K-means

Clustering

edureka!

Agenda of Today's Session

- What is Clustering?
- Types of Clustering
- What is K- Means Clustering?
- How does a K-Means Algorithm works?
- K-Means with Python


What is Clustering?

edureka!

What is Clustering?

"Clustering is the process of dividing the datase

into groups, consisting of similar data-points"

- Points in the same group are as similar as possible
- Points in different group are as dissimilar as possible

edureka!

What is Clustering?

Group of dine
in a restauran

Items arranged in
a mall

edureka!

Where is it Used?

Recommendation System

Recommended Movies

Fickr's Photos

edureka!

How business use Clustering?

Retail Store

Banking

Insurance Companies

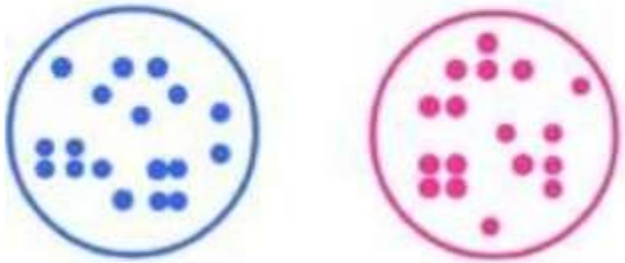
edureka!

Types of Clustering

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering

Exclusive Clustering

- Hard Clustering
- Data Point / Item belongs exclusively to one cluster
- For Example: K-Means Clustering



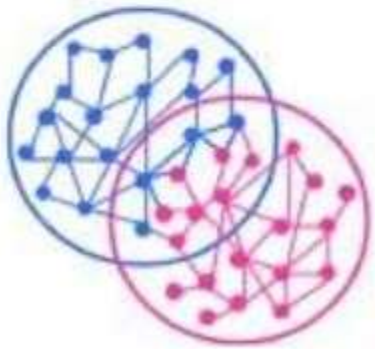
edureka!

Types of Clustering

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering

Overlapping Clustering

- Soft Cluster
- Data Point/ Item belongs to multiple cluster
- For Example: Fuzzy/ C-Means Clustering

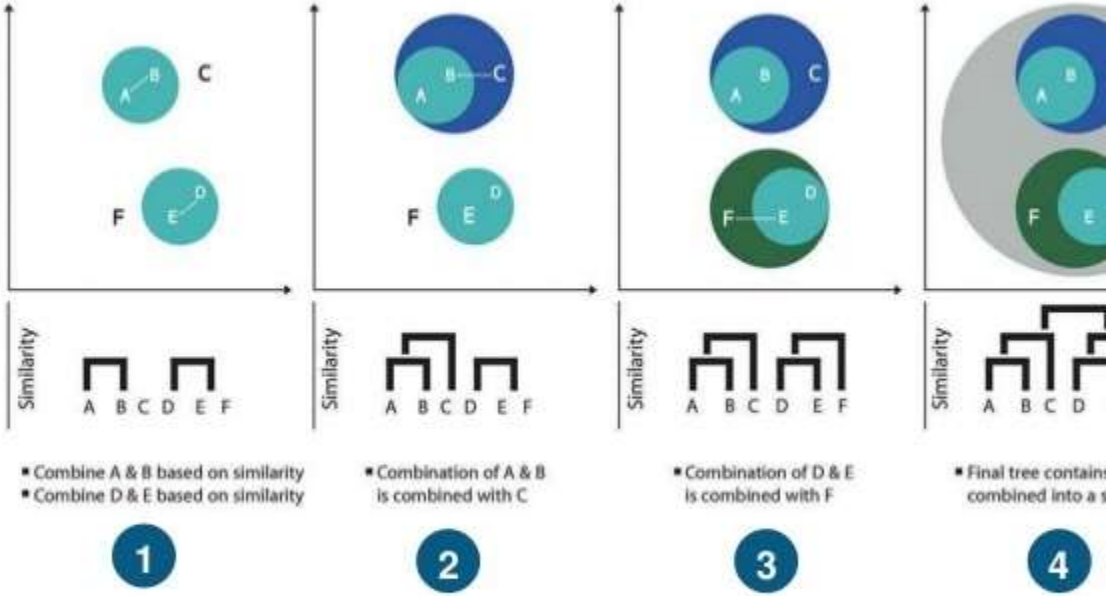


edureka!

Types of Clustering

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering

Hierarchical Clustering



edureka!

What is K-Means Clustering?

"K-Means is a clustering algorithm whose mail go
to group similar elements or data points into
cluster."

NOTE: 'K' in K-Means represent the number of clusters

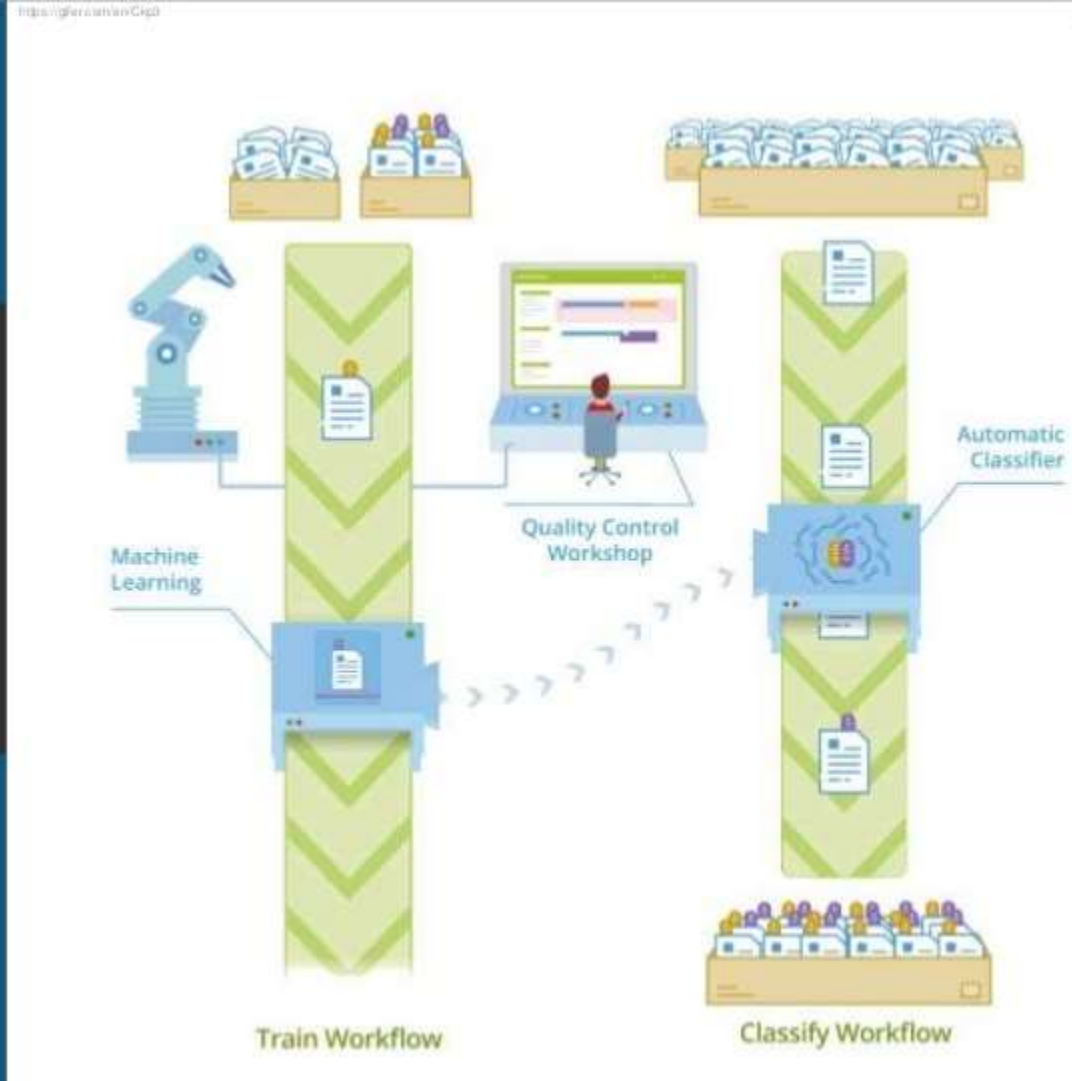
edureka!

What is K-Means Clustering?

Pile of dirty clothes

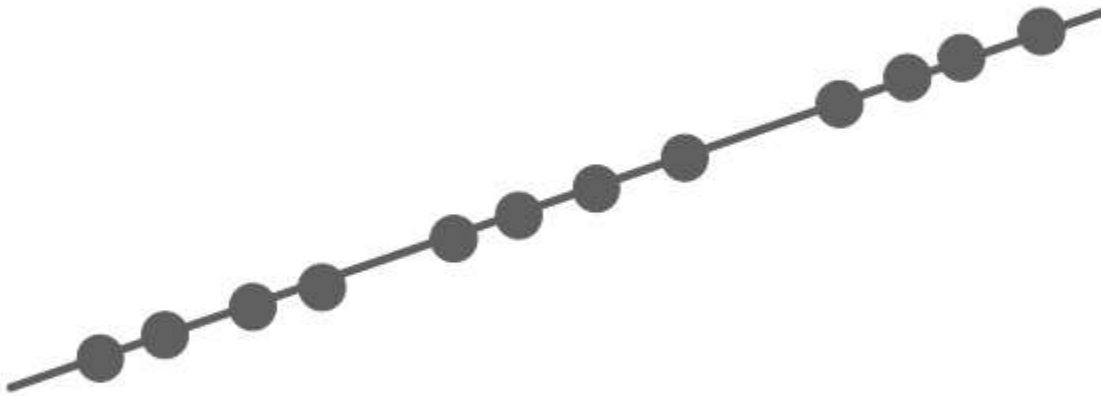
edureka!

Where Can I apply K-Means?



edureka!

K-Means Algorithm

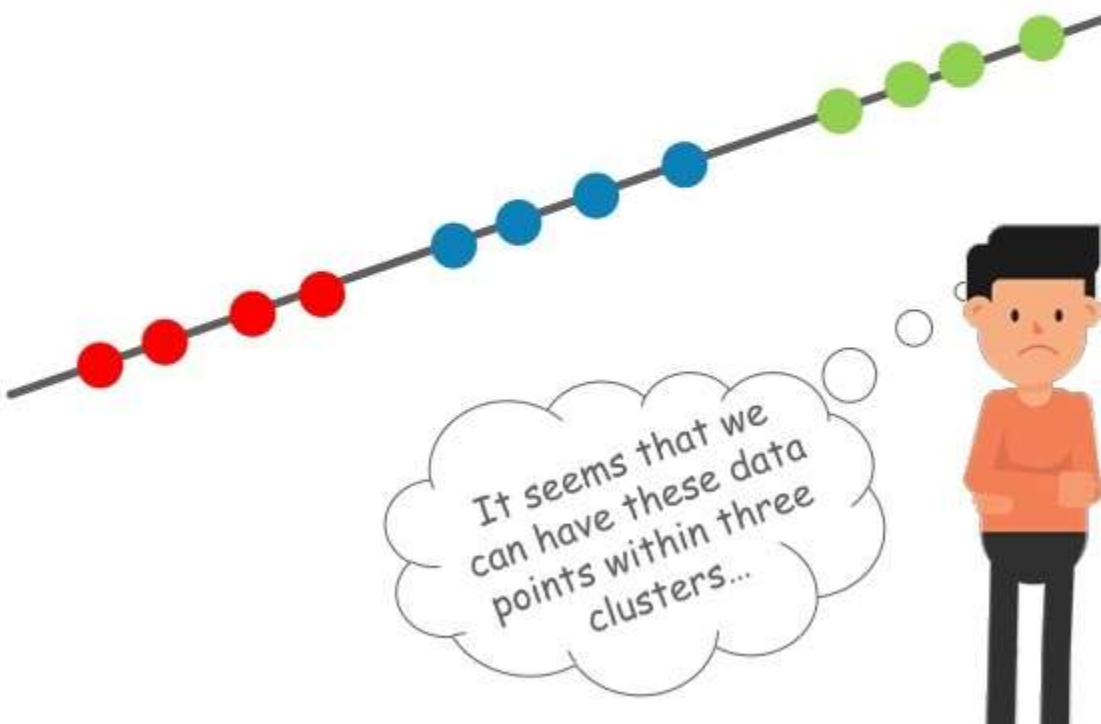


Number of Clusters = 3

edureka!

K-Means Algorithm

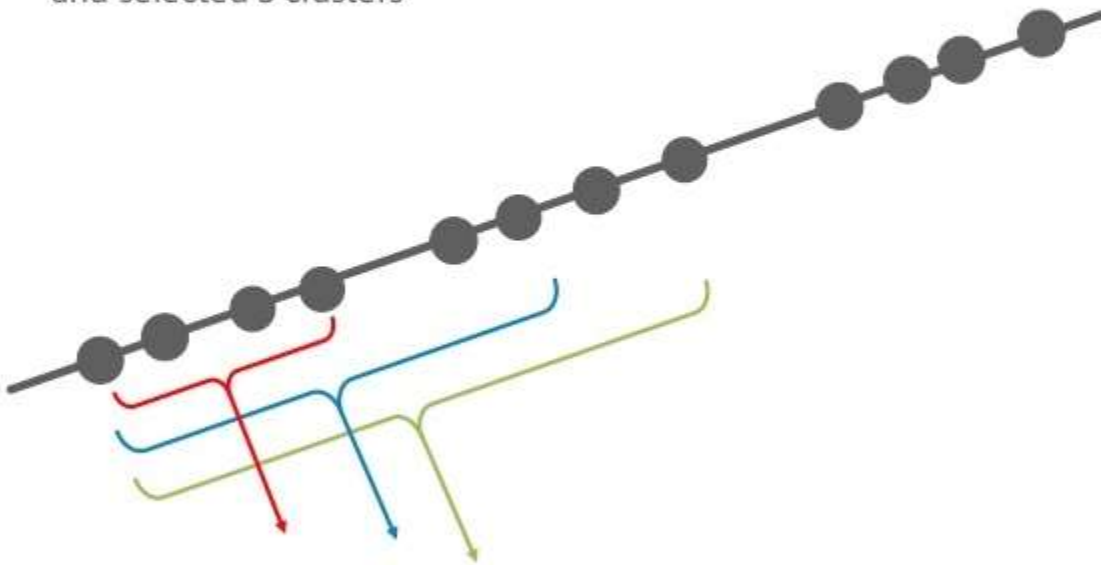
Number of Clusters, K = 3



edureka!

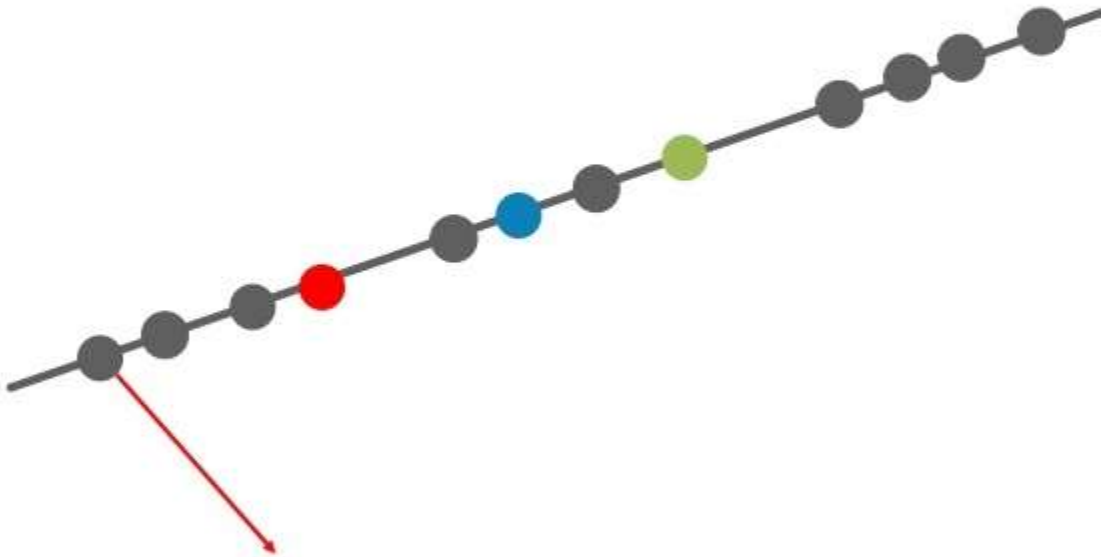
K-Means Algorithm

- **Step 1:** Select the number of clusters to be identified, i.e select a value for K =3 in this case
- **Step 2:** Randomly select 3 distinct data point
- **Step 3:** Measure the distance between the 1st point and selected 3 clusters



edureka!

K-Means Algorithm



Step 4: Assign the 1st point to nearest cluster (red in this case).

edureka!

K-Means Algorithm

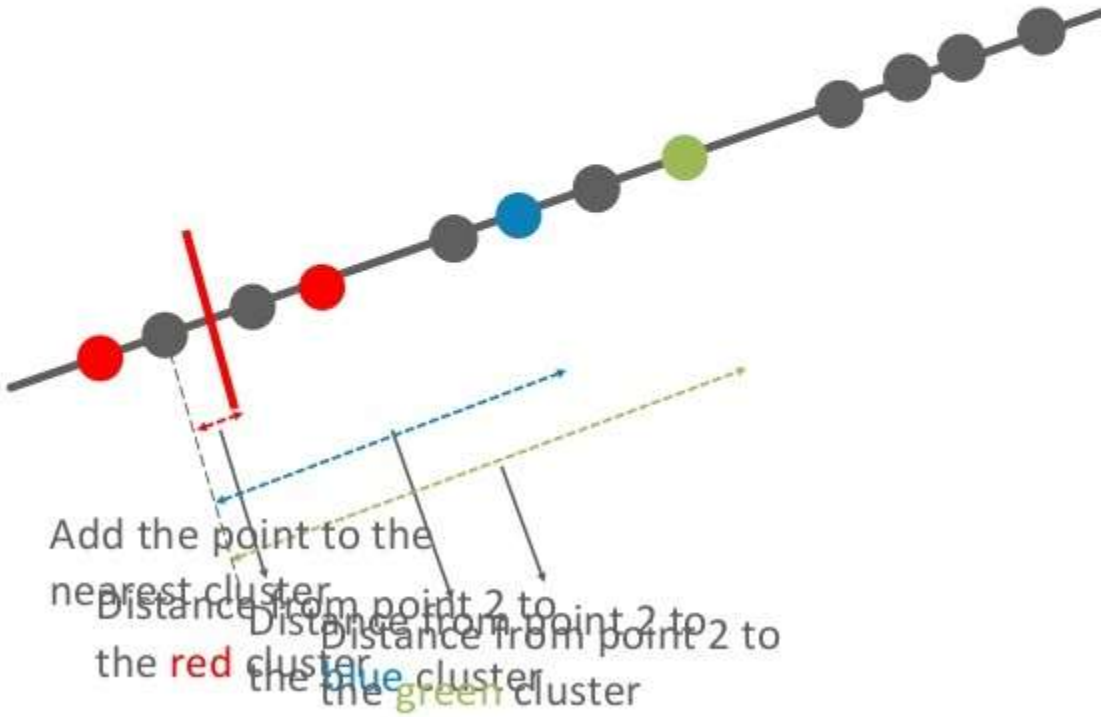
Step 5: Calculate the mean value including the new point for the red cluster

edureka!

K-Means Algorithm

Find to which cluster does point 2 belongs to, how?

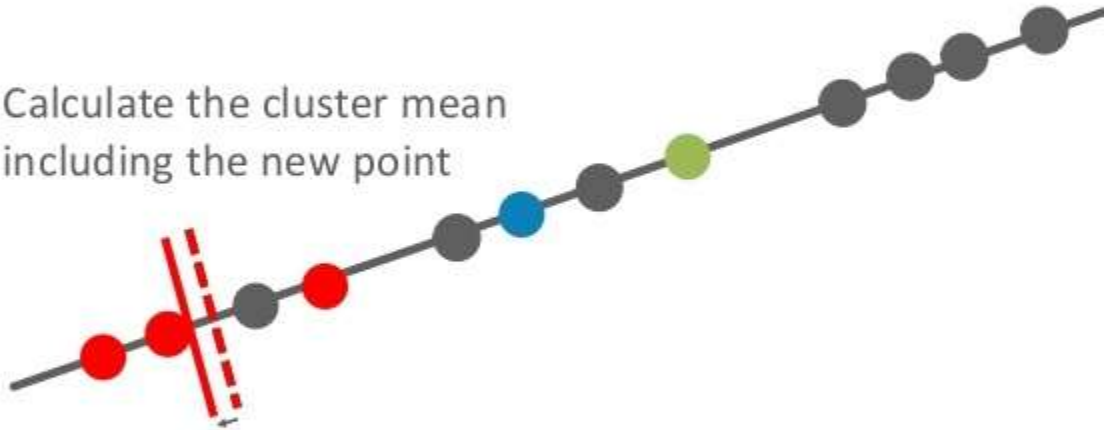
- Repeat the same procedure but measure the distance to the red mean



edureka!

K-Means Algorithm

Calculate the cluster mean including the new point



edureka!

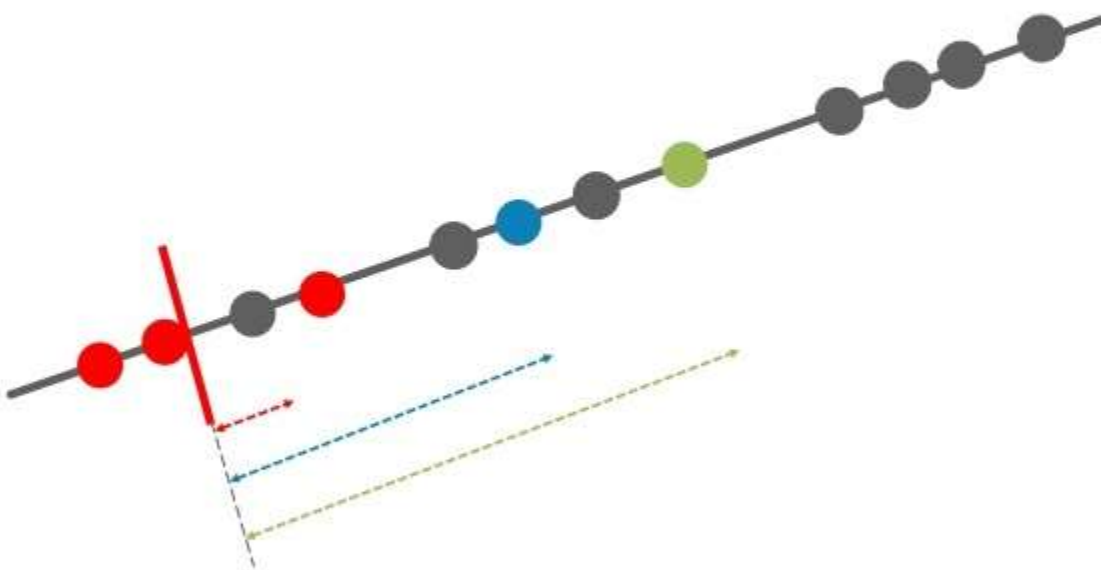
K-Means Algorithm

Find to which cluster does point 3 belongs to, how?

- Repeat the same procedure but measure the distance to the red mean

edureka!

K-Means Algorithm



Measure the distance and add the 3rd point to the nearest cluster (red)

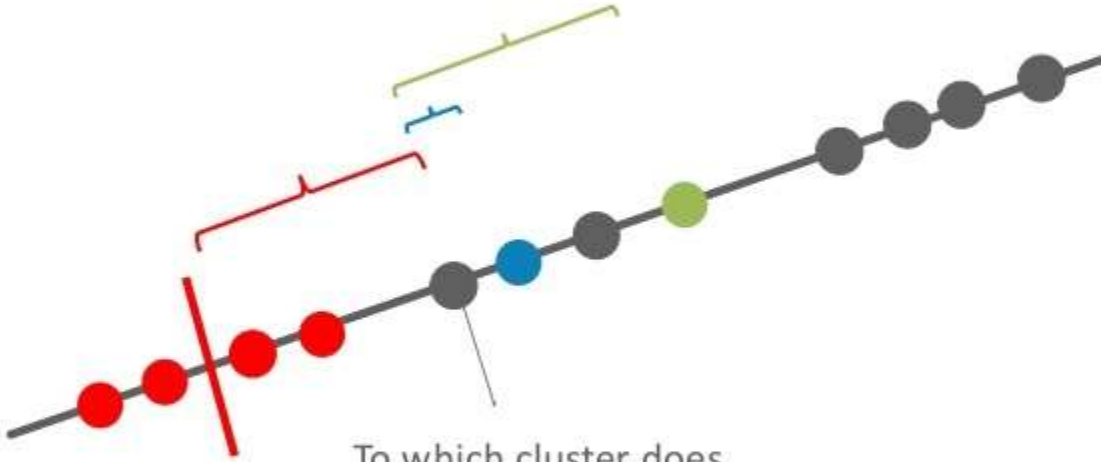
edureka!

K-Means Algorithm

edureka!

K-Means Algorithm

- Measure the distance
- Assign the point to the nearest cluster
- Calculate the cluster mean using the new point



edureka!

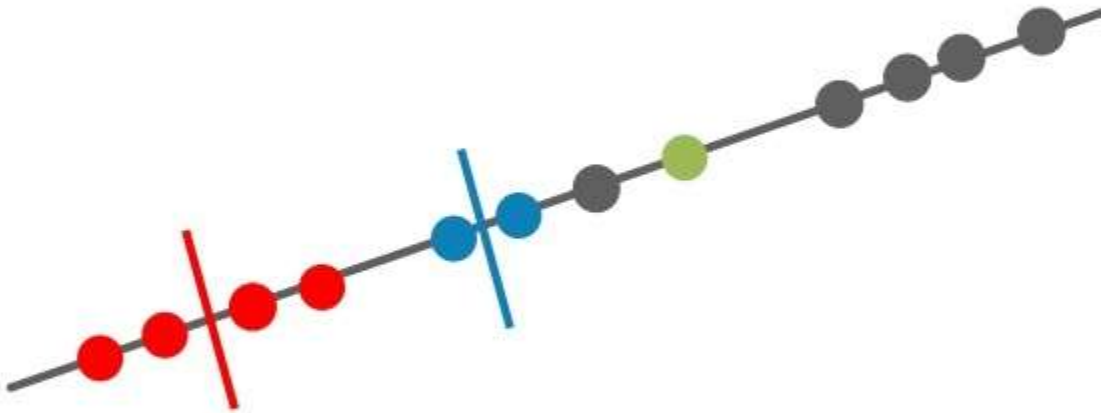
K-Means Algorithm

- Measure the distance
- Assign the point to the nearest cluster
- Calculate the cluster mean using the new point

edureka!

K-Means Algorithm

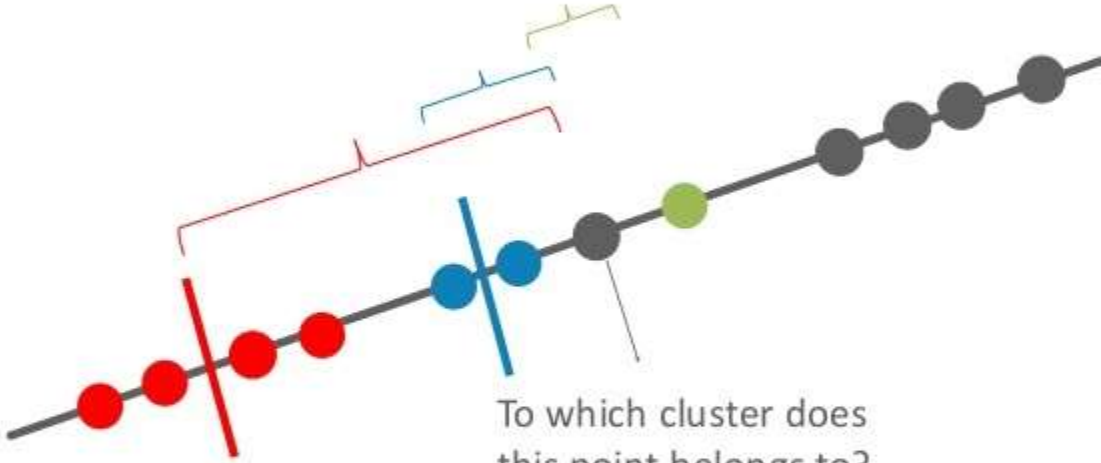
- Measure the distance
- Assign the point to the nearest cluster
- Calculate the cluster mean using the new point



edureka!

K-Means Algorithm

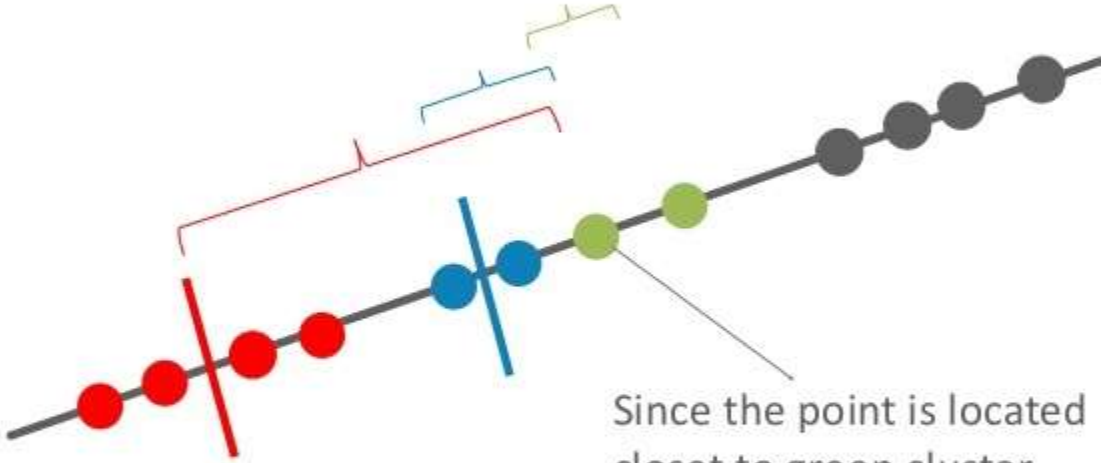
- Measure the distance from the cluster mean (centroids)
- Assign the point to the nearest cluster
- Calculate the cluster mean using the new point



edureka!

K-Means Algorithm

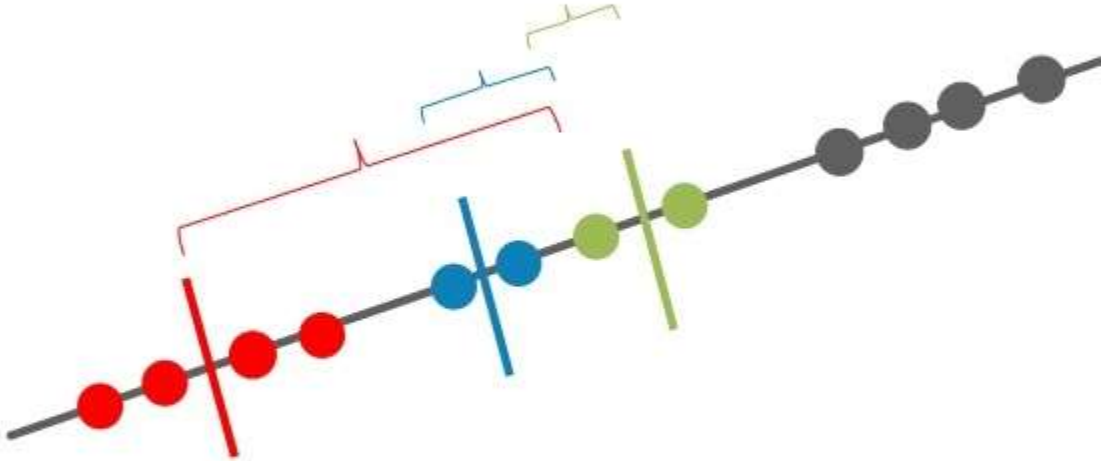
- Measure the distance from the cluster mean (centroids)
- Assign the point to the nearest cluster
- Calculate the cluster mean using the new point



edureka!

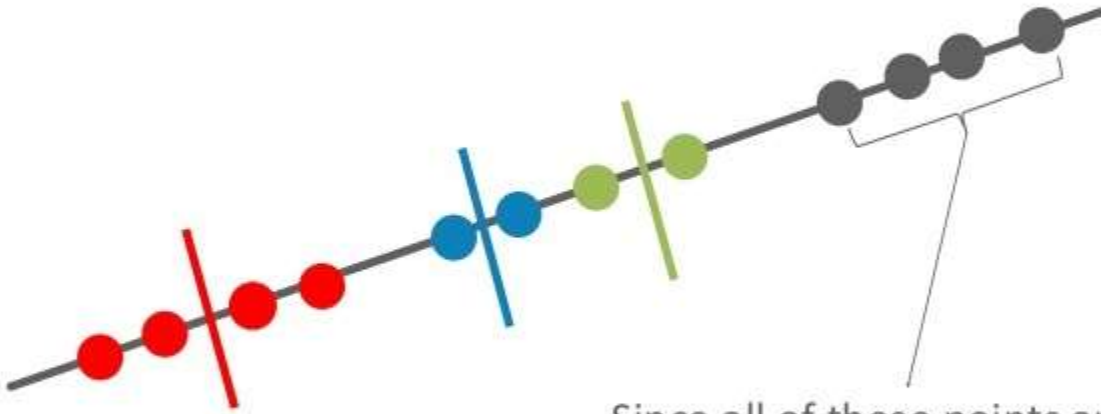
K-Means Algorithm

- Measure the distance from the cluster mean (centroids)
- Assign the point to the nearest cluster
- Calculate the cluster mean using the new point



edureka!

K-Means Algorithm



Since all of these points ar
located closet to **green** clu
so all of them will be assig
to green cluster

edureka!

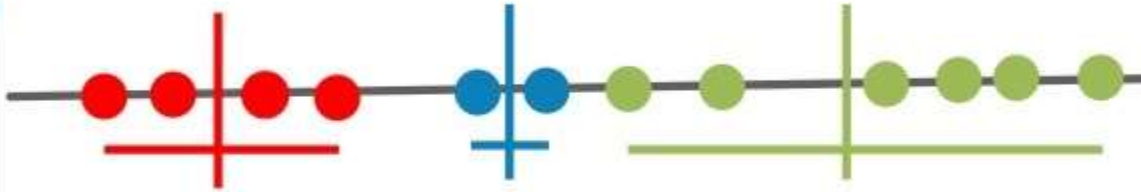
K-Means Algorithm

Result from 1st iteration

Original/Expected Result

edureka!

K-Means Algorithm



Total variation within the cluster

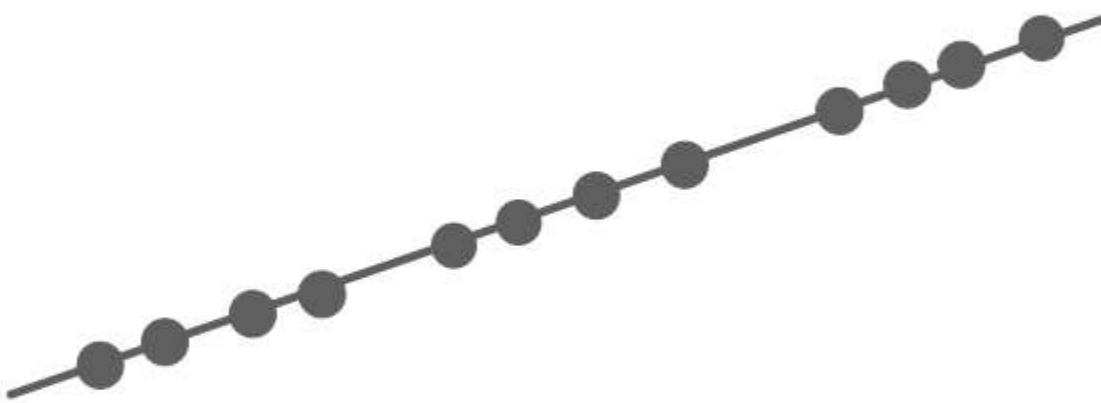
According to the K-Means Algorithm it iterates over again and ag

unless and until the data points within each cluster stops changi

edureka!

K-Means Algorithm

Iteration 2: Again we will start from the beginning. But this time we will be selecting different initial random point (as compared to what we chose in the 1st iteration)

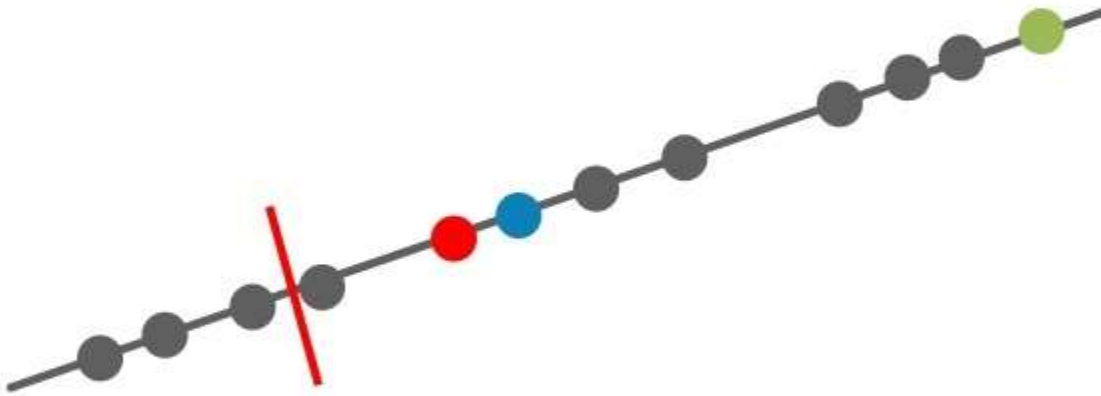


- **Step 1:** Select the number of clusters to be identified, i.e. K =3 in this
- **Step 2:** Randomly select 3 distinct data point
- **Step 3:** Measure the distance between the 1st point and selected 3 clusters

edureka!

K-Means Algorithm

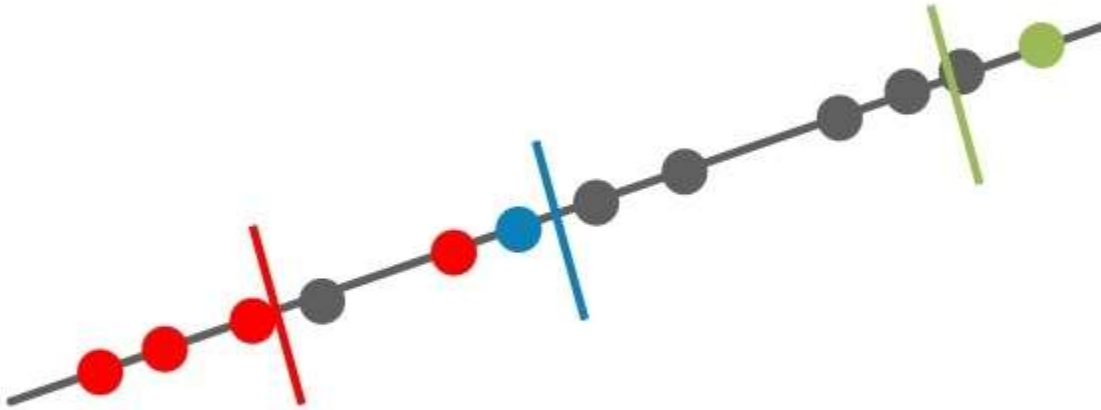
Algorithm picks 3 initial clusters and adds the remaining points to the cluster with the nearest mean, and again recalculating the mean each time a new point is added to the cluster



edureka!

K-Means Algorithm

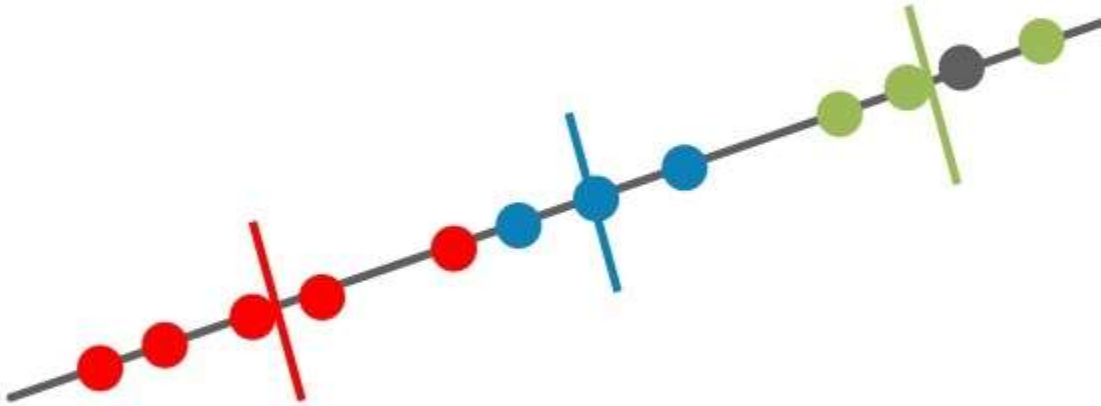
Algorithm picks 3 initial clusters and adds the remaining points to the cluster with the nearest mean, and again recalculating the mean each time a new point is added to the cluster



edureka!

K-Means Algorithm

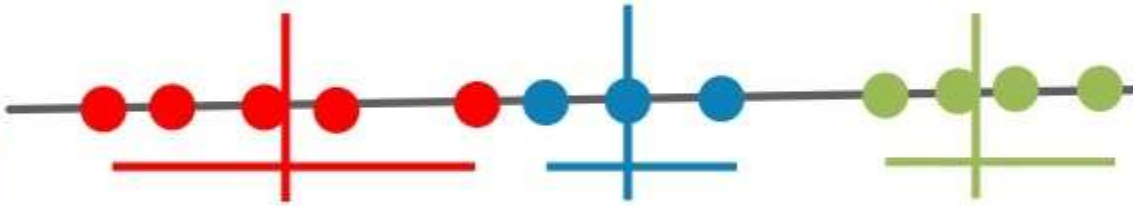
Algorithm picks 3 initial clusters and adds the remaining points to the cluster with the nearest mean, and again recalculating the mean each time a new point is added to the cluster



edureka!

K-Means Algorithm

Algorithm picks 3 initial clusters and adds the remaining points to the cluster with the nearest mean, and again recalculating the mean each time a new point is added to the cluster

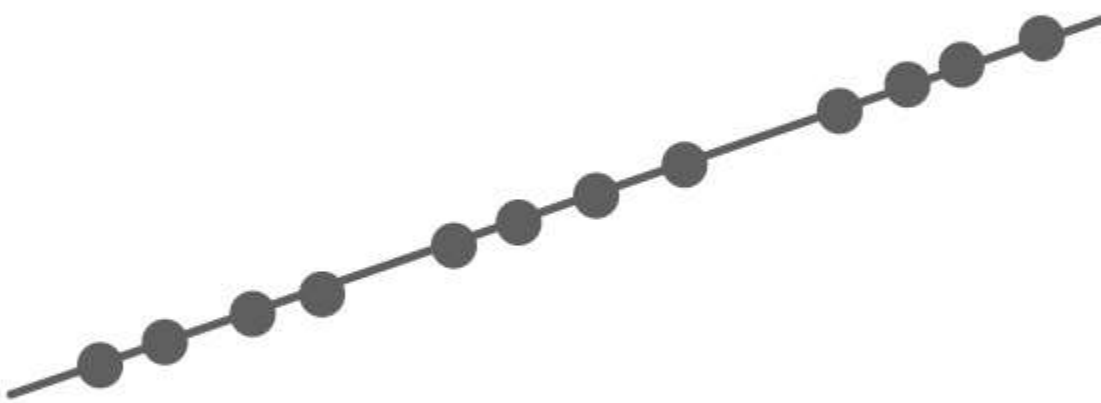


Total variation within the cluster

edureka!

K-Means Algorithm

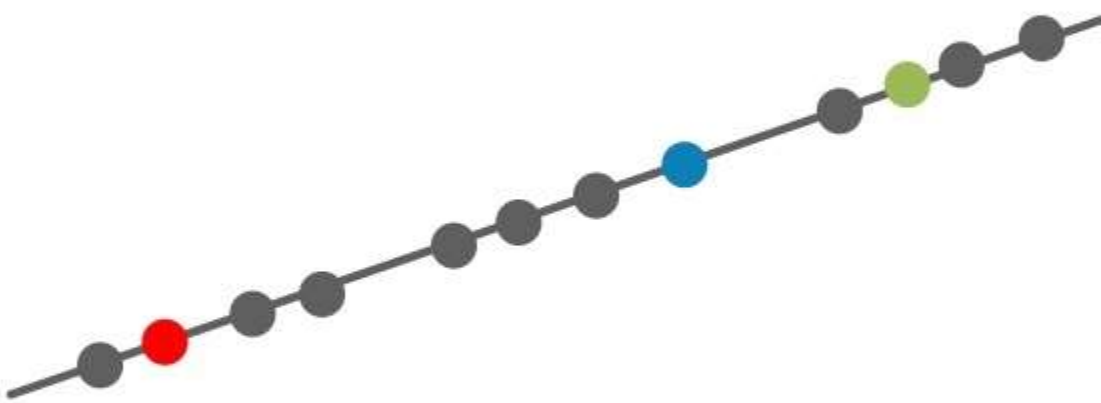
Iteration 3: Again we will start from the beginning and select different initial random point (as compared to what we chose in the 1st and 2nd iteration)



Pick 3 initial clusters

edureka!

K-Means Algorithm

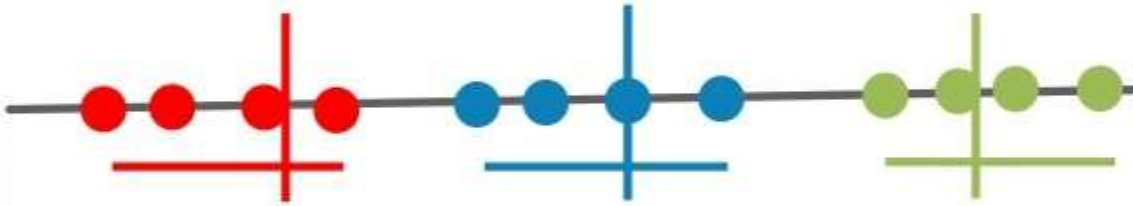


Cluster the remaining points

edureka!

K-Means Algorithm

Finally sum the variation within each cluster

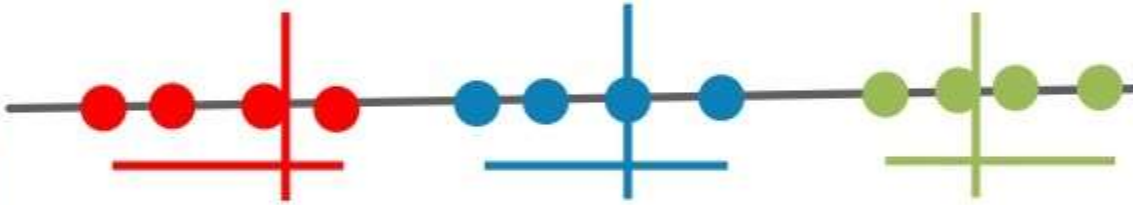


Total variation within the cluster

edureka!

K-Means Algorithm

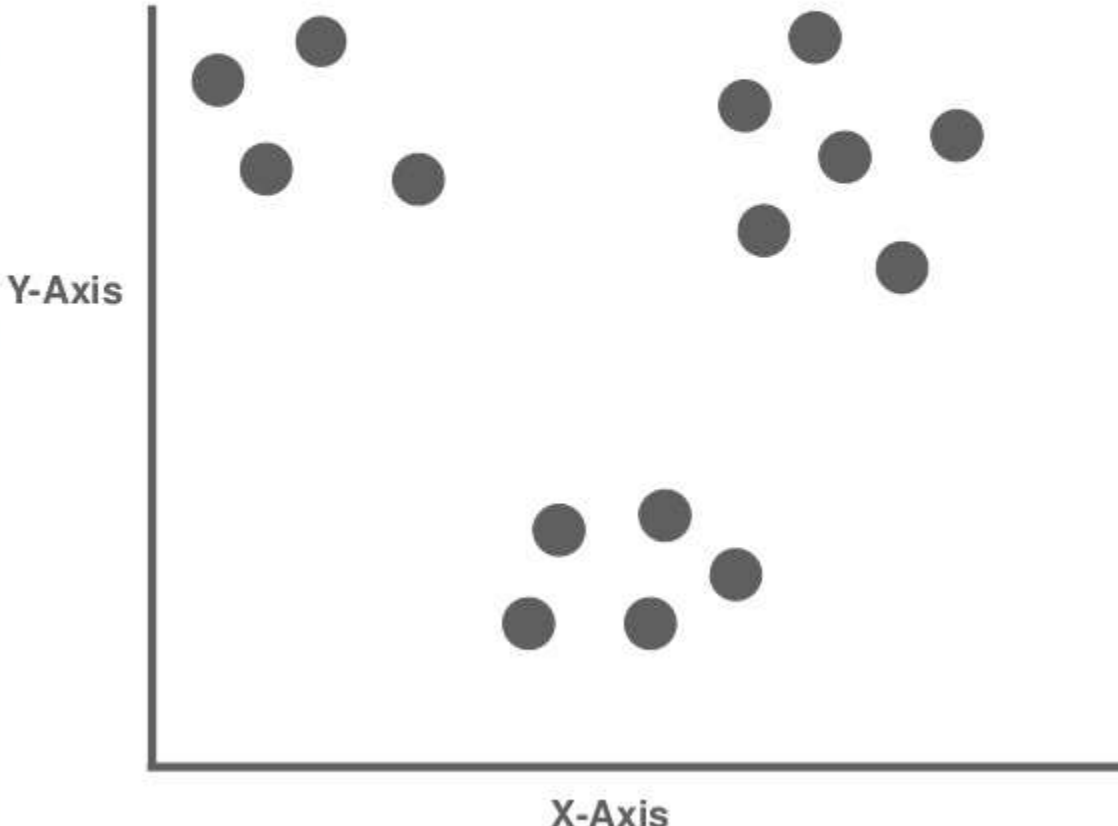
The algorithm can now compare the result and select the best variance out of it



edureka!

K-Means Algorithm

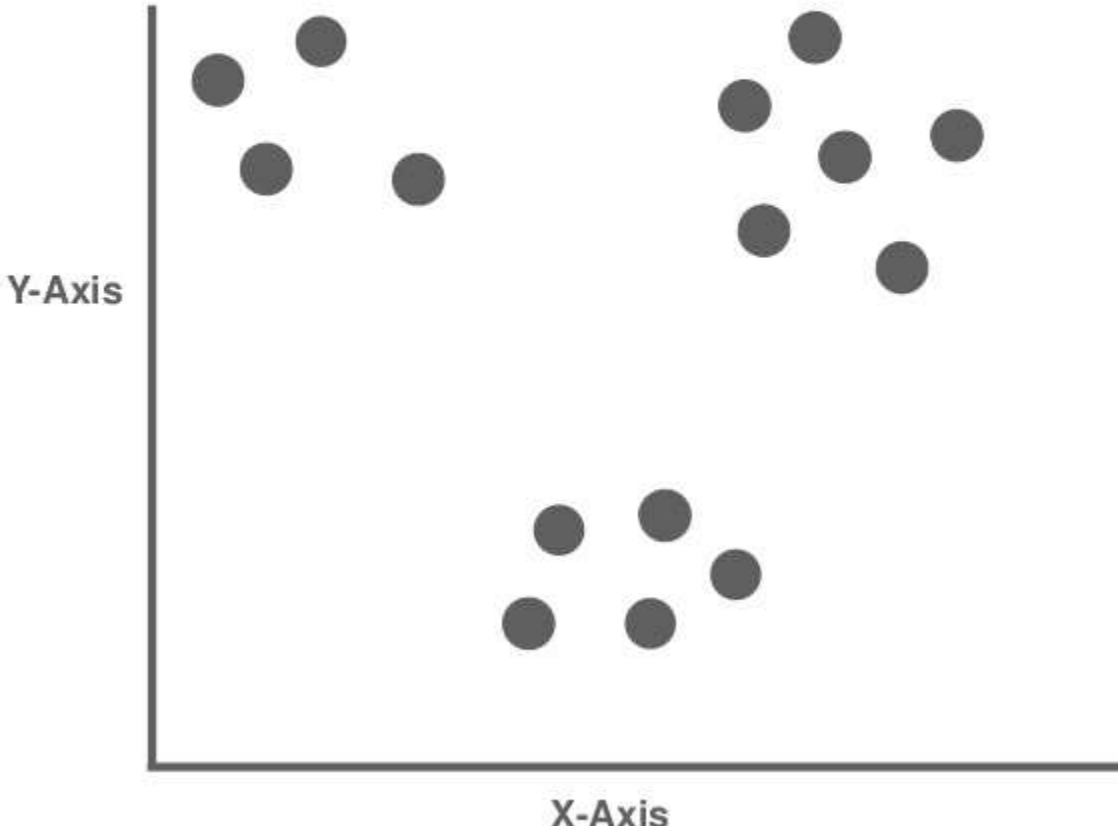
Now what if we have our data plotted on the X and Y axis



edureka!

K-Means Algorithm

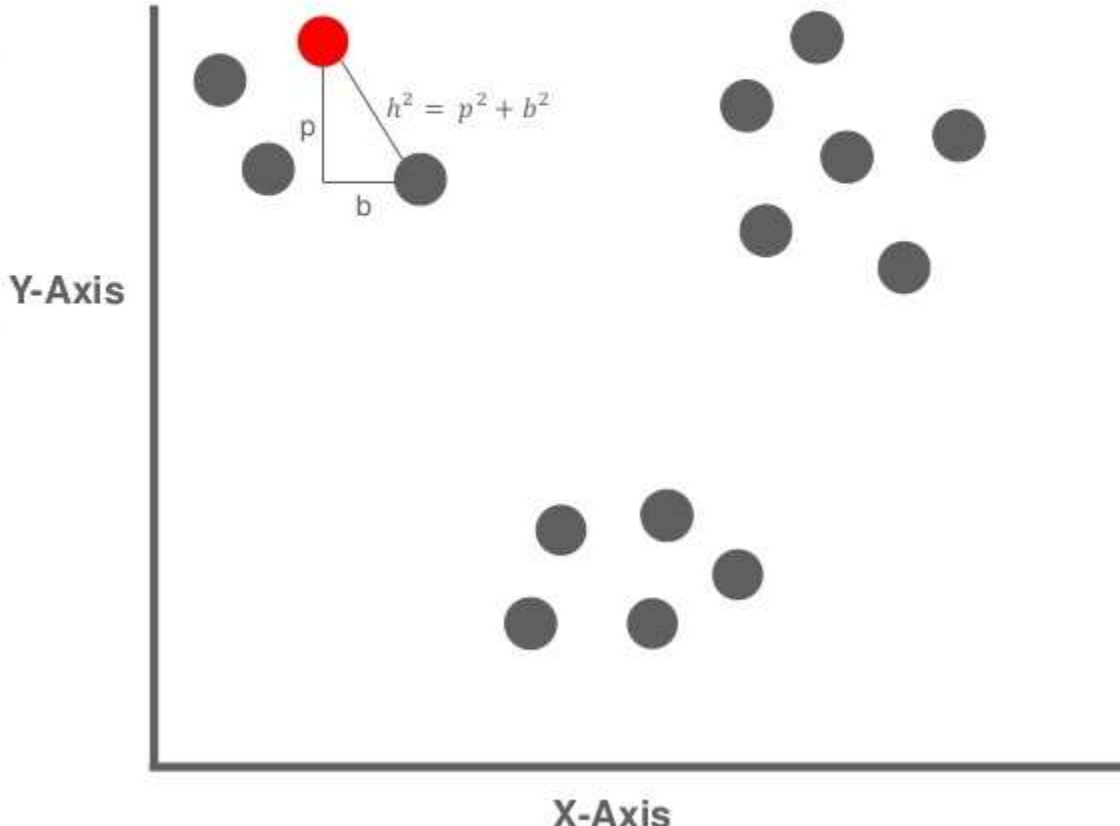
Similarly, pick initial 3 random points..



edureka!

K-Means Algorithm

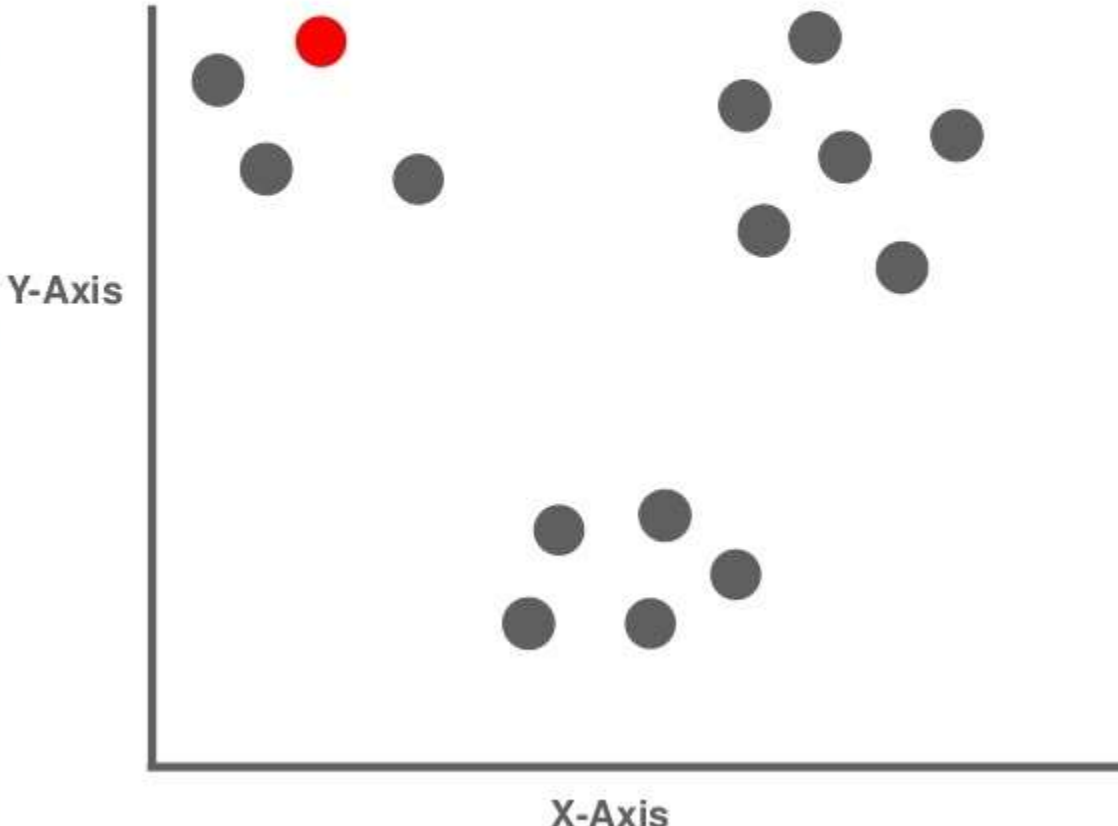
We will be using the Euclidean distance (in 2D its same as that of a Pythagorean Theorem)



edureka!

K-Means Algorithm

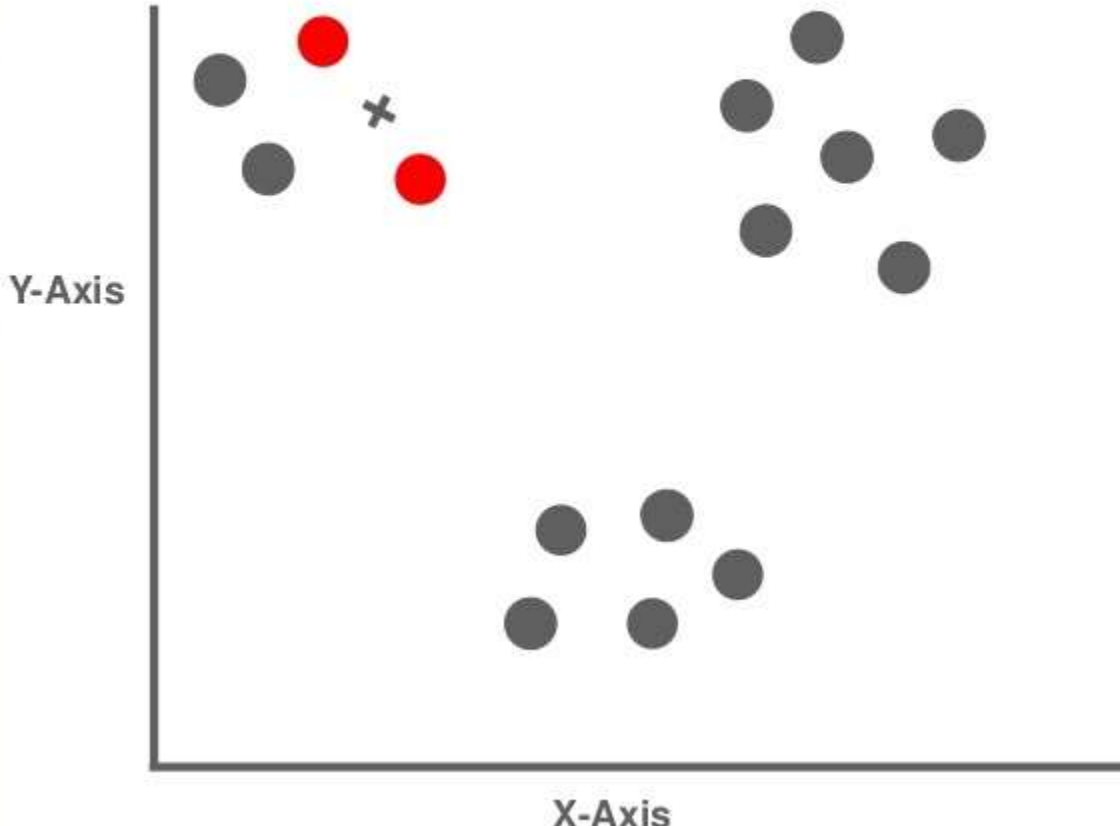
Again assign the point to the nearest cluster



edureka!

K-Means Algorithm

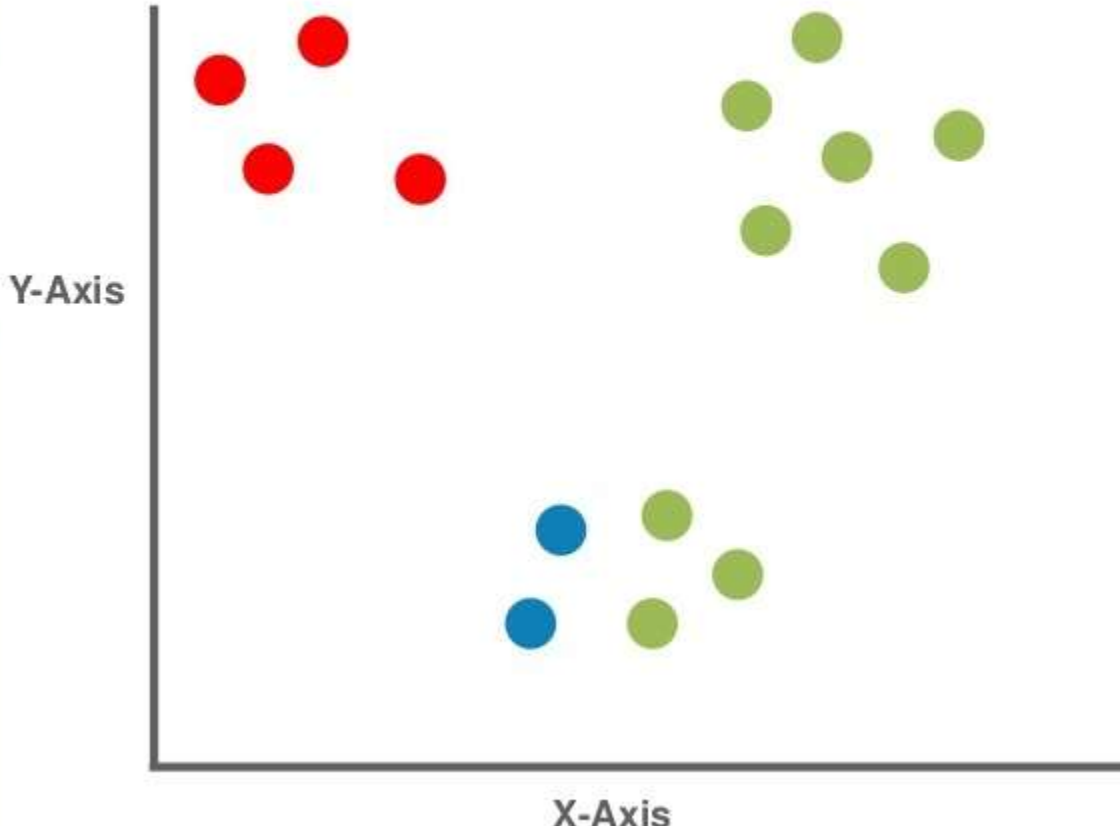
Finally calculate the centroid (mean of cluster) including the new point



edureka!

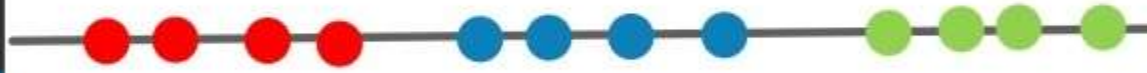
K-Means Algorithm

Finally in first iteration you get something like this...again you have to iterate this process to get the final cluster



How will you find K value

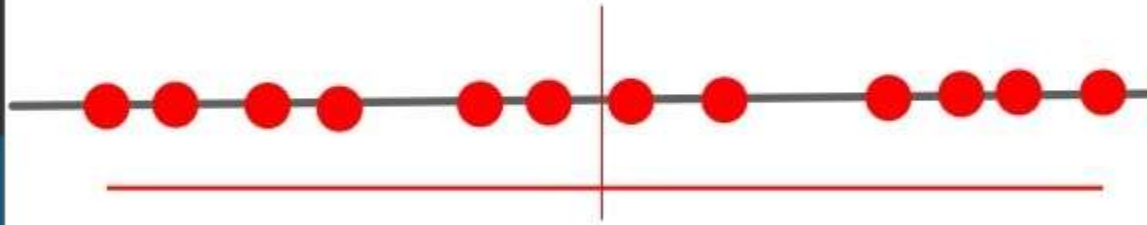
In the previous scenario k value was known to be 3, but this is not always true



edureka!

How will you find K value

For deciding the value of k, you have to use hi
and trail method, starting from K = 1

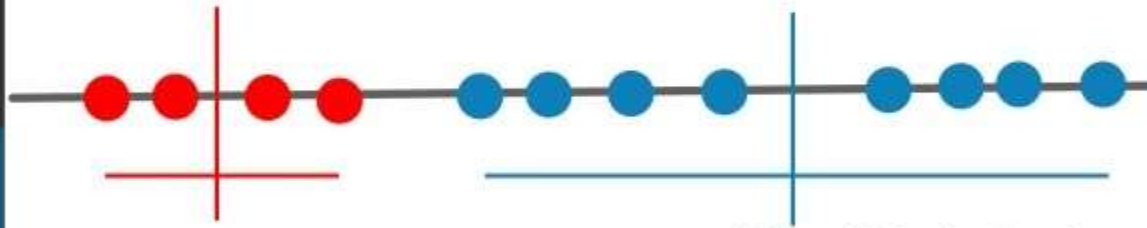


K=1 is the worst case scenario, even you cross-verify it with total variation

edureka!

How will you find K value

Now try with K = 2

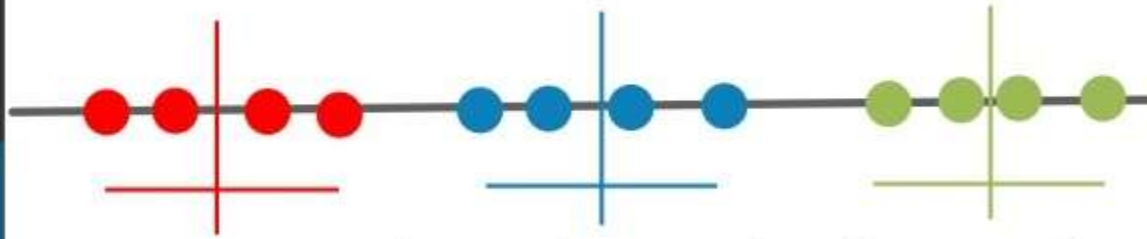


K=2 is still better then K = 1 (Total Variation)

edureka!

How will you find K value

Now try with K = 3



K=3 is even better than K =2 (Total Variation)

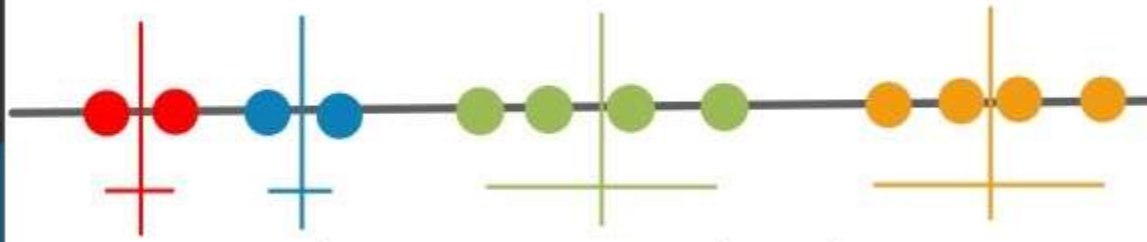
K = 1

K = 2

edureka!

How will you find K value

Now try with K = 4



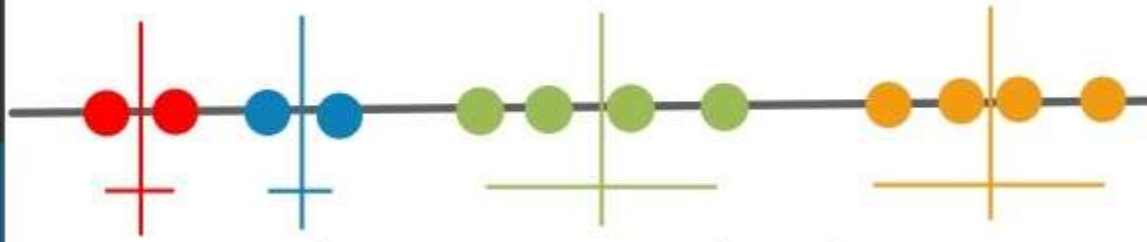
Total variation in K=4 is less than K =3

edureka!

How will you find K value

Now try with K = 4

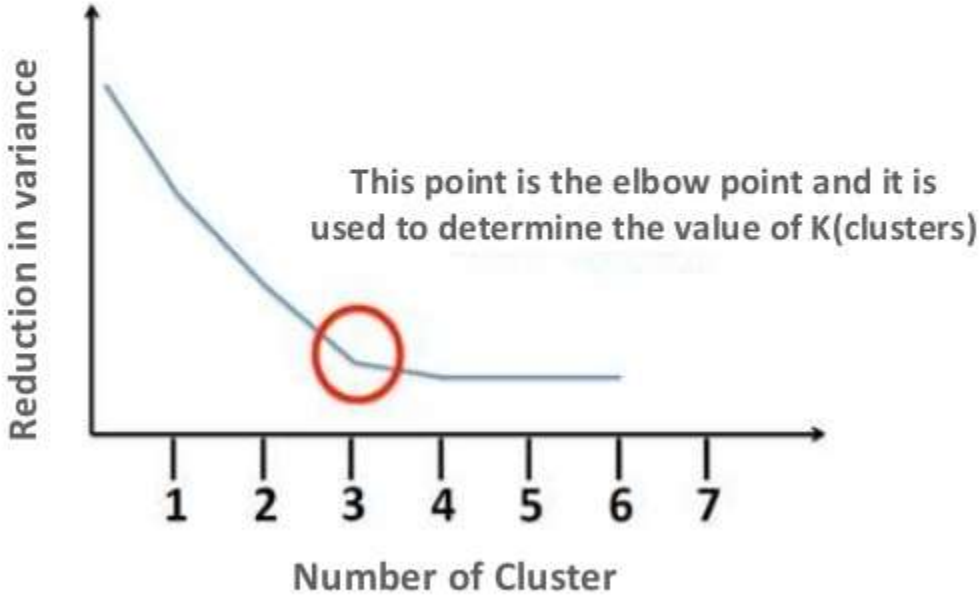
Each time you increase the cluster the variation decreases, no. of clusters = no. of data points ther in that case the variation = 0



Total variation in K=4 is less than K =3

edureka!

How will you find K value



Let’s learn to code

edureka!

K-Means Algorithm

Summarizing the K-Means Algorithm

```
randomly chose k examples as initial centroid
while true:
    create k clusters by assigning each
        example to closest centroid
    compute k new centroids by averaging
        examples in each cluster
    if centroids don't change:
        break
```

edurek

600,000+
SATISFIED LEARNERS
Thank you! ☺

For more information please visit our website
www.edureka.co