**ML – K-means algorithm, last part of unsupervised learning**

We’ve imposed the objective of gaining variance from data, this changes with clustering

We would like to derive a partition of the data in groups (clusters), with each group sharing at least one characteristic that sets it apart from other groups

Centroid: mean vector of a cluster of data points in a high-dimensional space

It’s not clear how the **number** of clusters is chosen yet, making clustering **ill-posed**

The definition of k-means comes from the fact that the algo looks for a fixed *k* number of clusters (mainly, of centroids needed in the dataset), and finds the centroids (mainly, finding the *means* of data)

It aims to keep the centroids as small as possible while allocating data points to nearest clusters

The k-means algorithm is NP-Hard, so it’s not optimal; we need to find:

* A set of centroids
* A set of assignment labels that create a mapping from data points to centroids

Given an arbitrarily given number of clusters to find (namely, *k*), and a random sample of the available centroids (using heuristic-like guessing)

The data will be clustered based on **smallest distance from centroids** (i.e. minimize within-cluster variance, while maximizing inter-cluster variance)