K-Means-Algorithmus

K-Means ist ein **iterativer Clustering-Algorithmus**, der Datenpunkte in **k Gruppen** aufteilt, basierend auf deren Ähnlichkeit.

Ablauf von K-Means

1. **Initialisierung**
    - Wähle zufällig **k Clusterzentren** (Centroids) aus den Datenpunkten.
2. **Zuweisung**
    - Weise jeden Punkt dem **nächstgelegenen Clusterzentrum** zu (nach euklidischer Distanz oder einer anderen Metrik).
3. **Update der Clusterzentren**
    - Berechne für jedes Cluster den neuen Mittelpunkt (**Mittelwert der Punkte**) und setze diesen als neues Clusterzentrum.
4. **Wiederholung**
    - Wiederhole Schritt 2 und 3, bis die **Clusterzentren stabil sind** (keine Änderungen mehr oder nur minimale Veränderungen).

Eigenschaften von K-Means

- Einfach und schnell
- Funktioniert gut, wenn Cluster **kugelförmig** und gleich groß sind
- Die Anzahl der Cluster **k muss vorgegeben** werden
- Empfindlich gegenüber **Ausreißern**
- Funktioniert schlecht bei **nicht-konvexen Clustern**

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN ist ein **dichtebasierter Clustering-Algorithmus**, der Cluster anhand der **Dichte von Datenpunkten** erkennt.

Ablauf von DBSCAN

1. **Parameter setzen**
    - `eps` : Maximaler Abstand zwischen zwei Punkten, damit sie als Nachbarn gelten.
    - `min_samples` : Mindestanzahl an Punkten in einem Gebiet, damit es als Clusterkern betrachtet wird.
2. **Punkte klassifizieren**
    - **Kernpunkt**: Hat mindestens `min_samples` Nachbarn in `eps` -Umgebung.
    - **Randpunkt**: Gehört zu einem Cluster, hat aber weniger als `min_samples` Nachbarn.
    - **Rauschen**: Gehört zu keinem Cluster.
3. **Clusterbildung**
    - Beginne mit einem **zufälligen Kernpunkt** und weise alle erreichbaren Punkte dem Cluster zu.
    - Wiederhole den Prozess für alle weiteren Kernpunkte.
    - **Rauschen bleibt ungruppiert**.

Eigenschaften von DBSCAN

- Kann Cluster mit **beliebiger Form** finden
- Erfordert keine vorherige Angabe von `k`
- **Robust gegen Ausreißer**, da sie als Rauschen erkannt werden
- Schwierigkeiten, wenn die Dichte der Cluster stark variiert
- Sensitiv auf Wahl von `eps` und `min_samples`

Vergleich K-Means vs. DBSCAN

Eigenschaft	K-Means	DBSCAN
Cluster-Form	Kugelförmig	Beliebige Formen
Anzahl der Cluster	Muss vorgegeben werden	Automatisch erkannt
Empfindlich auf Ausreißer	Ja	Nein
Performanz	Schnell	Langsamer (für große Datenmengen)
Geeignet für große Daten	Ja	Eher Nein
Parameterwahl	`k` (Clusteranzahl)	`eps`, `min_samples`

Wann welchen Algorithmus verwenden?

- **K-Means**: Wenn du **schnell** Cluster mit **kugelförmiger Verteilung** brauchst.
- **DBSCAN**: Wenn du **unregelmäßige Clusterformen** hast oder **keine feste Clusteranzahl** vorgeben kannst.