International College of Economics and Finance

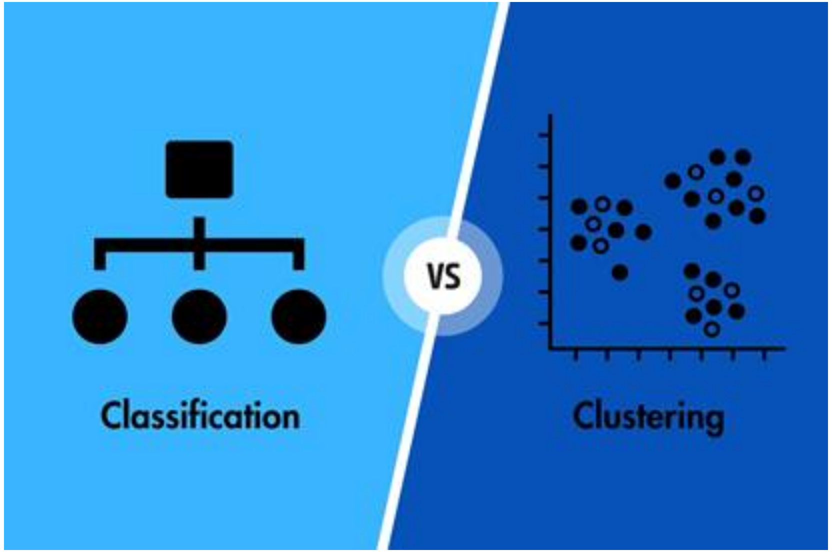
Clustering Methods in Machine Learning

Prepared by:
Rafaelyan George
Moscow, 2023

Clustering and classification

Classification is supervised:
 -Discrete, finite, known number of classes with known features
 -deterministic rigid models
 -stationary models

Clustering is unsupervised:
 -Unknown number of classes with unknown features
 -High-dim data
 -Big Data



Why is clustering useful?

Anomaly and Outlier detection	Identify areas in your data that are not representative or indicate the necessity of changes to the model
Segmentation	Divide your data into segments to use different models for better overall prediction power
Dimensionality Reduction	Overcome the curse of dimensionality by identifying low-variance dimensions/ dimensions that don't contribute to clusters and get rid of them

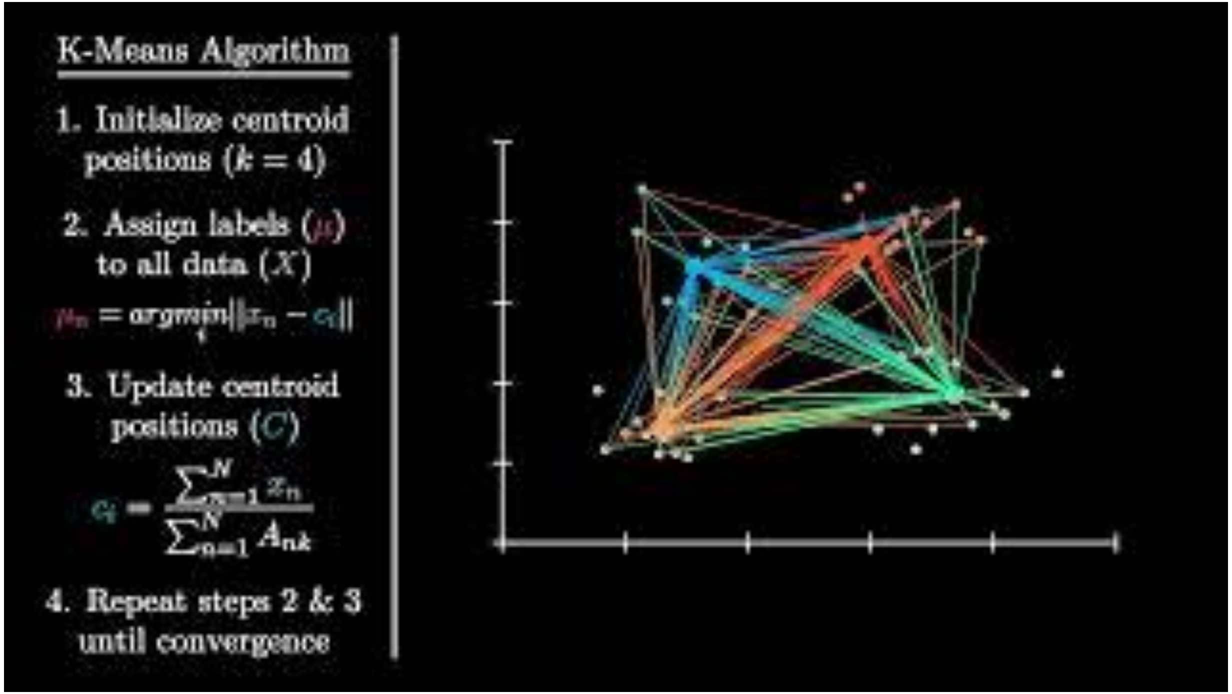
Algorithms to be covered today

K-means	The fundamental algorithm to all ML
DBSCAN, HDBSCAN, OPTICS	A popular algorithm for data clustering and its less popular more specialised variations
Affinity propagation	A rather exotic algorithm for some special data cases

K-means clustering

Steps to K-means clustering:
1)Choose the number of clusters (K) and repetitions (N)
2)Select K random points and denote them as initial cluster points
3)Assign all other points to the nearest initial cluster points
4)Calculate the mean of each cluster and denote each mean as the new initial cluster point
5)Recluster until clusters don’t change
6)Repeat the process N times and choose clustering with lowest sum of variances in each cluster

K-means clustering



https://www.youtube.com/watch?v=2lZZ_FzIlJY&pp=ygUVayBtZWFucyB2aXN1YWxpc2F0aW9u

DBSCAN, HDBSCAN, OPTICS

Steps to DBSCAN:

1)State radius (eps) and number of neighbours (n)

2)Select all points that have n or more neighbours within a euclidean distance of eps and mark them as core points

3)Select a random core point and assign it to a new cluster

4)Select all core points within the euclidean distance of eps from the previous point and assign them to the same cluster

5)Repeat the last step until no core points, then assign all non-core points to the cluster if the euclidean distance to the nearest core point that is a part of that cluster is less or equal eps

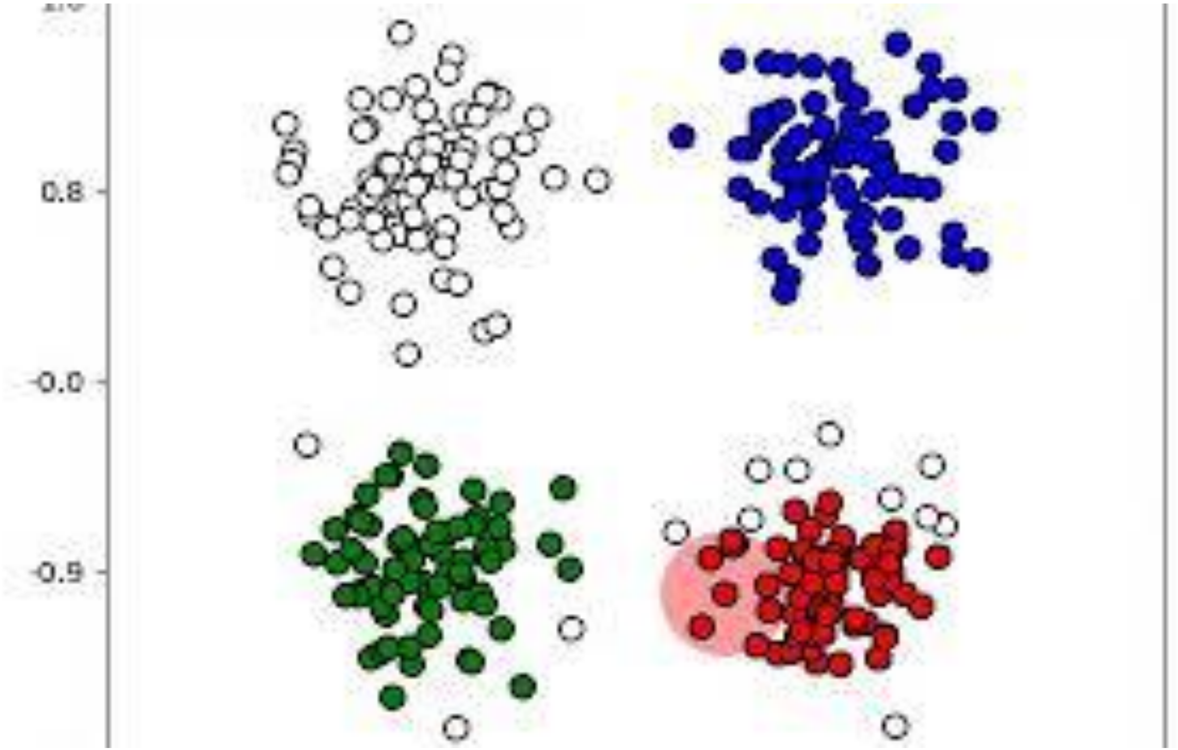
6)Repeat step 3 until no more core points left

7)Mark all non-core points that aren’t part of any cluster as separate clusters/ outliers

DBSCAN, HDBSCAN, OPTICS

DBSCAN	Plots clusters according to a preset parameter of the euclidean distance
OPTICS	Plots clusters according to a preset change in the density of neighbours
HDBSCAN	Plots clusters according to a best guess for the euclidean distance that arises from using all possible values of it

DBSCAN, HDBSCAN, OPTICS



https://www.youtube.com/watch?v=Idy6xQ5YQ7I

Affinity propagation

Steps to Affinity propagation:

1)Define 2 variables r(i, k) - responsibility and a(i, k) - accumulated evidence for each point, both values are initially zero

2)$r(i, k) \leftarrow s(i, k) - \max[a(i, k') + s*i, k') \ \forall \ k' \neq k]$ where s(i, k) is the negative of the euclidean distance between i and k

3)$a(i, k) \leftarrow \min[0, r(k,k) + \sum r(i', k)$ s.t. $i' \notin \{i, k\}]$

4)Measure convergence of both r(i, k) and a(i, k) via:

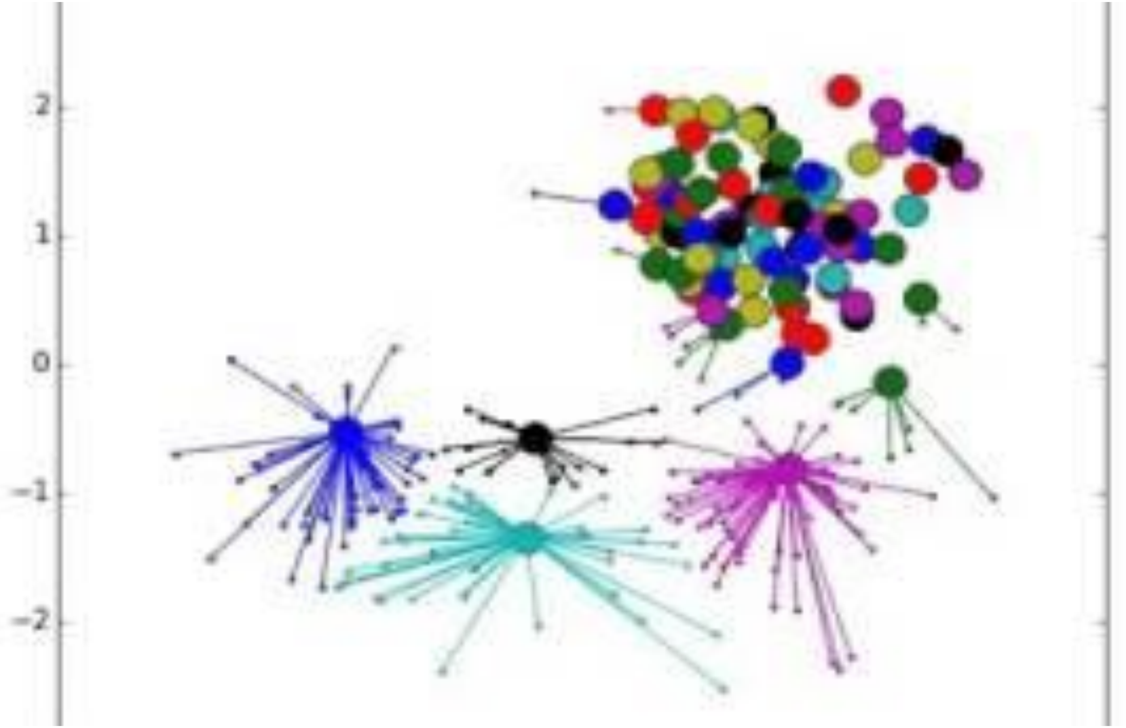
$r_{t+1}(i, k) = \lambda * r_t(i, k) + (1 - \lambda) * r_{t+1}(i, k)$, λ is user-defined

$a_{t+1}(i, k) = \lambda * a_t(i, k) + (1 - \lambda) * a_{t+1}(i, k)$, λ is user-defined

5)Repeat 2, 3 and 4 until convergence for each element

6)Choose points that satisfy $r(k, k) + a(k, k) \geqslant 0$ as exemplars and assign other points to them based on argmax[s(i, k)]

Affinity propagation



https://www.youtube.com/watch?v=NaldkmCouLw

HSE
University

ICEF

М И Э Ф