Algorithme K-means Explication

algori des kmeans

L'**algorithme K-means** est une méthode non supervisée de **clustering** (ou partitionnement de données) utilisée pour regrouper des données similaires en un nombre prédéfini de k clusters (ou groupes). Voici un aperçu détaillé de son fonctionnement :

🔑 Principe de K-means

L'objectif de l'algorithme K-means est de :

- Diviser un ensemble de points de données en **k groupes distincts**.
- Minimiser la **somme des distances quadratiques** entre les points de chaque cluster et leur centre (appelé **centroïde**).

⚙️ Étapes de l'algorithme K-means

1. **Initialisation :**
   - Choisir aléatoirement k centroïdes initiaux.
2. **Affectation des points aux clusters :**
   - Chaque point de données est affecté au centroïde le plus proche (en utilisant généralement la **distance euclidienne**).
3. **Recalcul des centroïdes :**
   - Pour chaque cluster, recalculer le centroïde en prenant la **moyenne des points** qui lui sont associés.
4. **Convergence :**
   - Répéter les étapes 2 et 3 jusqu'à ce que les centroïdes ne changent plus (ou jusqu'à ce qu'un critère d'arrêt soit atteint, comme un nombre maximal d'itérations).

📊 Fonction Coût (Inertie intra-cluster)

La fonction objectif que K-means cherche à minimiser est :

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

- C_i : l'ensemble des points du cluster i

- μ_i : le centroïde du cluster i
- $||x - \mu_i||^2$: la distance quadratique entre un point x et son centroïde

---

📌 Avantages et inconvénients

✅ Avantages :

- Simple et rapide à implémenter.
- Efficace pour des ensembles de données de grande taille.

❌ Inconvénients :

- Nécessite de définir le nombre de clusters k à l'avance.
- Sensible aux valeurs aberrantes.
- Peut converger vers un minimum local (dépend de l'initialisation des centroïdes).

🛠️ Applications courantes

- Segmentation de marché.
- Compression d'image.
- Regroupement de documents.
- Détection d'anomalies.

Si tu veux, je peux te montrer un exemple pratique avec du code Python. 😊