Algoritma K-Means

K-Means adalah algoritma klasterisasi berbasis partisi yang bertujuan meminimalkan variansi intra-klaster (disebut juga *inertia*) dengan cara mengelompokkan data ke dalam K klaster.

1. Formulasi Masalah

Diberikan himpunan data $X = \{x_1, x_2, \dots, x_n\} \subset \mathrm{R}^d$, K-Means mencari pusat klaster $\{\mu_1, \mu_2, \dots, \mu_K\}$ sedemikian sehingga fungsi objektif berikut diminimalkan:

$$\mathrm{L} = \sum_{k=1}^{K} \sum_{x_i \in \mathrm{C}_k} \|x_i - \mu_k\|^2$$

dengan C_k adalah himpunan anggota dari klaster ke-k dan μ_k adalah centroid dari klaster tersebut.

2. Inisialisasi Pusat Klaster

Terdapat dua pendekatan inisialisasi:

- **Random**: centroid dipilih secara acak dalam ruang data dengan:

$$\mu_k \sim \mathrm{U}(\min(X), \max(X)), \quad \forall k \in \{1, \dots, K\}$$

- **K-Means++**: centroid pertama dipilih secara acak, selanjutnya titik dipilih dengan probabilitas proporsional terhadap kuadrat jarak ke centroid terdekat:

$$P(x_i) = \frac{D(x_i)^2}{\sum_j D(x_j)^2}, \quad D(x_i) = \min_k \|x_i - \mu_k\|$$

3. Assignment Step

Setiap titik data x_i ditugaskan ke klaster dengan centroid terdekat:

$$\mathrm{label}(x_i) = \arg\min_{k \in \{1,\dots,K\}} \|x_i - \mu_k\|^2$$

Jarak dihitung dengan norma Euklidean:

$$\|x_i - \mu_k\| = \left(\sum_{j=1}^{d} (x_i^{(j)} - \mu_k^{(j)})^2\right)^{1/2}$$

4. Update Step

Setelah assignment, setiap centroid diperbarui sebagai rata-rata dari seluruh anggota klaster:

$$\mu_k = \frac{1}{|\mathrm{C}_k|} \sum_{x_i \in \mathrm{C}_k} x_i$$

5. Konvergensi

Algoritma diulang hingga konvergen, yaitu ketika centroid tidak berubah secara signifikan:

$$\|\mu_k^{(t)} - \mu_k^{(t-1)}\| < \epsilon, \quad \forall k$$

Atau hingga iterasi maksimum tercapai.

6. Inertia

Inertia didefinisikan sebagai jumlah total kuadrat jarak dari setiap titik ke centroid klasternya:

$$\text{Inertia} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

Nilai inertia digunakan sebagai metrik seberapa baik klasterisasi dilakukan (semakin kecil, semakin baik).