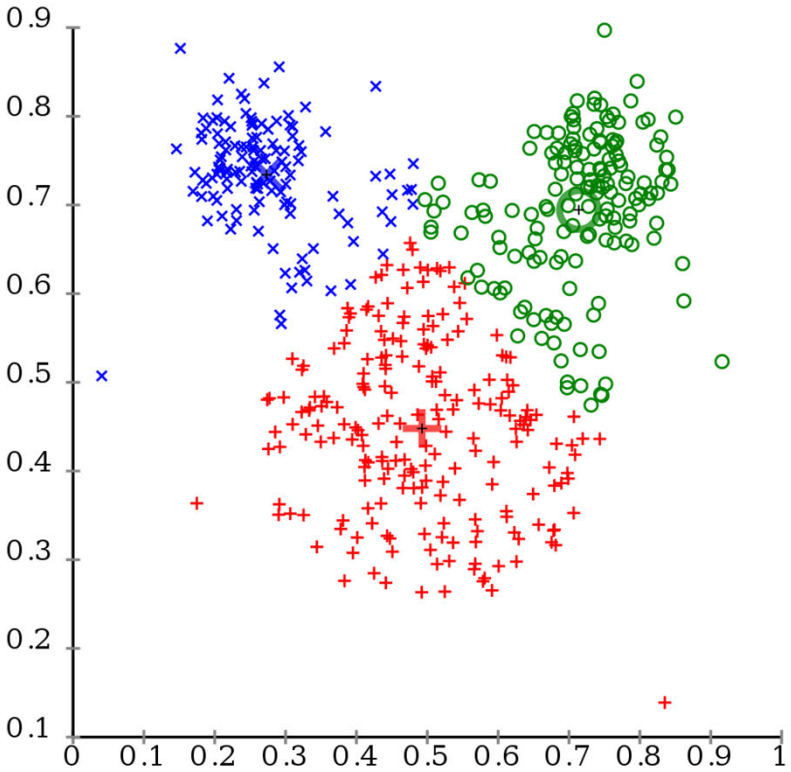
Machine Learning

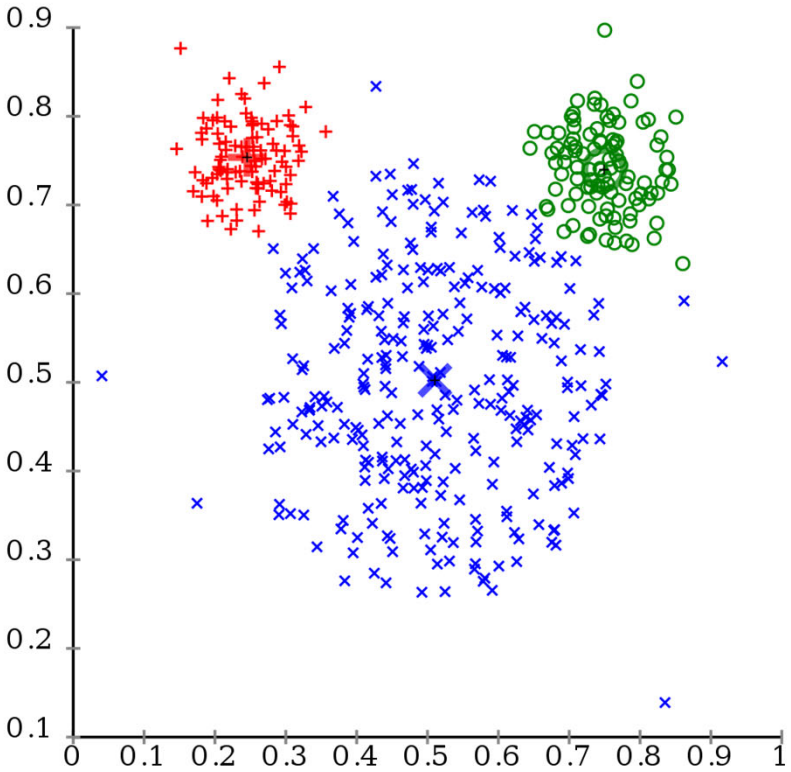
Clustering

Jian Liu

Part 1: K-means



Part 2: Gaussian Mixture Models

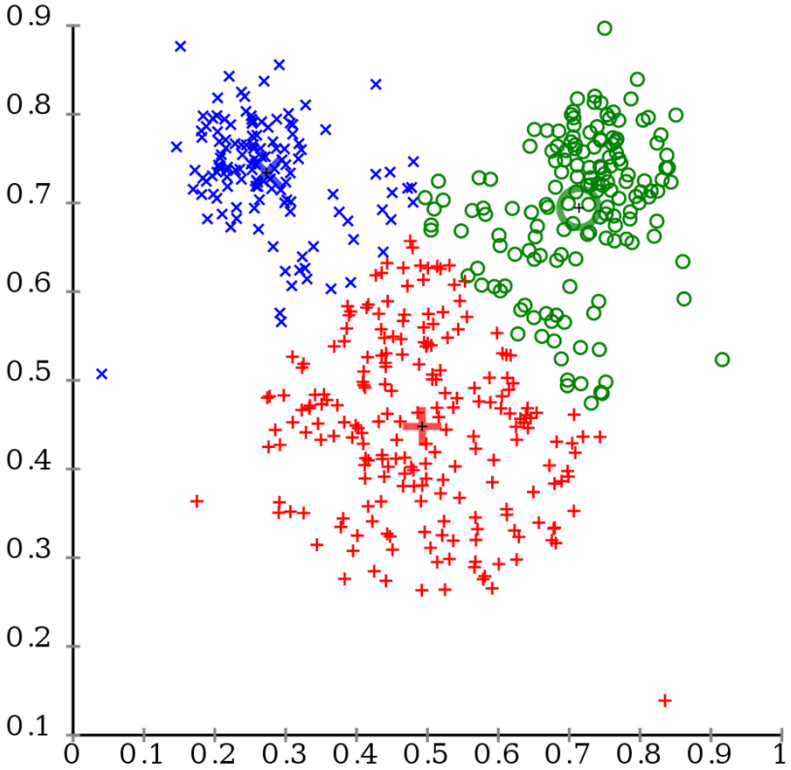


Machine Learning

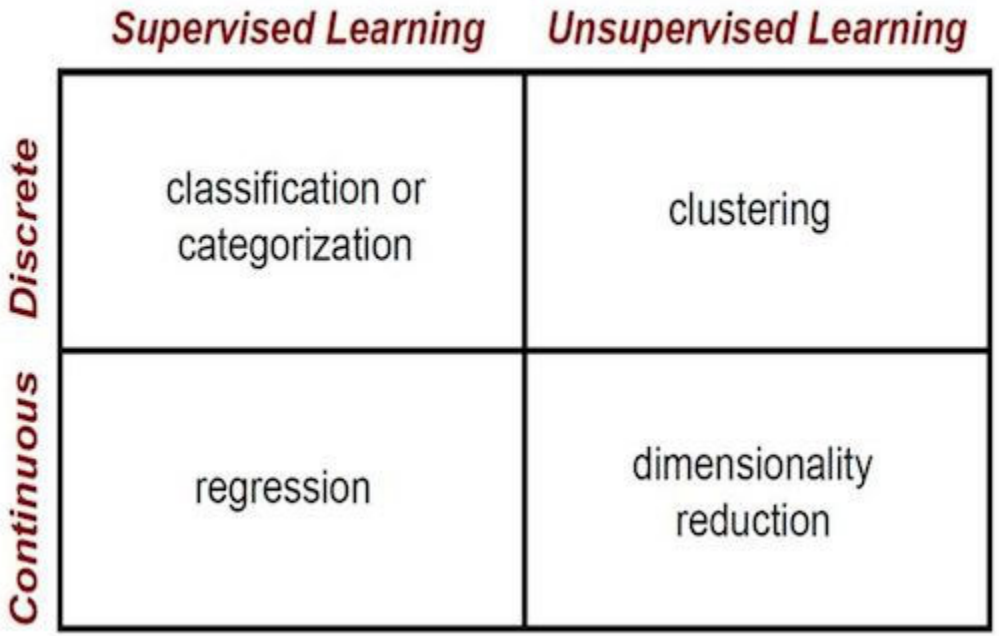
Clustering

Jian Liu

Part 1: K-means



Supervised vs. Unsupervised learning



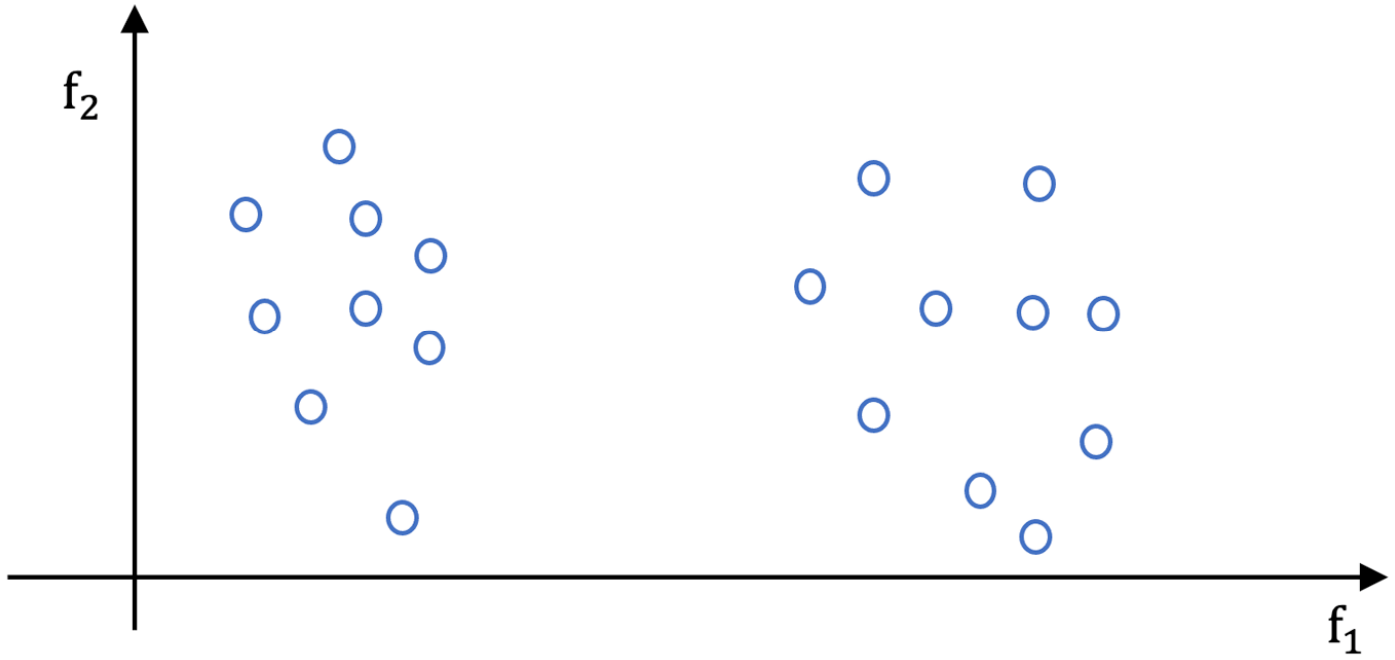
Clustering

Unsupervised learning techniques
Investigate the inner properties of a dataset without any labels.

Two clusters case

Euclidian distance: how close a pair of two data points are

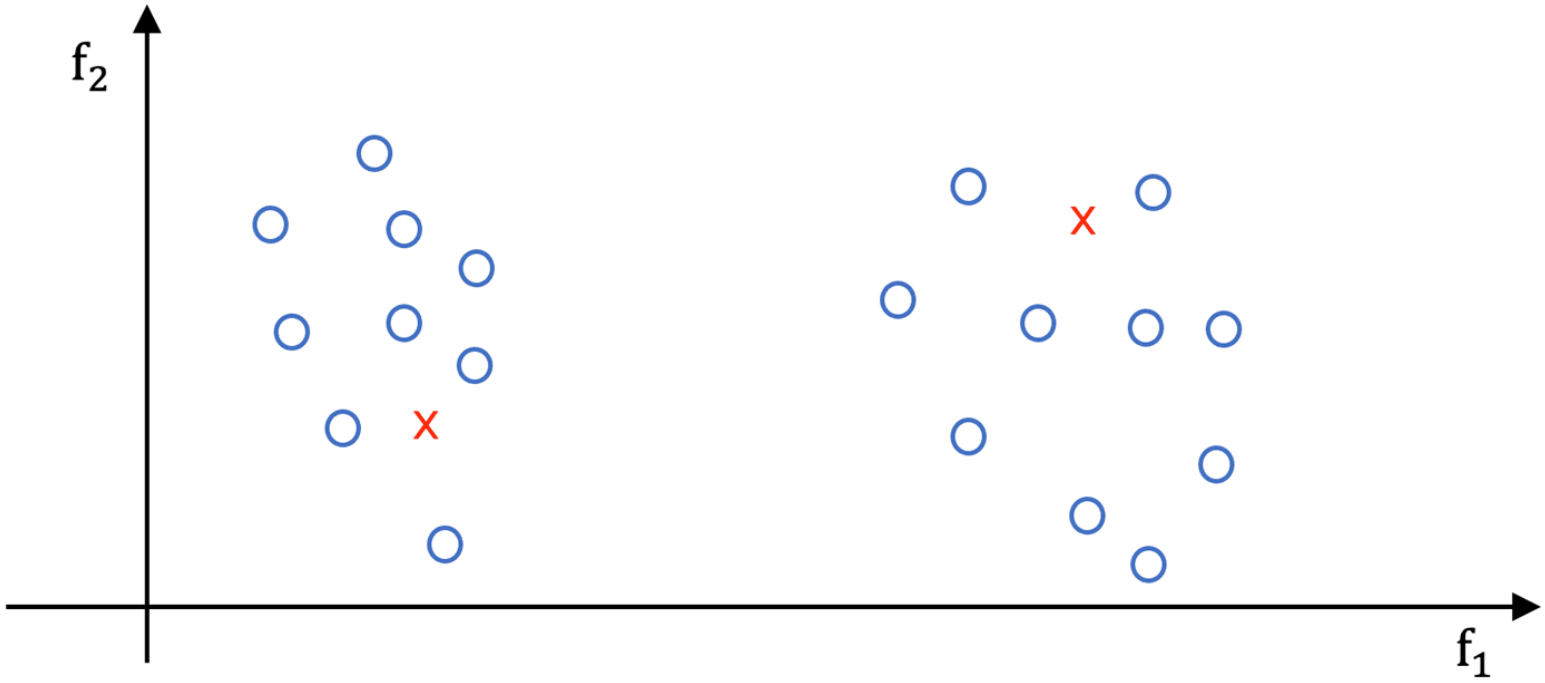
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$



Two clusters case

Euclidian distance: how close a pair of two data points are

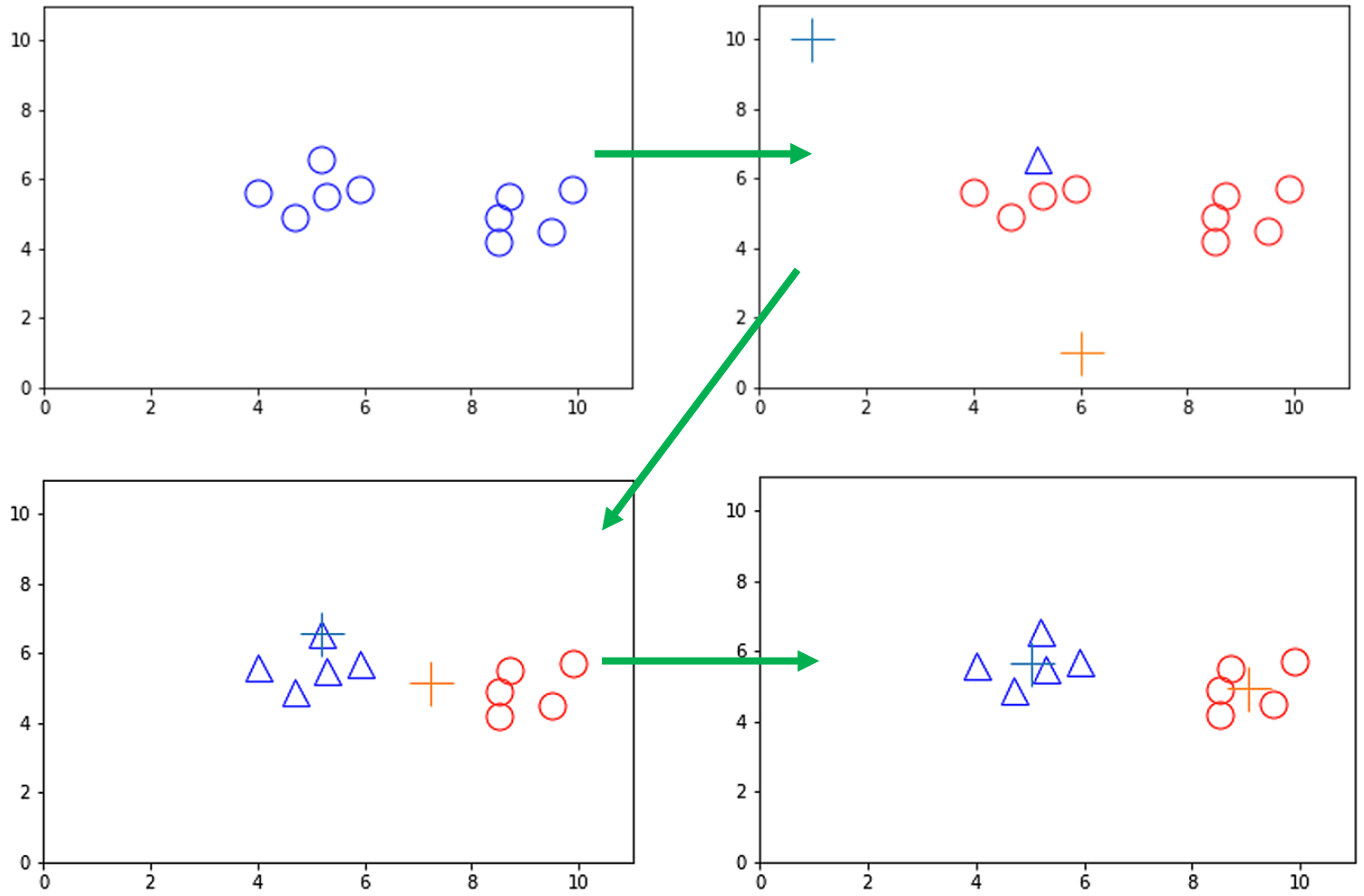
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$



the centroids initialised to a random point

Two clusters case

Steps of K-means clustering algorithms on a simple 2-d dataset



General case multi clusters: K-means algorithm

Do it iteratively:

1 – change our cluster and calculate their centres
2 – re-assign the data points into clusters and then recalculate the centres.

We do that iteratively until the algorithms converge

i.e. the clusters stop changing (or change very little).

Clusters loss function: measuring the quality of the cluster

SSE (sum of squared errors) loss function

$$J = \sum_{i=1}^{K} \sum_{\mathbf{x} \in \boldsymbol{C}_i} d(\boldsymbol{c}_i, \mathbf{x})^2$$

We take the derivative of the cost function with respect to the centroid

$$\nabla J(\boldsymbol{c}_k) = \sum_{\mathbf{x} \in \boldsymbol{C}_k} 2(\boldsymbol{c}_k - \mathbf{x}) = 0$$

$$\sum_{\mathbf{x} \in C_k} c_k = \sum_{\mathbf{x} \in \boldsymbol{C}_k} \mathbf{x}$$

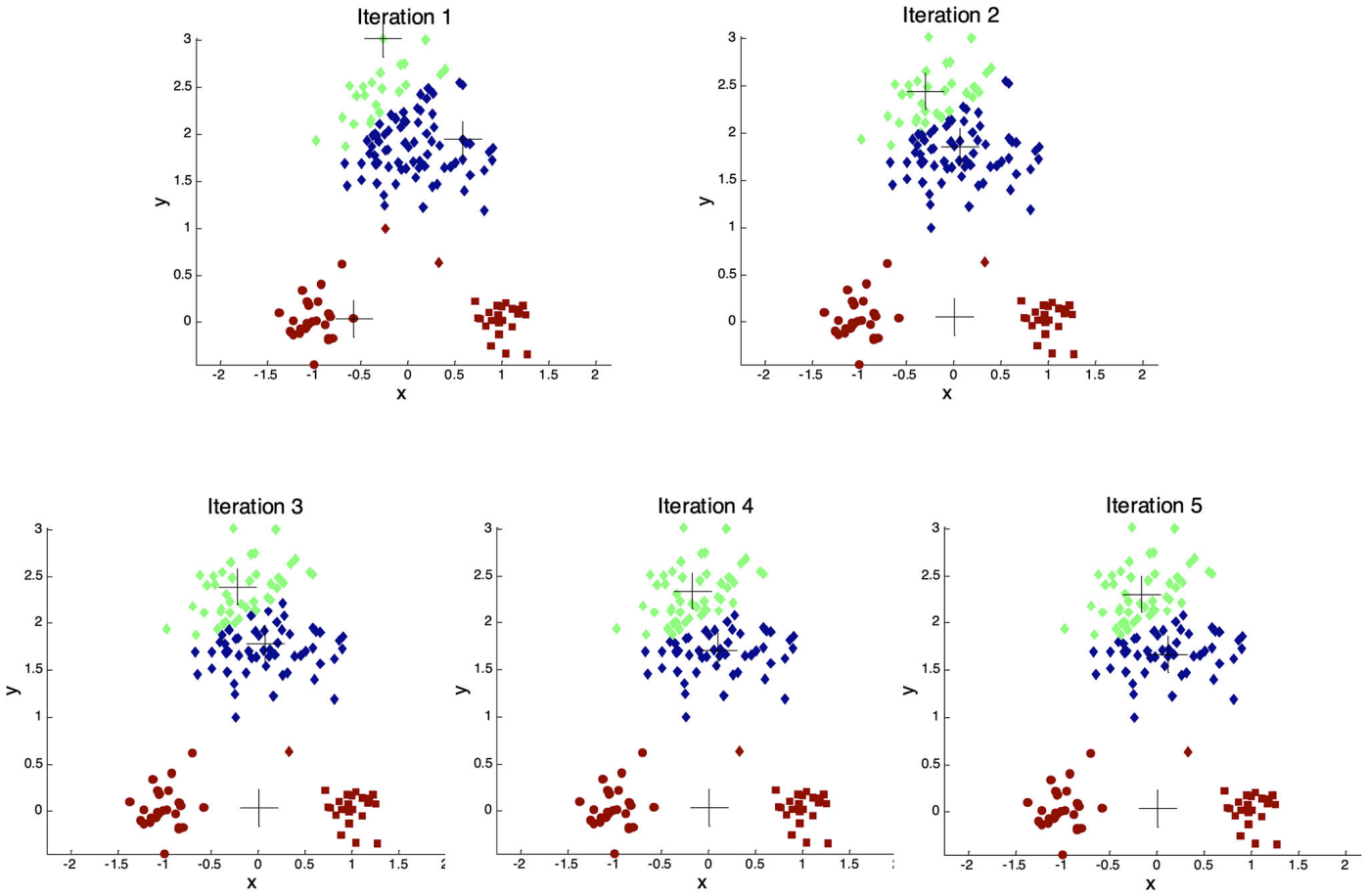
$$c_k |C_k| = \sum_{\mathbf{x} \in \boldsymbol{C}_k} \mathbf{x}$$

$|\boldsymbol{C}_k|$ is the number of data points in cluster $\boldsymbol{C}_k$.

$$c_k = \frac{1}{|C_k|} \sum_{\mathbf{x} \in C_k} \mathbf{x}$$

the K-means algorithm is indeed minimising the loss function SSE by assigning each centroid to the mean of the cluster

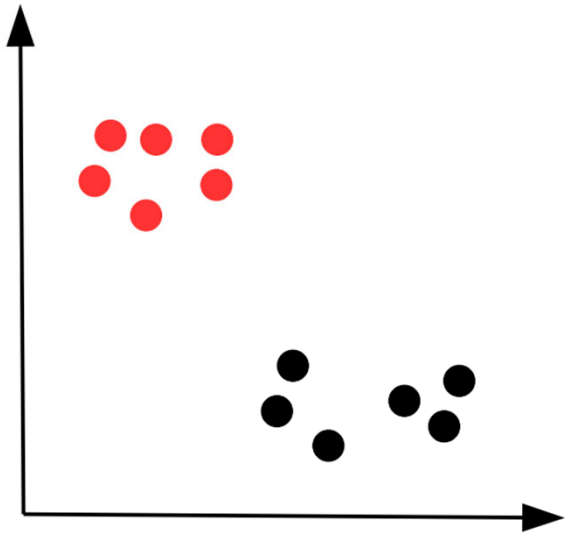
K-means not working well
due an unlucky choice of the initial centroids

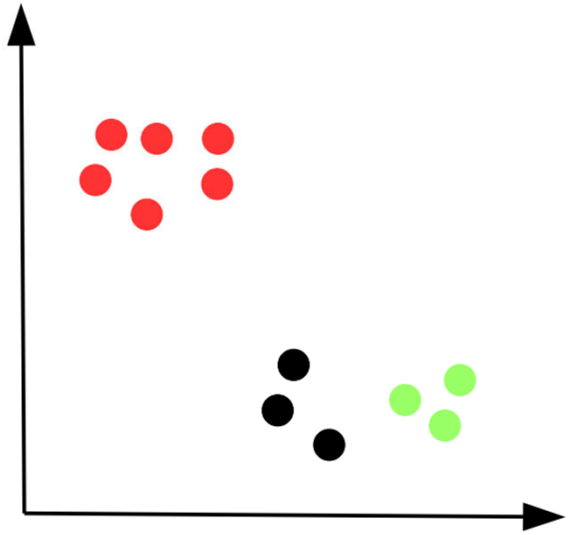


One strategy to overcome this issue is by randomising the initial centroids and performing multiple runs of the K-means and then selecting the one that produce the least SSE.

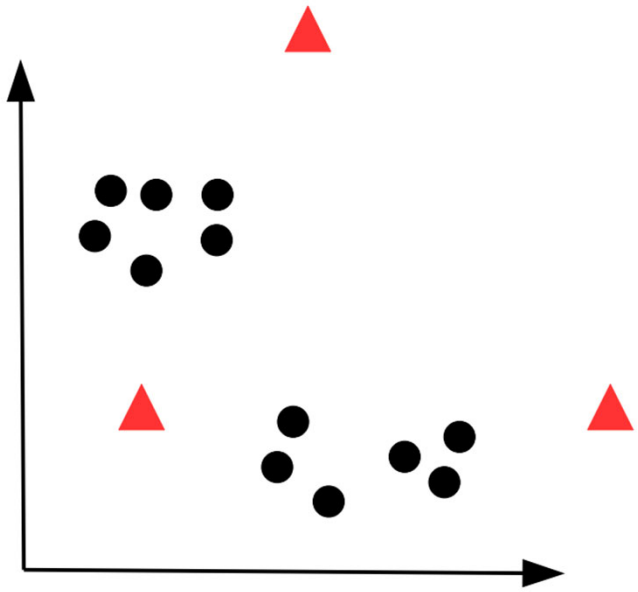
Scale

How many clusters do we have?



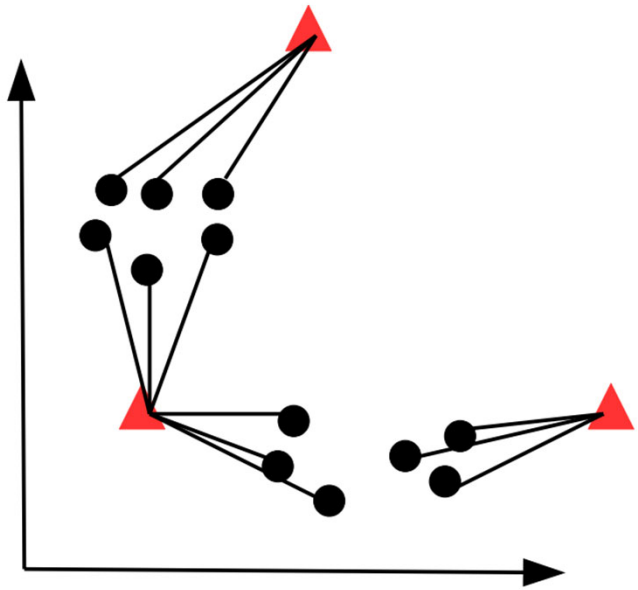


K-means



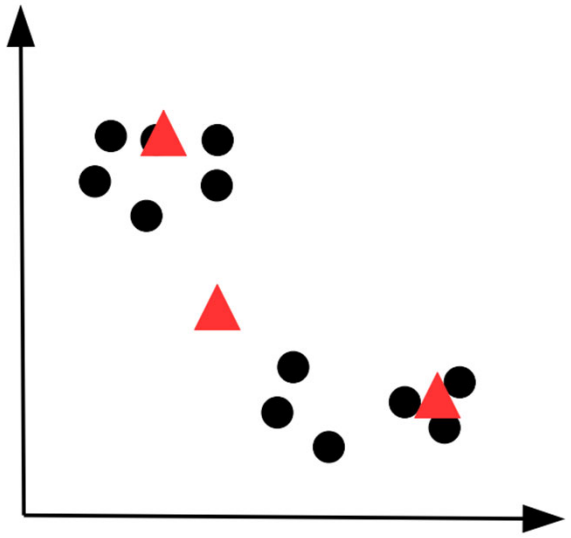
1. Choose the number of clusters (in the example: k=3)
2. Place k centroids randomly (the triangles)

K-means



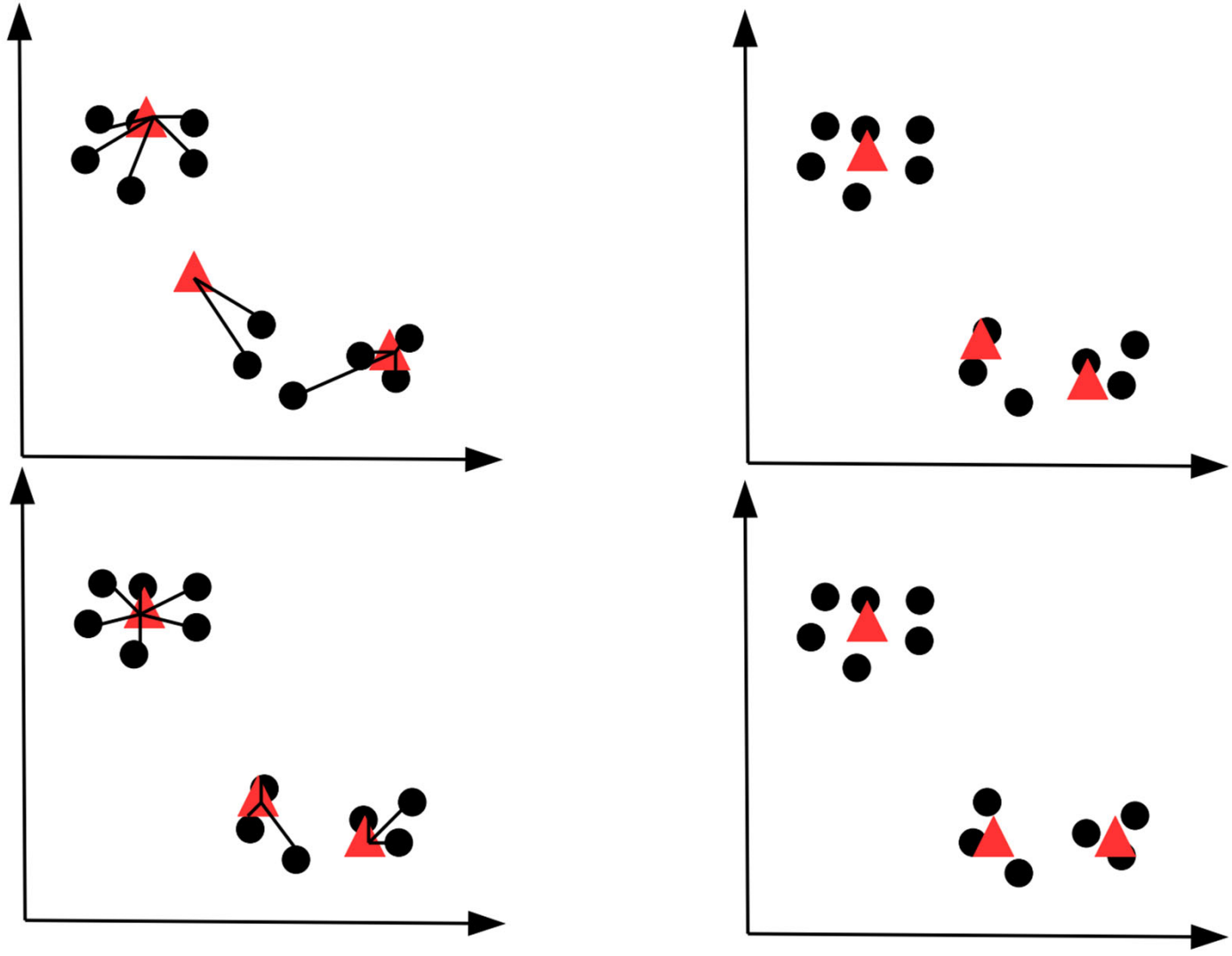
3. Identify the closest centroid to each point
4. Compute the new centroids for the clusters

K-means

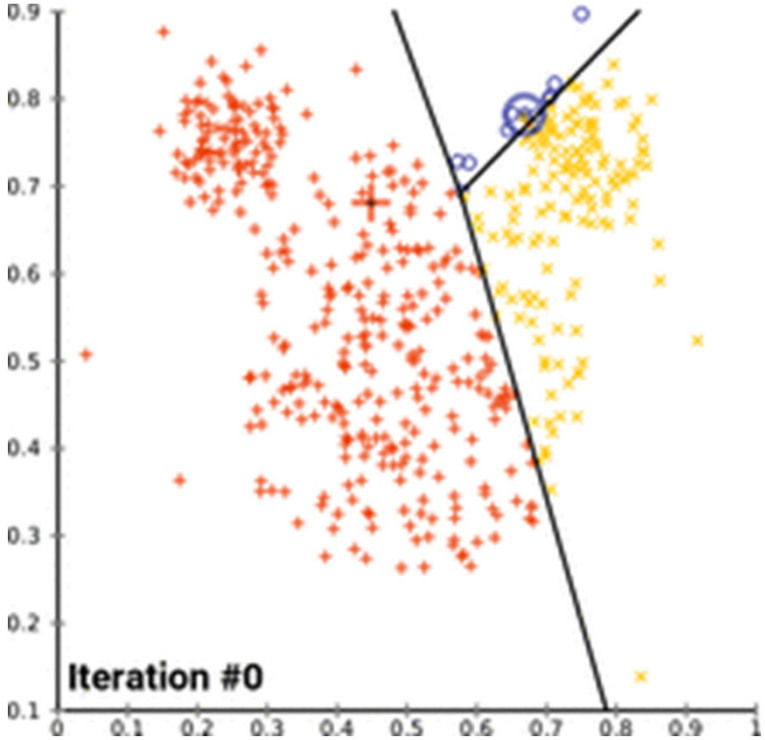


5. Repeat until the centroids do not move

K-means

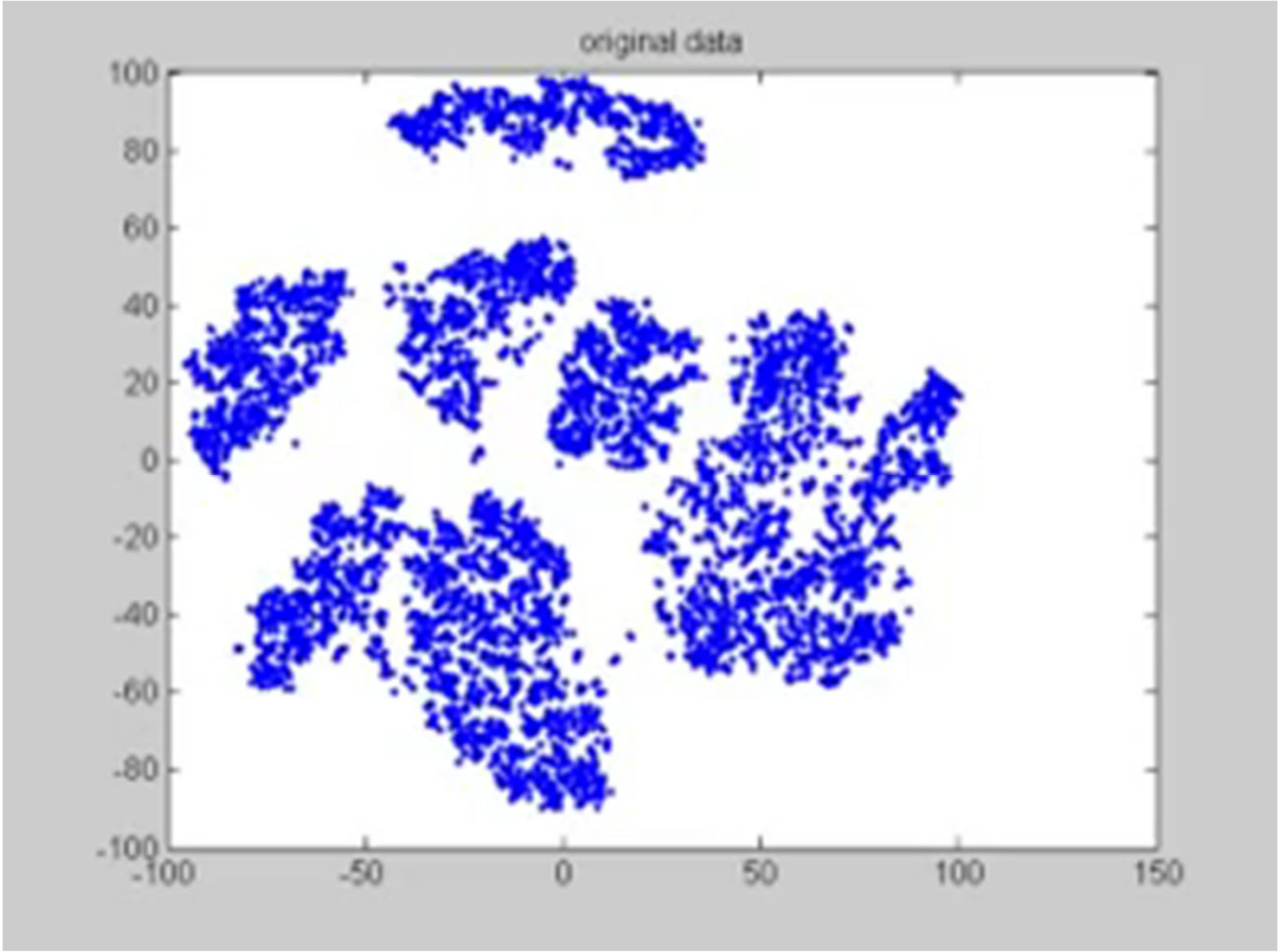


K-means



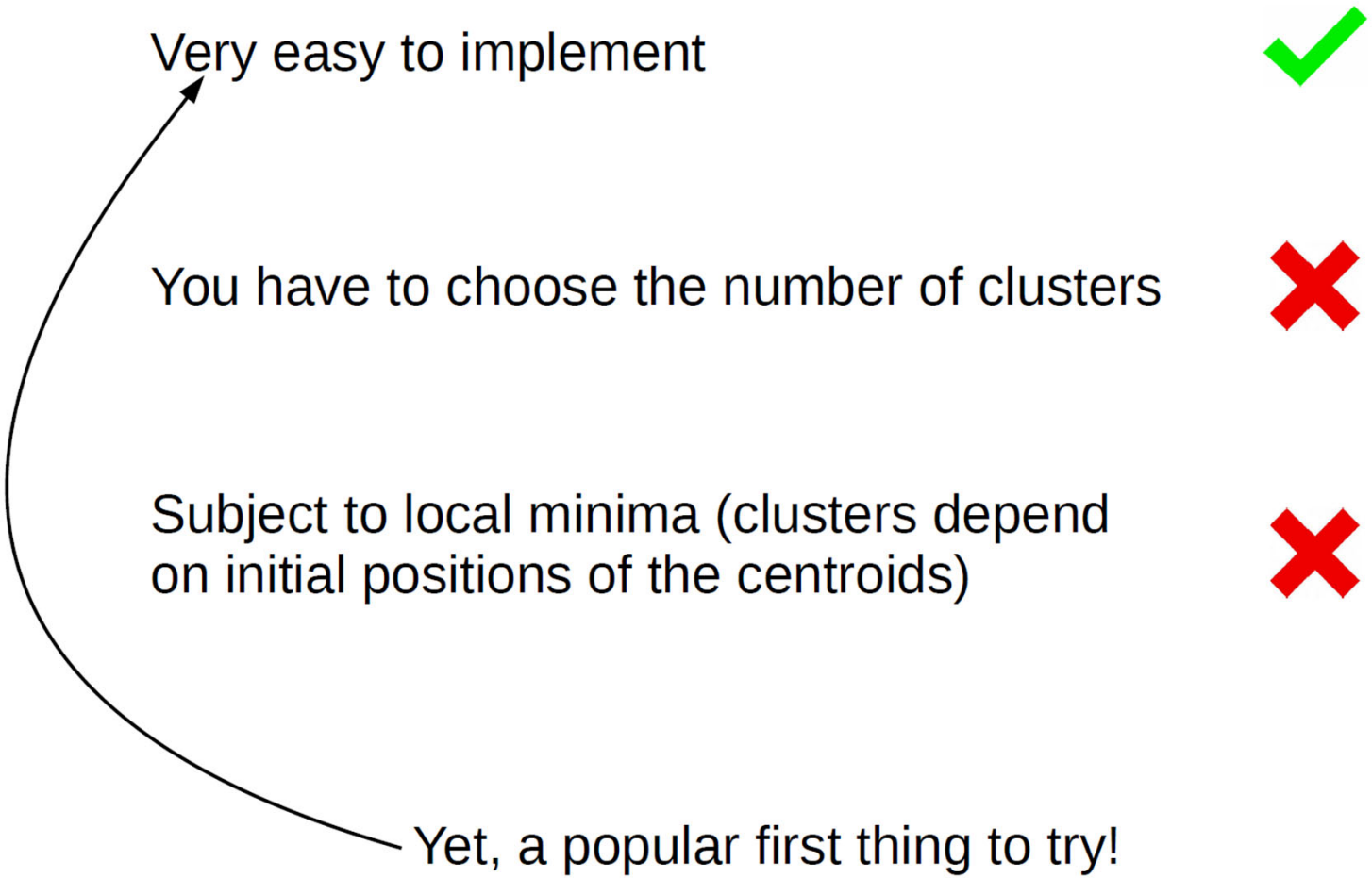
wikipedia

K-means



https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/

K-means


Very easy to implement
You have to choose the number of clusters
Subject to local minima (clusters depend on initial positions of the centroids)
Yet, a popular first thing to try!

Bisecting K-means algorithm

- Bisect (splitting into 2) the dataset into two clusters
- pick a cluster to bisect it in turn
- Each time we do the bisection in several trials (to create a **set of bisection candidates**) with random initial centroids.
- We then pick the pair of clusters candidates that have the lowest errors and we add them to the **list of clusters**.
- We keep bisecting each available cluster from the list of clusters by employing 2-means trails until we reach K-clusters.
- Each time we pick a cluster from the list of clusters to be bisected based on some criterion. A viable criterion is to select the cluster with the highest error. The process is repeated until we reach K-cluster.

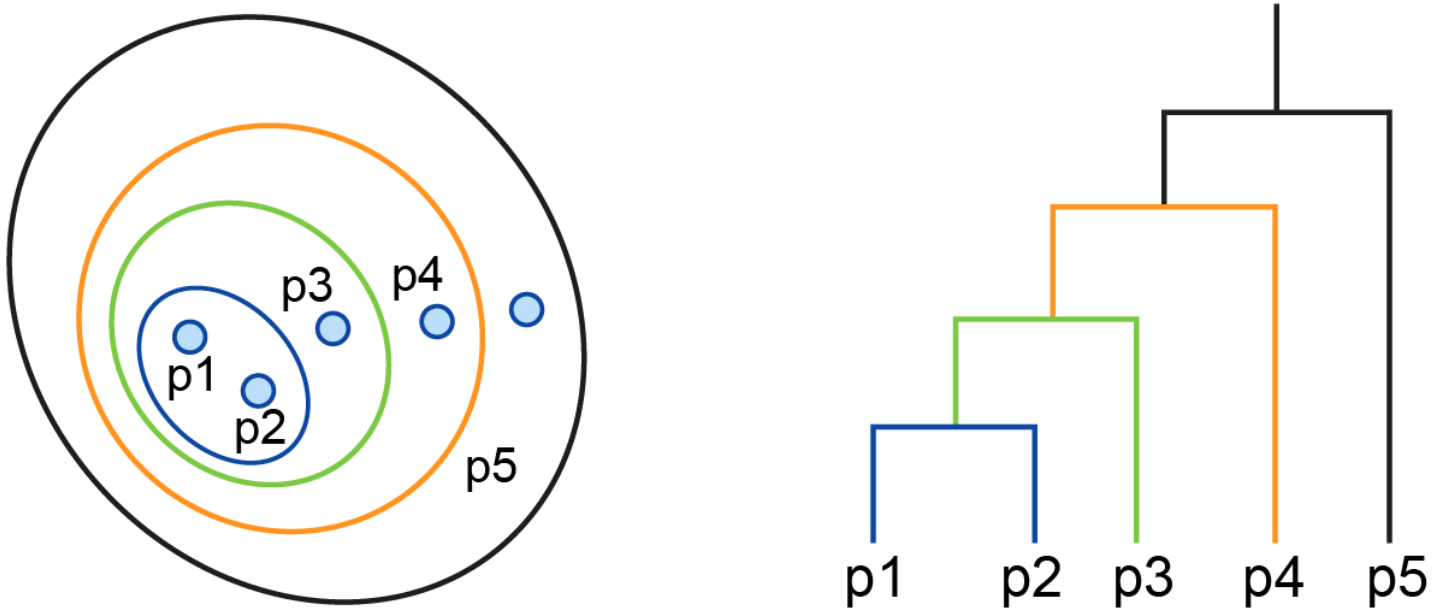
Bisecting K-means algorithm

- Partitional clustering techniques: K-means. The clusters are assumed to be partitional, i.e. they are well-separated from each other.

- Hierarchical clustering assumes that the clusters have a clear hierarchy and they are nested one inside the other.

Hierarchical clustering: agglomerative clusters

- Agglomerative techniques start from smaller clusters and build other larger clusters on top of them in a bottom-up approach.
- Divisive clustering on the other hand, starts with a large cluster that encompasses the entire dataset and works its way towards finer and more refined clusters in a top-down approach.
- Both agglomerative and divisive clustering have similar if not identical results.

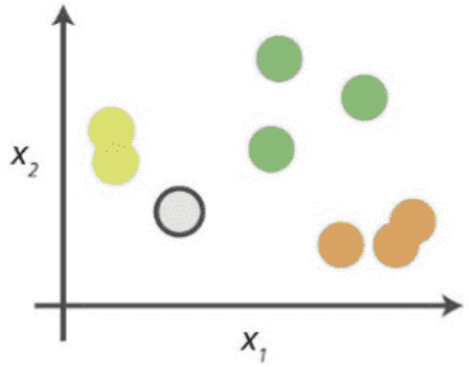


KNN

https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4

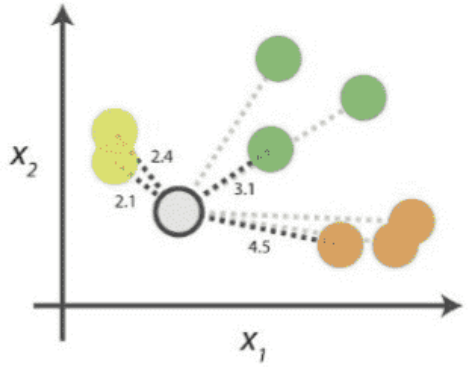
KNN

0. Look at the data



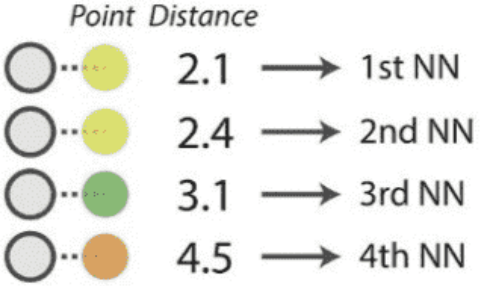
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances



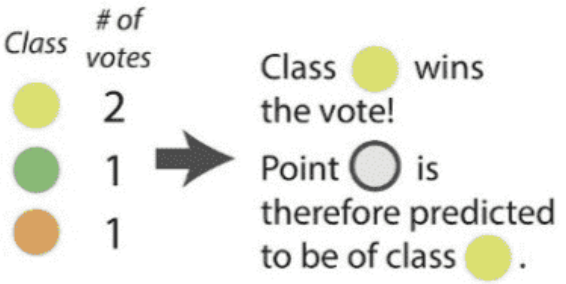
Start by calculating the distances between the grey point and all other points.

2. Find neighbours



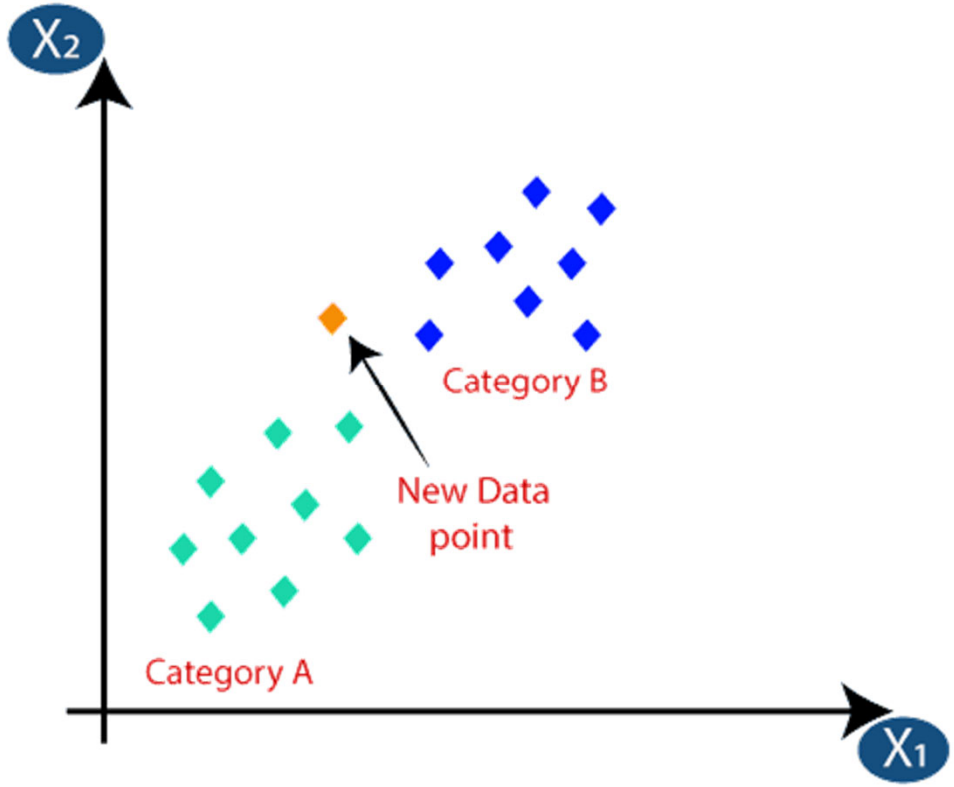
Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

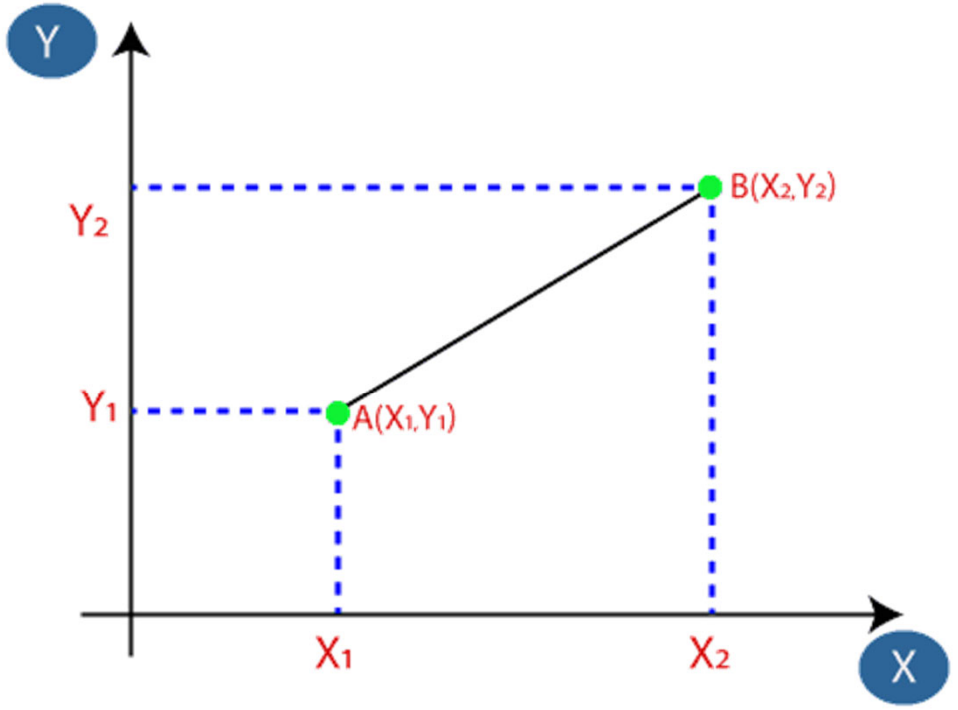


Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

KNN

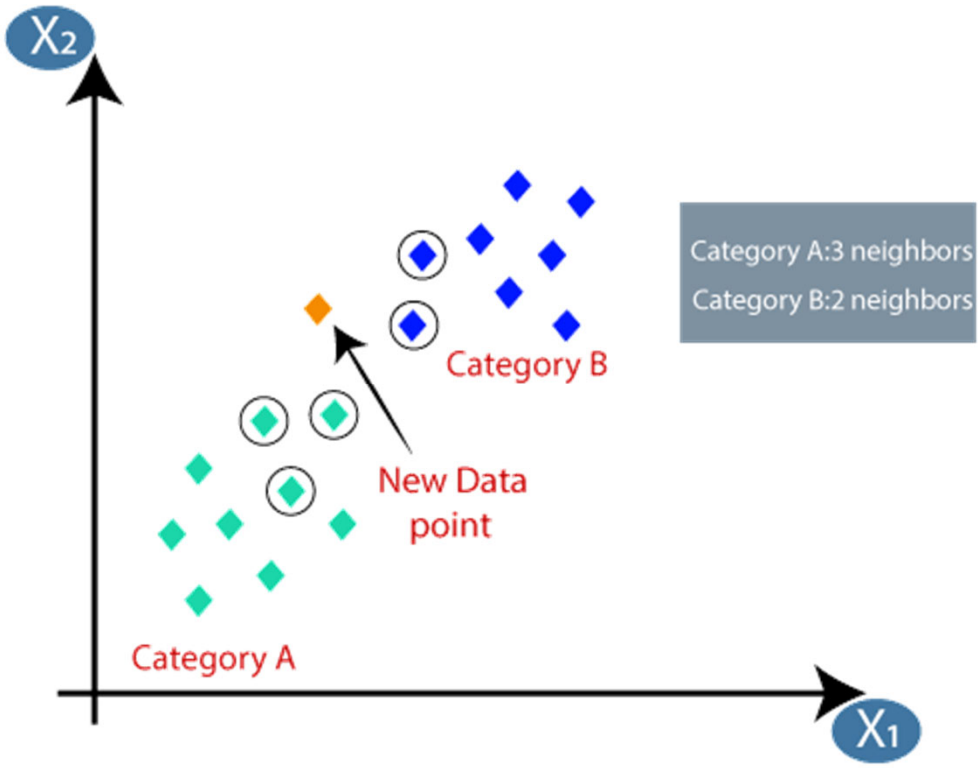


KNN

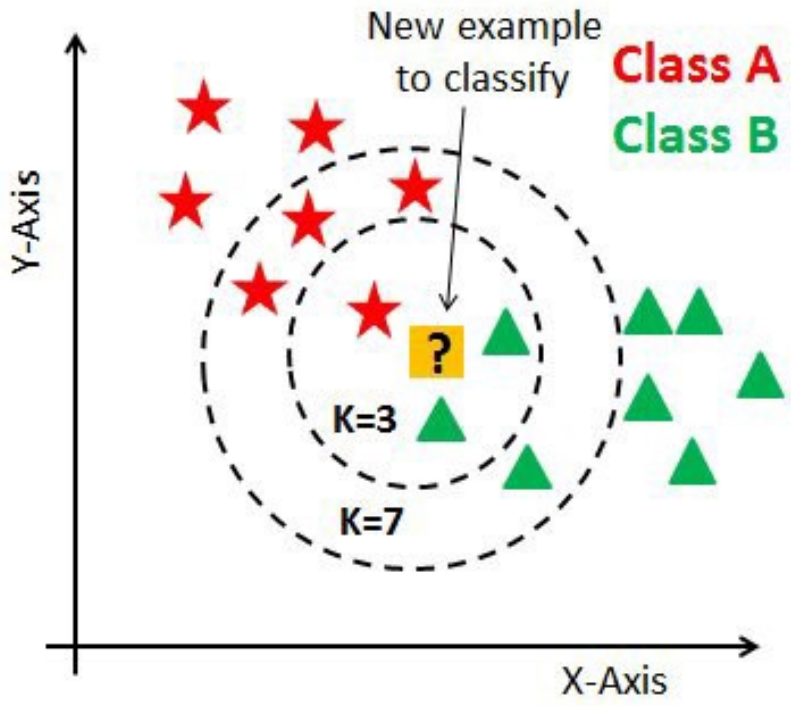


Euclidean Distance between A_1 and B_2 = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

KNN (K=5)

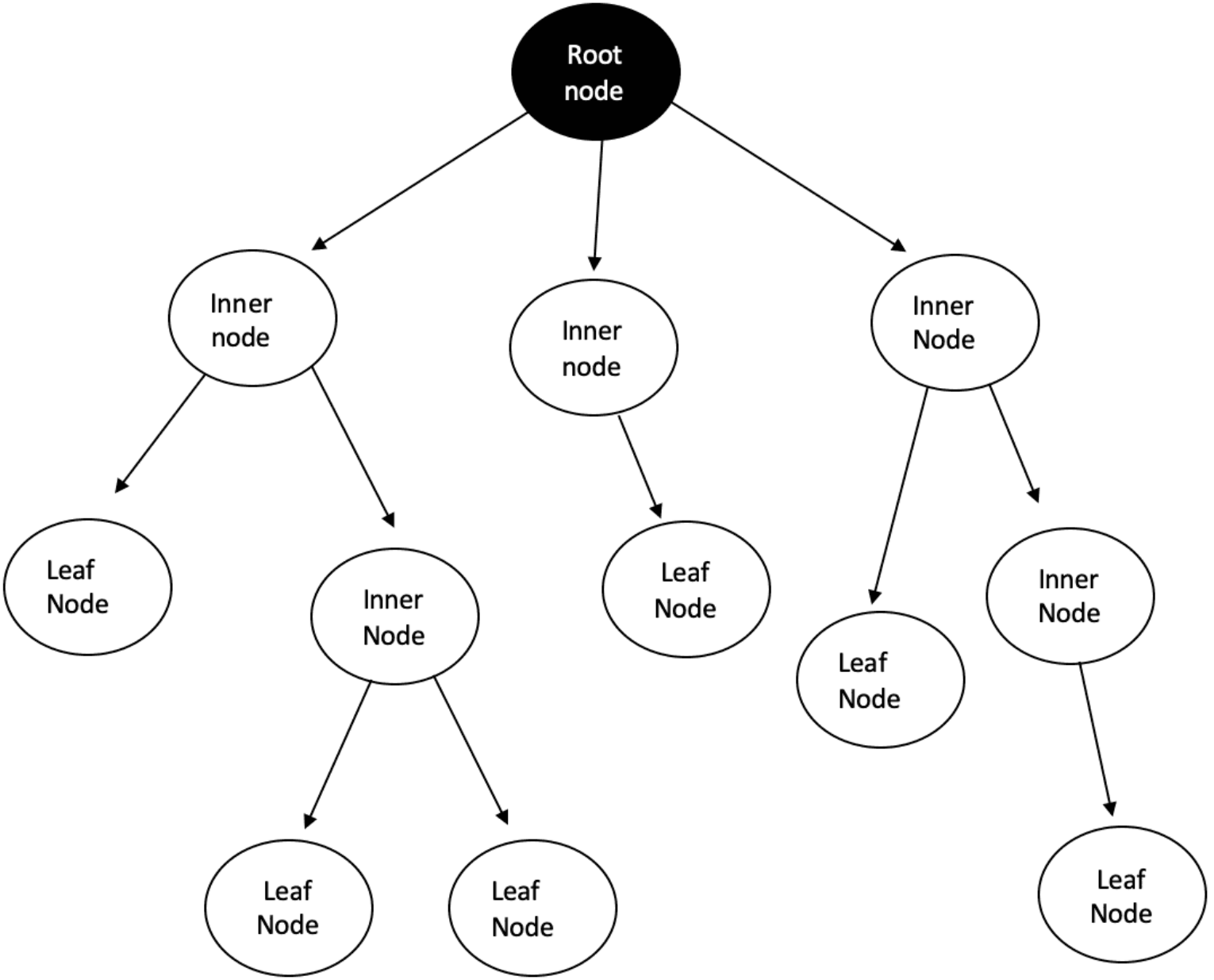

X2
X1
Category A
Category B
New Data point
Category A:3 neighbors
Category B:2 neighbors

KNN (K=?)

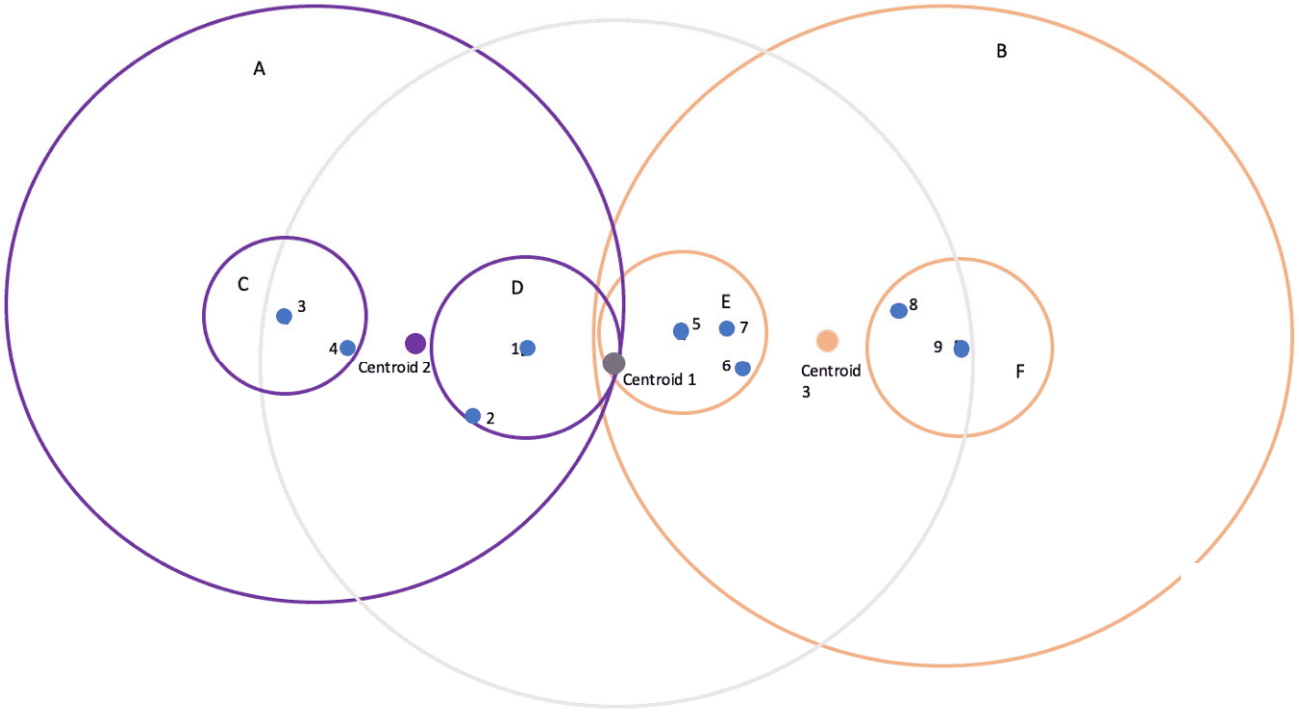


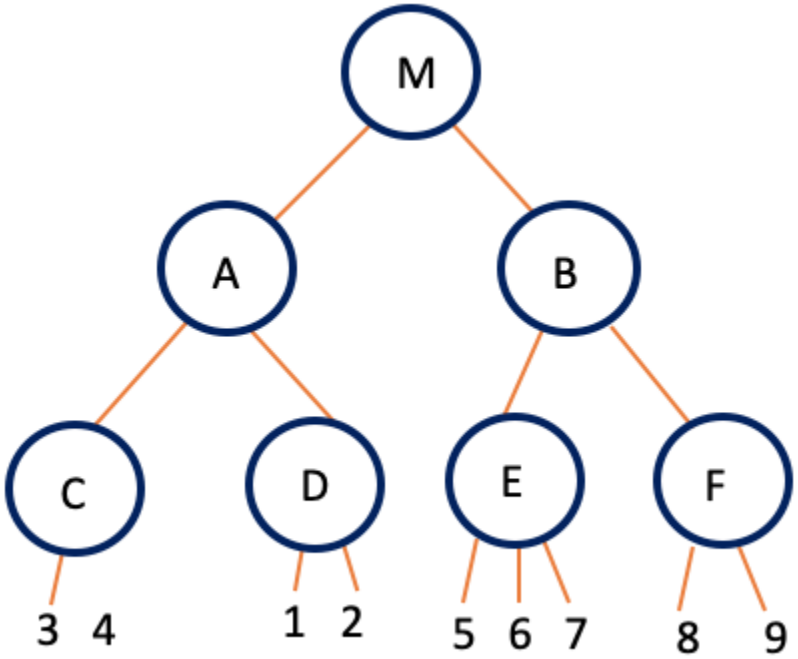
K=3, then class B
K=7, then class A

KNN by KD-tree



KNN by Ball Tree





K-means vs K-nearest neighbors (KNN)

- K-means is an unsupervised learning algorithm used for clustering
- K-nearest neighbors (KNN) is a type of supervised learning algorithm used for both regression and classification.
- KNN tries to predict the correct class for the test data by calculating the **distance** between the test data and all the training points. Then select the K number of points which is closet to the test data.
- The KNN algorithm calculates the probability of the test data belonging to the classes of 'K' training data and class holds the highest probability will be selected.
- In the case of regression, the value is the mean of the 'K' selected training points.

Table 2 Similarity functions

Name	Function formula or measure method	Explanation
Jaccard similarity	$J(A, B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$	1. Measure the similarity of two sets 2. |X| Stands for the number of elements of set X 3. Jaccard distance = 1 − Jaccard similarity
Hamming similarity	The minimum number of substitutions needed to change one data point into the other	The number is smaller, the similarity is more Hamming distance is the opposite of Hamming similarity Especially for the data of string
For data of mixed type	Map the feature into (0, 1) Transform the feature into dichotomous one $S_{ij} = \frac{1}{d}\sum_{l=1}^{d} S_{ijl}$ $S_{ij} = \left(\sum_{l=1}^{d} \eta_{ijl} S_{ijl}\right) / \left(\sum_{l=1}^{d} \eta_{ijl}\right)$	[3,4]

Category	Typical algorithm
Clustering algorithm based on partition	K-means, K-medoids, PAM, CLARA, CLARANS
Clustering algorithm based on hierarchy	BIRCH, CURE, ROCK, Chameleon
Clustering algorithm based on fuzzy theory	FCM, FCS, MM
Clustering algorithm based on distribution	DBCLASD, GMM
Clustering algorithm based on density	DBSCAN, OPTICS, Mean-shift
Clustering algorithm based on graph theory	CLICK, MST
Clustering algorithm based on grid	STING, CLIQUE
Clustering algorithm based on fractal theory	FC
Clustering algorithm based on model	COBWEB, GMM, SOM, ART