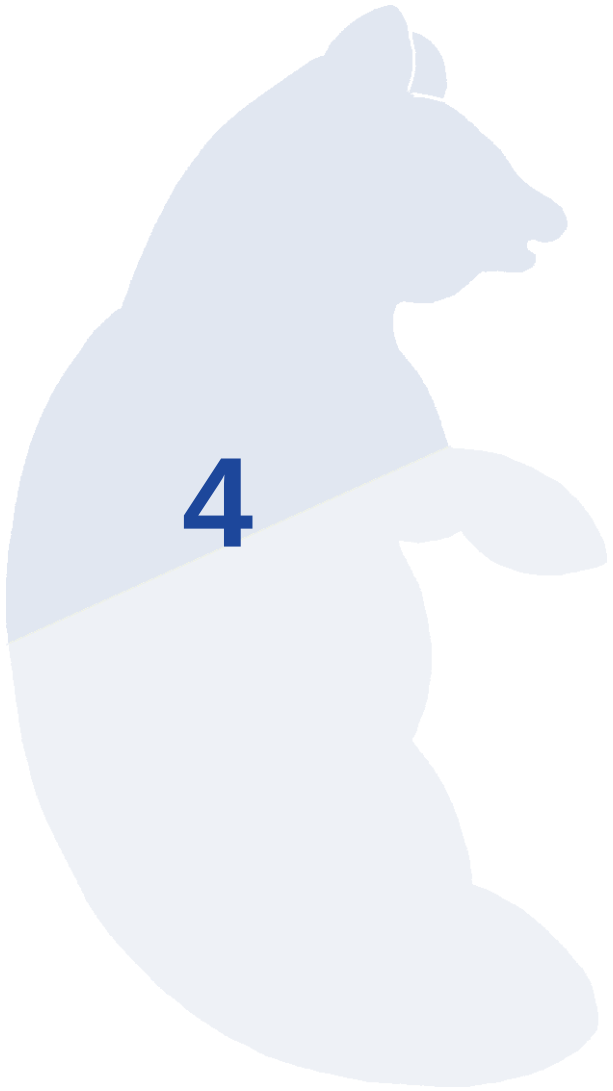


4

K-Means

4. K-Means

La méthode de clustering k-means est une technique d'apprentissage automatique non supervisée utilisée pour identifier les patterns d’individus que l’on va regrouper selon des similarités donc il est parfait pour segmenter les clients.

Cet algorithme a été conçu en 1957 au sein des Laboratoires Bell par Stuart P.Lloyd comme technique de modulation par impulsion et codage(MIC) . Il n’a été présenté au grand publique qu’en 1982. En 1965 Edward W.Forgy avait déjà publié un algorithme quasiment similaire c’est pourquoi le K-means est souvent nommé algorithme de Lloyd-Forgy.

Ces techniques nécessitent que l'utilisateur spécifie le nombre de clusters, indiqué par la variable k.

Cet algorithme est non déterministe, ce qui signifie qu'il peut produire des résultats différents à partir de deux exécutions distinctes, même si les exécutions sont basées sur la même entrée.

La qualité des affectations de clusters est déterminée en calculant la somme de l'erreur quadratique (SSE) après que les centroïdes convergent, ou correspondent à l'affectation de l'itération précédente. La SSE est définie comme la somme des carrés des distances euclidiennes de chaque point à son centroïde le plus proche. Comme il s'agit d'une mesure de l'erreur, l'objectif des k-means est d'essayer de minimiser cette valeur.

4. K-Means

L'algorithme :

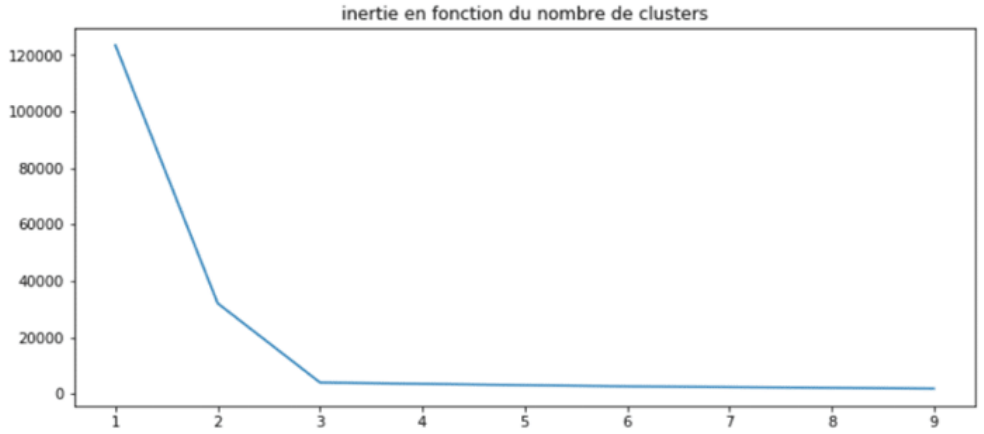
Algorithm 1 *k*-means algorithm

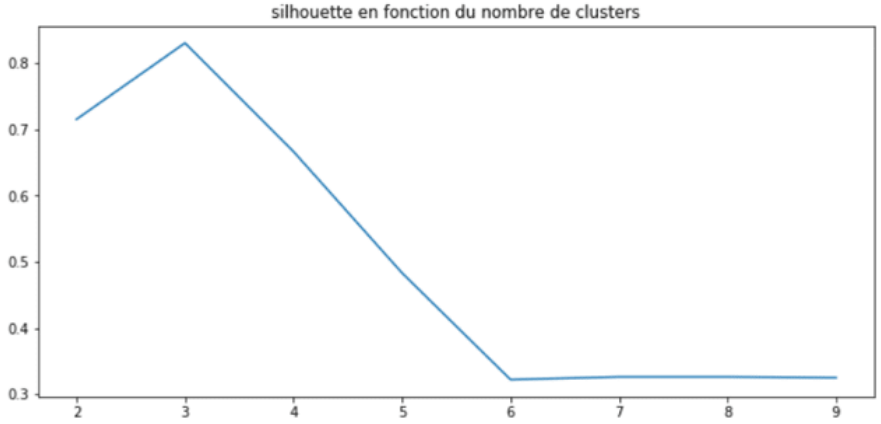
1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

La qualité des affectations de clusters est déterminée en calculant la somme de l'erreur quadratique (SSE) après que les centroïdes convergent, ou correspondent à l'affectation de l'itération précédente. La SSE est définie comme la somme des carrés des distances euclidiennes de chaque point à son centroïde le plus proche. Comme il s'agit d'une mesure de l'erreur, l'objectif des k-means est d'essayer de minimiser cette valeur.

Initialiser le paramètre k nombre de cluster :

- Méthode du coude
- ou coefficient de silhouette





4. K-Means

Hyperparamètres :

• **n_clusters** : Nous devons fournir à l'algorithme le nombre de clusters que nous souhaitons. La littérature standard suggère que nous utilisions la méthode du coude pour déterminer le nombre de clusters dont nous avons besoin et cela fonctionne bien pour les jeux de données théoriques nettoyés.
• **init** : C'est ici que vous pouvez définir les centroïdes initiaux des clusters.
• **random_state** : Il s'agit de définir une graine aléatoire. C'est utile si nous voulons reproduire des clusters exacts encore et encore.
• **n_init** : la valeur par défaut est 10. L'algorithme initialisera donc les centroïdes 10 fois et choisira la valeur la plus convergente comme meilleur ajustement. Augmentez cette valeur pour balayer tout l'espace des caractéristiques. Notez que si nous fournissons les centroïdes, l'algorithme ne s'exécutera qu'une seule fois ; en fait, il nous en avertira au moment de l'exécution. Si nous définissons les centroïdes initiaux ou si nous définissons un nombre de clusters supérieur à ce que nous attendons (dans l'intention de consolider certains clusters ultérieurement, comme indiqué ci-dessus), nous pouvons laisser cette valeur par défaut. Avec suffisamment de temps, K-means convergera toujours, mais peut-être vers un minimum local. Cela dépend fortement de l'initialisation des centroïdes. Par conséquent, le calcul est souvent effectué plusieurs fois, avec différentes initialisations des centroïdes. Une méthode permettant de résoudre ce problème est le schéma d'initialisation k-means++, qui a été implémenté dans scikit-learn (utilisez le paramètre init='k-means++'). Cela initialise les centroïdes pour qu'ils soient (généralement) distants les uns des autres, ce qui conduit à des résultats probablement meilleurs que l'initialisation aléatoire, comme indiqué dans la référence.

4. K-Means

• **tol** : Si nous fixons cette valeur à un niveau plus élevé, cela signifie que nous sommes prêts à tolérer une plus grande variation d'inertie, ou de perte, avant de déclarer la convergence (un peu comme la vitesse de convergence). Ainsi, si le changement d'inertie est inférieur à la valeur spécifiée par tol, alors l'algorithme arrêtera d'itérer et déclarera la convergence même s'il a effectué moins de max_iter rounds. Gardez cette valeur basse pour balayer tout l'espace des caractéristiques.
• **max_iter** = Il y a n_init runs en général et chaque run itère max_iter fois, c'est-à-dire que dans un run, les points seront assignés à différents clusters et la perte calculée pour max_iter fois. Si vous maintenez max_iter à une valeur plus élevée, vous avez la garantie d'avoir exploré tout l'espace des caractéristiques, mais cela se fait souvent au prix de rendements décroissants.

4. K-Means

Inconvénients :

- **Initialiser le paramètre k** (nombre de clusters).
- algorithme **non déterministe**, ce qui signifie qu'ils peuvent produire des résultats différents à partir de deux exécutions distinctes, même si les exécutions sont basées sur la même entrée.
- Le résultat dépend fortement de l'endroit où les centroïdes sont placés au départ.
- Un minima local et pas le minimum gobal peut être atteint donc pas le meilleur partitionnement.
- Plus il y a de centroïdes, plus il y aura de minima locaux.
- Sensible à la métrique de distance choisie (euclidienne, manhattan…)
- Mais lenteur quand même parce que nécessité de faire passer plusieurs fois les observations.
- La solution peut dépendre de l'ordre des individus (MacQueen) ➔ Mélanger aléatoirement les individus avant de les faire passer pour ne pas être dépendant d'une organisation non maîtrisée des observations.

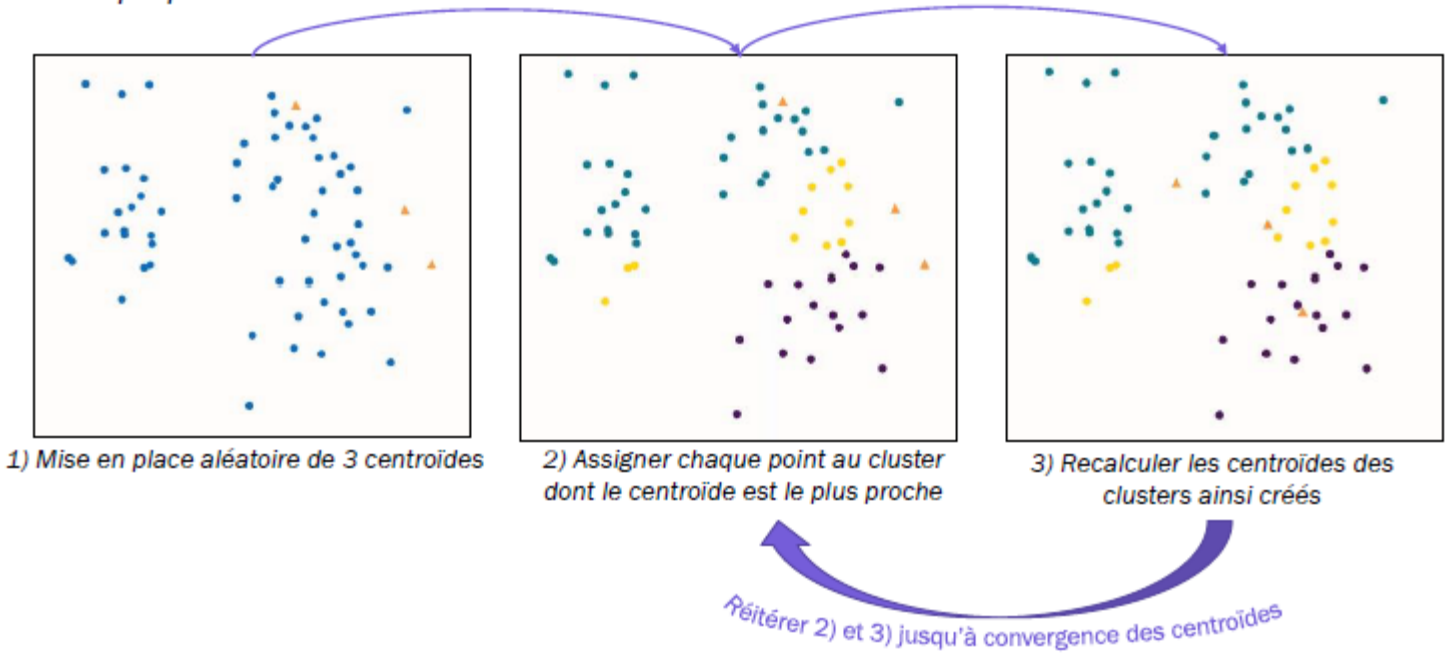
Avantages :

- Simple à mettre en œuvre.
- Entraînement rapide.
- Scalabilité : Capacité à traiter les très grandes bases. Seuls les vecteurs des moyennes sont à conserver en mémoire centrale. Complexité linéaire par rapport au nombre d'observations (pas de calcul des distances deux à deux des individus, cf. CAH).

4. K-Means

k-means – Principe :

Exemple pour 3 clusters :



4. K-Means

4. K-Means

4. K-Means