Hello guys. So we are going to continue our discussion with respect to the clustering algorithms. In our previous video, we discussed about K-means clustering. In this video we are going to discuss about the hierarchical clustering. Now whenever we talk about hierarchical clustering let's understand the goal, uh, what we are trying to do over here. So let's say that if this is my data set, after applying the hierarchical clustering we will be able to get this three groups of data right three clusters of data. But if I really want to compare this with K-means clustering over here, even though my k is equal to three, the number of groups, but here there will be no centroids. Okay. In K-means we specifically had centroids with respect to each and every cluster. Now let's go ahead and understand like what is the basic mathematical intuition behind forming this particular clusters? Okay. So in uh hierarchical clustering you specifically have two types. One is agglomerative clustering and divisive clustering. Okay. So uh, if you are able to just understand one okay, then you will be able to understand the other one itself because it is a reverse of that. Agglomerative basically means combining divisive basically means dividing okay. So just keep this point in mind, because the geometric intuition that I'm actually going to discuss is with respect to agglomerative, okay. And then automatically you will be able to understand about the divisive okay. So now let's consider these are my six data points. And then we'll try to understand what are the basic steps. Uh that is that needs to be taken in order to understand this. Okay. So here I'm just going to write down all the steps. So the first step is that for each data point. For each data point. Initially we will consider initially we will consider okay it as a separate class. Okay. So what do I mean by this is that I have six points over here. So each and every points will be considered as a separate cluster okay. Or separate class or cluster. Here I'll not say class, but at least I'll say it as cluster okay cluster. So let's say p two is my next point. So this will be a separate cluster P three over here will be a separate cluster P four will be another cluster. Over here P five will be another cluster. And p six will be another cluster okay. So let's consider these are my six cluster okay then coming to the second step which is super important. We find the nearest point. And. Create a new cluster. Okay. So let's say which is my nearest point, this two are my nearest point. So I will make this as one cluster okay. Over here okay. So this will be my new cluster over here. And uh the next nearest point if I probably consider will be P1P2 because this is specifically very very near. Then I will go ahead and create this cluster. Right. I will go ahead and create this cluster. Then you know that it is uh, P four is very much near to this cluster of P5P6. So what we will do, we will try to combine this into another cluster like this okay. And then uh over here also you can basically see P three is very much near to this particular cluster. This is my P three okay. So I will go ahead and combine this cluster over here. And finally both this cluster will be near. So we will go ahead and combine something like this okay. So this is the steps that we specifically need to follow. So in the third point here I can definitely write after doing this step is that we have to keep on doing the same process. Doing the same process. Unless and until we get a. Process until we get a. We get a single cluster, right? So these are the three steps that we specifically follow. And this approach is called as agglomerative approach. You know from smallest point we are trying to cover up all the points and create as a single cluster. Divisive approach. Also I will try to discuss it. But before that, let me also introduce to a new diagram which is called as dendrogram because still we need to decide. Okay, Chris, you have you know, you have actually covered up all the points and you have put it inside a single cluster. But still, I do not know that how many number of clusters we need to take. Right. So how do we decide that? And for that we'll be using a new technique which is called as dendrogram. So let me go ahead and again create this particular points. And let me write down all the steps that are going to happen okay. So here the first thing that I'm actually going to do. Um okay so let's say this is my P one. This is my p two. This is my p three. And similarly this is my p four. P5 and P6 write the same points, okay? And on the right hand side I will construct a dendrogram. This dendrogram is basically constructed based on based on all the points. So let's say this is my P1 point. This is my P2 point. This is my P3 point. This is my P4 point. This is my P5 point and this is my P6 points okay. And on the y axis here you will be basically having something called as Euclidean distance okay. And Euclidean distance can be one two. Any values, whatever values or whatever distances that we have in our uh, in our diagram. And that with respect to that, we will be basically using it. Okay. Now what do we do? In the first step we find out the nearest point. So we see that first of all each and every point is a separate cluster right. So here you can see each and every point is there. Now you go and find out the nearest point that is P4 p5 okay. So what you do you combine them as a new cluster. So here I'm basically going to write P4, p5. Let's say the distance between this cluster is this much. Okay. So I have combined this. I have grouped this into a single cluster. Now after that, you know that which is the nearest point after this obviously this P1P2. So I'm going to combine P1P2. Obviously it will be greater than the Euclidean distance with when compared to P4 and p5. I hope everybody is able to understand how do we find out the nearest point. Obviously we can use Euclidean or Manhattan distance. I have already discussed this in the K-means clustering. Right. So now what I'm actually going to do, I'm going to combine p one and p two. And let's say this will be my this will be my. This distance. Right. So I've combined this. Then you'll be able to find out which is the nearest point, whether P six is near to P4 or P5, or whether p3 is near to p1 and p2. Let's consider that p six is near to p4 and p5. So what I'm going to do I'm just going to combine this as a separate cluster. Right. So here is my separate cluster. So this will become another cluster. So in short my P six is basically getting combined over here okay. So here I'm going to draw another diagram right. And finally you will be able to see P3 will also get combined. So let me create another diagram for this. And here you'll be able to see P3 will get created. So when this P3 is getting combined to the cluster P1 and P2. So here you will be able to see you will be able to find something like P1. We'll be able to find something called the p2, p1, p2, which is as a cluster will get combined to P3. And here you'll be able to create this kind of program. Always remember that based on the Euclidean distance, this this height of the uh, building is basically created. I'll just consider it as building. Okay. So finally, when we combine both these things, that basically means you're combining this cluster and this cluster. Let's say this cluster is getting combined over here. Right. Now. Finally, these are my entire distance. Here you can see that, uh, and when we combine it, it becomes a huge cluster like this. Right? So totally, all the points have been covered. And in the right hand side, we have created something called as dendogram. Now is the point that how do we decide what should be the number of clusters? This is a super important thing that we really need to find out. How do we decide that this should be the number of clusters that is, uh, that that we have to take? In this particular case, let's say my k is equal to two with the help of k is equal to two. Obviously I'll be able to get two groups right p1 p2, p3 p p4 p5 p6. Right. But just by seeing this particular diagram, how do I find out whether k is equal to two. This is the most important thing. Now remember before I go and let me explain you how we decided the value as k. The approach that we have used over here is basically bottom to top right. So this particular approach is something called as agglomerative okay. And the difference between agglomerative and divisive is that divisive we go from top to bottom okay. So that basically means first of all we go from here in this, you know that we have combined all the points like P1, p2, p3, p4, p5, p6. Right. In the next one here you will be able to see I have p1, p2, p3 and this I specifically have p4, p5, p6. And then here I have p1 p2. Here I have p3 here. Similarly I have p4 p5 points right. So these are the clusters from the divisive approach. Also what will happen is that we'll combine all the points on the top. And then we go ahead and divide all the points from the bottom from the top to bottom, like how we did it from agglomerative clustering from bottom to top. Okay. Now this was all about dendrogram and how we can construct a dendrogram from this particular points. Now comes the main point. How do we select the k value. See guys k value. Basically selected based on this Euclidean Euclidean distance threshold okay. So we define some kind of threshold. Let's say the threshold. If I consider the threshold to be four then what will happen is that we'll just try to create a horizontal line from here. And I'm saying that the points right that are present in this particular, uh, clusters, you know, should not be having more than Euclidean distance threshold as four. That is what this line basically specifies all the points that are within the clusters and that are, uh, in different, different clusters. The threshold should be made sure that it should be always less than four. Right. So what we are doing over here, we are trying to set up some kind of threshold value. Now this is super, super important. Threshold value helps us to basically identify how many number of clusters will be there, let's say. And we'll also try to find out the relation once we decrease the threshold what will happen. So over here when I probably cut this line right when I cut this line, considering some threshold value here, you'll be seeing that it passes through two points right. When it passes through two points. That basically means my cluster should be two okay. My cluster should be two. Now obviously for this particular problem statement, your cluster needs to be two because you just have two important clusters that you can directly see from your eyes. But what happens if I reduce my if I reduce my threshold, let's say if I create a new line, if I say that my threshold is somewhere around 2.5 and I start creating a straight line over here, now, over here, you know that how many points it is passing, how many vertical lines it will be passing. It will be passing through this, this, this, this. Right. So four points one, two, three, four. Now in this particular case you can see that if it is passing through four points, you will be seeing that I have to use the k value as four. So here the threshold or the Euclidean distance as it is saying it is just like we are keeping a threshold at. What if the distance between the points, if we are decreasing it? That basically means the number of clusters will increase, right? So here when I try to create a horizontal line on this, the k will be four. Now if I keep on decreasing this threshold then let's say I make this threshold come away. At this particular point, if I if I'm just setting up my threshold as point five, then how many points, how many vertical lines it will pass through. Here you can see it will pass through six vertical lines. That basically means again, we have come to our similar points that we had right p1, p2, p3, p4, p5, p6. Right. So it is our duty to select the best, uh, threshold. And you have to always make sure that when the threshold is basically selected, we should get a good number of clusters. But again, uh, selecting the threshold can be a tough challenge or so with the help of dendrogram, there is a simple hack. Okay? You just need to find out the longest vertical line such that none of the horizontal line passes through it. So let's say that which is the longest vertical line. Let's say this is the longest vertical line. But here you can see that most of the horizontal line will pass right. This this will get passed over here. Right. And this will also get passed over here. So I cannot select this as my longest vertical line. What about this. No, it cannot be selected this because here horizontal line is getting passed through it. What about this vertical line? Yes, this vertical line can be selected because none of the horizontal line is passing through it. Right? When I say none of the horizontal line basically means other horizontal line, not the horizontal line that I am creating. Right. So here you can consider in this building if I try to extend this line, this is basically horizontal line which passes through this line. Right. Similarly here you can see if this is the line over here. Also a horizontal line can be created and it passes through all this line. So I cannot select this specific lines as my longest line right. Similarly if I go and select this here also you can see the horizontal line is passing right. So it cannot be selected. It cannot be select. The only longest vertical line is this one. So as soon as I find out the longest vertical line, all I have to do is that create a horizontal line through it and once I create a horizontal line, will just try to see that how many points it is passing through. If it is passing through two points, that basically means I have to select my k value as two. That is the technique we should definitely apply in Dendrograms, which will actually help us to select the k suitable values. Okay, and the same thing you can also do with the help of Python, which I will be showing you in the practical section. But understand the main technique is that select the. Select the. Longest. Vertical line. Does that? No horizontal line passes through it. Horizontal line passes through it. Okay. This is the technique that we should apply. And with the help of this what we are actually doing, we are able to set up our threshold value which is nothing but Euclidean distance right in the y axis. So I hope you have understood this particular video. I hope, uh, you understood about agglomerative and divisive and what is the differences between them. So in the next video, I'll be trying to, uh, explain you the differences between hierarchical and claiming clustering. And then we'll try to note down some of the points which will be important for you all. Uh, so that if anybody asks you in the interview, you will be able to answer it. So I hope I was able to make you understand all these points. Please make sure that you revise this if you're not able to understand.

Now, guys, let's go ahead and understand the differences between K-means versus hierarchical clustering. Now, one important thing I really want to find out. I'm not going to talk about the differences based on the working, but based on this two important parameters. One is scalability and one is flexibility. If I consider with respect to data size, size. If this data set size is huge, we should definitely use K-means clustering. Right? And for small data set the clear winner is hierarchical clustering. Right. So here obviously for the larger data set we have to use K-means. For smaller data set we use hierarchical clustering. Why? Because in hierarchical clustering we create dendrogram. So if we have a huge data points right many number of data points a huge data set, this dendrogram will not will not be able to clearly see, you know. So it will be very, very difficult to make the decision of how many number of clusters we need to have. K-means clustering. If I talk about K-means clustering, this is a super important point. Uh uh, you know, K-means clustering is only applicable for numerical data set, okay, only numerical data, whereas hierarchical clustering is just not only applicable for numerical data for other data also. Why? Because it just try to find out which is the nearest point. Let's say this is the nearest point if this two points are some movies also. Okay. So we can definitely do it with the help of cosine similarity. Right. So it need not only depend on numerical data, but it is also, you know, wherever probably you can just uh, try to find out wherever we can use this cosine similarity. Also for all those things where cosine similarity can be applied uh, for that also hierarchical clustering can be applied. Okay. So hierarchical clustering this is the most major advantage when compared to K-means okay. Now coming to the third point with respect to visualization, uh, in K-means we basically use k centroids. Right now if I'm specifically using centroids in K-means sometime, uh, if I say uh, and the technique that we specifically use is something called as elbow method, right? This elbow method, sometime it becomes difficult to find the number of centroids. Okay. Sometime it becomes difficult because we have to probably see that where there is a sudden decrease in the x value. And then when it is stabilizing, in the case of hierarchical clustering it becomes a little bit easy to find the number of clusters okay. But again if you have a huge data set k means clustering is the clear winner because obviously you cannot go and create dendogram for such a huge data set. But, uh, I would suggest always make sure that this is the most important point. And uh, in interviews also they may ask you k means clustering. You know, it it it is only applicable for numerical data. Uh, when you have variety of data, definitely you will be able to apply hierarchical clustering okay. So just try to find out this see about cosine similarity where cosine similarity is basically used. Let's say I have two movies. One is this action movies and one is this comedy movie. I can definitely find out the cosine similarity between these two movies. So that is what I'm saying. Variety of data, you know, it need not only be numerical data, but in K-means, since we are using Euclidean Manhattan distance, you know, uh, over here you will be able to see that only numerical data will work. Obviously, uh, when I try to show you the distance between this point, I told Euclidean distance. But here you can also use something called as cosine similarity. Okay. So this is a super important point. So yes this was the basic differences between K-means and hierarchical clustering. So I hope, uh, you understood this. And please explore more about this variety of data and definitely try to understand what cosine similarity is. Okay. Yes. I'll see you all in the.