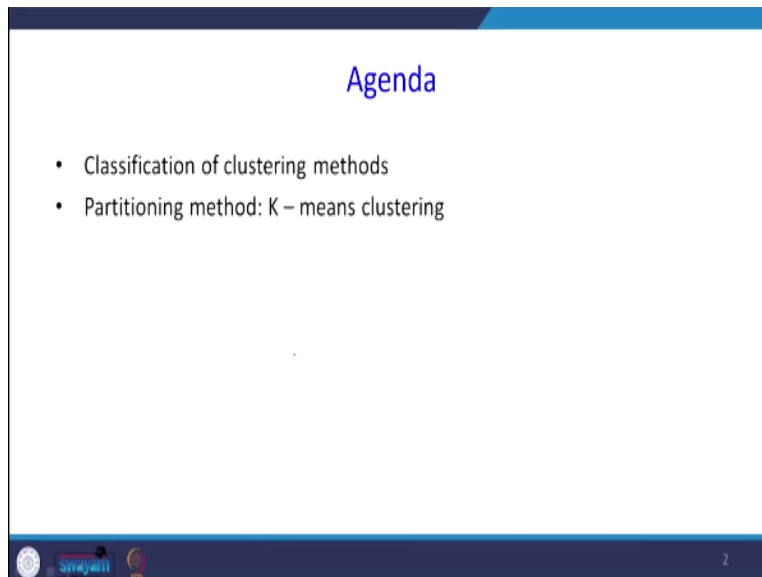


**Data Analytics with Python**  
**Prof. Ramesh Anbanandam**  
**Department of Management Studies**  
**Indian Institute of Technology – Roorkee**

**Lecture – 54**  
**K-Means Clustering**

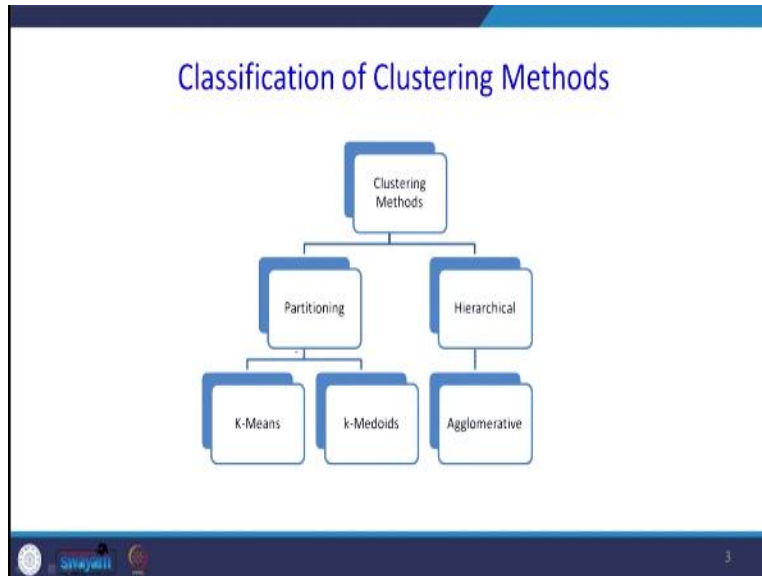
In this lecture, we will talk about K – means Clustering. Before that, I will explain what are the classifications of this clustering method. There are two type of classifications that I will explain in this class. In one classification is a K - means Clustering that I will solve one problem numerically with the help of some example. After solving the problem numerically, I will go to python there I will explain how to use python for doing this K - means clustering.

**(Refer Slide Time: 00:59)**



So the agenda for this lecture is classification of clustering methods under which the partitioning method K – means clustering. That we will see in this class.

**(Refer Slide Time: 01:03)**



So this picture shows the classification of clustering methods. So the clustering methods generally classified into two category, one is partitioning method another one is hierarchical method. In partitioning there is another classifications one is K-Means another one is K-Medoids. In hierarchical there are two methods, one is Agglomerative method another method is Divisive method that will explain when I am explaining this hierarchical method. So in this lecture we are going to discuss about K-Means clustering.

**(Refer Slide Time: 01:39)**

### Which Clustering Algorithm to Choose

- The choice of a clustering algorithm depends on
  - Type of data available
  - Particular purpose
- It is permissible to try several algorithms on the same data, because cluster analysis is mostly used as a descriptive or exploratory tool

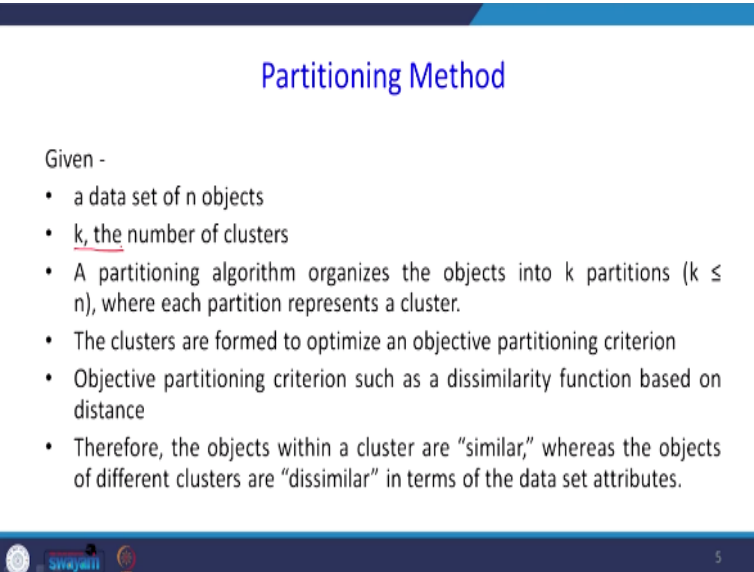
4

So which clustering algorithm to choose? Because previously I was saying two method one is see the partitioning another one is hierarchical. The K-Means algorithm is generally used in advance if you know how many clustering is required. That time you can go for this partitioning

method. If you do not have idea how much cluster you need to do then you can go for hierarchical. So the another point, the choice of clustering algorithm depends upon type of data available and particular purpose.

Particular purpose in the sense whether you want to have in advance how many cluster is required or let us go for all type of classifications later we will give to user to chose the right number of clustering. It is permissible to try several algorithm on the same data, because cluster analysis is mostly used as a descriptive or exploratory tool.

**(Refer Slide Time: 02:37)**



The slide is titled "Partitioning Method" in blue text. Below the title, it says "Given -" followed by a bulleted list of six points. The slide has a blue header and footer. The footer contains logos for "swayam" and "swayam" and the number "5".

Partitioning Method

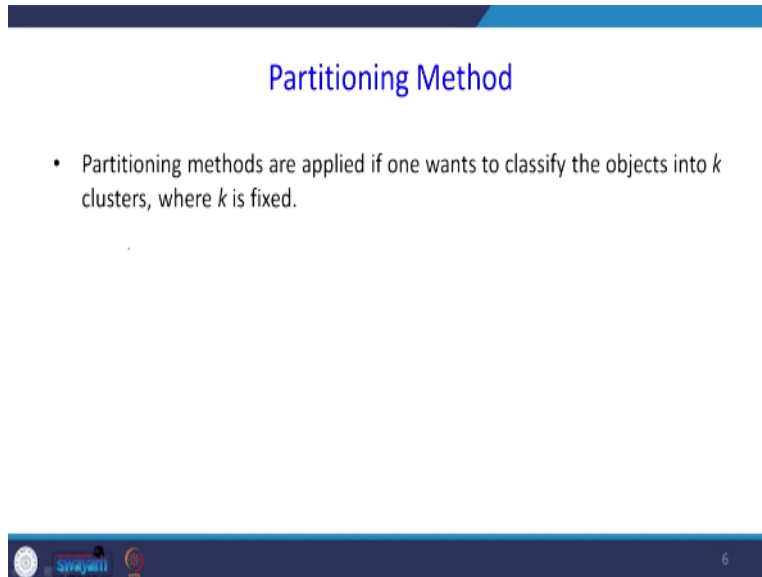
Given -

- a data set of  $n$  objects
- $k$ , the number of clusters
- A partitioning algorithm organizes the objects into  $k$  partitions ( $k \leq n$ ), where each partition represents a cluster.
- The clusters are formed to optimize an objective partitioning criterion
- Objective partitioning criterion such as a dissimilarity function based on distance
- Therefore, the objects within a cluster are "similar," whereas the objects of different clusters are "dissimilar" in terms of the data set attributes.

First we will talk about partitioning method. In partitioning method what are the data which are given is, a data set of  $n$  objects and  $k$ ; this is user-defined,  $k$  is a number of clusters. In advanced we are going to know how many cluster we are going to have. A partitioning algorithm organizes the objects into  $k$  partitions where  $k \leq n$ , where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion.

I will explain what the partitioning criterion is in next slide. The objective partitioning criterion such as dissimilarity function based on distance. So what is happened, within the cluster the dissimilarity should very less between the cluster the dissimilarity should be more. Therefore, the objects within the cluster are similar, whereas the objects of different clusters are dissimilar in terms of dataset attributes.

(Refer Slide Time: 03:40)



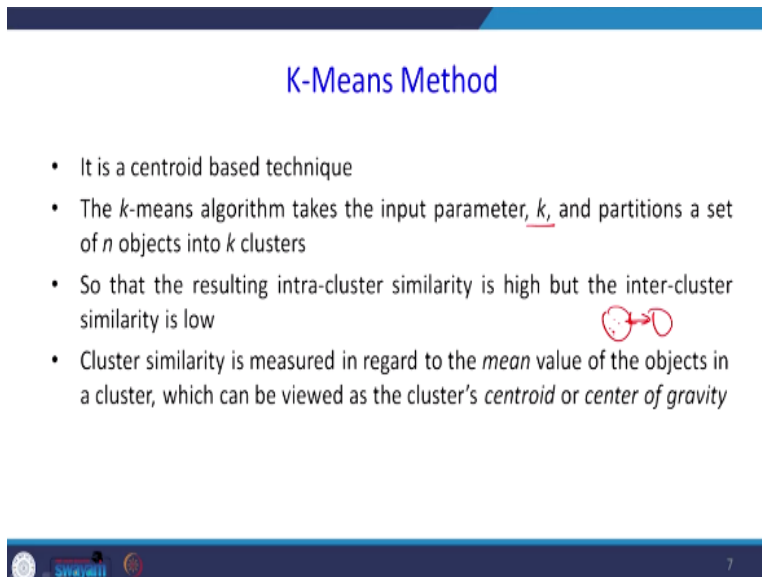
### Partitioning Method

- Partitioning methods are applied if one wants to classify the objects into  $k$  clusters, where  $k$  is fixed.


swayam 6

So partitioning methods are applied if one wants to classify the objects into  $k$  clusters, where  $k$  is fixed.

(Refer Slide Time: 03:49)



### K-Means Method

- It is a centroid based technique
- The  $k$ -means algorithm takes the input parameter,  $k$ , and partitions a set of  $n$  objects into  $k$  clusters
- So that the resulting intra-cluster similarity is high but the inter-cluster similarity is low 
- Cluster similarity is measured in regard to the *mean* value of the objects in a cluster, which can be viewed as the cluster's *centroid* or *center of gravity*

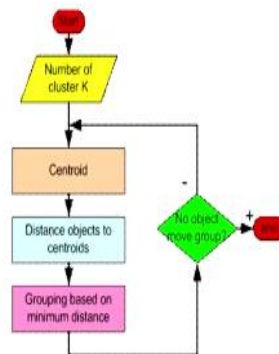
swayam 7

It is a centroid based technique because the; the centroid is nothing your mean, here kind of center of gravity. The  $k$ -means algorithm takes the input parameter,  $k$ , and partitions a set of  $n$  objects into  $k$  clusters, because as I told you here the  $k$  is one of input parameter. So, that the resulting intra-cluster similarity is high but inter-cluster similarity is; so what is happening is suppose there is a cluster 1 and cluster 2 so within that clusters there is a highly homogenous that the inter-cluster similarity is very low.

But between this cluster there should be a low similarity that means the, the dissimilarity is high. So cluster similarity is measured in regard of mean value of the objects in the cluster, which can be viewed as the clusters centroid or center of gravity as I told you.

**(Refer Slide Time: 04:45)**

### Working Principle of K-Means Algorithm



So working principle of K-Means algorithm is; this is flowchart start, in advance you should number of clusters. Then you form the centroids. Randomly you can choose certain point then you form the centroids. Then the distance objects to the centroids. You find out suppose there is object, how far away that object is from the centroid. Then grouping based on the minimum distance. If there are two points, we have to take the point which is closed to that centriod into that cluster. Now that you have to continue for all points, if no object move the group then you can stop it otherwise you continue this cycle.

**(Refer Slide Time: 05:32)**

## Working Principle of K-Means Algorithm

- First it randomly selects  $k$  of the objects, each of which initially represents a cluster mean or center
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean
- It then computes the new mean for each cluster
- This process iterates until the criterion function converges



9

Working principle of K-Means algorithm. First, it randomly selects  $k$  set of objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the objects and the cluster mean. It then computes the new mean for each cluster. Here what I mean, mean is this centroid. This process iterates until the criterion function converges.

(Refer Slide Time: 06:05)

## Working Principle of K-Means Algorithm

- Criterion function

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2$$

where

- $E$  is the sum of the square error for all objects in the data set;
- $p$  is the point in space representing a given object;
- $m_i$  is the mean of cluster  $C_i$  (both  $p$  and  $m_i$  are multidimensional).
- For each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed.
- This criterion tries to make the resulting  $k$  clusters as compact and as separate as possible.



10

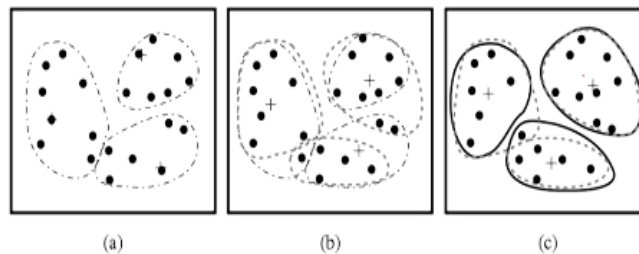
Let us see what is this criterion function. For example,  $E$  is  $\sum_{i=1}^k$  that means odd number clusters.  $P$  for all clusters. So the  $p - m$  modulus squared values. Here what is the  $p$  is the point in space representing the given object,  $n$  is the mean of the cluster. It is nothing but ((  
(06:36) mean absolute deviation but we are squaring that absolute deviation then we are

summing for all clusters. For each objects in each cluster, the distance from the object to its cluster center is squared, and the distance are summed. This criterion tries to make the resulting k clusters as compact as separate as possible.

**(Refer Slide Time: 06:59)**



$K = 3$



For example, this is  $k = 3$ , suppose we; there are  $n$  type of dataset. Randomly, I am making 3 clustering. After finding 3 clustering then I finding centroid of each clusters then from each centroid I am looking at all other objects and their distance then; if any point is closer to that centroid I am bringing that point into that cluster. Then I am updating this cluster and so on and continuing until all the objects are grouped into 3 clusters and the intra-distance is less and inter-distance is more. This I will explain with the help of a numerical example.

**(Refer Slide Time: 07:45)**

## K-Means Clustering Algorithm

Algorithm: k-means. The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

- Input:
  - k: the number of clusters,
  - D: a data set containing n objects.
- Output: A set of k clusters.



Algorithm, k-means. The k-means algorithm for partitioning, where each cluster's center represented by the mean value of the object in the cluster. What are the input for the k-means algorithm? The number of clusters and data set containing n objects. What is going to be output? A set of k clusters.

**(Refer Slide Time: 08:05)**

## K-Means Clustering Algorithm

Algorithm: k-means. The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

- Input:
  - k: the number of clusters,
  - D: a data set containing n objects.
- Output: A set of k clusters.



The K-Means clustering methods, as I; this also I have explained in my; that flowchart. Arbitrarily choose k objects from D from the dataset as the initial cluster centers. Repeat for all the points. Assign each object to the cluster to which the object is most similar. The distance is very small based on the mean value of the object in the cluster. Update the cluster means, i.e., calculate the mean value of the objects for each cluster until no change.



(Refer Slide Time: 08:40)

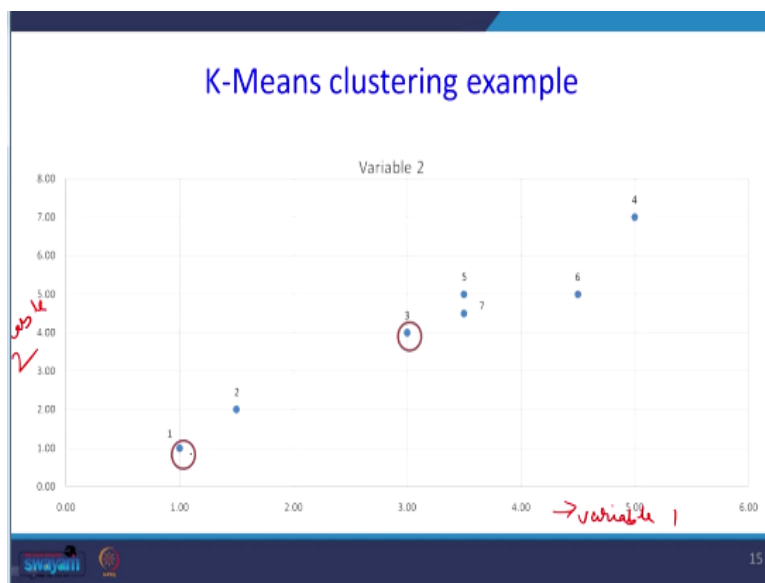
**K-Means clustering example**

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

14

Now taking a numerical example. There are 7 individuals. 1, 2, 3, 4, 5, 6, 7. There are two variables, variable 1 and variable 2. As I told you there is a small difference between clustering and factor analysis. In factor analysis we will group the variable into different category but in the cluster analysis the respondents the individuals have to group. That is the difference. Now there are 7 people is there. We are going to cluster this 7 people into some numbers. Let us see what is that number.

(Refer Slide Time: 09:17)



Suppose here the  $k = 2$  initially here assume that the  $k$  is given  $k = 2$ , I want to make 2 clustering. Suppose randomly I have chosen, in x-axis the variable 1, in y-axis the variable 2. So

the point 1 and 3 are randomly taken, yes it is mentioned variable 2 and variable 1. Point 1 and 3 are taken randomly so there are  $k = 2$ , 2 cluster.

(Refer Slide Time: 10:05)

### K-Means clustering example

- Initialization:** Randomly we choose following two centroids ( $k=2$ ) for two clusters. In this case the 2 centroid are:

Cluster	Var1	Var2
K1	1.0	1.0
K2	3.0	4.0

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.0

- Calculate Euclidean distance using the given equation  

$$\text{Distance} [(x_1, y_1), (x_2, y_2)] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

So the point 1 and 3 say the position of point 1 is 1, 2, the position of point 3 is 3, 4. So what is the initialization? Randomly we choose following two centroids where  $k = 2$  for two clusters. In this case the two centroids are that point itself, K1 1, 1; K2, 3, 4. So calculate the Euclidean distance using this given equations between all the points and between the cluster, so the formula for finding the Euclidean distance is root of  $x_2 - x_1$  whole square +  $y_2 - y_1$  whole square.

(Refer Slide Time: 10:47)

### K-Means clustering example

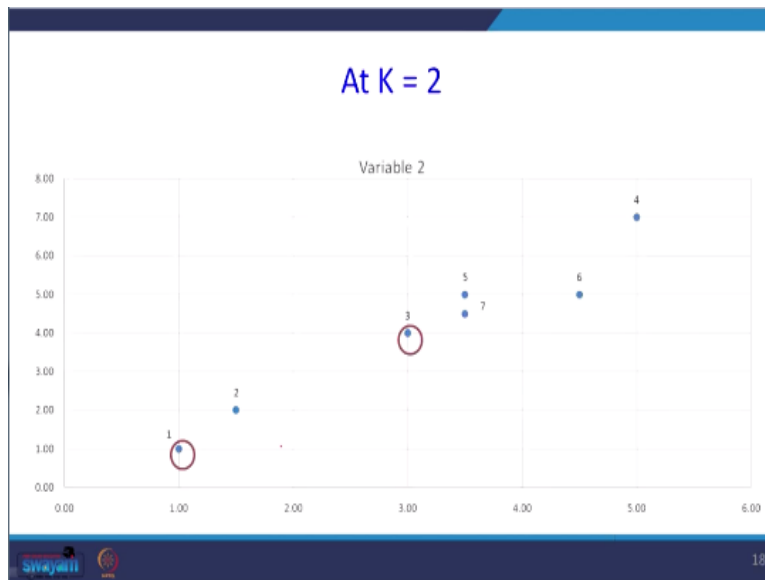
Distance of k1 from k1 (1.0, 1.0) =  $\sqrt{(1.0 - 1.0)^2 + (1.0 - 1.0)^2} = 0$   
 k1 to k2 (1.0, 1.0), (3.0, 4.0) =  $\sqrt{(3.0 - 1.0)^2 + (4.0 - 1.0)^2} = 3.61$   
 Distance of k 2 from k2 (3.0, 4.0) =  $\sqrt{(3.0 - 3.0)^2 + (4.0 - 4.0)^2} = 0$

Cluster	Centroid			Assignment
	K1	K2		
K1	0	3.61		k1
K2	3.61	0		k2

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Since there are 2 point is there, there are two things we are going to do that. The distance between k1 and k1 from that point itself what was the distance from that same point, so the distance is 0. And between the two clusters that is 1 and 3 that is K1 and K2. The position of K1 is 1, 1, the position of K2 is 3, 4. The distance is  $3-1$  whole square +  $4-1$  whole square that is the 3.61. Then the distance of K2 from K2 itself, obviously that will be 0. So what I have taken cluster K1 and K2, the distance between two clusters, so the distance between K1 and K2 is 3.61 the same value is this one. Now we will update this.

**(Refer Slide Time: 11:39)**



So there are two parts, this is Cluster 1 and Cluster 2.

**(Refer Slide Time: 11:45)**

### K-Means clustering example

- Calculate Euclidean distance for next dataset (1.5, 2.0)

$$\text{Distance from cluster1} = \sqrt{(1.5 - 1.0)^2 + (2.0 - 1.0)^2} = 1.12$$

$$\text{Distance from cluster2} = \sqrt{(1.5 - 3.0)^2 + (2.0 - 4.0)^2} = 2.5$$

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(1.5, 2.0)	1.12	2.5	k1

Now what we are going to do, we will take next variable 1.; that is the individual 2, 1.5 in 2. So from Cluster 1 I am going to find out how much faraway this point 2. Similarly, from cluster 2 that is individual 3, what is the distance of 2. So the distance from cluster 1 it is see  $1.5 - 1$  whole square +  $2 - 1$  whole square. So the distance from cluster 2 because 1 and 3 is initial cluster, you have to remember this.

This was our initial cluster. So that is  $1.5 - 3$  whole square +  $2 - 4$  whole square. So the Euclidean distance the data set is 1.5, 2. The distance between cluster 1 and this data set is 1.12. The distance between cluster 2 and the data set is 2.5. So this point is closer to cluster 1 because it is the distance is less 1.1. So what we are going to do we are going to assign this point that is individual into the cluster 1 that is the K1. So this point is assigned to K1.

**(Refer Slide Time: 13:05)**



So what happened, this point is 1.5, 2. So now in this cluster 2 and 1 are assigned into same cluster. After assigning we are going to update the centroid of this cluster that is point 1 and 2.

**(Refer Slide Time: 13:24)**

## K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	$(1.0 + 1.5)/2 = 1.25$	$(1.0 + 2.0)/2 = 1.5$
K2	3.0	4.0



Now we will update the centroid of that cluster K1 because, why we are updating that K1 initially 1 individual now one more individual has entered into that cluster K1 so we are updating for that attributes, so the centroid of K1 is; for variable 1, it is 1 + 1.5 divided by 2 it is 1.25. Then for variable 2 the centroid is in K1 for variable 2 the centroid is 1 + 2 divided by 2 1.5. The K2 remain as it is.

**(Refer Slide Time: 13:57)**

### K-Means clustering example

- Calculate Euclidean distance for next dataset (5.0, 7.0)

Distance from cluster1 =  $\sqrt{(5.0 - 1.25)^2 + (7.0 - 1.5)^2} = 6.66$

Distance from cluster2 =  $\sqrt{(5.0 - 3.0)^2 + (7.0 - 4.0)^2} = 3.61$

Individual	Variable 1	Var2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	4.5
6	4.5	5.5
7	3.5	4.5

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(5.0, 7.0)	6.66	3.61	K-2

Now we will look at individual 4 whose attribute are 5, 7. From cluster 1 let us find out how much distance it is. Similarly, from cluster 3 we will find out how much distance it is. But in cluster 1 already there are two point has come. So that the our centroid has been already updated. So the distance from cluster 1 is 5, see that this, this 1.25 you got from this value, it is a centroid

so this value centroid value for variable 1;  $5 - 1.25$  whole square similarly  $7 -$  this was the new centroid of cluster 1, you see that this is 1.5. So that square is 6.66.

From cluster 2 the distance is  $5 - 3$  whole square +  $7 - 4$  whole square, the distance is 3.61. So this value we brought in the table format. So that data set is 5, 7 from cluster 1 the distance is 6.66, from cluster 2 the distance is 3.61. So this point is very close to cluster 2, so we are going to assign this point to the cluster 2 so 5, 7.

**(Refer Slide Time: 15:16)**



Now what happened this point is very close to this one. So we assigned this into this cluster.

**(Refer Slide Time: 15:25)**

### K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	$(3.0 + 5.0)/2 = 4$	$(4.0 + 7.0)/2 = 5.5$

So after assigning as I told you, we have to update the centroid of cluster 2 now, because cluster 2 initially we had only one point, now one more point is entered. So the new centroid is 3 + 5 divided by 2 that is 4 for variable 1, for variable 2 the centroid is 4 + 7 divided by 2 5.5. Now this is the 4 and 5.5 is the new centroid for K2.

**(Refer Slide Time: 15:54)**

**K-Means clustering example**

Individual	Variable 1	Var
1	1.0	
2	1.5	
3	3.0	
4	5.0	
5	3.5	
6	4.5	
7	3.5	

- Calculate Euclidean distance for next dataset (3.5, 5.0)

Distance from cluster1 =  $\sqrt{(3.5 - 1.25)^2 + (5.0 - 1.5)^2} = 4.16$

Distance from cluster2 =  $\sqrt{(3.5 - 4.0)^2 + (5.0 - 5.5)^2} = 0.71$

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(3.5, 5.0)	4.16	0.71	K-2

25

Now we will another variable 5, that is 3.5. We will find out how far away this point or this individual from cluster 1 and cluster 2. First we will find out from cluster 1. For a cluster 1 it is a 3.5 – the centroid of cluster 1 whole square + 5 – centroid of that is 1.5 whole square that is a 4.16. Now this point 3.5 – 4, 4 is centroid of our cluster 2, see this one 3.5 – 4 whole square + 5 – 5.5 this 5.5 you got from this updated centroid of cluster 2, so 0.71.

So let us bring this value into the table format. So the distance between this point and the cluster 1 is 4.16 and the distance between 3.5, 5 this point 2 cluster 2 is 0.71. So this point is this dataset is closer to the cluster 2, so we will assign this point also to cluster 2. So after assigning what has happened, so this point is assigned to cluster 2.

**(Refer Slide Time: 17:07)**

## K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	$(3.0+5.0+3.5)/3 = 3.83$	$(4.0+7.0+5.0)/3 = 5.33$

Now we will go to the next variable. After assigning before going to the next variable we will update the centroid of cluster 2, because in cluster 2 now there are three points is there, that is 3, 4 5,7 3.5, 5. First in variable 1 we will find out centroid nothing but the average 3.83 for, in K2 the centroid of variable 2 is 5.33. Now this is the new centroid for our K2.

**(Refer Slide Time: 17:40)**

## K-Means clustering example

- Calculate Euclidean distance for next dataset (4.5, 5.0)

$$\text{Distance from cluster1} = \sqrt{(4.5 - 1.25)^2 + (5.0 - 1.5)^2} = 4.78$$

$$\text{Distance from cluster2} = \sqrt{(4.5 - 3.83)^2 + (5.0 - 5.33)^2} = 0.75$$

Individual	Variable 1	Var 2
1	1.0	
2	3.5	
3	1.0	
4	5.0	
5	3.5	
6	4.5	
7	3.5	

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(4.5, 5.0)	4.78	<u>0.75</u>	K- 2

Now we will take new individual and whose data point is 4.5 and 5. Now let us see how much distance or how much away from cluster 2. From cluster 1 say 4.5 – centroid of cluster 1 is 1.25 whole square + 5 – 1.5 whole square 4.78, that is a 4.78. Now from cluster 2 let us see how much distance. 4.5 – 3.83, how we got 3.83, because we are updated the centroid of cluster 2, so 3.83 whole square + 5 – 5.33. How we got 5.33? This value, 5.33 whole squares so distance is this



one. By looking at the table this dataset is closer to the cluster 2 so we will assign this dataset also to K2.

**(Refer Slide Time: 18:35)**



So after assigning, what is happening, so this point is assigning to the cluster 2. Again we will update.

**(Refer Slide Time: 18:41)**

### K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	$(3.0+5.0+3.5+4.5)/4= 4.00$	$(4.0+7.0+5.0+5.0)/4= 5.25$

Now in the cluster 2, there are 4 dataset, that is 3, 4, 5, 7. Now we will find the centroids. There are four dataset add it divided 4 that is 4, here 4.0+7.0+5.0 divided 4 that is 5.25, this is our centroid of K2, updated centroid.

**(Refer Slide Time: 19:11)**

### K-Means clustering example

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	3.5
4	5.0	7.0
5	3.5	5.0
6	4.5	4.5
7	3.5	4.5

- Calculate Euclidean distance for next dataset (3.5, 4.5)

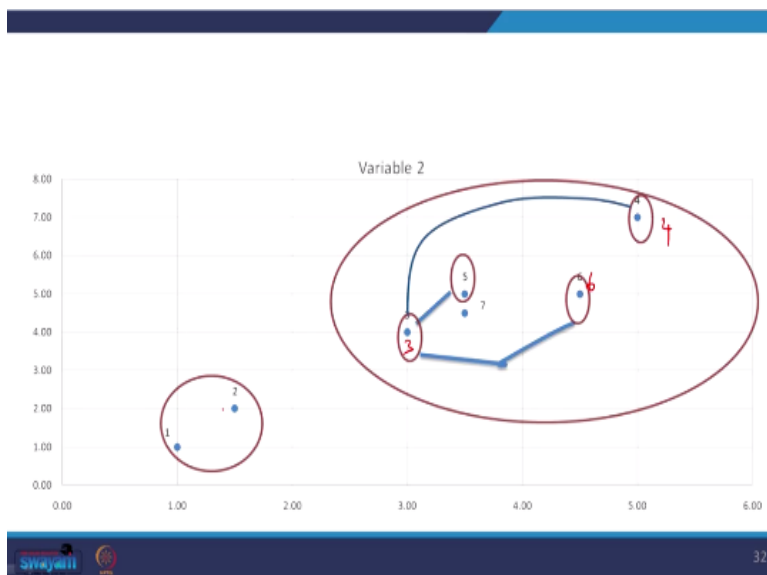
Distance from cluster1 =  $\sqrt{(3.5 - 1.25)^2 + (4.5 - 1.5)^2} = 3.75$

Distance from cluster2 =  $\sqrt{(3.5 - 4.00)^2 + (4.5 - 5.25)^2} = 0.86$

Dataset	Euclidean Distance		
	Cluster 1	Cluster 2	Assignment
(3.5, 4.5)	3.75	0.86	K-2

Now we will take the last point, it is the 3.5 and the 4.5. Let us see how much distance. This point is from cluster 1, cluster 2. From cluster 1,  $3.5 - 1.25$  whole square +  $4.5 - 1.5$  whole square that is 3.75. From cluster 2, how we got this 4 from this value  $3.5 - 4$  whole square +  $4.5 - 5.25$ , so this value is  $-5.25$  whole square is 0.86. So that value is 0.86. Again, so this point is closer to the cluster 2 so we will assign this point into cluster 2.

**(Refer Slide Time: 19:50)**



So now we have assigned, so this is one cluster. What are the point in this cluster? This is 3, 5, 7, 6, 4. In another cluster it is 1 and 2.

**(Refer Slide Time: 20:06)**

### K-Means clustering example

- Update the cluster centroid

Cluster	Var1	Var2
K1	1.25	1.5
K2	$(3.0+5.0+3.5+4.5+3.5)/5= 3.9$	$(4.0+7.0+5.0+5.0+4.5)/5= 5.1$

33

After that again we will find out centroid of that cluster 2. So 1, 2, 3, 4, 5 dataset so add all the value divided by 5 it is 3.9. Again you add all the value divided by 5 it is 5.51. Now there are two clusters. The centroid of cluster 1 is 1.25, 1.5. The centroid of cluster 2 is 3.9 and 5.1. This value will verify when I am showing python demo.

**(Refer Slide Time: 20:38)**

### K-Means clustering example

Individual	Variable 1	Variable 2	Assignment
1	1.0	1.0	1
2	1.5	2.0	1
3	3.0	4.0	2
4	5.0	7.0	2
5	3.5	5.0	2
6	4.5	5.0	2
7	3.5	4.5	2

34

Now this is the summary of our result. What has happened? So these individual is one group, cluster 1, this people in cluster 2. So what is the property is, the people in this cluster are more similar. People in this cluster also more similar. But between these two clusters the distance is far away. Now I have solved this problem with manually. Now we will go to python environment. So the same problem I will explain.

(Refer Slide Time: 21:23)

### Python code for K- Means Clustering

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

In [2]: data = pd.read_excel('clustering_ex.xlsx')

In [3]: data
```

	Variable_1	Variable_2
0	1.0	1.0
1	1.5	2.0
2	3.0	4.0
3	5.0	7.0
4	3.5	5.0
5	4.5	5.0
6	3.5	4.5

35

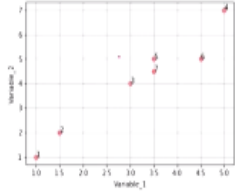
So I have brought this screenshot of our, for the K-Means clustering. We imported this required library import pandas as pd, import numpy as np, import matplotlib.pyplot as plt, so data is this one. So this was our data.

(Refer Slide Time: 21:41)

### Python code for K- Means Clustering

```
n [4]: fig = plt.figure(figsize = (5, 5))
x = data['Variable_1']
y = data['Variable_2']
n = range(1,8)
fig, ax = plt.subplots()
ax.scatter(x, y, marker='o', c='red', alpha=0.5)
plt.grid()
plt.xlabel("Variable_1")
plt.ylabel("Variable_2")
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))

<matplotlib.figure.Figure at 0x20d7a5044a8>
```



36

First we have plotted the scattered plot, the scattered plot with label.

(Refer Slide Time: 21:48)

### Python code

```

In [5]: from sklearn.cluster import KMeans
        kmeans = KMeans(n_clusters=2)
        kmeans.fit(data)

Out[5]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
              n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
              random_state=None, tol=0.0001, verbose=0)

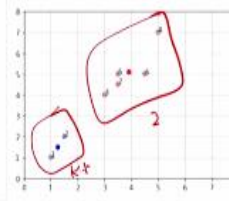
In [6]: labels = kmeans.predict(data)
        centroids = kmeans.cluster_centers_

In [8]: centroids
Out[8]: array([[3.0, 5.1],
              [1.25, 1.5]])

In [9]: fig = plt.figure(figsize=(5, 5))
        colors = ['r', 'b']
        colors = np.array(colors[labels], labels)
        colors = list(colors)
        fig, ax = plt.subplots()
        ax.scatter(x, y, color = colors, alpha = 0.5, edgecolor = 'k')
        for idx, centroid in enumerate(centroids):
            plt.scatter(centroid, color = colors[idx])

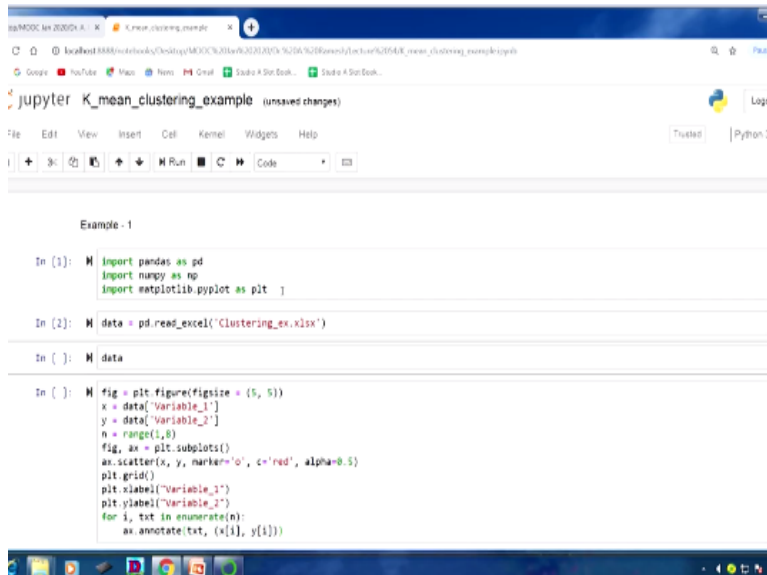
        for i, txt in enumerate(n):
            ax.annotate(txt, (x[i], y[i]))
            plt.grid()
            plt.xlabel('Variable_1')
            plt.ylabel('Variable_2')
            plt.show()

```



Now this was the final result of cluster analysis. What happened? See this is one group, this is another group. This blue represents the centroid of this cluster 1, this red represents centroid of cluster 2. Now we will go to python environment. I will tell you how to do this K-means clustering in python.

**(Refer Slide Time: 22:12)**



```

Example - 1

In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

In [2]: data = pd.read_excel("Clustering_ex.xlsx")

In [ ]: data

In [ ]: fig = plt.figure(figsize=(5, 5))
        x = data['Variable_1']
        y = data['Variable_2']
        n = range(1,8)
        fig, ax = plt.subplots()
        ax.scatter(x, y, marker='o', c='red', alpha=0.5)
        plt.grid()
        plt.xlabel("Variable_1")
        plt.ylabel("Variable_2")
        for i, txt in enumerate(n):
            ax.annotate(txt, (x[i], y[i]))

```

Now I am going to explain how to use python for doing k-means algorithm. I have taken two examples; one example is what I have explained in my presentation. First we will import necessary libraries pandas, numpy, matplotlib.pyplot and so on. Next we will import the data.

**(Refer Slide Time: 22:35)**

```

In [ ]:
2  30  40
3  50  70
4  35  60
5  45  50
6  35  45

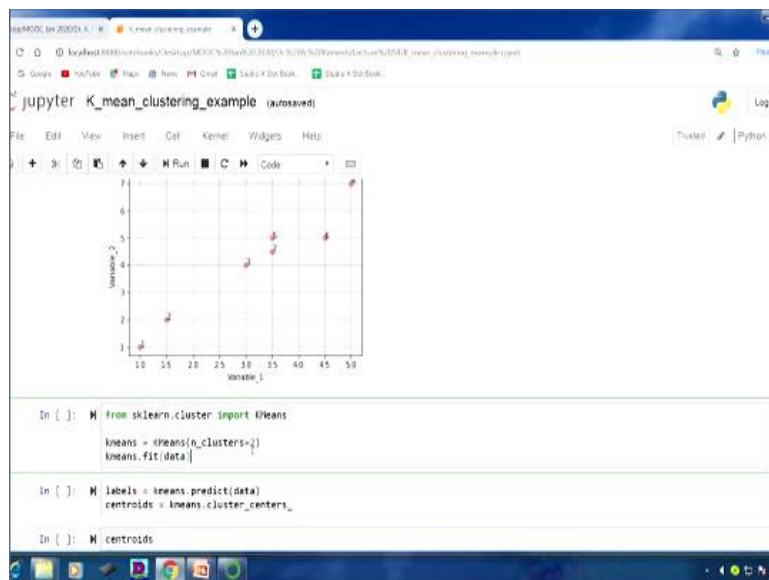
In [ ]:
fig = plt.figure(figsize = (5, 5))
x = data['variable_1']
y = data['variable_2']
n = range(1,8)
fig, ax = plt.subplots()
ax.scatter(x, y, marker='o', c='red', alpha=0.5)
plt.grid()
plt.xlabel('variable_1')
plt.ylabel('variable_2')
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))

In [ ]:
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
kmeans.fit(data)

```

The data, when you look at the data there are 7 individual is there, variable 1 and variable 2. After that we will position these individuals into, in a two-dimensional graph.

**(Refer Slide Time: 22:50)**



So what is happening, now we are able to see that all individuals, there are 7 individuals and their position, for example, the position of individual 1 is 1, 1, for position of individual 2 is 1.5, 2 and so on. For running k-means clustering algorithm we have to import this library. From sklearn.cluster import KMeans, so kmeans = KMeans n\_clusters = 2. So this is k = 2. If you want to have 3 clusters that is our next example, you have to substitute in sub 2 = 3, we will run this.

**(Refer Slide Time: 23:27)**

```

Out[5]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
              n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
              random_state=None, tol=0.0001, verbose=0)

In [6]: M labels = kmeans.predict(data)
        centroids = kmeans.cluster_centers_

In [7]: M centroids
Out[7]: array([[3.9, 5.1],
              [1.25, 1.5]])

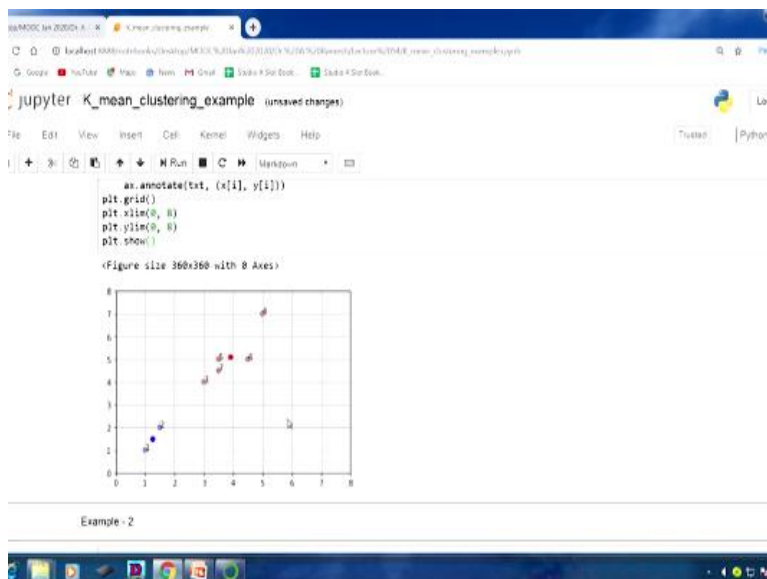
In [ ]: M fig = plt.figure(figsize = (5, 5))
        colormap = ('r', 'b')
        colors = map(lambda x: colormap[x], labels)
        colors = list(colors)
        fig, ax = plt.subplots()
        ax.scatter(x, y, color = colors, alpha = 0.5, edgecolor = 'k')
        for idx, centroid in enumerate(centroids):
            plt.scatter(centroid, color = colormap[idx+1])

        for i, txt in enumerate(x):
            ax.annotate(txt, (x[i], y[i]))
        plt.grid()
        plt.xlim(0, 8)
        plt.ylim(0, 8)

```

After running, we will verify, now the two clusters has been formed. Now we will verify the centroid of the two clusters. So the centroid of the two clusters is 3.9, 5.1 that was my cluster 2 centroid. For cluster 1 the centroid is 1.25, 1.5.

**(Refer Slide Time: 23:50)**



Let us see in picture form. This is the final output. This blue says, this is a centroid of cluster 1. Here the red one says that this is centroid of cluster 2. So what happening now two clusters are formed, in cluster 1 individual 1 and 2 is there; in cluster 2 individual 3, 5, 7, 6, 4 is there. This was exactly the result which I have done in the presentation.

**(Refer Slide Time: 24:21)**

```

In [9]: data1 = pd.read_excel('datapoints.xlsx')
data1
Out[9]:
   x  y
0  2 10
1  2  5
2  8  4
3  5  8
4  7  5
5  6  4
6  1  2
7  4  9

In [ ]: fig = plt.figure(figsize = (5, 5))
X = data1['x']
Y = data1['y']

n = range(1,9)
fig, ax = plt.subplots()
ax.scatter(X, Y, color = 'red')

```

I will take another example where instead of  $k = 2$  will go for 3 clusters, this is a different data set. So in that there are 8 individual is there. There are x variable and y variable.

**(Refer Slide Time: 24:37)**

```

3  8  4
6  1  2
7  4  9

In [ ]: fig = plt.figure(figsize = (5, 5))
X = data1['x']
Y = data1['y']

n = range(1,9)
fig, ax = plt.subplots()
ax.scatter(X, Y, color = 'red')
plt.grid()
plt.xlabel("x")
plt.ylabel("y")
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))

In [ ]: kmeans = KMeans(n_clusters=3)
kmeans.fit(data1)

In [ ]: labels = kmeans.predict(data1)
centroids = kmeans.cluster_centers_

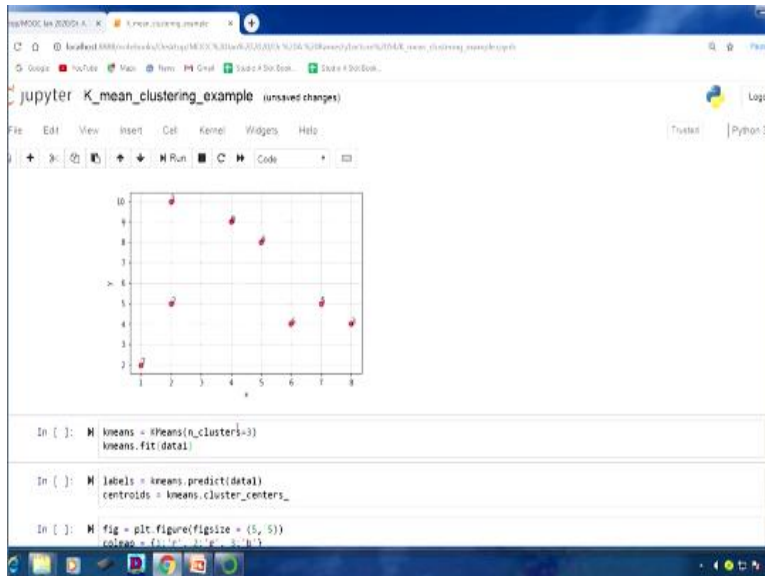
In [ ]: fig = plt.figure(figsize = (5, 5))

```

The same way will plot into the two-dimensional plot, this was that way.

**(Refer Slide Time: 24:41)**





So there are 8 individual and their position.

**(Refer Slide Time: 24:47)**

```

In [11]: kmeans = KMeans(n_clusters=3)
         kmeans.fit(data1)

In [12]: M labels = kmeans.predict(data1)
         centroids = kmeans.cluster_centers_

In [13]: M fig = plt.figure(figsize = (5, 5))
         colormap = 'g_r_b'

Out[13]: array([[7.         , 4.33333333],
                [3.66666667, 9.         ],
                [1.5        , 3.5        ]])

In [14]: M fig = plt.figure(figsize = (5, 5))
         colormap = ('r', 'g', 'b')
         colors = map(lambda X: colormap[X], labels)
         colors1 = list(colors)
         fig, ax = plt.subplots()
         ax.scatter(X, Y, color = colors1, alpha = 0.5, edgecolor = 'k')
         for idx, centroid in enumerate(centroids):
             plt.scatter(centroid, color = colormap[idx+1])

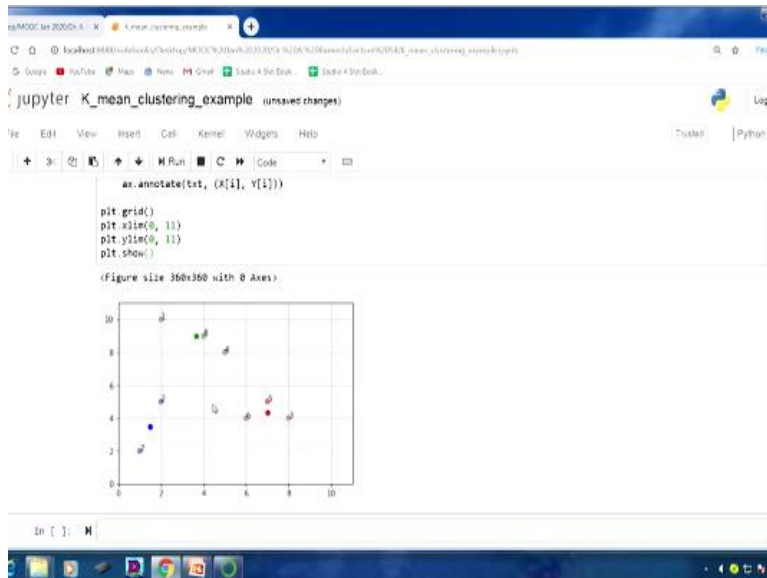
         for i, txt in enumerate(n):
             ax.annotate(txt, (X[i], Y[i]))

         plt.grid()

```

We will go for k-means algorithm. You see that  $k = 3$ , probably we are going to have 3 clusters. So when you run that after running we can find out the centroid, so what I am going to do I am going to enter b then I go to show what is the value of centroid of these three clusters, so paste it, now you run it. So now there are three clusters. The centroid of this cluster is 7, 4.3 3.6, 9 1.5 and comma 3.5.

**(Refer Slide Time: 25:23)**



Now I showed the picture from the final output, now just show the final output. There are three clusters which are in different color. So this blue says the centroid of cluster 1, this red says the centroid of cluster 2, this green says centroid of cluster 3. In this lecture, I have explained the classification of clustering methods. We know that there are two types of classification one is partitioning method another one is hierarchical method.

In the partitioning method there are another two classification one is a K-means clustering algorithm another one is K-Medoids. In hierarchical also there are two classification one is agglomerative and divisive method. But in this lecture I have covered only k-means algorithm. I have taken one numerical problems with the help of that numerical problems I have explained step-by-step procedure how to go for k-means algorithm.

After that I have explained the same problem in python, how to make k-means algorithm where  $k = 2$ . Apart from that, I have taken one more example in python environment then there I have explained how to make three clusters by taking the value of  $k = 3$ . The lecture I will explain the agglomerative method of clustering with the help of an example. Thank you.