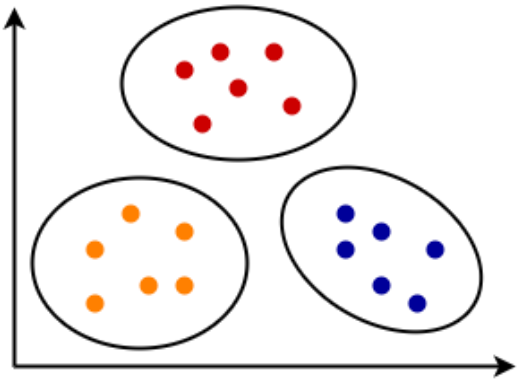
K-Means Clustering-

- K-Means clustering is an unsupervised iterative clustering technique.
- It partitions the given data set into k predefined distinct clusters.
- A cluster is defined as a collection of data points exhibiting certain similarities.

Before K-Means



After K-Means

It partitions the data set such that-

- Each data point belongs to a cluster with the nearest mean.
- Data points belonging to one cluster have high degree of similarity.
- Data points belonging to different clusters have high degree of dissimilarity.

K-Means Clustering Algorithm-

K-Means Clustering Algorithm involves the following steps-

Step-01:

- Choose the number of clusters K.

Step-02:

- Randomly select any K data points as cluster centers.

- Select cluster centers in such a way that they are as farther as possible from each other.

Step-03:

- Calculate the distance between each data point and each cluster center.
- The distance may be calculated either by using given distance function or by using euclidean distance formula.

Step-04:

- Assign each data point to some cluster.
- A data point is assigned to that cluster whose center is nearest to that data point.

Step-05:

- Re-compute the center of newly formed clusters.
- The center of a cluster is computed by taking mean of all the data points contained in that cluster.

Step-06:

Keep repeating the procedure from Step-03 to Step-05 until any of the following stopping criteria is met-

- Center of newly formed clusters do not change
- Data points remain present in the same cluster
- Maximum number of iterations are reached

Advantages-

K-Means Clustering Algorithm offers the following advantages-

Point-01:

It is relatively efficient with time complexity O(nkt) where-

- n = number of instances
- k = number of clusters
- t = number of iterations

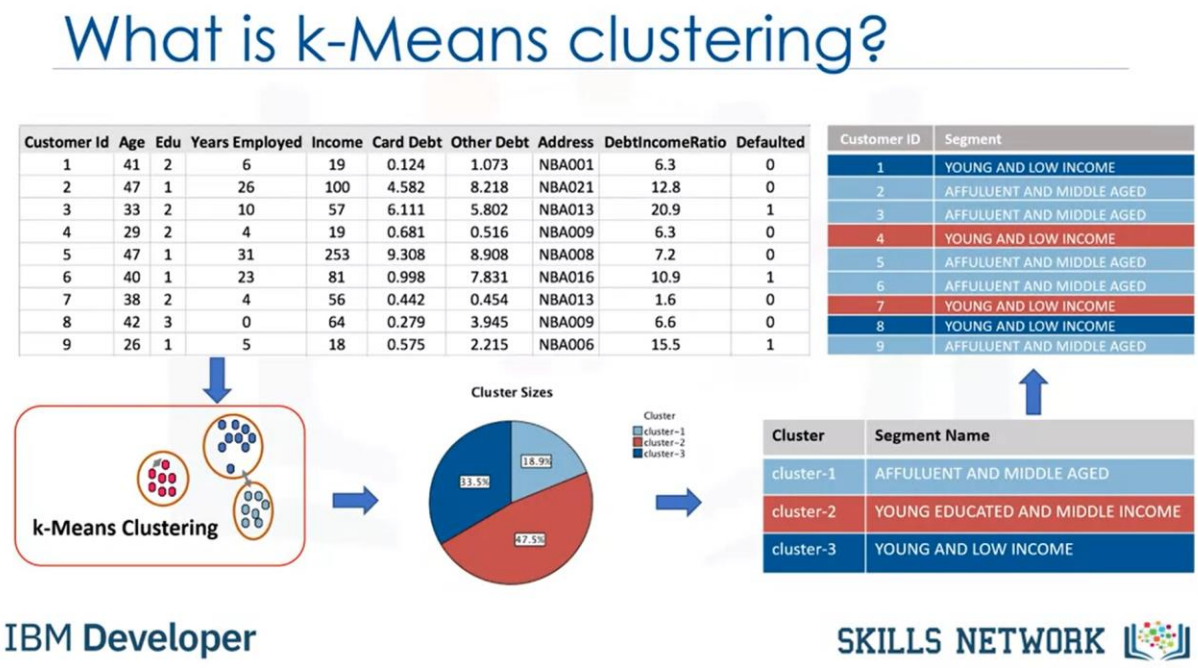
Point-02:

- It often terminates at local optimum.
- Techniques such as Simulated Annealing or **Genetic Algorithms** may be used to find the global optimum.

Disadvantages-

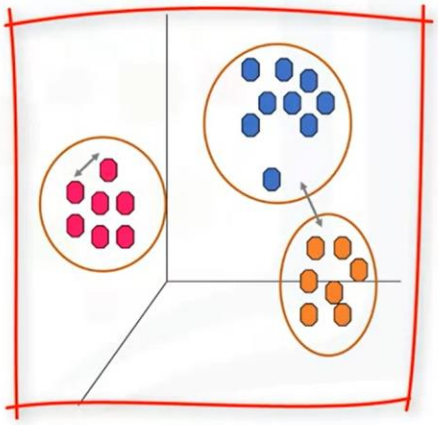
K-Means Clustering Algorithm has the following disadvantages-

- It requires to specify the number of clusters (k) in advance.
- It can not handle noisy data and outliers.
- It is not suitable to identify clusters with non-convex shapes.

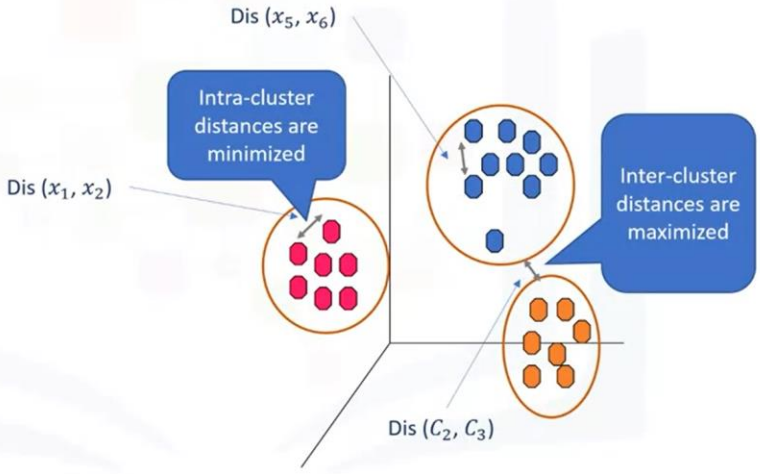


k-Means algorithms

- Partitioning Clustering
- K-means divides the data into non-overlapping subsets (clusters) without any cluster-internal structure
- Examples within a cluster are very similar
- Examples across different clusters are very different



Determine the similarity or dissimilarity



1-dimensional similarity/distance

Customer 1
Age
54

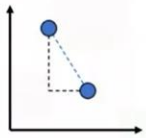
Customer 2
Age
50

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

$$\text{Dis}\,(x_1, x_2) = \sqrt{(54 - 50)^2} = 4$$

2-dimensional similarity/distance

Customer 1	
Age	Income
54	190



Customer 2	
Age	Income
50	200

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

$$= \sqrt{(54 - 50)^2 + (190 - 200)^2} = 10.77$$

Multi-dimentional similarity/distance

Customer 1		
Age	Income	education
54	190	3

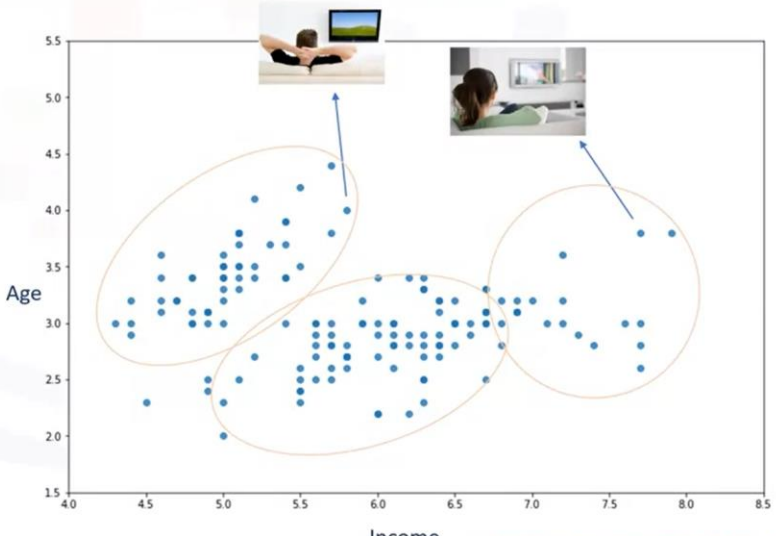
Customer 2		
Age	Income	education
50	200	8

$$\text{Dis}\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

$$= \sqrt{(54-50)^2+(190-200)^2+(3-8)^2} = 11.87$$

How does k-Means clustering work?

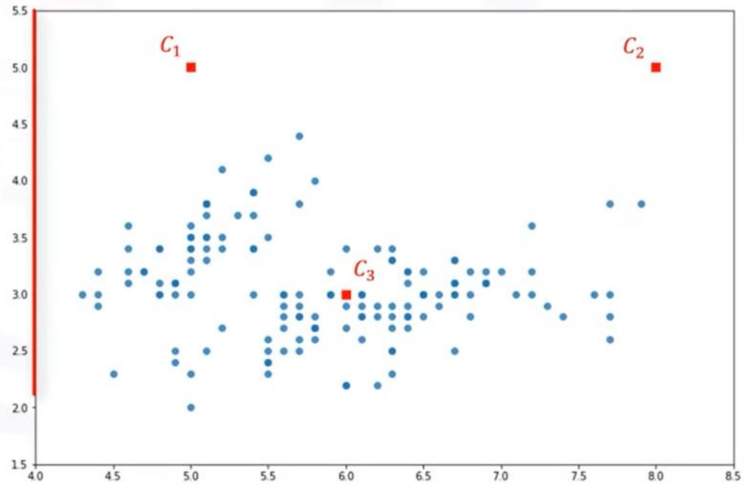
Customer ID	Age	Income
1	3	4
2	2	6
3	3.5	2
...	...	..



k-Means clustering – initialize k

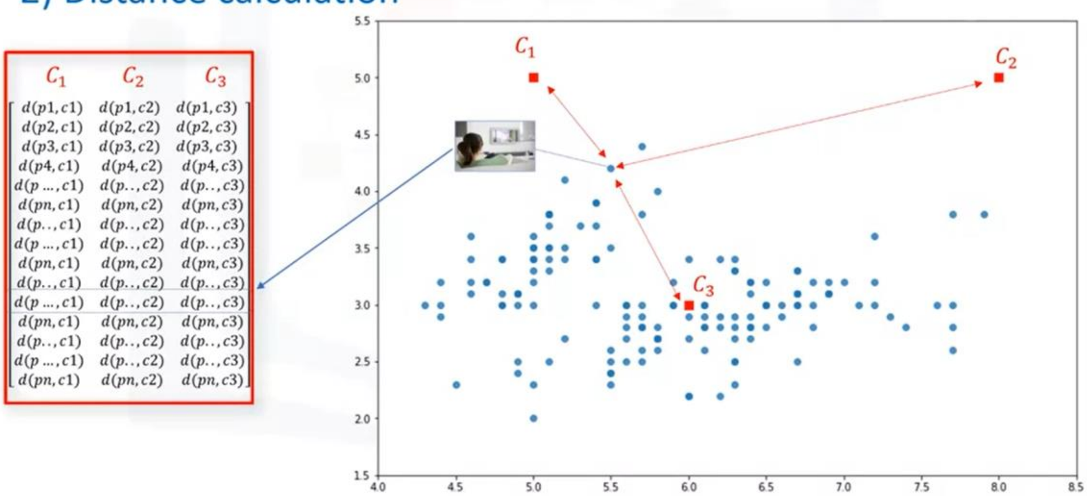
1) Initialize k=3 centroids randomly

C_1 = [8., 5.]
C_2 = [5., 5.]
C_3 = [6., 3.]



K-Means clustering – calculate the distance

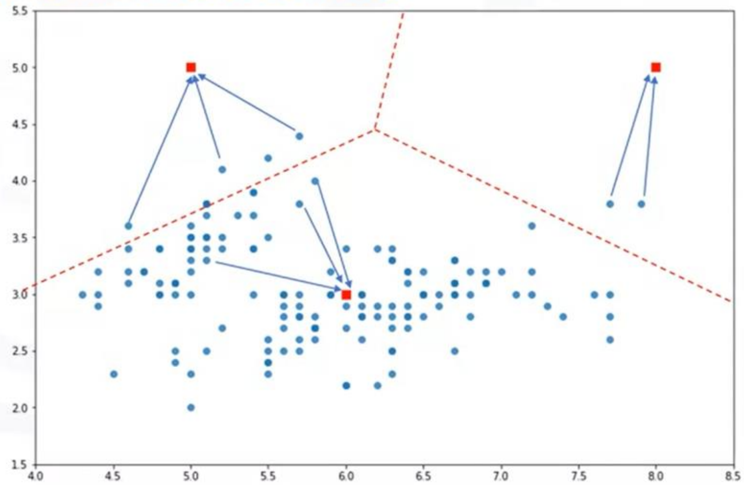
2) Distance calculation



k-Means clustering – assign to centroid

3) Assign each point to the closest centroid

C_1	C_2	C_3
$d(p1, c1)$	$d(p1, c2)$	$d(p1, c3)$
$d(p2, c1)$	$d(p2, c2)$	$d(p2, c3)$
$d(p3, c1)$	$d(p3, c2)$	$d(p3, c3)$
$d(p4, c1)$	$d(p4, c2)$	$d(p4, c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$
$d(p.., c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$
$d(p.., c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$
$d(p.., c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$



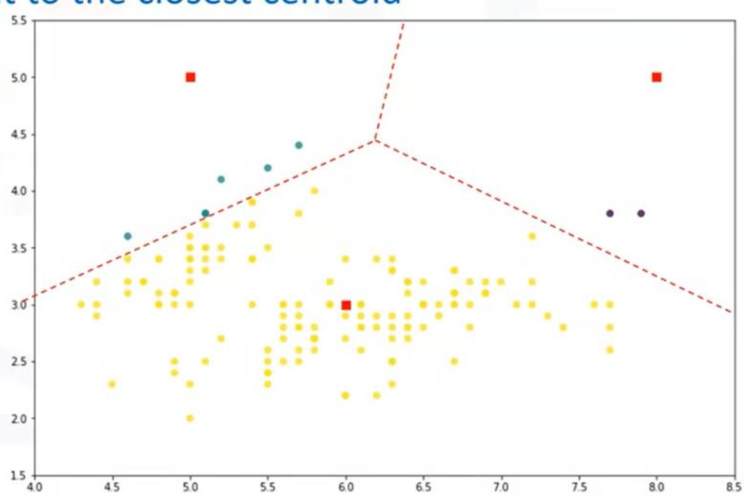
IBM **Developer**

SKILLS NETWORK

k-Means clustering – assign to centroid

3) Assign each point to the closest centroid

C_1	C_2	C_3
$d(p1, c1)$	$d(p1, c2)$	$d(p1, c3)$
$d(p2, c1)$	$d(p2, c2)$	$d(p2, c3)$
$d(p3, c1)$	$d(p3, c2)$	$d(p3, c3)$
$d(p4, c1)$	$d(p4, c2)$	$d(p4, c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$
$d(p.., c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$
$d(p.., c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$
$d(p.., c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(p\dots, c1)$	$d(p.., c2)$	$d(p.., c3)$
$d(pn, c1)$	$d(pn, c2)$	$d(pn, c3)$

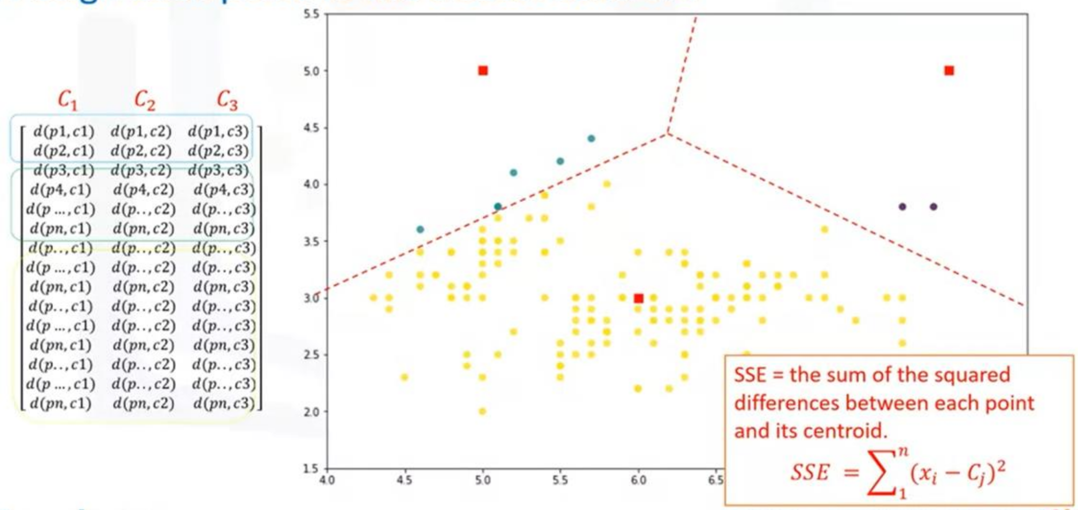


IBM **Developer**

SKILLS NETWORK

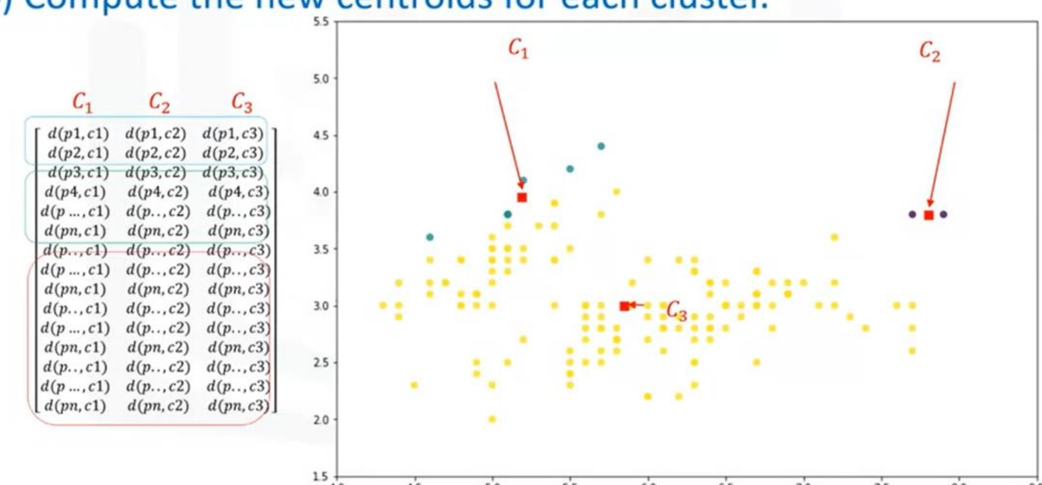
k-Means clustering – assign to centroid

3) Assign each point to the closest centroid



k-Means clustering – compute new centroids

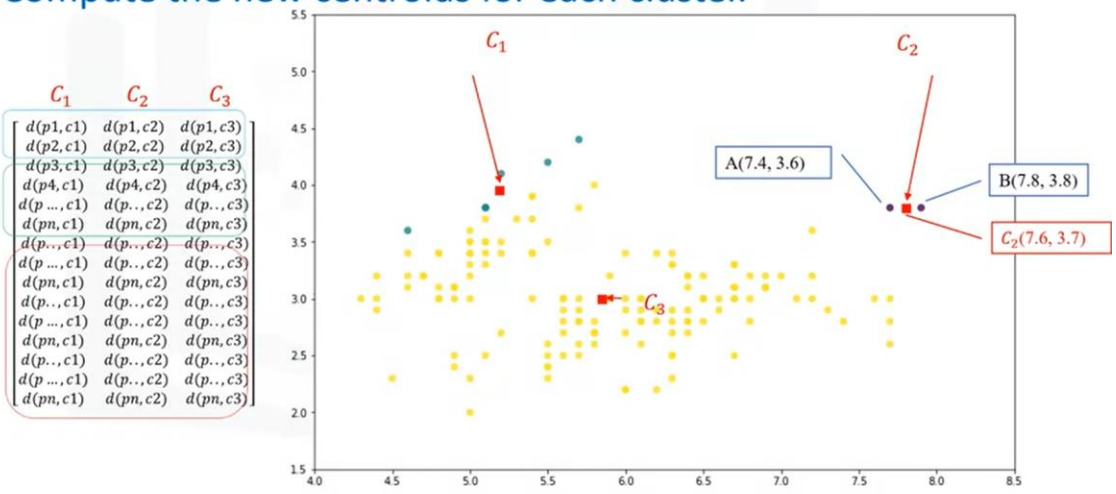
4) Compute the new centroids for each cluster.



SKILLS NETWORK

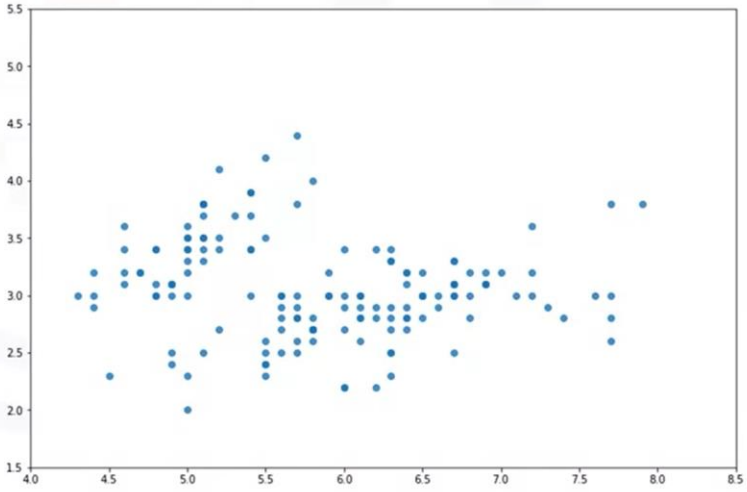
k-Means clustering – compute new centroids

4) Compute the new centroids for each cluster.



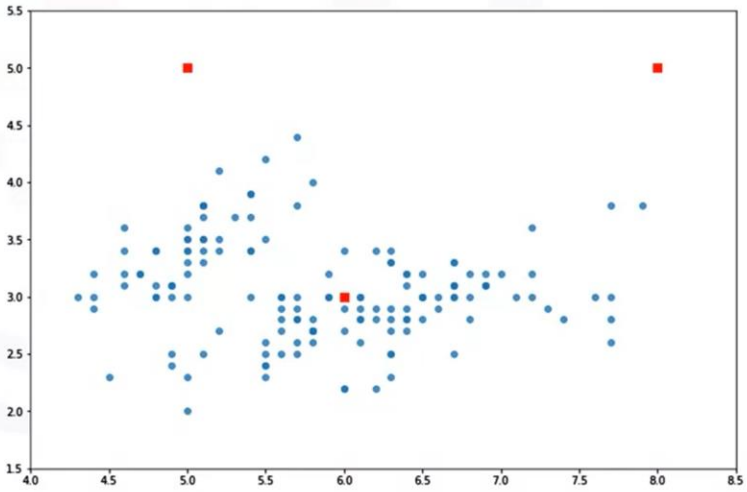
k-Means clustering – repeat

5) Repeat until there are no more changes.



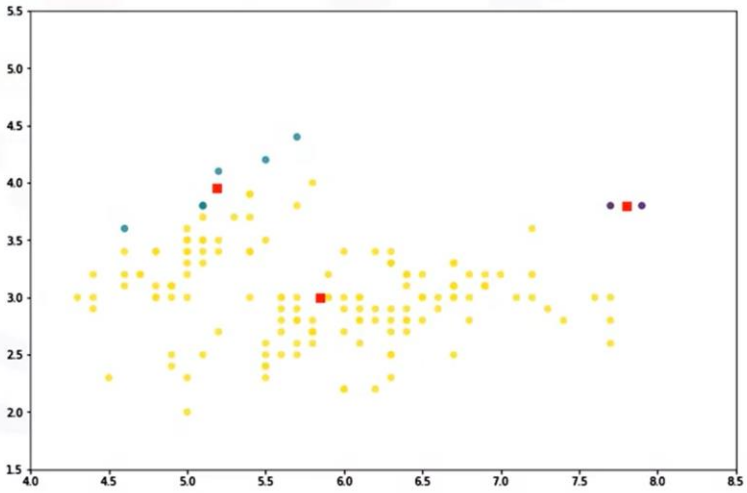
k-Means clustering – repeat

5) Repeat until there are no more changes.



k-Means clustering – repeat

5) Repeat until there are no more changes.

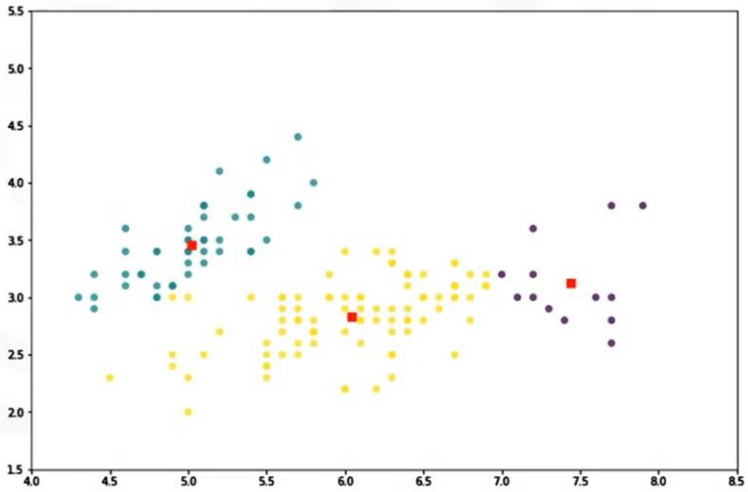


IBM Developer

SKILLS NETWORK

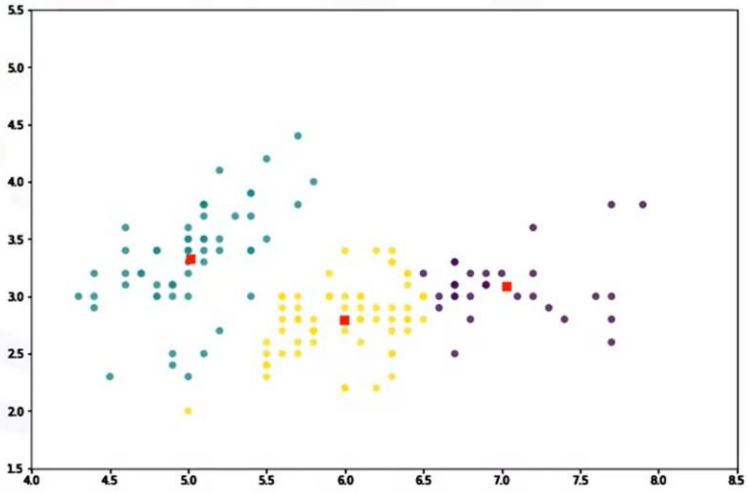
k-Means clustering – repeat

5) Repeat until there are no more changes.



k-Means clustering – repeat

5) Repeat until there are no more changes.

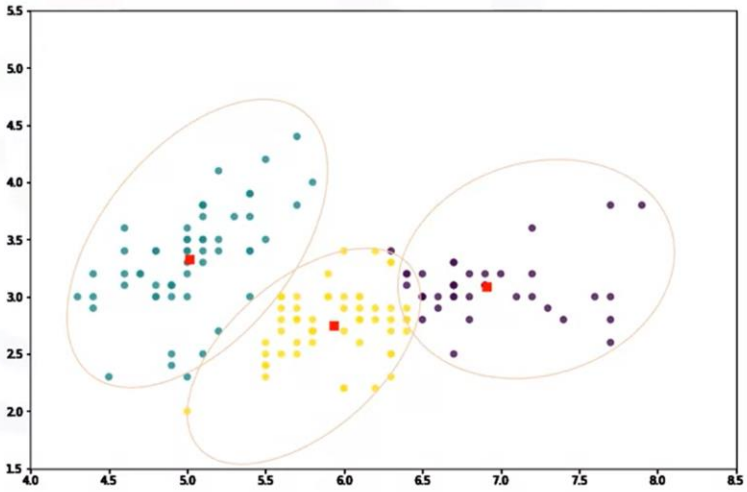


IBM Developer

SKILLS NETWORK

k-Means clustering – repeat

5) Repeat until there are no more changes.



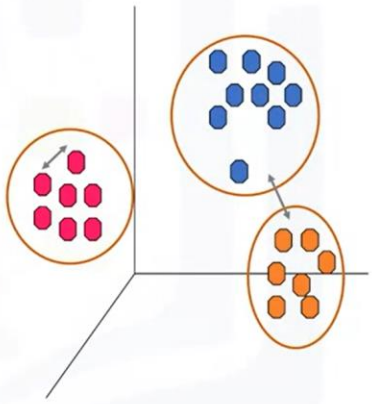
k-Means clustering algorithm

1. Randomly placing *k* centroids, one for each cluster.
2. Calculate the distance of each point from each centroid.
3. Assign each data point (object) to its closest centroid, creating a cluster.
4. Recalculate the position of the *k* centroids.
5. Repeat the steps 2-4, until the centroids no longer move.

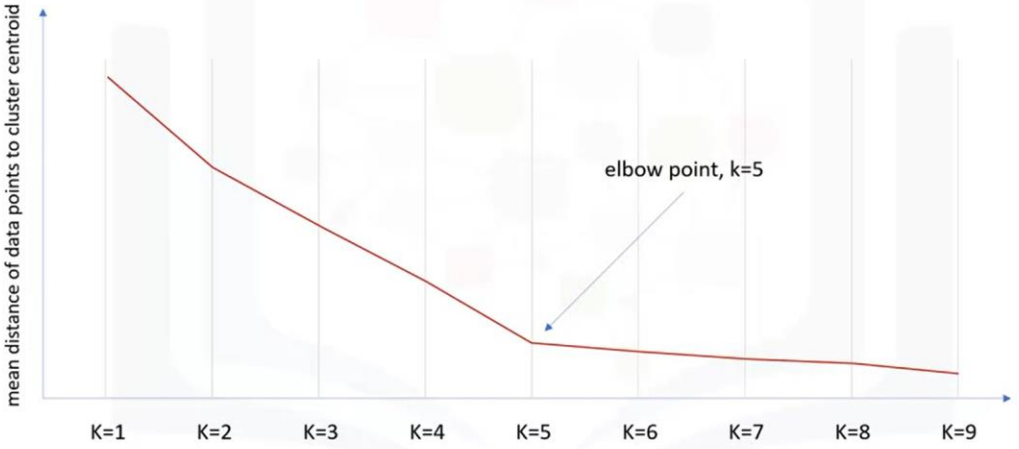
,

k-Means accuracy

- External approach
  - Compare the clusters with the ground truth, if it is available.
- Internal approach
  - Average the distance between data points within a cluster.



Choosing k



k-Means recap

- Med and Large sized databases (*Relatively efficient*)
- Produces sphere-like clusters
- Needs number of clusters (k)