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,

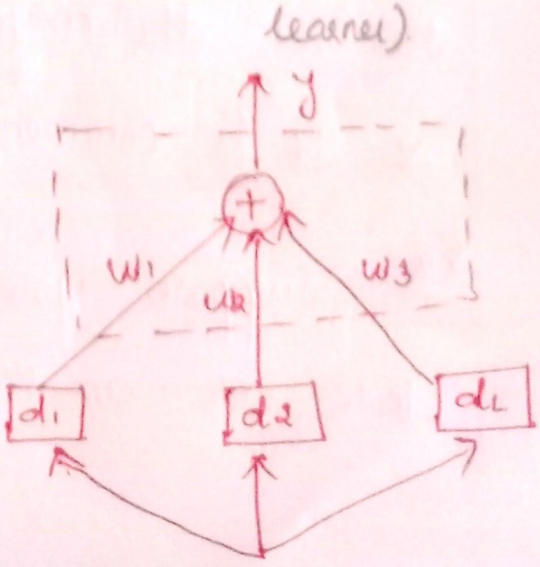
→ when k output exist for each learner,

→ 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.

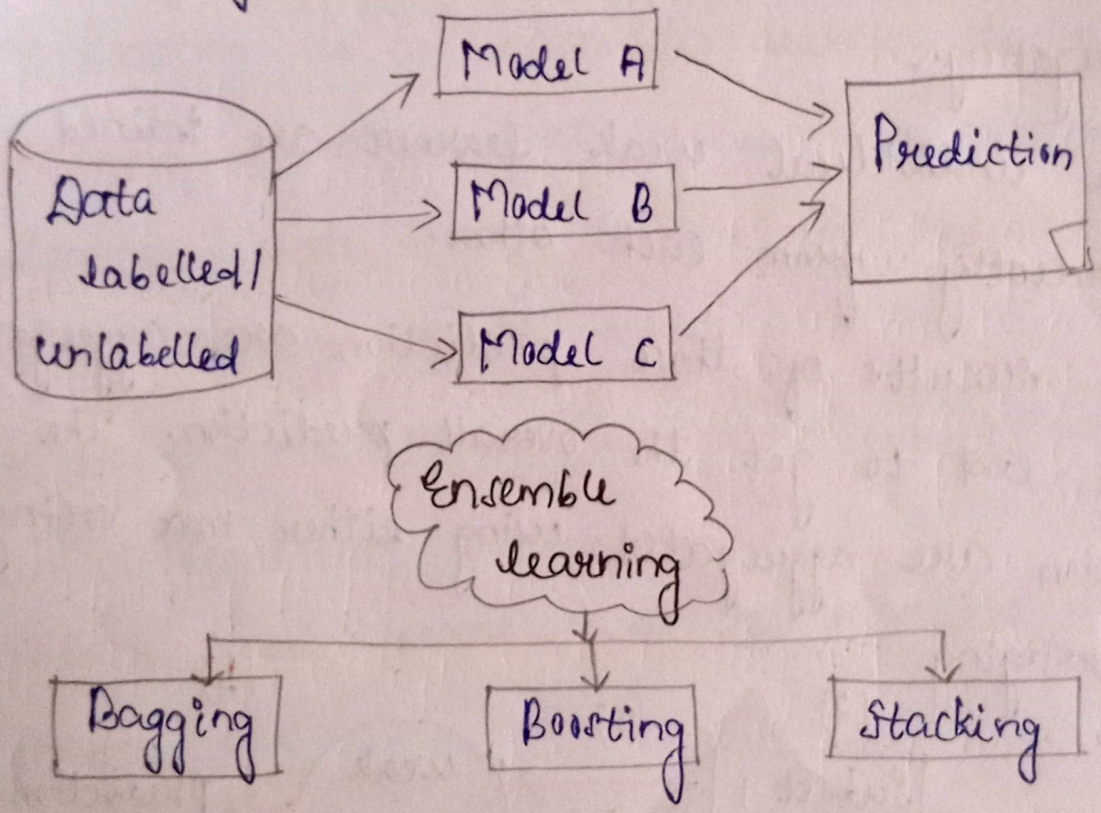
When to use Voting:

* 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.

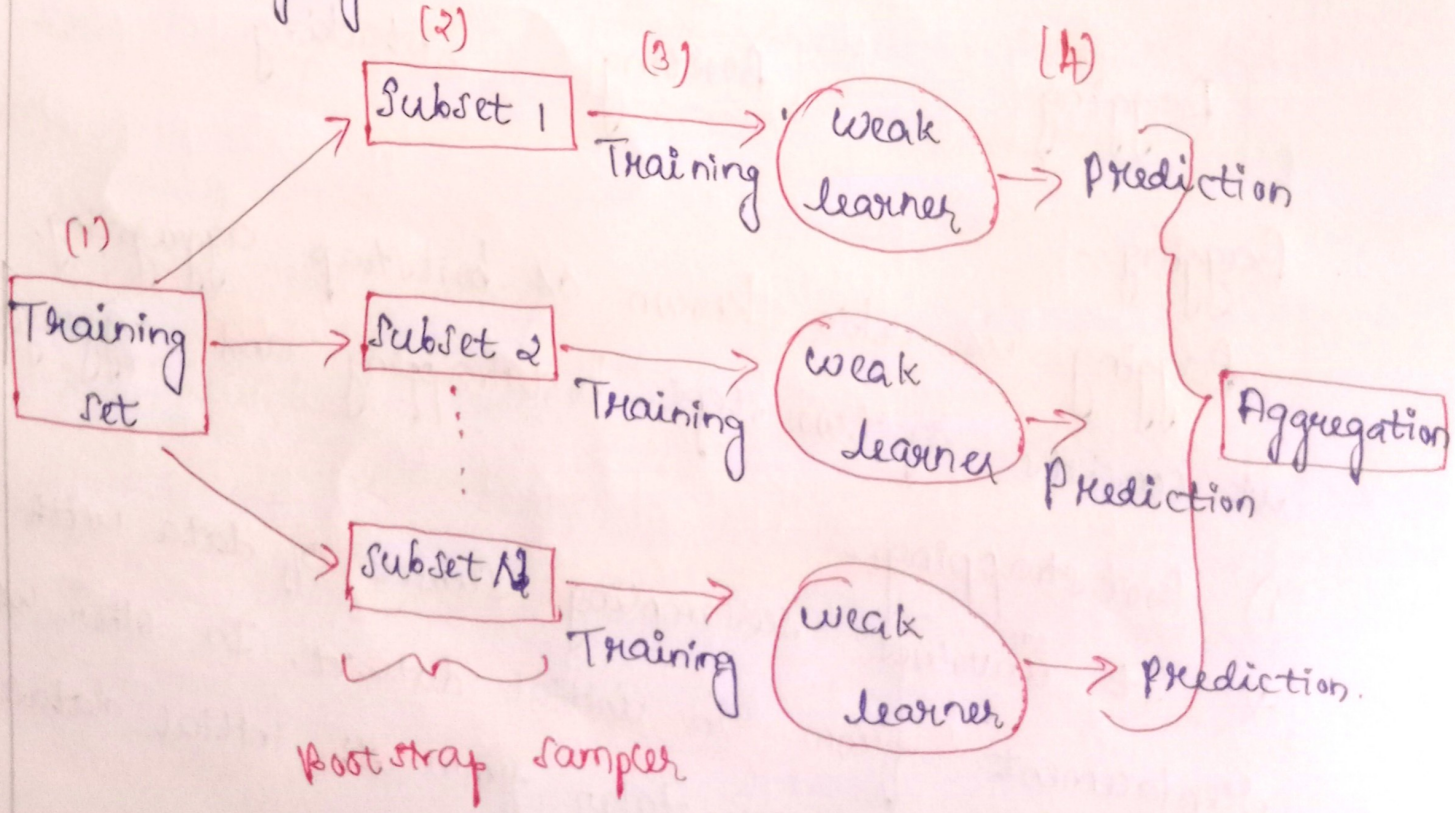
* These subsets of data are called bootstrapped dataset or simply, bootstraps.

* 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:-

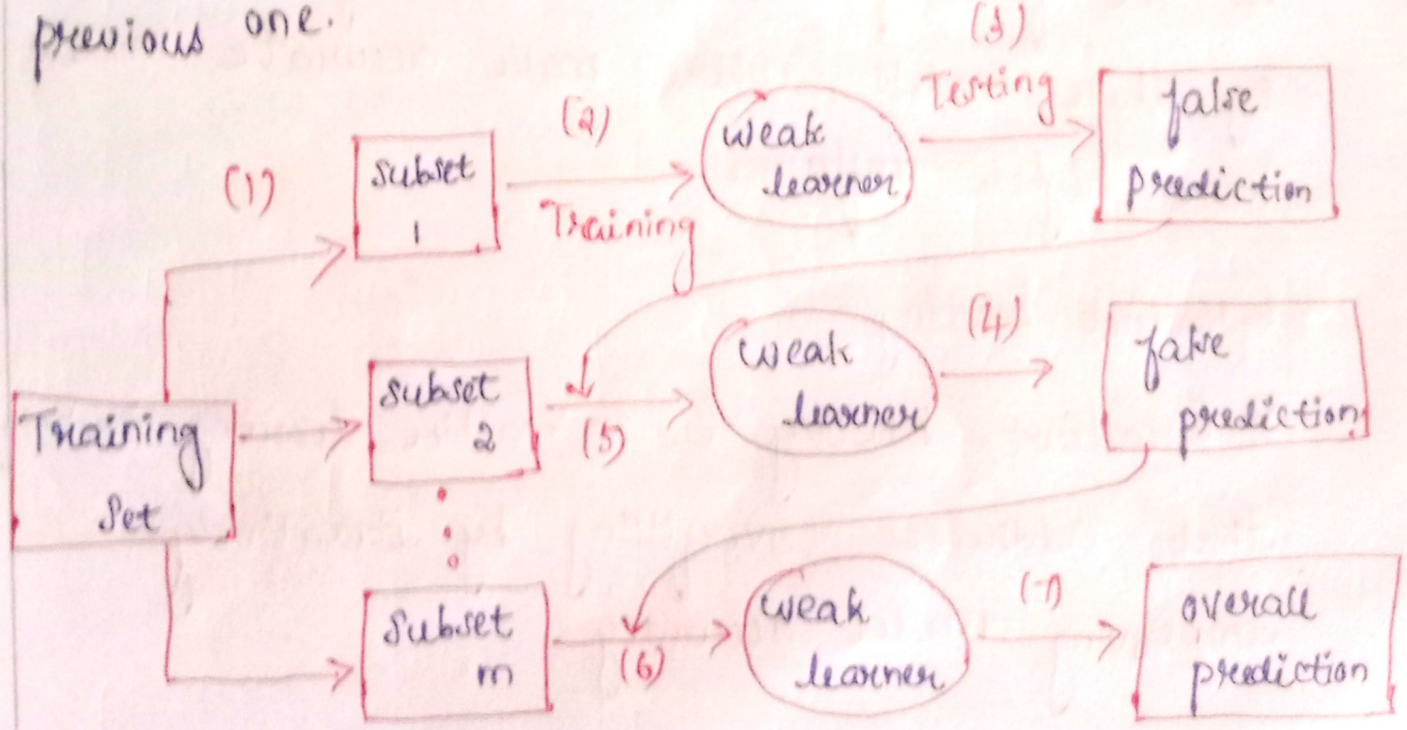
* Loss of interpretability
* Computationally expensive
* Less flexible.

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.

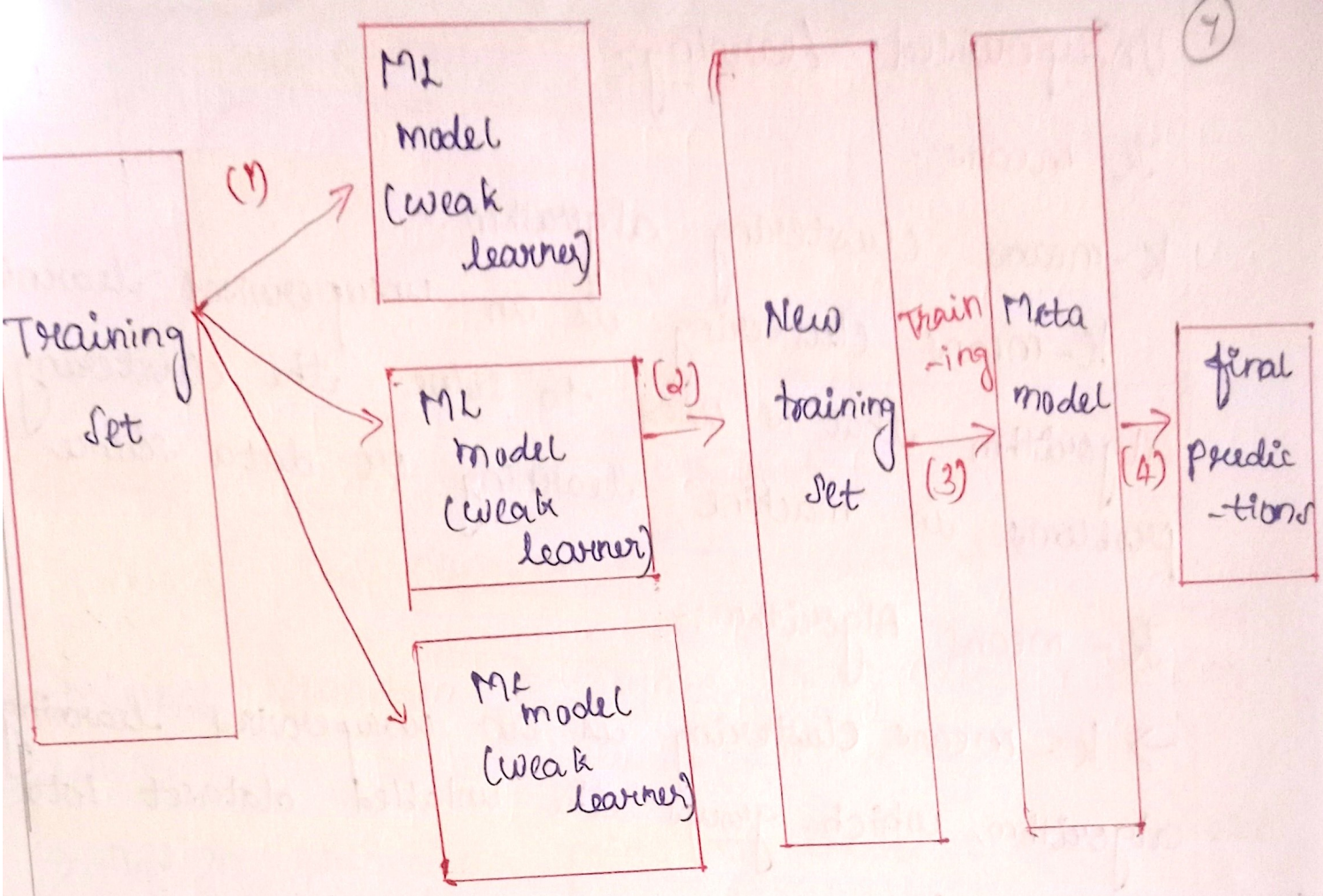
Stacking:-

* 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.

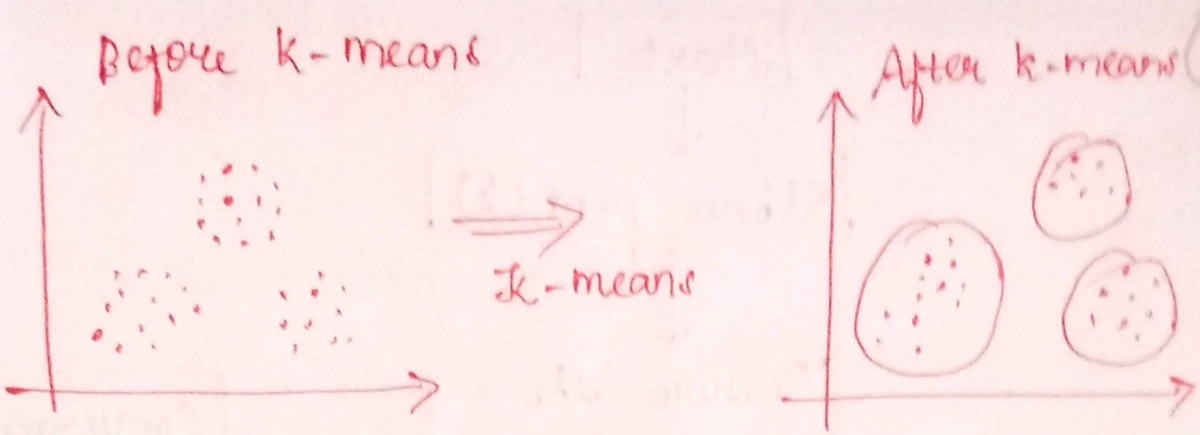
→ Here k defines the no of pre-defined clusters that need to be created in the process.

→ 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.

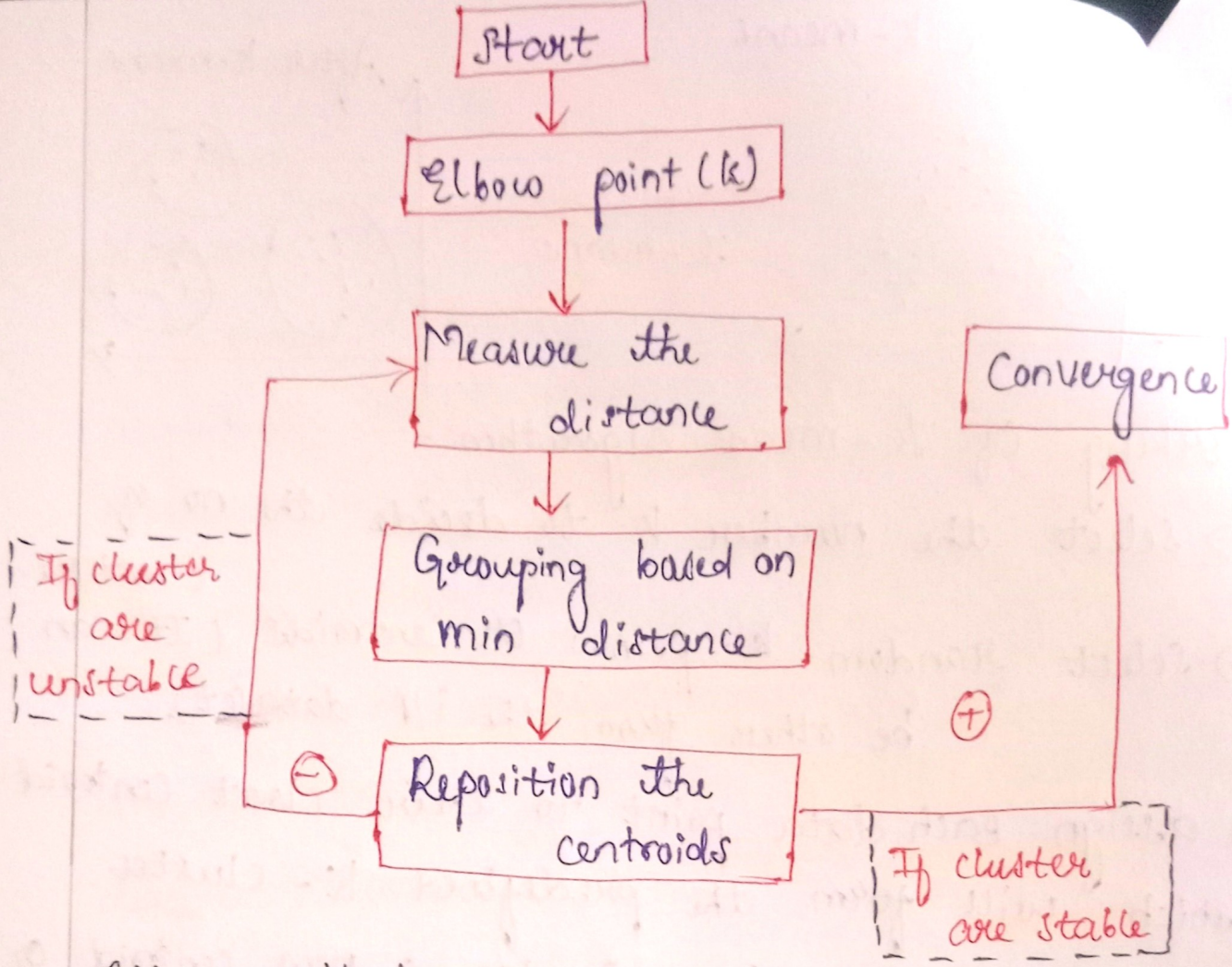
→ Calculate the variance & place a new centroid of each cluster.

→ 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.

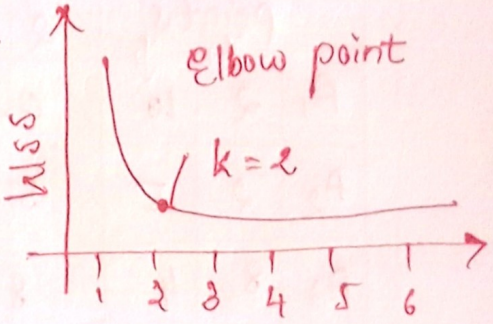
$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

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:-

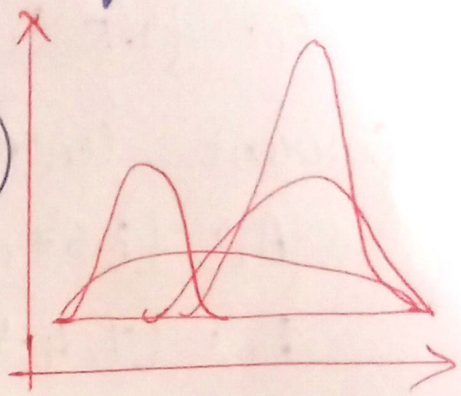
$P_1(1,2,3)$ $P_2(0,1,2)$ $P_3(3,0,5)$ $P_4(4,1,3)$

$P_5(5,0,1)$ 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:-

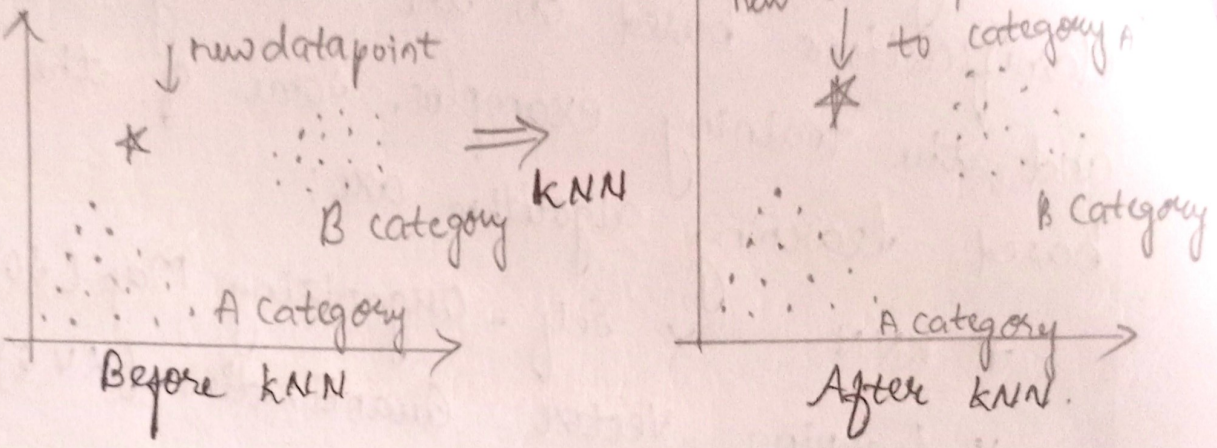
→ K-Nearest Neighbours (KNN) algorithm is a type of unsupervised ML algorithm which can be used for both classification as well as regression problems. It is mainly used for classification problems in industry.

→ 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.

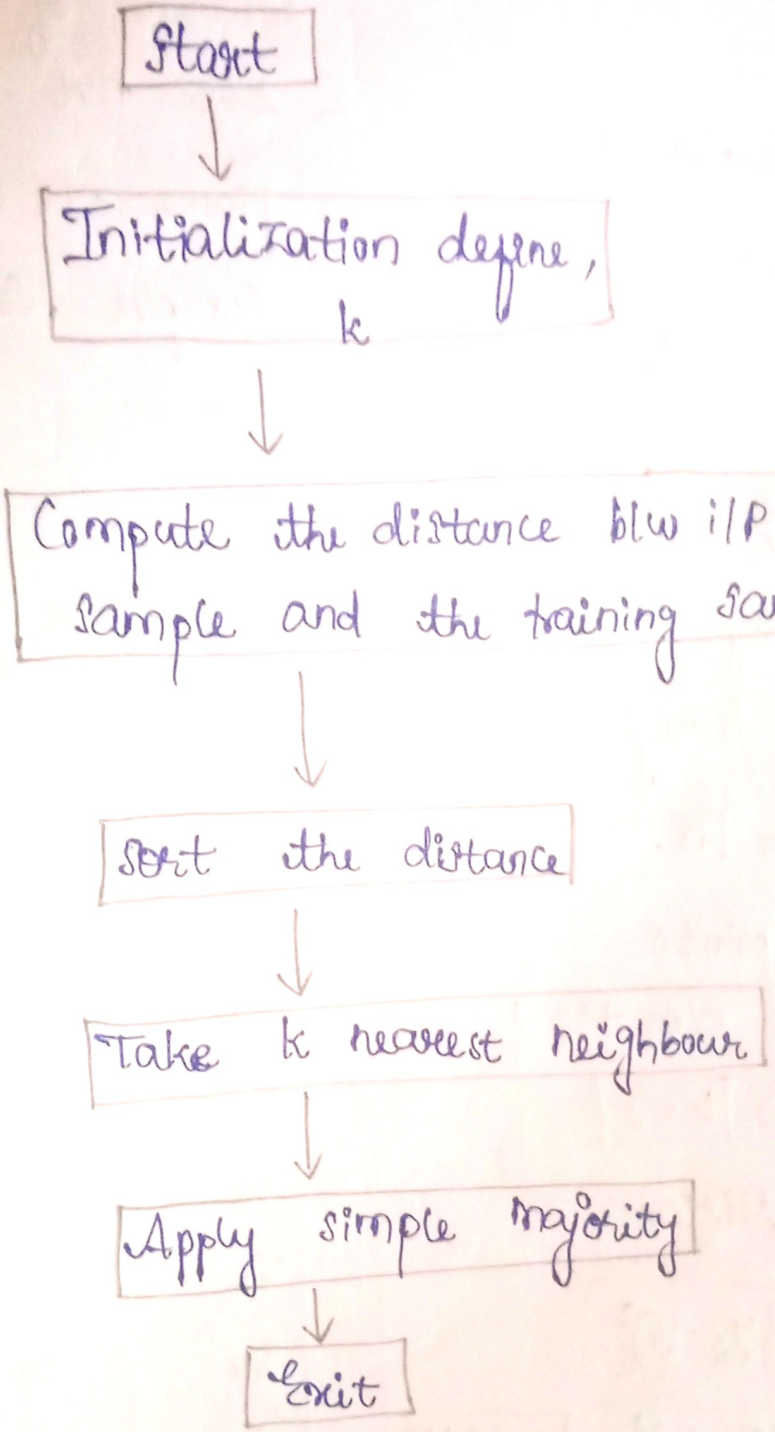
→ Calculate the Euclidean distance of K no of neighbour.

→ 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) :-

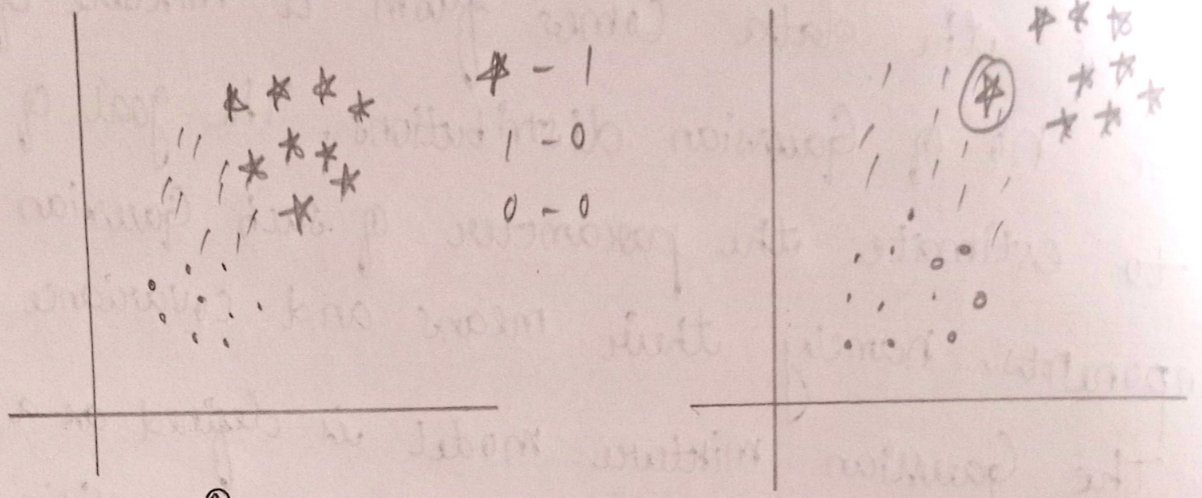
This defines the shape & orientation of each gaussian distribution, allowing for ellipsoidal clusters.

(3) Mixing Coefficient :-

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

belonging to each other.

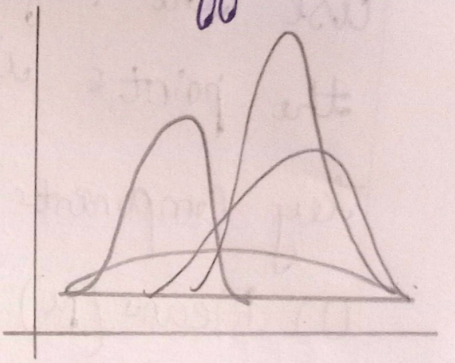
* The sum of all mixing coefficient is 1.



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 :-

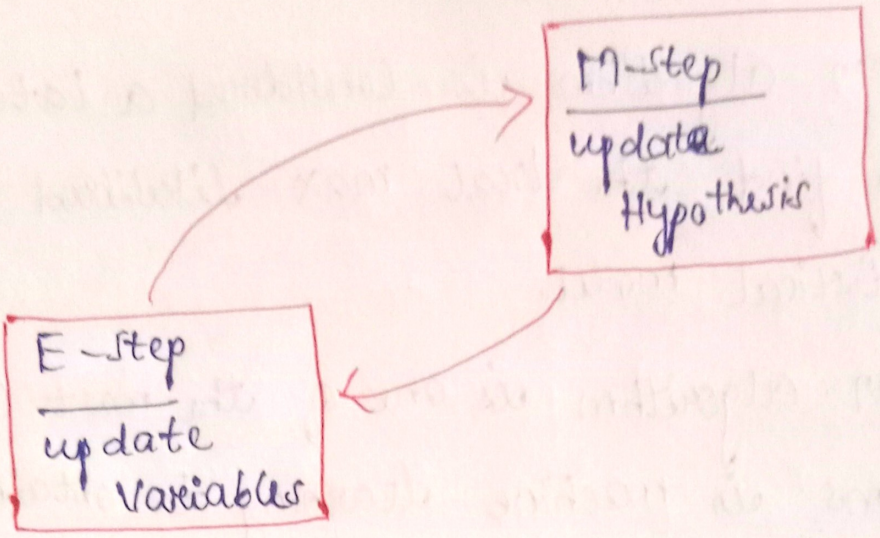
→ The EM algorithm is considered a latent variable model to find the local max likelihood parameter of a statistical model.

→ 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).