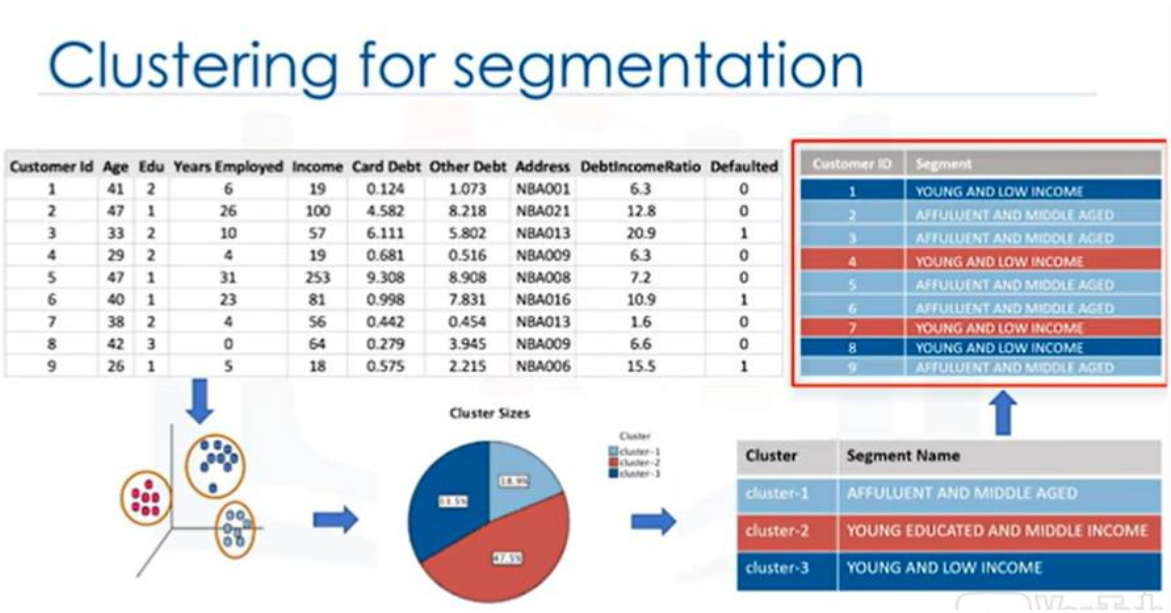
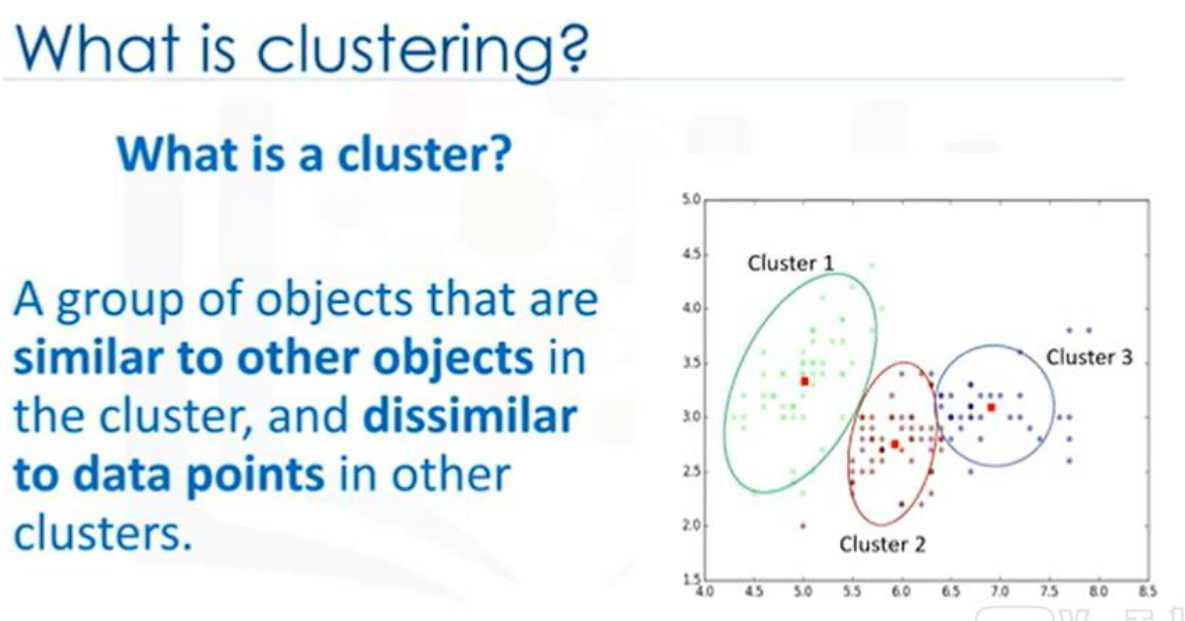
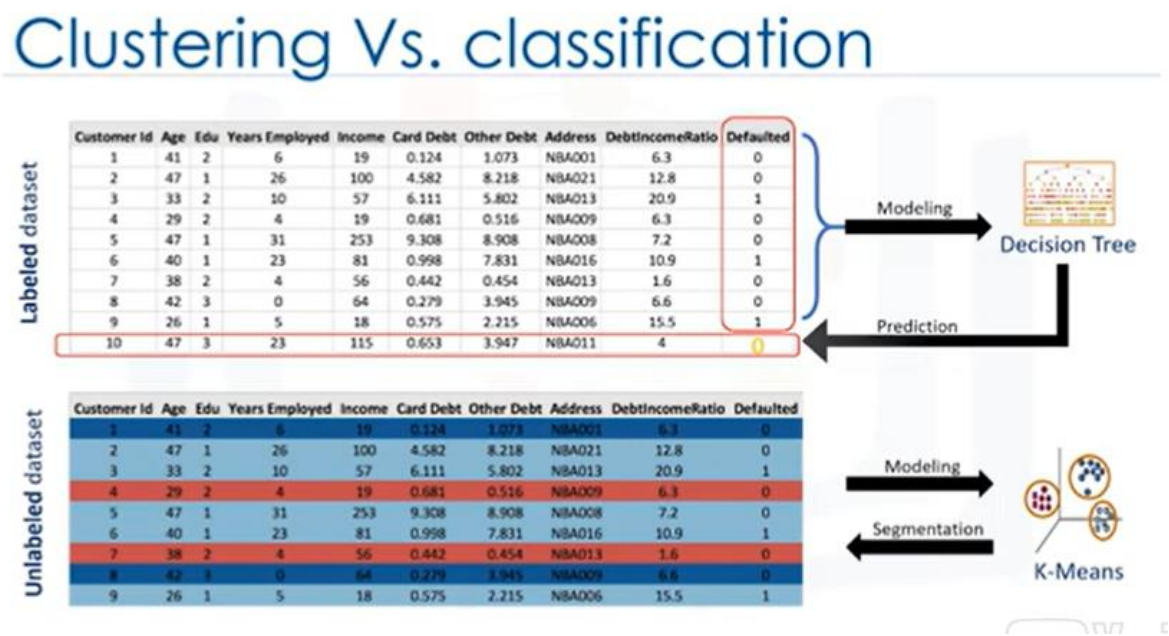
Clustering:

In this module, you will learn about different clustering approaches. You will learn how to use clustering for customer segmentation, grouping same vehicles, and for weather stations. You will understand 3 main types of clustering including Partitioned-based Clustering, Hierarchical Clustering, and Density-based Clustering.

Intro to Clustering:







Clustering Applications

- Retail/Marketing:
  - Identifying buying patterns of customers
  - Recommending new books or movies to new customers
- Banking
  - Fraud detection in credit card use
  - Identifying clusters of customers (e.g., loyal)
- Insurance
  - Fraud detection in claims analysis
  - Insurance risk of customers
- Publication
  - Auto-categorizing news based on their content
  - Recommending similar news articles
- Medicine
  - Characterizing patient behavior
- Biology
  - Clustering genetic markers to identify family ties

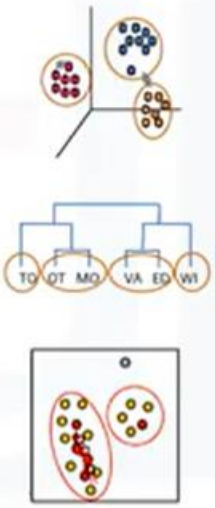
Why clustering?

- Exploratory data analysis
- Summary generation
- Outlier detection
- Finding duplicates
- Pre-processing step

Clustering algorithms

Clustering algorithms

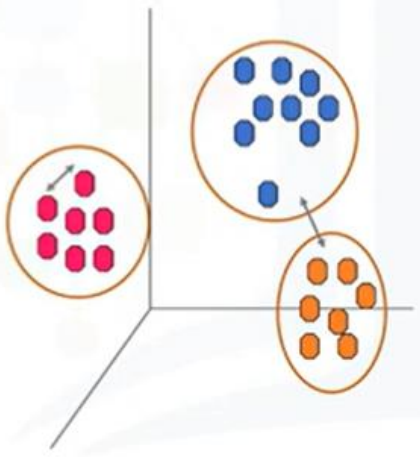
- Partitioned-based Clustering
  - Relatively efficient
  - E.g. k-Means, k-Median, Fuzzy c-Means
- Hierarchical Clustering
  - Produces trees of clusters
  - E.g. Agglomerative, Divisive
- Density-based Clustering
  - Produces arbitrary shaped clusters
  - E.g. DBSCAN



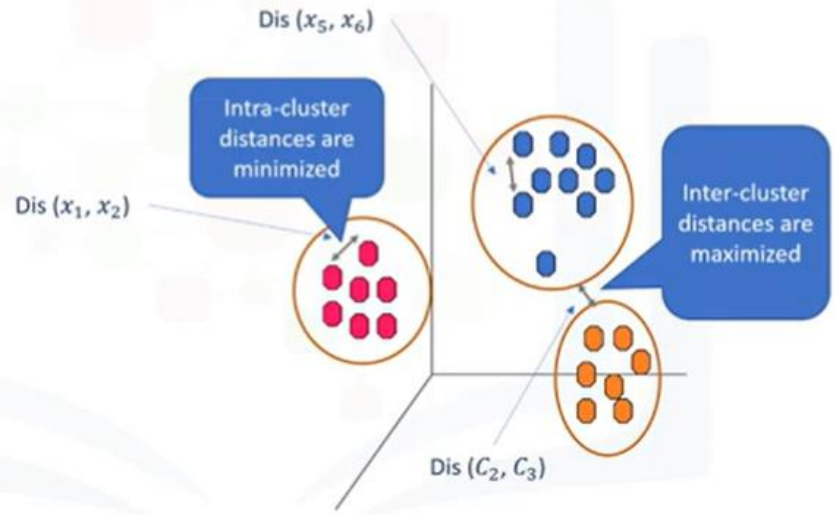
Intro to k-means

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

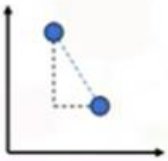
Customer 2
Age
50

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

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

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

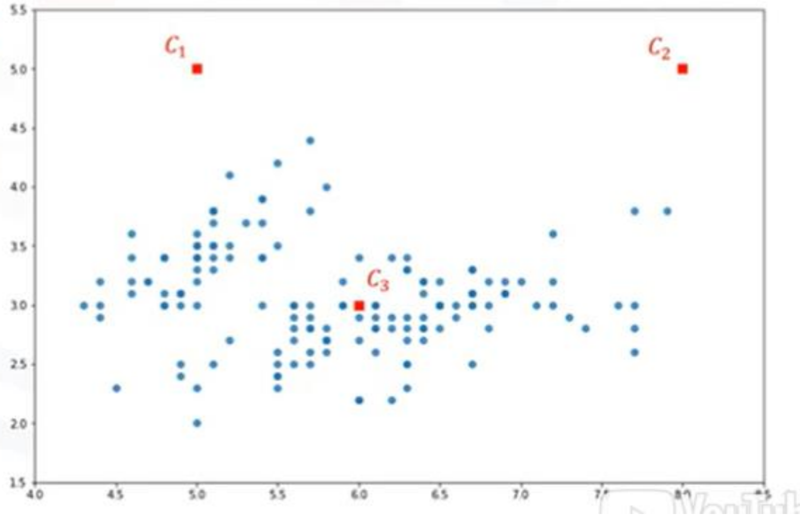
$$\text{Dis}(x_1, x_2) = \sqrt{\sum_{i=0}^{n}(x_{1i} - x_{2i})^2}$$

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

k-Means clustering – initialize k

1) Initialize k=3 centroids randomly

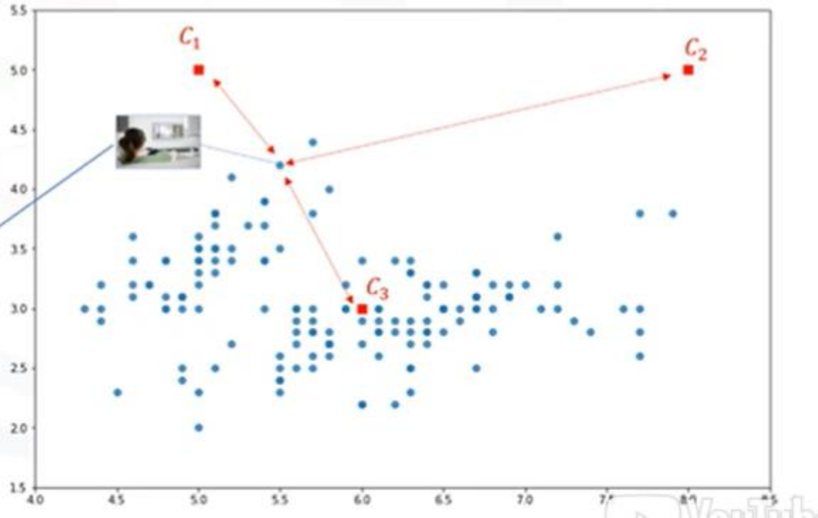
C_1 = [8., 5.]
C_2 = [5., 5.]
C_3 = [6., 3.]



K-Means clustering – calculate the distance

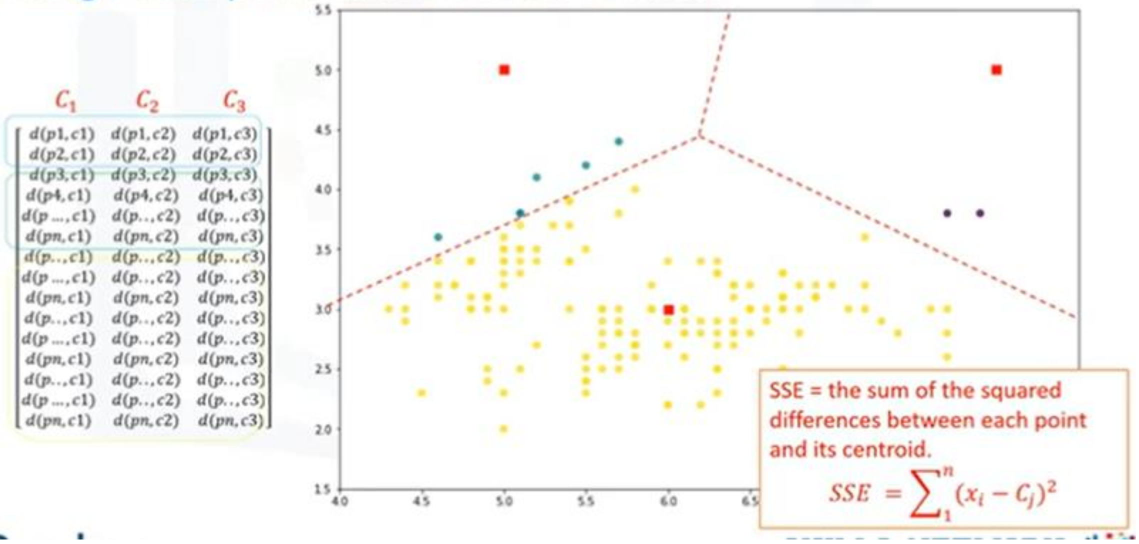
2) Distance calculation

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



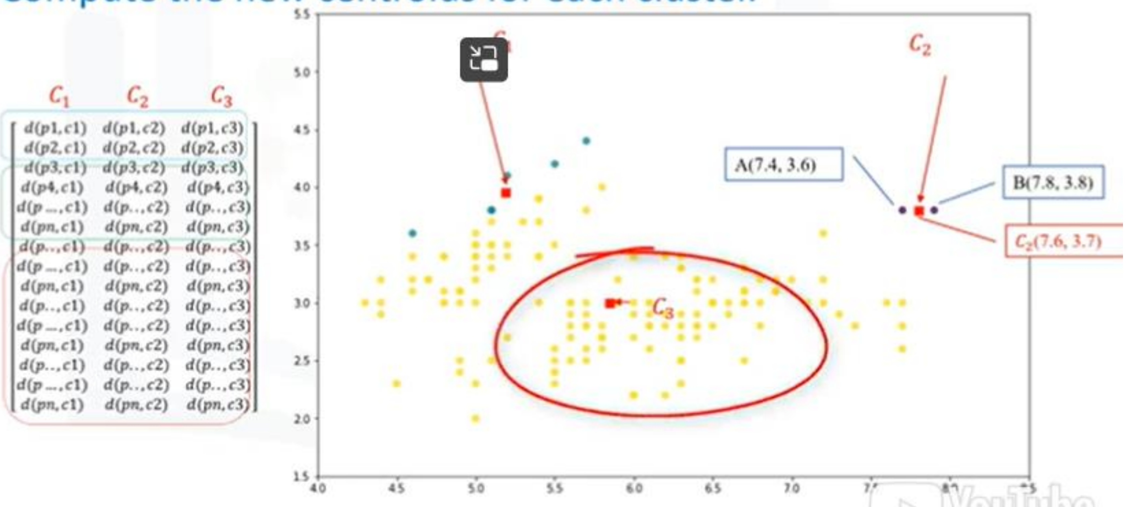
k-Means clustering – assign to centroid

3) Assign each point to the closest centroid



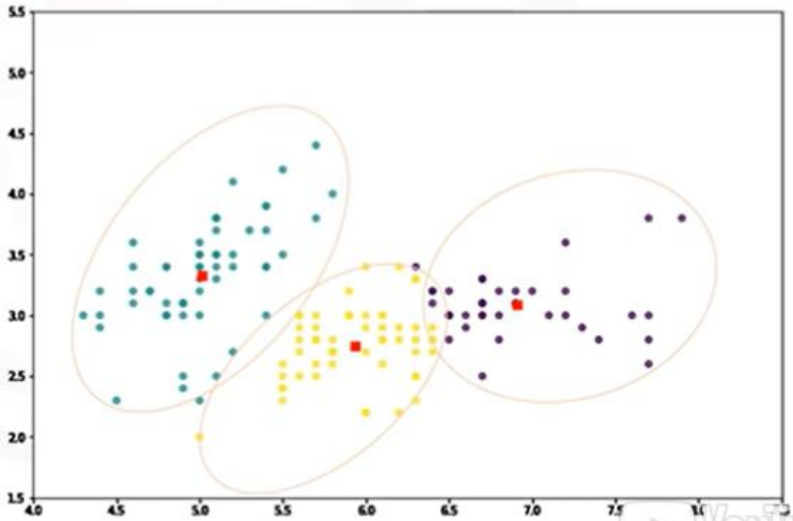
k-Means clustering – compute new centroids

4) Compute the new centroids for each cluster.



k-Means clustering – repeat

5) Repeat until there are no more changes.



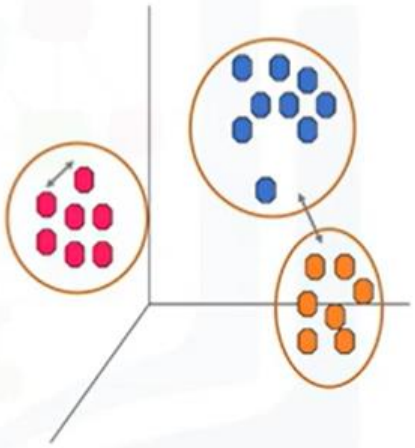
More on k-means

k-Means clustering algorithm

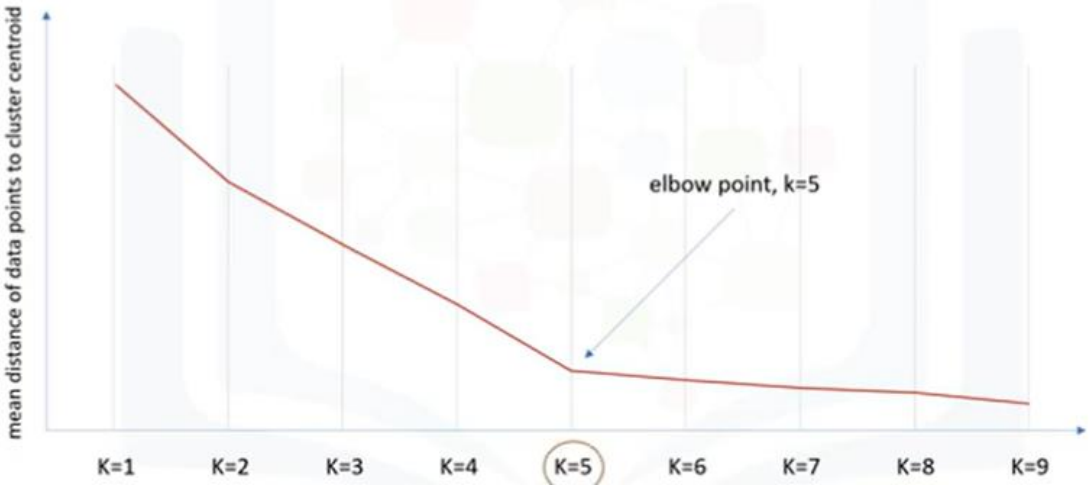
1. Randomly placing *k* centroids, one for each cluster.
2. Calculate the distance of each point from each centroid.
3. Assign each data point (object) to its closest centroid, creating a cluster.
4. Recalculate the position of the *k* centroids.
5. Repeat the steps 2-4, until the centroids no longer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



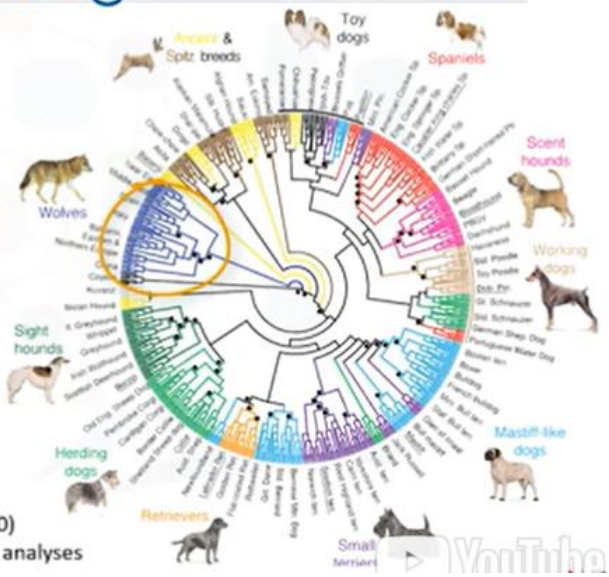
k-Means recap

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

Intro to Hierarchical Clustering

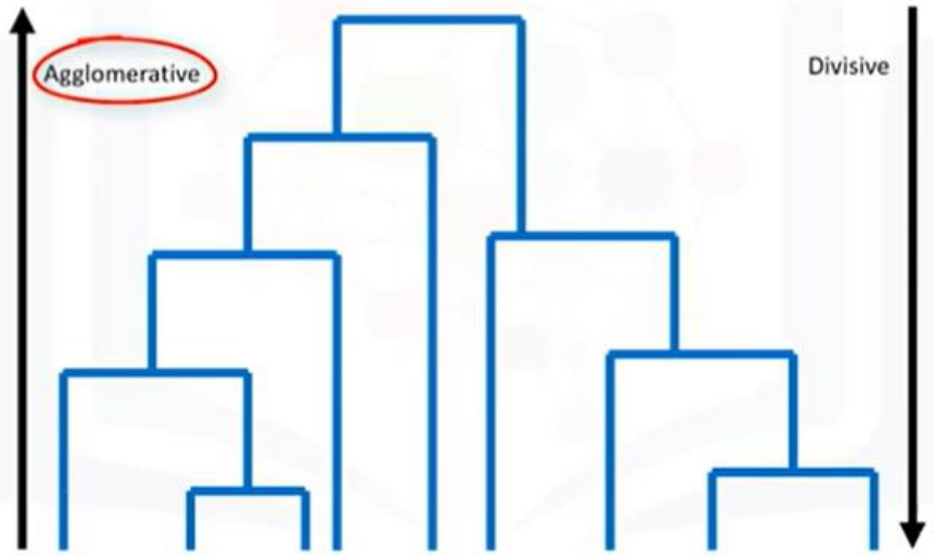
Hierarchical clustering

Hierarchical clustering algorithms build a hierarchy of clusters where each node is a cluster consists of the clusters of its daughter nodes.



Source: von Holdt B.M. et al. (2010)
Genome-wide SNP and haplotype analyses

Hierarchical clustering



Agglomerative algorithm

1. Create *n* clusters, one for each data point
2. Compute the Proximity Matrix
3. **Repeat**
   i. Merge the two closest clusters
   ii. Update the proximity matrix
4. **Until** only a single cluster remains

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Similarity/Distance

Patient 1		
Age	BMI	BP
54	190	120

Patient 2		
Age	BMI	BP
50	200	125

Dis (p1,p2)

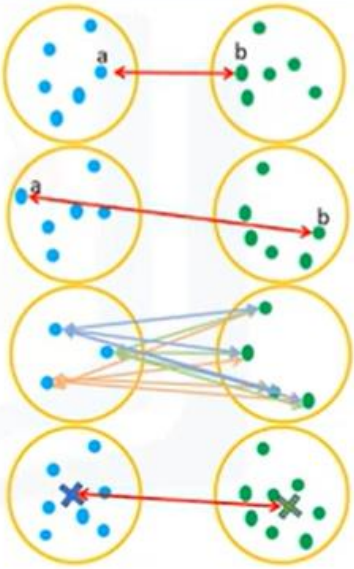
$$= \sqrt{\sum_{i=0}^{n}(x_i - y_i)^2}$$

$$= \sqrt{(54-50)^2+(190-200)^2+(120-125)^2}$$

$$= 11.87$$

Distance between clusters

- Single-Linkage Clustering
  - Minimum distance between clusters
- Complete-Linkage Clustering
  - Maximum distance between clusters
- Average Linkage Clustering
  - Average distance between clusters
- Centroid Linkage Clustering
  - Distance between cluster centroids



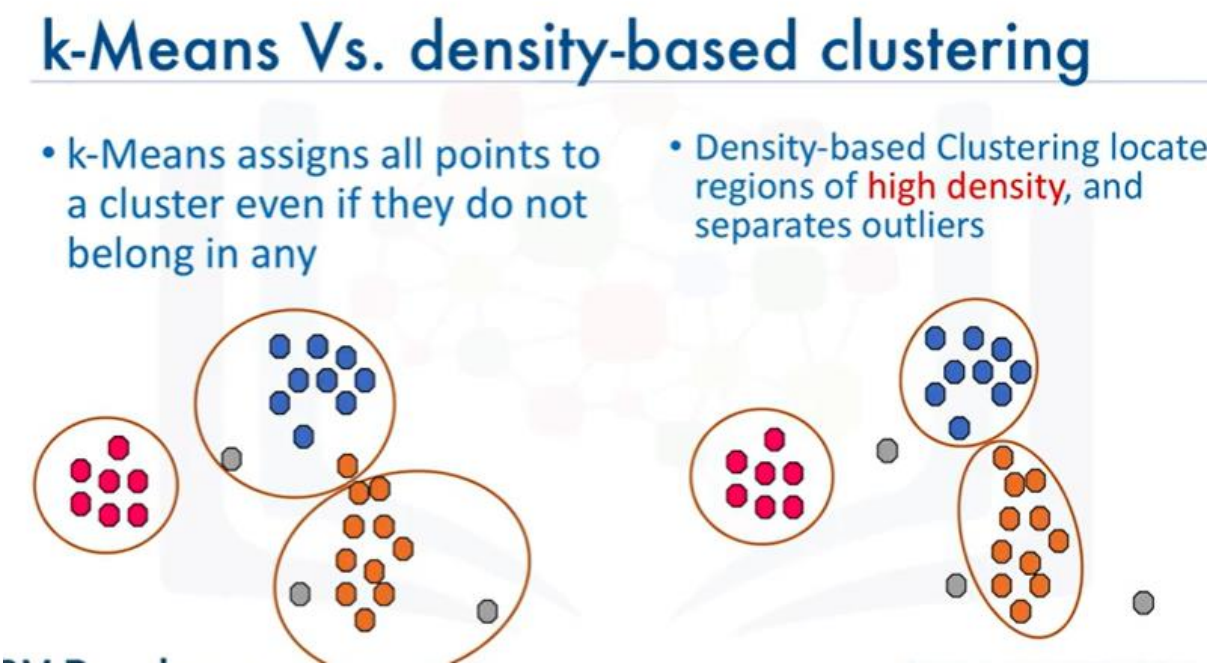
Advantages vs. disadvantages

Advantages	Disadvantages
Doesn't required number of clusters to be specified.	Can never undo any previous steps throughout the algorithm.
Easy to implement.	Generally has long runtimes.
Produces a dendrogram, which helps with understanding the data.	Sometimes difficult to identify the number of clusters by the dendrogram.

Hierarchical clustering Vs. *K*-means

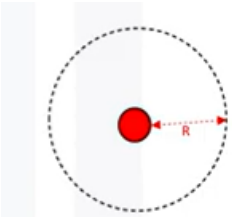
K-means	Hierarchical Clustering
1. Much more efficient	1. Can be slow for large datasets
2. Requires the number of clusters to be specified	2. Does not require the number of clusters to run
3. Gives only one partitioning of the data based on the predefined number of clusters	3. Gives more than one partitioning depending on the resolution
4. Potentially returns different clusters each time it is run due to random initialization of centroids	4. Always generates the same clusters

DBSCAN Clustering

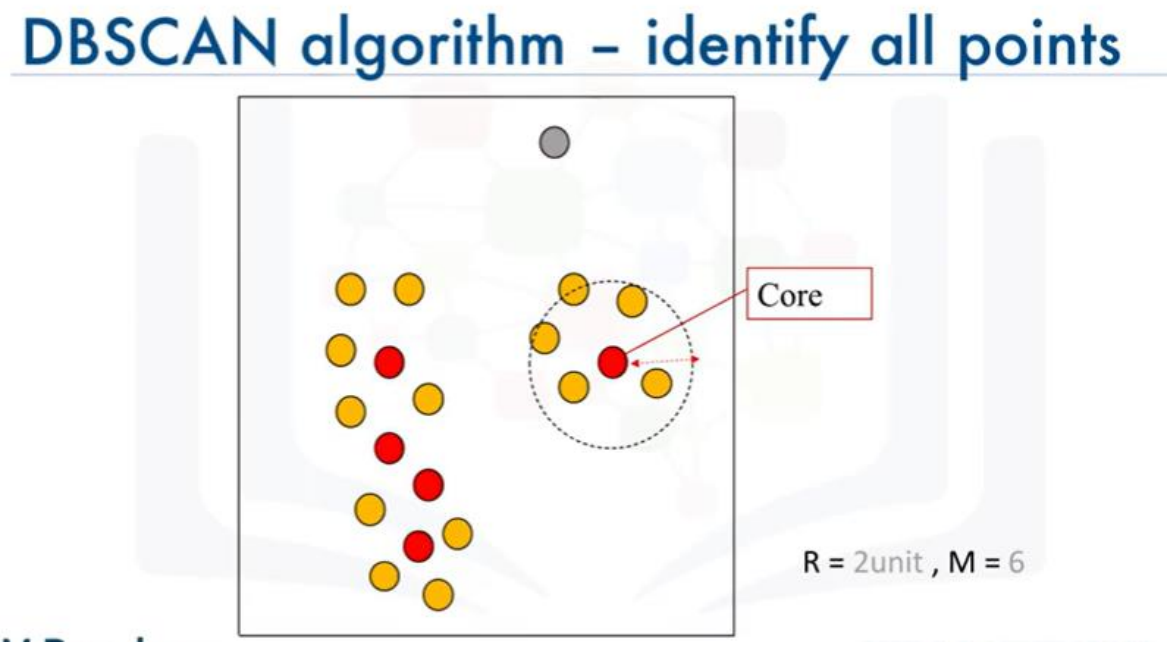


What is DBSCAN?

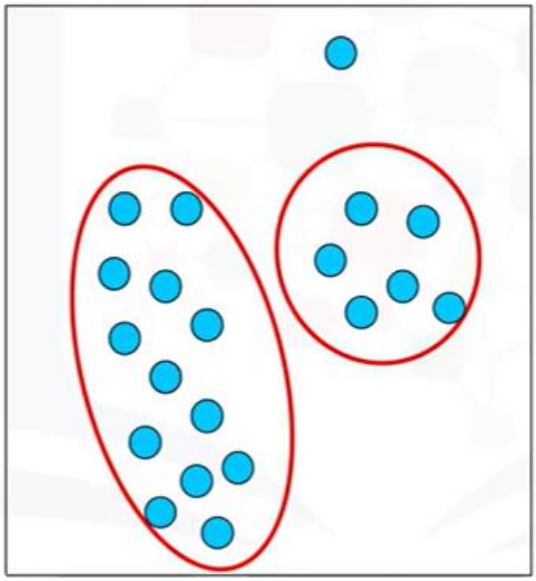
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
    - It is one of the most common clustering algorithms
    - Works based on density of objects
- R (Radius of neighbourhood)
    - Radius (R) that if includes enough number of points within, we call it a dense area



- M (Min number of neighbours)
  - The minimum number of data points we want in a neighbourhood to define a cluster



Advantages of DBSCAN



1. Arbitrarily shaped clusters
2. Robust to outliers
3. Does not require specification of the number of clusters