



UNIVERSITAS
INDONESIA
Veritas, Probatum, Justitia

KECERDASAN BUATAN

4 | CLUSTERING

Dr. Prima Dewi Purnamasari
Program Studi Teknik Komputer FTUI

Pendahuluan

Materi berikut diambil dari [Cognitivelass.ai](https://cognitivelass.ai)

Clustering

- Imagine that you have a customer dataset, and you need to apply customer segmentation on this historical data.
- Customer segmentation is the practice of partitioning a customer base into groups of individuals that have similar characteristics.

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

Clustering

Knowing this information allows a business to devote more time and attention to retaining these customers.

Another group might include customers from non-profit organizations, and so on.

A general segmentation process is not usually feasible for large volumes of varied data.

Therefore, you need an analytical approach to deriving segments and groups from large data sets.



It is a significant strategy as it allows a business to target specific groups of customers so as to more effectively allocate marketing resources.



For example, one group might contain customers who are high-profit and low-risk, that is, more likely to purchase products, or subscribe for a service.

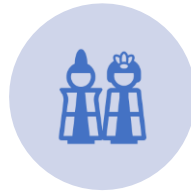
Clustering

One of the most adopted approaches that can be used for customer segmentation is clustering.

Clustering can group data only “unsupervised,” based on the similarity of customers to each other.

It will partition your customers into mutually exclusive groups, for example, into 3 clusters.

The customers in each cluster are similar to each other demographically.



Customers can be grouped based on several factors: including age, gender, interests, spending habits, and so on.



Let's learn how to divide a set of customers into categories, based on characteristics they share.

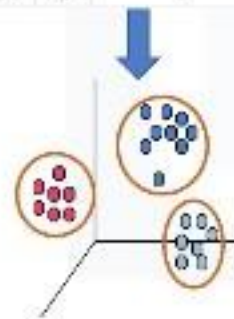


The important requirement is to use the available data to understand and identify how customers are similar to each other.

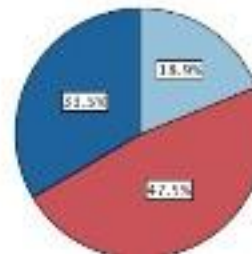
Clustering

Now we can create a profile for each group, considering the common characteristics of each cluster.

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1



Cluster Sizes



Cluster
cluster-1
cluster-2
cluster-3

Cluster	Segment Name
cluster-1	AFFLUENT AND MIDDLE AGED
cluster-2	YOUNG EDUCATED AND MIDDLE INCOME
cluster-3	YOUNG AND LOW INCOME

Clustering

- For example, the first group is made up of AFFLUENT AND MIDDLE AGED customers. The second is made up of YOUNG EDUCATED AND MIDDLE INCOME customers. And the third group includes YOUNG AND LOW INCOME customers.
- Finally, we can assign each individual in our dataset to one of these groups or segments of customers.

Customer ID	Segment
1	YOUNG AND LOW INCOME
2	AFFLUENT AND MIDDLE AGED
3	AFFLUENT AND MIDDLE AGED
4	YOUNG AND LOW INCOME
5	AFFLUENT AND MIDDLE AGED
6	AFFLUENT AND MIDDLE AGED
7	YOUNG AND LOW INCOME
8	YOUNG AND LOW INCOME
9	AFFLUENT AND MIDDLE AGED

Clustering

Customer segmentation is one of the popular usages of clustering.

Cluster analysis also has many other applications in different domains.

So let's first define clustering, and then we'll look at other applications.

Clustering means finding clusters in a dataset, unsupervised.



Now imagine that you cross-join this segmented dataset, with the dataset of the product or services that customers purchase from your company.



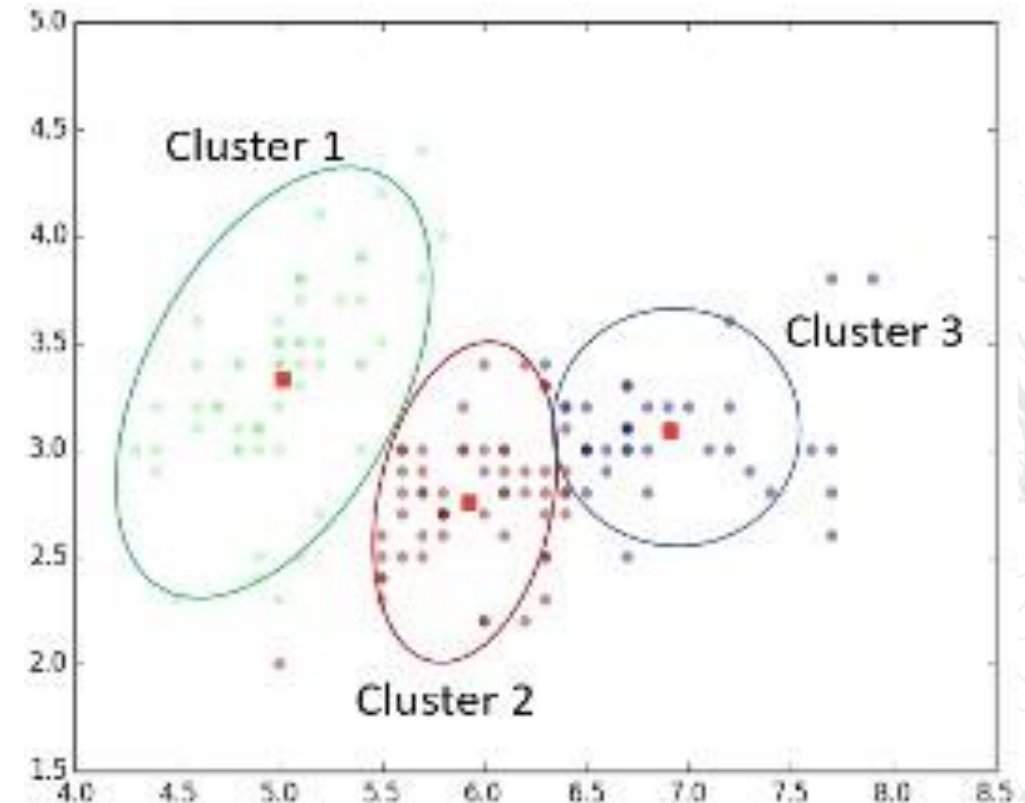
This information would really help to understand and predict the differences in individual customers' preferences and their buying behaviors across various products.



Indeed, having this information would allow your company to develop highly personalized experiences for each segment.

Clustering

- Clustering means **finding clusters** in a dataset, **unsupervised**.
- So, what is a cluster? A cluster is group of data points or objects in a dataset that are **similar to other objects in the group**, and dissimilar to data points in other clusters.



Clustering VS Classification

- Now, the question is, “What is different between clustering and classification?”
- Let’s look at our customer dataset again. Classification algorithms predict categorical class labels.

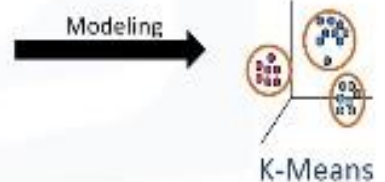
Labeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	5	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.581	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1
10	47	3	23	115	0.553	3.947	NBA011	4	



Unlabeled dataset

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	5	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.581	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1



Clustering VS Classification

- This means, assigning instances to pre-defined classes such as “Defaulted” or “Non-Defaulted.”
- For example, if an analyst wants to analyze customer data in order to know which customers might default on their payments, she uses a labeled dataset as training data, and uses classification approaches such as a decision tree, Support Vector Machines (or SVM), or, logistic regression to predict the default value for a new, or unknown customer.
- Generally speaking, classification is a supervised learning where each training data instance belongs to a particular class.
- In clustering, however, the data is unlabelled and the process is unsupervised.
- For example, we can use a clustering algorithm such as k-Means, to group similar customers as mentioned, and assign them to a cluster, based on whether they share similar attributes, such as age, education, and so on.

Clustering Applications

- In the Retail industry, clustering is used to find associations among customers based on their demographic characteristics and use that information to identify buying patterns of various customer groups.
 - Also, it can be used in recommendation systems to find a group of similar items or similar users, and use it for collaborative filtering, to recommend things like books or movies to customers.
- In Banking, analysts find clusters of normal transactions to find the patterns of fraudulent credit card usage.
 - Also, they use clustering to identify clusters of customers, for instance, to find loyal customers, versus churn customers.
- In the Insurance industry, clustering is used for fraud detection in claims analysis, or to evaluate the insurance risk of certain customers based on their segments.
- In Publication Media, clustering is used to auto-categorize news based on its content, or to tag news, then cluster it, so as to recommend similar news articles to readers.
- In Medicine: it can be used to characterize patient behavior, based on their similar characteristics, so as to identify successful medical therapies for different illnesses.
- In Biology: clustering is used to group genes with similar expression patterns, or to cluster genetic markers to identify family ties.

Why Clustering?

- If you look around, you can find many other applications of clustering, but generally, clustering can be used for one of the following purposes:
 - exploratory data analysis,
 - summary generation or reducing the scale,
 - outlier detection, especially to be used for fraud detection, or noise removal, finding duplicates in datasets,
 - pre-processing step for either prediction, other data mining tasks, or, as part of a complex system.
- Let's briefly look at different clustering algorithms and their characteristics.
- Partitioned-based clustering is a group of clustering algorithms that produces sphere-like clusters, such as k-Means, k-Median, or Fuzzy c-Means.
- These algorithms are relatively efficient and are used for Medium and Large sized databases.
- Hierarchical clustering algorithms produce trees of clusters, such as Agglomerative and Divisive algorithms.

Why Clustering?



This group of algorithms are very intuitive and are generally good for use with small size datasets.



Density based clustering algorithms produce arbitrary shaped clusters.



They are especially good when dealing with spatial clusters or when there is noise in your dataset, for example, the DBSCAN algorithm.

Hierarchical clustering algorithms produce trees of clusters, such as Agglomerative and Divisive algorithms.



UNIVERSITAS
INDONESIA

Veritas, Probatum, Justitia

K-Means

K-Means Clustering

- Customer segmentation is the practice of partitioning a customer base into groups of individuals that have similar characteristics.
- One of the algorithms that can be used for customer segmentation is k-Means clustering.
- k-Means can group data only “unsupervised,” based on the similarity of customers to each other. Let’s define this technique more formally.
- Imagine that you have a customer dataset, and you need to apply customer segmentation on this historical data.

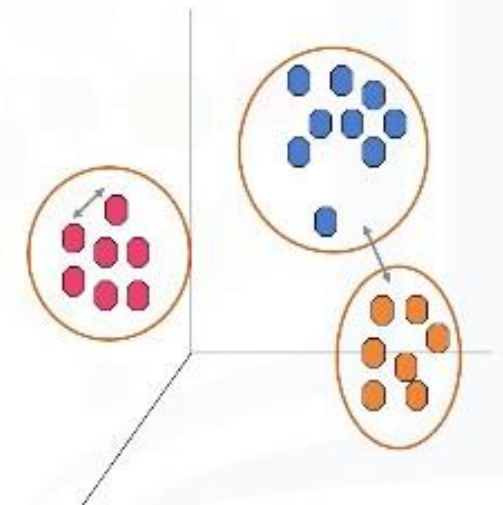
Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio	Defaulted
1	41	2	6	19	0.124	1.073	NBA001	6.3	0
2	47	1	26	100	4.582	8.218	NBA021	12.8	0
3	33	2	10	57	6.111	5.802	NBA013	20.9	1
4	29	2	4	19	0.681	0.516	NBA009	6.3	0
5	47	1	31	253	9.308	8.908	NBA008	7.2	0
6	40	1	23	81	0.998	7.831	NBA016	10.9	1
7	38	2	4	56	0.442	0.454	NBA013	1.6	0
8	42	3	0	64	0.279	3.945	NBA009	6.6	0
9	26	1	5	18	0.575	2.215	NBA006	15.5	1

K-Means Clustering

- There are various types of clustering algorithms, such as partitioning, hierarchical, or density-based clustering.
- k-Means is a type of partitioning clustering, that is, it divides the data into k non-overlapping subsets (or clusters) without any cluster-internal structure, or labels.
- This means, it's an unsupervised learning algorithm.
- Objects within a cluster are very different, but objects in different clusters are very similar (for using k-Means, we have 10 similar customers).

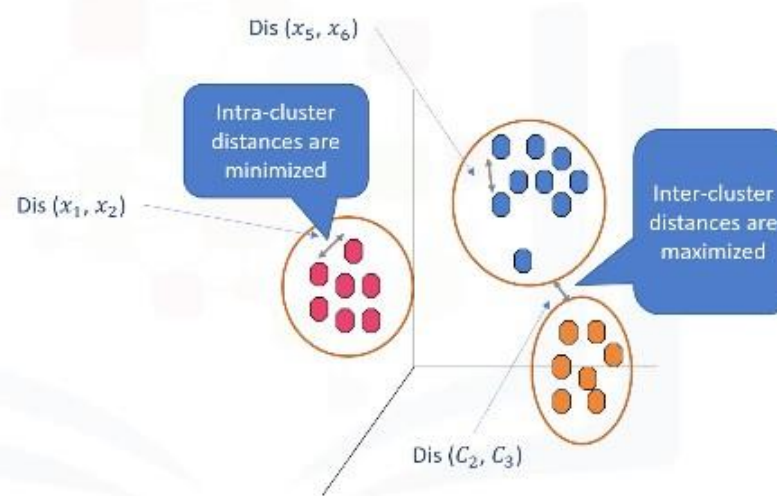
k-Means algorithms

- 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



K-Means Algorithms

- Now we face a couple of key questions.
- First, “How can we find the similarity of samples in clustering?”
- And then, “How do we measure how similar two customers are with regard to their demographics?”
- Though the objective of k-Means is to form clusters in such a way that similar samples go into a cluster, and dissimilar samples fall into different clusters, it can be said that k-Means tries to minimize the intra-cluster distances and maximize the inter-cluster distances.
- In other words, conventionally, the distance metric used to shape the clusters is the Euclidean distance.
- So, we can say, k-Means tries to minimize the intra-cluster distances and maximize the inter-cluster distances.



K-Means Algorithms

Now, the question is, "How we can calculate the dissimilarity or distance of two cases, such as two customers?" Assume that we have two customers, we'll call them customer 1 and 2.



Let's also assume that we have only one feature for each of these two customers, and that feature is Age. We can easily use a specific type of Minkowski distance to calculate the distance of these two customers.

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

K-Means Alogarithms

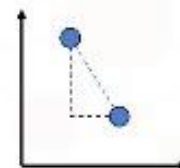
- Indeed, it is the Euclidian distance. Distance of x_1 from x_2 :

$$\text{Dis}(x_1, x_2) = \sqrt{(34 - 30)^2} = 4$$

- How about 2 feature? For example, if we have income and age for each customer, we can still use the same formula, but this time in a 2-dimensional space.



Customer 1	
Age	Income
54	190



Customer 2	
Age	Income
50	200

$$\begin{aligned} \text{Dis}(x_1, x_2) &= \sqrt{\sum_{i=0}^n (x_{1i} - x_{2i})^2} \\ &= \sqrt{(54 - 50)^2 + (190 - 200)^2} = 10.77 \end{aligned}$$

K-Means Alogarithms

- Also, we can use the same distance matrix for multi-dimensional vectors.
- Of course, we have to normalize our feature set to get the accurate dissimilarity measure.

Customer 1			Customer 2		
Age	Income	education	Age	Income	education
54	190	3	50	200	8

$$\begin{aligned}\text{Dis}(x_1, x_2) &= \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \\ &= \sqrt{(54 - 50)^2 + (190 - 200)^2 + (3 - 8)^2} = 11.87\end{aligned}$$

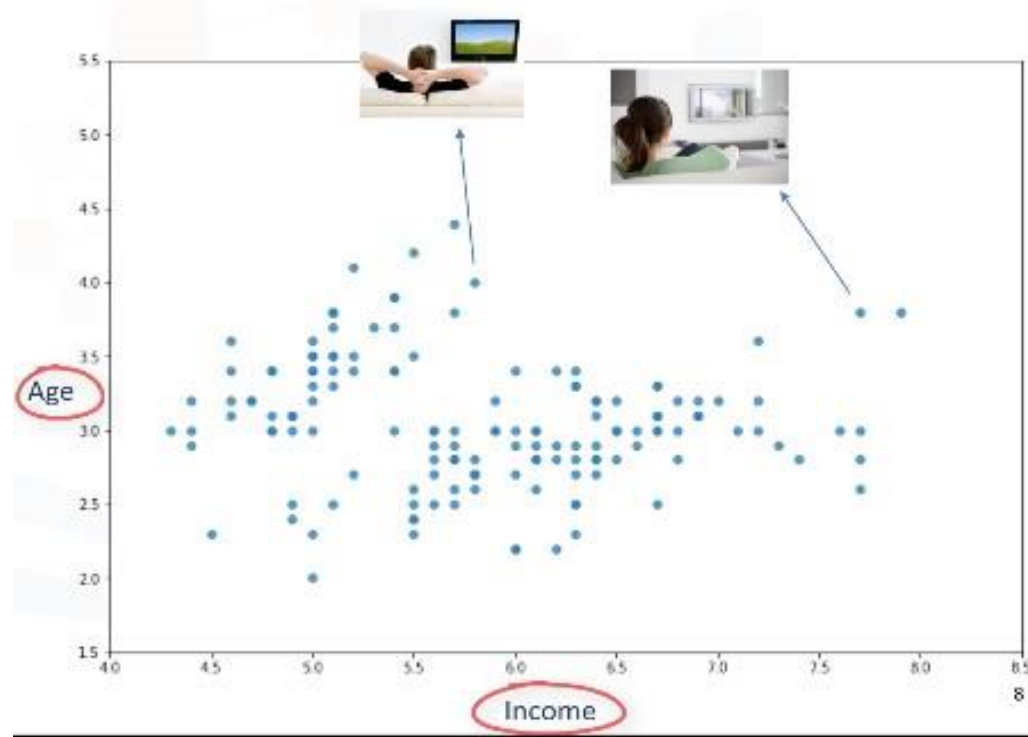
K-Means Algorithms

- There are other dissimilarity measures as well that can be used for this purpose, but it is highly dependent on data type and also the domain that clustering is done for it.
- For example, you may use Euclidean distance, cosine similarity, average distance, and so on.
- Indeed, the similarity measure highly controls how the clusters are formed, so it is recommended to understand the domain knowledge of your dataset, and data type of features, and then choose the meaningful distance measurement.
- Now, let's see how k-Means clustering works. For the sake of simplicity, let's assume that our dataset has only two features, the age and income of customers.

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

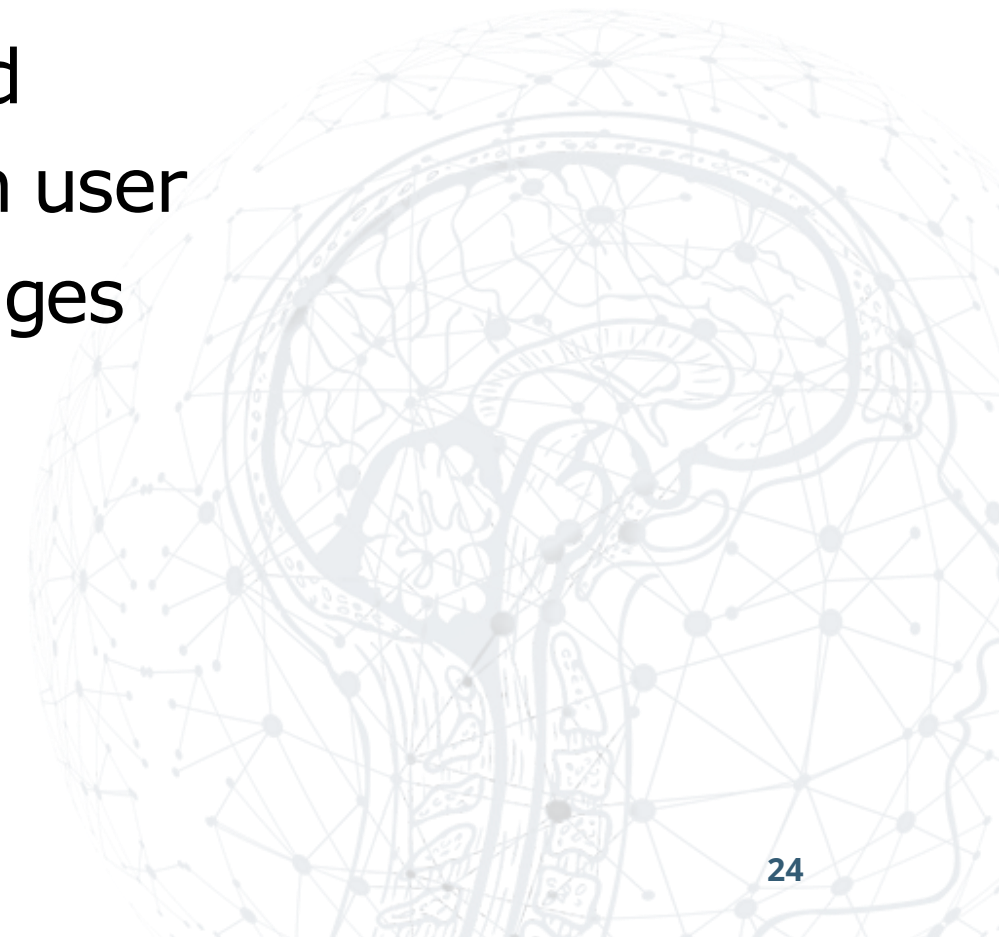
K-Means Alogarithms

- This means, it's a 2-dimentional space. We can show the distribution of customers using a scatterplot. The y- axes indicates Age and the x-axes shows Income of customers.



Pseudocode of KMeans

1. Initialize $K=3$ centroids randomly
2. Distance Calculation
3. Assign each point to closest centroid
4. Compute the new centroids for each user
5. Repeat until there are no more changes



Initialize $K=3$ Centroids Randomly



We try to cluster the customer dataset into distinct groups (or clusters) based on these two dimensions.



In the first step, we should determine the number of clusters.



The key concept of the k-Means algorithm is that it randomly picks a center point for each cluster.



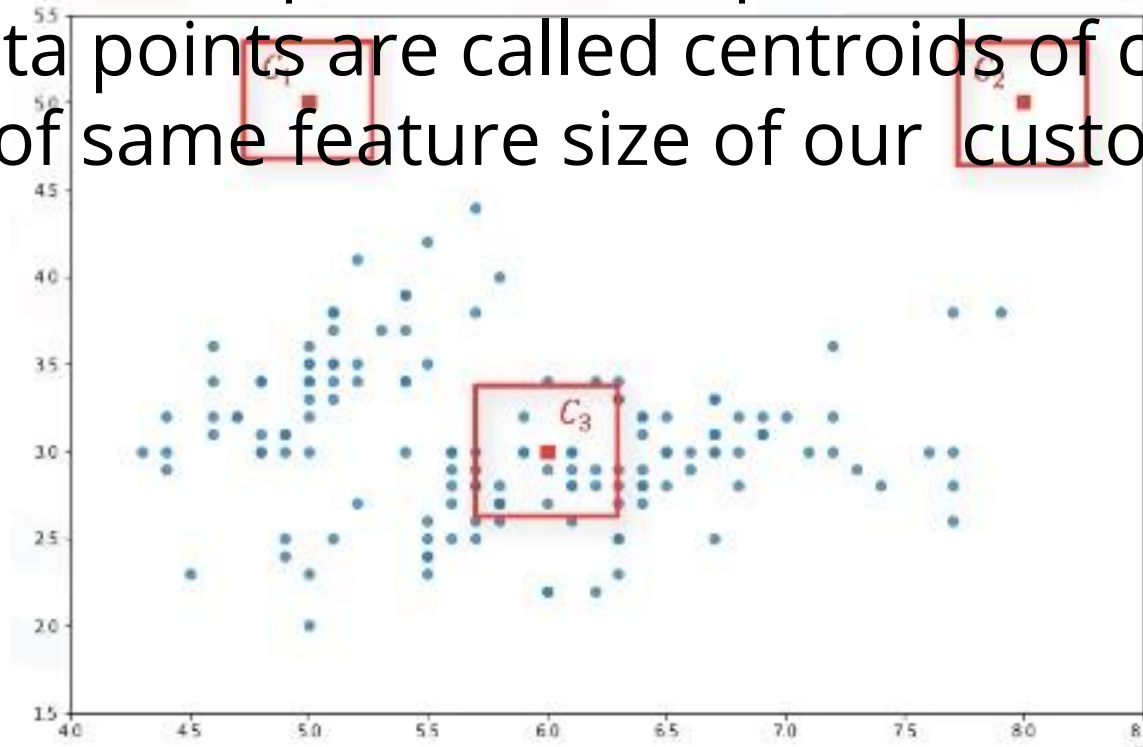
It means, we must initialize k , which represents "number of clusters."



Essentially, determining the number of clusters in a data set, or k , is a hard problem in k-Means that we will discuss later.

Initialize $K=3$ Centroids Randomly

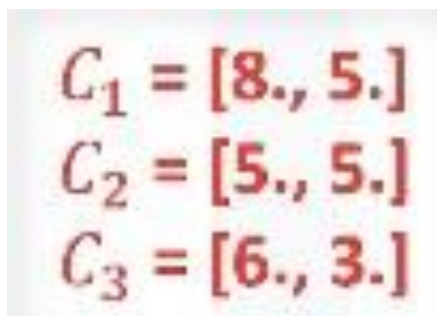
- For now, let's put k equals 3 here, for our sample dataset. It is like we have 3 representative points for our clusters. These 3 data points are called centroids of clusters, and should be of same feature size of our customer feature set.



Initialize K=3 Centroids Randomly

There are two approaches to choose these centroids:

- 1) We can randomly choose 3 observations out of the dataset and use these observations as the initial means. Or,
- 2) We can create 3 random points as centroids of the clusters, which is our choice that is shown in this plot with red color.


$$\begin{aligned}C_1 &= [8., 5.] \\C_2 &= [5., 5.] \\C_3 &= [6., 3.]\end{aligned}$$

After the initialization step, which was defining the centroid of each cluster, we have to assign each customer to the closest center.

For this purpose, we have to calculate the distance of each data point (or in our case, each customer) from the centroid points.

As mentioned before, depending on the nature of the data and the purpose for which clustering is being used, different measures of distance may be used to place items into clusters.



Distance Calculation

Therefore, you will form a matrix where each row represents the distance of a customer from each centroid. It is called the "distance-matrix."

C_1	C_2	C_3
$d(p_1, c_1)$	$d(p_1, c_2)$	$d(p_1, c_3)$
$d(p_2, c_1)$	$d(p_2, c_2)$	$d(p_2, c_3)$
$d(p_3, c_1)$	$d(p_3, c_2)$	$d(p_3, c_3)$
$d(p_4, c_1)$	$d(p_4, c_2)$	$d(p_4, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_n, c_1)$	$d(p_n, c_2)$	$d(p_n, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_n, c_1)$	$d(p_n, c_2)$	$d(p_n, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_{...}, c_1)$	$d(p_{...}, c_2)$	$d(p_{...}, c_3)$
$d(p_n, c_1)$	$d(p_n, c_2)$	$d(p_n, c_3)$

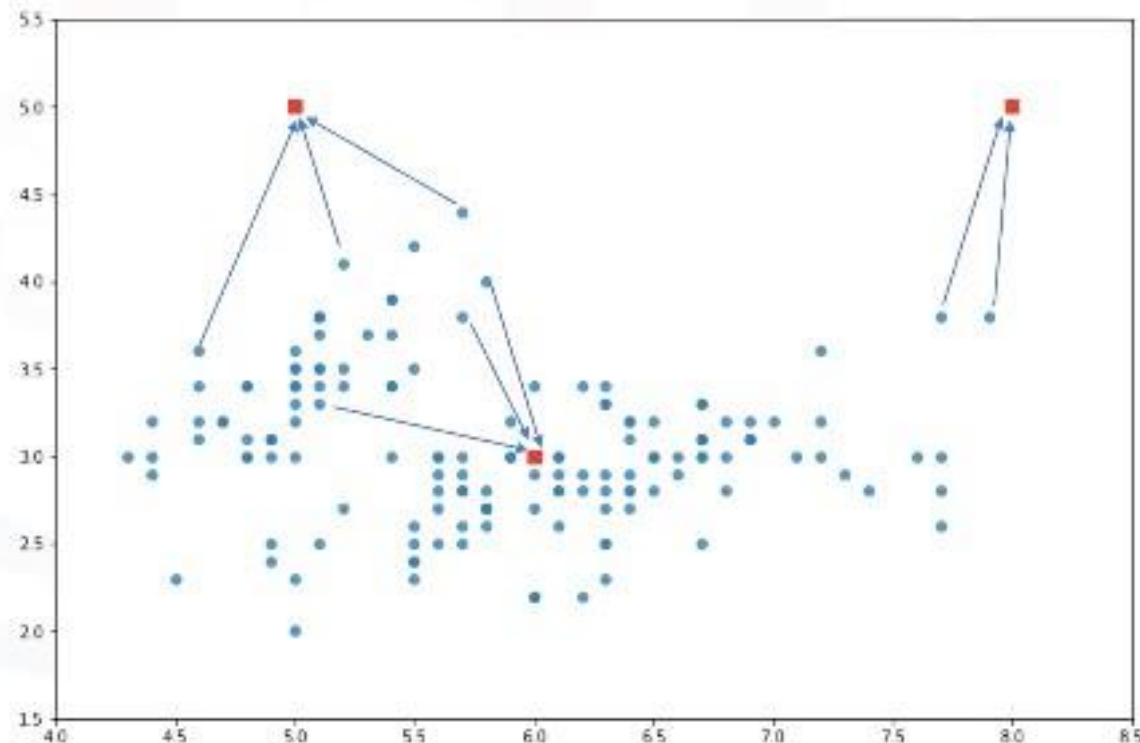
Assign each point to the closest centroid

The main objective of k-Means clustering is to minimize the distance of data points from the centroid of its cluster and maximize the distance from other cluster centroids.

So, in this step we have to find the closest centroid to each data point.

We can use the distance-matrix to find the nearest centroid to data points.

Finding the closest centroids for each data point, we assign each data point to that cluster.



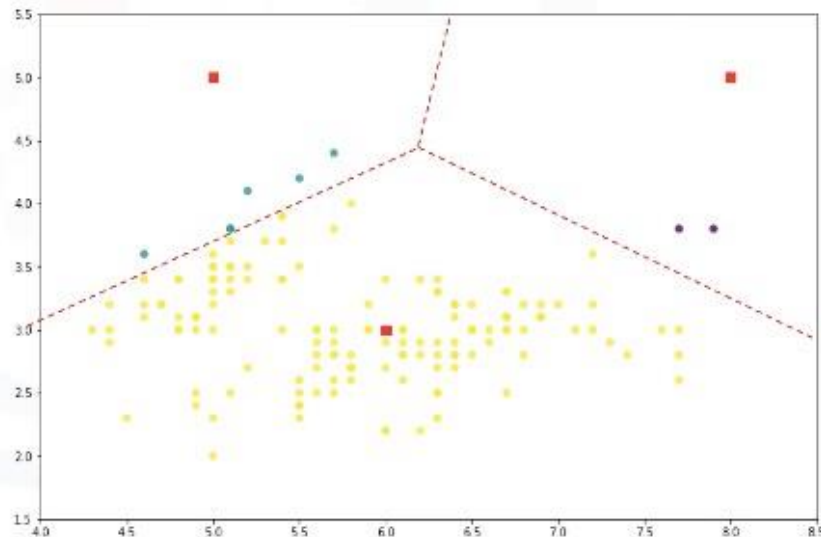
Compute the new centroid for each cluster

In other words, all the customers will fall to a cluster, based on their distance from centroids.

We can easily say that it does not result in good clusters, because the centroids were chosen randomly from the first.

Indeed, the model would have a high error.

Here, error is the total distance of each point from its centroid. It can be shown as within-cluster sum of squares error. Intuitively, we try to reduce this error.



SSE = the sum of the squared differences between each point and its centroid.

$$SSE = \sum_1^n (x_i - c_j)^2$$

Compute the new centroid for each cluster

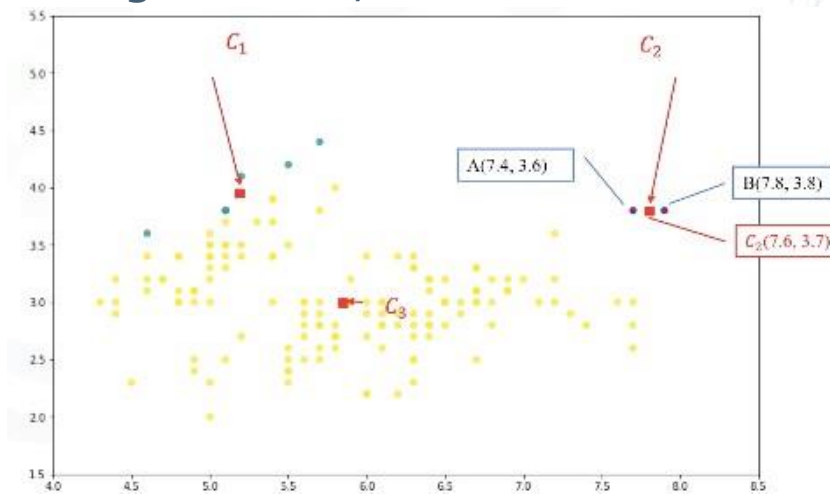
It means we should shape clusters in such a way that the total distance of all members of a cluster from its centroid be minimized.

Now, the question is, "How we can turn it into better clusters, with less error?"

Okay, we move centroids. In the next step, each cluster center will be updated to be the mean for data points in its cluster.

Indeed, each centroid moves according to their cluster members. In other words, the centroid of each of the 3 clusters becomes the new mean.

For example, if Point A coordination is 7.4 and 3.6, and Point B features are 7.8 and 3.8, the new centroid of this cluster with 2 points, would be the average of them, which is 7.6 and 3.7.



Compute the new centroid for each cluster

- Now we have new centroids. As you can guess, once again, we will have to calculate the distance of all points from the new centroids.
- The points are re-clustered and the centroids move again. This continues until the centroids no longer move.
- Please note that whenever a centroid moves, each point's distance to the centroid needs to be measured again.

Repeat until there are no more changes



Yes, k-Means is an iterative algorithm, and we have to repeat steps 2 to 4 until the algorithm converges.



In each iteration, it will move the centroids, calculate the distances from new centroids, and assign the data points to the nearest centroid.



It results in the clusters with minimum error, or the most dense clusters.



However, as it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters.

It means this algorithm is guaranteed to converge to a result, but the result may be a local optimum (i.e. not necessarily the best possible outcome).

To solve this problem, it is common to run the whole process, multiple times, with different starting conditions.

This means, with randomized starting centroids, it may give a better outcome.

And as the algorithm is usually very fast, it wouldn't be any problem to run it multiple times.

More on K-Means

Let's define the algorithm more concretely before we talk about its accuracy.

Please note, however, that you can also use different types of distance measurements, not just Euclidean distance.

Euclidean distance is used because it's the most popular.

Then, assign each data point (or object) to its closest centroid, creating a group.

Next, once each data point has been classified to a group, recalculate the position of the k centroids. The new centroid position is determined by the mean of all points in the group.

Finally, this continues until the centroids no longer move.

A k -Means algorithm works by randomly placing k centroids, one for each cluster.

The farther apart the clusters are placed, the better.

The next step is to calculate the distance of each data point (or object) from the centroids.

Euclidean distance is used to measure the distance from the object to the centroid.

Pseudocode

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.

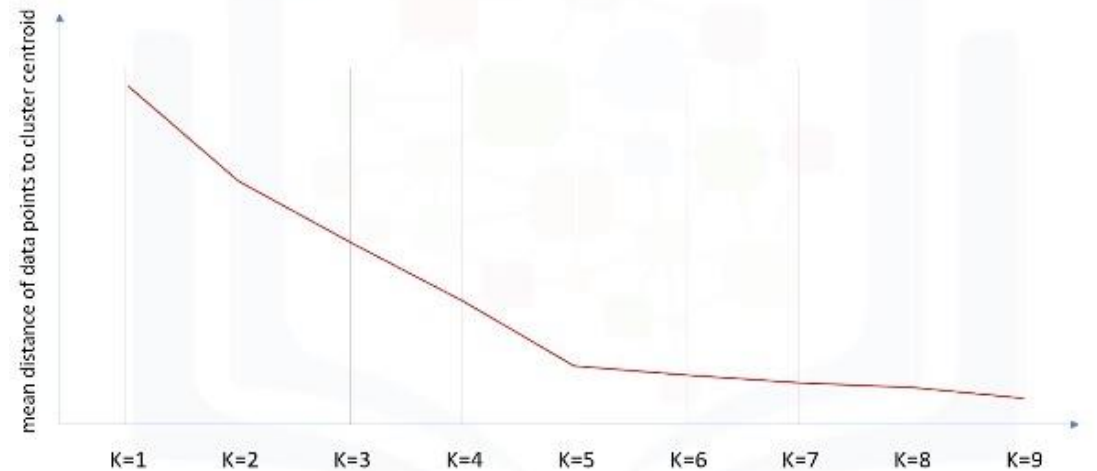
Accuracy of K-Means

- "How can we evaluate the 'goodness' of the clusters formed by k-Means?"
- "How do we calculate the accuracy of k-Means clustering?"
 1. Compare the clusters with the ground truth, if it's available.
 2. However, because k-Means is a unsupervised algorithm, we usually don't have ground truth in real world problems to be used. But, there is still a way to say how bad each cluster is using:
 1. **average distance between data points within a cluster**
 2. **average of the distances of data points from their cluster centroids**

K-Means Accuracy

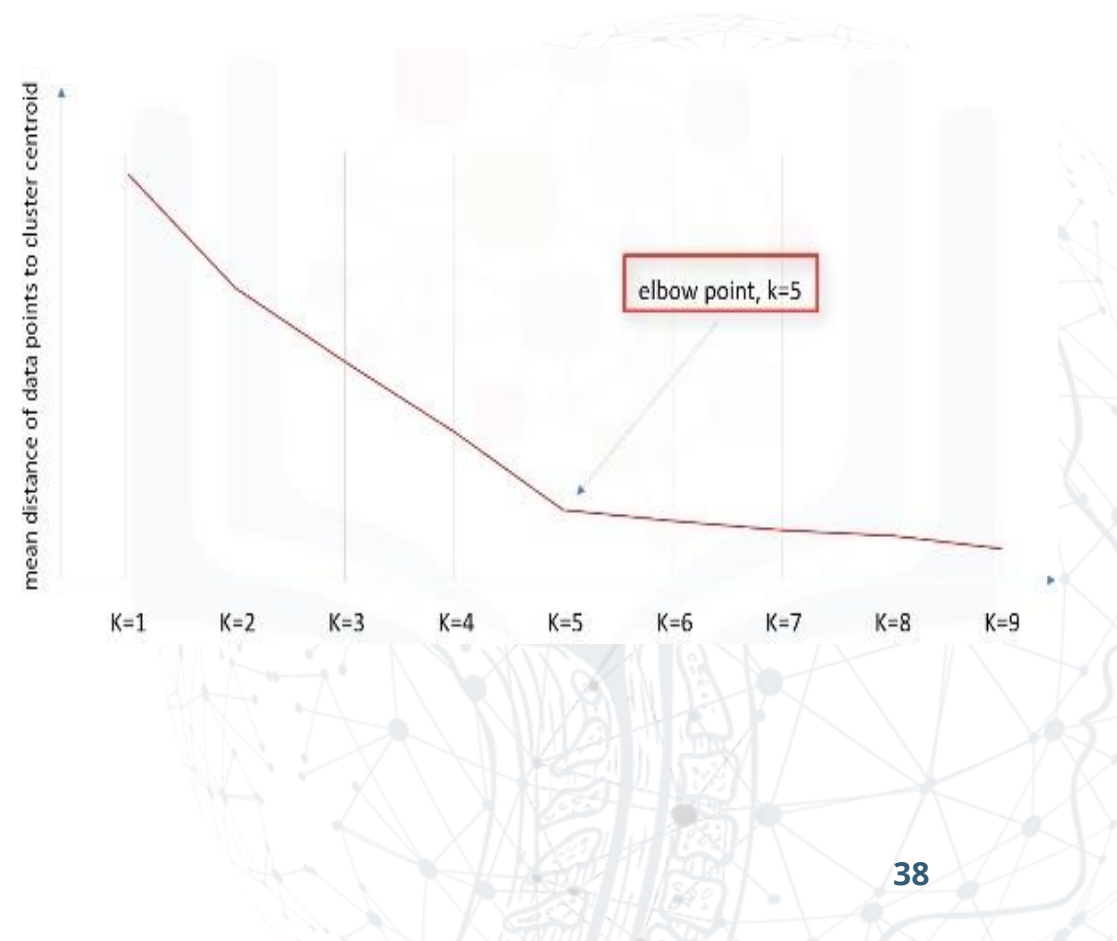
- Run the clustering across the different values of K, and looking at a metric of accuracy for clustering.
- Metric → **mean distance between data points and their cluster centroid**, which indicate how dense our clusters are, or to what extent we minimized the error of clustering.
- Find the best value for k.

Choosing k



Elbow Method

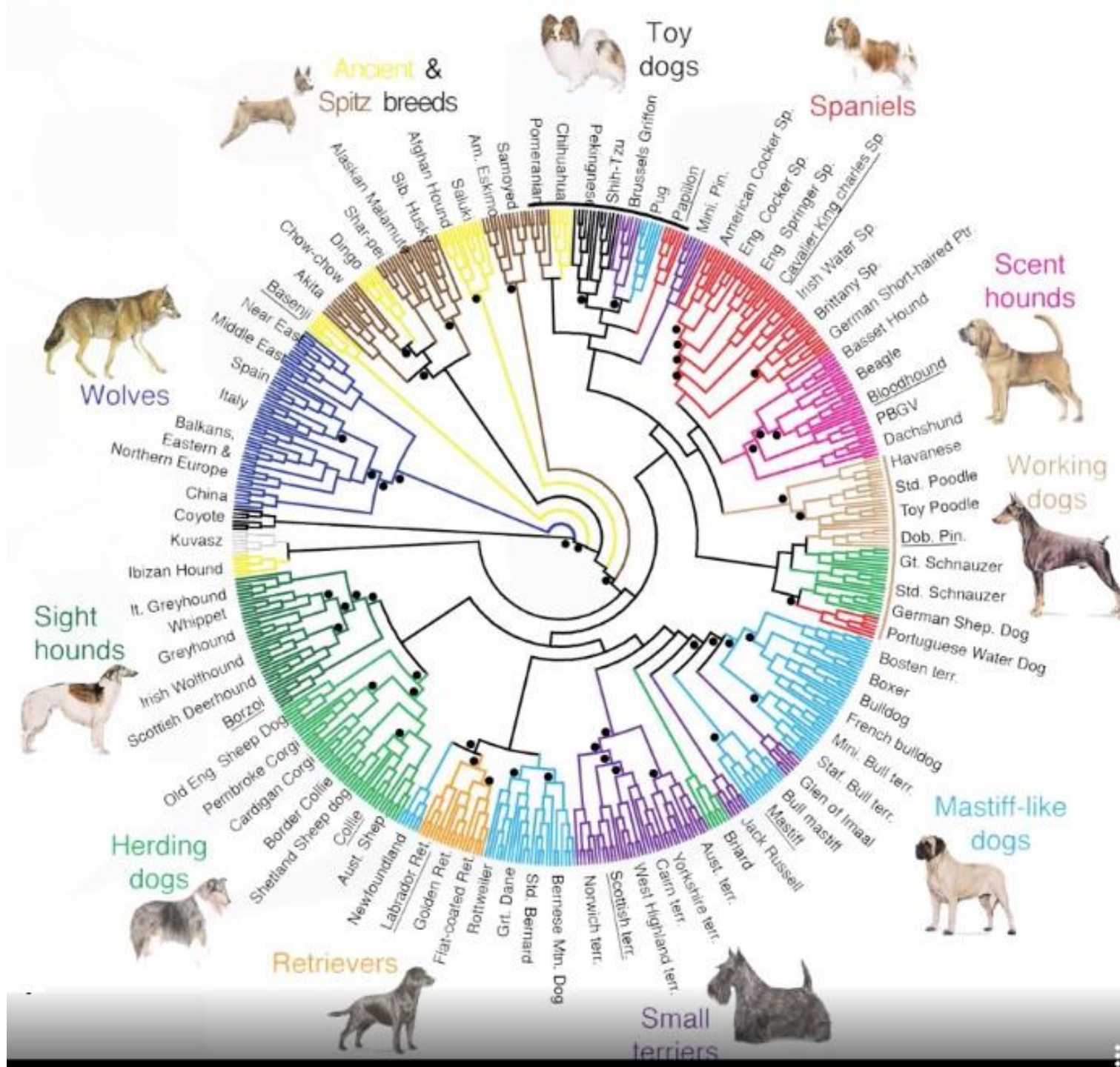
- Problem: with increasing the number of clusters, the distance of centroids to data points will always reduce.
- This means, increasing K will always decrease the "error."
- So, the value of the metric as a function of K is plotted and the "**elbow point**" is determined, where the rate of decrease sharply shifts.



K-Means

- So, let's recap k-Means clustering: k-Means is a partitioned-based clustering, which is:
 - Relatively efficient on medium and large sized datasets;
 - Produces sphere-like clusters, because the clusters are shaped around the centroids;
 - Its drawback is that we should pre-specify the number of clusters, and this is not an easy task.
- Let's look at this chart. An international team of scientists, led by UCLA biologists, used this dendrogram to report genetic data from more than 900 dogs from 85 breeds -- and more than 200 wild gray wolves worldwide, including populations from North America, Europe, the Middle East, and East Asia.
- They used molecular genetic techniques to analyze more than 48,000 genetic markers.

Hierarchical Clustering



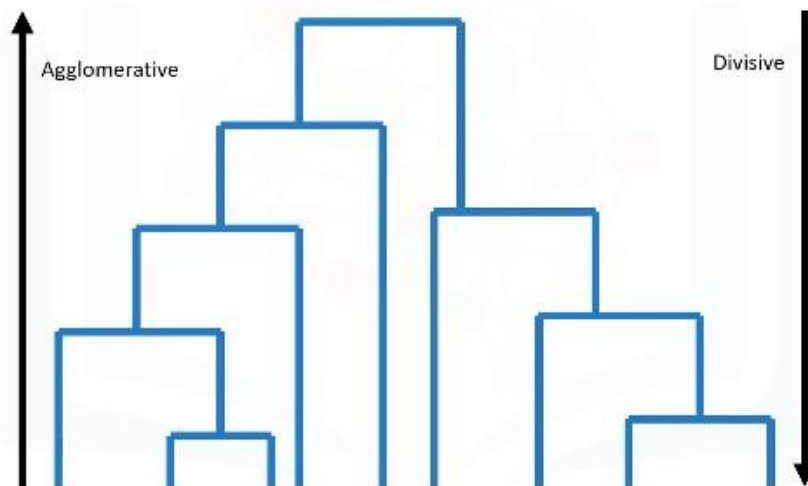
Hierarchical Clustering

This diagram, shows hierarchical clustering of these animals based on the similarity in their genetic data.

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

Strategies for hierarchical clustering generally fall into two types: Divisive and Agglomerative.

Hierarchical clustering



Type

- **Divisive** is top-down, so you start with all observations in a large cluster and break it down into smaller pieces. Think about divisive as "dividing" the cluster.
- **Agglomerative** is the opposite of divisive, so it is bottom-up, where each observation starts in its own cluster and pairs of clusters are merged together as they move up the hierarchy.
 - Agglomeration means to amass or collect things, which is exactly what this does with the cluster.

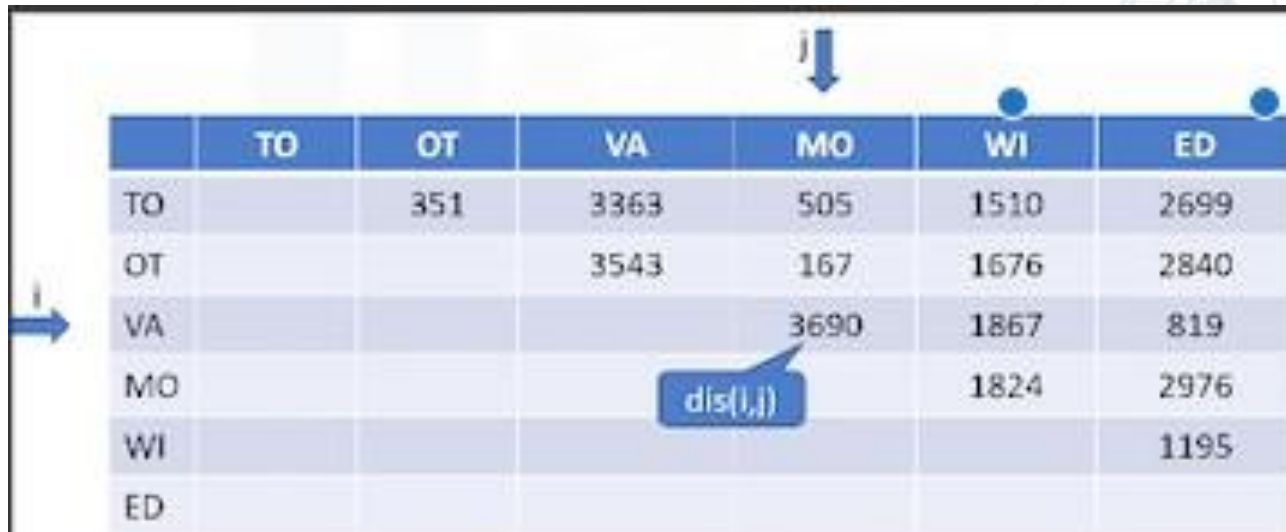
Hierarchical Clustering

- The Agglomerative approach is more popular among data scientists and so it is the main subject. Let's look at a sample of Agglomerative clustering.
- This method builds the hierarchy from the individual elements by progressively merging clusters.
- In our example, let's say we want to cluster 6 cities in Canada based on their distances from one another. They are: Toronto, Ottawa, Vancouver, Montreal, Winnipeg, and Edmonton



Hierarchical Clustering

- We construct a distance matrix at this stage, where the numbers in the row i column j is the distance between the i and j cities.
- In fact, this table shows the distances between each pair of cities.



	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						

$dis(i,j)$

Hierarchical Clustering

The algorithm is started by assigning each city to its own cluster. So, if we have 6 cities, we have 6 clusters, each containing just one city.

Let's note each city by showing the first two characters of its name.

TC OT MO VA ED WI						
	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						


The first step is to determine which cities -- let's call them clusters from now on -- to merge into a cluster.

Usually, we want to take the two closest clusters according to the chosen distance. Looking at the distance matrix, Montreal and Ottawa are the closest clusters.

So, we make a cluster out of them.

Hierarchical Clustering


- Please notice that we just use a simple 1-dimensional distance feature here, but our object can be multi-dimensional, and distance measurement can be either Euclidean, Pearson, average distance, or many others, depending on data type and domain knowledge.



	TO	OT	VA	MO	WI	ED
TO		351	3363	505	1510	2699
OT			3543	167	1676	2840
VA				3690	1867	819
MO					1824	2976
WI						1195
ED						

Hierarchical Clustering


- Anyhow, we have to merge these two closest cities in the distance matrix as well. So, rows and columns are merged as the cluster is constructed.
- As you can see in the distance matrix, rows and columns related to Montreal and Ottawa cities are merged as the cluster is constructed.



	TO	OT	MO	VA	ED	WI
	TO	OT/MO	VA	WI	ED	
TO		351	3363	1510	2699	
OT/MO			3543	1676	2840	
VA				1867	819	
WI					1195	
ED						

Hierarchical Clustering


- Then, the distances from all cities to this new merged cluster get updated. But how?
- For example, how do we calculate the distance from Winnipeg to the Ottawa-Montreal cluster?
- Well, there are different approaches, but let's assume, for example, we just select the distance from the centre of the Ottawa-Montreal cluster to Winnipeg.



	TO	OT	MO	VA	ED	WI
	TO	OT/MO	VA	WI	ED	
TO		351	3363	1510	2699	
OT/MO			3543	1676	2840	
VA				1867	819	
WI					1195	
ED						

Hierarchical Clustering

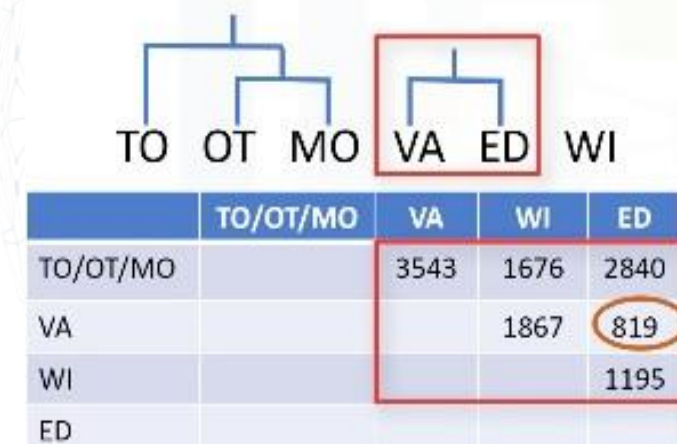
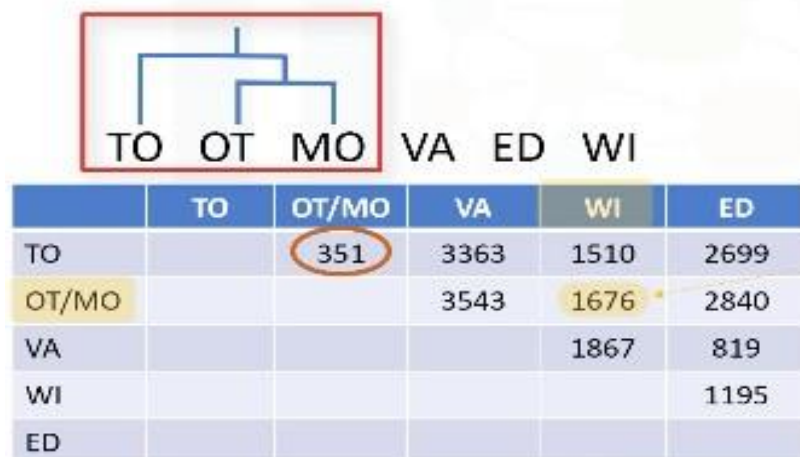
- Updating the distance matrix, we now have one less cluster.



	TO	OT	MO	VA	ED	WI
	TO	OT/MO	VA	WI	ED	
TO		351	3363	1510	2699	
OT/MO			3543	1676	2840	
VA				1867	819	
WI					1195	
ED						


Hierarchical Clustering

- Next, we look for the closest clusters once again.
- In this case, Ottawa-Montreal and Toronto are the closest ones, which creates another cluster.
- In the next step, the closest distance is between the Vancouver cluster and the Edmonton cluster. Forming a new cluster, their data in the matrix table gets updated.

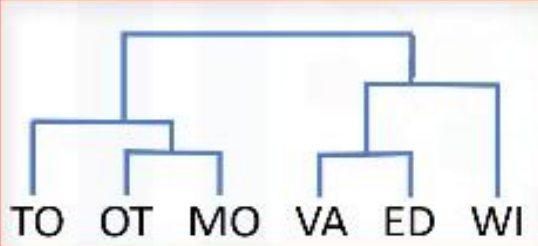


Hierarchical Clustering

- Essentially, the rows and columns are merged as the clusters are merged and the distance updated.
- This is a common way to implement this type of clustering, and has the benefit of caching distances between clusters.
- In the same way, agglomerative algorithm proceeds by merging clusters.
- And we repeat it until all clusters are merged and the tree becomes completed. It means, until all cities are clustered into a single cluster of size 6.



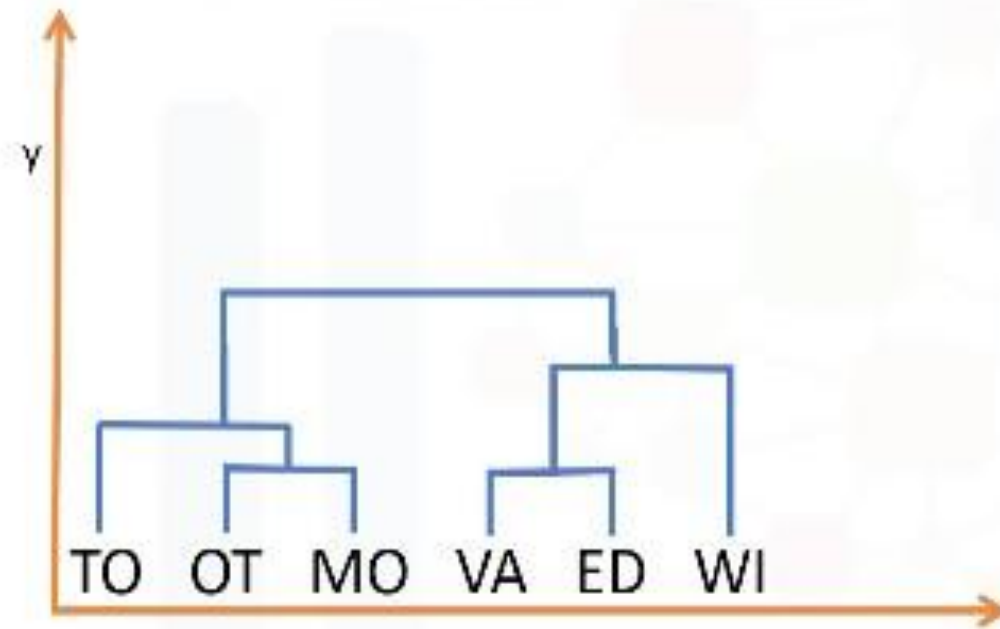
	TO/OT/MO	VA	WI	ED
TO/OT/MO		3543	1676	2840
VA			1867	819
WI				1195
ED				



	TO/OT/MO	VA/ED/WI
TO/OT/MO		1676
VA/ED/WI		

Hierarchical Clustering

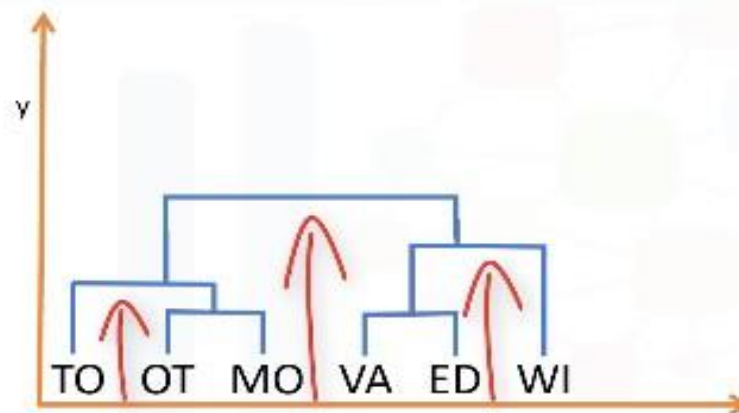
- Hierarchical clustering is typically visualized as a dendrogram as shown on this slide.



Dendrogram

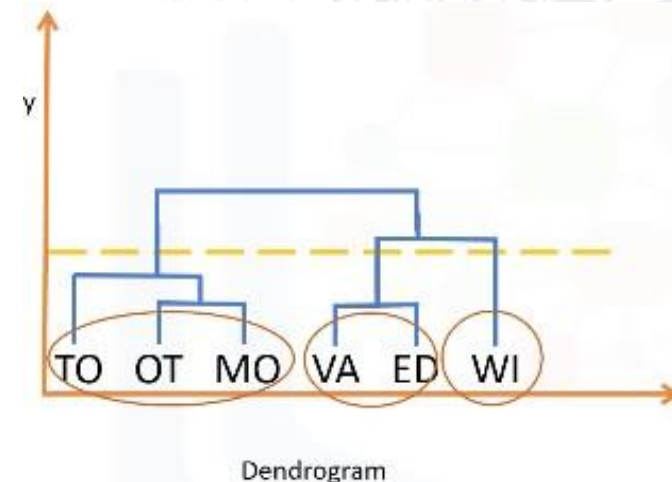
Hierarchical Clustering

- Each merge is represented by a horizontal line.
- The y-coordinate of the horizontal line is the similarity of the two clusters that were merged, where cities are viewed as singleton clusters.
- By moving up from the bottom layer to the top node, a dendrogram allows us to reconstruct the history of merges that resulted in the depicted clustering.



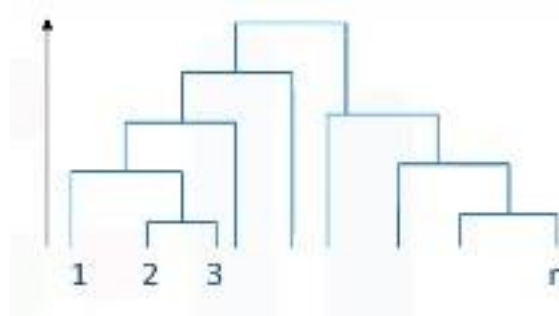
Hierarchical Clustering

- Essentially, Hierarchical clustering does not require a pre-specified number of clusters. However, in some applications we want a partition of disjoint clusters just as in flat clustering. In those cases, the hierarchy needs to be cut at some point.
- For example here, cutting in a specific level of similarity, we create 3 clusters of similar cities.



More on Hierarchical Clustering

Let's look at Agglomerative algorithm for Hierarchical Clustering.
Remember that Agglomerative clustering is a bottom-up approach.
Let's say our dataset has n data points.



First, we want to create n clusters, one for each data point. Then each point is assigned as a cluster.
Next, we want to compute the distance/proximity matrix, which will be an n by n table.

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

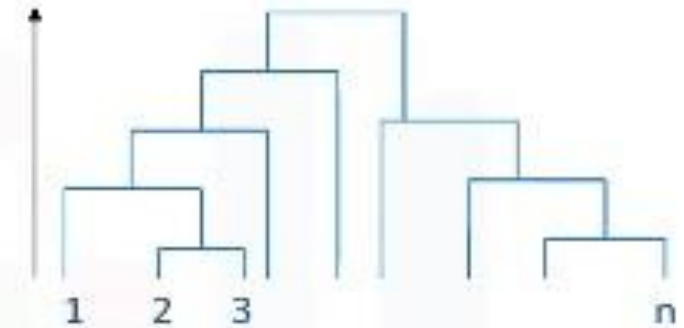
More on Hierarchical Clustering

- After that, we want to iteratively run the following steps until the specified cluster number is reached, or until there is only one cluster left.
- First, MERGE the two nearest clusters. (Distances are computed already in the proximity matrix.) Second, UPDATE the proximity matrix with the new values.
- We stop after we've reached the specified number of clusters, or there is only one cluster remaining, with the result stored in a dendrogram.
- So, in the proximity matrix, we have to measure the distances between clusters, and also merge the clusters that are "nearest."

Pseudocode

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



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

More on Hierarchical Clustering

- So, the key operation is the computation of the proximity between the clusters with one point, and also clusters with multiple data points.
- At this point, there are a number of key questions that need to be answered. For instance,
 - “How do we measure the distances between these clusters and How do we define the ‘nearest’ among clusters?”
 - We also can ask, “Which points do we use?”

More on Hierarchical Clustering

First, let's see how to calculate the distance between 2 clusters with 1 point each.

Let's assume that we have a dataset of patients, and we want to cluster them using hierarchy clustering.

So, our data points are patients, with a feature set of 3 dimensions.



Patient 1		
Age	BMI	BP
54	190	120



Patient 2		
Age	BMI	BP
50	200	125

Dis (p1,p2)

More on Hierarchical Clustering

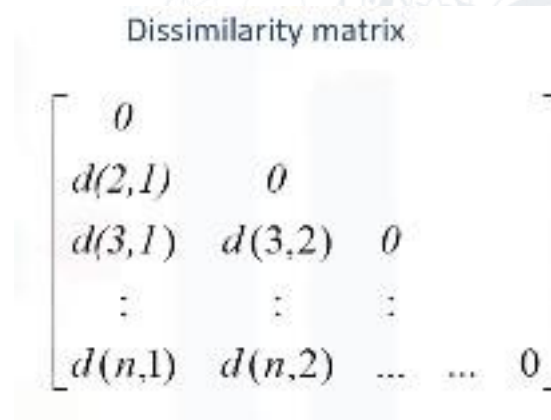
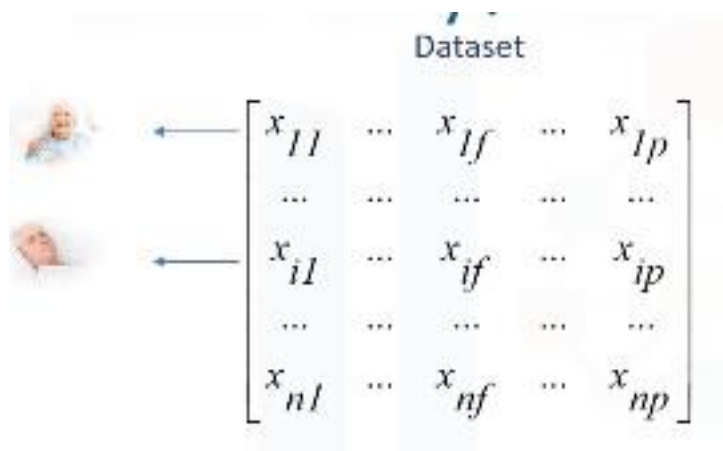
For example, Age, Body Mass Index (or BMI), and Blood Pressure.

We can use different distance measurements to calculate the proximity matrix. For instance, Euclidean distance.

$$\begin{aligned} \text{Dis}(p1, p2) &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \\ &= \sqrt{(54 - 50)^2 + (190 - 200)^2 + (120 - 125)^2} \\ &= 11.87 \end{aligned}$$

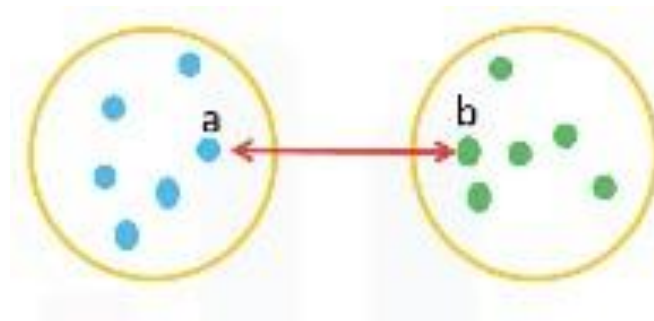
So, if we have a dataset of n patients, we can build an n by n dissimilarity-distance matrix.

It will give us the distance of clusters with 1 data point. However, as mentioned, we merge clusters in



More on Hierarchical Clustering

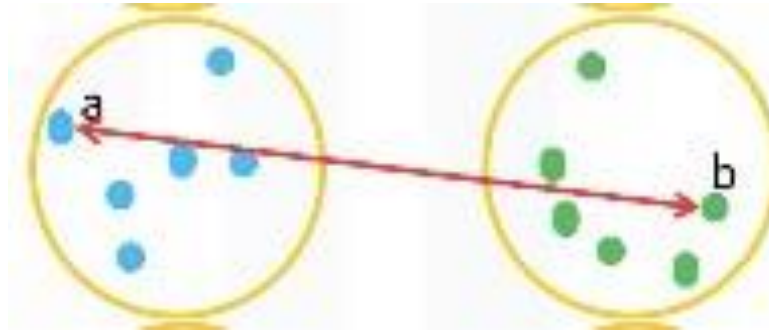
- Agglomerative clustering. Now, the question is, “How can we calculate the distance between clusters when there are multiple patients in each cluster?”
- We can use different criteria to find the closest clusters, and merge them.
- In general, it completely depends on the data type, dimensionality of data, and most importantly, the domain knowledge of the dataset.
- In fact, different approaches to defining the distance between clusters, distinguish the different algorithms.
- As you might imagine, there are multiple ways we can do this. The first one is called Single-Linkage Clustering.
 - Single linkage is defined as the shortest distance between 2 points in each cluster, such as point “a” and “b”.



More on Hierarchical Clustering

Next up is Complete-Linkage Clustering.

This time, we are finding the longest distance between points in each cluster, such as the distance between point "a" and "b".



The third type of linkage is Average Linkage Clustering, or the mean distance.

This means we're looking at the average distance of each point from one cluster to every point in another cluster.

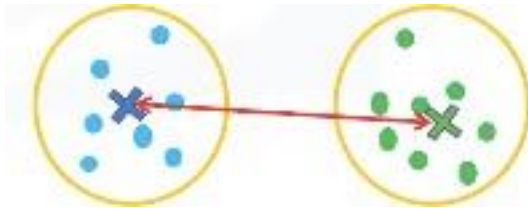


More on Hierarchical Clustering

The final linkage type to be reviewed is Centroid Linkage Clustering.

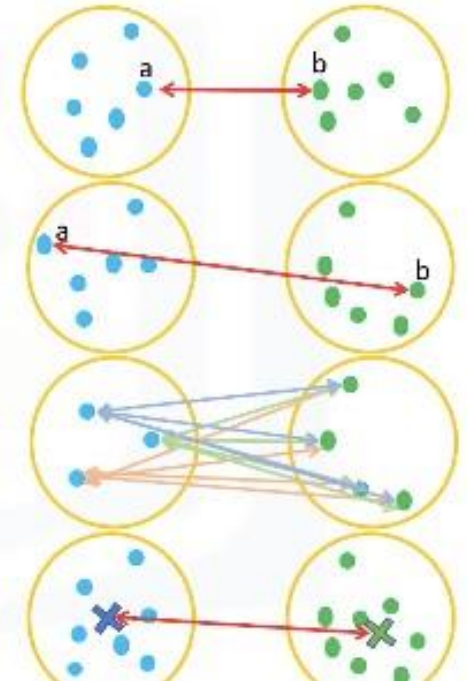
Centroid is the average of the feature sets of points in a cluster.

This linkage takes into account the centroid of each cluster when determining the minimum distance.



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

- There are 3 main advantages to using hierarchical clustering.
 - First, we do not need to specify the number of clusters required for the algorithm.
 - Second, hierarchical clustering is easy to implement.
 - And third, the dendrogram produced is very useful in understanding the data.
- There are some disadvantages as well.
 - First, the algorithm can never undo any previous steps. So for example, the algorithm clusters 2 points, and later on we see that the connection was not a good one, the program cannot undo that step.
 - Second, the time complexity for the clustering can result in very long computation times, in comparison with efficient algorithms, such k-Means.
 - Finally, if we have a large dataset, it can become difficult to determine the correct number of clusters by the dendrogram.



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



UNIVERSITAS
INDONESIA

Veritas, Probatum, Justitia

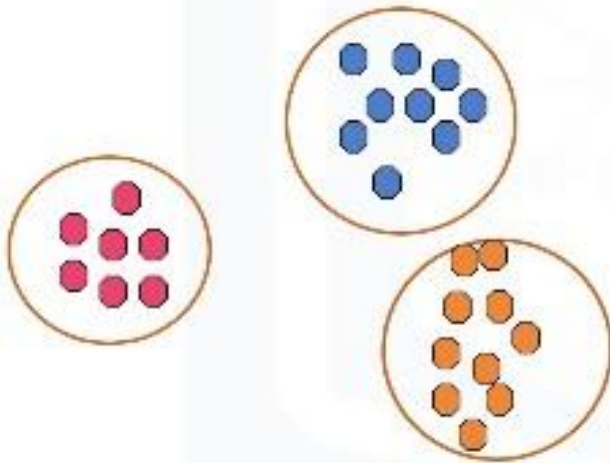
DBSCAN

DBSCAN Clustering

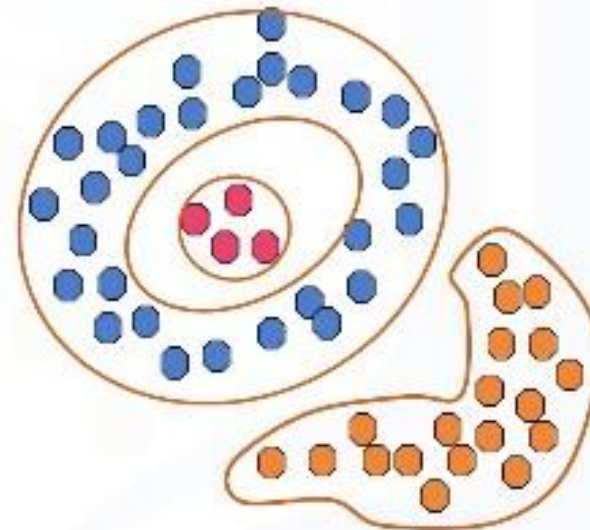
- Most of the traditional clustering techniques, such as k-means, hierarchical, and fuzzy clustering, can be used to group data in an un-supervised way.
- However, when applied to tasks with arbitrary shape clusters, or clusters within clusters, traditional techniques might not be able to achieve good results.
- That is, elements in the same cluster might not share enough similarity -- or the performance may be poor.

Density-based clustering

- Spherical-shape clusters



- Arbitrary-shape clusters

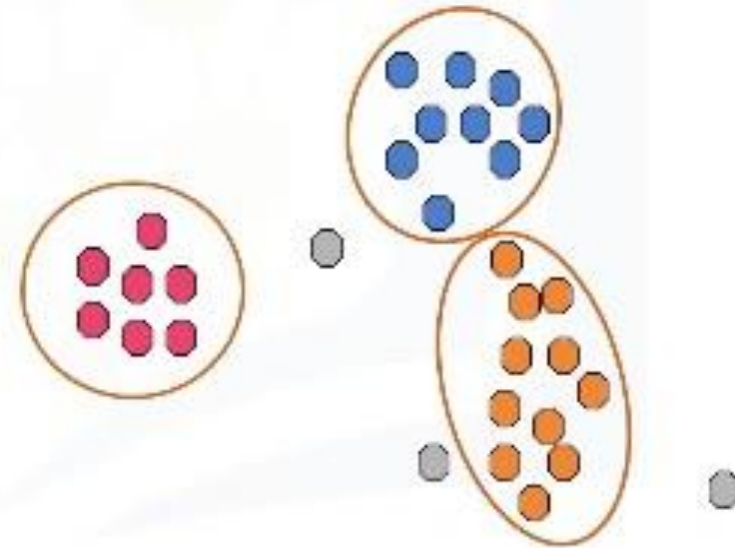
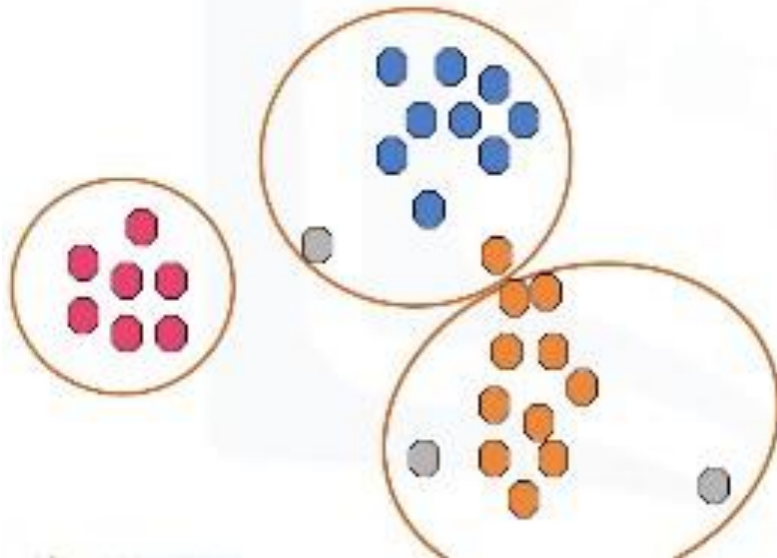


DBSCAN Clustering

- Algoritma clustering seperti K-Means, mungkin mudah dipahami dan diterapkan dalam praktiknya, tetapi algoritma tsb tidak dapat mengidentifikasi outlier.
- Semua sampel harus dimasukkan ke klaster, bahkan jika mereka sebenarnya tidak termasuk dalam klaster manapun.
- Titik anomali menarik sentroid cluster ke arah mereka
- Sebaliknya, clustering berbasis densitas menempatkan daerah dengan densitas tinggi yang dipisahkan satu sama lain oleh daerah dengan densitas rendah.
- Jenis clustering berbasis densitas yang populer adalah DBSCAN.

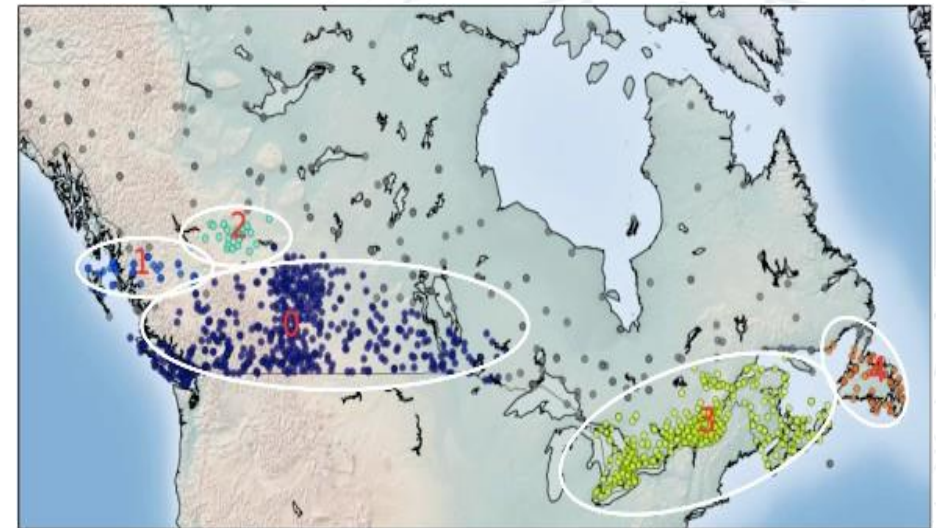
k-Means Vs. density-based clustering

- k-Means assigns all points to a cluster even if they do not belong in any
- Density-based Clustering locates regions of **high density**, and separates outliers



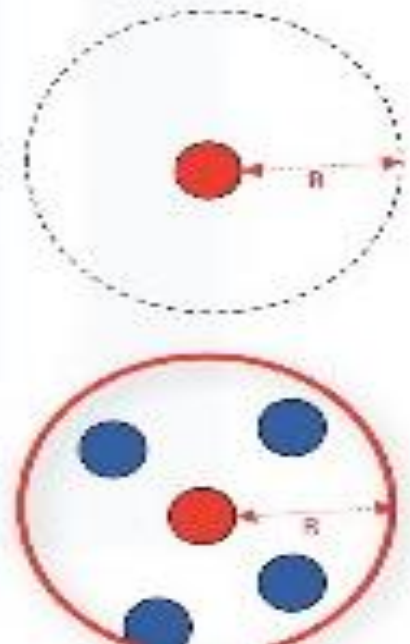
DBSCAN Clustering

- DBSCAN is particularly effective for tasks like class identification on a spatial context.
- The wonderful attribute of the DBSCAN algorithm is that it can find out any arbitrary shape cluster without getting affected by noise.
- For example, this map shows the location of weather stations in Canada.
- DBSCAN can be used here to find the group of stations, which show the same weather conditions.
- As you can see, it not only finds different arbitrary shaped clusters, it can find the denser part of data-centered samples by ignoring less-dense areas or noises.



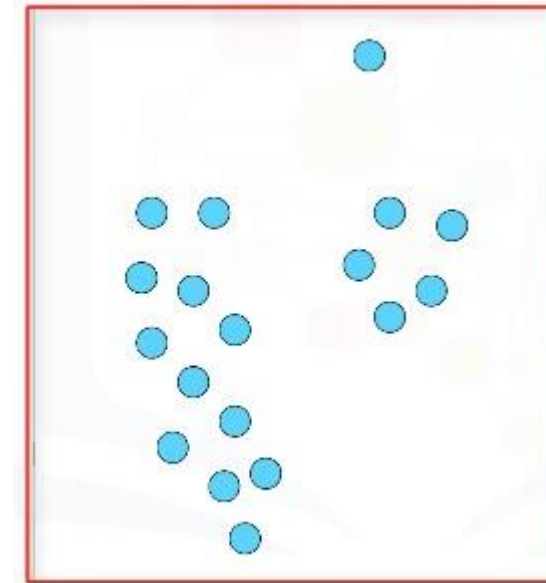
DBSCAN Clustering

- DBSCAN → Density-Based Spatial Clustering of Applications with Noise.
- Ide dasar: jika sebuah titik adalah anggota cluster, maka titik tsb harus dekat dengan banyak titik lain dalam cluster itu.
- Parameter utama:
 1. Radius (R) = radius yang ditentukan, jika terdapat "cukup" titik di dalamnya, maka disebut "dense area."
 2. Minimum Neighbor (M) = menentukan jumlah minimum titik data yang kita inginkan di sekeliling untuk menentukan klaster.



How DBSCAN Works

- Contoh:
 - $R = 2$ units. Misal 2 cm dari point of interest.
 - $M = 6$ points (termasuk point of interest)
- Jenis point:
 - Core, border, atau outlier point.
- Algoritma:
 - Kunjungi setiap point
 - Temukan jenisnya
 - Kelompokkan point sebagai cluster berdasarkan jenisnya.



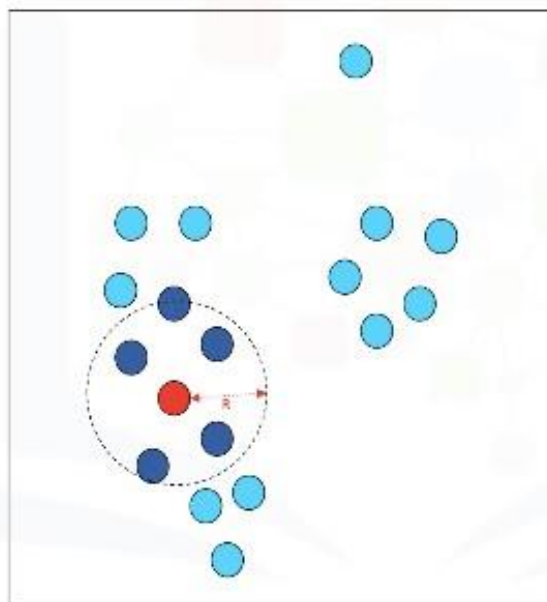
Each point is either:

- *core point*
- *border point*
- *outlier point*

$R = 2\text{unit}$, $M = 6$

How DBSCAN Works

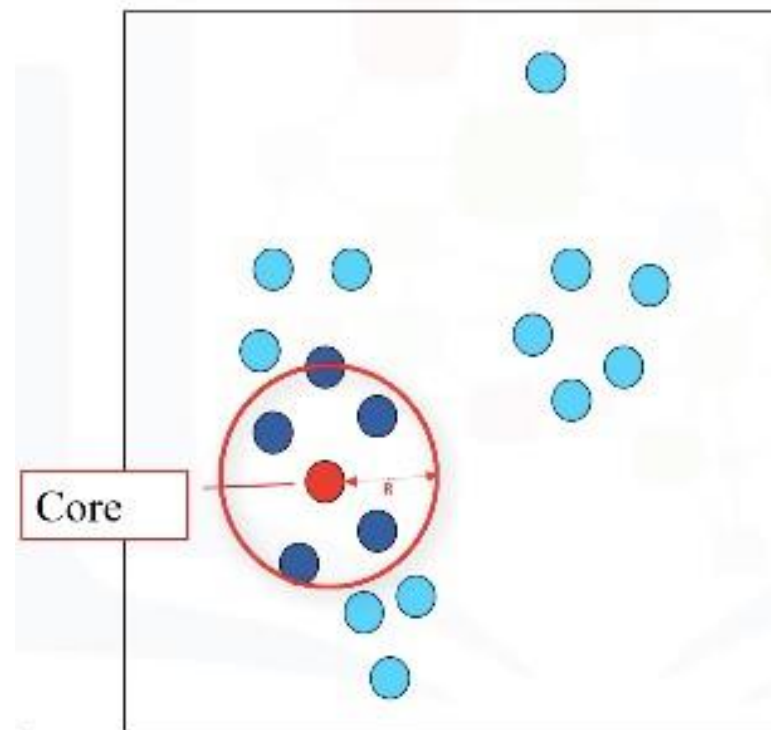
- Mari kita pilih titik secara acak. Pertama kita periksa untuk melihat apakah titik tsb merupakan core point.
- Sebuah titik merupakan **core point** jika dalam radius R , setidaknya ada sejumlah M titik.



$R = 2\text{unit}$, $M = 6$

How DBSCAN Works

- Misalnya, karena ada 6 titik di tetangga pada radius 2 cm dari titik merah, maka titik merah ini adalah core

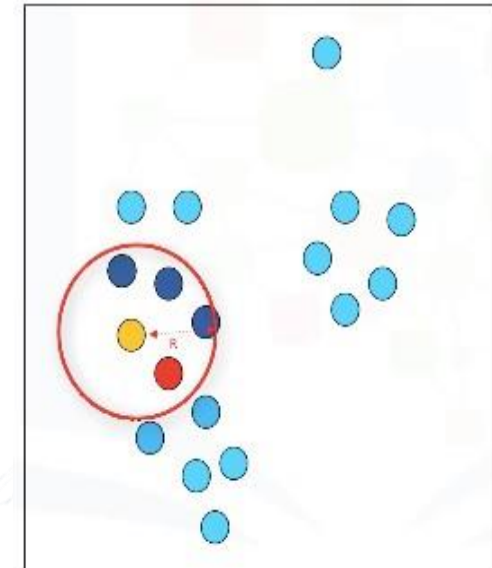


$R = 2\text{unit}$, $M = 6$

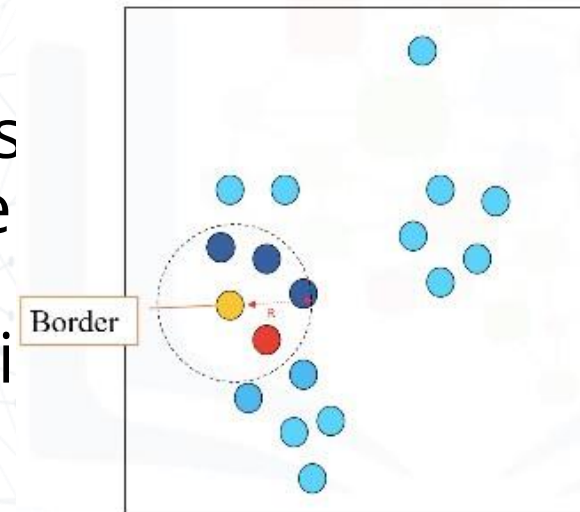


How DBSCAN Works

- Perhatikan titik kuning
- Hanya ada 5 titik di lingkungan ini, termasuk titik kuning.
- Titik kuning merupakan **border point**
- Border point: Its neighborhood contains less than M data points, or It is reachable within R-distance from some core point.
- It means that even though the yellow point is within the 2-centimeter neighborhood of the red point, it is not by itself a core point, because it does not have at least 6 points in its neighborhood.



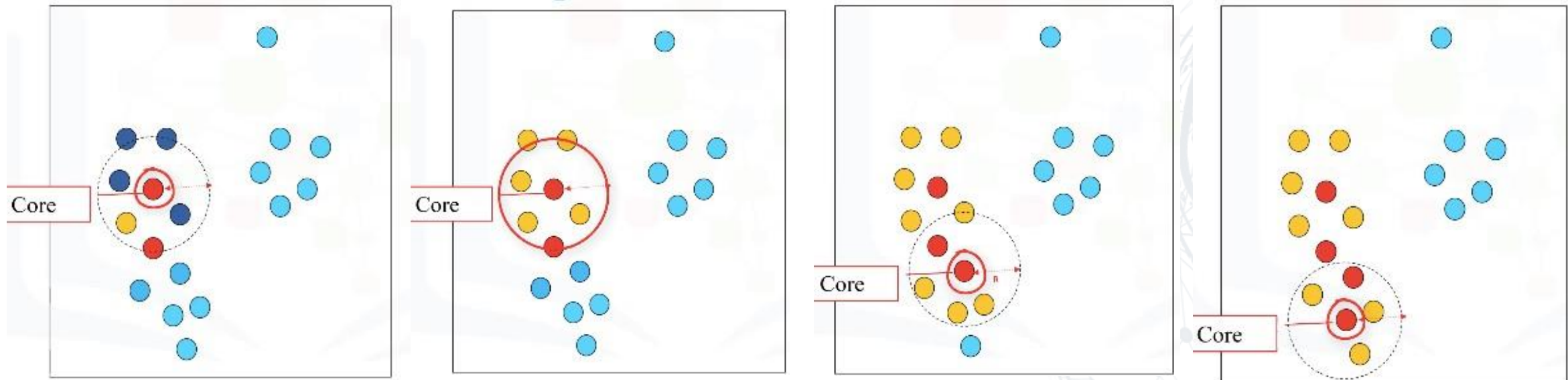
R = 2unit , M = 6



R = 2unit , M = 6

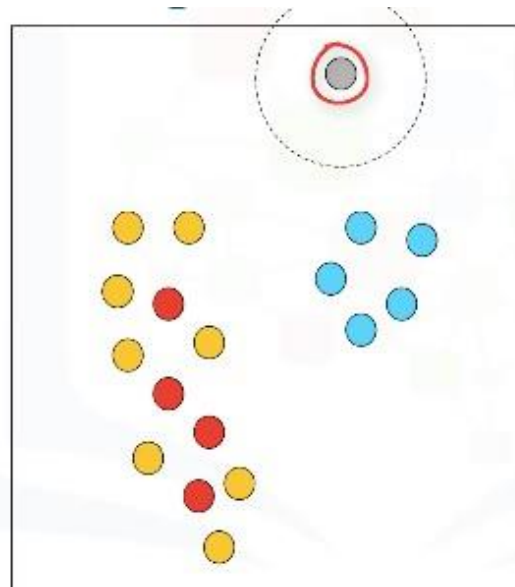
How DBSCAN Works

- We continue with the next point.
- As you can see it is also a core point.
- And all points around it, which are not core points, are border points. Next core point. And next core point.



How DBSCAN Works

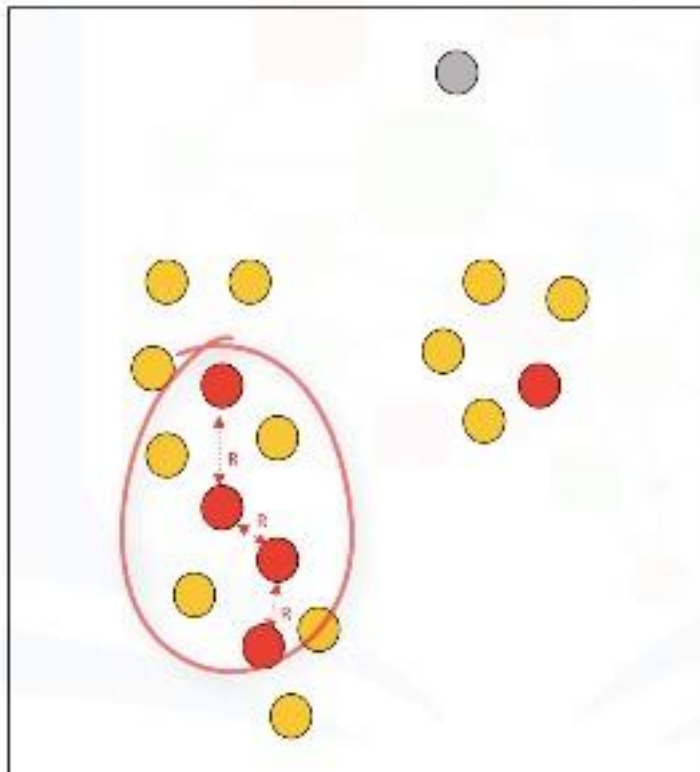
- Let's take this point. You can see it is not a core point, nor is it a border point. So, we'd label it as an outlier.
- What is an outlier? An outlier is a point that: Is not a core point, and also, is not close enough to be reachable from a core point.



How DBSCAN Works

We continue and visit all the points in the dataset and label them as either Core, Border, or Outlier.

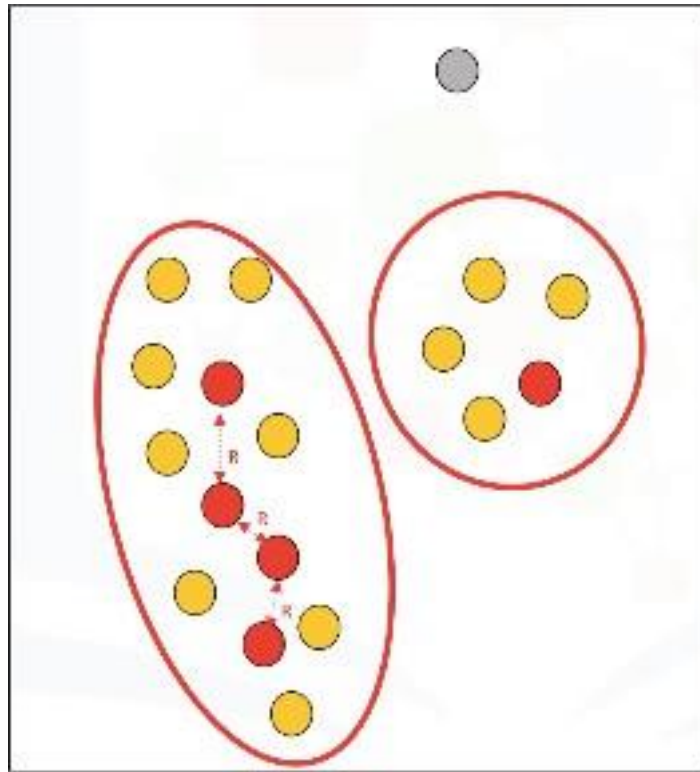
The next step is to connect core points that are neighbors, and put them in the same cluster.



How DBSCAN Works

So, a cluster is formed as at least one core point, plus all reachable core points, plus all their borders.

It simply shapes all the clusters and finds outliers as well.



DBSCAN Advantages

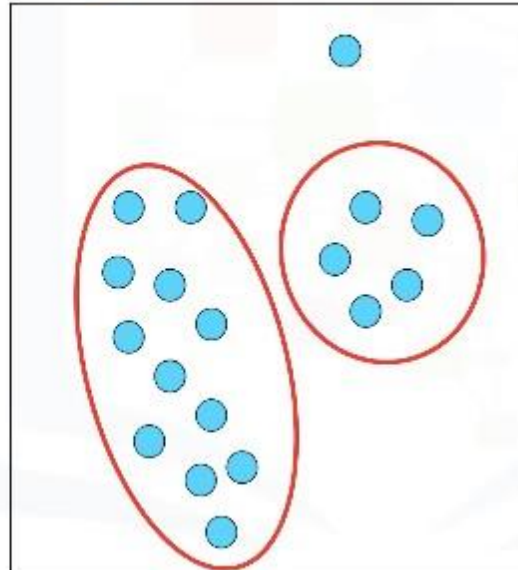
Let's review this one more time to see why DBSCAN is cool.

DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by a different cluster.

DBSCAN has a notion of noise, and is robust to outliers.

On top of that, DBSCAN makes it very practical for use in many really world problems because it does not require one to specify the number of clusters, such as K in k-Means.

Advantages of DBSCAN



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

- Lihat Lab untuk Kmeans, Hierarchical Clustering dan DBSCAN di Cognitiveclass/EMAS