[grouping]

Clustering

unlabelled – dependent feature missing

Overview

similarity metric → distance

grouping

- **Clustering** is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data
- It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different
- In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance

  → hyperparameter * * *
- The decision of which similarity measure to use is application-specific
- Unlike supervised learning, clustering is considered an unsupervised learning method since we don't have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance

  ↳ target/dependent

Applications of Clustering

- Marketing
  - Customer segment discovery
- Library
  - To cluster different books based on topics and information
- Biology
  - Classification among different species of plants and animals
- City planning
  - Analyze the value of houses based on location
- Document Analysis
  - Various research data and documents can be grouped according to certain similarities
  - Labeling large data is really difficult. Clustering can be helpful in these cases to cluster text & group it into various categories
  - Unsupervised techniques like LDA are also beneficial in these cases to find hidden topics in a large corpus

Issues

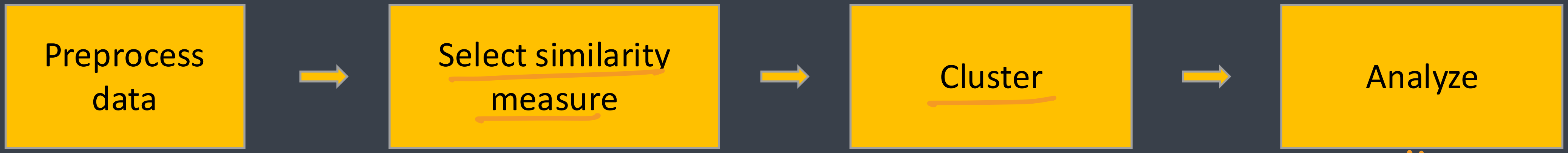
- The results may be less accurate since data isn't labeled in advance and input data isn't known
- The learning phase of the algorithm might take a lot of time as it calculates and analyses all possibilities
- Without any prior knowledge the model is learning from raw data
- As the number of features increases, complexity increases
- Some projects involving live data may require continuous data feeding to the model, resulting in time-consuming and inaccurate results

Overview

unlabelled data → unsupervised learning → label the data → supervised learning

(clustering)

Regression classification



Preprocess data → Select similarity measure → Cluster → Analyze

↓

distance

create clusters

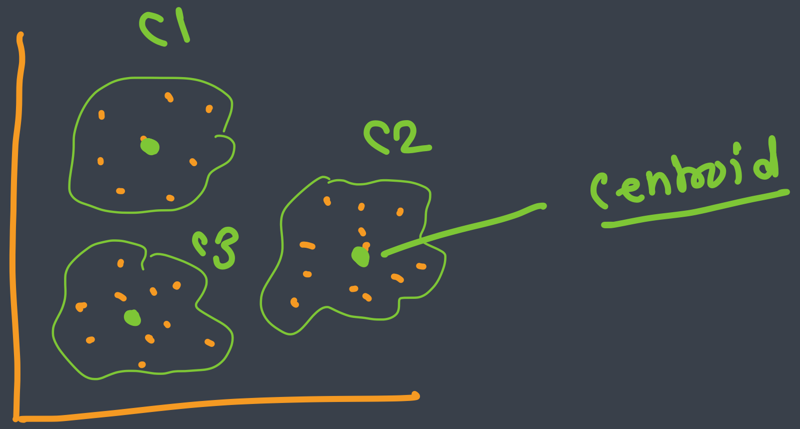
⇅

Clustering Types

Centroid-based Clustering

centroid → center point of cluster

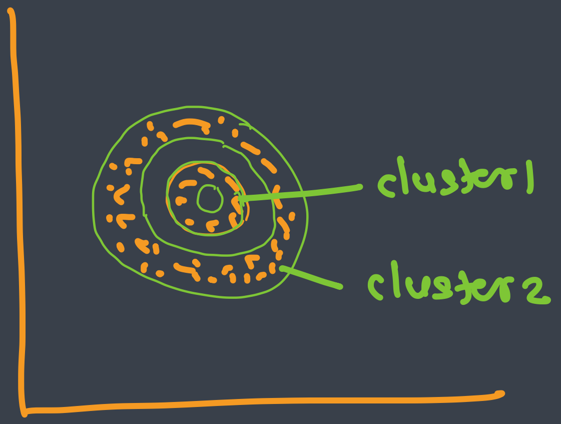
- Centroid-based clustering organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering defined below
- Centroid-based algorithms are efficient but sensitive to initial conditions and outliers
- K-Means is the most widely-used centroid-based clustering algorithm



Density based Clustering

- Density-based clustering connects areas of high ~~example~~ density into clusters
- This allows for arbitrary-shaped distributions as long as dense areas can be connected
- These algorithms have difficulty with data of varying densities and high dimensions
- Further, by design, these algorithms do not assign outliers to clusters

DBScan



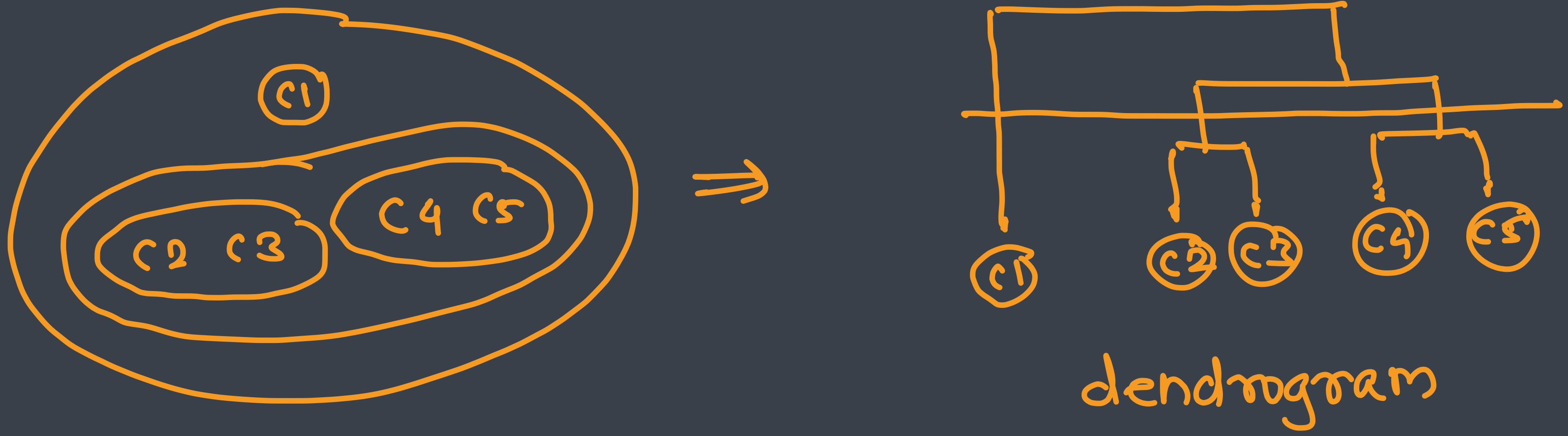
Distribution based Clustering

- This clustering approach assumes data is composed of distributions, such as Gaussian distributions
- The distribution-based algorithm clusters data into three Gaussian distributions
- As distance from the distribution's center increases, the probability that a point belongs to the distribution decreases
- The bands show that decrease in probability.
- When you do not know the type of distribution in your data, you should use a different algorithm

Hierarchical Clustering

- Hierarchical clustering creates a tree of clusters
- Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies
- In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level

Hierarchical clustering

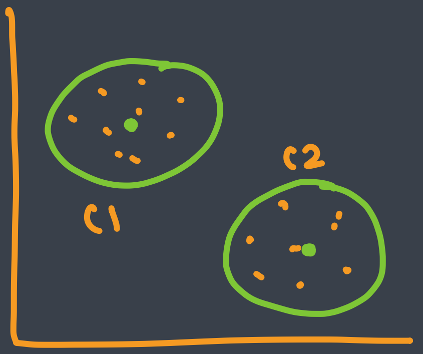


k-means

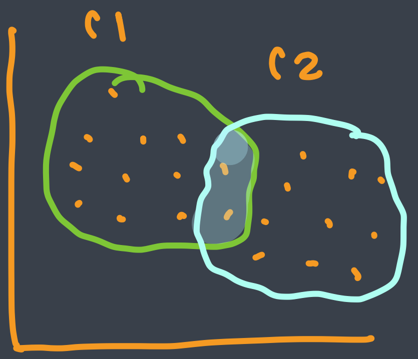
k- no of clusters

Overview

- **k-means** algorithm is an iterative algorithm that tries to partition the dataset into distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**
- It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible
- It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum
- The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster



mutually exclusive
non-overlapping



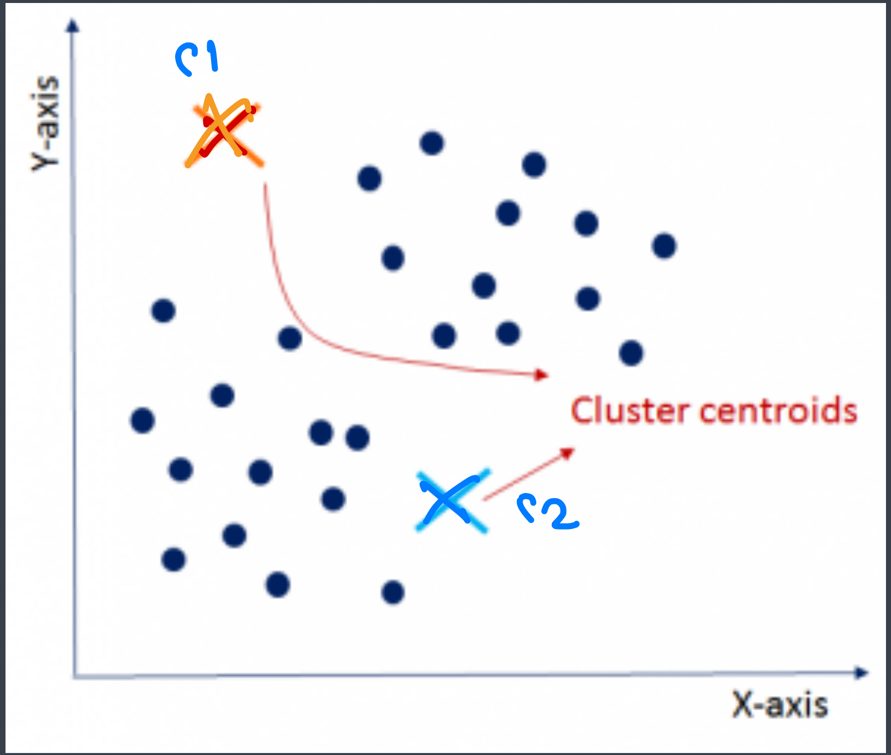
overlapping

How does it work?

- Specify number of clusters *K*
- Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing
- Compute the sum of the squared distance between data points and all centroids
- Assign each data point to the closest cluster (centroid)
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster

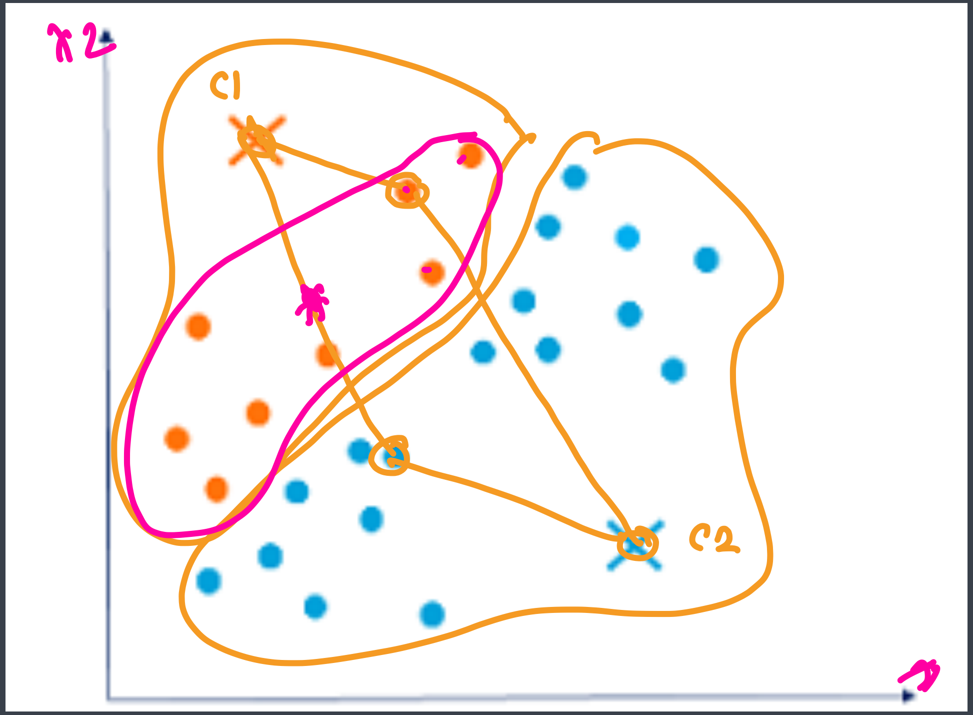
K-Means Clustering - Algorithm

- **Initialization** K = 2
  - randomly initialise two points called the cluster centroids



K-Means Clustering - Algorithm

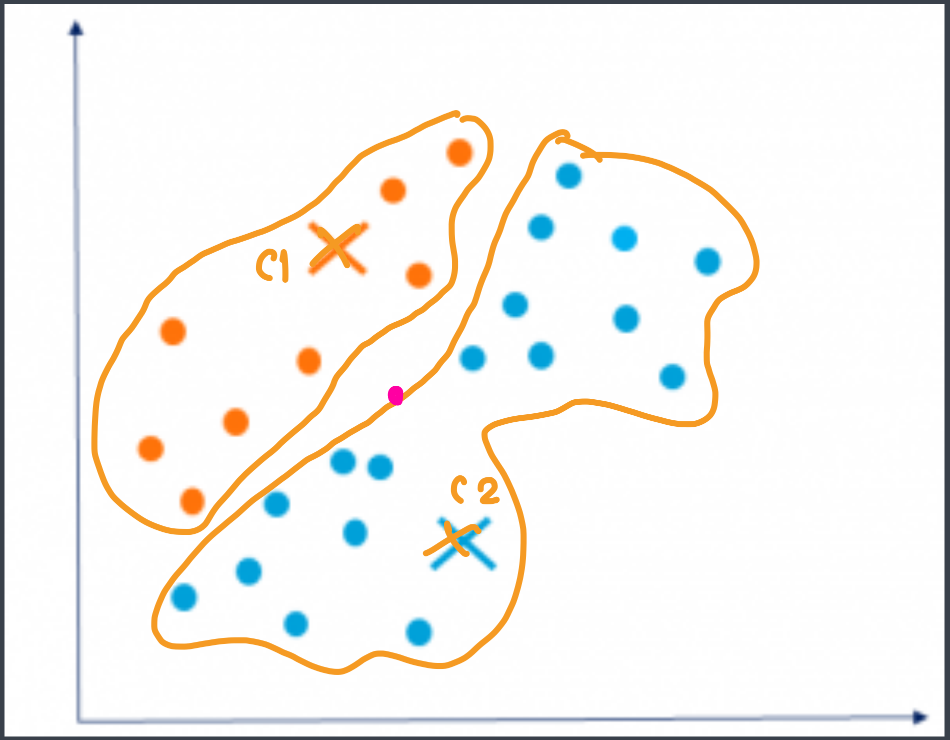
- **Cluster Assignment**
  - Compute the distance between both the points and centroids
  - Depending on the minimum distance from the centroid divide the points into two clusters



K-Means Clustering - Algorithm

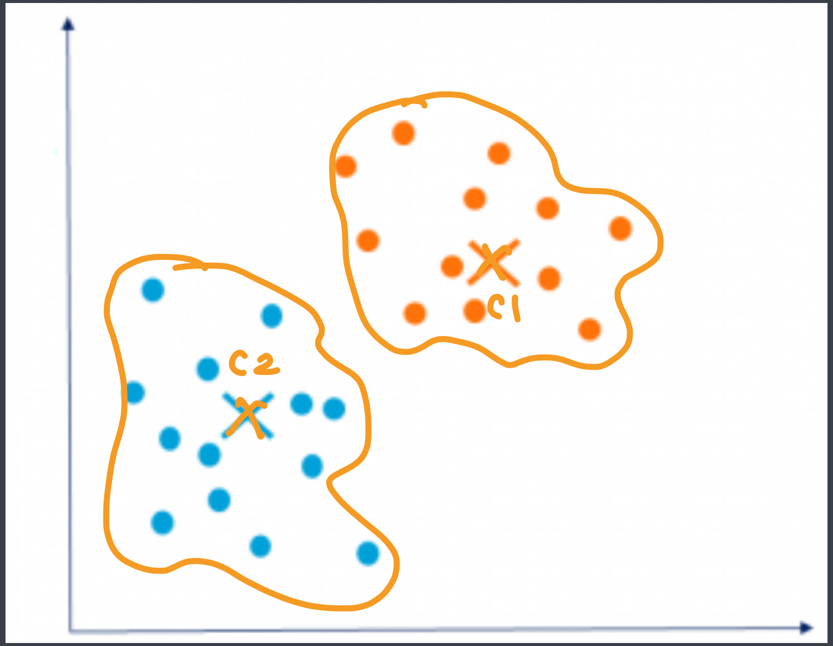
- **Move Centroid**
  - Consider the older centroids are data points
  - Take the older centroid and iteratively reposition them for optimization

- **Optimization**
  - Repeat the steps until the cluster centroids stop changing the position



K-Means Clustering - Algorithm

- **Convergence**
  - Finally, k-means clustering algorithm converges and divides the data points into two clusters clearly visible in multiple clusters



K-Means Clustering - Example

- Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows:

15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65

N = 19

iteretion -1 → initialization , k=2 , c1 = 17 c2 = 20

ages	distance		
	c1 (17)	c2 (20)	cluster
15	-2	-5	c1
15	-2	-5	c1
16	-1	-4	c1
19	2	-1	c2
19	2	-1	c2
20	3	0	c2
20	3	0	c2
21	4	1	c2
22	5	2	c2
28	11	8	c2

c1 = 15

c2 = 21

ages	distance		cluster
	c1 (15)	c2 (21)	
15	0	-6	c1
15	0	-6	c1
16	1	-5	c1
19	4	-2	c2
19	4	-2	c2
20	5	-1	c2
20	5	-1	c2
21	6	0	c2
22	7	1	c2
28	13	7	c2

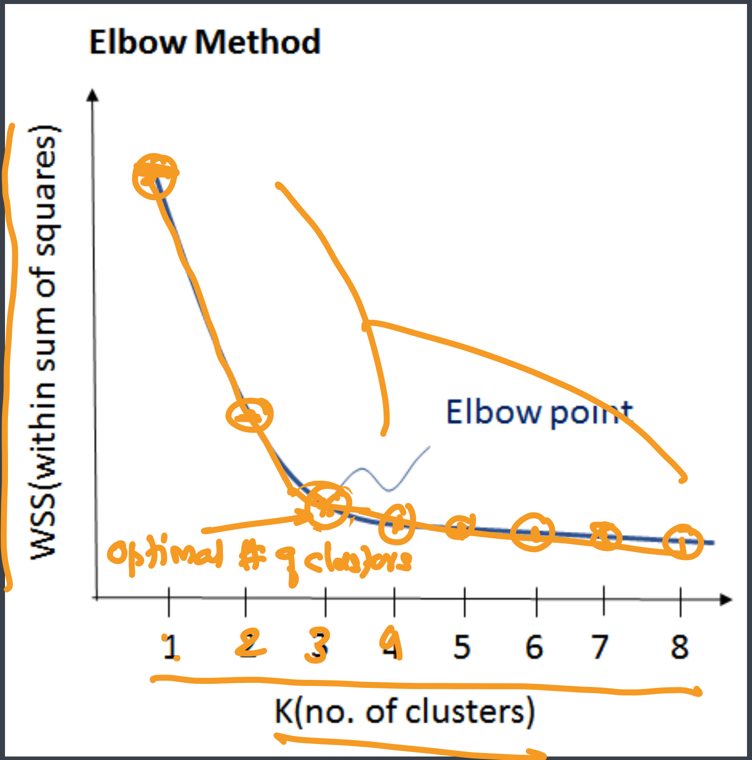
c1 = 15

c2 = 21

Optimization

Elbow Method

- Total within-cluster variation
  - Also known as Within Sum of Squares (WSS)
  - The sum of squared distances (Euclidean) between the items and the corresponding centroid
- Draw a curve between WSS (within sum of squares) and the number of clusters
- It is called elbow method because the curve looks like a human arm and the elbow point gives us the optimum number of clusters



	c1	c2
d1	d1-c1	d1-c2
d2	d2-c1	d2-c2
d3	d3-c1	d3-c2
d4	d4-c1	d4-c2
d5	d5-c1	d5-c2

$K=2 = \Sigma (x_i - c_i)^2$

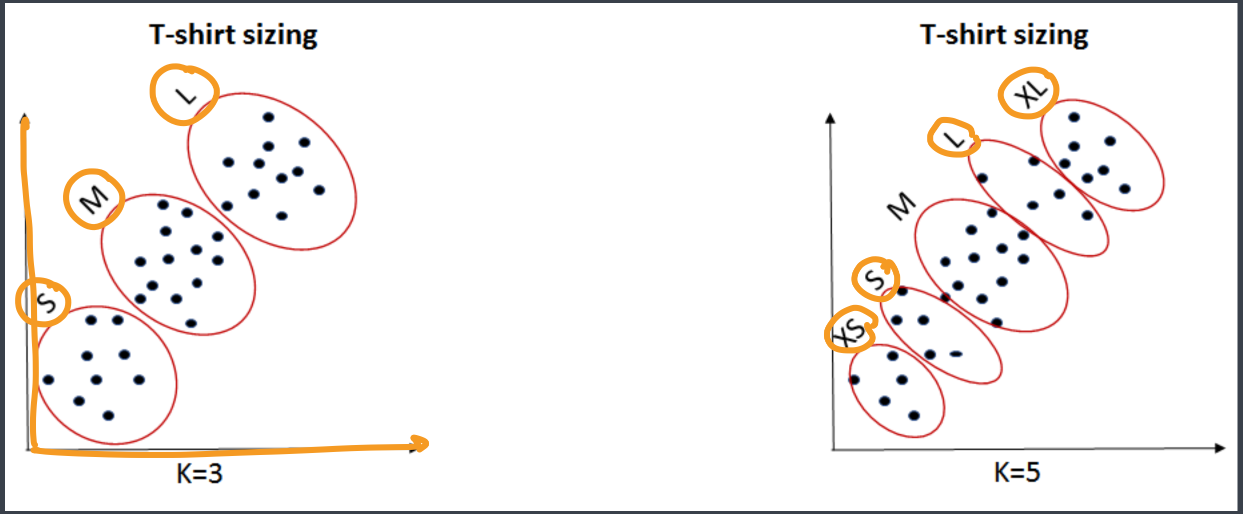
$k=3 =$ —— er. ——

$K=4 =$ —— ——

⋮ ⋮ ⋮

Purpose Method

- Get different clusters based on a variety of purposes
- Partition the data on different metrics and see how well it performs for that particular case



- K=3: If you want to provide only 3 sizes(S, M, L) so that prices are cheaper, you will divide the data set into 3 clusters.
- K=5: Now, if you want to provide more comfort and variety to your customers with more sizes (XS, S, M, L, XL), then you will divide the data set into 5 clusters.

Advantages

- It’s straightforward to implement
- It’s scalable to massive datasets and also faster for large datasets
- It adapts to new examples very frequently

Disadvantages

- K-Means clustering is good at capturing the structure of the data if the clusters have a spherical-like shape. It always tries to construct a nice spherical shape around the centroid. This means that the minute the clusters have different geometric shapes, K-Means does a poor job clustering the data.
- Even when the data points belong to the same cluster, K-Means doesn’t allow the data points far from one another, and they share the same cluster
- K-Means algorithm is sensitive to outliers
- As the number of dimensions increases, scalability decreases

Hierarchical Clustering

Hierarchical Clustering

- Separating data into different groups based on some measure of similarity
- Types
  - Agglomerative
  - Divisive

Hierarchical Clustering

- Dendrogram
  - diagram that shows the hierarchical relationship between objects

Agglomerative Clustering

- Also called as bottom-top clustering as it uses bottom-up approach
- Each data point starts in its own cluster
- These clusters are then joined greedily by taking two most similar clusters together

Agglomerative Clustering

- Start by assigning each item to a cluster
    - if you have N items, you now have N clusters, each containing just one item
- Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less
- Compute distances (similarities) between the new cluster and each of the old clusters
- Repeat steps 2 and 3 until all items are clustered into a single cluster of size N

Agglomerative Clustering

- **Single linkage**
  - Also known as nearest neighbour clustering
  - The distance between two groups is defined as the distance between their two closest members
  - It often yields clusters in which individuals are added sequentially to a single group
- **Complete linkage**
  - Also known as furthest neighbour clustering
  - The distance between two groups as the distance between their two farthest-apart members
- **Average linkage**
  - Referred to as the unweighted pair-group method
  - Distance between two groups is defined as the average distance between each of their members

Divisive Clustering

- Also called as top-bottom clustering as it uses top-bottom approach
- All data point starts in it’s the same cluster
- Then using parametric clustering like k-means divide the cluster into multiple clusters
- For each cluster repeating the process find sub cluster till the desired number of clusters found

DBSCAN

DBSCAN

- Clustering analysis or Clustering is an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense
- It comprises many different methods based on differential evolution
- E.g. K-Means (distance between points), Affinity propagation (graph distance), Mean-shift (distance between points), DBSCAN (distance between nearest points) etc.
- Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches
- Clusters are dense regions in the data space, separated by regions of the lower density of points
- The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise
- The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points

Why DBSCAN

- Partitioning methods like K-means clustering and hierarchical clustering work for finding spherical-shaped clusters or convex clusters
- In other words, they are suitable only for compact and well-separated clusters
- Moreover, they are also severely affected by the presence of noise and outliers in the data
- Real life data may contain irregularities, like:
    - Clusters can be of arbitrary shape
    - Data may contain noise.

Parameters needed

- **eps**
  - It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to ‘eps’ then they are considered neighbors
  - If the eps value is chosen too small then large part of the data will be considered as outliers
  - If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters
  - One way to find the eps value is based on the k-distance graph
- **MinPts**
  - Minimum number of neighbors (data points) within eps radius
  - Larger the dataset, the larger value of MinPts must be chosen
  - As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1
  - The minimum value of MinPts must be chosen at least 3

How does it work ?

- Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors
- For each core point if it is not already assigned to a cluster, create a new cluster
- Find recursively all its density connected points and assign them to the same cluster as the core point
- A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the eps distance
- This is a chaining process. So, if b is neighbor of c, c is neighbor of d, d is neighbor of e, which in turn is neighbor of a implies that b is neighbor of a
- Iterate through the remaining unvisited points in the dataset
- Those points that do not belong to any cluster are noise