Le clustering

K-plus proches voisins, Naïve Bayes et régression logistique sont des classifieurs faisant partie de la famille des apprentissages **supervisés** : l'appartenance des observations aux familles de sortie est connue à priori.

Le *clustering* est typiquement un cas d'apprentissage non supervisé.
Le problème de base est classique : partant d'une matrice X de données, comment extraire de façon automatique des groupes nommés *clusters* ?

On distingue deux principales approches :

Clustering hiérarchique : Le nombre k de groupes n'est pas connu à l'avance. On construit alors une suite de clusters emboités les uns dans les autres. On aboutit à une hiérarchie arborescente dans laquelle les observations sont de plus en plus ressemblantes que l'on est dans des niveaux bas.

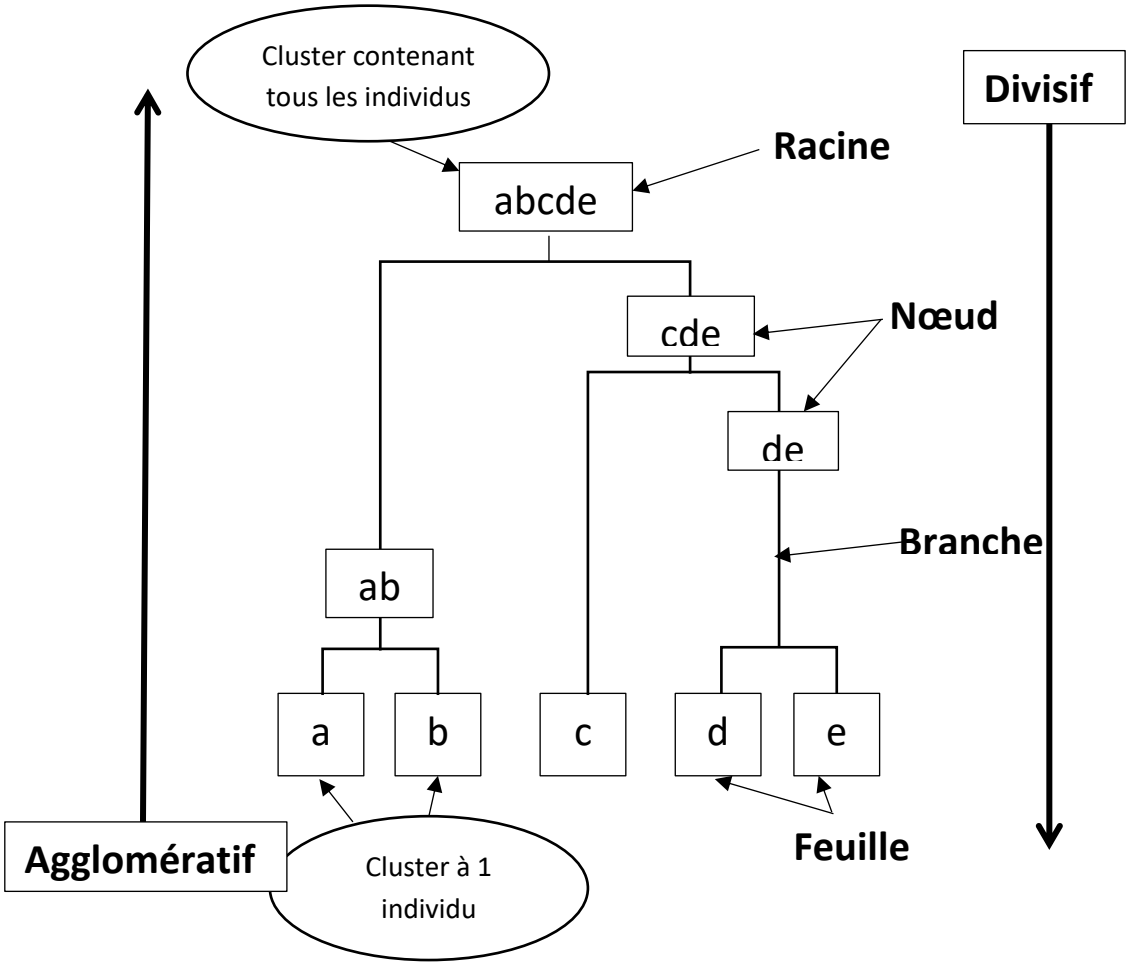
Clustering non-hiérarchique : Les individus sont répartis en k classes, pour une valeur de k fixée à l'avance. Chaque individu n'appartient qu'à un seul groupe.

Clustering hiérarchique

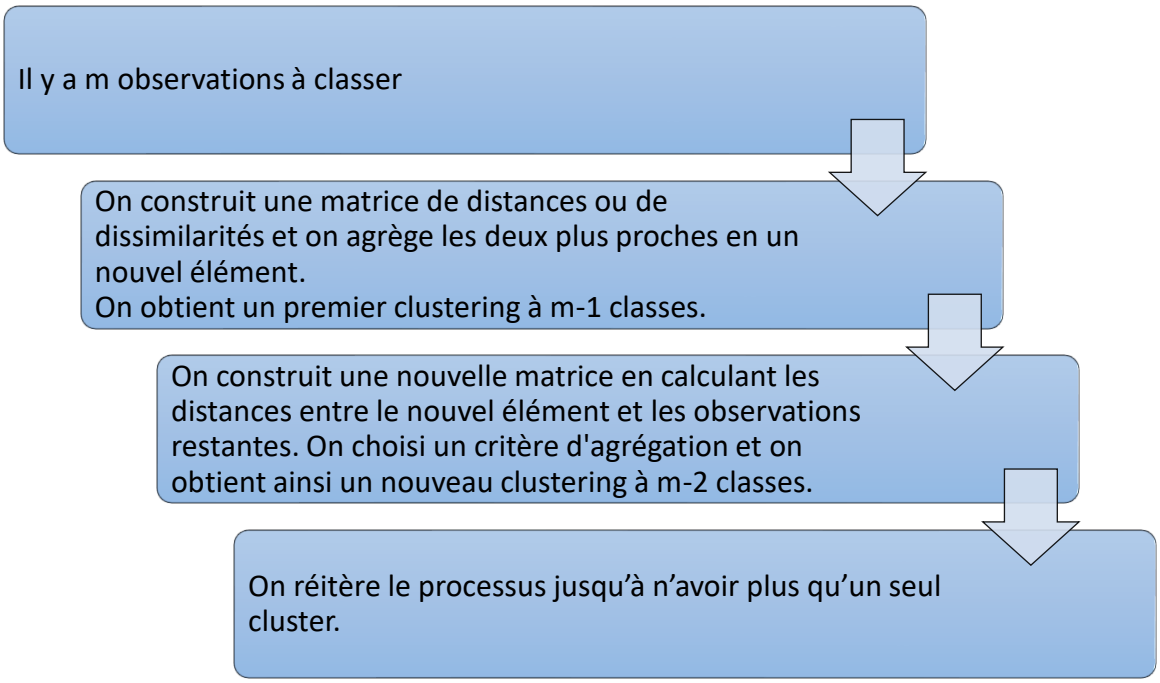
Il peut être fait de deux manières :

- Ascendant (ou agglomératif) appelé souvent AGNES (agglomerativ nesting)
- Descendant (ou divisif) appelé souvent DIANA (divisif analysis)

Dans les deux cas les algorithmes aboutissent à la construction d'un dendrogramme :



C’est la méthode agglomérative qui est la plus populaire elle peut se décrire comme suit :



Trois questions essentielles se posent :

1. Comment calcule-t-on les distances ?
2. Quel est le critère d’agrégation ?
3. Combien de classes doivent être conservées ?

Les distances :

La définition de la distance est choisie en amont de l'application de l'algorithme. Elle va définir « la ressemblance entre individus ». Elle doit donc s'adapter aux particularités des données.

Les différentes distances :

Distance euclidienne (dans l'espace $\mathbb{R}^n$)	$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$ Les données étant centrées et réduites.
Distance du χ^2	$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} \frac{1}{f_n} (f_{1i} - f_{2i})^2}$$ Très utilisée pour comparer les proportions. elle mesure l'écart entre des valeurs observées et des valeurs théoriques.
Distance de Manhattan (ou city-block)	$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} \lvert x_{1i} - x_{2i} \rvert}$$ Elle permet de minimiser l'influence des grands écarts.

Dans certaines circonstances comme les tableaux de présence/absence, la dissimilarité sera mesurée par la distance de Jaccard.

Dans ce cas x_1, x_2 sont définies par n variables binaires.
On définit alors les quantités suivantes :

- M_{11} : nombre de variables qui valent 1 cher x_1 et x_2
- M_{10} : nombre de variables qui valent 1 cher x_1 et 0 chez x_2
- M_{01} : nombre de variables qui valent 0 cher x_1 et 1 chez x_2
- M_{00} : nombre de variables qui valent 0 cher x_1 et x_2

Avec $M_{11} + M_{01} + M_{10} + M_{00} = n$

L'indice de Jaccard est alors :

$$J = \frac{M_{11}}{M_{10} + M_{01} + M_{00}} = \frac{M_{11}}{n - M_{00}}$$

La distance de Jaccard est alors :

$$d(x_1, x_2) = 1 - J$$

Le critère d'agrégation

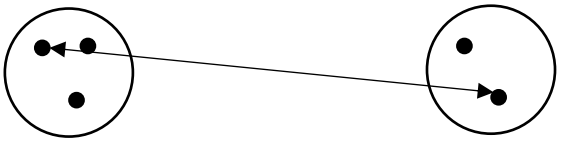
Calculer une distance entre deux individus est relativement aisé. Cependant dès la deuxième itération il faut calculer la distance entre des groupes d'individus.

Pour cela il faudra définir un critère d'agrégation.

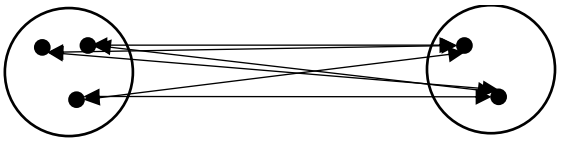
Trois critères sont utilisés quelle que soit la mesure de distance :

Single linkage :
basé sur les deux éléments les plus proches.

Complete linkage :
basé sur les deux éléments les plus éloignés.



Average linkage :
basé sur la moyenne des distances entre les éléments de chaque groupe.



Critère de Ward

Pour les distances euclidiennes, le critère le plus utilisé est le critère de Ward, qui est basé sur la minimisation de l'inertie intraclasses (écart entre chaque point et le centre de gravité de la classe).

Remarque :

Il faut retenir qu'il n'y a pas un seul clustering. Il est recommandé de tester plusieurs approches pour un même jeu de données. Selon le critère choisi on peut aboutir à des arbres ayant des formes très différentes :

- Le single linkage : produira des arbres aplatis, avec des accrochages successifs d'individus.
- Le complete linkage formera des groupes isolés et compacts.
- Le critère de Ward produira des classes d'effectifs similaires.

La troncature

C'est la partie la plus subjective et laissée à l'appréciation de l'analyste.

A quel niveau c'agrégation doit-on couper l'arbre ou comment définir le nombre de clusters le plus pertinent ?

On peut cependant respecter quelques critères :

- L'allure générale de l'arbre laisse souvent apparaître un niveau de coupe logique indiqué par des sauts importants dans les valeurs des indices de niveau.
- Eviter un trop grand nombre de clusters.
- Privilégier les clustering produisant du sens.

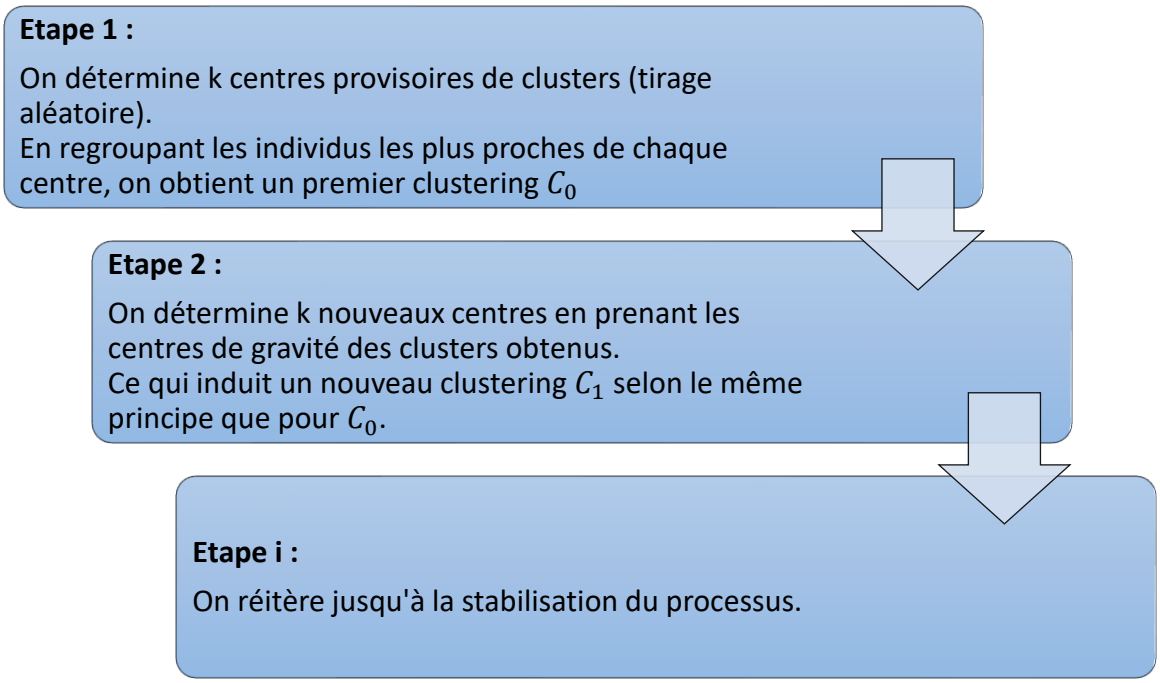
Le clustering non-hiérarchique

Cette fois ci on connait à l'avance le nombre de clusters à constituer. La solution qui consiste à énumérer toutes les possibilités de regroupement imaginables et conserver la meilleure est celle qui vient le plus naturellement à l'esprit. Elle est cependant à proscrire, le nombre de combinaisons devenant très vite énorme.

Plusieurs algorithmes permettent d'arriver à ce clustering en utilisant le principe de la méthode des centres mobiles.

Les centres mobiles

Pour k classes définies à l'avance, l'algorithme procède de façon itérative comme suit :



Le processus se stabilise généralement assez vite. L'algorithme s'arrête soit si deux itérations successives aboutissent au même clustering soit si un critère de contrôle choisi (variance intraclasse par exemple) se stabilise, soit si on atteint un nombre d'itérations fixé.

Cet algorithme se décline sur plusieurs variantes :

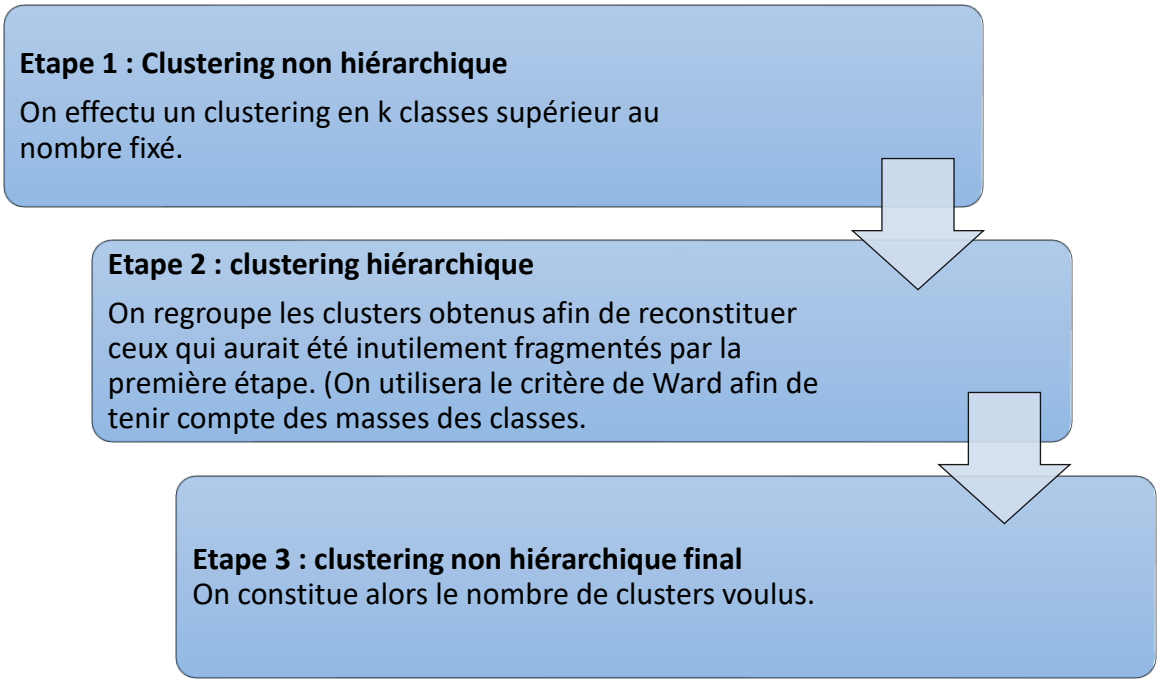
- K-means : fonctionne exactement comme les centres mobiles mais recalcule le centre de gravité à chaque nouvel individu.
- K-médoids : semblable à la précédente mais le représentant de la classe sera l'élément le plus central (le medoid)
- Les nuées dynamiques : elle remplace le centre de gravité par un groupe d'individus (les étalons) ce qui forme le « noyau »

Et d'autres ….

Clustering mixte

Dans la pratique il peut être utile de mixer les deux types de clustering pour tirer profit des avantages de chacun.

- La capacité du non hiérarchique à analyser un grand nombre d'individus.
- La possibilité avec le hiérarchique de choisir le nombre de classes optimal.



Coté Python

Bibliothèque sklearn

- preprocessing.scale : permet de centrer et réduire les données.
- cluster : permet d'invoquer la classe KMeans avec un nombre de clusters passé en paramètre(n_clusters=…)
  L'instance de KMeans utilise ensuite la méthode fit
  la méthode labels_ : renvoi le tableau des numéros de clusters pour chaque élément.
  La méthode transform : crée le tableau des distances de chaque élément aux centres des clusters
- metrics.silhouhette_score : calcule la moyenne du « Silhouette Coefficient » de tous les éléments.
  Ce coefficient représente l'écart relatif entre les distances moyennes intraclusters et extraclusters (en prenant le cluster le plus proche) pour chaque élément.
  La valeur de silhouette_score doit être la plus élevée possible.

Bibliothèque : scipy.cluster.hierarchy

- Linkage : génère la matrice des liens entre les éléments :
  Method : précise la méthode utilisée ('single', 'complete', 'average', 'weighted' ou 'ward')
  metric : type de distance : euclidean, minkowski, cityblock
- Dendrogram :
  labels : données de l'axe des x
  color_threshold permet de fixer le niveau de troncature et colore les différents clusters.
- fcluster : clustering au seuil t avec un critère de formation des clusters
  t : seuil de troncature (threshold)
  criterion : le plus souvent 'distance' c'est la hauteur des branches du dendrogramme qui est comparée au seuil. D'autres options sont possibles.

Exemple d'étude : classification des fromages.

Vous disposez d'un ensemble de fromages (29 observations) décrits par leurs propriétés nutritives (ex. protéines, lipides, etc. ; 9 variables). L'objectif est d'identifier des groupes de fromages homogènes, partageant des caractéristiques similaires.

Vous utiliserez essentiellement deux approches : la classification ascendante hiérarchique (CAH – Package SciPy) et la méthode des centres mobiles (k-Means – Package Scikit-Learn).

Le fichier « fromage.txt » provient de la page de cours de Marie Chavent de l'Université de Bordeaux. : http://www.math.u-bordeaux.fr/~machaven/teaching/

Vous y trouverez d'excellents supports et exercices corrigés qui compléteront à profit cette première prise en main de Python dans le contexte de la classification automatique.

1ère étape : Importation, statistiques descriptives et graphiques

- Importer les données dans pandas (séparateur : « \t »
- Afficher les statistiques descriptives du jeu de données.
- Utiliser la méthode « scatter_matrix » de la bibliothèque pandas.tools.plotting pour afficher les relations des variables deux à deux.

2ème étape : Classification ascendante hiérarchique

- Centrer et réduire les données.
- Générer la matrice des liens
- Afficher le dendrogramme (orienté : 'right' pour un meilleur confort de lecture)
- Fixer un seuil pour la troncature et afficher le dendrogramme avec la coloration des différents clusters
- Utiliser fcluster pour générer un tableau de « clustering »
- Ordonnez les fromages par cluster et afficher un tableau récapitulatif

3ème étape : Les centres mobiles : K-means

- Entrainer un classifieur k-means sur le jeu de données centrées et réduites pour obtenir 4 clusters.
- Afficher un tableau des fromages associés à leurs groupes
- Créer un tableau de correspondances entre les deux méthodes CAH et kmeans (en utilisant pandas.crosstab)
- Rechercher le nombre adéquat de groupes en faisant varier le nombre de clusters (de 2 à 10) et en calculant le silhouette_score. (Stocker les coefficients dans un tableau et afficher un graphique représentant la variation des scores en fonction du nombre de clusters)