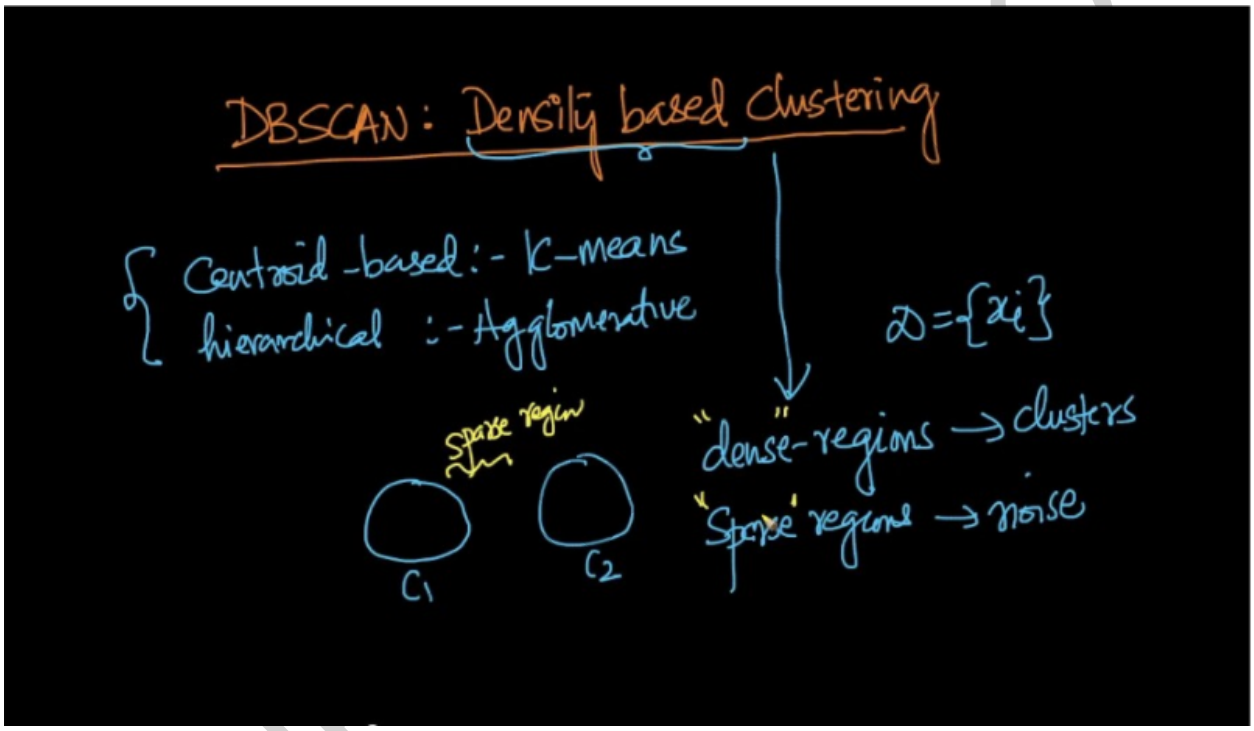
52.1 Density based clustering

In the previous modules, we have studied centroid-based clustering (K-means) , hierarchical clustering. In this chapter, we are going to study density based clustering.

DBSCAN : Density-based spatial clustering of applications with noise



Timestamp : 01:16

As usual, we are given a bunch of points.
Let's call them D : {xi} i = 1 to n where n represents the total number of data points.

Our aim is to partition these xi points into dense regions and sparse regions. Typically the dense regions become the clusters and sparse regions become the noise.

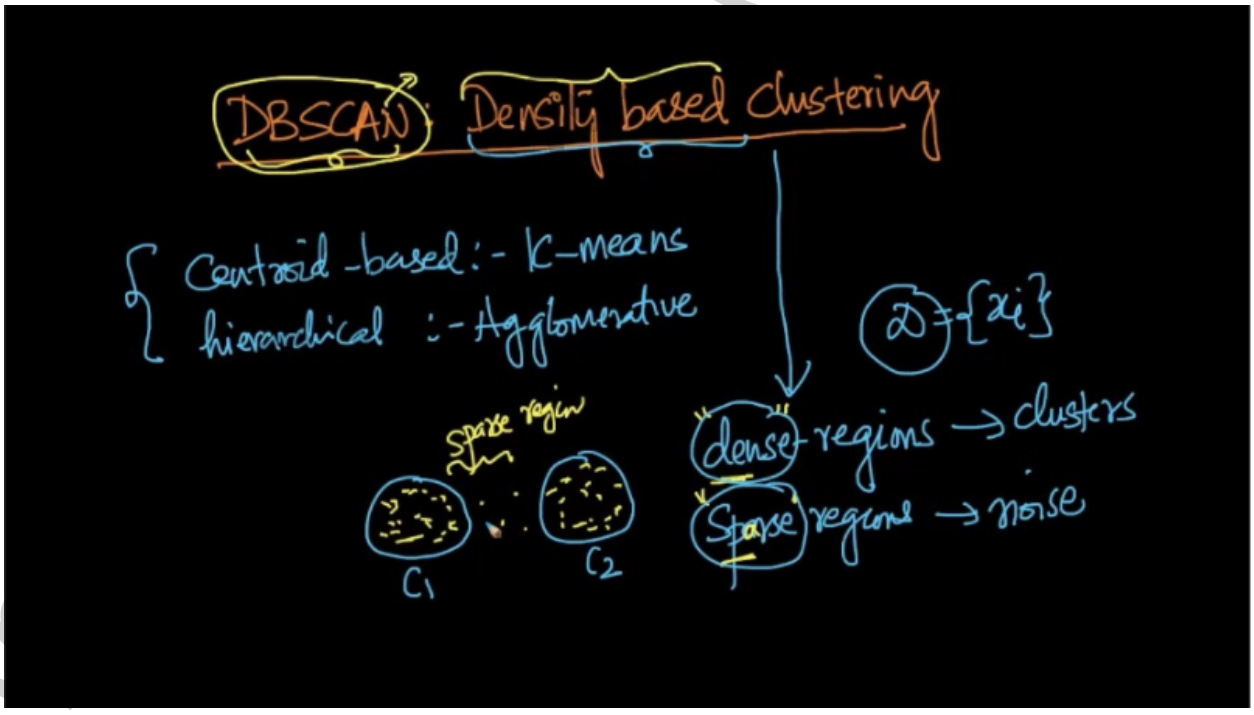
What is dense and sparse in simple terms?

Dense : It basically means, the points are clustered together.
Sparse : It basically means, there is some empty space in the region of where those xi's live.

How to mathematically quantify the above terms ?

Before proceeding with this, let us look at the important terms we are going to learn in this chapter.

1.) **Min point**
2.) **Epsilon**
3.) **Core point**
4.) **Border point**
5.) **Noise point**



Timestamp: 04:15

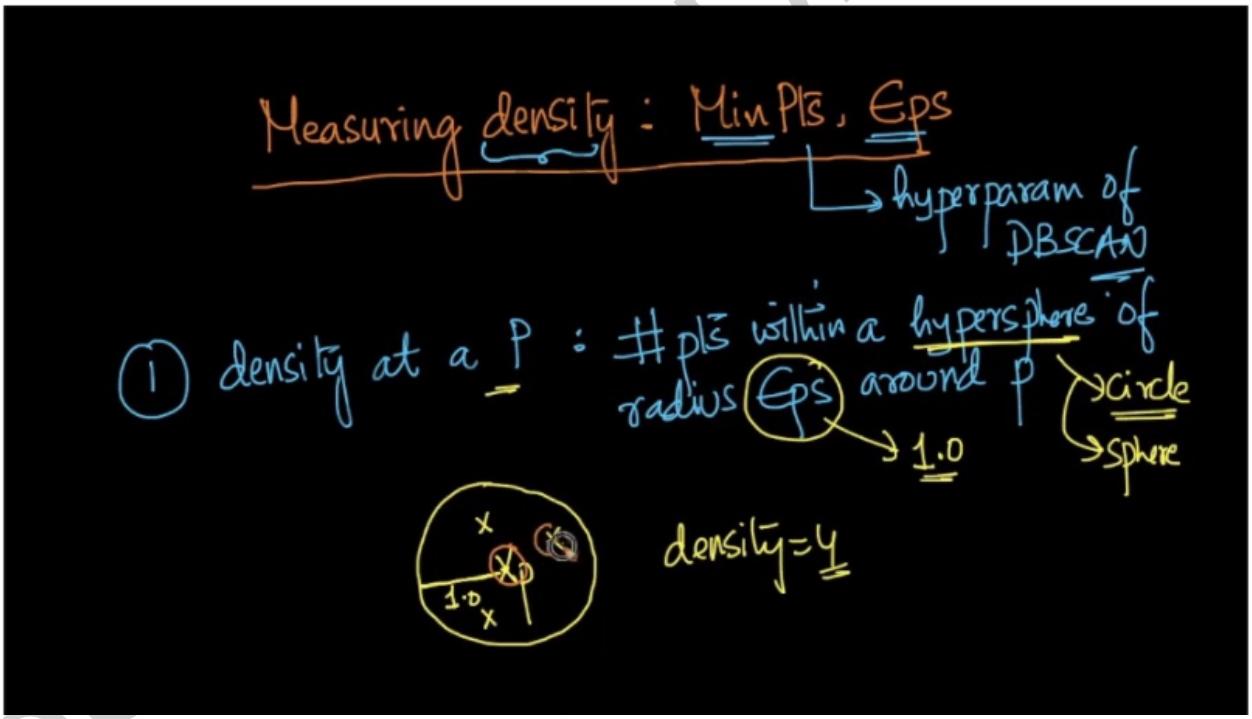
As we can see from the above image, dense regions are usually contained in the cluster and sparse regions are where noise points are present.

52.2 MinPts and Eps : Density

MinPts and Eps are the hyperparameters for the DBSCAN algorithm.

1.) **Density at a point P :** Number of points within a hypersphere of radius Epsilon around P . (Hypersphere - A sphere in n-dimensional space . For example if n = 2 , then we call it as a circle)

Let's consider an example.



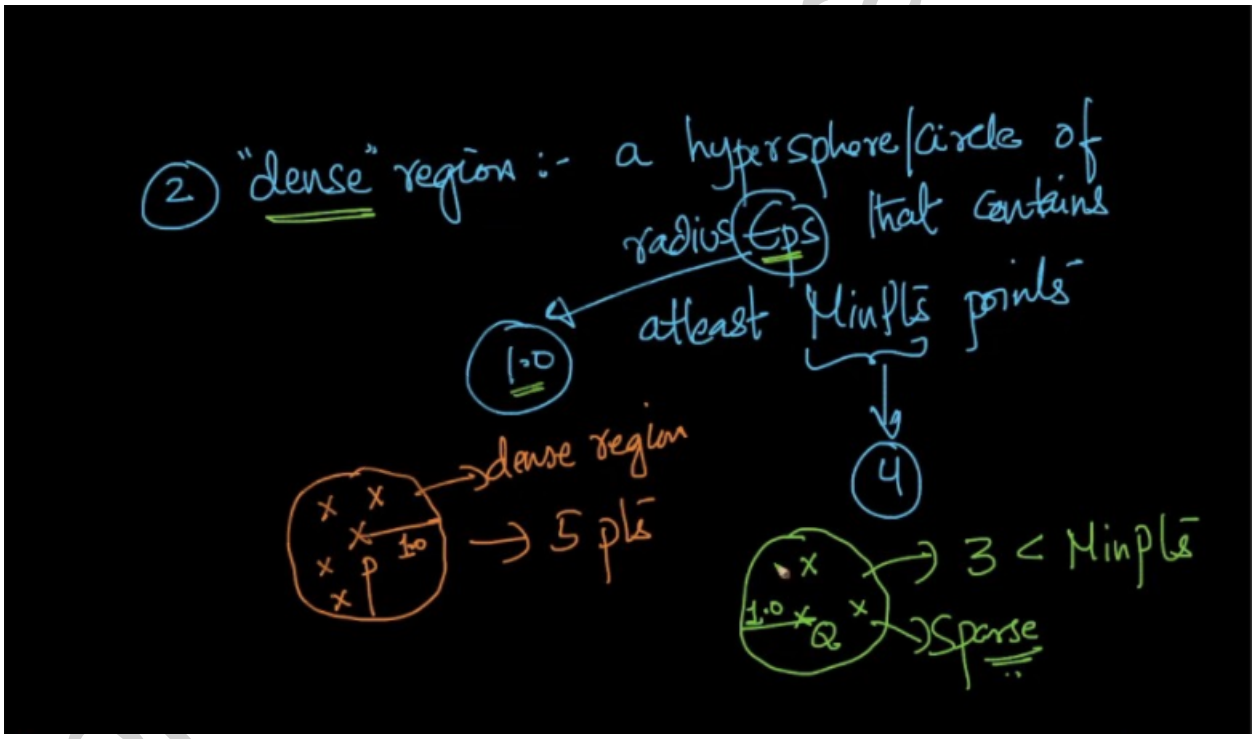
Timestamp : 02:21

Using the definition given above, we can calculate the density at a point P.

1.) First, draw a hypersphere of radius 1(Epsilon) with the point P as the centre.
2.) Next, count the number of points which lie inside the hypersphere.
3.) Include the point P also in the count.
4.) The count simply represents the density at a point P
5.) Here, density is 4.

2.) **Dense region :** A hypersphere of radius Epsilon that contains at least MinPts points.

Let's take an example.
Let's assume MinPts is 4 and Epsilon is 1.



Timestamp : 04:42

1.) Let's consider the hypersphere which is colored orange. We will first draw a hypersphere of radius 1 around the point P. Now, we will count how many points lie inside that hypersphere. We can see there are 5 points(including the point P). Since 5 > MinPts, we conclude that as a dense region.

2.) Let's consider the hypersphere which is colored green. We will first draw a hypersphere of radius 1 around the point P. Now, we will count how many points lie inside that hypersphere. We can see there are 5 points(including the point P). Since 3 < MinPts, we conclude that as a sparse region.

52.3 Core, Border and Noise points

D = {xi} i = 1 to n.

Given a point xi from the dataset D, we can categorize them as corepoint or border point or noise point.

1.) **Core point :** A point P is said to be core if it has greater than or equal to MinPts points in an Epsilon radius around it. In simple terms, we will first draw a hypersphere of radius Epsilon with point P as a centre. Then, we will count the number of points which lie inside that hypersphere including the point P itself. If that count is greater than or equal to MinPts, then we declare the point P as core point. It always belongs to a dense region. Note : Dense region : A hypersphere of radius Epsilon that contains at least MinPts points.

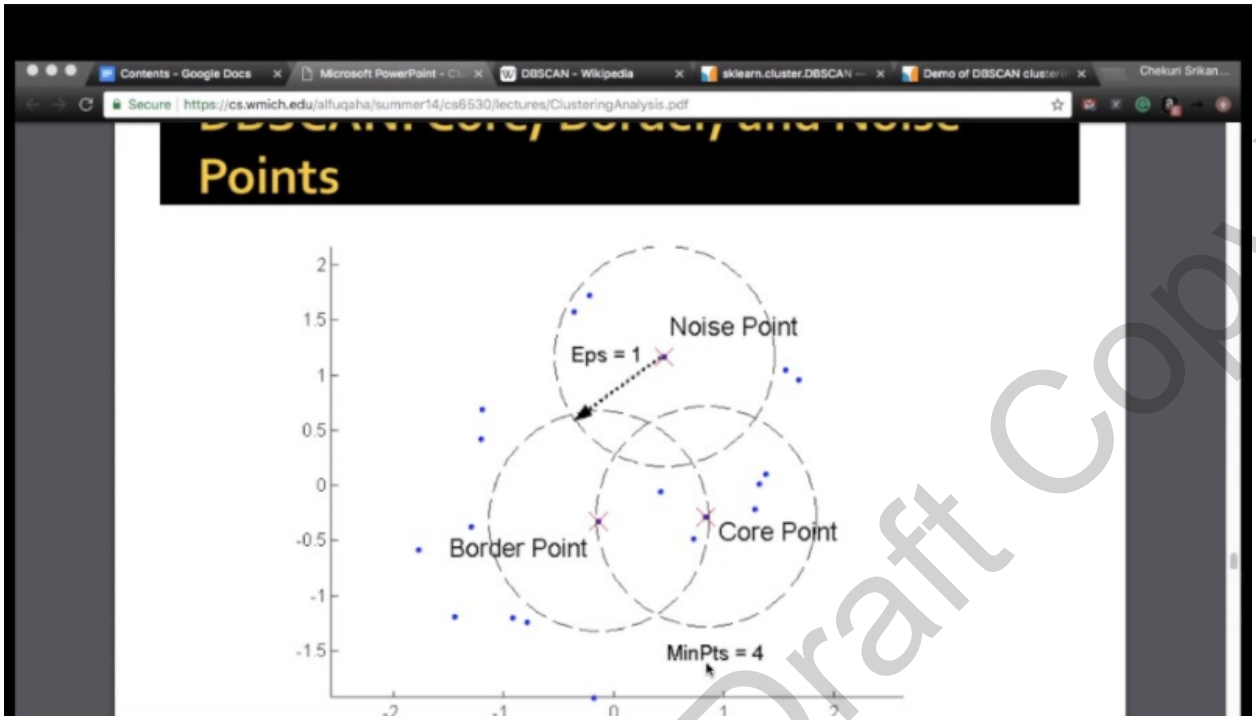
2.) Border point :

a.) A point P is said to be a border point, if p is not a core point. This means, P has less than MinPt points in Epsilon radius.

b.) Also p lies in the neighborhood of another core point Q i.e, distance between P and Q is lesser than or equal to Epsilon. Note : If a point A lies in the neighborhood of another point B, then distance between them is lesser than or equal to Epsilon.

3.) **Noise point :** Any point P which is neither a core point nor a border point is termed as a noise point

We will look at some illustrative examples.

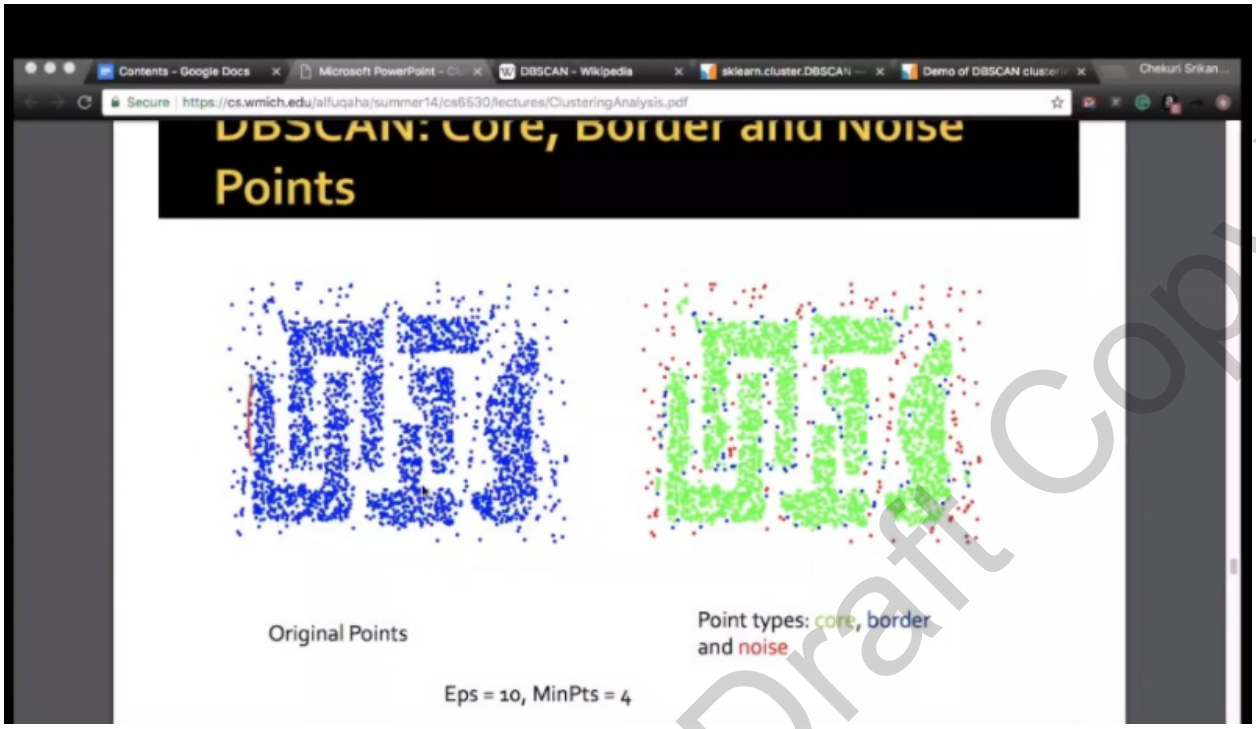


Timestamp : 03:57

The points marked with X are taken as reference points.

1.) The point marked with X at the rightmost corner is a core point. We can see there are more than MinPts inside the hypersphere.
2.) For the border point, there are less than MinPts around it. Also , note the border point belongs to the neighborhood of the core point i.e, their distance is less than or equal to 1 (Epsilon). These two conditions must satisfy the border point.
3.) The noise point is neither a core point nor a border point.

Let’s consider a real dataset.

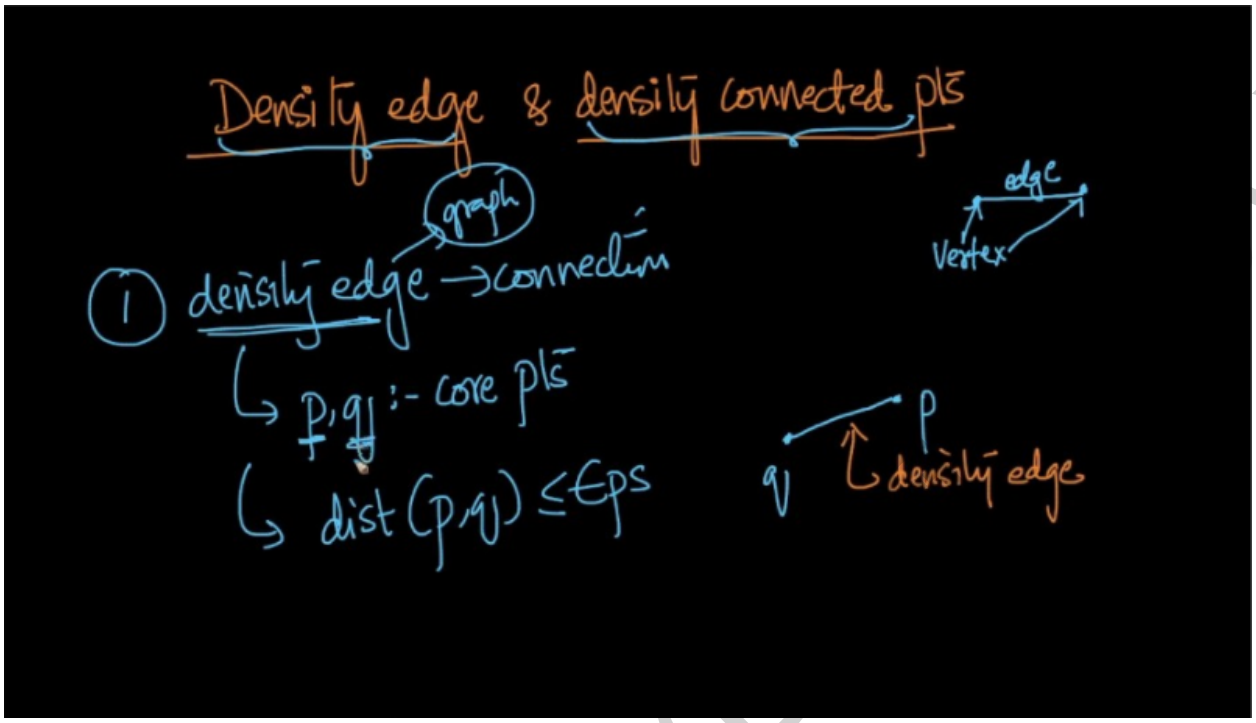


Timestamp : 05:54

Using the above definitions given, please try to categorize the points.

52.4 Density edge and Density connected points

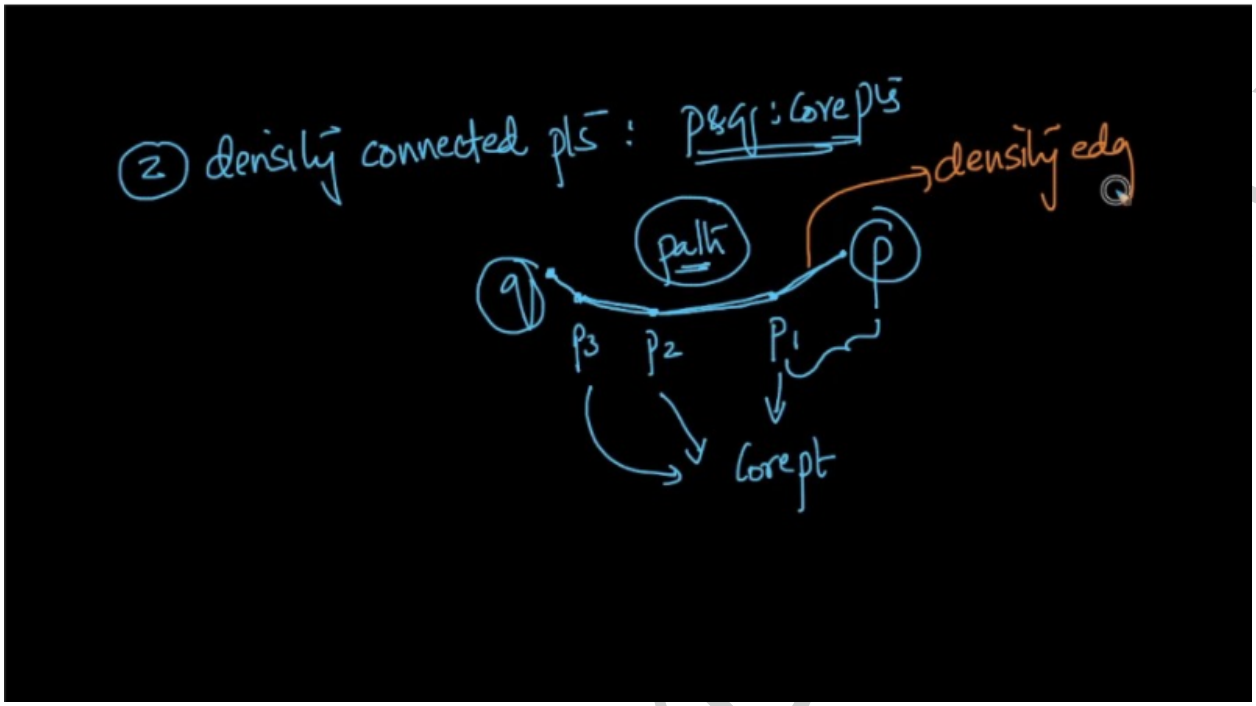
1.) **Density edge :** If P and Q are two core points and if the distance between P and Q is lesser than or equal to Epsilon, then we create a graph with two vertices P and Q and connect them with an edge.



Timestamp : 01:33

Please refer to the above image for the graph.

2.) **Density connected points :** P and Q must be a core point. This is slightly a different definition.

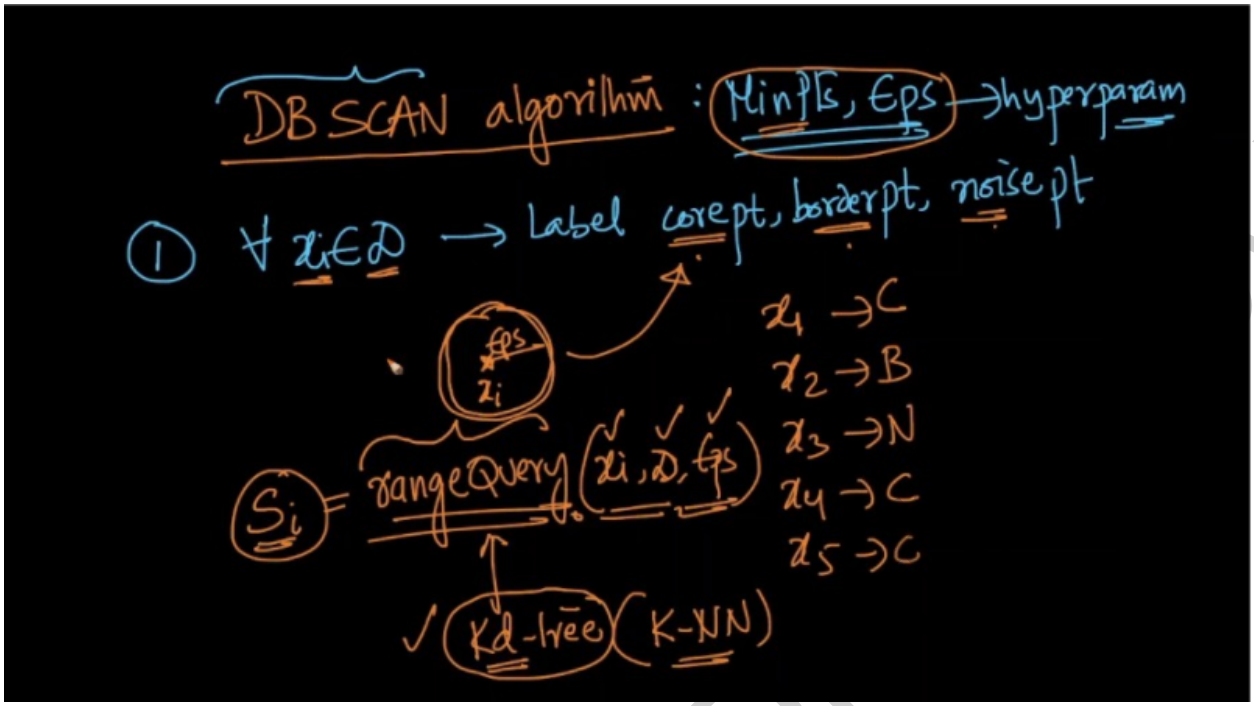


Timestamp : 03:12

P and Q are said to be density connected points if there exists a path between P and Q and in that path there are also other points which have density edges.

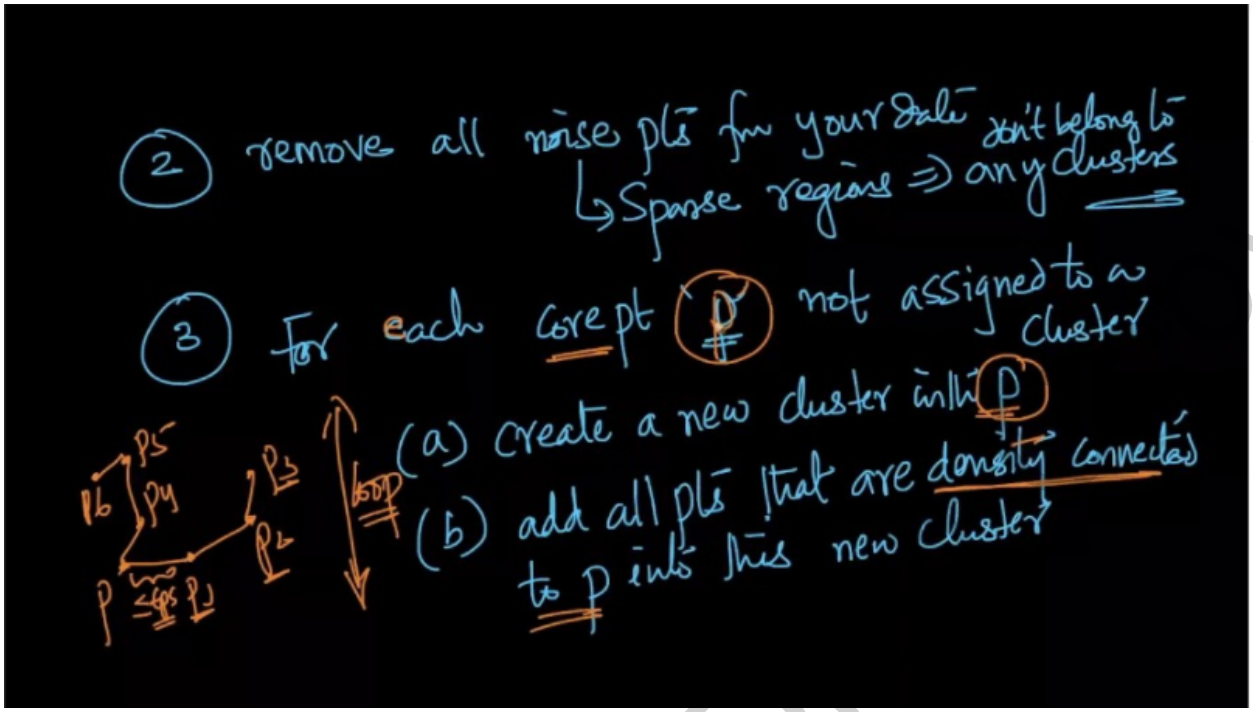
In simple terms, p1 - p has density edge , p2-p1 has a density edge , p3-p2 has a density edge q-p3 has a density edge and p-q has a density edge. Please refer to the definition for density edge above. This is simply a generalization to the definition of density edge.

52.5 DBSCAN algorithm



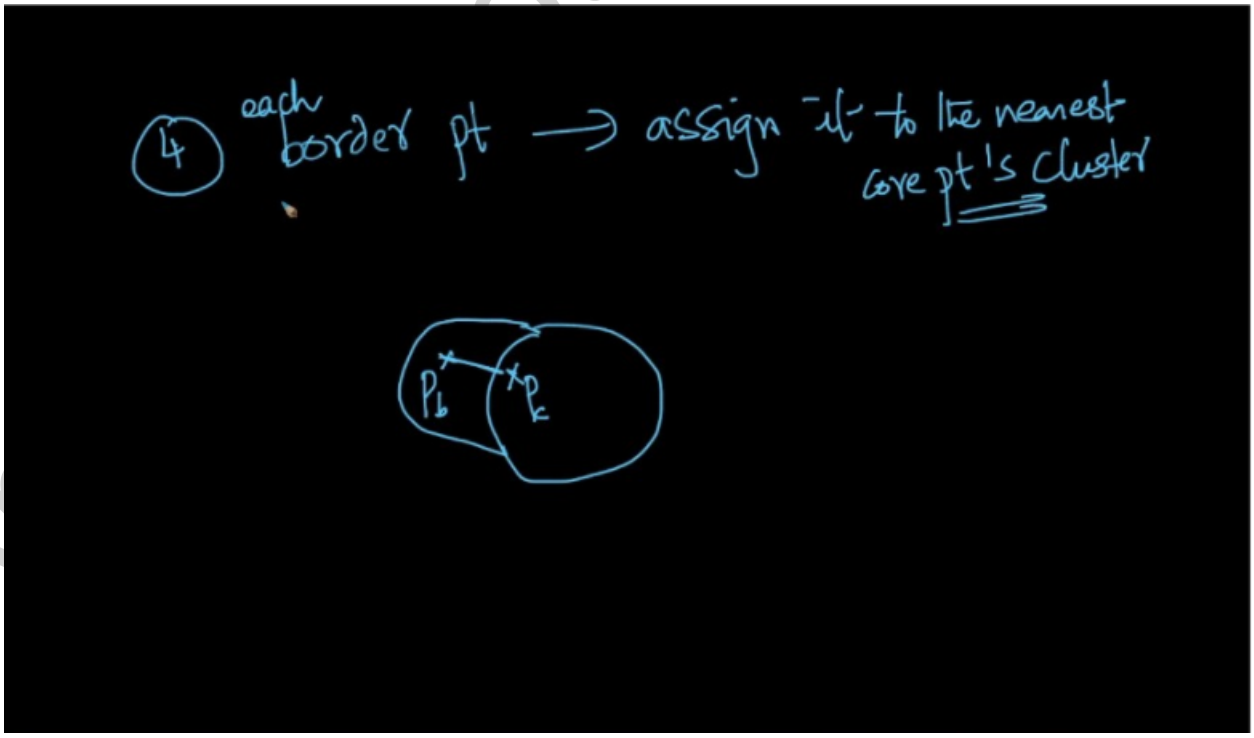
Timestamp : 03:06

1.) For each point xi belonging to a dataset D, label them as Core point, border point,noise point.
2.) For checking the above conditions and for labelling, we can use range query.
3.) Range Query basically returns a set of points which are Epsilon distance away from a point P. For example, a range query (P, Epsilon) will return a set of points S which is Epsilon distance away from P. It's implemented using kd-trees. We learnt about kd-trees in the K-NN module.
4.) Remove all noise points i.e, sparse regions.
5.) For each core point P that is not assigned to a cluster (Initially, none of the core points is assigned to a cluster).
    5.1) Create a new cluster with P
    5.2) Add all the points that are densely connected with P.



Timestamp : 07:12

6.)Assign each border point to the nearest core point's cluster.



Timestamp : 08:28

The core computation we need to perform is a range query. With kd-tree, the range query operation can be done in O(log(n)).

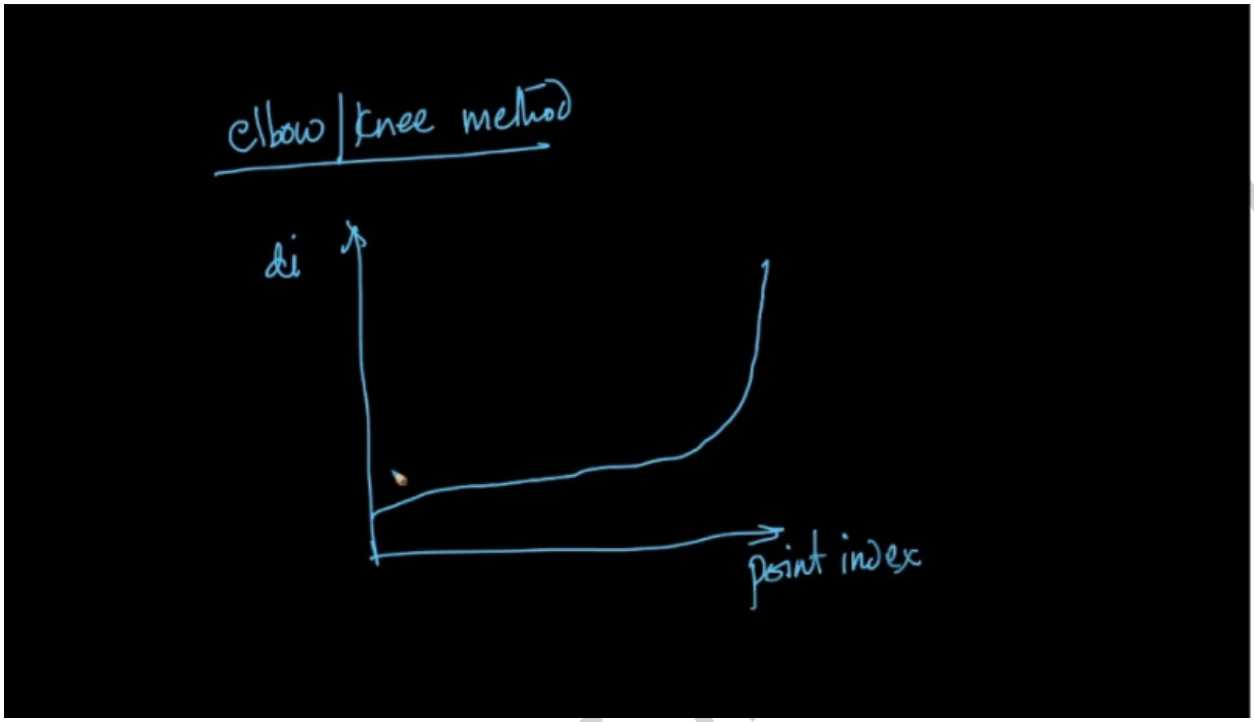
52.6 Hyper Parameters : MinPts and Eps

1.) **MinPts :** A rule of thumb is to have MinPts>=d+1 where d is the dimensionality of our dataset. Typically, MinPts can be set to 2*d.

1.1) If your dataset is noisy, then have a large value for MinPts. It's because, if we have a large value for MinPts, then the chance of the noise point being classified as a core point is very less. Note : MinPts is a main criterion deciding whether a point is a core point or not.

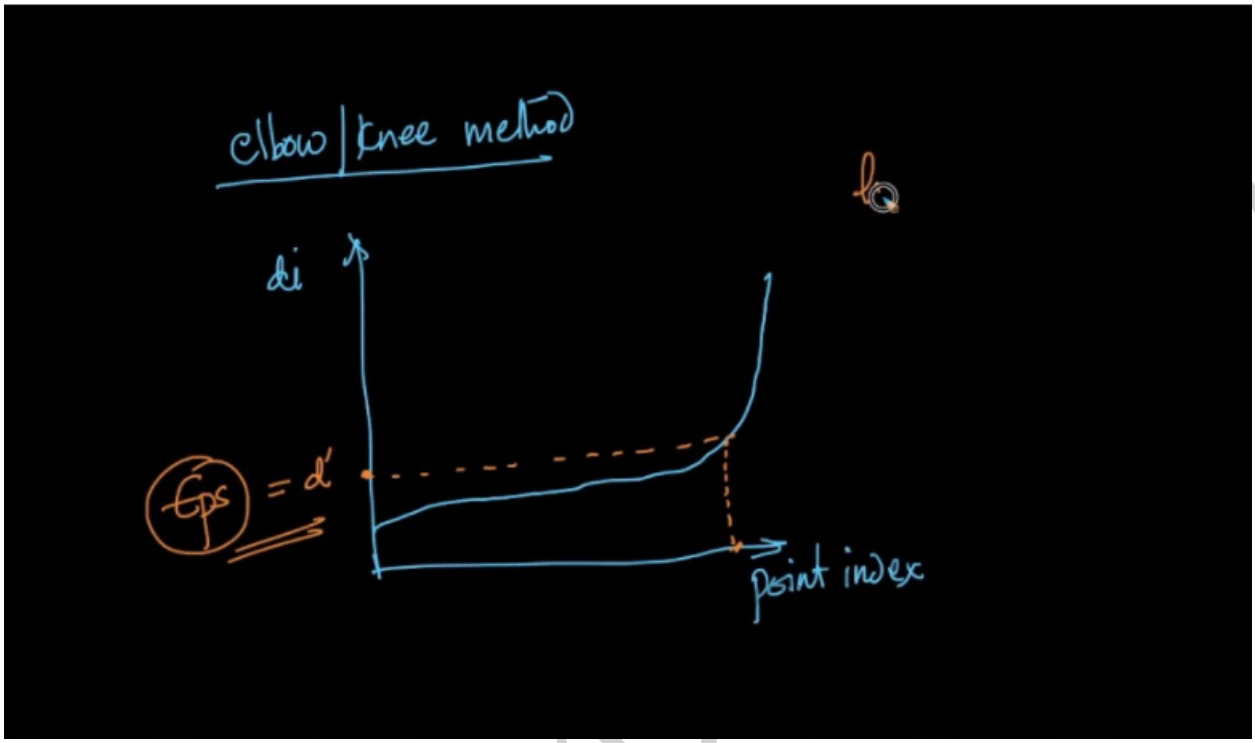
1.2) It is often chosen by a domain expert.

2.) **Epsilon:** Let's say we are given a value for MinPts as m.
2.1) Given a xi, first compute the distance between xi and the nearest neighbor of xi.
2.2) For each point xi, compute the distance and get the list of distances.
2.3) Now, sort the distances in increasing order.
2.4) Simply plot the distances and the point index.



Timestamp : 06:19

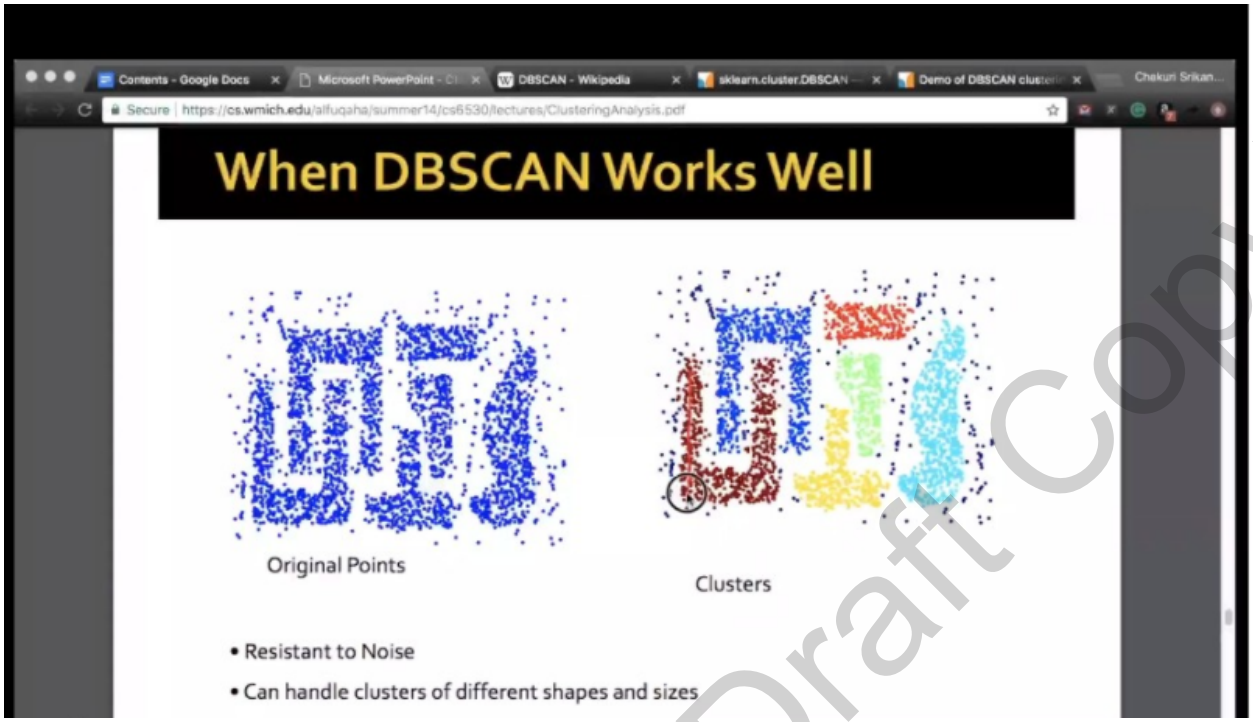
We can now use the elbow method which is discussed earlier in the course and then find the inflexion point. It is a point where the curve changes its direction. Then corresponding to it, we choose the epsilon.



Timestamp : 06:57

If di is high, then the chance that xi is noisy is also high.

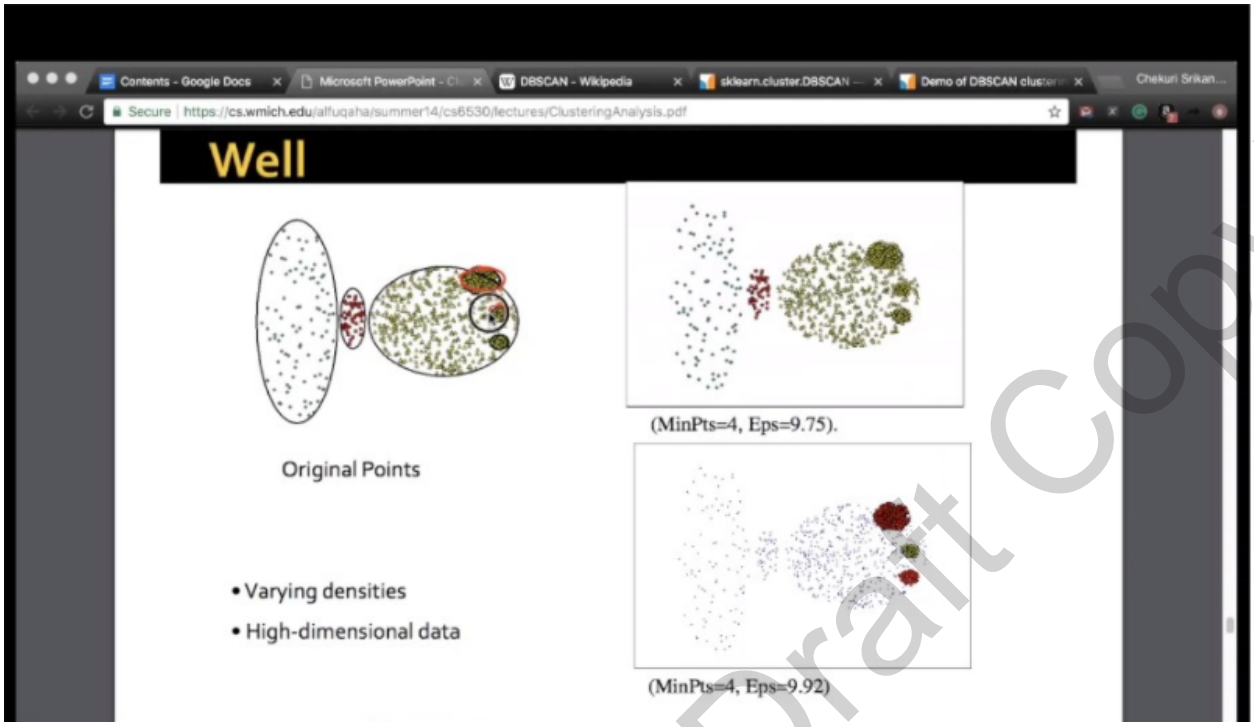
52.7 Advantages and Limitations of DBSCAN



Timestamp : 02:23

1.) It's resistant to noise
2.) Can handle clusters of different shapes and sizes.
3.) It doesn' require one to specify the number of clusters a priori.
4.) It requires only two parameters: MinPts and Epsilon.
5.) It is designed for use with databases as it's created by the database community.

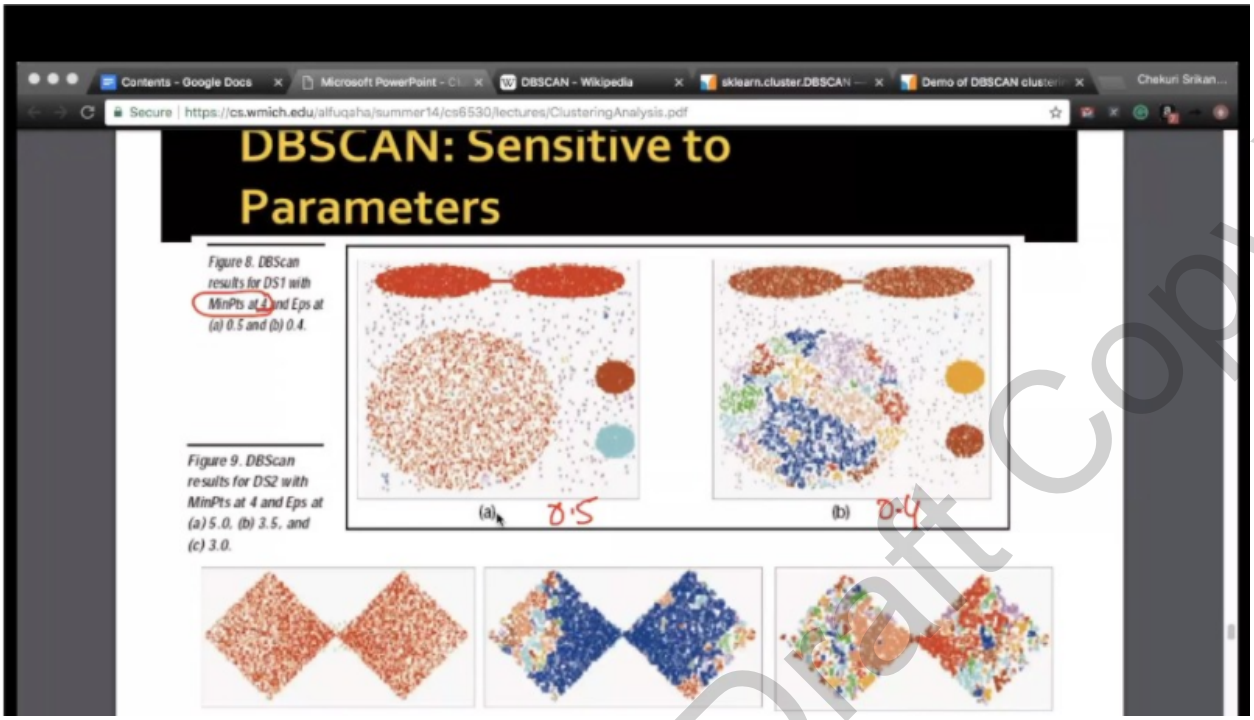
Limitations :



Timestamp : 02:31

1.) Even a small change in the hyperparameters, we can get a completely different type of clusters. So, it's quite sensitive to the choice of hyperparameters.
2.) Varying densities.
3.) High-dimensional data. It's due to the curse of dimensionality.

Let's look at some illustrative examples.



Timestamp : 05:35

As we can see from the above image, since there are varying densities in the dataset, with a slight change in the hyperparameters, we get different results.

4.) It's not entirely deterministic.
5.) If the data and the scale are not well understood ; choosing a meaningful distance threshold Epsilon would be difficult.

52.8 Time and Space complexity

Time Complexity : O(n*log(n)) -> Mostly comes from the fact that we are using a range query. In simple terms, for each point we are going to perform a range query which takes O(log(n)) and there are a total of n points.

Space Complexity : O(n) . We just need to store the data. Our data is often stored in a database. But it still works even if it's not.

52.9 Code Sample

As the video lecture 52.9 is only about the code sample discussion, we are not providing any notes for it. For any queries regarding the code samples, please feel free to post them in the comments section below the video lecture (or) you can mail us at mentors.diploma@appliedroots.com