K-Means: Segmentación de Poblaciones (Clustering No Jerárquico)

¿Qué es K-means?

K-means es un **algoritmo no supervisado** usado para agrupar objetos o individuos en grupos (“*clusters*”) según la **similitud de sus características**. Es ampliamente usado en muchas áreas como marketing, recomendación de contenidos y análisis de datos.

Características Principales

- Es un método **no supervisado**: no usa etiquetas ni categorías predefinidas, solo usa las variables que describen a los individuos.
- Es **no jerárquico**: a diferencia del *clustering* jerárquico, necesitas decirle cuántos grupos (“k”) quieres formar.
- Solo puede trabajar con **variables numéricas**. Para variables cualitativas hay variantes, como k-modes.

Ventajas

- Fácil de implementar.
- Requiere menos cálculos que el *clustering* jerárquico.
- Puede analizar grandes volúmenes de datos.
- Suele dar buenos resultados en bases de datos reales.

Desventajas

- Debes saber cuántos grupos quieres de antemano.
- Es sensible a valores atípicos (*outliers*).
- La solución puede variar dependiendo de cómo lo inicialices.
- Puede quedarse en óptimos locales (no siempre encuentra la mejor solución global).

- Solo trabaja bien si los *clusters* tienen tamaños y densidades similares.

¿Cómo funciona K-means? (Pasos del Algoritmo)

1. **Inicialización:** Decide cuántos clusters (k) vas a buscar e inicializa los "centros" de los *clusters*. Hay varias formas de inicializar:
   - **Aleatorio:** Asignar *clusters* aleatoriamente.
   - **Centros aleatorios:** Elegir aleatoriamente k individuos como los primeros centros.
   - **K-means++:** Escoger los primeros centros lo más alejados posible entre sí.
2. **Asignación:** Cada individuo se asigna al centro más cercano (usualmente usando distancia euclídea).
3. **Re-cálculo:** Se recalculan los centros de cada *cluster* como el promedio de los individuos asignados.
4. **Iteración:** Repite los pasos 2 y 3 hasta que no cambien las asignaciones (convergencia), hasta que la mejora sea mínima o hasta un número máximo de iteraciones.

Conceptos Importantes para Evaluar el *Clustering*

- **Cohesión:** Qué tan cerca están los individuos dentro de un mismo *cluster*.
- **Separación:** Qué tan lejos están los *clusters* entre sí.
- **Inercia *intra-cluster* (W):** Distancia promedio de los individuos a su centro (mejor si es pequeña).
- **Inercia *inter-cluster* (B):** Distancia promedio entre los centros de los *clusters* (mejor si es grande).
- **Centro de gravedad:** Promedio de las variables para todos los individuos o para cada *cluster*.
- **Ley de Fisher:** Inercia total = inercia *inter-cluster* + inercia *intra-cluster*. Mejorar una ayuda a mejorar la otra.

Preprocesamiento y buenas prácticas

- **Eliminar *outliers*** antes de agrupar, pues afectan mucho el resultado.

- **Normalizar las variables**: si una variable tiene valores muy grandes dominará sobre las demás. Normaliza para poner todo en el mismo rango, por ejemplo de 0 a 1.
- **Decidir el valor de k:** Si no sabes cuántos clusters poner, usa técnicas para determinar el número óptimo.

¿Cómo decidir cuántos clusters formar? (Valor óptimo de k)

Algunas técnicas populares:

1. **Método del codo (Elbow):** Graficas la inercia intra-cluster (W) vs número de clusters. Elige el punto donde dejar de agregar clusters ya no reduce mucho la inercia ("el codo").
2. **Coeficiente de silueta:** Mide qué tan similar es un individuo a su cluster y qué tan diferente al cluster más cercano. El mejor valor es cercano a 1.
3. **Índice de Dunn:** Busca clusters compactos y bien separados. Cuanto mayor, mejor.
4. **Índice de Davies-Bouldin:** Busca clusters densos y bien separados. Cuanto menor, mejor.

> Tip: Puedes visualizar tus datos con reducción de dimensionalidad (por ejemplo, usando Análisis de Componentes Principales - PCA) para hacerte una idea de cuántos grupos podrías tener.

Ejemplo Numérico (Pasos con Inicialización)

- Escoges 3 individuos como centros iniciales.
- Calculas la distancia de cada individuo a cada centro (por ejemplo, usando la **distancia euclídea**).
- Asignas cada individuo al centro más cercano.
- Calculas el nuevo centro (promedio de sus variables) para cada grupo.
- Repites: recalculas distancias y reasignas, hasta que ya no haya cambios.

¿Qué puede salir mal?

- Si inicializas los centros muy juntos, puedes terminar con clusters malos (soluciones subóptimas).
- Por eso, **k-means++** (centrales alejados) suele dar mejores resultados y evitar óptimos locales.
- Puedes correr varias veces el algoritmo con diferentes inicializaciones y quedarte con la mejor partición.

Criterios para terminar el algoritmo

- No hay cambios en las asignaciones.
- La inercia intra-cluster cae por debajo de un umbral.
- Se alcanza un número máximo de iteraciones.

Resumen gráfico

1. Decide cuántos clusters quieres (o encuentra el número óptimo).
2. Inicializa los centros.
3. Asigna cada dato al centro más cercano.
4. Calcula los nuevos centros.
5. Repite hasta que se estabilicen los grupos.