# DATA 2

## Apprentissage non supervisé, clustering, applications

Alexandre Bruckert
2023

Nantes Université
Master Cultures Numériques
Semestre 2

alexandre.bruckert@univ-nantes.fr
https://abruckert.github.io

# Supervisé ou non ?

- Apprentissage non-supervisé : on utilise des données non-labelisées (i.e. **volume d'information très important disponible**) pour apprendre quelque chose sur **la distribution** des données

- Apprentissage supervisé : on utilise des **paires données-variable d'intérêt** (i.e. **données annotées**) pour apprendre à prédire la variable d'intérêt

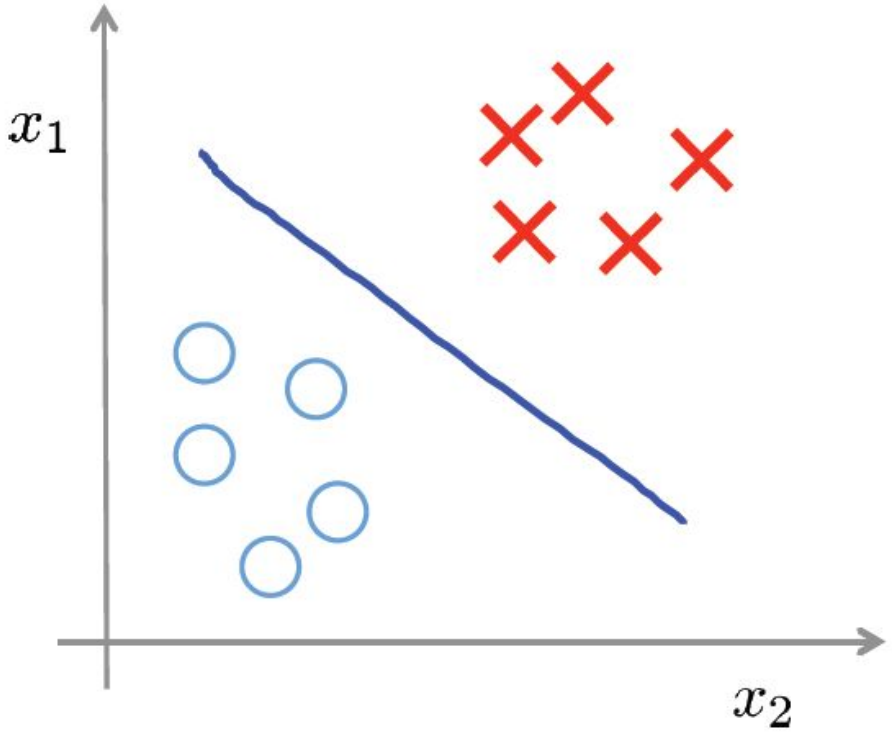
- (Apprentissage par renforcement : on ne fournit **presque pas d'information** au système, excepté une "récompense" de temps en temps)

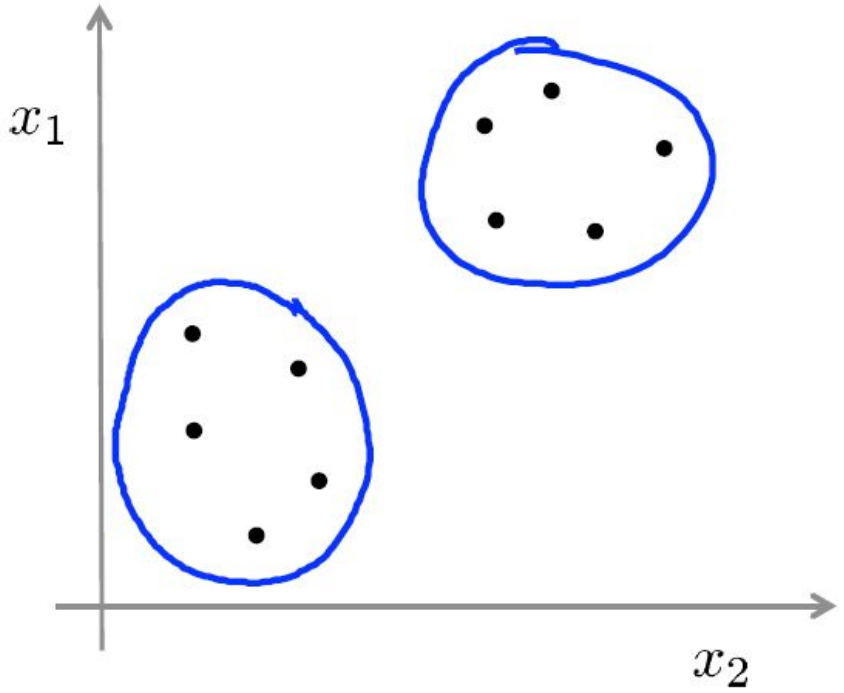
*Analogie de la "forêt noire", Yann LeCun*

# Supervisé ou non ? (version avec les maths)

- Apprentissage supervisé :
  - Une donnée disponible $\{X, Y\}$, ou $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
  - On cherche une fonction $f : X \to \hat{Y}$
  - On résout le problème $\arg\min_\Theta L(f_\Theta(X), Y)$
- Apprentissage non-supervisé:
  - Une donnée disponible $\{X\}$, ou $\{x_1, \ldots, x_n\}$
  - On cherche une fonction $f : X \to \hat{Y}$
  - On utilise l'hypothèse $x_a \sim x_b \implies y_a \sim y_b$ et $x_a \nsim x_b \implies y_a \nsim y_b$

# Supervisé ou non ? (version avec les dessins)




*Figures by Andrew Ng*

# Avantages, inconvénients...

- Apprentissage supervisé :
  - Des données de qualité, avec beaucoup d'information !
  - Une évaluation (relativement) simple, en comparant prédiction et vérité terrain
  - Il faut collecter des labels (tâche d'annotation), ce qui peut être coûteux
  - La qualité des labels est primordiale
- Apprentissage non-supervisé:
  - Des données en quantité, mais avec relativement peu d'information explicite
  - Il faut définir la similarité entre deux échantillons
  - Les modèles obtenus sont (souvent) plus facilement généralisables et robustes

# Clustering

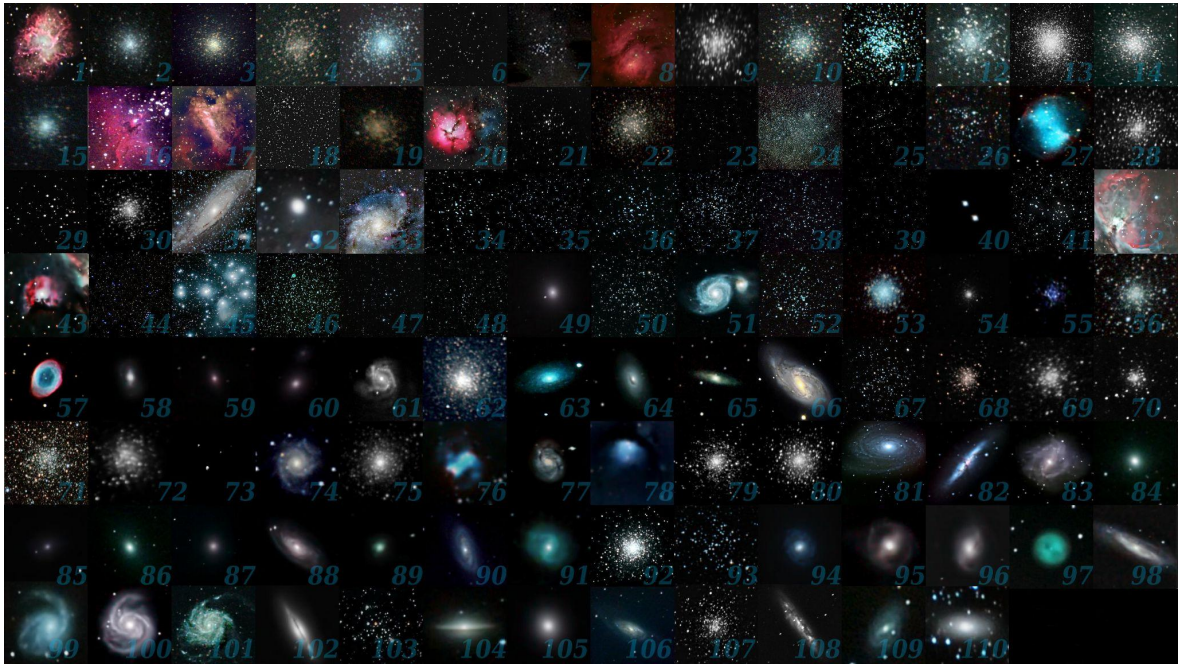
- On a un **ensemble de points** (dans $\mathbb{R}^d$) et une **distance** (ou a minima, une notion de distance) entre ces points.

- On cherche à **grouper ces points** au sein de **clusters**, tels que
  - Les points proches (au sens de la distance définie) sont dans le même cluster
  - Les points éloignés sont dans des clusters différents

- Problèmes :
  - Comment définir la distance ?
  - Comment définir le nombre de clusters ?
  - Comment classer les outliers ?
  - Comment traiter les données de très haute dimension ?
  - …

**Le clustering est un problème difficile** !

# Exemples d’application : astrophysique

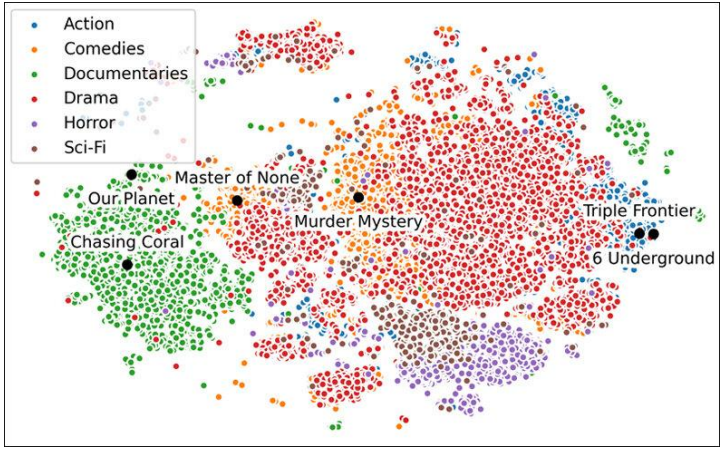


*Catalogue Messier d’objets astronomiques*

- $X$ : Une image de galaxie
- On cherche à grouper des objets similaires (galaxies, nébuleuses, étoiles…)

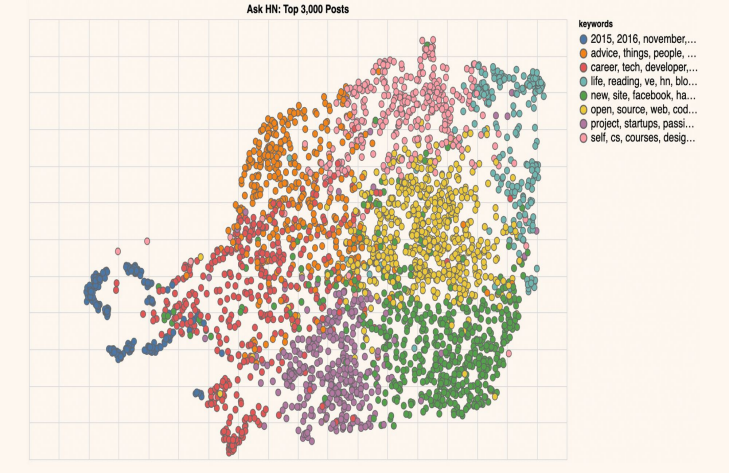
# Exemples d’application : Netflix

- On cherche à caractériser des “genres” de films / séries
- On fait l’hypothèse que des utilisateurs ont des **préférences** de genres
- On représente un film par l’**ensemble des utilisateurs qui ont regardé ce film**
- $X = (x_1, \ldots, x_n)$ où $x_i \in \{0, 1\}$ et $n$ est le nombre total d’utilisateurs
- Objectif : **regrouper les films similaires, pour pouvoir les recommander**



# Exemples d’application : Textes et documents

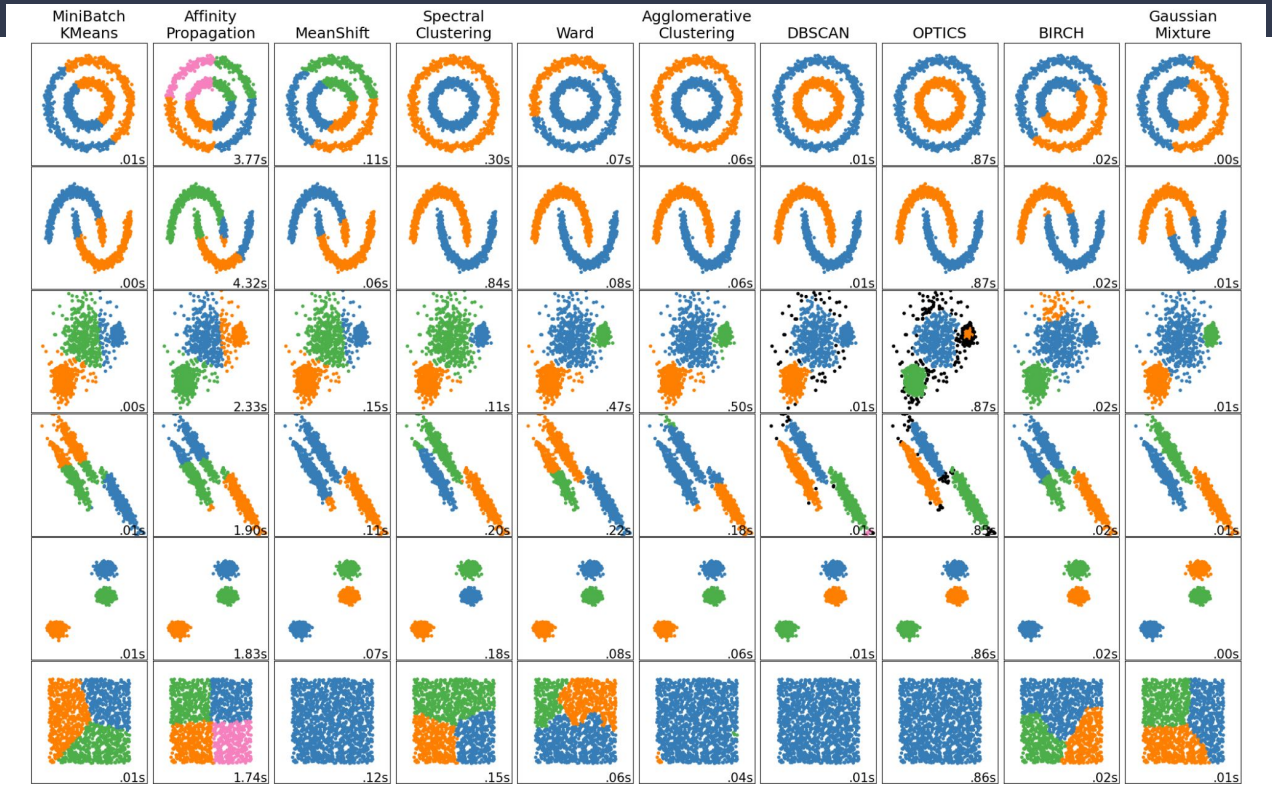
- On cherche à caractériser des “genres” de livres
- On fait l’hypothèse que des **livres similaires contiennent des mots similaires**
- On représente un livre par **l’ensemble de ses mots**
- $X = (x_1, \ldots, x_n)$ où $x_i \in \{0, 1\}$ et $n$ est la taille du dictionnaire
- Objectif : **regrouper les livres similaires, pour pouvoir les recommander**



# La malédiction de la dimension

- Clustering en 2 dimensions : ça semble simple…
- La plupart des applications prennent place dans des espèces de **très grande dimension** (texte, image, …)
- Les espaces de très grande dimension sont… **ETRANGES**
- **Quand le nombre de dimension augmente, le volume de l'espace augmente, et la distance entre les points aussi**
- **Le nombre de points nécessaires augmente très vite !**

# Clustering : différentes stratégies




*https://scikit-learn.org/stable/modules/clustering.html*

# Clustering hiérarchique

- Principe : Créer un **arbre binaire** de manière itérative
- En “coupant” l’arbre à une hauteur donnée, on obtient un nombre de clusters spécifique
- **Avantages** :
  - Peut fonctionner dans les espaces non-euclidiens
  - Pas de nombre de clusters à prédéfinir
- **Inconvénients** :
  - Beaucoup de calculs nécessaires, peu adapté aux volumes de données importants

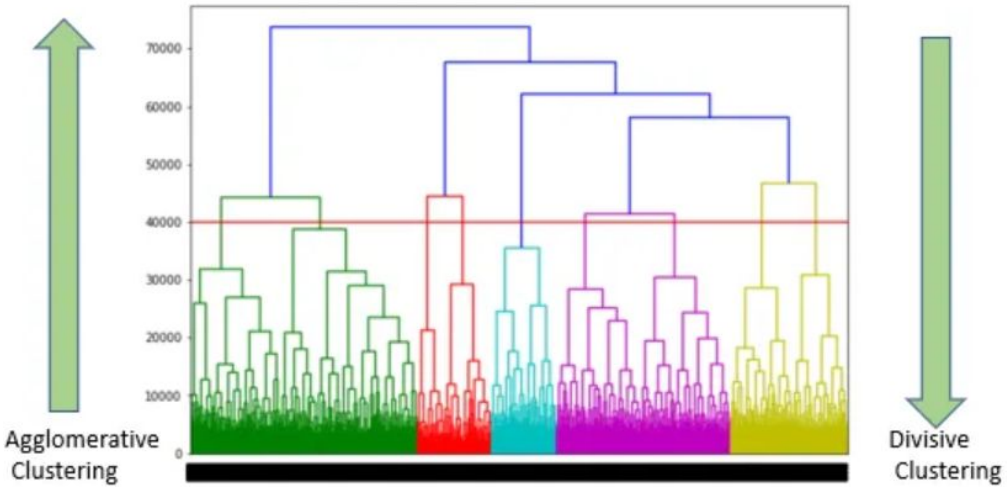



*Figure by Renu Khandelwal : https://pub.towardsai.net/everything-on-hierarchical-clustering-60bf613377a2*

# Clustering hiérarchique

- Clustering par **agglomération** :
  - Chaque point est initialement considéré comme un cluster
  - A chaque étape, on fusionne les deux clusters les plus proches (au sens de la distance considérée)
  - On continue jusqu'à obtenir un cluster unique

**Approche Bottom-Up**

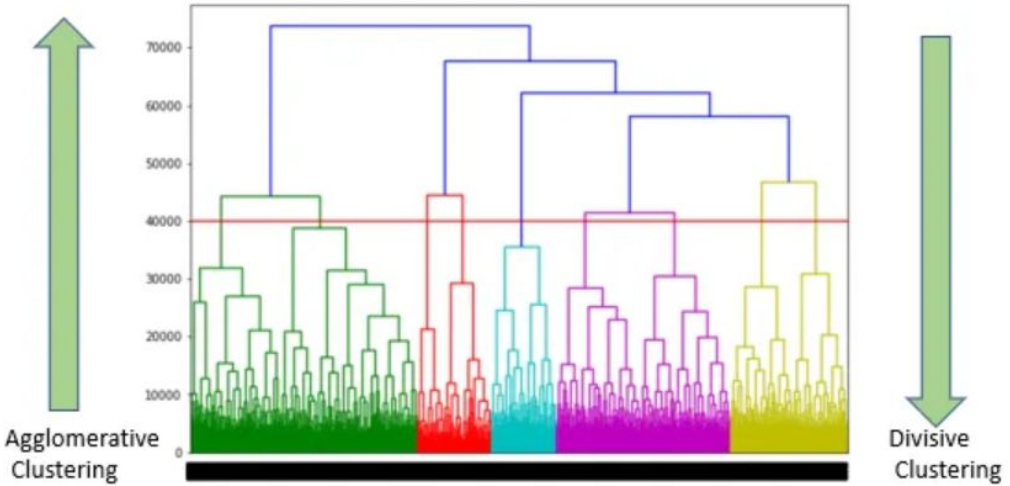



*Figure by Renu Khandelwal : https://pub.towardsai.net/everything-on-hierarchical-clustering-60bf613377a2*

# Clustering hiérarchique

- Clustering par **division** :
  - Tous les points sont dans un cluster unique
  - A chaque étape, choisit un cluster à diviser
  - On divise le cluster en deux nouveau cluster
  - On continue jusqu'à obtenir un cluster par point

**Approche Top-Down**

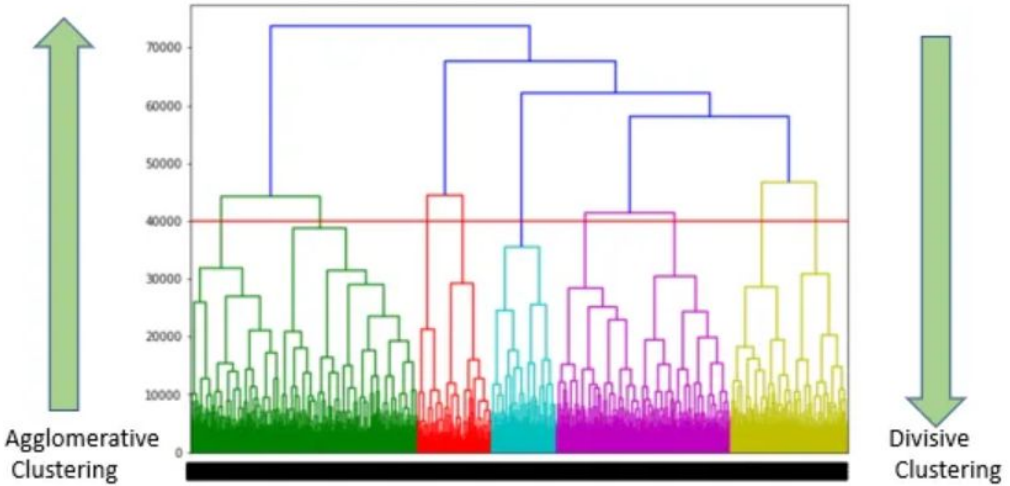



*Figure by Renu Khandelwal : https://pub.towardsai.net/everything-on-hierarchical-clustering-60bf613377a2*

# Clustering hiérarchique

- Choisir le nombre de clusters ?
  - Utiliser le dendrogramme !
  - Choisir un seuil qui “coupe” le dendrogramme de telle sorte à maximiser les distances entre les clusters
- Quels paramètres, distance, méthode choisir ?
  - Dépendant du problème et de la distribution des données

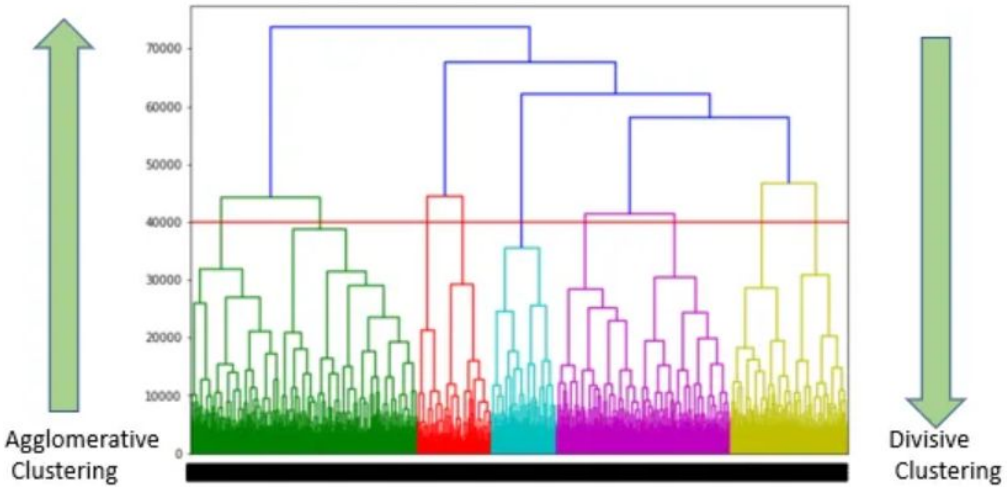
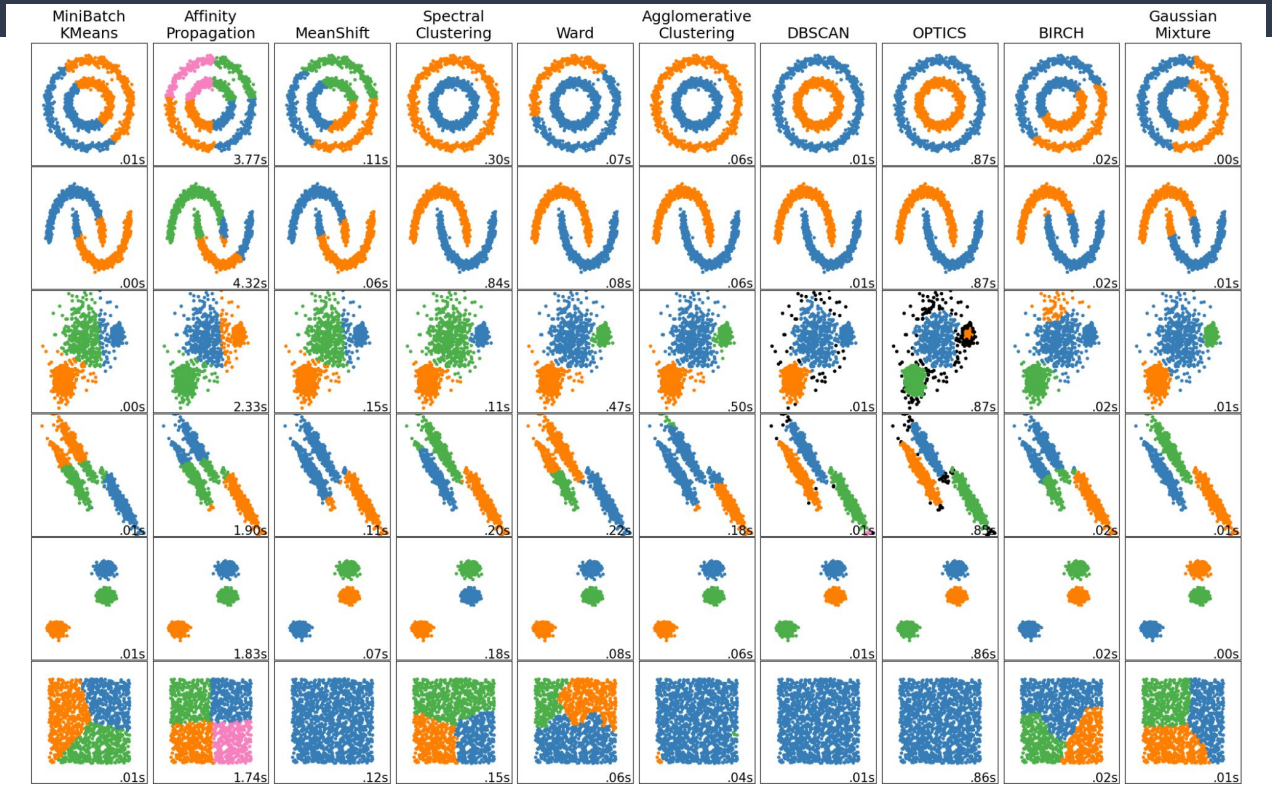



*Figure by Renu Khandelwal : https://pub.towardsai.net/everything-on-hierarchical-clustering-60bf613377a2*

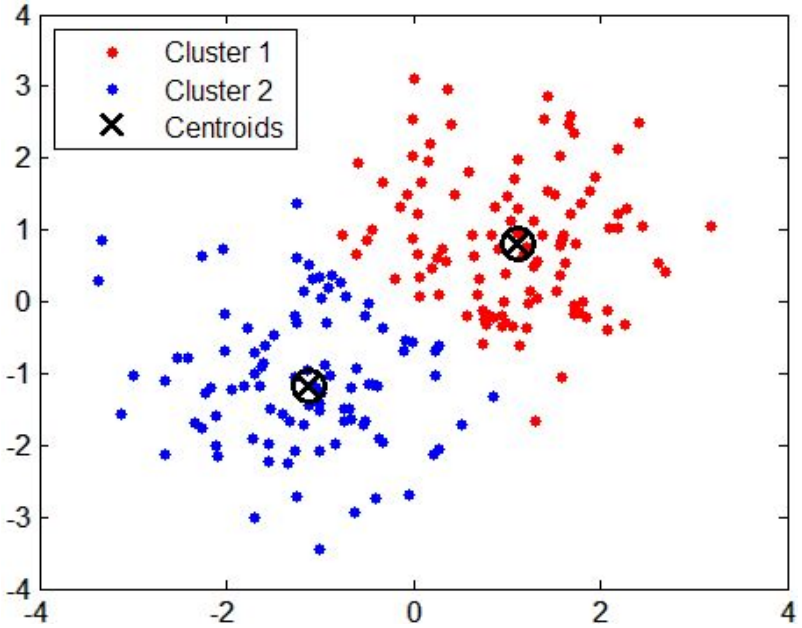
# Clustering : différentes stratégies




*https://scikit-learn.org/stable/modules/clustering.html*

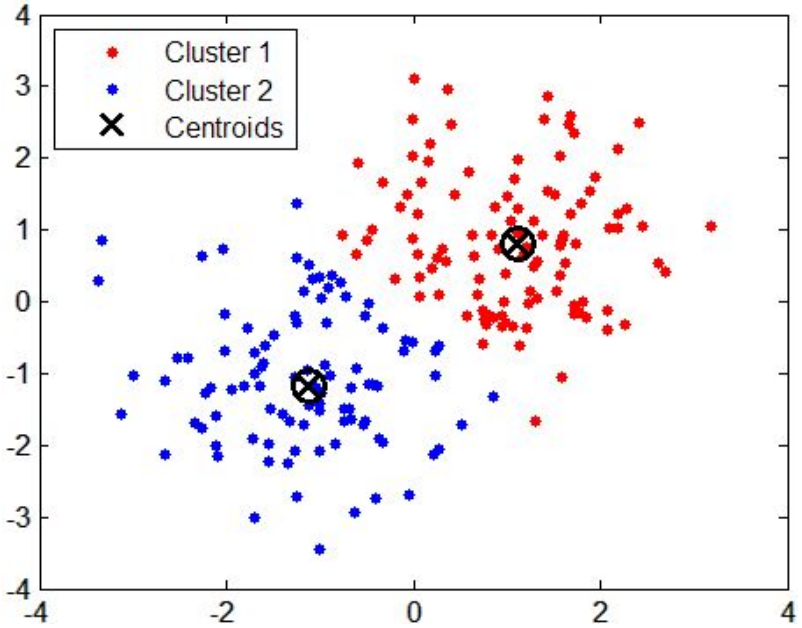
# Clustering par centroïdes

- Au lieu de définir une distance entre clusters, **on définit un point “centroïde”** qui représente le cluster
- Plutôt que de calculer toutes les distances point à point, on va calculer les **distances point-à-centroïde** !
- **Deux points proches du même centroïde sont alors considérés comme similaires**, et associés au même cluster



# Clustering par centroïdes

- **Avantages** :
  - (beaucoup) moins complexe que les algorithmes hiérarchiques
  - Très peu de paramètres à préciser
  - Facilité d'utilisation après apprentissage
- **Inconvénients** :
  - Limité aux séparations linéaires
  - Curse of dimensionality ++
  - Pas de données non-euclidiennes (sauf ajustement, modèle spécial, etc)



# Algorithme(s) k-means

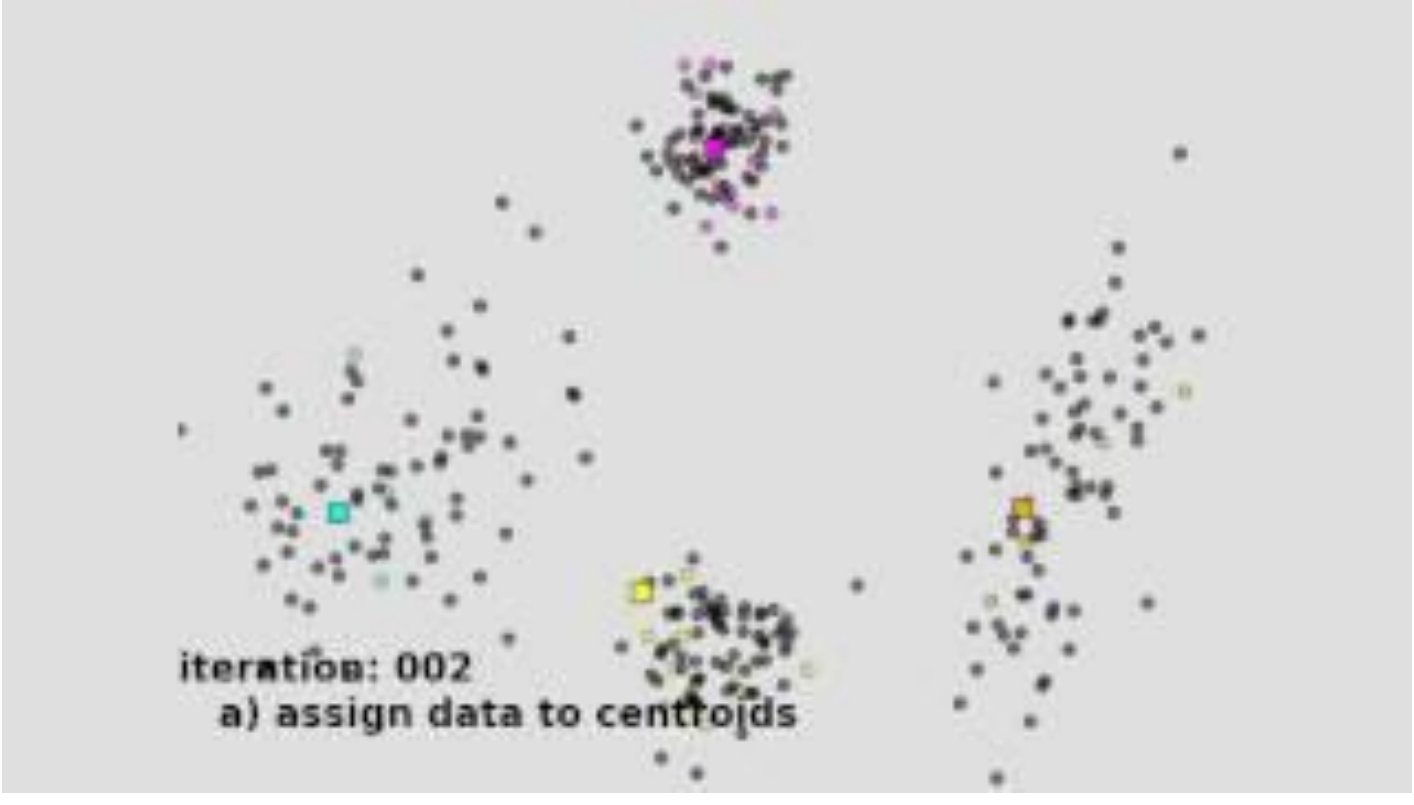
- **!!! Il s'agit d'une classe de méthodes basées centroïdes, et non d'une méthode unique !!!**
- **Problème :** Etant donné un nombre de clusters $k$, des points dans un espace euclidien muni d'une distance, on cherche à trouver des centroïdes qui vont minimiser la somme des distances quadratiques entre chaque point d'un cluster et le centroïde.
- Trouver la solution exacte à ce problème d'optimisation est **NP-difficile** !
- Solution approximative : **l'algorithme de Lloyd** $\mathcal{O}(n)$

# Algorithme(s) k-means

- On initialise en sélectionnant aléatoirement k centroïdes dans l’espace
- Jusqu’à convergence, on répète :
    - On assigne chaque point au cluster dont le centroïde est le plus proche
    - Une fois que tous les points sont assignés, on déplace les centroïdes, en prenant le barycentre des points du cluster.

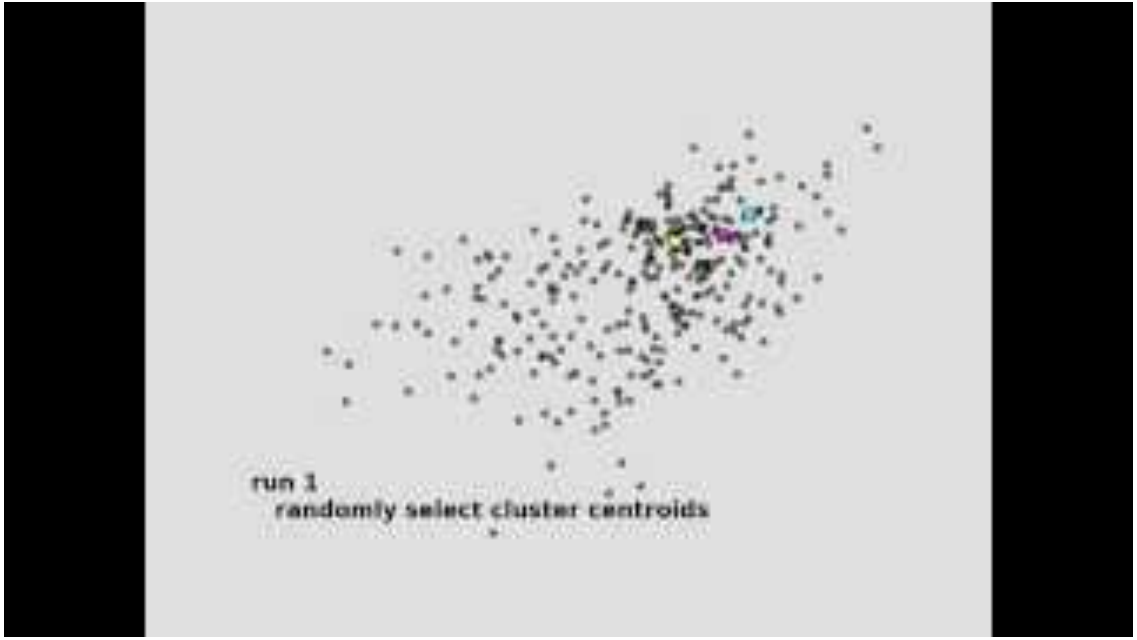
Convergence = la position des centroïdes ne bouge plus.

# Algorithme(s) k-means



# Algorithme(s) k-means

- Malheureusement, la **convergence des méthodes de k-means dépend très fortement de l'initialisation**



# Algorithme(s) k-means

- Malheureusement, la **convergence des méthodes de k-means dépend très fortement de l'initialisation**

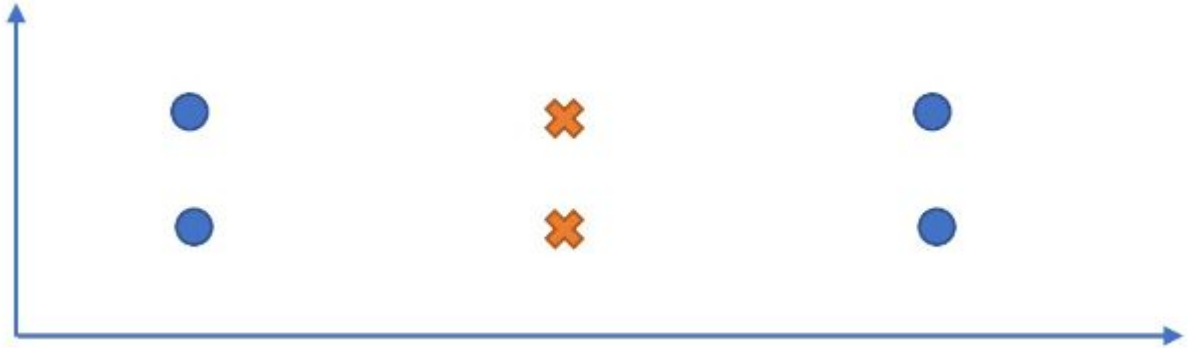
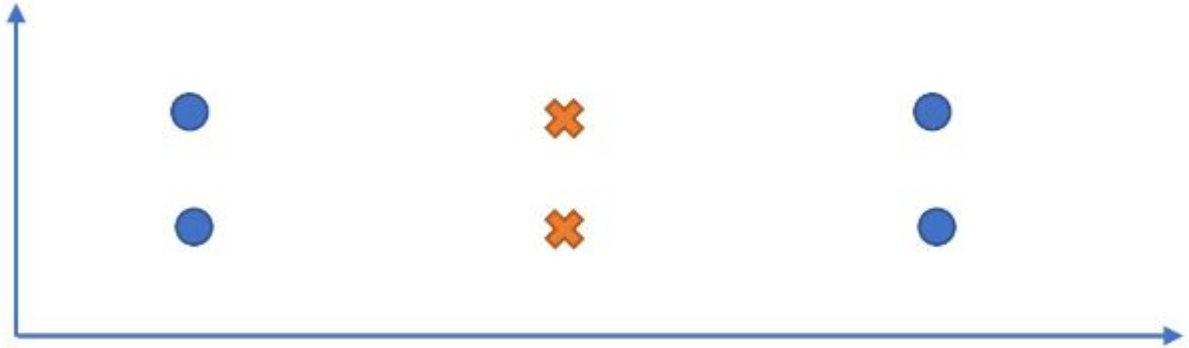


*Figure by Jure Leskovec & Mina Ghashami, Stanford CS246*

# Algorithme(s) k-means

- Malheureusement, la **convergence des méthodes de k-means dépend très fortement de l'initialisation**



- Stratégies pour régler ce problème d'initialisation :
  - Faire tourner le modèle plusieurs fois
  - k-means++

*Figure by Jure Leskovec & Mina Ghashami, Stanford CS246*

# Algorithme(s) k-means

- **k-means++** : On va venir prendre un petit **sous-ensemble de S points** dans nos données, les grouper par n'importe quelle méthode de clustering, puis **utiliser les centroïdes obtenus comme initialisation**.

- Le nombre de points **S est proportionnel au nombre de clusters k,** par un facteur qui dépend du **logarithme du nombre total de points**

- **Comment sampler les points de S :** on prend le premier point de manière aléatoire uniforme, puis pour chaque nouveau point, la probabilité d'ajouter un point donné est proportionnelle à $D(p)^2$, qui est le carré de la distance entre ce point et le point de S le plus proche.

# Algorithme(s) k-means

- **Comment choisir k ?!**

# Algorithme(s) k-means

- **Comment choisir k ?!**
- **Si le problème l'impose, on prend le nombre de clusters adapté**
  - ex : classification entre 2 classes d'images, tailles de T-shirt, …

# Algorithme(s) k-means

- **Comment choisir k ?!**
- **Si le problème l'impose, on prend le nombre de clusters adapté**
  - ex : classification entre 2 classes d'images, tailles de T-shirt, …
- **Sinon, on essaye plusieurs valeurs de k**
  - Méthode du "coude" (avec la distance moyenne aux centroïdes, par exemples)

# k-means avancés : algorithme BFR

- Comment faire si les données sont (très) volumineuses ? Problèmes de mémoire, de temps…
- **Algorithme BFR** (Bradley-Fayyad-Reina)
- **Hypothèse** : les clusters suivent une **distribution normale** (en d dimensions), dans un **espace euclidien**, centrée autour des centroïdes (i.e. les clusters sont des ellipses)
- **Idée principale** :
  - On cherche avant tout à trouver la **position des centroïdes**; l'assignement des points aux clusters se fera dans un second temps.
  - Plutôt que de considérer les points en eux-mêmes, on va alors conserver en mémoire des **statistiques liées à différents groupes de points**
  - On va venir ajouter les points petit à petit, en faisant évoluer les clusters

# k-means avancés : CURE

- Comment faire si les clusters ont une **forme arbitraire** ?
- **CURE** : Clustering Using REpresentatives
- **Hypothèses** : distance euclidienne, pas d'hypothèse sur la forme ou la distribution des clusters
- **Idée principale** : Au lieu d'utiliser des centroïdes, on va utiliser une **collection de points représentatifs du cluster**

# k-means avancés : CURE

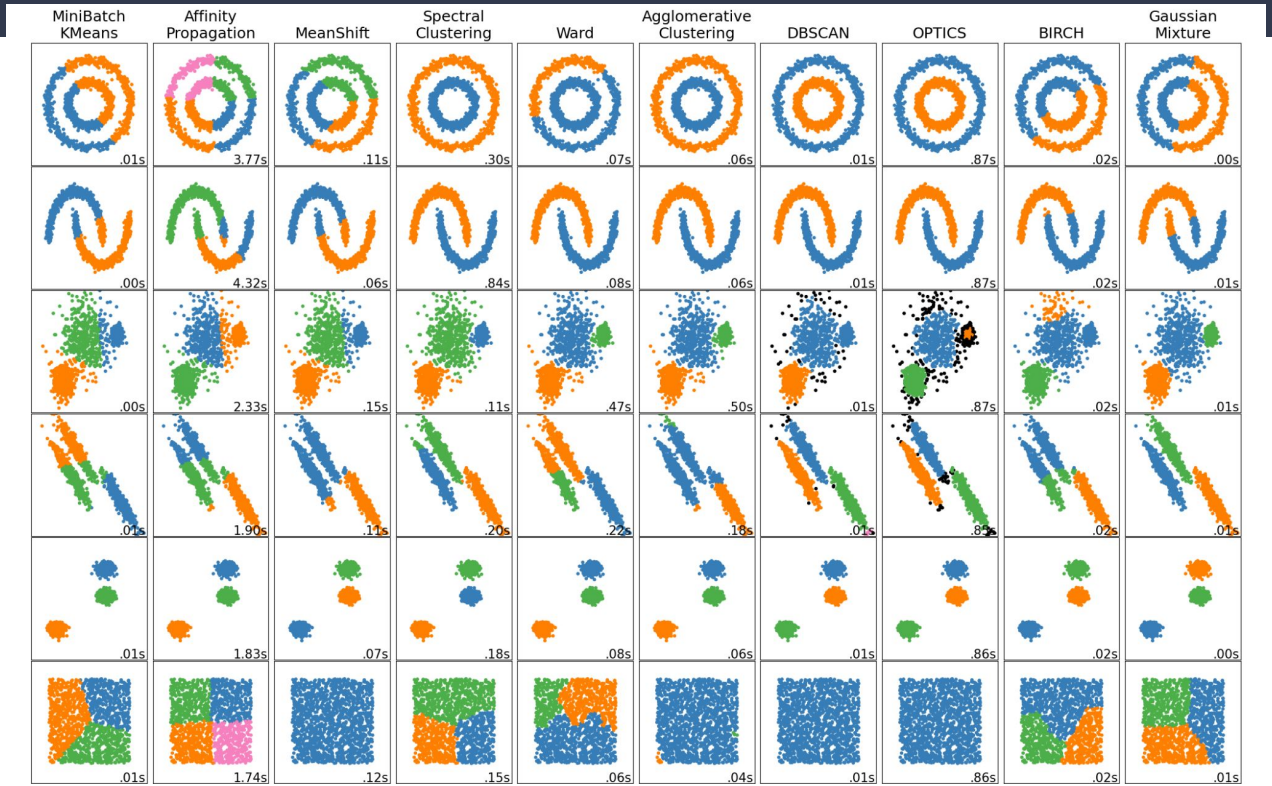
- **Passe 1:**
  - On sample un sous-ensemble aléatoire des points, et on applique un clustering hiérarchique
  - Pour chaque cluster ainsi formé, on sélectionne des points, les plus dispersés dans le cluster possible
  - On rapproche ces points du centroïde d'environ 20%
  - Ces points serviront ainsi de **représentants du cluster**.

- **Passe 2:**
  - On considère maintenant l'ensemble des points
  - Pour chaque point, on trouve le **représentant** le plus proche, et on assigne ce point au cluster associé.

# k-means avancés

- Beaucoup d’autres algorithmes:
  - Online k-means
  - Soft k-means
  - Hartigan-Wong algorithm
  - …

# Clustering : différentes stratégies




*https://scikit-learn.org/stable/modules/clustering.html*

# Bien d’autres approches

- **Modèles basés densité**
- **Modèles sur graphes**
- **Modèles basés distributions**
- **Modèles basés réseaux de neurones**
- **…**

**Pour une liste plus complète : D. Xu & Y. Tian, “A Comprehensive Survey of Clustering Algorithms”, *Annals of Data Science*, vol.2, pp 165-193 (2015)**

# Evaluer les clusters formés

Pour un modèle supervisé, on peut comparer les prédictions du modèle à une vérité terrain.
**Mais comment caractériser un “bon clustering”, sans vérité à laquelle se comparer ?**

# Evaluer les clusters formés

Pour un modèle supervisé, on peut comparer les prédictions du modèle à une vérité terrain.
**Mais comment caractériser un “bon clustering”, sans vérité à laquelle se comparer ?**

- **Pas de réponse absolue :** dépend du projet, des données, du contexte, etc
- Il existe cependant certaines métriques permettant de mesurer la “**qualité intrinsèque**” d’un clustering.
- Intuitivement :
    - Un bon clustering doit **maximiser la distance inter-cluster (séparation inter-cluster)**
    - Un bon clustering doit **minimiser la distance intra-cluster (rassemblement intra-cluster)**

# Indice de Rand

Si on a à disposition un peu de données de **vérité terrain annotées** (i.e. des paires $(x_i, y_i)$), on peut calculer l'**indice de Rand.**
L'idée est de comparer la séparation des différentes classes dans les données annotées à la séparation des clusters prédits.

$$RI(c_1(X), c_2(X)) = \frac{a+b}{\binom{n}{2}}$$

$c_1(X)$ : premier clustering des données
$c_2(X)$ : second clustering des données
$a$ : nombre de fois où une paire d'éléments appartient au même cluster dans $c_1$ et $c_2$
$b$ : nombre de fois où une paire d'éléments n'appartient pas au même cluster dans $c_1$et $c_2$

Exemple : {A, B, C, D, E}
c1 : {1, 1, 1, 2, 2}
c2 : {1, 1, 2, 2, 3}

a = 1 ; b = 5

RI = 0.6

# Indice de Rand

- **Avantages :**
  - Interprétabilité
  - Un “mauvais” clustering donne un score de 0, un clustering “parfait” donne un score de 1
  - Mesure bornée
  - Pas d’hypothèse faite sur la structure ou la forme des clusters

- **Inconvénients :**
  - Nécessite une vérité terrain (i.e. approche au moins semi-supervisée)
  - S’il n’est pas ajusté, l’indice a tendance à être souvent proche de 1, même dans des cas où les clusterings sont significativement différents.

# Coefficient de Silhouette

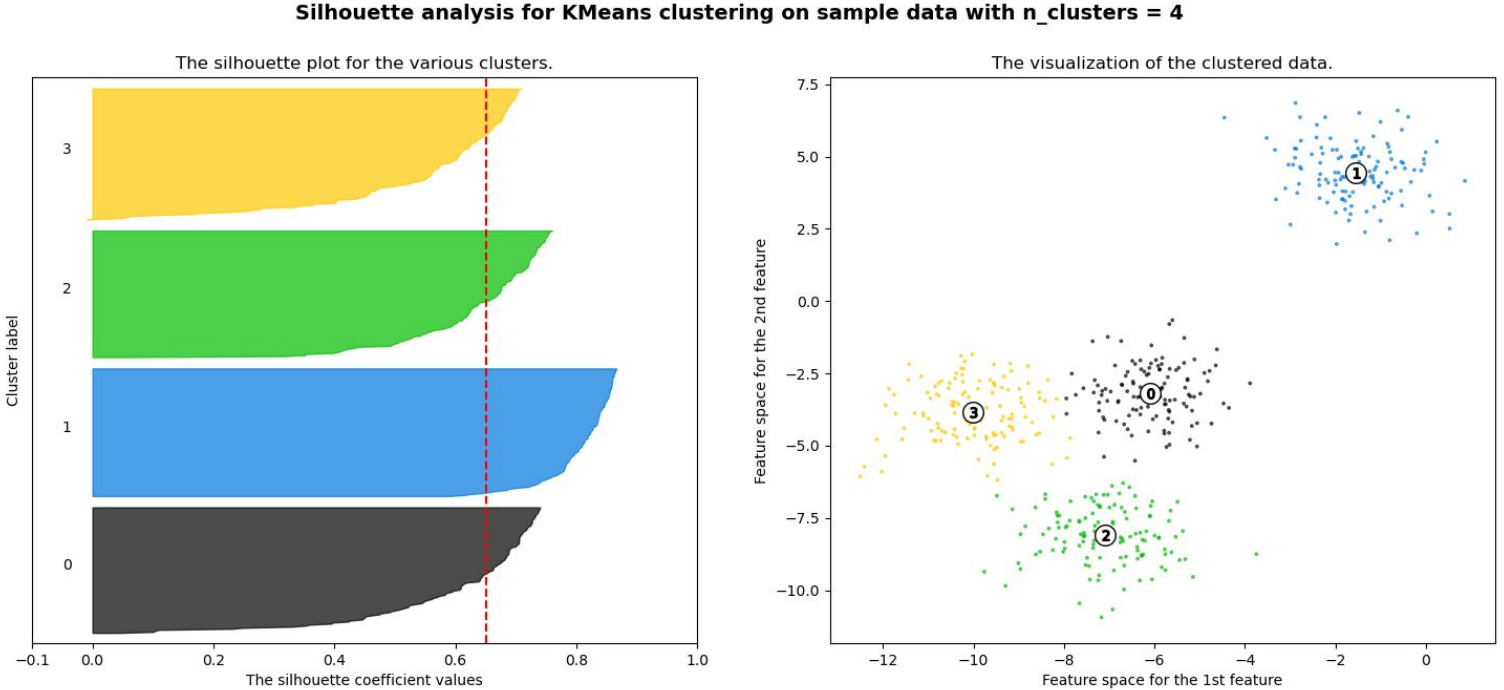
Pour chaque échantillon, on va venir calculer le score de silhouette $s$

$$s = \frac{b - a}{\max(a, b)}$$

$a$ : distance moyenne de l'échantillon aux autres points de son cluster
$b$ : distance moyenne de l'échantillon aux autres points du cluster le plus proche

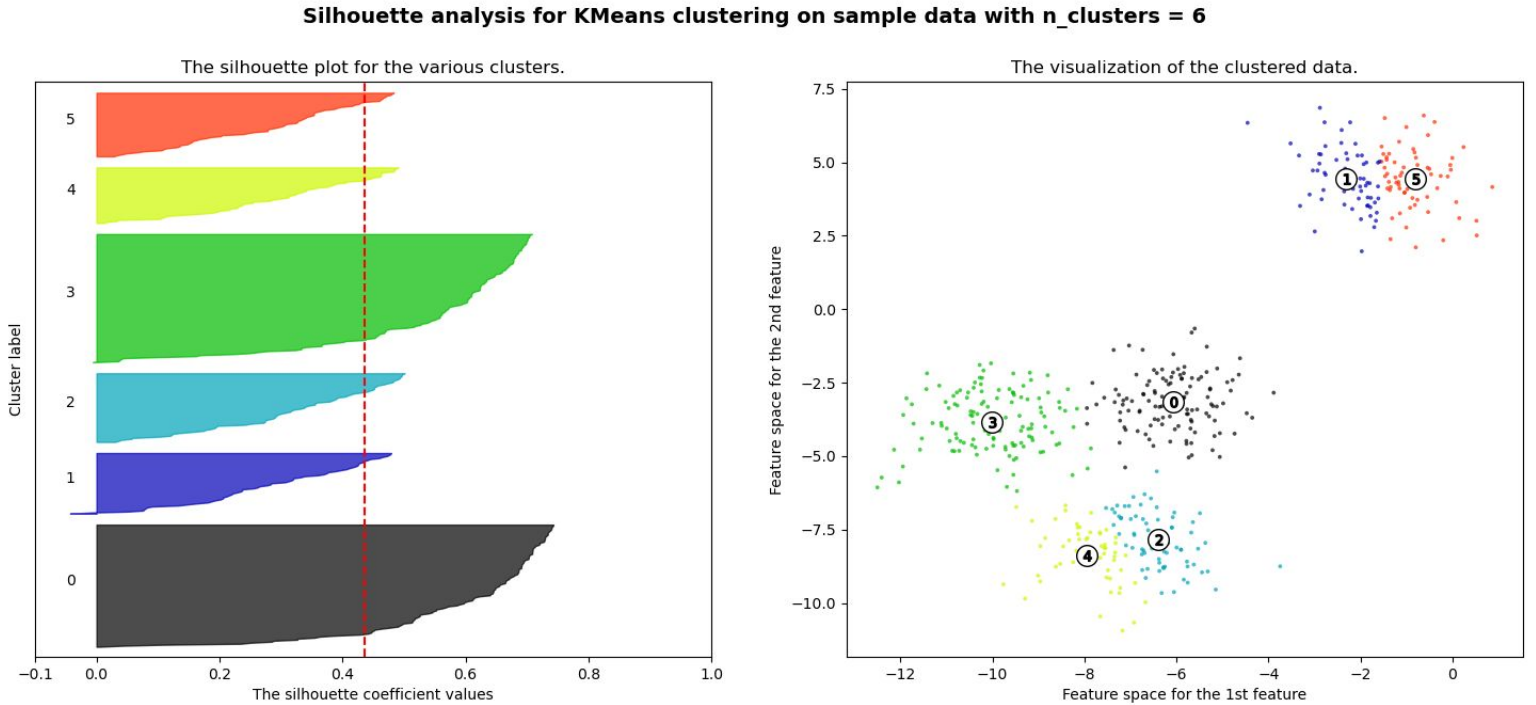
Le coefficient total d'un clustering est obtenu en moyennant l'ensemble des scores de silhouette des échantillons.

# Coefficient de Silhouette




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

# Coefficient de Silhouette




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

# Coefficient de Silhouette

- **Avantages :**
  - Mesure bornée entre -1 (clustering incorrect) et +1 (clustering très dense)
  - Coefficients autour de 0 indiquent des clusters qui se chevauchent
  - Les scores sont plus élevés quand les clusters sont bien séparés

- **Inconvénients :**
  - Le score est généralement plus élevé quand les cluster sont convexes, ce qui favorise certaines méthodes, notamment basées densité (DBScan, par exemple)

# D’autres mesures

- **Avec des classes de vérité terrain :**
  - Mesures d’information mutuelle
  - Mesures d’homogénéite, complétude, V-mesure
  - Score de Fowlkes-Mallows
  - …
- **Sans classes de vérité terrain :**
  - Index de Calinski-Harabasz
  - Index de Davies-Bouldin
  - Matrices de confusion pairwise
  - Matrices de contingence
  - **…**

**Plus d’infos : https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation**