

Clustering algorithms (*Algorithmes de partitionnement*)

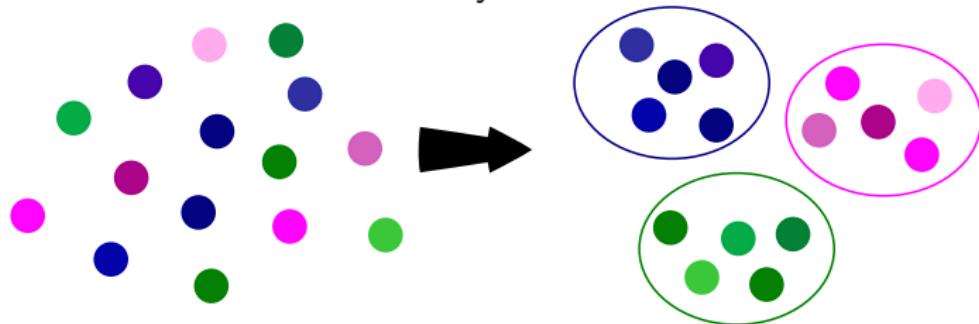
Diane Lingrand and many contributors



Polytech SI

2023 - 2024

- **Unsupervised learning** : no class label needed.
- Find the class labels directly from the data.



- Cluster : a collection of data objects
 - Similar to one another within the same cluster
 - Dissimilar to the objects in other clusters
 - Need a way to calculate object similarity/distance
- Typical applications :
 - As a stand-alone tool to get insight into data distribution
 - As a preprocessing step for other algorithms

What Is Good Clustering ?

- A good clustering method will produce high quality clusters with
 - high intra-class similarity
 - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

- 1 Similarity - distance
- 2 Partitionning approaches to clustering
- 3 Hierarchical clustering methods
- 4 Density based clustering methods
- 5 Evaluation

- images from six classes :
 - muffin,
 - bagel,
 - fried chicken,
 - chihuahua,
 - labradoodle and
 - puppy
- How would you do a binary clustering of these images ?

- images from six classes :
 - muffin,
 - bagel,
 - fried chicken,
 - chihuahua,
 - labradoodle and
 - puppy
- How would you do a binary clustering of these images ?
 - food : muffin, bagel and fried chicken
 - pet : chihuahua, labradoodle and puppy

Similarity : an example

- food : muffin, bagel and fried chicken
- pet : chihuahua, labradoodle and puppy



Choosing the distance is of crucial importance !

- Standardize data

- Calculate the mean absolute deviation : $s_j = \frac{1}{n} \sum_{i=1}^n |x_{ij} - \mu_j|$ where

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$$

- Calculate the standardized measurement (z-score) : $z_{ij} = \frac{x_{ij} - \mu_j}{s_j}$
- Using mean absolute deviation is more robust than using standard deviation
- A classical example ($\mu_{age} = 60$; $\mu_{salary} = 11074$; $s_{age} = 5$; $s_{salary} = 37$) :

	age	salary
alice	50	11000
bob	70	11100
charly	60	11122
dany	60	11074

⇒

	age	salary
alice	-2	-2
bob	2	0.7
charly	0	1.3
dany	0	0

Why standardizing data ?

	age	salary
alice	50	11000
bob	70	11100
charly	60	11122
dany	60	11074

Manhattan distance	alice	bob	charly	dany
alice	0	120	132	84
bob	120	0	32	36
charly	132	32	0	48
dany	84	36	48	0

Why standardizing data ?

	age	salary
alice	50	11000
bob	70	11100
charly	60	11122
dany	60	11074

	age	salary
alice	-2	-2
bob	2	0.7
charly	0	1.3
dany	0	0

Manhattan distance	alice	bob	charly	dany
alice	0	120	132	84
bob	120	0	32	36
charly	132	32	0	48
dany	84	36	48	0

Manhattan distance	alice	bob	charly	dany
alice	0	6.7	5.3	4
bob	6.7	0	2.6	2.7
charly	5.3	2.6	0	1.3
dany	4	2.7	1.3	0

Why standardizing data ?

	age	salary
alice	50	11000
bob	70	11100
charly	60	11122
dany	60	11074

	age	salary
alice	-2	-2
bob	2	0.7
charly	0	1.3
dany	0	0

Manhattan distance	alice	bob	charly	dany
alice	0	120	132	84
bob	120	0	32	36
charly	132	32	0	48
dany	84	36	48	0

Manhattan distance	alice	bob	charly	dany
alice	0	6.7	5.3	4
bob	6.7	0	2.6	2.7
charly	5.3	2.6	0	1.3
dany	4	2.7	1.3	0

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
Xscaled = scaler.fit_transform(X)
```

- Minkowski distance

$$d_q(i, j) = \sqrt[q]{\sum_{k=1}^m |x_{ik} - x_{jk}|^q}$$

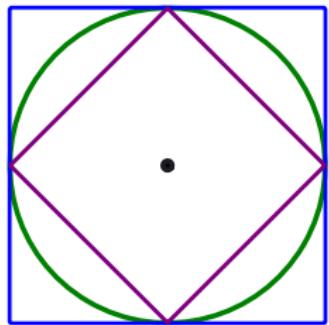
- if $q = 1$: Manhattan distance

$$d_1(i, j) = \sum_{k=1}^m |x_{ik} - x_{jk}|$$

- if $q = 2$: Euclidean distance

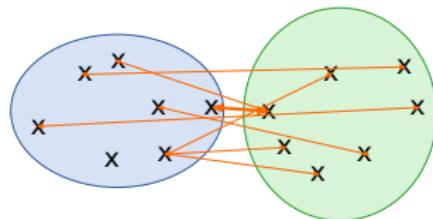
$$d_2(i, j) = \sqrt{\sum_{k=1}^m |x_{ik} - x_{jk}|^2}$$

- if $q = \infty$: Max. distance

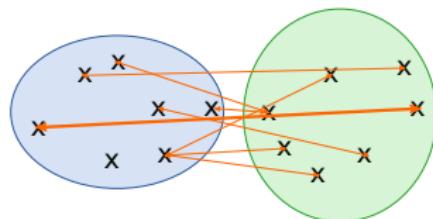


Distances between clusters (1)

- Single linkage : smallest distance between an element in one cluster and an element in the other, i.e., $d(K_p, K_q) = \min_{x_i \in K_p, x_j \in K_q} d(x_i, x_j)$



- Complete linkage : largest distance between an element in one cluster and an element in the other, i.e., $d(K_p, K_q) = \max_{x_i \in K_p, x_j \in K_q} d(x_i, x_j)$

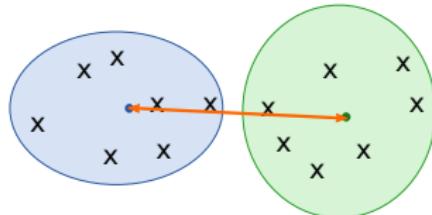


- Average linkage : average distance between an element in one cluster and an element in the other

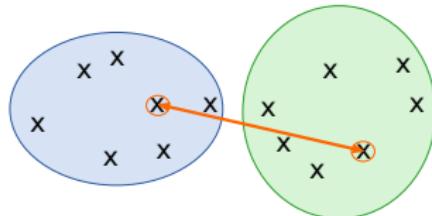
Distances between clusters (2)

- Centroid : distance between the centroids of two clusters, i.e.,
 $d(K_p, K_q) = d(C_p, C_q)$

- centroid = midpoint of a cluster : $C_p = \frac{1}{N_p} \sum_{i=1}^{N_p} x_i^{(p)}$



- Medoid : distance between the medoids of two clusters, i.e.,
 $d(K_p, K_q) = d(M_p, M_q)$
- Medoid : one chosen, centrally located object in the cluster



- Ward : sum of squared distances within all clusters

- Partitioning (iterative construction of partitions)
 - k-Means, k-Medoids, etc.
- Hierarchical (construct a dendrogram of instances)
 - Diana, Agnes, BIRCH, ROCK, CAMELEON
- Density-Based (based on connectivity and density function)
 - DBSCAN, OPTICS, DenClue
- Grid-Based
 - STING, WaveCluster, CLIQUE
- Model-Based
 - Expectation Maximization
 - Self-organizing maps (SOM)
- Spectral clustering
- Frequent-Pattern-Based

- **Partitioning** (iterative construction of partitions)
 - k-Means, k-Medoids, etc.
- **Hierarchical** (construct a dendrogram of instances)
 - Diana, Agnes, BIRCH, ROCK, CAMELEON
- **Density-Based** (based on connectivity and density function)
 - DBSCAN, OPTICS, DenClue
- Grid-Based
 - STING, WaveCluster, CLIQUE
- Model-Based
 - Expectation Maximization
 - Self-organizing maps (SOM)
- Spectral clustering
- Frequent-Pattern-Based

- 1 Similarity - distance
- 2 Partitionning approaches to clustering
- 3 Hierarchical clustering methods
- 4 Density based clustering methods
- 5 Evaluation

- Partitioning method : Construct a partition of a dataset D of n objects into a set of k clusters, minimizing the sum of squared distances :

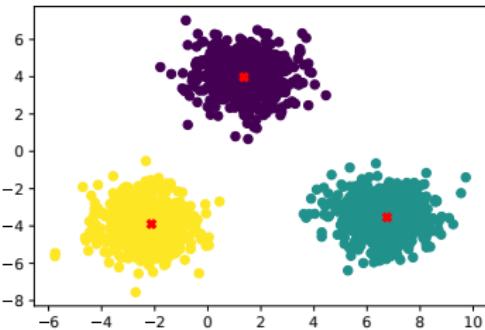
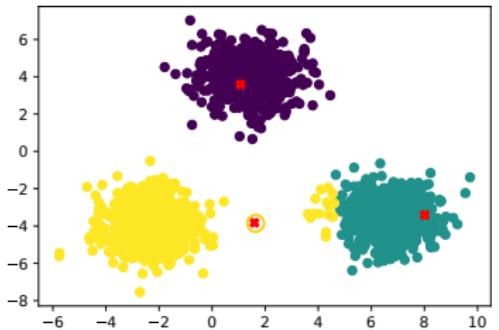
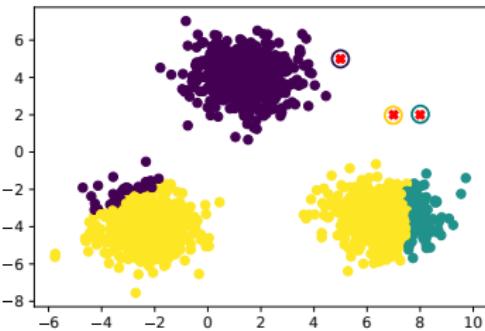
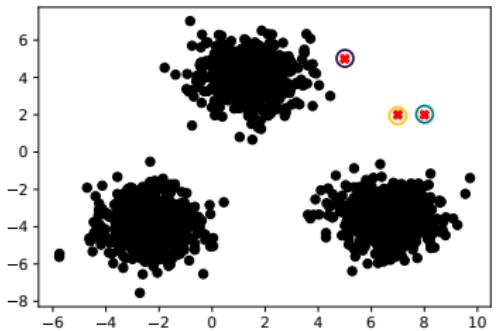
$$\sum_{p=1}^k \sum_{i=1}^{N_p} (x_i^{(p)} - C_p)^2$$

- Given a k, find a partition of k clusters that optimizes the chosen partitioning criterion
 - Global optimal : exhaustively enumerate all partitions
 - Heuristic methods : k-means and k-medoids algorithms
 - k-means (MacQueen '67) : Each cluster is represented by the centroid of the cluster
 - k-medoids or PAM (Partition around medoids) (Kaufman and Rousseeuw '87) : Each cluster is represented by one of the objects in the cluster

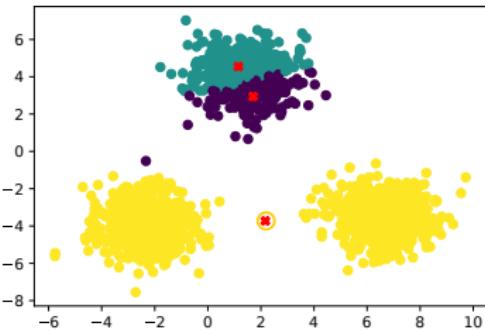
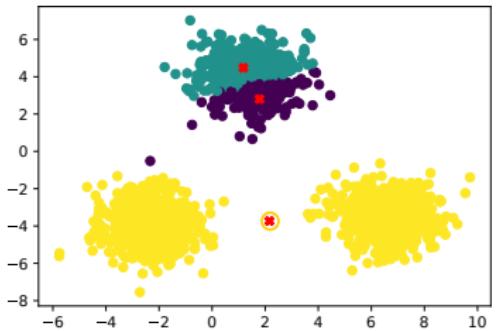
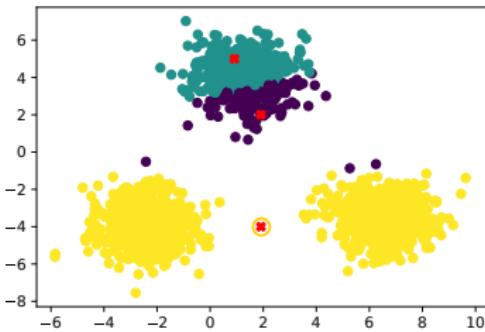
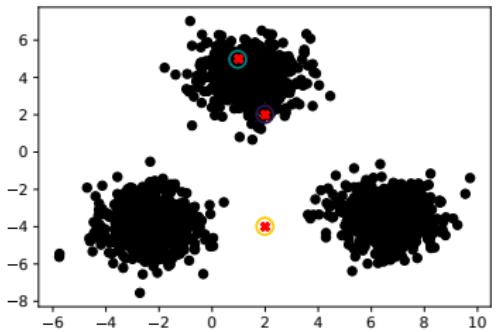
Given k , the k-means algorithm is implemented in four steps :

- ① Choose k initial centroids (could be samples), each labeled as k
- ② Assign each object to the cluster having the nearest centroid point
- ③ Compute centroids of the clusters of the current partition (the centroid is the center, i.e., mean point, of the cluster)
- ④ Go back to Step 2, stop when no more change

Example of k-means

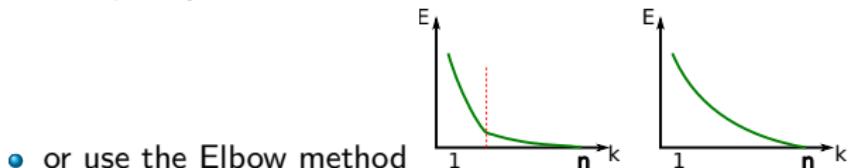


Example of k-means : bad initialisation



k-mean advantages and drawbacks

- Strength : Relatively efficient : $O(tkn)$, where n nb. of objects, k nb. of clusters, and t nb. of iterations. Normally, $k, t \ll n$.
- Comment : Often terminates at a local optimum.
 - The global optimum may be found using optimization methods such as : simulated annealing and evolutionary algorithms.
 - In practice : several trials using aleatory initialisations and some heuristics (e.g. k-mean++)
- Weaknesses :
 - Applicable only when mean is defined, then what about categorical data ?
 - Need to specify k , the number of clusters, in advance

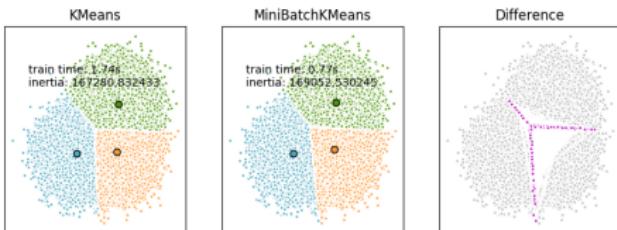


• or use the Elbow method

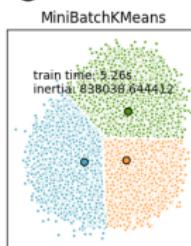
- Unable to handle noisy data and outliers
- Not suitable to discover clusters with non-convex shapes

mini-batch k-mean

- a variant of k-mean for scalability improving
- mini-batch = subset of dataset
 - randomly sampled at each iteration of the training process
- results almost as good as the standard algorithm
 - code from https://scikit-learn.org/stable/auto_examples/cluster/plot_mini_batch_kmeans.html#sphx-glr-auto-examples-cluster-plot-mini-batch-kmeans-py with modification of the number of data to 200000 (upper limit to have k-means running).

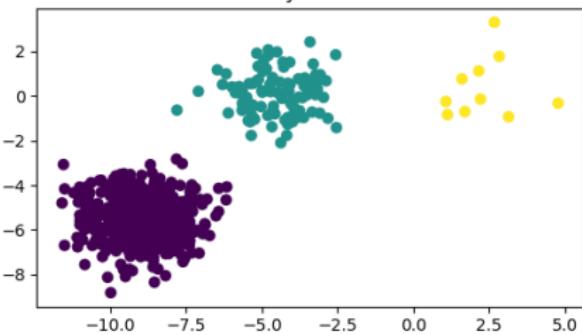
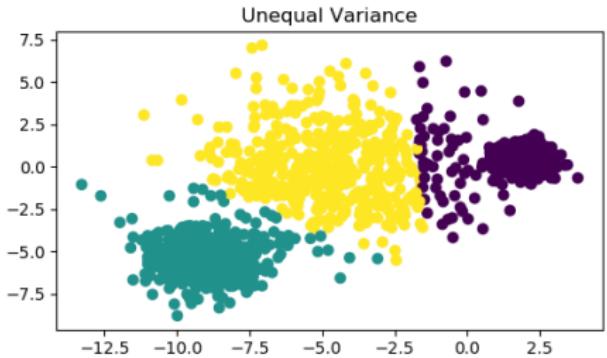
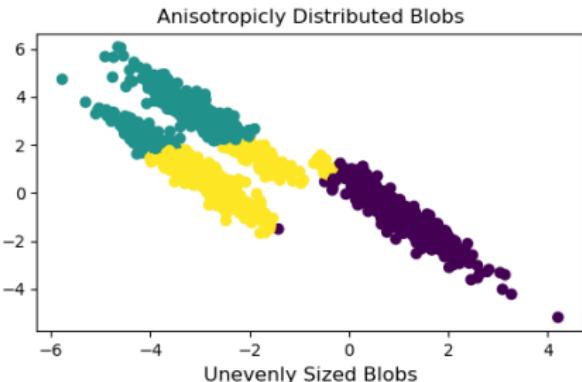
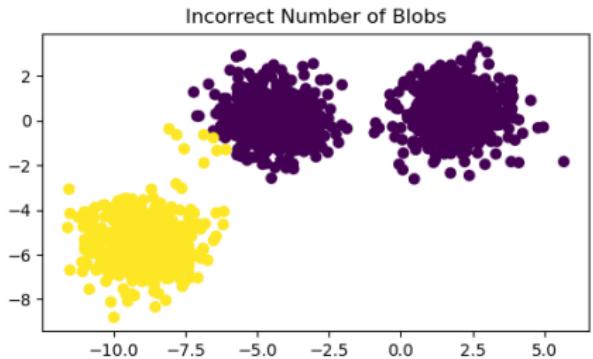


- Only with mini-batch and using 1 million of data :

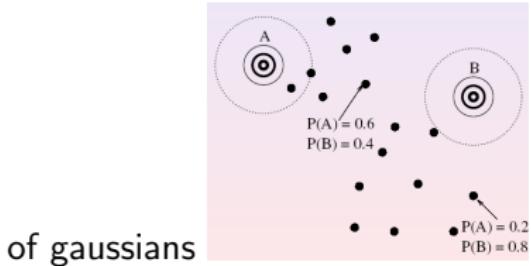


Demonstration of k-means assumptions

https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_assumptions.html

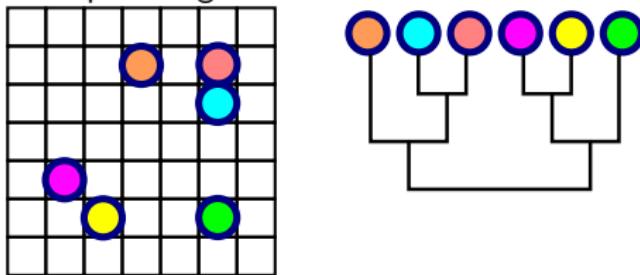


- k-medoids (PAM) :
 - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - less sensitive to outliers than k-mean
 - PAM works effectively for small data sets, but does not scale well for large data sets
- C-mean (fuzzy k-mean)
 - a data can belong to more than one cluster to a degree (soft vs hard assignment)
- GMM : Gaussian Mixture Model
 - Based on the assumption that the distribution of samples is a mixture



- 1 Similarity - distance
- 2 Partitionning approaches to clustering
- 3 Hierarchical clustering methods
- 4 Density based clustering methods
- 5 Evaluation

- Input : distance matrix - Output : a dendrogram (tree of clusters)
- This method does not require the number of clusters k as an input, but may need a termination condition
- Two approaches :
 - Bottom up approach = agglomerative
 - Example using Manhattan distance and complete linkage

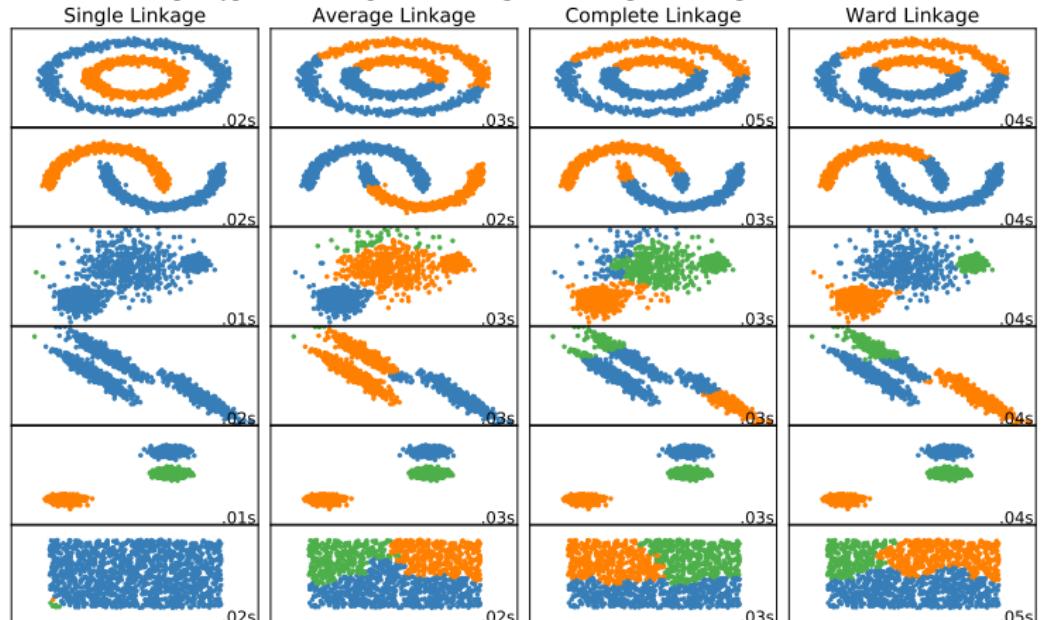


- Top down approach (DIANA = DIvide ANALysis)
 - start with all samples in one cluster. Hierarchical division of clusters related to a dispersion criterion (divide into clusters where closest samples are the more distant ones).

Impact of the linkage criteria on agglomerative clustering

From <https://scikit-learn.org/stable/modules/clustering.html#>

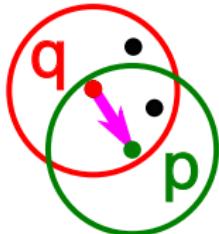
different-linkage-type-ward-complete-average-and-single-linkage.



- 1 Similarity - distance
- 2 Partitionning approaches to clustering
- 3 Hierarchical clustering methods
- 4 Density based clustering methods
- 5 Evaluation

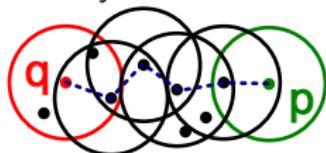
- Clusters : areas of high density separated by areas of low density
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features :
 - Discover clusters of arbitrary shape
 - Handle noise
 - One scan
 - Need density parameters as termination condition
- Examples of density-based methods :
 - DBSCAN, OPTICS, DENCLUE, CLIQUE

- Two parameters :
 - ϵ : Maximum radius of the neighbourhood
 - MinPts : Minimum number of points in an ϵ -neighbourhood of that point
- Neighborhood : $N_\epsilon(p) = \{q \in D | d(p, q) \leq \epsilon\}$
- Core point : if at least MinPts points are within distance ϵ of p (including p).
- Directly density-reachable : A point p is directly density-reachable from a point q w.r.t. ϵ , MinPts if :
 - p belongs to $N_\epsilon(q)$
 - core point condition : $|N_\epsilon(q)| \geq \text{MinPts}$



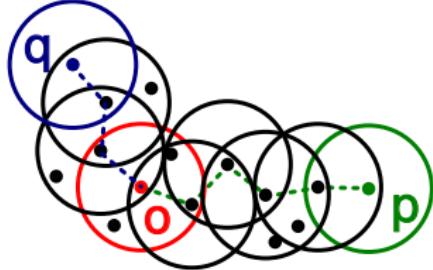
- Density-Reachable :

- A point p is density-reachable from a point q w.r.t. ϵ , MinPts if there is a chain of points p_1, \dots, p_n , $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

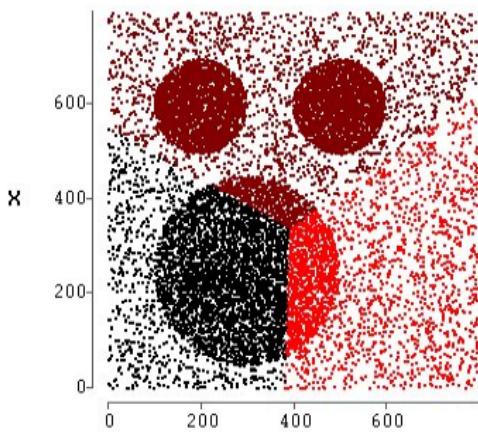


- Density-Connected :

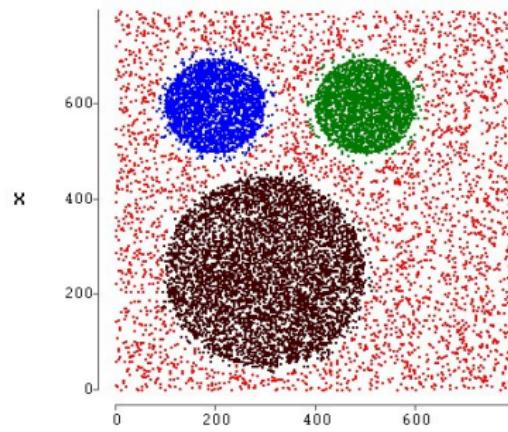
- A point p is density-connected to a point q w.r.t. ϵ , MinPts if there is a point o such that both, p and q are density-reachable from o w.r.t. ϵ and MinPts



- choose a core point p , not already assigned to a cluster
- add all density-reachable points from p to this cluster
- go back to step 1 until no more points without a cluster
- all points not reachable from any other point are outliers or noise points



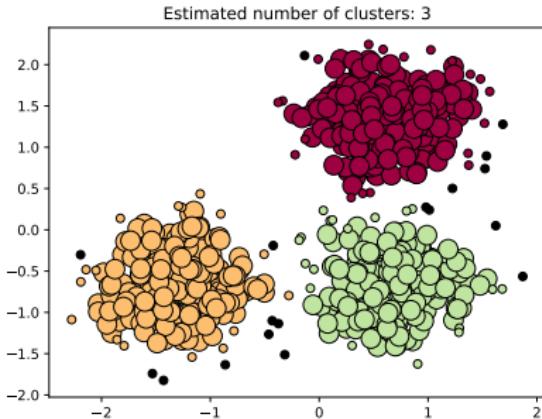
using k-means



using DBSCAN

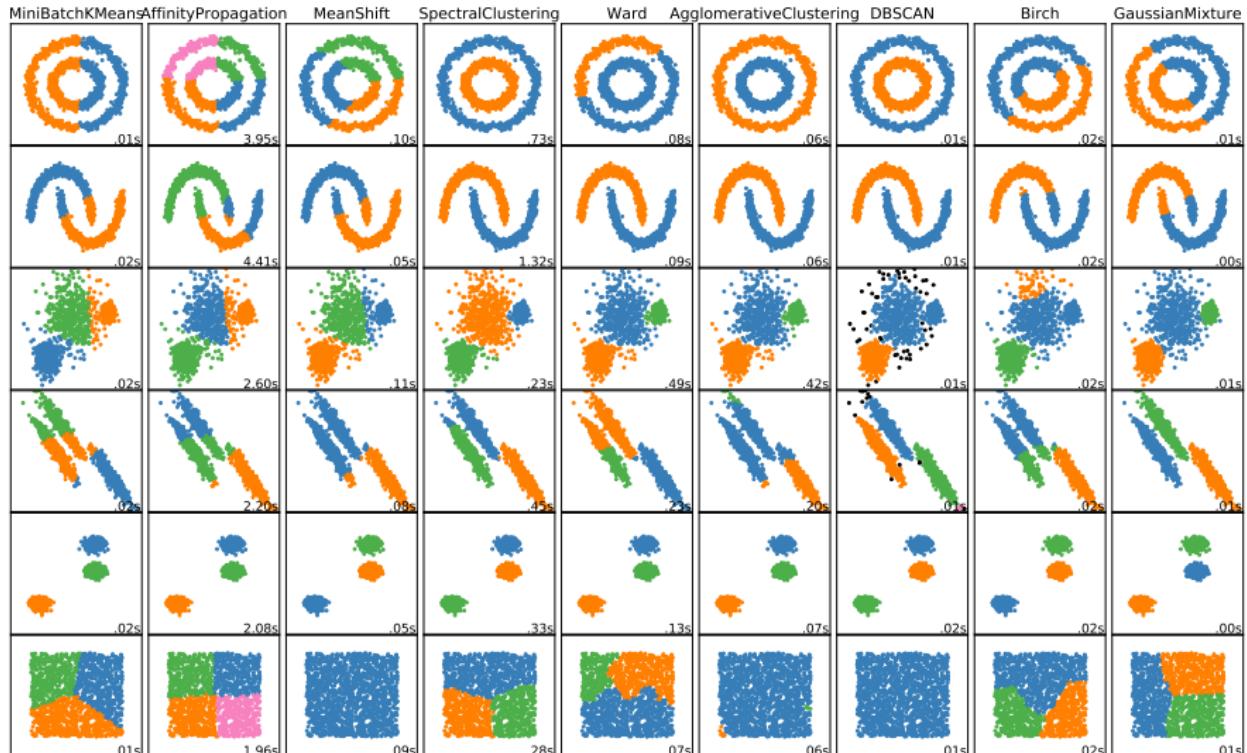
DBSCAN in scikit-learn

```
from https://scikit-learn.org/stable/auto_examples/cluster/plot_dbSCAN.html#  
sphx-glr-download-auto-examples-cluster-plot-dbscan-py
```



```
from sklearn.cluster import DBSCAN  
db = DBSCAN(eps=0.3, min_samples=10)
```

- 1 Similarity - distance
- 2 Partitionning approaches to clustering
- 3 Hierarchical clustering methods
- 4 Density based clustering methods
- 5 Evaluation



- ground truth known
 - ARI (Adjusted Rand Index)
 - MI (Mutual Information)
 - V-measure
- ground truth not known
 - Davies-Bouldin index
 - Dunn index
 - Calinski-Harabaz
 - silhouette

- Internal validation : Separation ? Compactness ?
 - E.g. Dunn, Davies-Bouldin, Calinski-Harabaz, and Silhouette indexes.
 - Problems :
 - Different performance WRT existence of noise, variable densities, and non well-separated clusters.
 - Overrate the algorithm that uses the same clustering model.

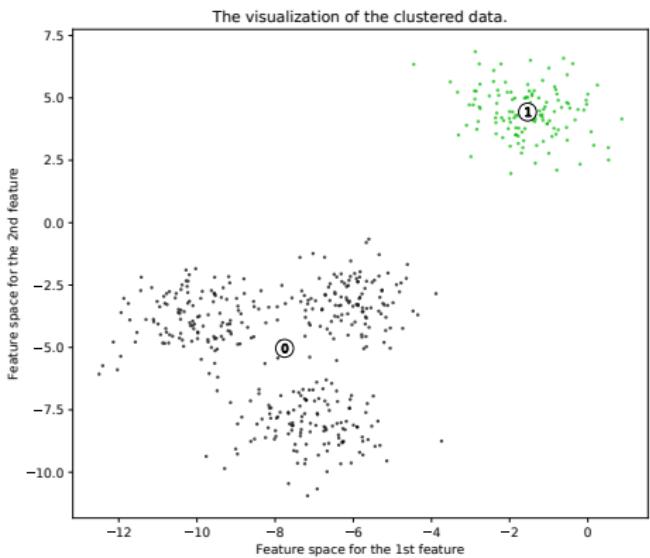
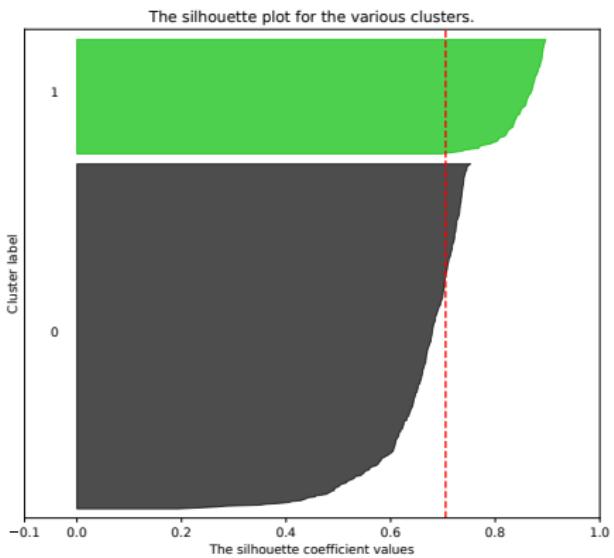
number of clusters	average silhouette score	Calinski Harabaz score
2	0.71	1604
3	0.59	1810
4	0.65	2704
5	0.56	2263
6	0.45	2043

- for a single sample : Silhouette coefficient $s = \frac{b-a}{\max(a,b)}$ with
 - a : mean distance between a sample and all other samples from the same cluster
 - b : mean distance between a sample and all other samples in the next nearest cluster
- for a dataset : mean of Silhouette coefficients for each sample
- sklearn : `metrics.silhouette_score`

k-means with $k = 2$ and silhouette profiles

inspired by https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

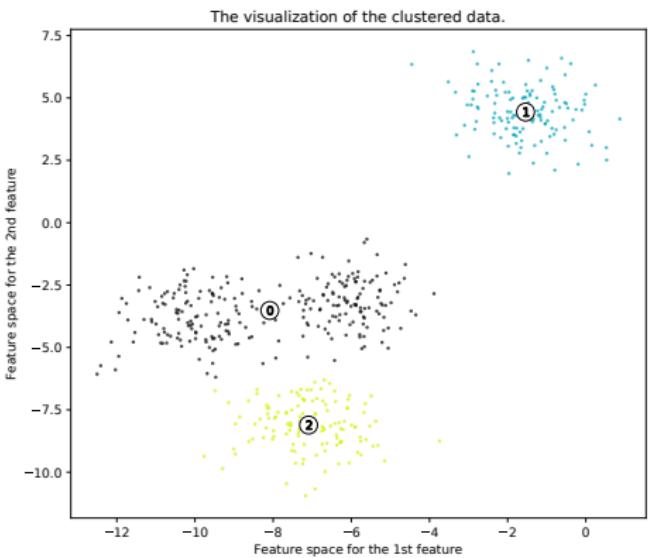
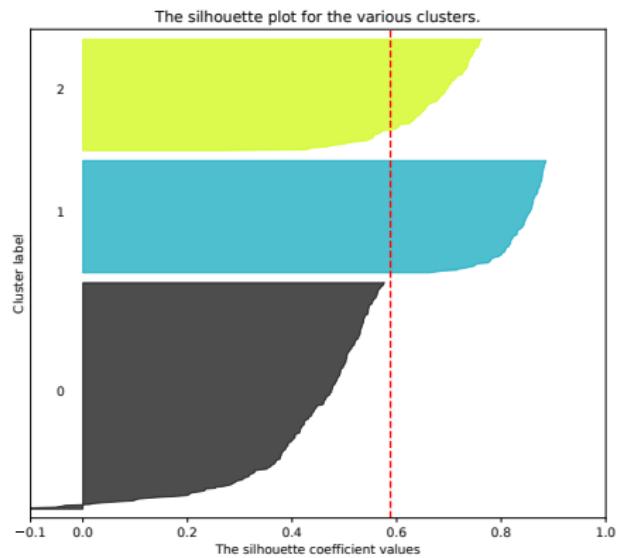
Silhouette analysis for KMeans clustering on sample data with $n_{clusters} = 2$



k-means with $k = 3$ and silhouette profiles

inspired by https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

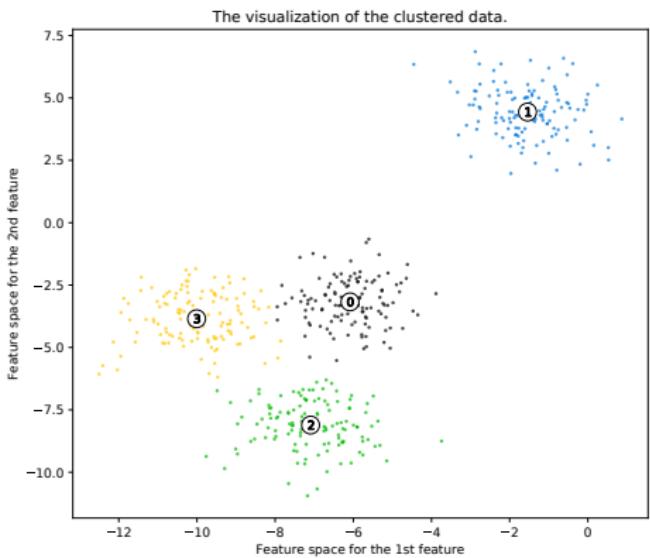
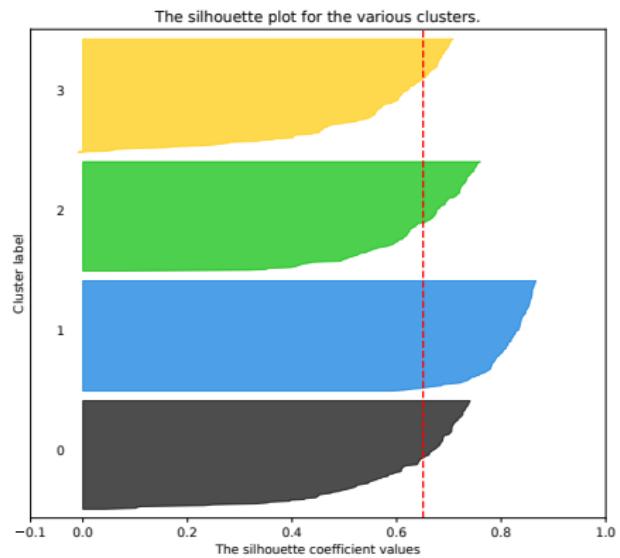
Silhouette analysis for KMeans clustering on sample data with n_clusters = 3



k-means with $k = 4$ and silhouette profiles

inspired by https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

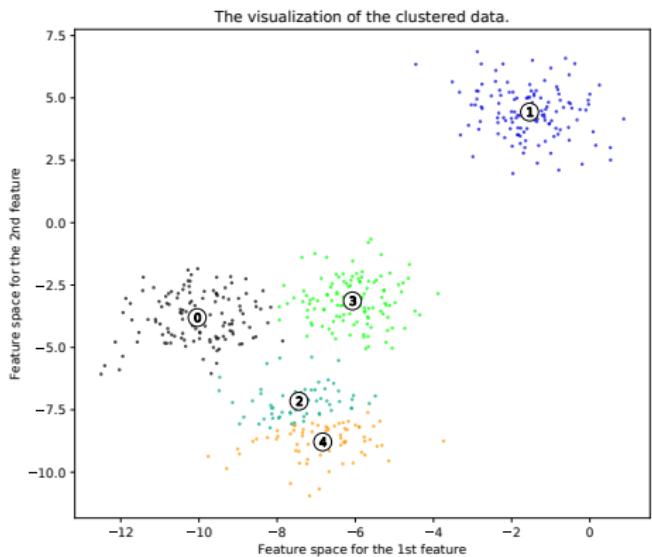
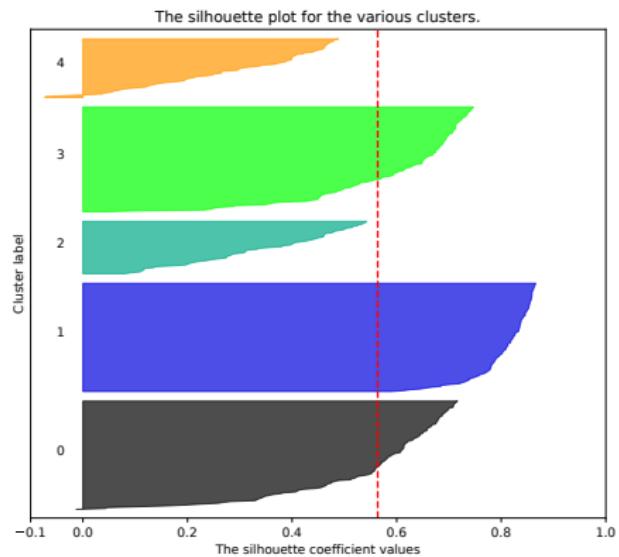
Silhouette analysis for KMeans clustering on sample data with $n_{clusters} = 4$



k-means with $k = 5$ and silhouette profiles

inspired by https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

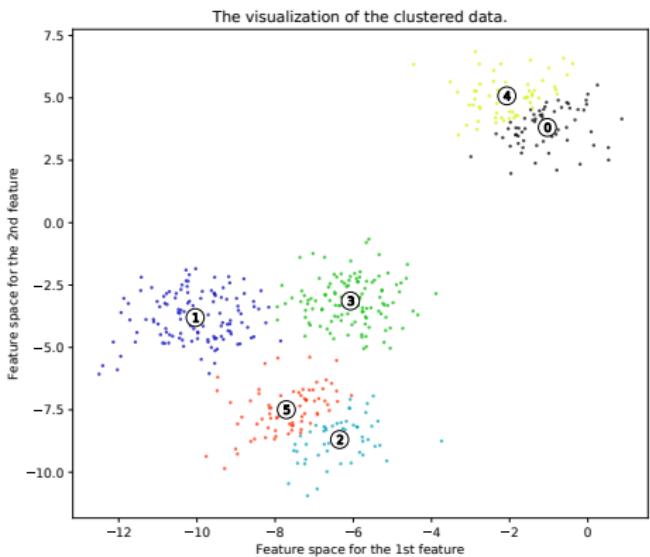
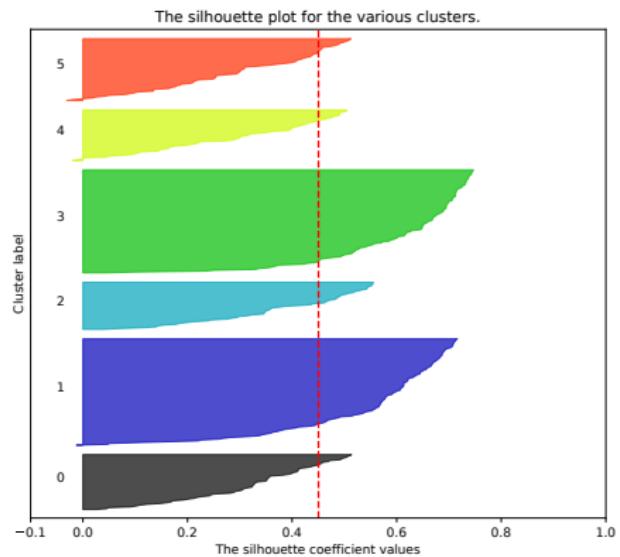
Silhouette analysis for KMeans clustering on sample data with $n_{clusters} = 5$



k-means with $k = 6$ and silhouette profiles

inspired by https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Silhouette analysis for KMeans clustering on sample data with $n_{clusters} = 6$



- For numerical data, standardisation is really important
- Many algorithms exists with different advantages and drawbacks
 - you need to find the one that fits your data
- Parameters of algorithms are also important
 - you need to tune them
- Evaluation metrics are useful but should be considered carefully
 - no better than a dedicated metric for your problem (final goal of a pipeline)