# Unsupervised Machine Learning

In regression target is continuous whereas in classification target is categorical. In case of Unsupervised machine learning we are not provided with any target feature. Since target is missing hence there is no supervision and that's the reason why it is called Unsupervised Machine Learning.

## Clustering [grouping]

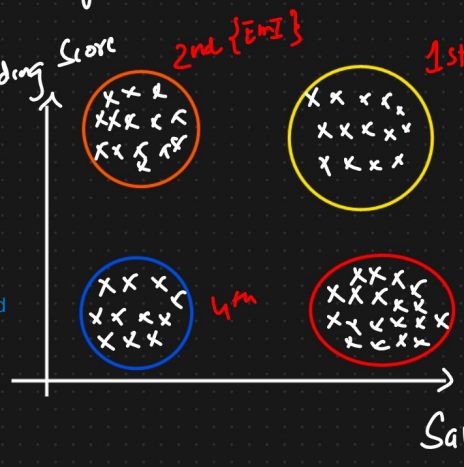
Different clustering algorithm who. Silhouette score is one of the performance metric that is being used for measuring the performance of clustering algo

1. K Mean Clustering
2. Hierarchal clustering
3. DBScan clustering

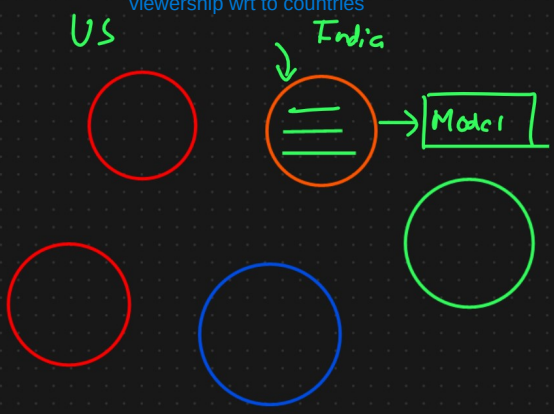
⇒ Performance Metrics Validate clustering algorithm.

⇓ Silhouette score

Ex;
Here we are making clusters based on Salary and Spending score. Suppose Apple needs to push notification to it's user to buy a product at 30% off then they will divide the cluster on priority basis on which notifications will be pushed to the respective users. Here company will first target to send frequent notification to 1st cluster followed by 2nd then 3rd and at last 4th. If they follow this approach then chances will be high that companies product get sold quickly.




Ex2;
Any Hindi movie will be recommended in India and then in other countries like USA since Hindi movie viewership is pretty high in India as compared to other countries This will be very clear if we form cluster of Hindi movie viewership wrt to countries

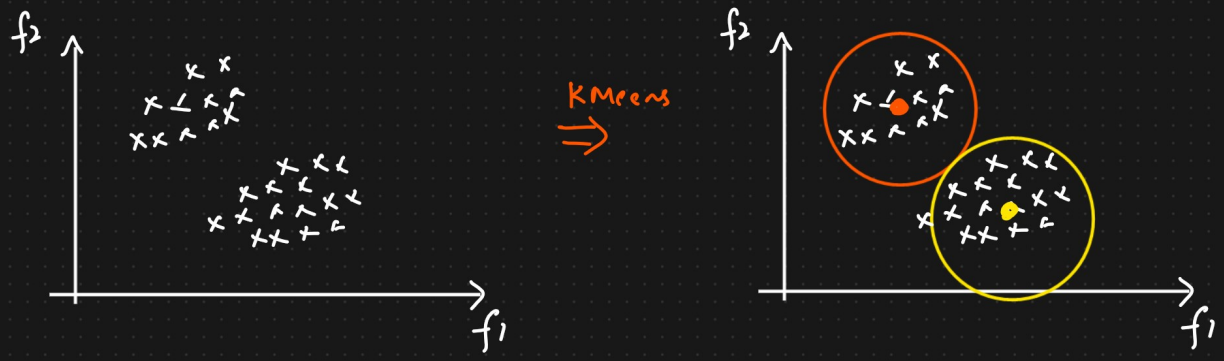



In unsupervised learning there is no output but INSIGHTS.

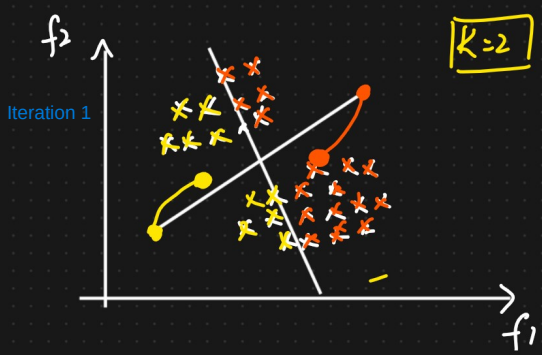
## 1 K Means Clustering

(Centroid based approach where,
K is no. of centroid based on which exact same no. of clusters will be formed.
That is number of clusters = k value)

### Geometric Intuition



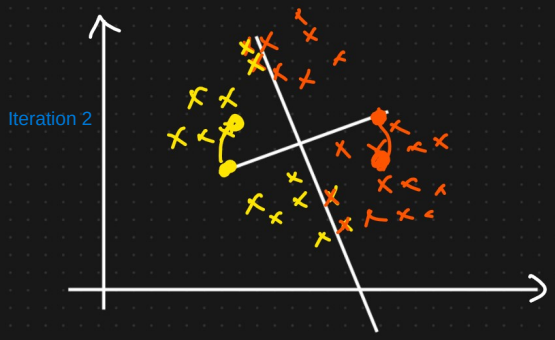

### K Means Mathematical Intuition

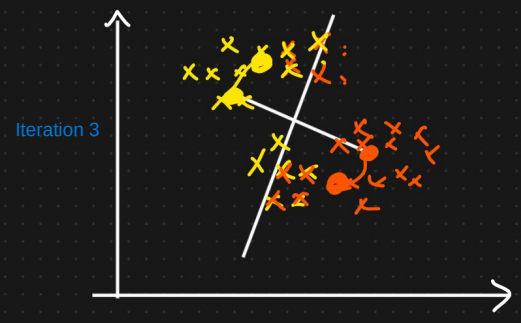


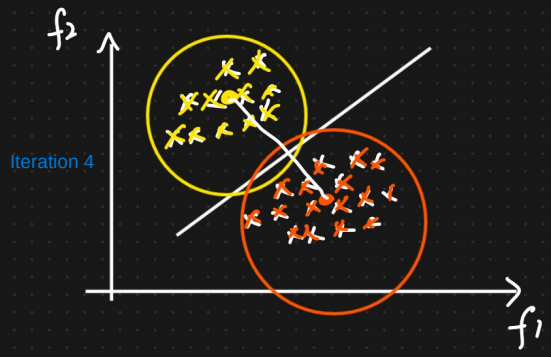

**Steps.** K centroids

1. Initialize some K value
2. Label all the points based on distance to the Nearest Centroid
3. Move the centroid

Move the centroid to the new point based on the average value of data points for respective clusters means moving centroid to the point where the respective cluster is more concentrated or dense.




In the last iteration we can clearly see that perpendicular bisector of line joining the 2 centroids points has clearly divided datapoints into 2 clusters with no overlapping. Here is the stage where we should converge our algorithm.

## How do we select the K value [Elbow method] ⇒ Knee locator

WCSS = Within Cluster Sum of Square.

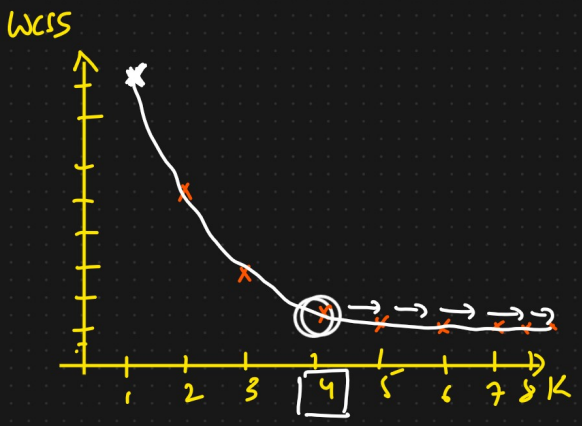
Initialize K: 1 to 10


K=1

K=2

$$WCSS = \sum_{i=1}^{K} (\text{Distance between point to the nearest centroid})^2$$

WCSS




## Euclidian Distance

K ⇒ Number of centroids.

Plot k with wcss and plot. Point at which graph starts becoming parallel to x axis or gives elbow shape or point of decrease becomes negligible or very small we may select that respective k value

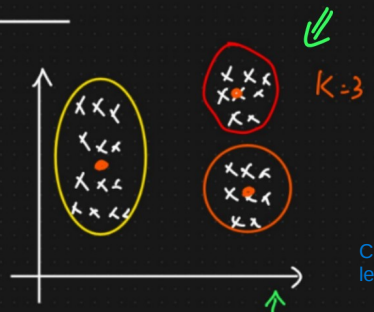
In above example we should select K=4 as the centroid value.

How are we deciding the location of centroid? Ans; By using Random initaialization trap approach.

## Random Initialization Trap (K Means ++) ⇒

Observe this data distribution and imagine the clusters to be formed

K Means ⇒

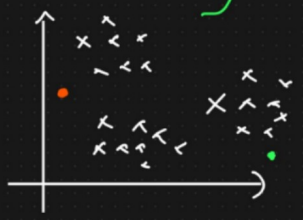



Cluster formed after applying k means ++ leads to correct formation of clusters.

If centroids are closely initialized then there is high possibility that clusters formed may be wrong. So we use K mean ++ that initializes centroids far from each other in random fashion.

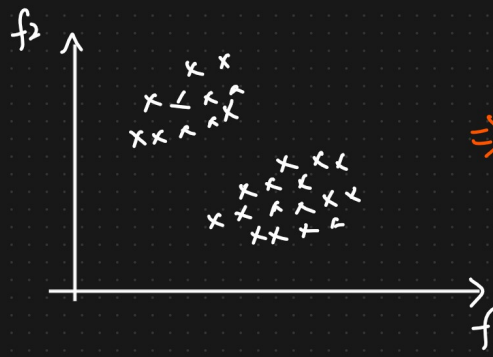
When centroids initialized closely, leads to incorrect cluster formation

⇒ K Means ++

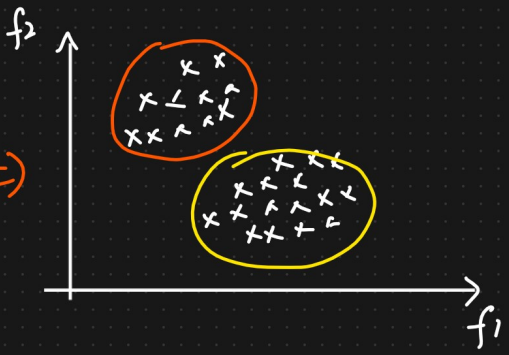


## ② Hierarichal Clustering [Agglomerative Clustering]

There is no centroid used in this approach for the formation of the clusters




⇒ Hierarichal Clustering ⇒

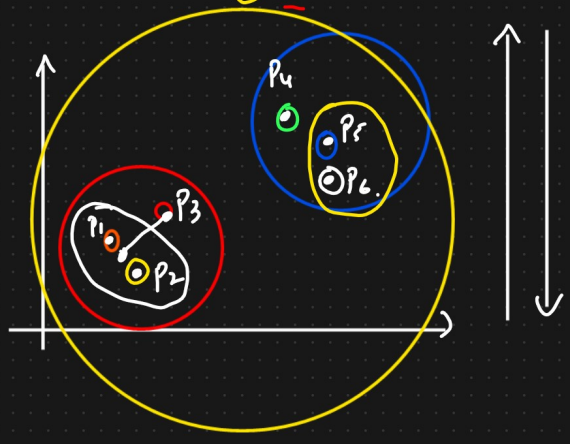



Types of Hierarchal Clustering

HC

① Agglomerative
② Divisive

⇒ Geometric Intuition




Steps

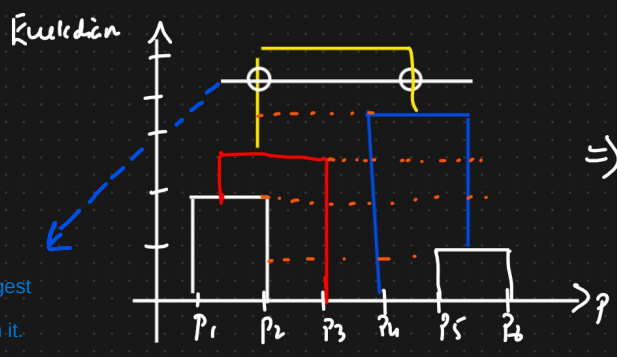
① For each point we will consider it as a separate cluster

② Find the nearest point and creak a new cluster

Euclidian Distance

In Agglomerative we agglomerate or aggregate or combined the smaller groups thus forming the larger group.

Whereas, in Divisive we do the reverse. That is we will start with the larger group followed by splitting in into smaller groups.




⇒ Dendrogram.

Graph b/w Euclidean distance and datapoints is called Dendrogram where each bar represents same group

How to consider the number of groups or clusters:
In the dendrogram find the longest straight line such that no horizontal lines passes through it. Make a slice of such a line and the number of points it passing through it will be equivalent to number of groups.

Please note here P1 and P2 is having same distance since Euclidean distance is calculated from the pt. in b/w P1 and P2. Resultant of which is then compared with P3 by calculating distance from pt in b/w them. Same continue to happen and we get the dendrogram

## K Means Vs Hierarchical Clustering

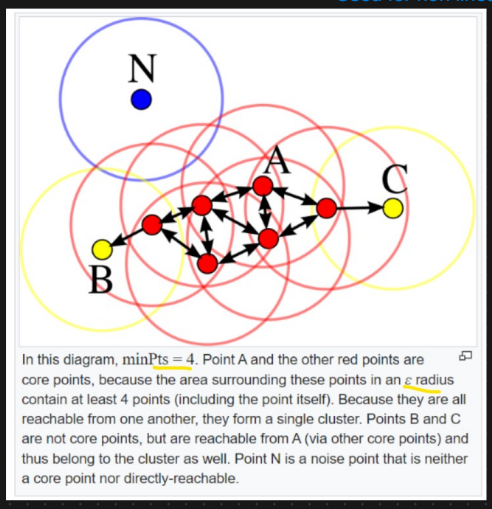
### Scalability and Flexibility

① Dataset size → Huge ⇒ K Means
→ Small ⇒ Hierarchical Clustering

Since visualizing dendrogram will not be possible in case of large dataset hence Hierarchical Clustering should be avoided in such cases.

### ③ DBScan Clustering

K means affected by outlier and group them as well. But DBSCAN clustering approach makes sure that the outliers are not grouped.
Used for non linear clustering.




In this diagram, minPts = 4. Point A and the other red points are core points, because the area surrounding these points in an ε radius contain at least 4 points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but are reachable from A (via other core points) and thus belong to the cluster as well. Point N is a noise point that is neither a core point nor directly-reachable.

- Core points
- Border points
- Noise/outliers

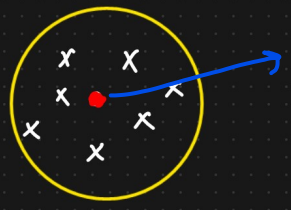
} Non linear Clustering.

Hyperparameter

① minpts = 4 ② ∈ = radius

minpts represents the minimum no. of datapoints

Core point ← ① No. of data points within the ∈ ≥ minpts

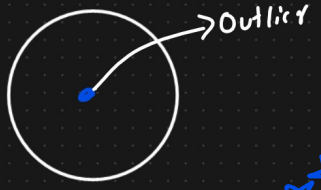


Please note that this is also a datapoint itself. What we are basically doing over here is that for all the data points we are drawing the circle with epsilon radius and then deciding whether that will be core point, border point or an outlier

② Border points

No. of data points within the radius is less than minpts = 4

## ③ Outliers [DBScan is robust to outliers].

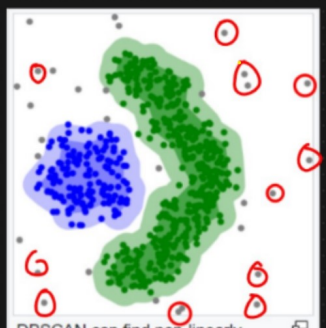



For any datapoint if draw a circle with epsilon radius and if within the resultant circle if there is no other datapoint present then the respective datapoint is treated as Outlier

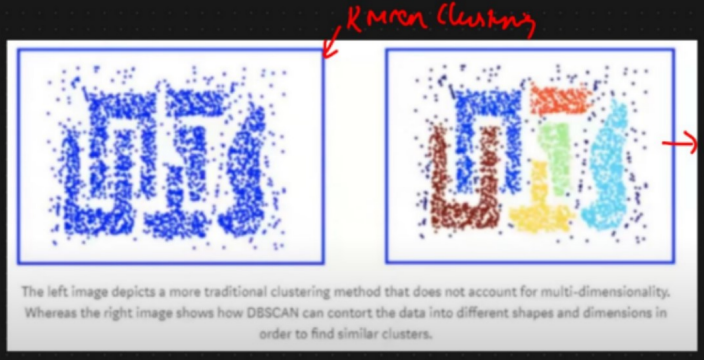
One of the most important use case of DBSCAN is an ANOMALY DETECTION

## Some Examples after we apply DBScan Clustering

Understanding cell structures, Genome sequencing etc.



DBSCAN can find non-linearly separable clusters. This dataset cannot be adequately clustered with k-means or Gaussian Mixture EM clustering.




The left image depicts a more traditional clustering method that does not account for multi-dimensionality. Whereas the right image shows how DBSCAN can contort the data into different shapes and dimensions in order to find similar clusters.

## Silhouette Score [Validate the clustering model].

① First step

For data point $i \in C_I$ (data point i in the cluster $C_I$), let

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j)$$

be the mean distance between i and all other data points in the same cluster, where $|C_I|$ is the number of points belonging to cluster $C_I$, and d(i,j) is the distance between data points i and j in the cluster $C_I$ (we divide by $|C_I| - 1$ because we do not include the distance d(i,i) in the sum). We can interpret a(i) as a measure of how well i is assigned to its cluster (the smaller the value, the better the assignment).

a(i)

② 

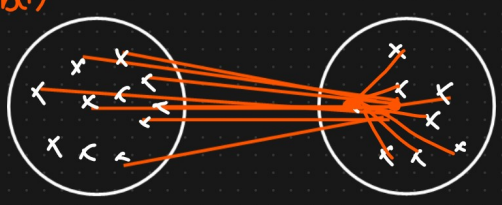
We then define the mean dissimilarity of point i to some cluster $C_J$ as the mean of the distance from i to all points in $C_J$ (where $C_J \neq C_I$).

For each data point $i \in C_I$, we now define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j)$$

b(i) > a(i) ⇒ Clustering is good.

b(i)



We now define a *silhouette* (value) of one data point i

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_I| > 1$$

and

$$s(i) = 0, \text{ if } |C_I| = 1$$

Which can be also written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

From the above definition it is clear that

$$-1 \leq s(i) \leq 1$$

If S(i) is more towards 1 then respective clustering model is better. Whereas, if S(i) is more towards -1 then respective clustering model is worse.