

### Unit - III

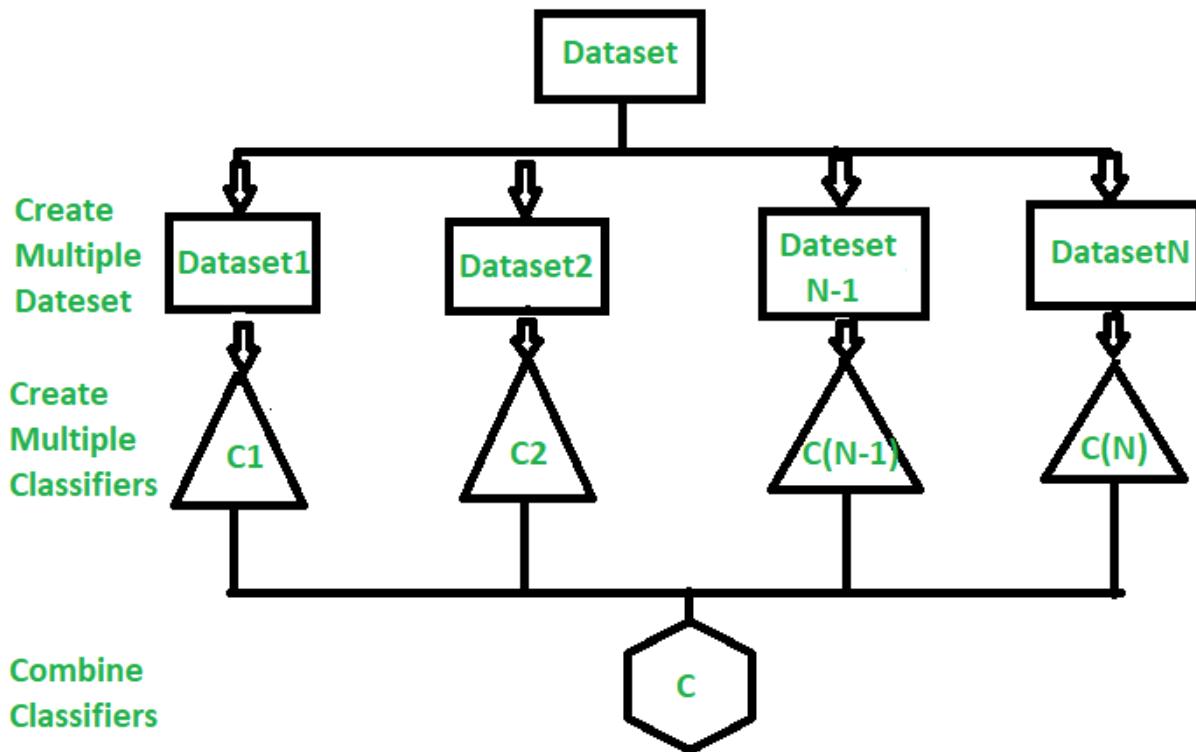
**Ensemble Algorithms: Bagging, Boosting, Random Forest Algorithm.**

**Unsupervised Learning:**

**Cluster Analysis: Similarity measures, categories of clustering algorithms, k-means, Hierarchical, Expectation-Maximization, Fuzzy c-means algorithm.**

## Ensemble Learning

Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. Basic idea is to learn a set of classifiers (experts) and to allow them to vote. Then take majority/average of these votes for final results.

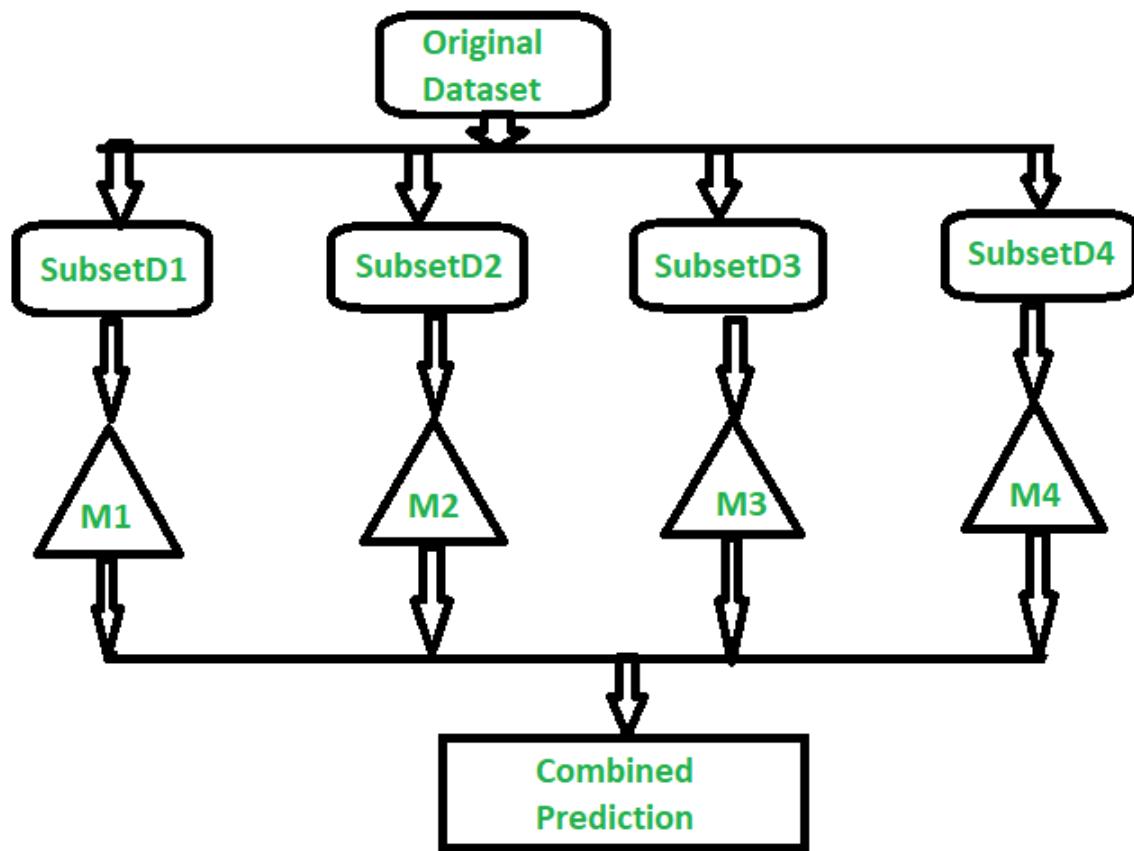


### Bagging:

Bagging (Bootstrap Aggregation) is used to reduce the variance of a decision tree. Suppose a set  $D$  of  $d$  tuples, at each iteration  $i$ , a training set  $D_i$  of  $d$  tuples is sampled with replacement from  $D$  (i.e., bootstrap). Then a classifier model  $M_i$  is learned for each training set  $D < i$ . Each classifier  $M_i$  returns its class prediction. The bagged classifier  $M^*$  counts the votes and assigns the class with the most votes to  $X$  (unknown sample).

### Implementation steps of Bagging –

1. Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
2. A base model is created on each of these subsets.
3. Each model is learned in parallel from each training set and independent of each other.
4. The final predictions are determined by combining the predictions from all the models.



## Boosting

Boosting is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers. It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

## Advantages of Boosting

Improved Accuracy – Boosting can improve the accuracy of the model by combining several weak models' accuracies and averaging them for regression or voting over them for classification to increase the accuracy of the final model.

Robustness to Overfitting – Boosting can reduce the risk of overfitting by reweighting the inputs that are classified wrongly.

## Implementation of Boosting

1. Initialise the dataset and assign equal weight to each of the data point.
2. Provide this as input to the model and identify the wrongly classified data points.
3. Increase the weight of the wrongly classified data points.
4. if (got required results)
  - Goto step 5
  - else
    - Goto step 2
5. End

## AdaBoost

The AdaBoost stands for Adaptive boosting. This technique gives weight to each data point. At each iteration a new classifier is trained on the training set, with weights that are applied to the training set for each data point is modified. The weights are set to  $1/N$ , where  $N$  is the number of data points in the training set. Then at each iteration, the error ( $\epsilon$ ) is computed as the sum of the weights of the misclassified points, and the weights for incorrect examples are updated by being multiplied by  $\alpha = (1 - \epsilon)/\epsilon$ . Weights for the correct examples are left alone, and the whole set is normalized so that it sums to 1. Training terminates after a set number of iterations, or when either all of the datapoints are classified correctly.

**The algorithm is given below:**

1. Initialize the weights of each input to  $1/N$
2. Training the algorithm

Repeat

    . Train the classifier using training data.

    Verify the results for training data

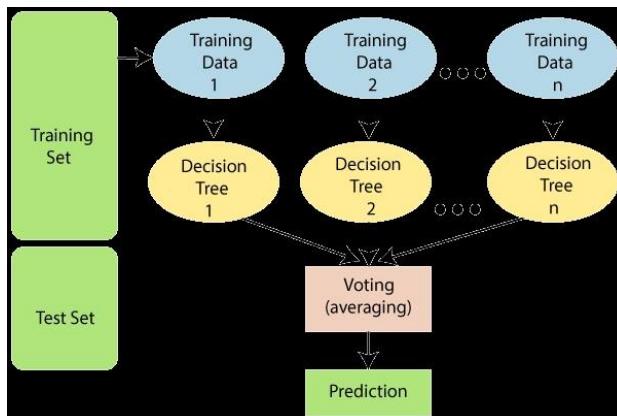
    Compute error ( $\epsilon$ ) for misclassified data

    Update the weights by multiplying  $\alpha = (1 - \epsilon)/\epsilon$

Until number of iterations or whether all data points are classified correctly.

## Random Forest Algorithm

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
- It can be used for both Classification and Regression problems in ML.
- It is based on the concept of **ensemble learning**, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model*.
- *Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average or majority to improve the predictive accuracy of that dataset.*
- Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output
- The working of the Random Forest algorithm:



## Random Forest Algorithm

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision tree associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

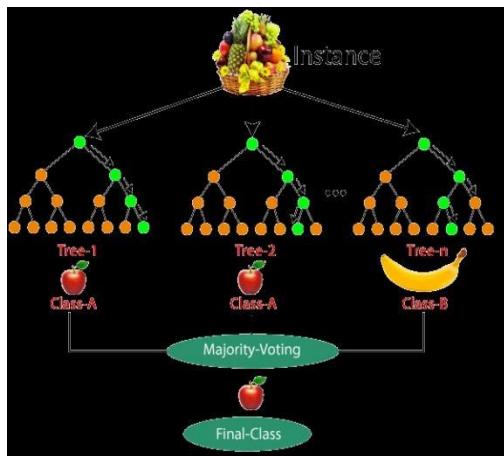
**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

**Example:** Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree.

Each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



## Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

**Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.

**Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.

**Land Use:** We can identify the areas of similar land use by this algorithm.

**Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

## Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

# Clustering

The process of grouping data objects based on their similarity is clustering. The data set is split into number of clusters.

Cluster is collection of data objects with similar features.

The data objects within a cluster are similar and objects in different clusters are dissimilar.

Cluster applications – pattern recognition, image processing and market research.

## categories of clustering algorithms

Clustering methods can be classified into the following categories –

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method

## Partitioning Method

Suppose we are given a database of ‘n’ objects and the partitioning method constructs ‘k’ partitions of data.

Each partition will represent a cluster and  $k \leq n$ . It means that it will classify the data into  $k$  groups, which satisfy the following requirements –

Each group contains at least one object.

Each object must belong to exactly one group.

## Hierarchical Methods

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed.

There are two approaches here –

Agglomerative Approach and Divisive Approach

## **Agglomerative Approach**

- This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another.
- It keep on doing so until all of the groups are merged into one or until the termination condition holds.

## **Divisive Approach**

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds.

- ***Density-based Method***

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

### **Grid-based Method**

In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure.

## Partitioning Methods

Database has  $n$  objects

$k$  partitions to be made where  $k \leq n$ ; each partition is a cluster.

Objects within a cluster are similar; objects of different clusters are dissimilar.

### **Partitioning Methods:**

(1) k-means

(2) k-medoids:

## **(A) Centroid-based technique: The k-means method:**

- Cluster similarity is measured using mean value of objects in the cluster.
- Randomly select  $k$  objects. Each object is a cluster mean or center.
- Each of the remaining objects is assigned to the most similar cluster based on the distance between the object and the cluster mean.
- Compute new mean for each cluster.
- This process iterates until all the objects are assigned to a cluster and the partitioning criterion is met.
- This algorithm determines  $k$  partitions that minimize the squared error function

## **Algorithm: K mean:**

### **Input:**

K: The number of clusters

D: A dataset containing N number of objects

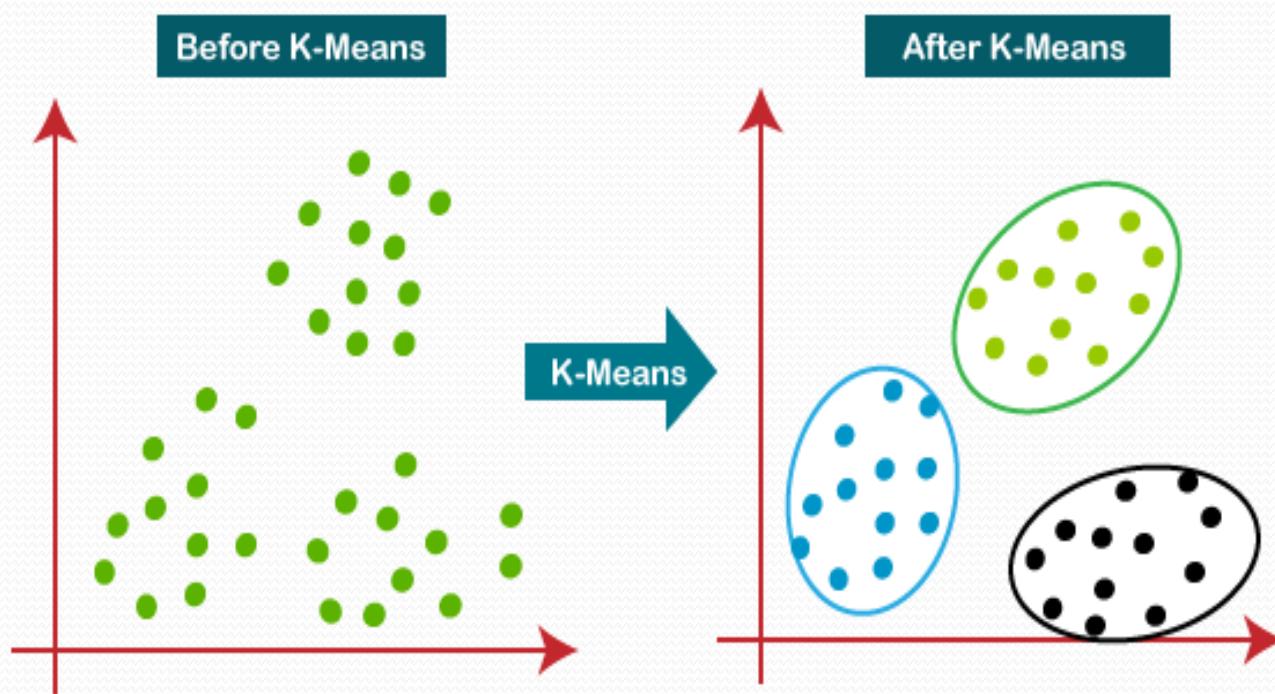
**Output:** A dataset of K clusters

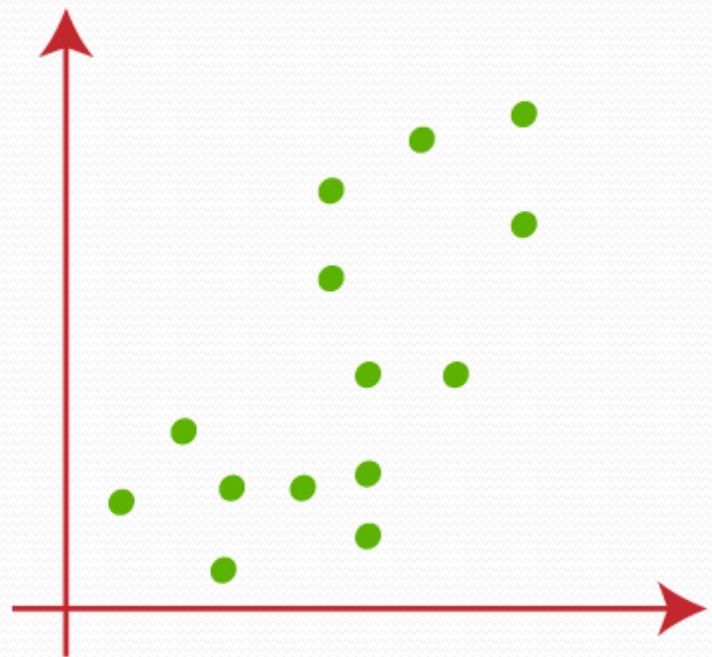
### **Method:**

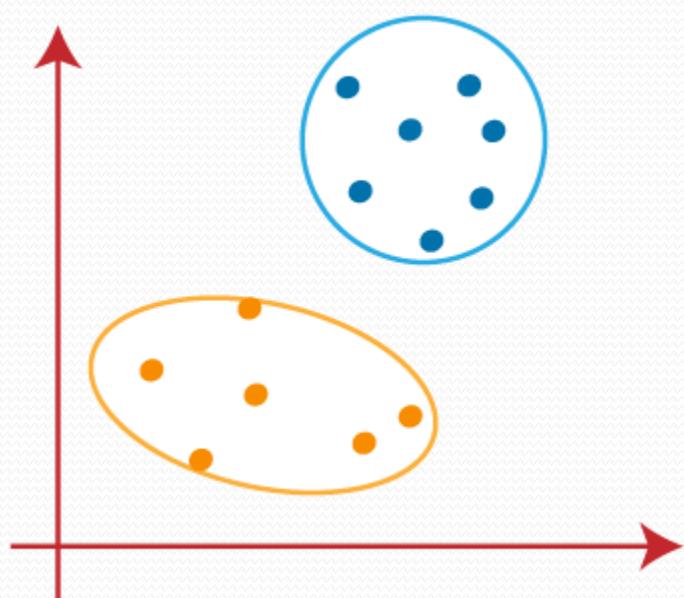
- '(1) Randomly assign K objects from the dataset(D) as cluster centres(C)
- (2) **repeat**
- (3)   (Re) Assign each object to cluster which the object is most similar based upon mean values.
- (4)   Update Cluster means, i.e., Recalculate the mean of each cluster with the updated values.
- (5) **until** no change;

# K-Means clustering

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters







## Elbow Method

The Elbow method is used to find the optimal number of clusters.

This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster.

The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$\text{WCSS} = \sum_{P_i \text{ in Cluster}_1} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster}_2} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster}_3} \text{distance}(P_i, C_3)^2$$

In the above formula of WCSS,

$\sum_{P_i \text{ in Cluster}_1} \text{distance}(P_i, C_1)^2$ : It is the sum of the square of the distances between each data point and its centroid within a cluster1.

To measure the distance between data points and centroid, we can use method such as Euclidean distance.

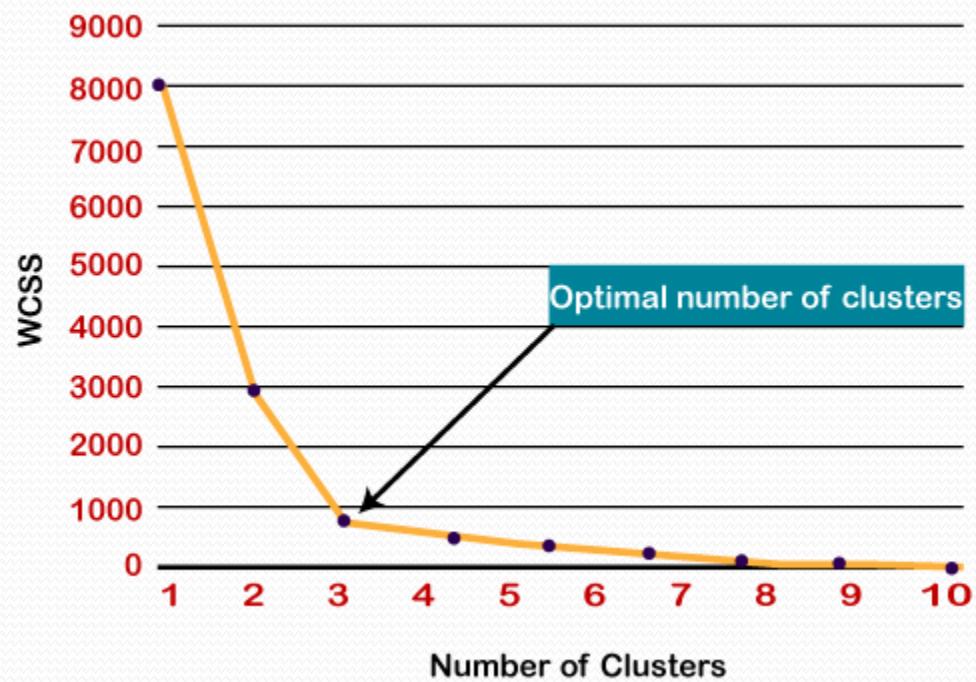
To find the optimal value of clusters, the elbow method follows the below steps:

It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).

For each value of K, calculates the WCSS value.

Plots a curve between calculated WCSS values and the number of clusters K.

The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.



## Example problem for K-means clustering

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:  $A_1=(2,10)$ ,  $A_2=(2,5)$ ,  $A_3=(8,4)$ ,  $A_4=(5,8)$ ,  $A_5=(7,5)$ ,  $A_6=(6,4)$ ,  $A_7=(1,2)$ ,  $A_8=(4,9)$ .

Suppose that the initial centers of each cluster are  $A_1$ ,  $A_4$  and  $A_7$ .

Find the clusters and their members.

K Means

$$A_1 = (2, 10), A_2 = (2, 5), A_3 = (8, 4), A_4 = (5, 8) \quad ①$$

$$A_5 = (7, 5), A_6 = (6, 4), A_7 = (1, 2), A_8 = (4, 9)$$

$$\text{centroid } (c_1) = A_1 = (2, 10)$$

$$\text{centroid } (c_2) = A_4 = (5, 8)$$

$$\text{centroid } (c_3) = A_7 = (1, 2)$$

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

Iteration 1 :

Distance from every point to centroids

¶

A1:

$$d(A_1, c_1) = \sqrt{(2-2)^2 + (10-10)^2} = 0 \quad \checkmark$$

$$d(A_1, c_2) = \sqrt{(2-5)^2 + (10-8)^2} = \sqrt{13}$$

$$d(A_1, c_3) = \sqrt{(2-1)^2 + (10-2)^2} = \sqrt{65}$$

~~A1~~  $\in$  cluster 1

A2:

$$d(A_2, c_1) = \sqrt{25} = 5$$

$$d(A_2, c_2) = \sqrt{18} = 4.24$$

$$d(A_2, c_3) = \sqrt{10} = 3.16$$

A2  $\in$  cluster 3

$$A_3: d(A_3, c_1) = \sqrt{36} = 6$$

$$d(A_3, c_2) = \sqrt{25} = 5 \quad \checkmark$$

$$d(A_3, c_3) = \sqrt{53} = 7.28$$

A3  $\in$  cluster 2

$$A_4: d(A_4, c_1) = \sqrt{13}$$

$$d(A_4, c_2) = 0 \quad \checkmark$$

$$d(A_4, c_3) = \sqrt{52}$$

A4  $\in$  cluster 2

A5:

$$d(A_5, c_1) = \sqrt{50} = 7.07$$

$$d(A_5, c_2) = \sqrt{13} = 3.60$$

$$d(A_5, c_3) = \sqrt{45} = 6.70$$

A5  $\in$  cluster 2

A6:

$$d(A_6, c_1) = \sqrt{52} = 7.21$$

$$d(A_6, c_2) = \sqrt{17} = 4.12$$

$$d(A_6, c_3) = \sqrt{29} = 5.38$$

A6  $\in$  cluster 2

A7:

$$d(A_7, c_1) = \sqrt{65} \quad \checkmark$$

$$d(A_7, c_2) = \sqrt{52}$$

$$d(A_7, c_3) = 0 \quad \checkmark$$

A7  $\in$  cluster 2

A8:

$$d(A_8, c_1) = \sqrt{55}$$

$$d(A_8, c_2) = \sqrt{2} \quad \checkmark$$

$$d(A_8, c_3) = \sqrt{58}$$

A8  $\in$  cluster 2

At end of iteration 1:

$$\text{cluster 1: } \{A_1\} = \{(2, 10)\}$$

$$\text{cluster 2} = \{A_3, A_4, A_5, A_6, A_8\} = \{(8, 4), (5, 8), (7, 5), (6, 4), (4, 9)\}$$

$$\text{cluster 3} = \{A_2, A_7\} = \{(2, 5), (1, 2)\}$$

Iteration 2:

centres of the new clusters:

$$c_1 = (2, 10)$$

$$c_2 = ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6)$$

$$c_3 = ((2+1)/2, (5+2)/2) = (1.5, 3.5)$$

At end of second iteration

$$\text{cluster 1: } \{A_1, A_8\}$$

$$\text{cluster 2: } \{A_3, A_4, A_5, A_6\}$$

$$\text{cluster 3: } \{A_2, A_7\}$$

Manhattan

Iteration 3

centres of new cluster

$$c_1 = (3, 9.5)$$

$$c_2 = (6.5, 5.25)$$

$$c_3 = (1.5, 3.5)$$

At end of iteration

$$\text{cluster 1: } \{A_1, A_4, A_8\}$$

$$\text{cluster 2: } \{A_3, A_5, A_6\}$$

$$\text{cluster 3: } \{A_2, A_7\}$$

$$\text{with centers } c_1 = (3.66, 9)$$

$$c_2 = (7.43, 4.33)$$

$$c_3 = (1.5, 3.5)$$

## Hierarchical clustering Methods

A hierarchical clustering method can be used for grouping data objects into a hierarchy or “tree” of clusters.

Ex:employees in organization can grouped into executives, managers, and staff.

Hierarchical clustering method can be either **agglomerative** or **divisive**, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or topdown (splitting) fashion.

An **agglomerative** hierarchical clustering method uses a bottom-up strategy.

It starts by considering each object form its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied. The single cluster becomes the hierarchy's root.

For the merging step, it finds the two clusters that are closest to each other (according to some similarity measure), and combines the two to form one cluster.

A **divisive** hierarchical clustering use a **top-down strategy**. It starts by placing all objects in one cluster, which is the hierarchy's root.

It then divides the root cluster into several smaller sub-clusters, and recursively partitions those clusters into smaller ones.

The partitioning process continues until each cluster at the lowest level is coherent enough—either containing only one object, or the objects within a cluster are sufficiently similar to each other.

---

## **Agglomerative versus divisive hierarchical clustering.**

AGNES (AGglomerative NESting), an agglomerative hierarchical clustering method, and DIANA (DIvisive ANAlysis), a divisive hierarchical clustering method, on a data set of five objects,  $\{a, b, c, d, e\}$ .

Initially, AGNES, the agglomerative method, places each object into a cluster of its own. The clusters are then merged step-by-step according to some criterion.

- For example, clusters  $C_1$  and  $C_2$  may be merged if an object in  $C_1$  and an object in  $C_2$  form the minimum Euclidean distance between any two objects from different clusters.

This is a single-linkage approach in that each cluster is represented by all the objects in the cluster, and the similarity between two clusters is measured by the similarity of the *closest pair of data points belonging to different clusters*.

The cluster-merging process repeats until all the objects are eventually merged to form one cluster.

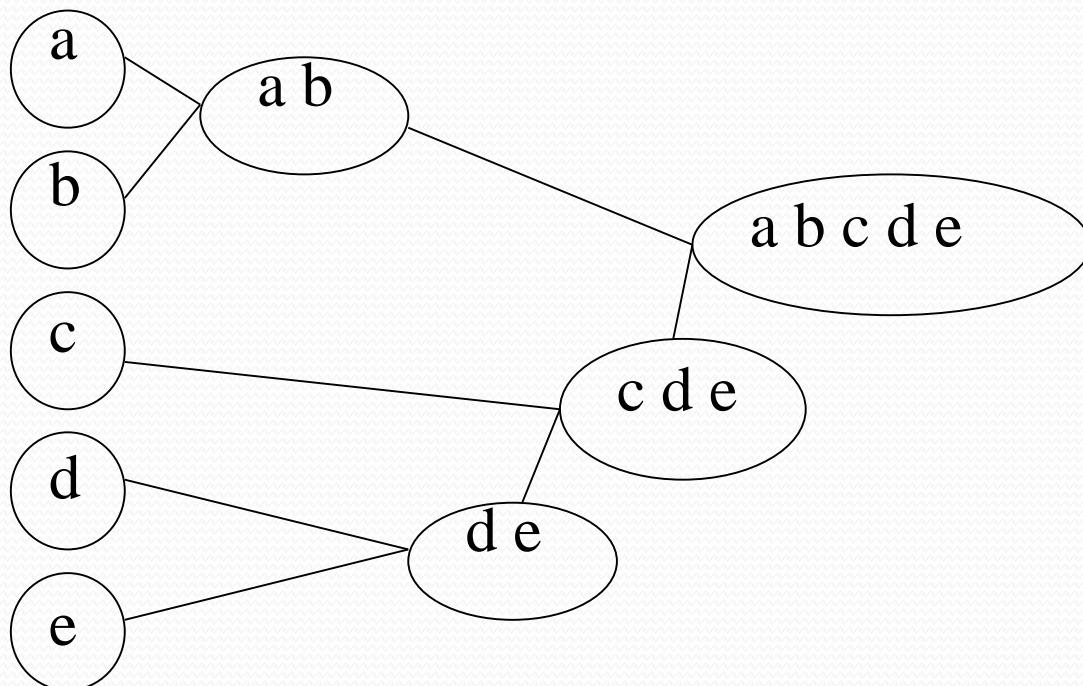
## DIANA

All the objects are used to form one initial cluster. The cluster is split according to some principle such as the maximum Euclidean distance between the closest neighboring objects in the cluster.

The cluster-splitting process repeats until, eventually, each new cluster contains only a single object.

Step 0      Step 1      Step 2      Step 3      Step 4

**agglomerative  
(AGNES)**



Step 4      Step 3      Step 2      Step 1      Step 0

**divisive  
(DIANA)**



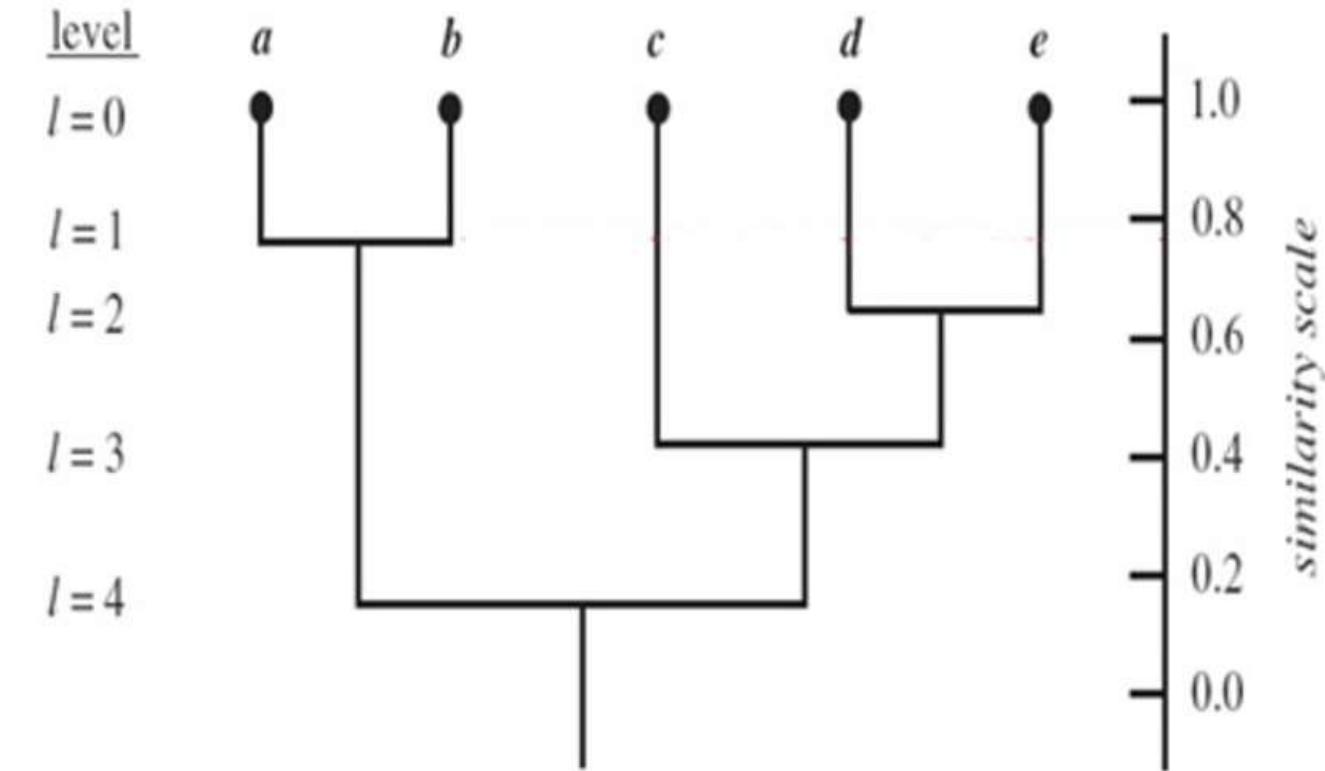
A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering.

It shows how objects are grouped together (in an agglomerative method) or partitioned (in a divisive method) step-by-step.

Example: consider dendrogram for the five objects  
*level 0 shows the five objects* as singleton clusters at level 0. At  $l=1$ , *objects a and b are grouped together to form the first cluster.*

We can also use a vertical axis to show the similarity scale between clusters.

For example, when the similarity of two groups of objects,  $\{a,b\}$  and  $\{c,d, e\}$ , is roughly 0.16, they are merged together to form a single cluster.



Dendrogram representation for hierarchical clustering of data objects  $\{a, b, c, d, e\}$ .

## Similarity measures

• The cluster is group of records or data points. Within, similarity  
The members within a cluster are similar and dissimilar with  
other members of clusters.  
clustering is unsupervised learning technique used to group similar  
data points together to form clusters.

### Types of similarity measures

Different types of data requires different similarity measures.

Some common types include:

numeric data: Euclidean, Manhattan, cosine

Binary data: Hamming, Jaccard

categorical data: Matching coefficient, Jaccard

Text data: cosine similarity, Jaccard

### Euclidean distance

It is suitable for continuous numeric data.

$$d(x, y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

### Manhattan distance

$$d(x, y) = |x_2 - x_1| + |y_2 - y_1|$$

It calculates the sum of absolute difference of two data points

### Cosine Similarity

measures the cosine of angle between two vectors.

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

$x \cdot y$  is the dot product of vectors  $x$  and  $y$

$\|x\|$  and  $\|y\|$  are the magnitudes (Euclidean norm) of A and B

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

## Jaccard Similarity

It measures the intersection over union of two sets, suitable for binary or categorical values.

$$J(X, Y) = \frac{A \cap B}{A \cup B}$$

## Hamming Distance

It is used to compare two binary strings of equal lengths.  
It counts the number of positions with different bits  
useful for binary or categorical data

## Pearson Correlation Coefficient

~~Measures~~ Evaluates the linear relationship between two variables.

$$\rho = \text{cov}(x, y) / (\sigma_x \cdot \sigma_y) \rightarrow \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \cdot \sqrt{\sum (y_i - \bar{y})^2}}$$

values ranges from -1 to 1.  
It is useful when the relationship strength is more important than absolute differences.

## Example

### Cosine Similarity

$\text{sim}(x, y) \rightarrow x \text{ and } y \text{ are vectors}$

$$\text{sim}(x, y)$$

$$x = \{3, 2, 0, 5\}$$

$$y = \{1, 0, 0, 0\}$$

$$\frac{x \cdot y}{\|x\| \|y\|}$$

$$x \cdot y = 3 + 0 + 0 + 0 = 3$$

$$\|x\| = \sqrt{3^2 + 2^2 + 0^2 + 5^2} = \sqrt{9 + 4 + 25} = \sqrt{38} = 6.16$$

$$\|y\| = \sqrt{1^2 + 0^2 + 0^2 + 0^2} = 1$$

$$\text{sim}(x, y) = \frac{3}{6.16 \times 1} = 0.49$$

## Expectation-Maximization Algoirthm

Expectation-Maximization (EM) algorithm is a **iterative method** used in unsupervised machine learning to find unknown values in statistical models. It helps to find the best values for unknown parameters especially when some data is missing or hidden. It works in two steps:

- **E-step (Expectation Step):** Estimates missing or hidden values using current parameter estimates.
- **M-step (Maximization Step):** Updates model parameters based on the estimated values from the E-step.

In the real world applications of machine learning, it is very common that there are many relevant features available for learning but only small subset of them are observable.

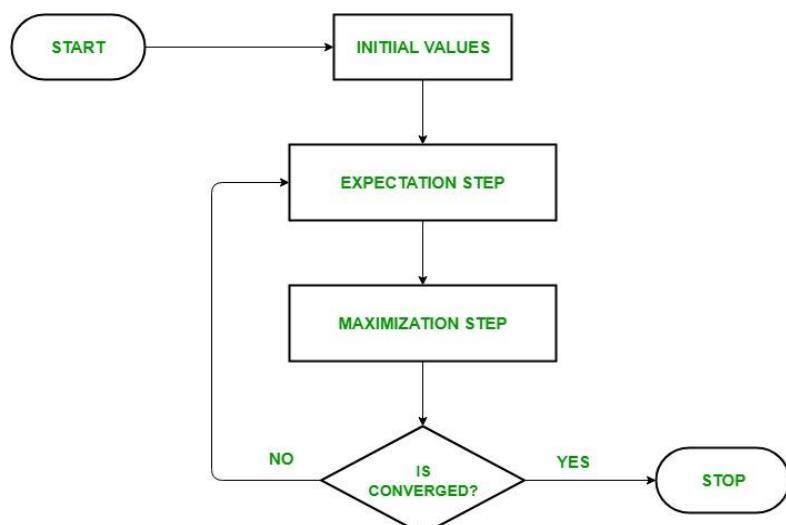
The Expectation-Maximization Algorithm can be used for the latent variables (variables that are not directly observable and are actually inferred from the values of other observed variables).

This algorithm is actually base for many unsupervised clustering algorithms in the field of machine learning.

### **Expectation-Maximization Algorithm**

1. Initially, a set of initial values of the parameters are considered. A set of incomplete observed is given to the system with the assumption that the observed data comes from a specific model.
2. Expectation Step (E-step) : Estimate the missing or hidden data using the observed data.
3. Maximization Step (M-step) : We use the complete data generated in the “E-step” in order to update the values of the parameters.
4. We check, whether the values are converging or not. If yes, then stop, otherwise repeat step 2 and step 3 until convergence occurs.

The process can be depicted using the following flow chart :



**Fig. 1. Flow chart for Expectation-Maximization Algorithm**

Uses of E-M algorithm

1. Handles incomplete data
2. Can model latent variables directly
3. Can be used for clustering

## Fuzzy C-means algorithm

clustering is a distance based clustering unsupervised ML algorithm where data points that are close to each other are grouped in a given number of clusters/groups.

in hard clustering, each data point is assigned only a single cluster. The K-means and K-medoids clustering algorithms are hard clustering algorithms.

in soft clustering, each data point belongs to a cluster with a certain probability also known as membership value. Fuzzy c-means clustering algorithm is an example of soft clustering.

### steps in Fuzzy C-Means clustering

step1 : Given the data points based on the number of clusters required to initialize the membership table with random values. suppose given data points are  $\{(1,3), (2,5), (6,8), (7,9)\}$

		data points			
		(1,3)	(2,5)	(6,8)	(7,9)
		cluster 1	0.8	0.7	0.2
		cluster 2	0.2	0.3	0.8

Step 2 : find out the centroid

The formula for finding out the centroid ( $V$ ) is

$$V_{ij} = \frac{\sum_{k=1}^m y_{ik}^m * x_k}{\sum_{k=1}^m y_{ik}}$$

$y_{ik}$  is membership value  
 $m$  is fuzziness parameter generally taken as 2  
 $x_k$  is the data point

$$v_{11} = \frac{(0.8)^2 * 1 + (0.7)^2 * 2 + (0.2)^2 * 4 + (0.1)^2 * 7}{0.8^2 + 0.7^2 + 0.2^2 + 0.1^2} = 1.568$$

First centroid  
first component

$$v_{12} = \frac{0.8^2 * 3 + 0.7^2 * 5 + 0.2^2 * 8 + 0.1^2 * 9}{0.8^2 + 0.7^2 + 0.2^2 + 0.1^2} = 4.051$$

~~First~~ First cluster centroid = 1.568, 4.051

$$v_{21} = \frac{0.2^2 * 1 + 0.3^2 * 2 + 0.8^2 * 4 + 0.9^2 * 7}{0.2^2 + 0.3^2 + 0.8^2 + 0.9^2} = 5.35$$

$$v_{22} = \frac{0.2^2 * 3 + 0.3^2 * 5 + 0.8^2 * 8 + 0.9^2 * 9}{0.2^2 + 0.3^2 + 0.8^2 + 0.9^2} = 8.215$$

centroids are: (1.568, 4.051) and (5.35, 8.215)

Step 3:

Find out the distance of each point from the centroid

$$D_{11} = \sqrt{(1 - 1.568)^2 + (3 - 4.051)^2} = 1.2$$

$$D_{12} = \sqrt{(1 - 5.35)^2 + (3 - 8.215)^2} = 6.79$$

$$D_{21} = \sqrt{(2 - 1.568)^2 + (5 - 4.051)^2} = 1.04$$

$$D_{22} = \sqrt{(2 - 5.35)^2 + (5 - 8.215)^2} = 4.64$$

$$D_{31} = \sqrt{(4 - 1.568)^2 + (8 - 4.051)^2} = 4.63$$

$$D_{32} = \sqrt{(4 - 5.35)^2 + (8 - 8.215)^2} = 1.36$$

$$D_{41} = \sqrt{(7 - 1.568)^2 + (9 - 4.051)^2} = 7.34$$

$$D_{42} = \sqrt{(7 - 5.35)^2 + (9 - 8.215)^2} = 1.82$$

step 4 : update membership values

$$\gamma_{ki} = \left( \sum_{j=1}^n \left\{ \frac{d_{kj}}{d_{kj}} \right\}^{\left(\frac{1}{m-1}\right)} \right)^{-1}$$

$$\gamma_{11} = \left( \left\{ \frac{(1.2)^2}{(1.2)^2} + \frac{(1.2)^2}{(6.79)^2} \right\}^{\left(\frac{1}{2-1}\right)} \right)^{-1} = 0.97$$

$$\gamma_{12} = \left( \left\{ \frac{(6.79)^2}{(1.2)^2} + \frac{(6.79)^2}{(6.79)^2} \right\}^{\left(\frac{1}{2-1}\right)} \right)^{-1} = 0.03$$

For point 2, new membership values are

$$\gamma_{21} = \left( \left\{ \frac{(1.04)^2}{(1.04)^2} + \frac{(1.04)^2}{(4.64)^2} \right\}^{\left(\frac{1}{2-1}\right)} \right)^{-1} = 0.93$$

$$\gamma_{22} = \left( \left\{ \frac{(4.64)^2}{(1.04)^2} + \frac{(4.64)^2}{(4.64)^2} \right\}^{\left(\frac{1}{2-1}\right)} \right)^{-1} = 0.05$$

The updated membership table.

		(1,3)	(2,5)	(4,8)	(7,9)
		0.97	0.95	0.08	0.06
cluster \ datapoint	1				
	2	0.03	0.05	0.92	0.94

Step 5 :

Repeat the steps 2-4 until the constant values are obtained for the membership values or the difference is less than tolerance value.