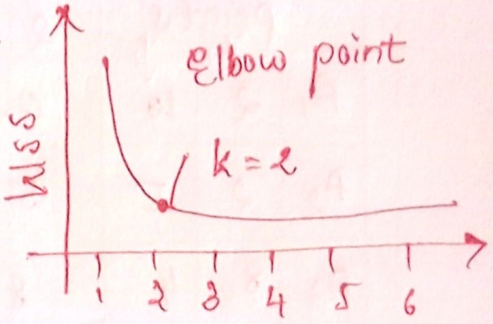
$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

where, x_i - datapoint

c_i - closest point to Centroid.

The WSS is measured for each value of k, The value of k, which has the least amount of WSS, is taken as the optimum value.

→ Here, WSS is on the y-axis and no. of cluster on the x-axis



Application of k-means clustering :-

→ Distance measure.
→ k-means for encryption
→ Customer segmentation
→ Image segmentation
→ recommendation engineers.

Advantages :-

→ simple and easy to implement
→ Fast and efficient
→ Scalability
→ flexibility

Disadvantages :-

→ Sensitivity to initial Centroids
→ requires specifying the no of clusters.
→ sensitive

Problem :-

$A_1(2,10)$ $A_2(2,5)$ $A_3(8,4)$ $B_1(5,8)$ $B_2(7,5)$ $C_1(1,2)$ $C_2(4,9)$ The distance function is euclidean distance & initial values are A_1, B_1, C_1 as the center of each cluster respectively.

Data points	x_1	y_1	Distance to x_2 2, y_2 10	Distance to x_2 5, y_2 8	Distance to x_2 1, y_2 2	cluster	New cluster
A_1	2	10	0.00	3.60	8.06	1	
A_2	2	5	5.00	4.24	3.16	3	
A_3	8	4	8.48	5.00	7.28	2	
B_1	5	8	3.60	0	7.21	2	
B_2	7	5	7.07	3.60	6.70	2	
B_3	6	4	7.21	4.12	5.38	2	
C_1	1	2	8.06	7.21	0.00	3	
C_2	4	9	2.23	1.41	7.61	2	

Euclidean distance $d(P_1,P_2) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

$A_1 = \sqrt{(2-2)^2 + (10-10)^2} = 0$

$A_2 = \sqrt{(2-2)^2 + (10-5)^2} = 5$

$A_3 = \sqrt{(2-8)^2 + (10-4)^2} = \sqrt{(-6)^2 + (6)^2} = 8.48$

$B_1 = \sqrt{(2-5)^2 + (10-8)^2} = \sqrt{(3)^2 + (2)^2} = 3.60$

$B_2 = \sqrt{(2-7)^2 + (10-5)^2} = \sqrt{5^2 + 5^2} = 7.07$

$B_3 = \sqrt{(2-6)^2 + (10-4)^2} = \sqrt{4^2 + 6^2} = 7.21$

$C_1 = \sqrt{(2-1)^2 + (10-2)^2} = \sqrt{(1)^2 + (8)^2} = 8.06$

$C_2 = \sqrt{(2-4)^2 + (10-9)^2} = \sqrt{(2)^2 + (1)^2} = 2.23$

$A_1 = \sqrt{(5-2)^2 + (8-10)^2} = \sqrt{3^2 + (2)^2} = 3.60$

$A_2 = \sqrt{(5-2)^2 + (8-5)^2} = \sqrt{(3)^2 + (3)^2} = 4.24$

$A_3 = \sqrt{(5-8)^2 + (8-4)^2} = \sqrt{3^2 + 4^2} = 5$

$B_1 = \sqrt{(5-5)^2 + (8-8)^2} = 0$

$B_2 = \sqrt{(5-7)^2 + (8-5)^2} = \sqrt{2^2 + 3^2} = 3.60$

$B_3 = \sqrt{(5-6)^2 + (8-4)^2} = \sqrt{1^2 + 4^2} = 4.12$

$C_1 = \sqrt{(5-1)^2 + (8-2)^2} = \sqrt{4^2 + 6^2} = 7.21$

$C_2 = \sqrt{(5-4)^2 + (8-9)^2} = \sqrt{1^2 + 1^2} = 1.41$

$A_1 = \sqrt{(1-2)^2 + (2-10)^2} = \sqrt{1^2 + 8^2} = 8.06$

$A_2 = \sqrt{(1-2)^2 + (2-5)^2} = \sqrt{1^2 + 3^2} = 3.16$

$A_3 = \sqrt{(1-8)^2 + (2-4)^2} = \sqrt{7^2 + 2^2} = 7.28$

$B_1 = \sqrt{(1-5)^2 + (2-8)^2} = \sqrt{4^2 + 6^2} = 7.21$

$B_2 = \sqrt{(1-7)^2 + (2-5)^2} = \sqrt{6^2 + 3^2} = 6.70$

$B_3 = \sqrt{(1-6)^2 + (2-4)^2} = \sqrt{5^2 + 2^2} = 5.38$

$C_1 = \sqrt{(1-1)^2 + (2-2)^2} = 0$

$C_2 = \sqrt{(1-4)^2 + (2-9)^2} = \sqrt{3^2 + 7^2} = 7.61$

New centroids:

A_1 : (2, 10)

B_1 : (6, 6)

C_1 : (1.5, 3.5)

from their centroids (consider there centroid as current initial centroids;

Data points			Distance to						Cluster	New Cluster
			2	10	6	6	1.5	3.5		
A_1	2	10	0.00		5.66		6.52		1	1
A_2	2	5	5.00		4.12		1.58		3	3
A_3	8	4	8.48		2.83		6.52		2	2
B_1	5	8	3.60		2.24		5.70		2	2
B_2	7	5	7.07		1.41		5.70		2	2
B_3	6	4	7.21		2.00		4.53		2	2
C_1	1	2	8.06		6.40		1.58		3	3
C_2	4	9	2.23		3.61		6.04		2	1

New centroids:

A_1 : (4, 9.5)

B_1 : (6.5, 5.25)

C_1 : (1.5, 3.5

from this centroids (consider their centroids as current initial centroids:

Current Centroids

A_1: (3, 9.5)

B_1: (6.5, 5.25

C_1: (1.5, 3.5)

Current Centroids:

A_1: (3.67, 9)

B_1: (7, 4.33)

C_1: (1.5, 3.5)

So this makes same in clustering

A_1 belongs to 1st

B_1 belongs to 1st

C_2 belongs to 1st

A_3 belongs to
B_2 belongs to
B_3 belongs to
} 2nd cluster.

A_2 belongs to
C_1 belongs to
} 3rd cluster.

Ass problem:-

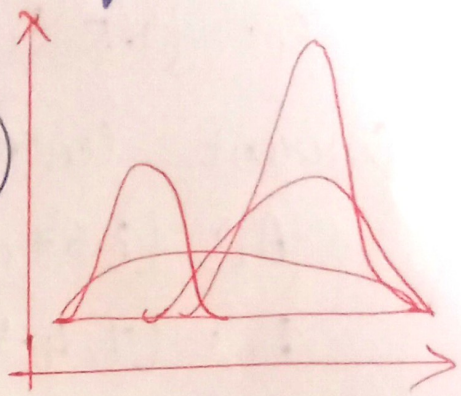
$P_1(1,2,3)$ $P_2(0,1,2)$ $P_3(3,0,5)$ $P_4(4,1,3)$

$P_5(5,0,1)$ Initial centroid $C_1(1,0,0)$ $C_2(0,1,1)$.

Gaussian distribution :-

It has a bell shaped curve, with the data Symmetrically distributed around the mean Value. Here gaussian distribution with a difference is mean and variance (σ^2).

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



where,

x is the data point,

μ is the mean (center of distribution)

σ^2 is the variance (squared of the distribution)

Defference blw GMM & k-means : (A SS)

Expectation Maximization :-

→ The EM algorithm is considered a latent variable model to find the local maximum likelihood parameter of a statistical model.

→ The EM algorithm is one of the most commonly used terms in machine learning to obtain maximum likelihood estimates of variable that are sometimes observable & sometimes not.

Instance Based learning:-

Instance based learning is also known as lazy learning or memory-based learning, is a machine learning approach that makes predictions or classifications based on the similarity b/w new instance and the training examples. Some of the instance-based learning algorithm are;

* KNN
* Self-Organizing Map (SOM)
* Learning Vector Quantization (LVQ)

KNN - Nearest Neighbour:-

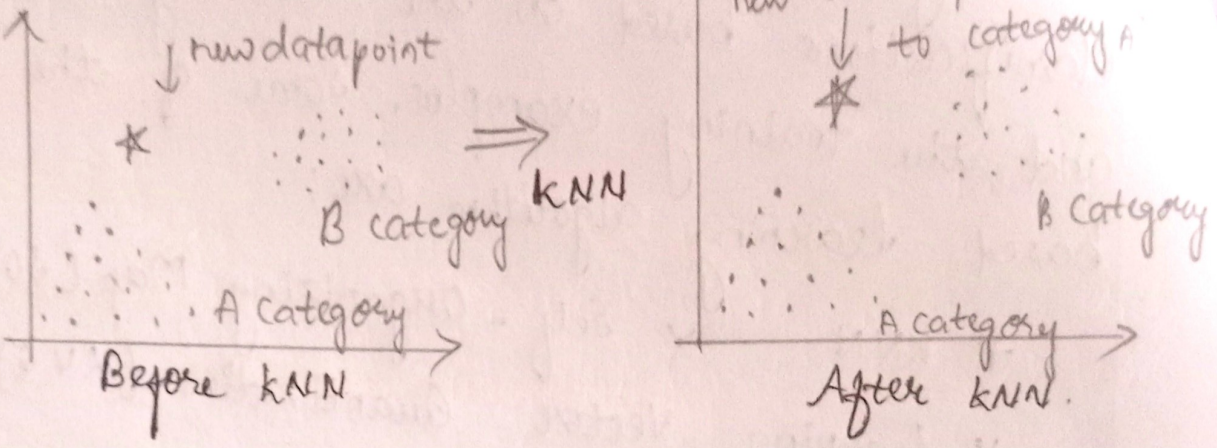
→ K-Nearest Neighbours (KNN) algorithm is a type of unsupervised ML algorithm which can be used for both classification as well as regression problems. It is mainly used for classification problems in industry.

→ Lazy learning algorithm - KNN is a lazy learning algorithm because it does not have a specialized training phase & uses all the data for training while classification.

→ Non-parametric learning algorithm - KNN is also a non-parametric learning algorithm because it doesn't

Assume anything about the underlying data.

→ It makes predictions based on the similarity (typically distance) b/w the new data point (new instance) and the stored instances.



working of KNN :-

→ Select the no of K of the neighbour.

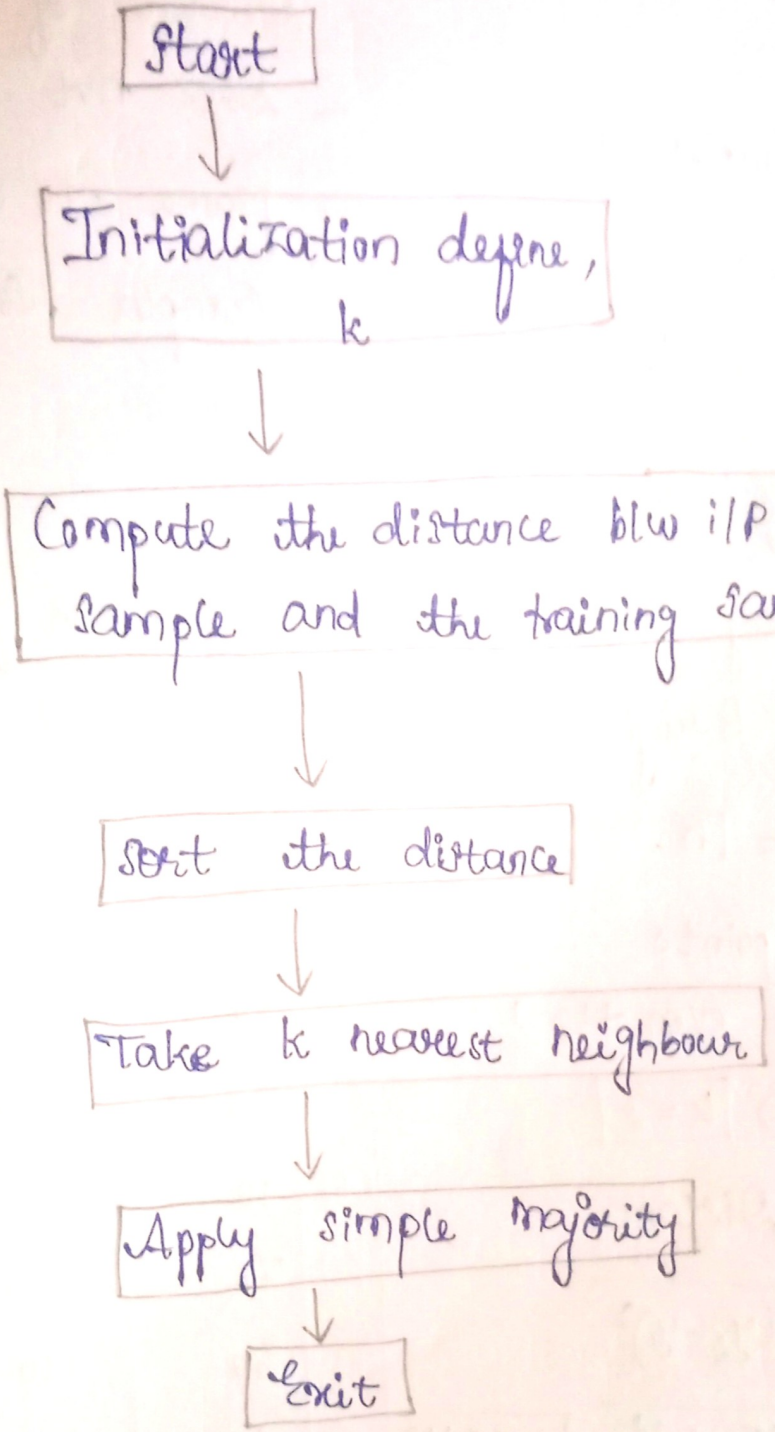
→ Calculate the Euclidean distance of K no of neighbour.

→ Take the K nearest neighbour as per the calculated euclidean distance.

→ Among these K neighbour, count the no of the data points in each category.

→ Assign the new data point to each that category for which the no of the neighbour is max.

→ Our model is ready.



Euclidean distance:-

Euclidean distance b/w the data points. The points euclidean distance is the distance b/w two points, which we have already studied in geometry. It can be calculated as,

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

Adv of KNN:
→ Easy to implement
→ Adapts easily
→ few hyperparameter
→ more effective

Disadv of KNN:
- Does not scale
- prone to overfitting
- Complex for large data

Problem:- Using KNN, predict the class for new data entry with Brightness = 20; Saturation = 35

	Brightness	Saturation	Class
(1)	40 x_1	20 y_1	Red
(2)	50	50	Blue
(3)	60	90	Blue
(4)	10	25	Red
(5)	70	70	Blue
(6)	60	10	Red
(7)	25	80	Blue
	20 x_2	35 y_2	Red

(1) K factor = $\sqrt{n}$

no of data points

If n is even +1, -1

$\sqrt{7} = 2.64 \Rightarrow \boxed{k = 3}$

(2) Euclidean Distance

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

(1) $\Rightarrow \sqrt{(20-40)^2 + (35-20)^2} = \sqrt{400+225}$

(2) $\Rightarrow \sqrt{(20-50)^2 + (35-50)^2} = \sqrt{900+225}$

(3) $\Rightarrow \sqrt{(20-60)^2 + (35-90)^2} = \sqrt{1600+3025}$

(4) $\Rightarrow \sqrt{(20-10)^2 + (35-25)^2} = \sqrt{100+100}$

(5) $\Rightarrow \sqrt{(20-70)^2 + (35-70)^2} = \sqrt{2500+1225}$

(6) $\Rightarrow \sqrt{(20-60)^2 + (35-10)^2} = \sqrt{1600+625}$

(7) $\Rightarrow \sqrt{(20-25)^2 + (35-80)^2} = \sqrt{25+2025}$

(1)	25 → Red min to 3(k)
(2)	38.54
(3)	68.00
(4)	14.14 → Red
(5)	61.03
(6)	47.16
(7)	45.27

Gaussian Mixture Models :-

→ A Gaussian mixture model is a model that assumes the data comes from a mixture of a finite no of Gaussian distributions. The goal of modeling is to estimate the parameter of such Gaussian components, namely their means and covariance matrices.

→ The Gaussian mixture model is defined as a mixture model that has a combination of the unspecified probability distribution functions.

→ Gaussian mixture models are probabilistic models & use the soft clustering approach for distributing the points in different cluster.

Key components of GMM :-

(1) Means (μ) :-

Each Gaussian distribution in the mixture has its own mean, which determines the center of the cluster.

(2) Covariances (Σ) :-

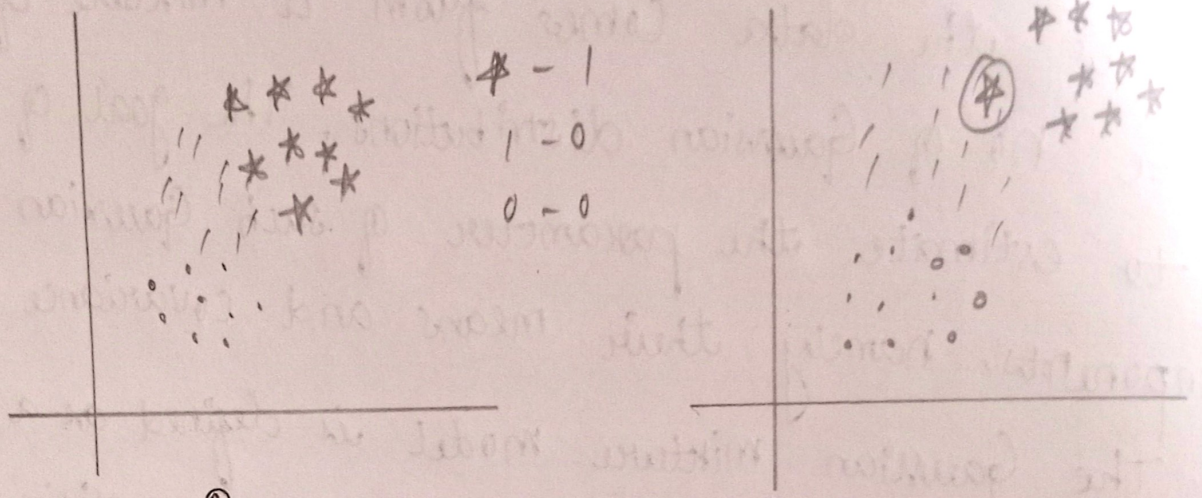
This defines the shape & orientation of each gaussian distribution, allowing for ellipsoidal clusters.

(3) Mixing Coefficient :-

* These are the weights assigned to each Gaussian distribution, indicating the proportion of data

belonging to each other.

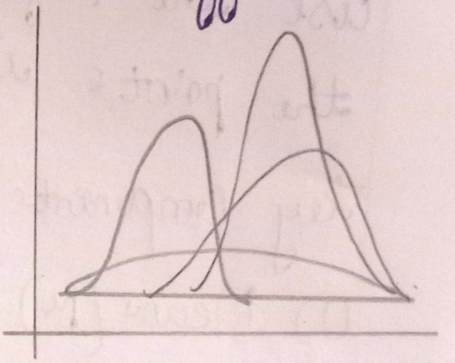
* The sum of all mixing coefficient is 1.



Gaussian Distribution :-

It has a bell shaped curve, with the data points symmetrically distributed around the mean value.

Here few Gaussian distribution with a difference in mean (μ) and variance (σ^2).



$$f(x/\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right.$$

where,

x is the data point,

μ is the mean (center of the distribution)

σ^2 is the Variance (spread of the distribution)

Diff blw GMM & k-means : (Ass)

Expectation Maximi

Expectation Maximization :-

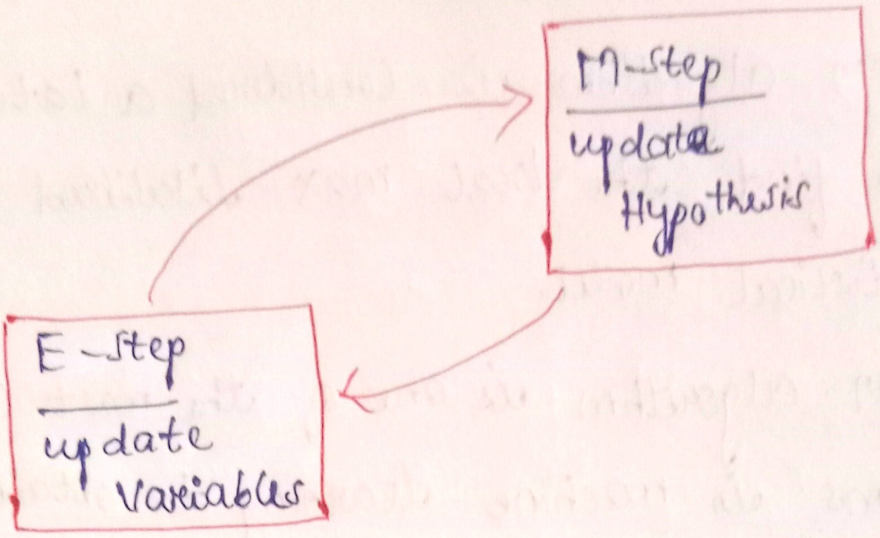
→ The EM algorithm is considered a latent variable model to find the local max likelihood parameter of a statistical model.

→ The EM algorithm is one of the most commonly used terms in machine learning to obtain max likelihood estimates of variable that are sometimes observable & sometimes not.

EM algorithm :-

→ The EM algorithm is the combination of various unsupervised ML algorithms, such as k-means clustering algorithm, which is used to determine the local max likelihood estimates (MLE) or max a posteriori estimates (MAP) for unobservable variable in statistical models.

→ A latent variable model consists of both observable and unobservable variable where observable can be predicted while unobserved are inferred from the observed variable. These unobservable variance are known as latent.



Expectation step (E-step):-

It involves the estimation (guess) of all missing values in the dataset, so that after completing this step, there should not be any missing value (Estimates missing or hidden values using current parameter estimates).

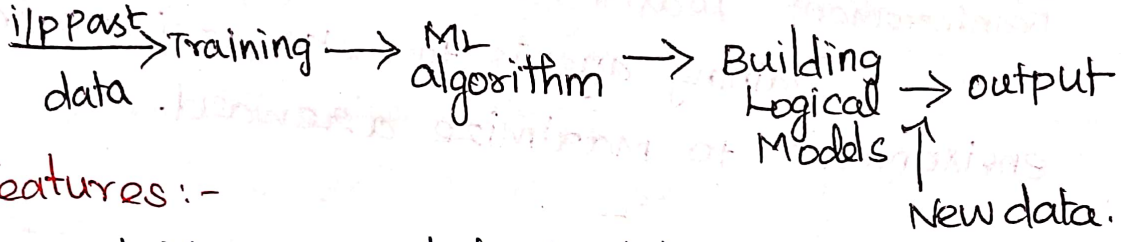
UNIT-III

SUPERVISED LEARNING

Introduction to Machine Learning:-

ML Learn from historical data, build the Prediction models and whenever it receives new data, Predicts the output for it.

The accuracy of predicted output depend upon the amount of data, as the huge amount of data helps to build a better model which Predict the output more accurately.



Features:-

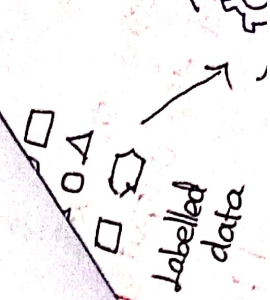
- → ML uses data to detect various patterns in a given dataset.
- → It learn from past data and improve automatically.
- → It is a data driven technology

Importance of Machine Learning.

- → Rapid increment in the production of data.
- → Solving complex problem, which are difficult for a human.
- → Decision Making in various sectors. including finance
- → Finding hidden patterns and extracting useful information from data.

Types of ML:-

* Supervised Learning
* unsupervised Learning.
* Reinforcement Learning.



Supervised Learning:-

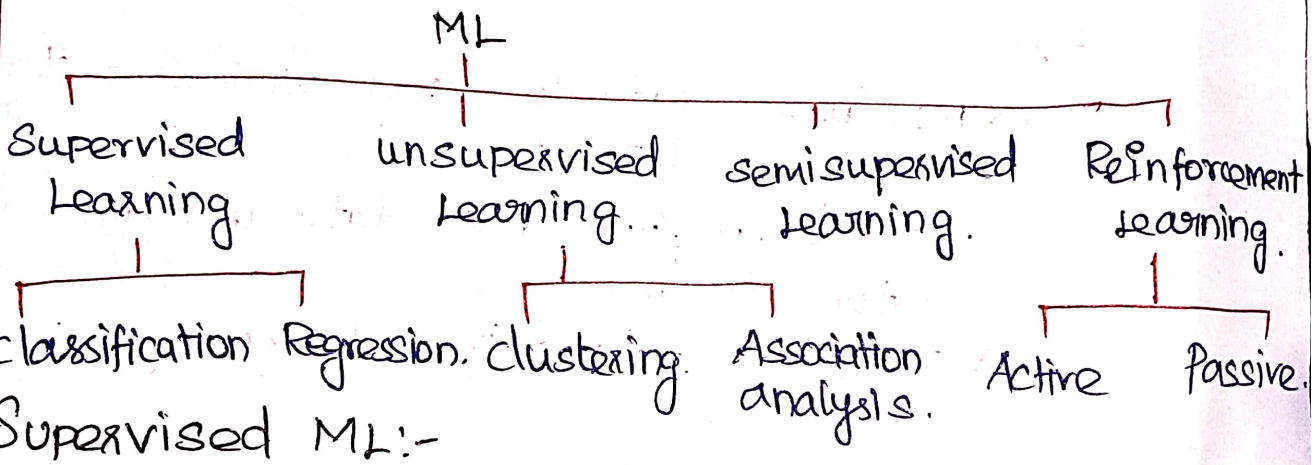
Training models on labeled data to predict outcomes or classify data points.

Unsupervised Learning:-

Discovering patterns and Structures in unlabeled data.

Reinforcement Learning:-

Training agents to make decision in an environment to maximize a reward.

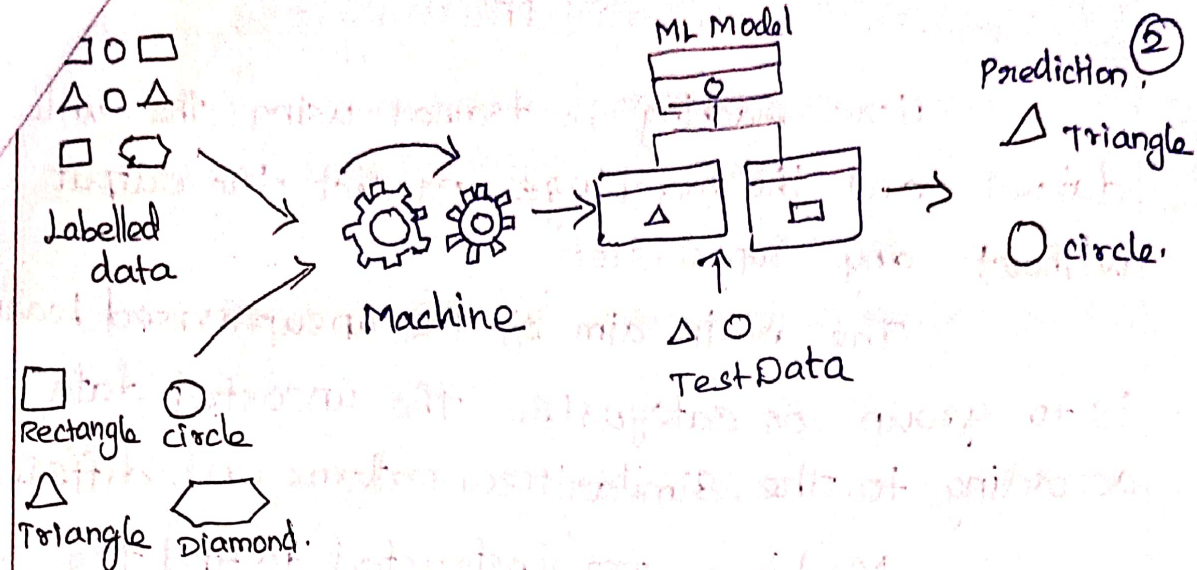


Supervised ML:-

It train the machine using the labelled dataset and based on the training, the machine Predicts the output.

The labelled data specifies that some of the input are already mapped to the output.

Train the machine with the input and corresponding output, and then the machine will Predict the output using the test dataset.



Types :

i) Classification

ii) Regression.

i) Classification :-

↳ It used to solve the classification problem in which the output variable is categorical.

↳ Such as 'yes' or 'No', female or Male

↳ The classification algorithm predict the categorical present in the dataset.

↳ Some popular classification algorithm are

Random forest, Decision tree, Logistic regression, Support vector Machine.

Regression :-

↳ It is used to solve Regression Problem in which time is a linear Relationship b/w i/p and o/p value

↳ This are used to predict Continuous output variable such as Market friend, weather Prediction, mark of Students etc.

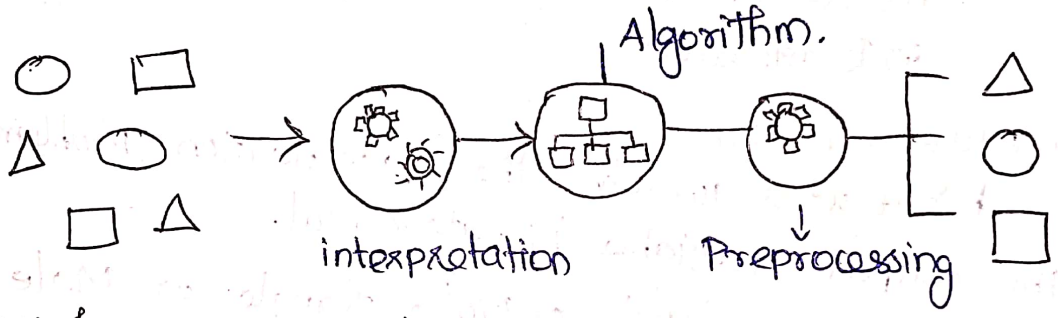
Some popular Regression Algorithm are, Simple linear Regression, Multivariate Regression, Decision Tree, Lasso Regression.

Unsupervised Maching Learning :-

Here maching is trained using the dataset, and the machinge predict the outp without any supervision.

The main aim of the unsupervised Learn is to group or categories the unsorted dataset according to the similarities, patterns and differences.

Machines are instructed to find the hidden patterns from the i/p dataset.



Types of unsupervised ML.

i) clustering
(ii) Association.

Clustering :-

It is used to find the inherent groups from the data.

The object in the group are most similar to each other and no similarities with the objects to other groups.

An example of the clustering algorithm is grouping the customers by their purchasing behavious.

ular clustering Algorithms:-

i) k-Means clustering
ii) Mean shift Algorithm.
iii) Principal Component Analysis
iv) Independent component Analysis.

Association:-

↳ It find interesting relation among variables within a large dataset.

↳ It is used to find the dependency of one data items on another data item.

↳ Based of dependency it maps those variable so that it can generate maximum.

SEMI SUPERVISED LEARNING:-

↳ To overcome the drawbacks of supervised learning and unsupervised learning algorithm, the concept of semi-Supervised is introduced.

↳ Hence it uses the combination of labelled and unlabelled dataset during training Period.

REINFORCEMENT LEARNING:-

↳ Reinforcement learning works on a feedback, based process - learning from experiences and improve its Performance.

↳ In Reinforcement learning, there is no labelled data like supervised learning and agents learn from their experiences only.

Types of Reinforcement Learning:-

1) Active Reinforcement learning
2) passive Reinforcement Learning.

Passive

↳ The agent policy (sequence of action) is fixed which means that it is told what to do

↳ The goal of passive RL agent is to execute a fixed policy and evaluate it.

Active

↳ An Agent need to decide what to do as there's no fixed policy that it can act on.

↳ An active RL agent is to act an learn an optimal policy.

Linear Regression Model:-

Linear Regression:-

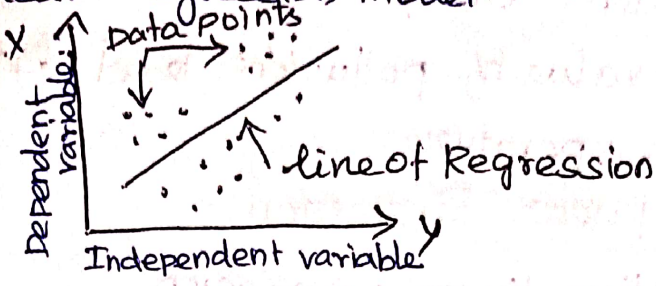
↳ linear Regression is an algorithm that Provides a linear relationship b/w an independent variable and a dependent variable to predict the outcome of future events.

↳ Here the dependent variable is also Called the response variable and independent variable also called a explanatory or predictor variables.

↳ Thus linear regression is a Supervised Learning algorithm that Simulates a mathematical relationship b/w variables and make predictions for continuous or numeric variables such as Sales

salary, age, etc.

Here A Scoped Straight line represent the linear regression model.



Least Squares:

In statistics, Linear regression is a linear approach to model the relationship b/w a scalar response (dependent variable) - ray x, and one or more explanatory variable (independent variable) - ray y.

The linear regression model provides a sloped straight line representing b/w the variables Mathematically, we can represent a linear regression as

$$y = a_0 + a_1 x + \epsilon$$

y - dependent variable (Target variable)

x - Independent variable (Predictor variable)

a_0 - Intercept of the line

a_1 - linear regression coefficient ϵ - random error

TYPES OF LINEAR REGRESSION:-

1) Simple linear Regression:-

If a single independent variable is used to Predict the value of a numerical dependent variable, then such a linear regression algorithm is called Simple linear regression.

Simple linear regression reveals the correlation b/w a dependent variable (i/p) and an independent variable (o/p).

It is relationship strength b/w the of the dependent variable is based on the of the independent variable.

Ex! The value of pollution level data at a specific temperature.

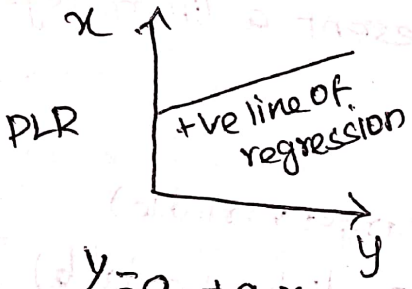
(2) Multiple linear Regression:-

1) positive linear regression

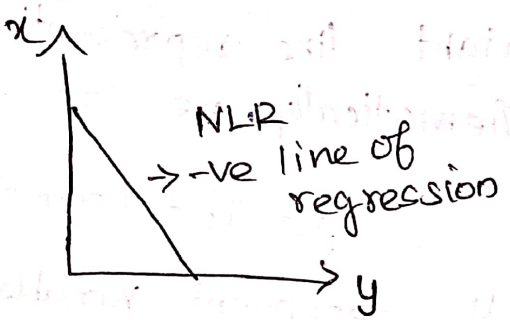
2) Negative linear regression.

1) Positive linear Relationship:-

If the dependent variable increase on the y-axis and independent variable increase on the x-axis, then such a relationship is termed as PLR.



$y = a_0 + a_1 x$



$y = a_0 + a_1 x$

$Y = -a_0 + a_1 x.$

(ii) Negative linear Relationship:-

If the dependent variable decreases on the y-axis and independent variable increase on the x-axis then, such a relationship is called NLR.

Simple linear Regression Problem:-

A Dataset containing information about the no of hours students spend studying and their corresponding scores on a test. Your task is to perform Simple Linear regression to Predict test scores based on the no of hours studies using following dataset.

No of hours studied (x)	Test scores (y)	x^2	xy
2	75	4	150
8	82	9	246
4	93	16	372
5	89	25	445
6	98	36	588

$\bar{x} = \frac{20}{5} = 4$ $\quad \bar{y} = \frac{437}{5} = 87.4$ $\quad \bar{x^2} = \frac{90}{5} = 18$ $\quad \bar{xy} = \frac{1801}{5} = 360.2$

$$a_1 = \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x^2} - (\bar{x})^2} \qquad a_0 = \bar{y} - a_1\bar{x}$$

$$\text{Mean} = \frac{\text{sum of all obs}}{\text{no of obs}}$$

Simple linear Reg $y = a_0 + a_1 x \rightarrow (1)$

$$a_1 = \frac{360.2 - (4)(87.4)}{18 - (4)^2} = \frac{360.2 - 349.6}{18 - 16}$$

$$a_1 = 5.3 = \frac{10.6}{2} = 5.3 \rightarrow (2)$$

$$a_0 = \bar{y} - a_1\bar{x} \Rightarrow 87.4 - (5.3)(4)$$

$$87.4 - 21.2$$

$$a_0 = 66.2 \rightarrow (3)$$

Sub eqn(2) & (3) in eqn (1)

$$y = 66.2 + 5.3x$$

Ass.Sum:- A dataset consider 4 Instances of salary and expenditure data is given as shown table Apply linear Regression tech to Predict the expenditure with Salary 3 lakh

Salary (x)	
5	
1	
2	
7	
3	?

Multiple linear Regression Problem.

Age x_1	year x_2	Salary y
35	1	20
25	1	10
40	2	35
50	4	?

$y = a + b_1x_1 + b_2x_2 + \dots + b_kx_k$

$a_1, b_1, b_2 \dots$ - coefficient.

x_1, x_2 - Independent variable

y - o/p, dependent variable.

$$a = \bar{y} - b_1\bar{x_1} - b_2\bar{x_2}$$

$$y = a + b_1x_1 + b_2x_2$$

$$b_1 = \frac{(x_2^2)(x_1y) - (x_1x_2)(x_2y)}{(x_1)^2 (X_2^2) - (x_1x_2)^2}$$

$$b_2 = \frac{(x_1^2)(x_2y) - (x_1x_2)(x_1y)}{(x_1^2)(x_2^2) - (x_1x_2)^2}$$

x_1	x_2	y	x_2^2	x_1^2	x_1x_2	x_1y	x_2y
35	1	20	1	1225	35	700	20
25	1	10	1	625	25	250	10
40	2	35	4	1600	80	1400	70
x_1= 100	x_2= 4	y= 65	x_2^2= 6	x_1^2= 3450	x_1x_2= 140	x_1y= 2350	x_2y= 100

$$b_1 = \frac{(x_2^2)(x_1y) - (x_1x_2)(x_2y)}{(x_1^2)(x_2^2) - (x_1x_2)^2}$$

$$b_1 = \frac{(6)(2350) - (140)(100)}{(3450)(6) - (140)^2} \quad (6)$$

$$= \frac{14100 - 14000}{20700 - 19600} = \frac{100}{1100} = 0.09.$$

$$b_2 = \frac{(x_1^2)(x_2y) - (x_1x_2)(x_1y)}{(x_1^2)(x_2^2) - (x_1x_2)^2}$$

$$= \frac{(3450)(100) - (140)(2350)}{(3450)(6) - (140)^2}$$

$$= \frac{345000 - 329000}{20700 - 19600} = \frac{16000}{1100} = 14.54$$

Mean $\bar{y} = \frac{65}{3} = 21.6$

$$\bar{x_1} = \frac{100}{3} = 33.3$$

$$\bar{x_2} = 4/3 = 1.33$$

Sub b_1 & b_2 & mean value in a.

$$a = \bar{y} - b_1\bar{x_1} + b_2\bar{x_2}$$

$$= 21.6 - (0.09)(33.3) + (14.5)(1.33)$$

$$= 21.6 - 2.99 + 19.28$$

$$= 37.89.$$

$y = a + b_1x_1 + b_2x_2$ (x_1, x_2) value from Table.

$$= 37.89 + (0.09)(50) + (14.5)(4)$$

$$= 37.89 + 4.5 + 58.$$

$$= 100.36$$

Solve a sum by Linear Regression Least Square method:-

Equation of linear regression.

$$y = a + bx$$

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

$$\bar{y} = a + b\bar{x}$$

x	y	$x-\bar{x}$	$y-\bar{y}$	$x-\bar{x} + y-\bar{y}$	$(x-\bar{x})^2$
2	3	-3	-3.25	9.75	9
4	7	-1	0.75	-0.75	1
6	5	1	1.25	-1.25	1
8	10	3	3.75	11.25	9
$\bar{x}=5$	$\bar{y}=6.25$			19	20

eqn of linear Regression $y = a + bx$.

Here x is indedependent variable

y is dependent variable

a is intercept, b is slop or coefficient.

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

$$\bar{y} = a + b\bar{x}$$

$$b = \frac{19}{20} \quad \boxed{b = 0.95}$$

$$\bar{y} = a + b\bar{x}$$

$$a = \bar{y} - b\bar{x}$$

$$= 6.25 - (0.95)(5)$$

$$= 6.25 - 4.75$$

$$a = 1.5$$

$$y = a + bx$$

$$y = 1.5 + 0.95x$$

Predict the final exam grade of a student who received an 86 in the mid term exam.

$$y = 1.5 + 0.95(86) = 1.5 + 81.7 = 96.7$$

Assignment Problem:-

The following table shows the mid term and final exam grades obtained by the students in database course. Use the method of least squares to find the eqn for the prediction of students final exam grade based on the students midterm grade in the course. predict the final exam grade of student who received as 36 in the mid term exam.

x mid Term	y Final exam.
72	84
50	63
81	77
74	78
94	90
86	75
59	49
83	79
65	77
33	52
88	74
81	90.

BAYESIAN LINEAR REGRESSION:-

Bayesian linear Regression is a statistical technique that integrates linear regression with Baysian inference.

It accounts for uncertainity in model parameter by considering them as random variable and assigning probability distribution to them.

Bayesian linear Regression is particularly valuable in machine learning (ML) for several reason especially when dealing with incertainly incorporating prior knowledge and improving model interpretability.

Bayesian models provide more interpretable results because they produce distributions over parameters.

A traditional linear regression model would give you a single predicted price, but it wouldn't tell how confident the model is in that prediction.

Bayesian linear regression, on the other hand would provide a distribution of possible prices, reflecting the uncertainity in the maket. This probabilistic prediction is far more useful in making informed decision, such as risk Management.

Linear Regression

* To predict an outcome y (eg: income) based on an i/p x (eg: years of experience)
* The relationship is modeled as $y = B_0 + B_1 x + \epsilon$
* Where B_0 is the intercept, B_1 is the slope and ϵ is the error term.

Baysian Approach.

* Bayesian approach find a distribution of possible Parameters.
* Start with a prior belief about the Parameter and update this belief using the data to get a Posterior distribution.

$$\text{Posterior} = (\text{likelihood} * \text{Prior}) / \text{Normalization}.$$

Model Specification:-

Assume a linear relationship b/w the i/p & the o/p y; $y = B_0 + B_1 x + \epsilon$

where

B_0 is intercept,

B_1 is slope

ϵ is error term, typically assumed to be normally distributed. $\epsilon \sim N(0, \sigma^2)$

(2) Prior Distribution:-

Assign Prior distribution to the model parameters B_0 & B_1. These represent your initial belief about the parameters before seeing the data.

$$B_0 \sim N(\mu_0, \sigma_0^2)$$

$$B_1 \sim N(\mu_1, \sigma_1^2)$$

where μ_0, μ_1 are the mean & σ_0^2, σ_1^2 are the variation of the period.

(3) Likelihood:

The likelihood function represents the Probability of observing the data given the Parameter assuming normally distributed errors.

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2).$$

This is the likelihood of observing y_i given x_i & the parameter B_0 & B_1.

Posterior Distribution:-

Using Bayes theorem, combine the Prior and the likelihood to obtain the posterior distribution of the parameters.

$$P(B_0, B_1 | data) \propto P(data | (B_0, B_1) \times P(B_0) \times P(B_1)$$

Prior $P(B_0)$ and $P(B_1)$

likelihood $P(data | B_0, B_1)$

Posterior $P(B_0, B_1 | data)$

This formula provides the updated belief about the parameters after considering the observed data.

Prediction:-

To Predict the output y^* for a new x^* use the posterior distribution of the parameter.

$$y^* \sim N(\hat{B_0} + \hat{B_1} X, \sigma^2_{Predictive}).$$

Posterior means.

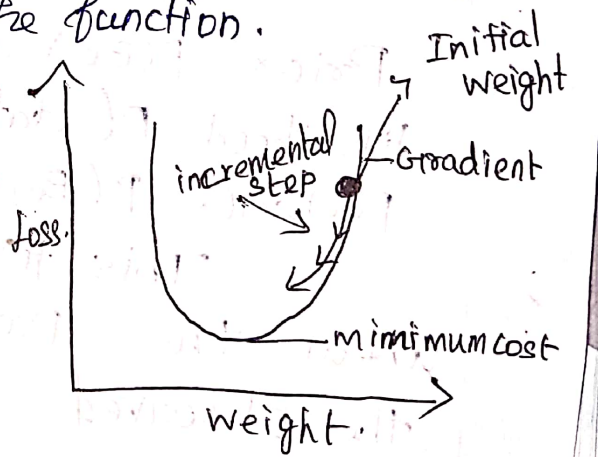
GRADIENT DESCENT:-

It is defined as one of the most commonly used iterative optimization algorithm of ML to train the ML and DL models. It helps in finding the local minimum of a function. (Reduce Cost) or (Closs function)

If we move towards a negative gradient or away from the gradient of the function at the current point, it will give the local minimum of the function.

Whereever we move towards a Positive gradient or towards the gradient of the function at the current point, we will get the Local maximum of the function.

The main objective of using a gradient descent algorithm is to minimize the cost function using iteration.



Predictive:- Accounts for both the uncertainity in the parameter and the noise in the data. This approach allows for the incorporation of prior knowledge and the estimation of uncertainity in the model parameter and prediction.

Adv

* Incorporates prior knowledge.
* Probabilistic prediction
* Natural regularization
* Model comparision
* Online learning.

DisAdv

* Computational cost
* Prior specification
* Interpretation
* Expertise
* Sensitivity to outliers.

Linear class is ML algorithm used to classify data into categories by drawing a linear boundary. It uses a linear fn to model the relationship b/w i/p features & target o/p

LINEAR CLASSIFICATION MODELS:-

Discriminate function:-

→ A classification algorithm that make it's classification based on a linear predictor function, combining a set of weight with the feature vector.

→ A linear classifier does classification decision based on the value of linear combination of characteristics.

→ Imagine that the linear classifier will merge into it's weights all the characteristics that define a particular class.

→ Linear classifiers can represent a lot of things but they can't represent everything.

→ The classic example of what they can't represent is the XOR function.

LDA is a supervised learning algorithm which means that it requires a labelled set of data points in order to learn the linear discriminant function.

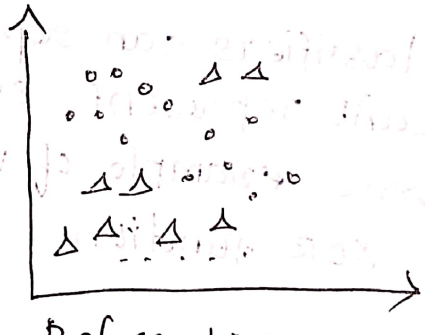
The main purpose of LDA is to find the line or plane that best seperates data points belonging to different classes.

The key idea behind LDA is that the decision boundary should be chosen such that it maximize the distance b/w the means of two classes while simultaneously minimizing the variance within each class data or within class scatter. This criterion is known as the fisher Criterian.

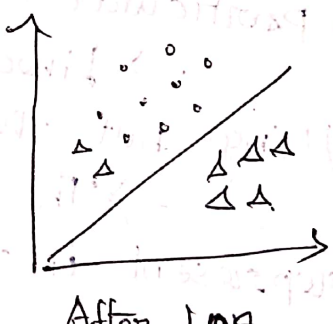
LDA is one of the most widely used ML algorithm due to its accuracy and flexibility

LDA can be used for a variety of task such as classification, dimensionality reduction and features selection.

Suppose we have two classes & we need to classify them efficiently, then using LDA, classes are divided as follow.



Before LDA



After LDA

Steps in LDA Algorithm:-

i) The first steps is to calculate the mean & Standard deviation of each of feature.

ii) within class scatter matrix and blw class scatter matrix is calculated.

(iii) These matrices are then used to calculate the eigenvectors and eigen values.

iv) LDA chooses the k eigenvectors with the largest eigenvalues to form a transformation matrix.

v) LDA uses this transformation matrix to transform the data into a new space with k dimension.

vi) once the transformation matrix transform the data into new space with ic dimension, LDA can then be used for classification or dimensionality reduction.

Probabilistic Discriminative Model:-

Discriminative model are a class of supervised ML Models which make predictions by estimating conditional probability $P(y|x)$. In order to use a generative model, more unknowns should be solved. one has to estimate probability of each class and Probability of observation given class.

For two-class classification problem, the posterior probability of class C_1 can be written as a logistic sigmoid action on a linear function of x.

$$P(C_1|x) = \sigma\left[\ln \frac{P(x|C_1)\,P(C_1)}{P(x|C_2)\,P(C_2)}\right] = \sigma(w^T x + w_0).$$

Logistic Regression:-

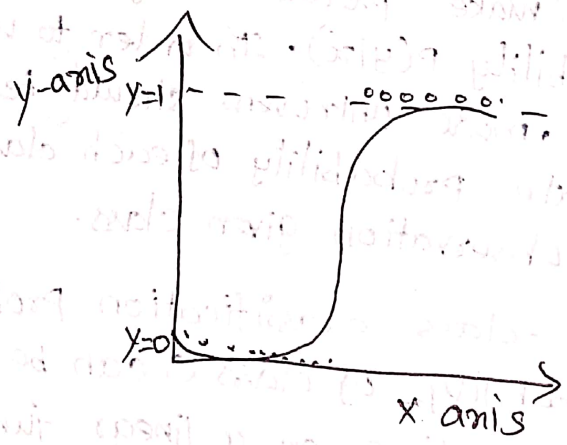
Logistic regression is a supervised learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event or observation. The model delivers a binary outcome limited to two possible outcomes: yes/no, 0/1 etc...

When considering the two class problem using a generative approach and under general assumption, the posterior probability of class C_1 is re-written.

: a logistic sigmoid on linear function of the feature vector $\phi = \phi(x)$

$$P(C_1|\phi) = y(\phi) = \sigma(w^T\phi), \text{ with } P(C_2|\phi) = 1 - P(C_1|\phi)$$

→ The logistic sigmoid function is defined as $\sigma(a) = \frac{1}{1+\exp(-a)}$, with $a = \ln \frac{P(\phi|C_1)\,P(C_1)}{P(\phi|C_2)\,P(C_2)}$



Generative Model:-

A generative Model is a statistical model of the joint probability distribution $p(x, y)$ on given observable variable x and target variable y.

Models with linear decision boundaries arise from assumption about the data.

In generative approach to classification we first model the class - conditional densities $P(x|C_k)$ and the class prior $p(C_k)$, and then we compute posterior probabilities $p(C_k|x)$ through Baye's theorem.

Naive Bayes:- (Refer unit:2)

i) Naive Bayes Algorithm
(ii) Baye's theorem
(iii) Working of Naive Bayes classifier
iv) Problem
v) App, Adv, Dis Adv.

Maximum Margin classifier:-

The maximal margin classifier is the optimal hyperplane defined in the (here) case where two classes are linearly seperable.

Given an $n \times p$ data matrix x with a binary response variable defined as $y \in [-1, 1]$, it might be possible to define a p-dimensional hyperplane.

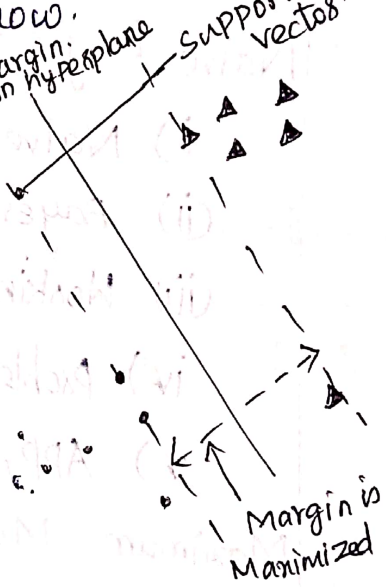
$$h(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$
$$= x_i^T \beta + \beta_0 = 0.$$

This separating hyperplane has the property that if β is constrained to be a unit vector, $\|\beta\| = \sum \beta^2 = 1$, then the product of hyperplane and response variable are positive perpendicular distances from the hyperplane, the smallest of which may be formed the hyperplane margin, M.

The maximal margin classifier is the hyperplane with the maximum margin max [M] subject to $\|\beta\| =$

A separating hyperplane rarely exist. In fact, even if a separating hyperplane does exist, its resulting margin is probably undesirably narrow.

Here is the maximal margin classifier the data set has 2 linearly seperable classes $y \in [-1, 1]$ described by 2 features x_1 & x_2. This code is unimportant - Just trying to produce the visualization.

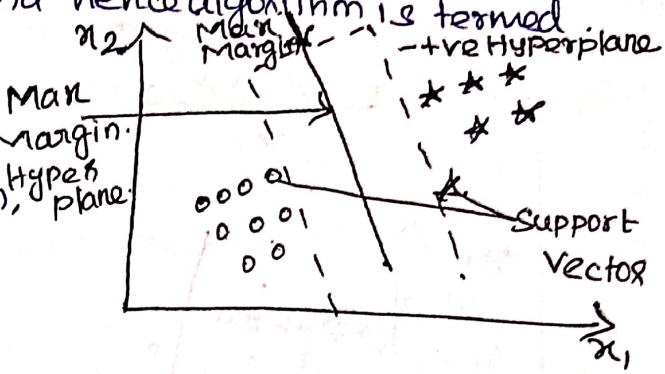


Support Vector Machinge

SVM is one of the most popular supervised learning algorithm, which is used for classification as well as regression problems. However, primarily it is used for classification problem in ML.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This bet decision boundary is called a hyperplane.

SVM chooses the extreme points/vector that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed support vector Maching



* SVM algorithm can be used for face detection, image classification text categorization etc.

Types of SVM:-

1) Linear SVM:- linear svm is used for linearly seperable data, which means if a dataset can be classified into two classed by using a single straight line, then such data is termed as linearly seperable data and classifier is used called as linear svm classifier.

Non linear SVM:-

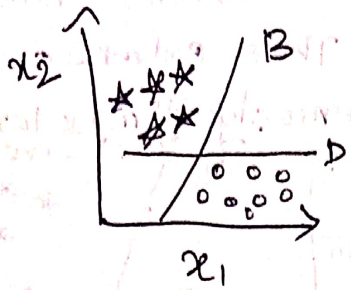
It is used for non linearly seperated data which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used called Non-linear SVM classifier.

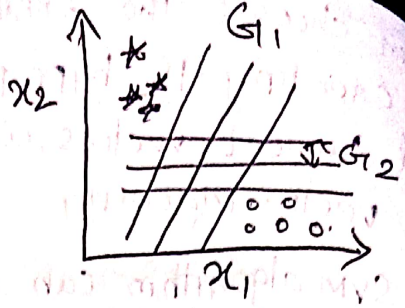
LINEAR SVM:-

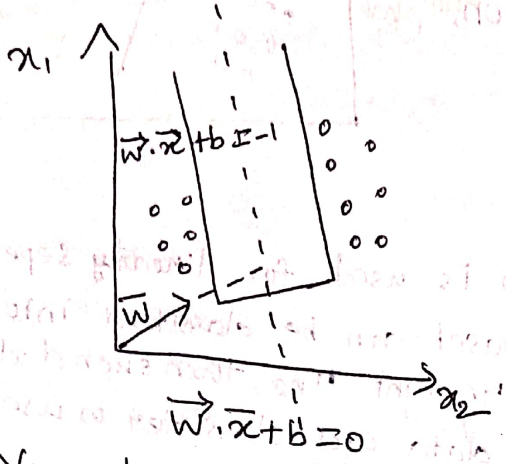
The Working of the SVM can be understood by using an example.

Suppose we have a dataset that has two tags (Green & blue) and the dataset has two features x_1 and x_2

We ~~cannot~~ want a classifier that can classify the pair $(x1, x2)$ of coordinates in either green or blue.







$$\max \frac{2}{\|w\|}$$

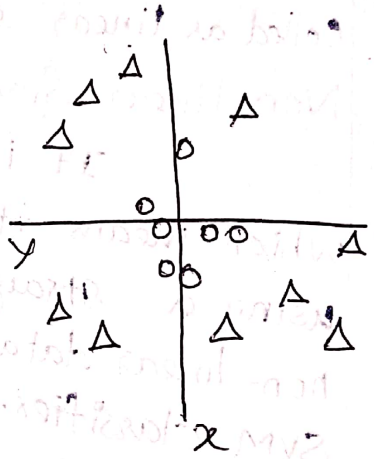
$(w \cdot x + b) \geq 1 \quad \forall x$ of class 1

$(w \cdot x + b) \leq -1 \quad \forall x$ of class 2

NON LINEAR SVM:-

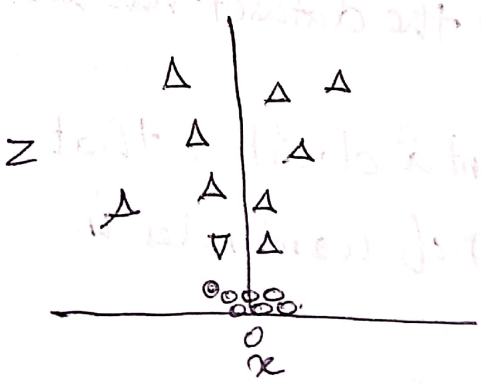
If data is linearly arranged then we can seperate it by using a straight line, but for non-linear data, we cannot draw a single straight line

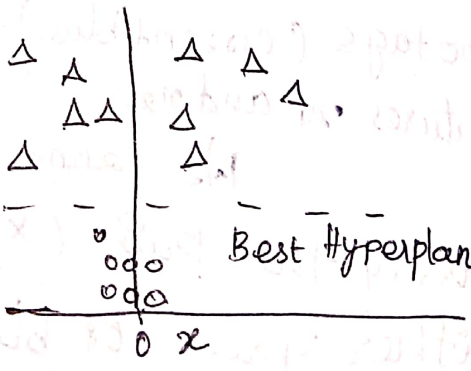
So to seperate these data points, we need to add one more dimension.



For linear data, we have used two dimension x and y, So for non-linear data, we will add a third dimension z.

It can be calculated as $z = x^2 + y^2$

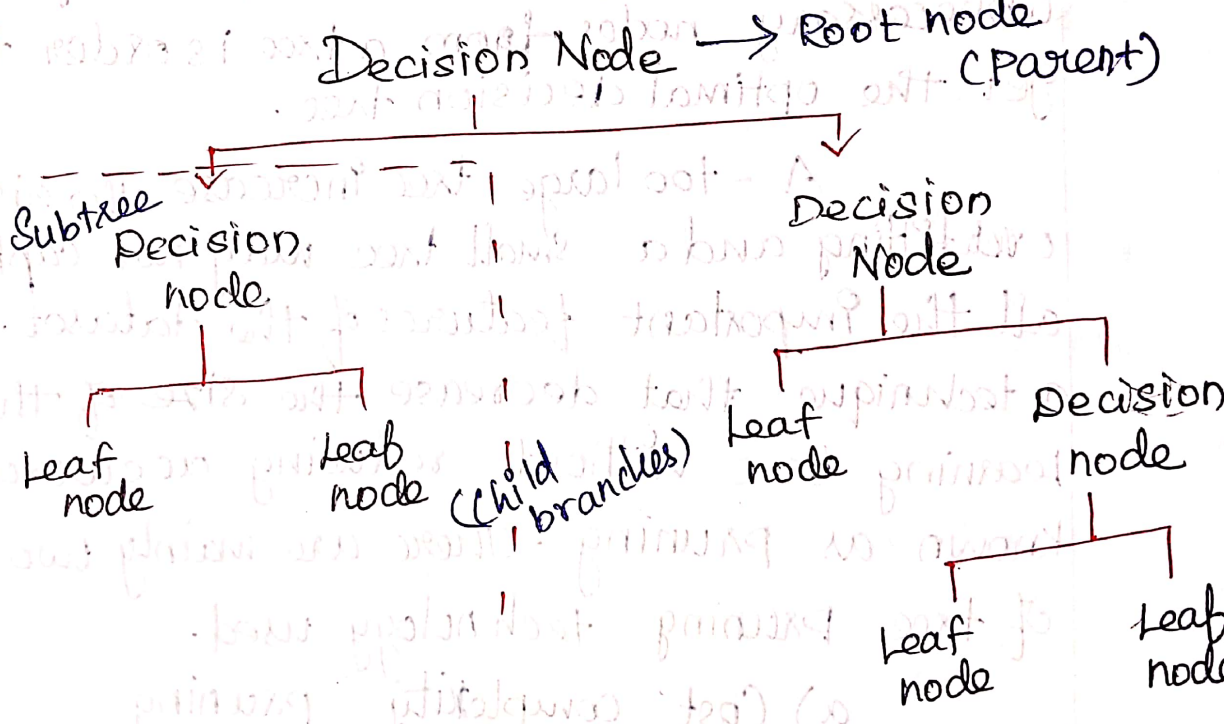




DECISION TREE:-

Decision Tree is a Supervised learning technique that can be used for both classification and regression problem, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represent the outcome.

In a Decision tree, there are 2 nodes which are the decision Node and leaf node. Decision nodes are used to make any decision and have multiple branches whereas leaf nodes are the output of those decision and do not contain any further branches.



Decision Tree Terminologies:-

1) **Root Node:-** Root node is from where the decision tree starts. It represents the entire dataset which further gets divided into two or more homogeneous sets.

2) **Leaf node:-** Leaf nodes are the final output node, and the tree cannot be ~~seperated~~ Segregated further after getting a leaf node.

3) **Splitting:-** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given condition.

Branch/Sub tree:- A tree formed by splitting the tree.

Pruning: Getting an optimal Decision Tree:-

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too large tree increase the risk of overfitting and a small tree may not capture all the important features of the dataset. Therefore a technique that decrease the size of the learning tree without reducing accuracy is known as pruning. There are mainly two types of tree pruning technology used.

a) Cost complexity pruning.

b) Reduced Error Pruning.

Adv	Disadv.
* Easy to understand and Interrupt	* overfitting.
* Handles Different Data types.	* Instability
* less Data preperation.	* Bias towards Dominant classes.
* Handles Non linear relationship.	* Greedy approach
* Feature Importance.	* Limited experiencess.

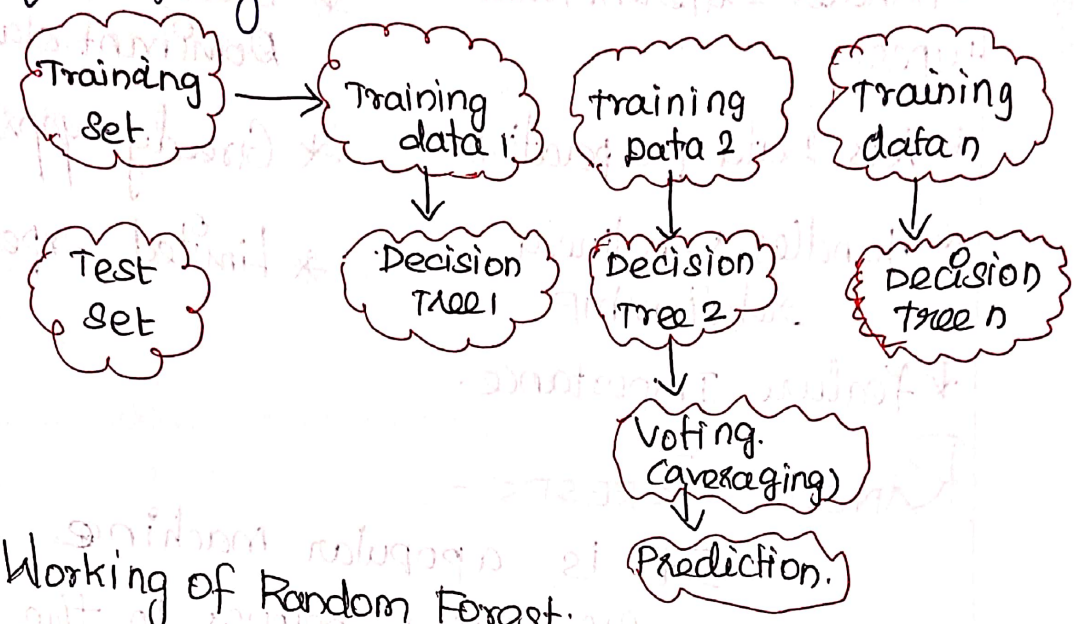
RANDOM FORESTS:-

It is a popular machine learning algorithm that belongs to the Supervised learning technique. It can be used for both classification and regression Problem in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classification to solve a complex problem and to improve the performance of the model.

Random forest is a classifier that contains a no of decision tree on various subset of the given dataset and takes the average to improve the predictive accuracy of the dataset. Instead of relying on one decision tree, the random forest takes the

Prediction from each tree and based on the majo Votes of prediction, and it predicts the final outpu

The greater number of trees in the forest leads to higher accuracy and Present the problem of overfitting.



Working of Random Forest:

Random forest work in two phase first is to create the random forest by combining N Decision tree, and second is to make prediction for each tree created in the first phase.

The working process can be explained in the below steps and diagrams:-

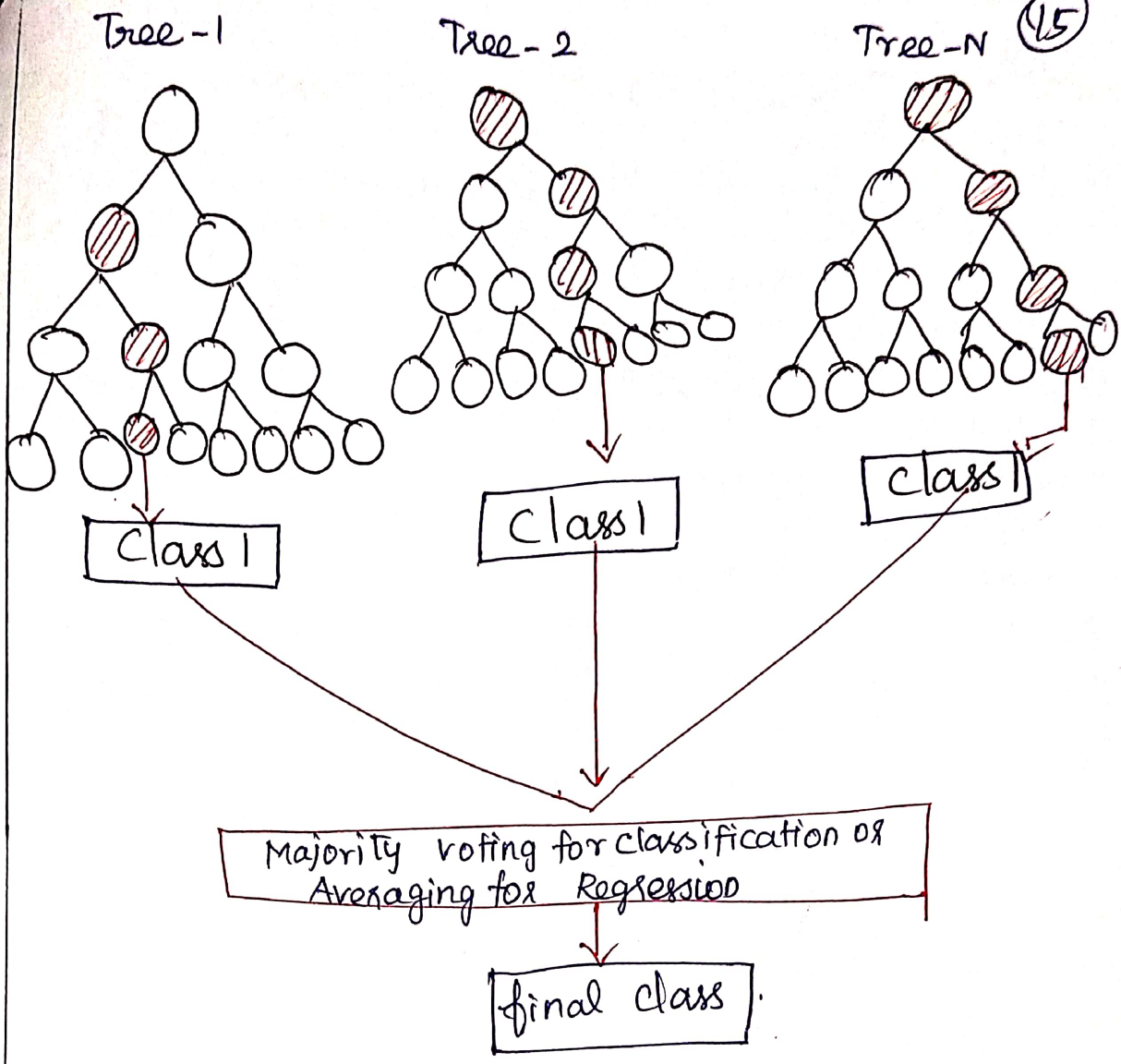
Step1: select random k data point from the training set.

Step2: Build the decision task tree associated with the selected data points.

Step3: choose the number N for decision trees that you want to build.

Step4:- Repeat Step1 & Step 2

Step5:- For new data points find the prediction of each decision tree and assign new data point to the category that win the majority votes.



Application: * Banking, * Medicine * Marketing.

Adv:-

* High Accuracy
* Robustness to overfitting
* Handle various datatypes
* Handle Missing values
* Feature importance.
* Scalability.

Disadv

* Complexity & interpretability
* computationally intensive
* Resource requirements
* Prediction speed.
* May not perform well on linear relationship.

Unit - II Probabilistic Reasoning

Acting under uncertainity - Bayesian Inference - Naive bayes models. Probabilistic Reasoning - Bayesian Networks - Exact inference in BN - approximate inference in BN - Causal Networks.

(1) Acting under Uncertainity :-

$A \rightarrow B$ means if A is true then B is true, if we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainity.

→ To represent uncertain knowledge, uncertain reasoning or Probabilistic reasoning is used.

Causes of uncertainity :-

Following are some leading causes of uncertainity to occur :

- → Information occured from unreliable sources.
- → Experimental Errors.
- → Equipment fault.
- → Temparature Variation.
- → Climate change.

General theory of decision Theory :-

Decision theory = Probability theory + Utility theory.

Unit - II Probabilistic Reasoning (1)

Acting under uncertainity - Bayesian Inference - Naive bayes models. Probabilistic Reasoning - Bayesian Networks - Exact inference in BN - approximate inference in BN - Causal Networks.

(1) Acting under Uncertainity :-

A → B means if A is true then B is true, if we are not sure about whether A is true or not then we cannot express this statement, this situation is called uncertainity.

→ To represent uncertain knowledge, uncertain reasoning or Probabilistic reasoning is used.

Causes of uncertainity :-

Following are some leading causes of uncertainity to occur :

- → Information occured from unreliable sources.
- → Experimental Errors.
- → Equipment fault.
- → Temparature Variation.
- → Climate change.

General theory of decision Theory :-

Decision theory = Probability theory + Utility theory.

Bayes' Rule :-

The product rule can actually be written in

$P(a \wedge b) = P(a|b)\ P(b)$ (or)

$P(a \wedge b) = P(b|a)\ P(a)$

Equating the two right-hand sides & dividing.

$$\boxed{P(b|a) = P(a|b)\ P(b) \ / P(a)}$$

this equation is known as Bayes rule or law.

Baye's rule becomes.

$P(\text{Cause} / \text{effect}) = P(\text{effect} | \text{cause})\ P(\text{cause}) \ / P(\text{effect})$

2) Bayesian Inference :-

→ Bayesian inference is a method of statistical inference in which Baye's theorem is used to update the probability for a hypothesis.

→ Bayesian Inference derives the posterior probability as a consequence of two probability & a "likelihood function" derived from a statistical method.

According to Baye's theorem,

$$P(H/E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

where,

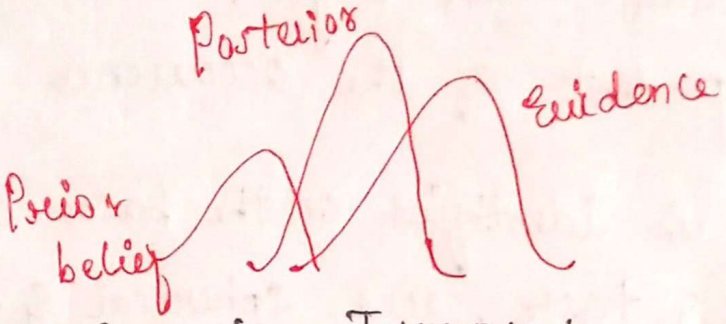
→ H stands for any hypothesis

→ P(H) is prior probability, estimate probability of exa hypothesis.

→ E is the evidence.

→ P(H/E) is the posterior probability.

→ P(E/H) is called likelihood.



Types of Bayesian Inference :-

1) Exact inference in BN - Guessing exact.

Example:-

→ Junction Tree algorithm (JTA)

- Belief Propagation method.
- Sum of Product method.

2) Approximate inference in BN - Guessing approximate.

- Stochastic &
- Deterministic.

Example 1) Markov chain Monte Carlo Algorithm.

Applications :-

Including Science, Engineering, Philosophy, medicine, sport & law.

3) Naive Baye's models:-

→ It is used supervised learning algorithm.

→ used for solving classification problems (Text).

→ It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Why it is called Naive Bayes?

Naive: Independent of the occurrence of other features.

Fruit is identified on the bases of colour, shape & taste, red, spherical & sweet fruit.

Bayes: Depends on the principle of Baye's theorem.

Baye's theorem:-

→ known as baye's rule or Baye's Law.

→ Determine the probability of a hypothesis with prior knowledge.

→ Depends on the conditional probability.

formula:-

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

where,

$P(A|B)$ → Posterior (target) Probability

Hypothesis A on the observed event B.

$P(B)$ → Marginal Probability.

Probability of evidence.

P(B/A) is likelihood Probability.

Evidence given that the probability of a hypothesis is true.

P(A) is prior probability.

hypothesis before observing the evidence.

Bayes theorem can be rewritten as:

$$P(Y/X) = \frac{P(X/Y)\,P(Y)}{P(X)}$$

P(X/Y) → likelihood; P(Y) → class prior probability; P(Y/X) → Posterior probability; P(X) → predictor prior probability

$$\text{posterior} = \frac{\text{Prior} \times \text{likelihood}}{\text{evidence}}$$

$$P(H/E) = \frac{P(E/H) * P(H)}{P(E)}$$

likelihood → P(E/H); prior → P(H); posterior → P(H/E); P(E) → prior probability evidence

Steps in Naive Bayes model:-

S_1: Calculate the prior probability for given class labels.

By converting the given dataset into frequency labels.

S_2: find likelihood.

S_3: Use Bayes theorem to calculate posterior.

S_4: Higher Probability.

Example :-

If the weather is sunny, then the player should play or

Solution :-

First Consider the below dataset.

	Outlook	Play
0	Rainy	Yes
1.	Sunny	Yes
2.	Overcast	Yes
3.	Overcast	Yes
4.	Sunny	Yes
5.	Overcast	Yes
6.	Rainy	Yes

	Outlook	Play
7.	Sunny	Yes
8.	Rainy	No
9.	Sunny	No
10.	Sunny	Yes
11.	Rainy	No
12.	Overcast	Yes
13.	Overcast	Yes.

frequency table for weather Conditions.

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	4

likelihood table weather Conditions:

Weather	No	Yes	
Overcast	0	5	5/14 = 0.35
Rainy	2	2	4/14 = 0.29
Sunny	2	3	5/14 = 0.35
All	4/14 = 0.29	10/14 = 0.71	

Applying Baye's theorem

$P(Yes|sunny) = P(sunny|Yes) * P(Yes) / P(sunny)$

$P(sunny|Yes) = 3/10 = 0.3$

$P(sunny) = 0.35$

$P(Yes) = 0.71$

So $P(Yes|sunny) = 0.3 * 0.71 / 0.35 = 0.60$

$P(No|sunny) = P(sunny|No) * P(No) / P(sunny)$

$P(No|sunny) = 2/4 = 0.5$

$P(No) = 0.29$

$P(sunny) = 0.35$

So $P(No|sunny) = 0.5 * 0.29 / 0.35 = 0.41$

Hence on a sunny day, player can play the game.

Advantages of Naive Bayes classifier :-

→ Fast & Easy ML algorithm.

→ Used for binary as well as multi-class classification.

→ Performs in multi-class prediction.

→ Text classification problems.

Disadvantages :-

All feature are independent or unrelated.

Applications:

-> used for Credit scoring
-> used in medical data classification
-> used in real-time prediction.
-> used in text classification (Span filtering. Sentiment analy

Types of Naive bayes model:-

3 types:

* Gaussian * multinomial * Bernoulli

Gaussian:

->features follow a normal distribution.
-> predictor take continuous values instead of discreate.

multinomial:-

->used when data is multinomial distributed
-> used for document classification problems.
-> Category such as sports, politics, education etc...

Bernoulli:-

-> Similar to the multinomial classifier.
-> But the predictor variables are independent boolean variables.
-> If a particular word is present or not in a document.

Probabilistic Reasoning:-

* Probability can be defined as a chance that an uncertain event will occur. It is the numerical measure of the likelihood that an event will occur.
* Value of probability always remains between 0 & 1.

$0 \leq P(A) \leq 1$, where $P(A)$ is the probability of an event A.

$P(A) = 0$, indicates total uncertainty in an event A

$P(A) = 1$, indicates total certainty in an event A.

Axioms of probability:-

→ Given a set U (universe)

A probability function is a function defined over the subsets of U that maps each subset to the real numbers.

→ $P(U) = 1$, $P(A)$ $[0, 1]$

→ $P(A \in B) = P(A) + P(B) - P(A \cap B)$

→ if $A \cap B = \{\}$ then $P(A \cup B) = P(A) + P(B)$

Probability formula:-

$$\text{Probability of occurrence} = \frac{\text{No of desired outcomes}}{\text{Total no of outcomes.}}$$

$P(\neg A)$ = probability of a not happening

$P(\neg A) + P(A) = 1$

Event: Each possible outcome of a variable

Sample space: Collection of all possible events.

Random Variable: Represents the event & objects

Prior probability: Computed before observing new information.

Posterior Probability: Combination of prior probability and new information.

Conditional Probability:-

* occurring an event when another event has already happened.

* Let's suppose, we want to calculate the event A when event B has already occured, "the probability of A under the conditions of B", it can be written as,

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

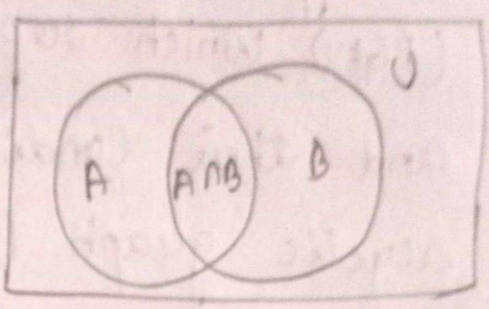
where,

$P(A \cap B)$ = Joint probability of A & B.

$P(B)$ = Marginal probability of B.

If the probability of A is given and we need to find the probability of B, then it will be given as,

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



Example :-

In a class, there are 70% of the students who like English & 40% of the students who like English & maths and then what is the present of students those who like english also like maths?

Solu :-

Let A is an event data that a students likes maths.

B is an event that a student likes english.

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

$$= \frac{0.4}{0.7}$$

$$= 57\%$$

Hence, 57% are the students who like english also like maths.

5) Bayesian Network:-

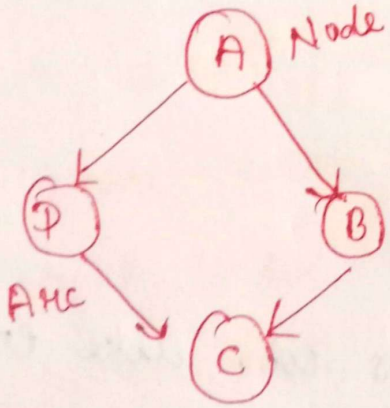
Bayesian n/w is a probabilistic graph. (PGM) which represents a set of values and their conditional dependencies using a acyclic graph (DAG).

Bayesian n/w : Directed acyclic graphs → Causal Structure.

Markov n/w : undirected graph → General dependencies.

Directed Acyclic Graph:-

1) Node represent random variable
2) Arcs represent direct Influence.
3) Nodes have conditional probability table.



→ A, B, C, D are random variables represented by the nodes of the n/w graph.

→ Node B, which is connected with node A by a directed arrow

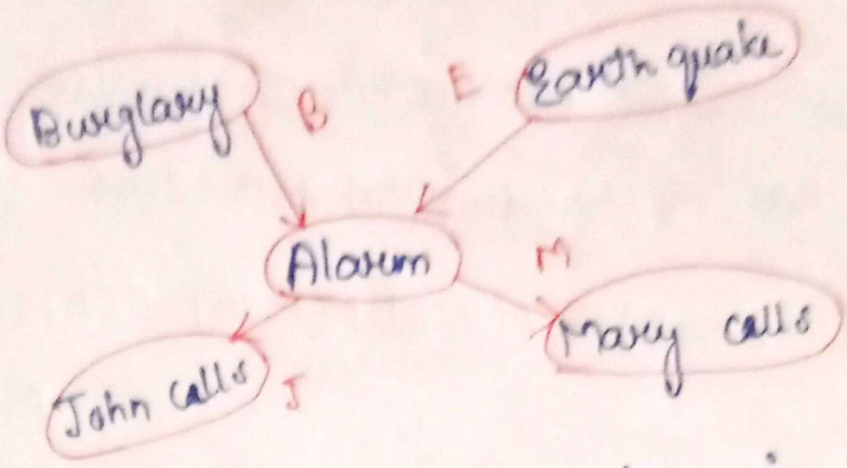
→ A is called parent of Node B.

→ Node C is independent of Node A.

Baysian network has mainly two components.

i) Causal Component

ii) Actual Numbers.

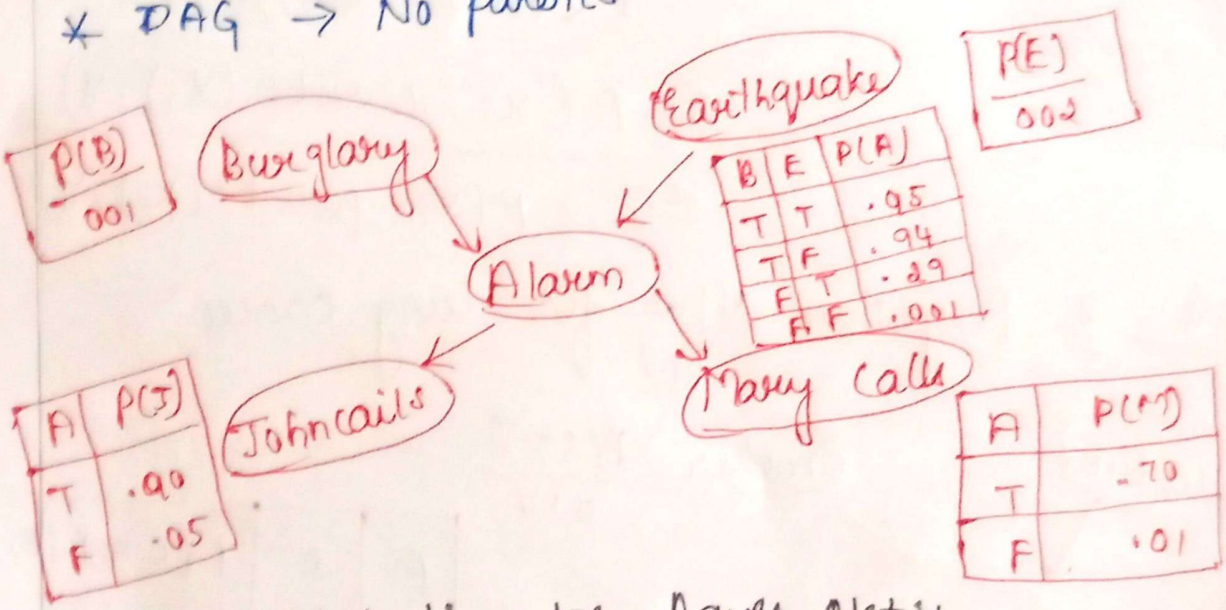


List of all events occurring in this nlw.

Conditional Probability Tables (CPT):-

* Each node has CPT that gives the probability of each of its values given every possible combination of values for parents.

* DAG → No parents.



Joint distribution for Bayes Nets:-

Joint distribution can be expressed as product of local conditional probabilities.

Bayesian Nlw implicity defines a joint distribution.

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(x_i))$$

Example:

Alarm has sounded but neither a burglary nor an earthquake has occured and both John and Mary call.

$$P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) = P(J|A)\, P(M|A)\, P(A|\neg B \wedge \neg E)\, P(\neg B)\, P(\neg E)$$

$$= 9.0 \times 7.0 \times 0.001 \times 0.999 \times 0.998$$

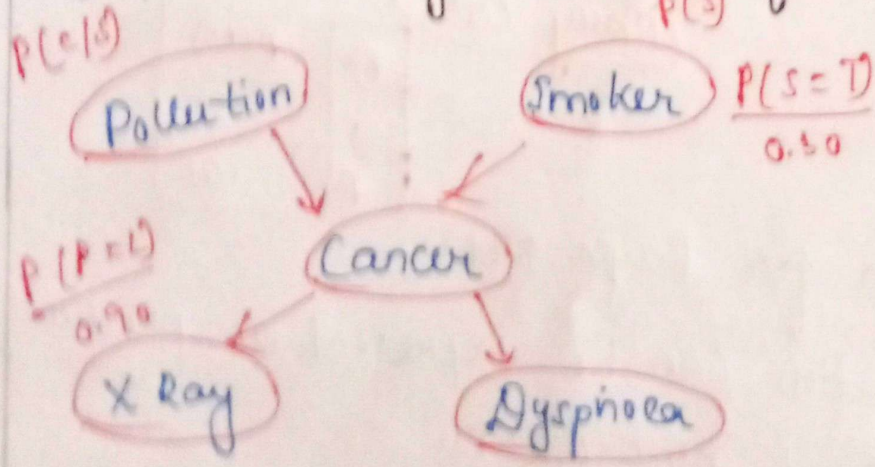
$$= 0.0062$$

Conditional probabilities can be computed from the joint distribution as follows.

$$P(x_i \mid \text{Parents}(X_i)) = \frac{P(x_i, \text{Parents}(X_i))}{P(\text{Parents}(X_i))}$$

$$= \frac{\Sigma_y P(x_i, \text{parents}(X_i), y)}{\Sigma_{x_i, y} P(x_i', \text{parents}(x_i), y)}$$

Example: 2 Bayesian N/w for lung cancer.



P	S	P(C=T | P,S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

chain rule:-

Reformulate the rule use of conditional probabilities.

$P(X_1, \ldots X_n) = P(X_n | X_{n-1}, \ldots X_1) P(X_{n-1}, \ldots X_1)$

Repeat process reduction of conjucatesve probabilities to a conditional dependency and a smaller conjunction

Final a big products.

$P(X_1, \ldots X_n) = P(X_n | X_{n-1}, \ldots X_1) P(X_{n-1} | X_{n-2}, \ldots X_1) \ldots$

$P(X_2 | X_1) P(X_1) = \prod_{i=1}^{n} P(X_i | X_{i-1}, \ldots X_1)$

This identity is called the chain rule,

Applications:

i) Prediction ii) Anomaly detection iii) Reasoning

iv) Time Series prediction v) Decision making under uncertainty.

6) Exact inference in BN:-

To compute the posterior probability distribution for a set of query variables, given some observed event usually, some assignment of values to a set of evidence variable.

X → Query variable

E → Set of evidence variable

e → particular observed event

Y → denotes the hidden (non query) variables.

Then, the complete set of variables is $\{X$

posterior probability distribution $P(X|e)$.

Inference by enumeration:-

Any conditional probability can be computed summing terms from full joint distribution.

$$P(X|e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y).$$

Consider the query.

$P($Burglary | John calls$)$ = true, Mary calls = true$)$

Burglary n/w.

$$P(B|j, m) = \alpha P(B, j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a).$$

CPT.

$$P(b|j, m) = \alpha \sum_a \sum_e P(b)\, P(e)\, P(a|b, e)\, P(j|a)\, P(m|a)$$

Product of probability will be $O(2n)$.

Complexity $O(n\,2n)$.

Enumeration - Ask algorithm:-

ENUMERATION - Ask algorithm, evaluates these expression trees using depth first, left-to-right recursion.

Space Complexity - No. of. Variables.

Time Complexity - $O(2n)$ better than $O(n\,2n)$.

nction ENUMERATION - ASK (X, e, bn) returns a distribution over X.

inputs: X, the query variable.

e, observed values for variables E.

bn, a bayes net with variable vars.

Q(x) ← a distribution over x, initially empty. for each value x_i of X do

$Q(x_i)$ ← ENUMERATE - ALL (vars, ex_i)

where e_{x_i} is e extended with $X = x_i$

return NORMALIZER (Q(x))

function ENUMERATE - ALL (vars, e) returns a real No

if EMPTY? (vars) then return 1.0.

V ← FIRST (vars)

if V is an evidence variable with value v in e.

then return P^ (v | Parents (V)) x ENUMERATE - ALL (REST (vars),

else return $\sum_v$ P(v | parents (v)) x ENUMERATE - ALL (REST (1).

where,

e_v is extenderded with V = v

Eg: $f_1(a,b) \rightarrow f_2(b,c) = f(a,b,c)$.

A	B	$f_1(A,B)$
T	T	.3
T	F	.7
F	T	.9
F	F	.1

B	C	$f_2(B,C)$
T	T	.2
T	F	.8
F	T	.6
F	F	.4

A	B	C	
T	T	T	.
T	T	F	.3 x
T	F	T	.7 x .6
T	F	F	.7 x .4
F	T	T	.9 x .2 =
F	T	F	.9 x .8 = .72
F	F	T	.1 x .6 = .06
F	F	F	.1 x .4 = .04

Clustering Algorithm:-

→ Variable elimination algorithm is simple.

→ Computation of posterior probabilities $O(n_2)$

→ using clustering algorithms, this can be reduced to $O(n)$.

→ Multiply connected n/w can be converted into a polytree.

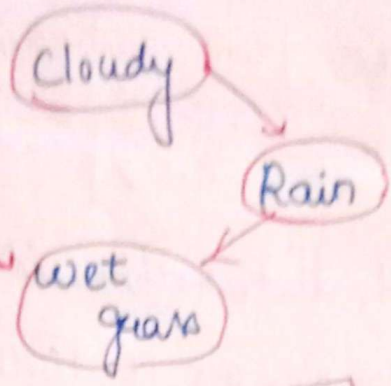
by combining Sprinkler & rain node into clustel node called Sprinkler + Rain.

→ Two boolean nodes replaced by a mega-node.

4 possible Values : TT, TF, FT, FF.

→ mega-nodes has only one parent.

P(C) = 5



	P(S)
T	.10
F	.50

C	P(R)
T	.80
F	.20

S	R	P(W)
T	T	.99
T	F	.90
F	T	.90
F	F	.00

P(C) = 5

S+R	P(W)
T T	.99
T F	.90
F T	.90
F F	.00

	P(S+R = *)			
C	TT	TF	FT	FF
T	.03	.02	.72	.18
F	.10	.40	.10	.40

1) Approximate Inference in Bayesian n/w:-

→ Approximate answers whose accuracy depends on the no samples generated.

→ Sampling applied to the Computation of posterior probabilities.

Two families of algorithm:-

i) Direct Sampling

ii) Markov chain Sampling

Direct Sampling methods :-

→ Generation of samples from a known probability

Ex : P(coin) = ⟨0.5, 0.5⟩

→ Sampling from this distribution is exactly like the coin: with probability 0.5 it will returns heads, and with probability 0.5 it will return tails.

function - PRIOR-SAMPLE (bn) return an event sampled from the prior specified by bn.

inputs: bn, a Bayesian n/w specifying joint distribution. $P(X_1, \ldots, X_n)$

$x \leftarrow$ an event with n elements.

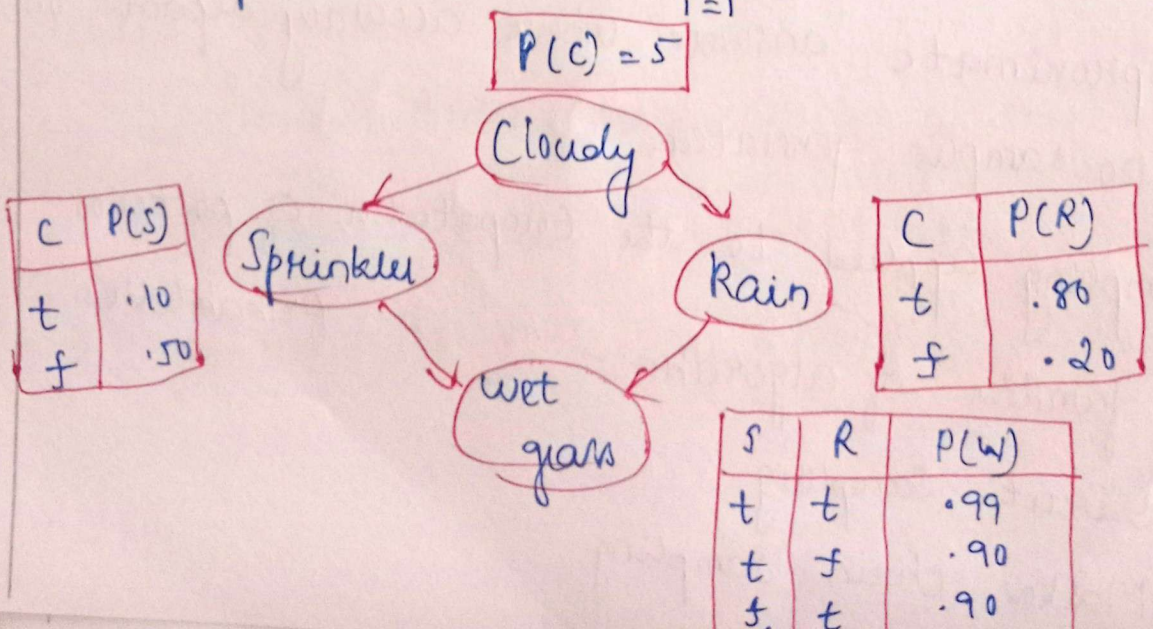
for each variable X_i in $X_1, \ldots X_n$ do

$x[i] \leftarrow$ a random sample $P(x_i \mid parents(x_i))$

return x.

PRIOR - SAMPLE :

$$S_{PS}(x_1, \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))$$



h Sampling step depends only on the parent values Sps

$(x_1, \ldots x_n) = P(x_1, \ldots x_n)$

$$\lim_{N \to \infty} \frac{NPs(x_1, \ldots x_n)}{N} = Sps(x_1, \ldots x_n) = P(x_1, \ldots x_n)$$

Sps (true, false, true, true) = 0.5 x 0.9 x 0.8 x 0.9

= 0.324.

Rejection Sampling :-

function REJECTION – SAMPLING (X, e, bn, N) returns an estimate of P(X/e).

inputs : X, the query variable.

e, observed value for variable E.

bn, a Bayesian n/w

N, the total no of samples to be generated local variable N a vector of counts for each value of X, initially zero.

for j = 1 to N do

X ← PRIOR – SAMPLE (bn)

if X is consistent with e then,

N(x) ← N[x] + 1 where X is the value of X in X.

return NORMALIZE (N).

Rejection Sampling Algorithm:-

Let $P(x|e)$ be the estimated distribu...

$$\hat{P}(x|e) = \alpha N P_S(x,e) = \frac{N P_S(x,e)}{N P_S(e)}$$

$$\hat{P}(x|e) \simeq \frac{P(x,e)}{P(e)} = P(x|e).$$

Rejection Sampling produce a consistent estimate of true probability

Estimate P(Rain / Sprinkler = true), using 100 samples of the 100 that we generate, suppose that 73 have Sprinkler = false and are rejected, while 27 have Sprinkler = true of the 27, 8 have rain = true & 19 have rain = false.

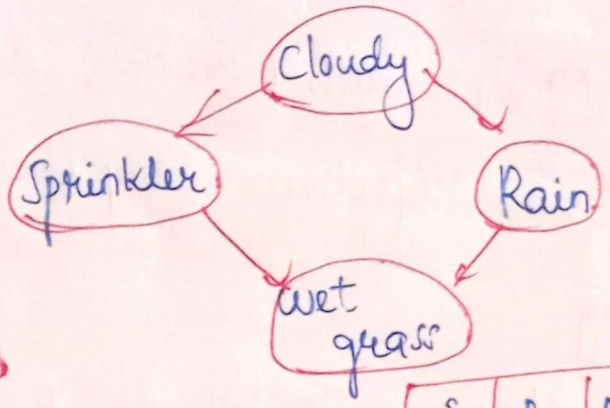
P (Rain | Sprinkler = true) NORMALIZE (8, 19>) = < 0.296, 0.704 > 26.

Likelihood Weighting:-

→ Likelihood weighting avoids the inefficiency of rejection sampling.

→ Generates only events that are consistent with the evidence.

P(C) = 5



C	P(S)
t	.10
f	.50

C	P(R)
t	.80
f	.20

S	R	P(W)
t	t	.99
t	f	.90
f	t	.90
f	f	.00

Cloudy is an evidence variable ← Wx P[Cloudy = true) = 0.5

Sprinkler is not an evidence variable ← P[Sprinkler | Cloudy = true)

weight for a given sample X is the product.

$$w(z,e) = \prod_{i=1}^{m} P(e_i \mid \text{parents}(E_i))$$

$$S_{WS}(z,e)\, w(z,e) = \prod_{i=1}^{l} P(z_i \mid \text{parents}(Z_i)) \prod_{i=1}^{m} P(e_i \mid \text{parents}(E_i))$$

Calculated as follows,

$$\hat{P}(x \mid e) = \alpha \sum_{y} N_{WS}(x,y,e)\, w(x,y,e)$$

$$\simeq \alpha' \sum_{y} S_{WS}(x,y,e)\, w(x,y,e)$$

$$\simeq \alpha' \sum_{y} P(x,y,e)$$

$$= \alpha' P(x,e) = P(x \mid e)$$

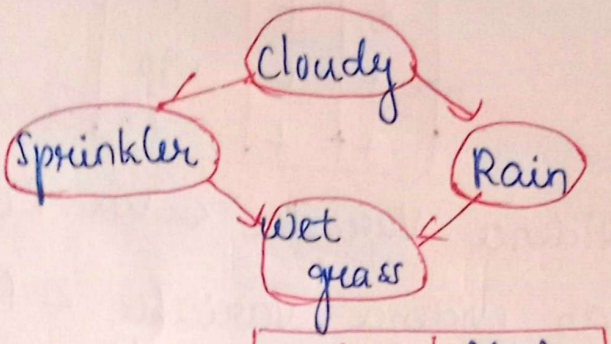
Hence, likelihood weighting returns consistent estimates.

Inference by Markov chain simulation

→ Markov chain Monte Carlo (MCMC) alg. quite differently from rejection sampling and likelihood weighting.

→ MCMC state change similar to SA.

P(C) = 5



C	P(S)
t	.10
f	.50

C	P(R)
t	.80
f	.20

S	R	P(W)
t	t	.99
t	f	.90
f	t	.90
f	f	.00

P(Rain/Sprinkler) = true, wet grass = true

Initial state is [true, true, false, true].

Cloudy :-

P(Cloudy | Sprinkler = true, Rain = false)

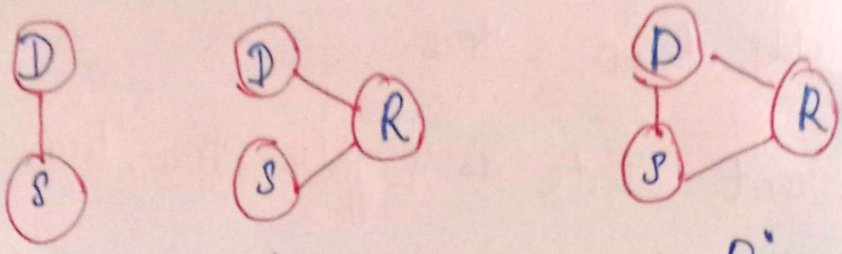
Suppose the result is Cloudy = false.

Then the new current state is [false, true, false, true].

Rain :-

Given the current values of its Markov blanket.

Types of cause :-



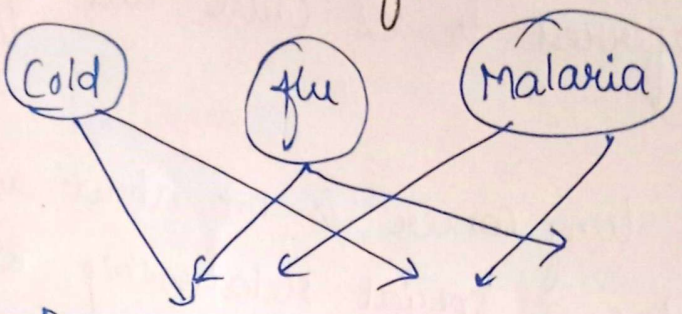
a) Direct Cause

b) Common Cause

c) Directed + Common

Example :-

Causal N/w for disease & symptoms :-

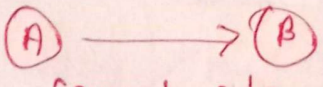


Causal Bayesian n/w with causes (diseases) Cold, flu and malaria and effects (symptoms) Nausea & Headache.

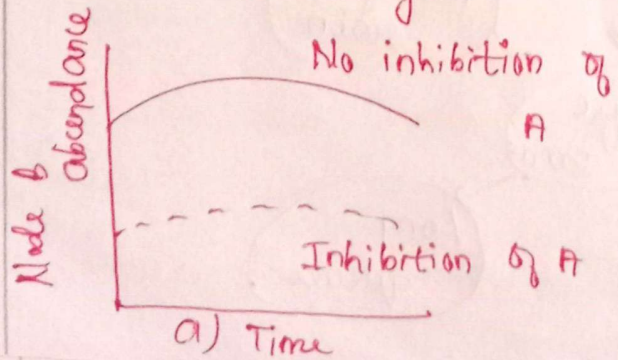
Types of Causal N/w :-

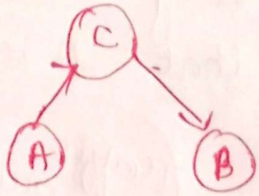
a) Directed Causal edge may represent direct effect

Inhibitation of parent node A can change the abundance of the child node B.

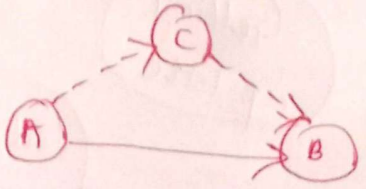


Causal edge





Unmeasured node



Indirect effect (b)

indirect causal edge may represent indirect effect:-

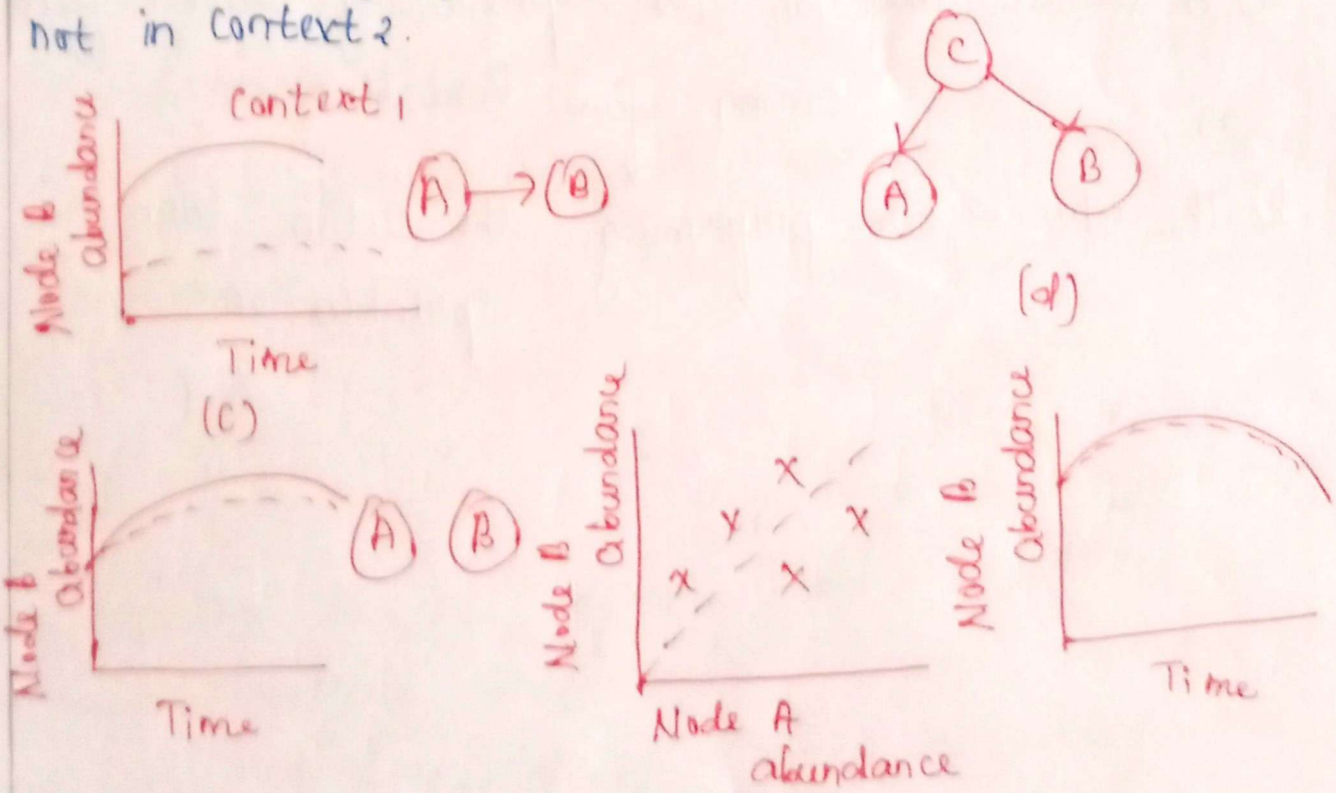
→ That occur via unmeasured intermidiate node

→ If node A causally influences node B via measured node C, the causal n/w should contain edges from A to C from C to B.

→ If node C is not measured.

c) Causal edges depend on biological Context:-

Causal edge from A to B appears in Context 1 not in Context 2.

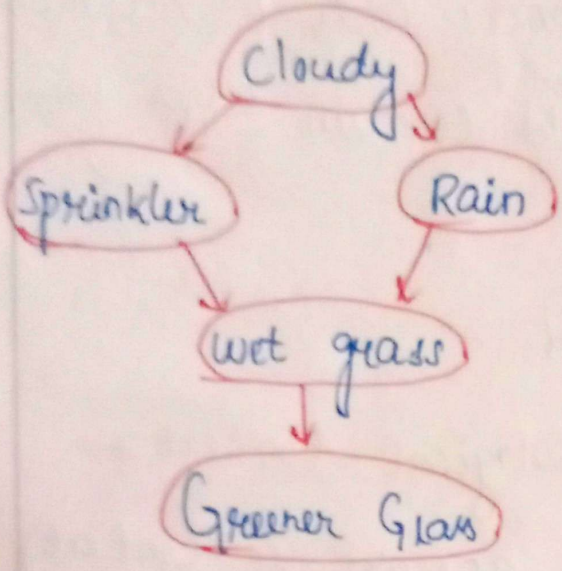


d) Correelation & Causation:-

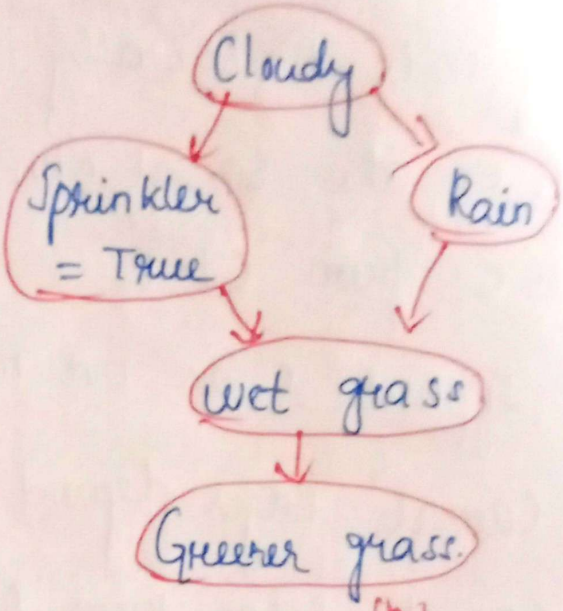
Node A & B are correlated owing to regulation by the same node (C) but in this example no sequence of mechanistic events links A to B and thus inhibition of A does not change the abundance of B.

There is no causal edge from A to B

Another Example:-



a)



(b)

a) A causal Bayesian n/w representing cause-effect among five variables.

b) The n/w after performing the action "turn Sprinkler on".

U23ITT62

ARTIFICIAL INTELLIGIENCE AND MACHINE LEARNING

UNIT-I PROBLEM SOLVING

Introduction to AI - AI Application - Problem Solving agents - Search algorithm - Uninformed Search Strategies - Heuristic Search Strategies - Local Search and optimization Problems - adversarial Search - Constraint Satisfaction Problem.

L1: Introduction to AI

AI is the branch of cs focused on building machines and ability of Computer to do task like human intelligence, such as making decision, Problem solving etc..

⟹ father of AI - John McCarthy.

Here, Artificial defines, "man made" and Intelligence defines "thinking Power".

Tasks & foundation of AI:-

1) Understanding Language (NLP)
   ex:- Chat GPT, Google translate.
2) Learning from data (ML)
   ex:- Netflix recommendation based on watching
3) Recognizing images or objects (computer vision)
   ex: Face recognizion, Imag classification.
4) Making Decision:-
   ex: Self driving car.

5) Reasoning And Problem ...
ex: Games like chess.

Tools In AI :-

1) Programming language.
   - Python (Most popular lang for AI).
   - R (used. in statistics and data analysis)
2) Machine Learning Lib (Build model from data)
   - Tensorflow (Deep learning, Neaural Network)
   - pytorch (Research friendly deep learning)
   - Matplot (Grap Rep).
3) Computer vision tool (Image Reco).
   - open cv (Image Processing).
4) Natural Language Processing (understand human lang)
   - Spacy (fast NLP tasks)
5) Reinforcement learning tools (Games/Robotic
   - open AI Gym (Train agents).
6) Data & visualization tools.
   - Pandas (Data analysis).
   - Matplot (Seaborn Data vaise Graph)
   - Numpy (Numerical computing).

Advantages of AI :-

→ High Accuracy with less error & efficiency
→ 24/7 Availability
→ High Speed / faster decision making.
→ Reduce human Risks.
→ useful as a public utility.

Dis Advantage of AI:-

→ High cost
→ Job Displacement
→ Lack of human Judgment
→ Data Privacy issues
→ No feelings and emotions.

AI Application:-

1) Healthcare
   - Disease Diagnosis (cancer, heart)
   - Medical imaging (x-ray, MRI, ECG, EEG)
2) Robotics:-
   - Autonomous Robots (Industry, delivery)
   - Surgical Robots (operation).
   - Service Robots (hotels)
3) E-Commerce:
   - Recommendation
   - Virtual Shopping (Amazon).
4) Smartphone:-
   - Face Recognition in clock.
5) Self driving car. (Autonomous vehicles)
6) Banking a finance
   - fraud detection
   - Credit Scoring.
7) Education:
   - Personalize learning.
8) NLP
   - Translation
   - voice to text
   - Sentiment Analysis.

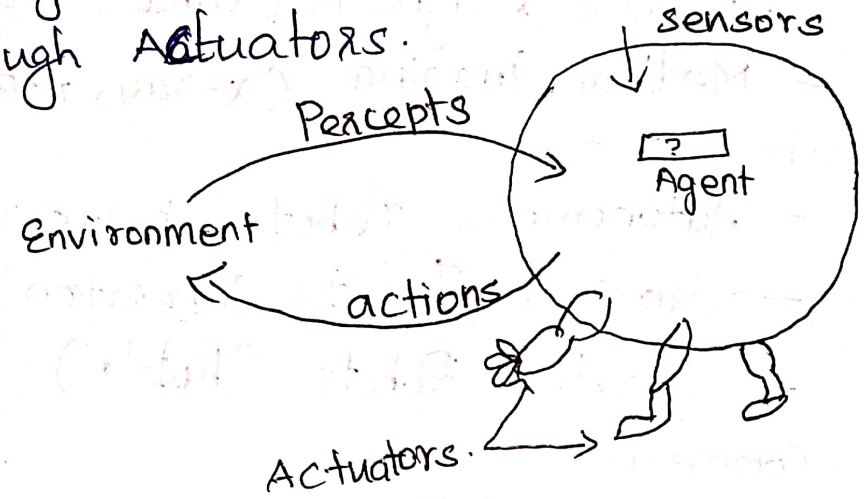
9) Government & Public services.
- Traffic System.
- Public safety monitoring.

Agent function: Input O/P Sequence to

L2: Problem Solving Agent:-

⇒ Agent and Environment.

An Agent is anything that can be viewed as perceiving its environment through sensors & acting upon that environment through Actuators.



$F: P^* \rightarrow A$

F:- Agent function.

P - Percept Sequence

A - Action.

Sensor:- used to observes the current environment.

Actuators: Acting a environment through. actuators.

Ex:- In human Agent: eye, ears, and other. organs are sensors, and hands, legs. month & other body parts for actuators.

In Robotic Agent:- Cameras and infrared range finder for sensors and various motors for actuators.

⇒ Input devices - sensor.
O/P devices - Actuators.

Agent function: It is used to maps the percept sequence to an action.

Agent are having dataset that contain (historical data of environment) + (map the current environment status.

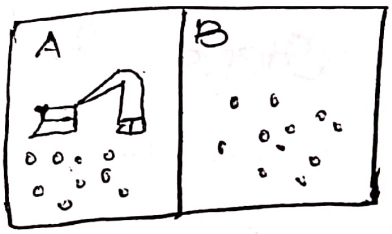
Agent Program:-

The Agent function for artificial agent will be implemented by an agent program.

The agent program is a concrete implementation, running on the agent architecture.

Vaccum Cleaner world:-

In this we have 2 location. one is A & B.



* Percepts :- location, content (A, B) eg [A, Dirty].

* Action :- left, Right, Suck, No operation.

* Tabulation of vaccum Agent function :-

Percept Sequence	Actions
A, clean	Right
A, Dirty	Suck
B, clean	left
B, Dirty	SUCK
If both are clean	No operation.

Types of Agent:-

* Simple Reflex Agent
* Model based reflex Agent
* Goal based Agent
* Utility based Agent
* Learning Agent.

L3: Problem Solving Agent:-

A kind goal based agent - choose their action in order to achieve Goals.

This allows the agent to a way of choosing among multiple possibilities, selecting the one which reaches a Goal state.

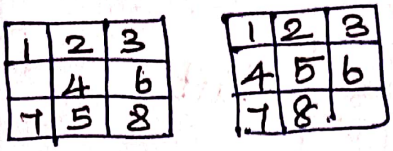
It requires Searching and planning techniques : eg: GPS - finding a path to certain destination.

Steps Performed by problem Solving agent.

Goal formulation
↓
Problem formulation.
↓
Search Solution.
↓
Execution.

Goal formulation:- It is the first and simplest step in problem solving. It organizes the steps and sequence required to formulate one goal out of multiple goals.

Problem



Problem Formulation:-

It is the most important step of problem solution which descides what action should be taken to achieve the formulated goal.

→ Initial State:- It is the starting state of the agent towards its goal.
(1,2,3/b,4,6/7,5,8).

→ Action:- It is the possible action available to the agent. (Left, Right, up, down).

→ Goal Test:- It determine if the given state is a goal state. (1,2,3/4,5,6/7,8,b)

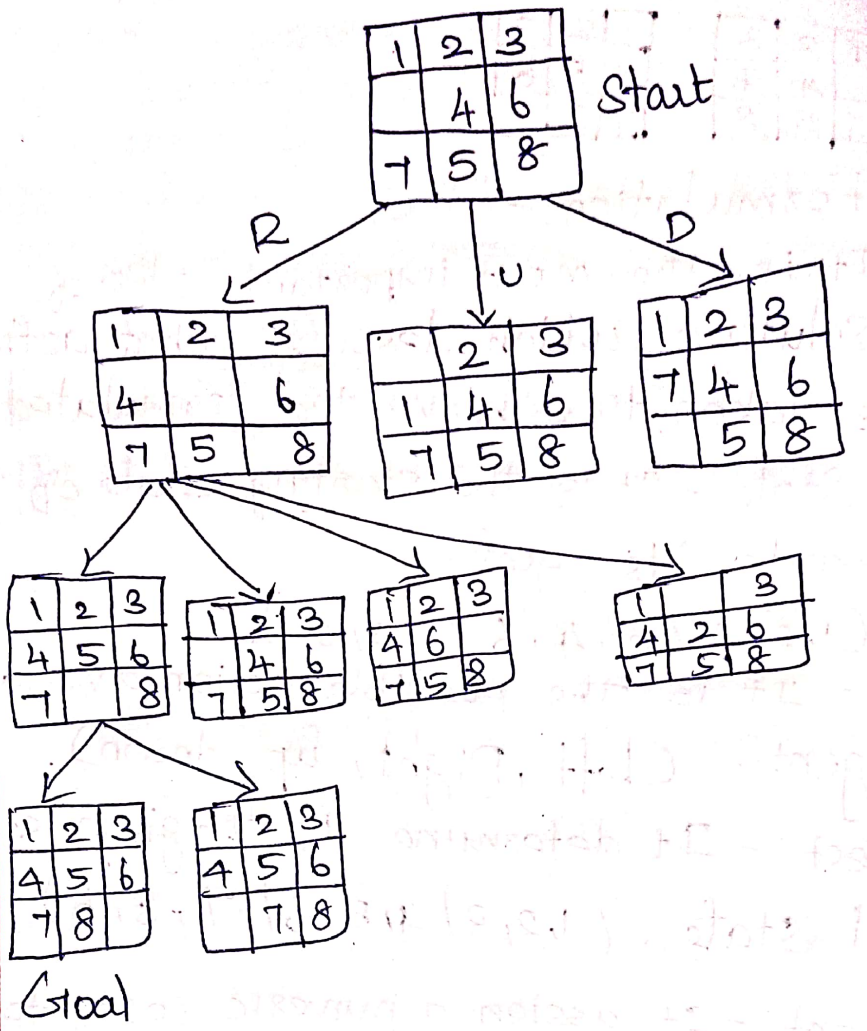
→ Path Cost:- It assign a numeric cost to each path that follows the goal.
(cost / per action).

Search Solution:-

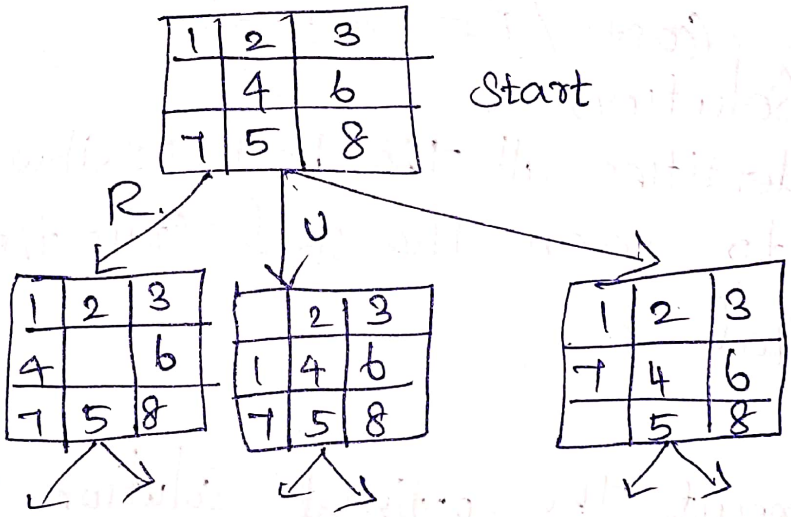
It identifies all the best posible sequence of action to reach the goal state from the current state:-

Execution:-

It execute the optimal solution from the searching algorithm to reach the goal state from the current state. An optimal solution has the lowest path cost among all the solutions.



Execution:



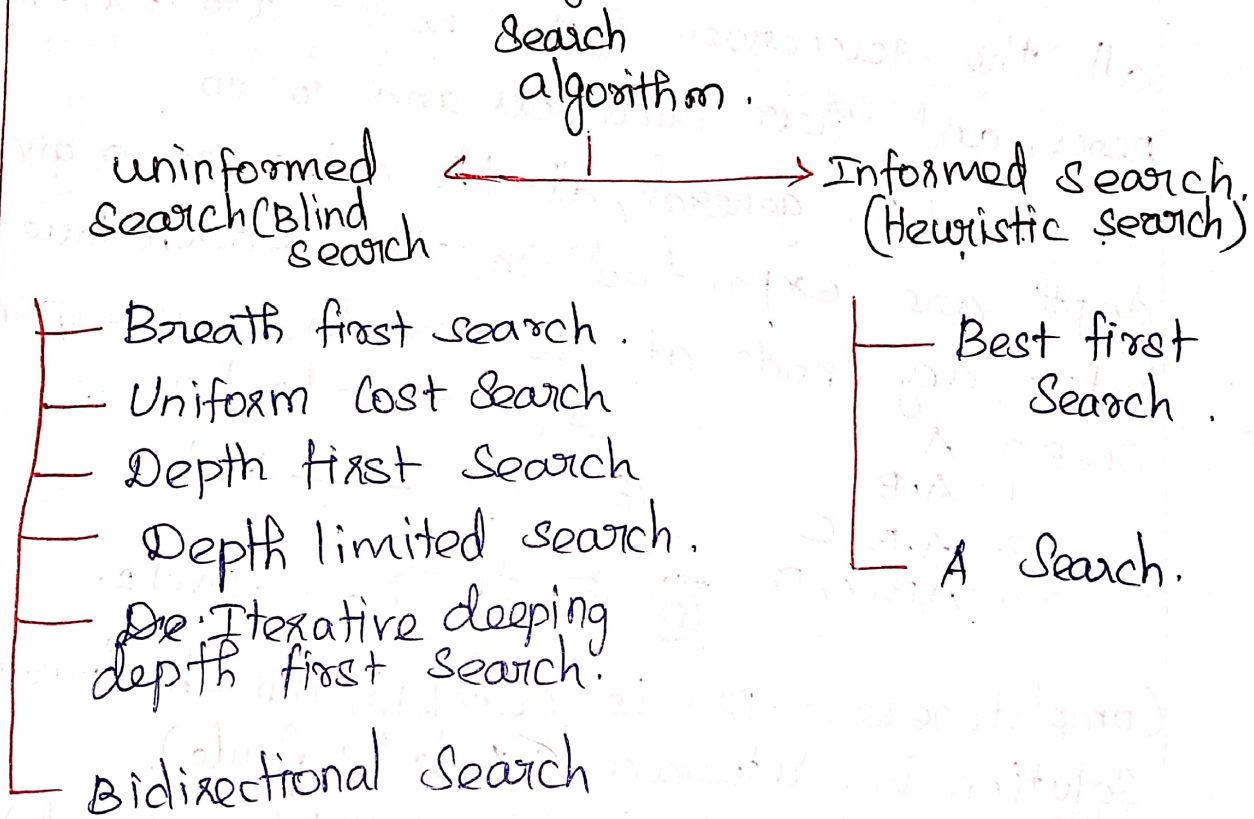
Search form the core component of many intelligent process.

Search is the systematic examination of states to find path from the start/root state to the goal state.

Many traditional search algorithm are used in AI application.

For complex problem, the traditional algorithm are unable to find the solution within some practical time and space limits.

Consequently, many spacial techniques are developed using heuristic function.



15. Uniformed Search strategies:-

→ uniformed search algorithm generates the search tree without using any domain knowledge. (get knowledge only from the knowledge definition).

→ This algorithm generates successors using the Successor fanction.

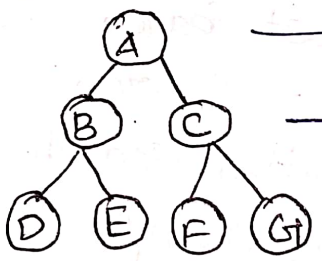
→ knowledge about Goal state and not Information about the path cost from the Current State to goal state.

i) Breadth First Search:-

→ The Root node is expanded first and then all the successors of the nodes are expended. next, and their successors and so on.

→ In general, all the nodes at a given depth are expanded in the search true before any node at the next level are expanded.

Step 0: A
1: A, B
2: A, B, C
3: A, B, C, D



Completeness:- BFS is Complete, The shallowest. Solution is returned (if b is finite).

Time Complexity:- $b + b_2 + b_3 + \cdots + b_d = O(b, d)$.

b - node at every State

d - depth of shallowes solution.

→ Time requirement is still a major factor.

based on memory size
→ space is the bigger problem (more than time),
optimal :- BFS is optimal if path cost is a non decreasing function of the depth of the node.

2) Uniform Cost Search :-

Instead of expanding the shallowest node, Uniform Cost Search expand the node n with the lowest path cost.

Uniform Cost doesn't not care about the number of steps a path has only about their total Cost.

Completeness :- UCS is complete, such if there is a solution UCS will find it.

Time Complexity :-

Let c^* is cost of the optimal solution, ε is a goal node. Then no of steps is $c^*/\varepsilon+1$

Here we have taken +1, as we start from State 0 & end to c^*/ε

worst case time complexity of UCS is $O(b^{1+}(c^*/\varepsilon))$

Space complexity :-

same logic for space complexity, so the worst case space Complexity of UCS is $O(b^{1+}(L^*/\varepsilon))$.

Optimal :-

UCS is always optimal as it only select a path with the lowest path Cost.

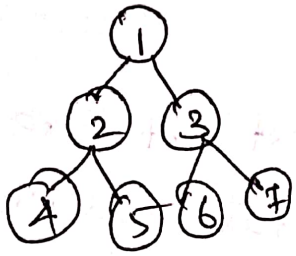
3) DEPTH FIRST SEARCH

Start with root node and complexity exp... left most child node, before exploring its sib...

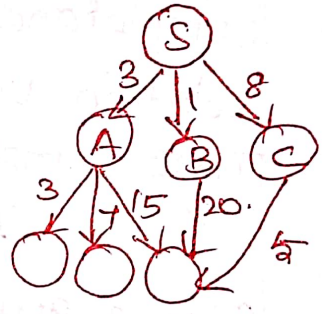
Siblings are explored in left to right fa...

Depth first search always expands the ~~request~~ deepest node in the current frontier of the search tree.

Depth 1st traversal. 1 → 2 → 4 → 5 → 3 → 6 → 7



Expanded node	Nodes list.
	So
So	A3 B1 C8
A3	D6 E10 G18 B1 C8
~~D6~~	E10 G18 B1 C8
~~D6~~	
E10	G18 B1 C8
G18	B1 C8



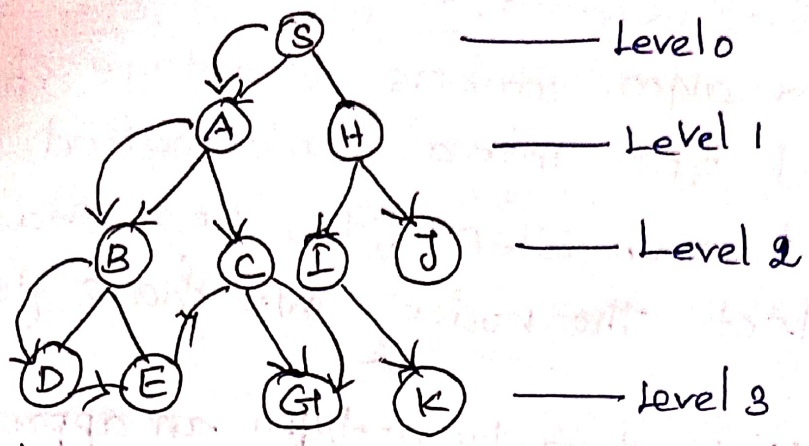
Solution path found is SAG1, Cost is

No of nodes expanded {including goal nodes} = 5

4) Depth limited Search Algorithm:-

Depth limited search is the combination of DFS and Limits for level.

Depth limited search is also similar to the DFS as it also implements "Last in first Out (LIFO) Stack data Structure" but in addition it has a level limit.



Completeness :-

DFS is complete within finite state space as expand every node within a limited search tree.

Time Complexity :- Time complexity of DFS will be equivalent to the node traversed by the algorithm $O(b-1)$

$$T(n) = 1 + n_2 + n_3 + \cdots + n_m = O(n_m)$$

m - maximum depth.

Space Complexity :-

DFS needs to store only single path from the root node, Hence space complexity of DFS is equivalent to the ~~single~~ size of the fringe set. which is $(O(b^* - 1))$

Optimal : DFS is a non optimal, it generate large no of steps or high cost to reach the goal node.

1-b Informed Search Algorithm :-

This informed search strategy is one that uses problem - specific knowledge beyond the definition of the problem itself. It can find solution more efficiently than an uninformed strategy.

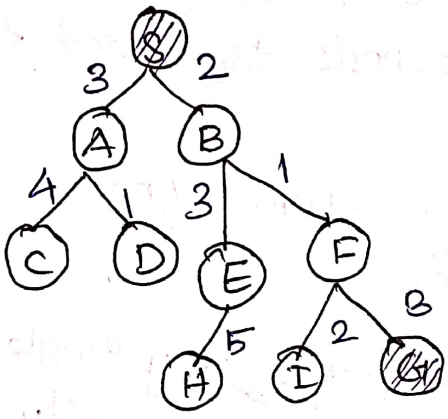
1) Best first Search:-

BFS algm combine advantage of DFS and BFS into a single method.

At each step of the BFS search we select the nodes we have generated so far.

This is done by applying an appropriate heuristic function to each of them.

Then expand the chosen node by using the rules to generate its successors.



Node	H(n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

2) A* Search:-

↳ It is most widely used form of BFS.

The evaluation function f(n) is obtained by.

g(n) - Cost to reach the node.

h(n) = Cost to get from node to goal

f(n) = g(n) + h(n).

Local Search and optimization Problem:-

Local Search Algorithm:-

The LSA Search only the final state not the path to get there.

Ex: 8 Queens problem (we can only about finding a valid final configuration of 8 queens (8 queen arranged on chess board and no queen can attack other 8 queens) and not the path from initial state to final state.

Local Search algorithm operate by searching from a start state to neighboring states

Without keeping track of a path, nor the set of states that have been reached.

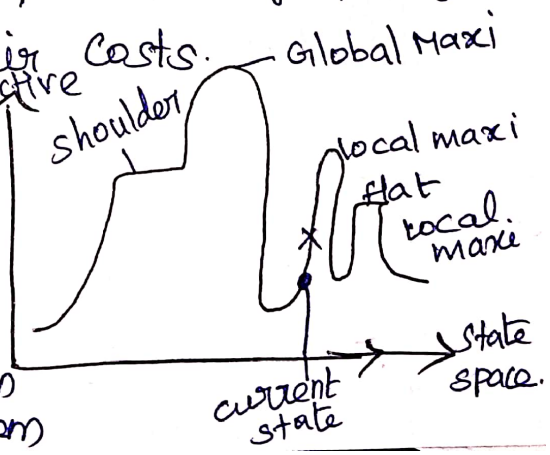
They might never explore a portion of the search space when a solution actually resides,

They search only the final state.

Ex: Hill Climbing Search algorithm:-

A state space landscape is a graph of states associated with their costs.

Hill climbling algorithm is a heuristic search algorithm which continuously move in the direction of increasing value to find the peak of the mountain or best solution to the problem



It keep track of one current state each iteration move to the neighbouring with highest value. that is if head direction that provides the steepest

App of Local Search Algorithm:-

- integrated Circuit Algorithm
- Factory floor layout
- Automatic Programming
- Telecommunication network optimization.
- Crop planning and Portfolio management.

Adv of LSA:-

→ use very little memory

→ They can often find reasonable Solution in large or infinite state space for which systematic algorithms are unsuitable.

→ Local Search algorithm can also Solve optimization problem

→ They find best state awarding to an objective function.

→ It is a game playing techniques where the agents are surrounded by a competitive environment → A conflicting goal is given to theagents (multiagent).

→ This agents complete with one another and try to defect one another in order to win the game.

→ Such conflicting goal giverise to the adversarial search.

→ Here, games playing means discussing those games when human intelligence and logic factor is used, excluding other factors such as luck factor.

→ Tic-tac-toe, chess, cheekers etc.. such type of games where no luck factor works, only mind works.

Game theory:-

→ According to game theory, a game played b/w two players, To complete the game one has to win the game and other to looser automatically.

Consider a game with 2 player Max and Min. Max move 1st followed by Min.

A game can be formally defined as a search problem with the following elements.

→ So the initial state.

→ player(s): Define which player has to move in a state(s).

→ Action(s) - Return the set of legal move in a state.

→ Result(s,a) - The transition model which define the result of a move.

→ Terminal - Test(s) - True when game is over, false otherwise.

→ utility (s,p) - This function also objective function or playoff function.

→ Defines the final numeric value for a game that ends in a terminal state s for player p.

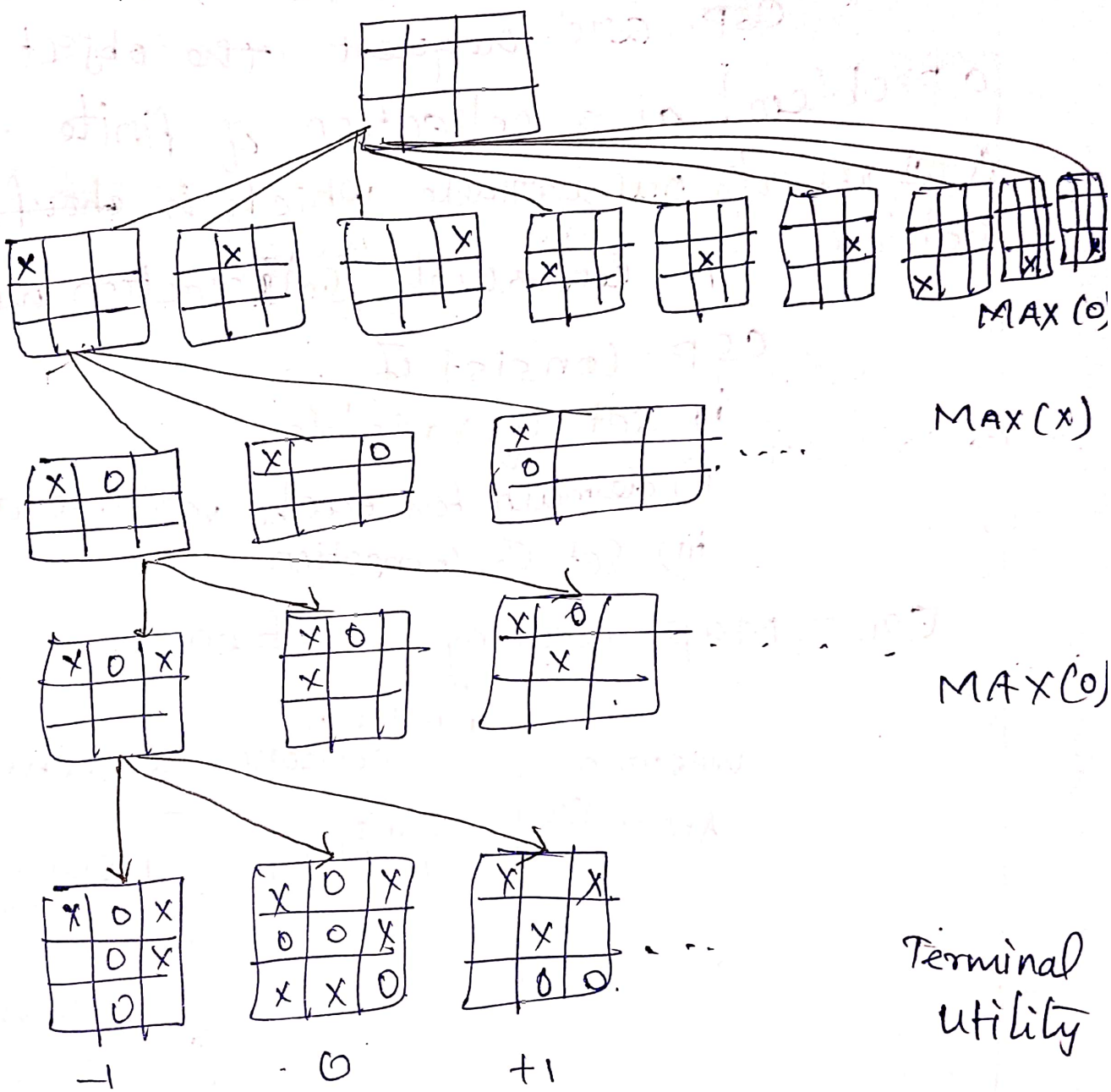
eg:- In tic-tac-toe, the outcome is win, loss or draw with values (-1, 0, or 1)

Max has 9 possible moves from the initial state.

(10)

play alternates b/w Max placing an X and Min placing an O.

The number in each leaf node indicates the utility value of terminal state from the point of view of MAX (Good for MAX, bad for Min).



Lg) CSP - Constraint Satisfaction

CSPs are mathematical question defined as a set of objects whose state must satisfy a number of constraints or limitation.

CSPs are request the object in a problem, as a collection of finite constraints out variable which is ~~should~~ solved by constraints satisfaction methods.

CSP Consist of
i) set of variable
ii) domains for each variable and
iii) set of connection

Ex:- Map coloring problem.