Unsupervised Learning

- In Unsupervised Machine learning, we don't have specific output, we have feature 1 and feature 2 So we make **Clusters** of similar kind of data in unsupervised Machine learning Algorithms.

Where does clustering get used?

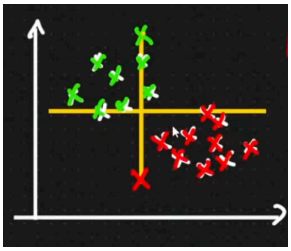
- In custom ensemble technique, First of all we creat clusters after that we apply clustering algorithms and it's good that we apply either regression or Classification. Suppose we have 3 groups, So for each groups we apply supervised algorithms.

What K-means?

- Here, k is nothing but centroids, let's say our K value is 2 that mean we will get 2 clusters and each clusters will have centroid point.
- So how we conclude that we have 2 groups because we cann't say okay so we have two groups as we have **high dimensions data** So, for that we need to **perform some steps.**

  Steps:

  1. We try different **K-values and** which is suitable value. Let start k value from 2.
  2. Now second step is we initialize K number of centroid after that we will find which poins are near to those centroids, in order to find out we will use euclidian distance. In easy way what we can do is we joint two centroids with straight line and draw another interception line like shown in below image.So, points which are near to green will be turned into green, likewise whichever points are near to red will be turned into red.



3. We will compute average of red and green group to update the centroids and after the update again it repats until all points are there own location then further there is no updation of centroids, even if there is one red point in green area that point will be turned into green.

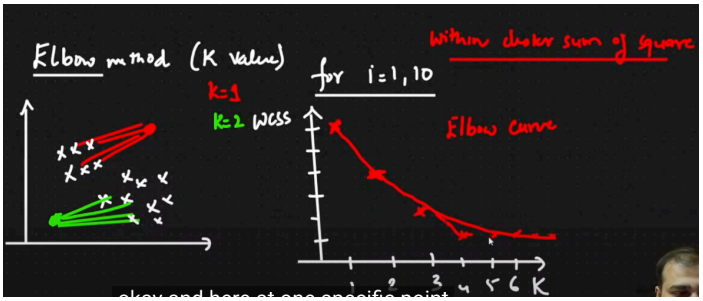
So, thats the steps will be performed, however how we will decide the **K-value** for that There is a method which is called **Elbow method.**

NOTE:

- Also, we need to find centroid very very far then we are able to find centroids exactly in the center, Otherwise we wanted two centroid as we have k=2 and we got 3 centroids.
- In order to do that we use **K-mean ++** algorithm, it will ensure that all centroids are far from each others.

Elbow method

- We will go through the iteration like from 1 to 10, and for each iteration we will create graph with respect to **K value** and **WCSS (**Within clusters sum of square**).**
- **For k=1**, now we have one centroid, so we will calculate the distance from each points to centroid which will definitely give us a grater distance. **Likewise, if** K value increase, **wcss** will be decreased and after some k value it will be normal.That's why it's known as **Elbow** method. We will check which is feasible K value where we see abrupt change, therefore as per the image we will take **k value 4** as after that value wcss is going to be normal.

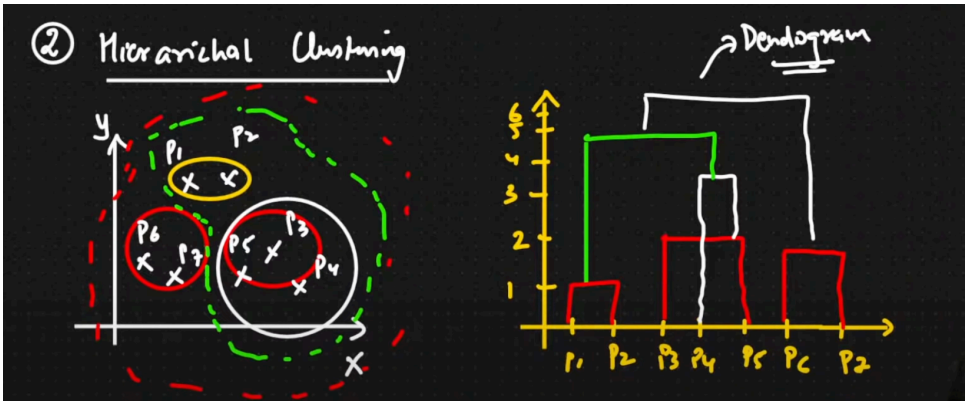


Hierarchical clustering

Let's say we have 8 points where p1 and p2 are nearest name as C1 so we make one cluster there, likewise p6 and p7 are in another separate cluster name as C2, and p5 and p3 are in another cluster name as C3.

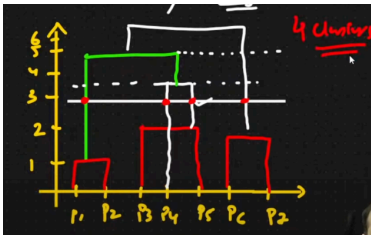
Now, p4 is nearest to the C3 cluster, now we will make that cluster, after that there are three clusters where p1and p2 are in C1, and p6 and p7 are in C2, while in C3 there are p4,p5,p3.

After this, we will see that C1 is nearest to C3 so we will combine that in one cluster and then this cluster will be nearest to the C2 so we do the same thing here. Take a look into image below. The graph on right side name as **Dendogram.**



How do we find that how many groups should be here?

We need to find the longest vertical line that has no horizontal line passed through it. Once we find it then we will pass horizontal line through it and see how many vertical line it passing throug it(as shown in below image). So there will be 4 clusters.



Which has taken maximum time Kmeans or Hierarchical clustering?

Answer is Hierarchical clustering as we have more data it will make more dendogram. So, if our dataset is **large** go with **Kmeans** or it is **small** then go with **hierarchical** clustering.

Validation in Clustering

In clustering, we validate through **silhouette score**. It can be validated for **Kmeans** and **Hierarchical.**

First and foremost, we will try to find out **a(i)** , how do we do that in cluster we find the distance from centroid to all points and get the average of it, as per the **equation** we see that in **d(i,j) , i** is centroid and **j** is all points. Where |Ci| is the **number of points belonging o cluster Ci (**We divide by |Ci|-1 as we don't include the distance d(i,j) in the sum.**)**

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

Secondly, we calculate **b(i),** we find out the nearest cluster of the cluster that we use for **a(i) then** we calculate the distance from **each points of one cluster to each points of other cluster and get the average of it.**

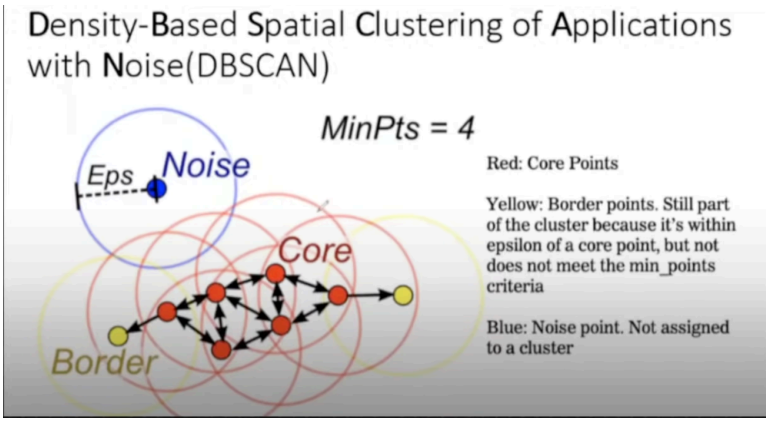
$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

After all this, our silhouette formula will be like this,

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

If we have **good model,** we will get **b(i) > a(i). In silhouette,** our value will be between **-1** and **1. So,** if our value near to 1 it means we have good model and **b(i) > a(i), while** if value is near to -1 it means we have worst model and there **a(i) > b(i).**

DBScan



- There are some terminologies like **Epsilon, core points, Minpts, border points, noise points.**
- Epsilon means it's the radius of circle(Cluster).
- MinPts means it's the hyperparameter.
- Let's say we have 4 **MinPts,** it means we have **four points** in the cluster then, epsilon of that cluster will be known as **Core Points.**
- If we have one **Core point,** the epsilon of the cluster will be become **Border Points** as shown in the image**.**
- If in the cluster we don't have any point, it will become **Noise point / Outliers.**