K-MEANS

INTRODUCCIÓN

K-means es un algoritmo de *aprendizaje no supervisado* usado principalmente para clustering (agrupamiento): es decir, organizar datos en grupos cuando no tenemos etiquetas (clases) predefinidas. La idea central es sencilla: queremos separar nuestros datos en K grupos de forma que los puntos dentro de cada grupo sean lo más "parecidos" posible entre sí, y lo más "diferentes" posible de los otros grupos.

K-means es muy popular porque suele ser:

- Rápido y escalable
- Fácil de implementar
- Útil como primera aproximación

Ejemplos típicos: segmentación de clientes, agrupamiento de imágenes por colores dominantes, agrupar sensores por patrones de lectura, etc.

ALGORITMO

K-means intenta minimizar la suma de distancias al centro del cluster (normalmente distancia Euclidiana). En términos simples:

1. Cada cluster tiene un centroide (un "punto promedio").
2. Cada dato pertenece al cluster cuyo centroide está más cerca.
3. El algoritmo ajusta centroides para que queden en una posición que represente bien a sus puntos.

La función objetivo típica es minimizar la inercia (también llamada Within-Cluster Sum of Squares, WCSS):

$$\sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (1)$$

Donde:

- $K =$ número de clusters
- $C_i =$ conjunto de puntos del cluster i
- $\mu_i =$ centroide del cluster i
- $\|x - \mu_i\|^2 =$ distancia (al cuadrado) del punto al centroide

Las entradas del algoritmo son el conjunto de datos X y el número de clusters K; la salida es una asignación de cada punto a un cluster y los centroides. A continuación, se presentan los pasos del algoritmo

Paso 1 - Inicialización

Se eligen Kcentroides iniciales. Aquí hay dos opciones comunes:

- Aleatorio: simple, pero puede salir mal
- K-means++: elige centroides iniciales más separados para mejorar estabilidad y resultados

Paso 2 – Asignación

Para cada punto, se calcula su distancia a cada centroide y se asigna al más cercano:

$$cluster(x) = argmin_i \|x - \mu_i\| \quad (2)$$

Paso 3 – Actualización

Se recalcula cada centroide como el promedio de los puntos asignados a ese cluster:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \quad (3)$$

Paso 4 – Repetir hasta converger

Se repiten los pasos 2 y 3 hasta que ocurra alguna condición:

- Ya no cambian las asignaciones

- Los centroides se mueven muy poquito
- Se alcanza un número máximo de iteraciones

No hay una única respuesta “correcta” para elegir K, pero se usan heurísticas comunes, como por ejemplo el método del codo, con el cual se busca el punto donde aumentar K ya no reduce mucho el error.

Para entender un poco mejor, utilizaremos un ejemplo. Supongamos que tenemos el siguiente conjunto de datos:

Punto	Variable1	Variable2
A	1	1
B	1	2
C	2	1
D	8	8
E	9	8
F	8	9

Figura 1. Conjunto de datos

Si graficamos los puntos tendríamos la siguiente figura de dos dimensiones:

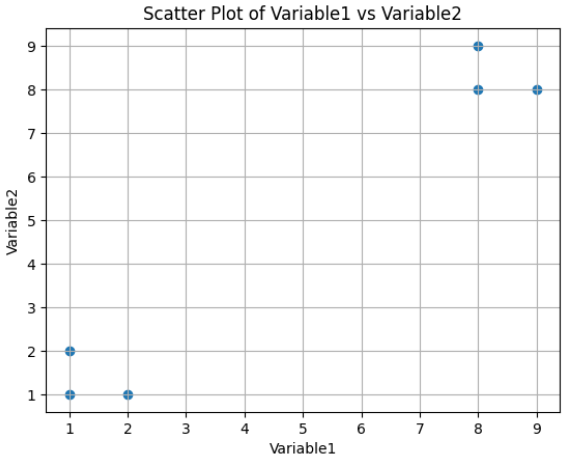

Figura 2. Gráfica del conjunto de datos

Utilizaremos dos clusters ($K = 2$) y mediremos la distancia de los puntos utilizando la distancia Euclidiana, la cual está definida como:

$$d\big((x,y),(a,b)\big) = \sqrt{(x-a)^2 + (y-b)^2} \tag{4}$$

Recordando, el primer paso es la inicialización de los centroides, para hacerlo fácil, elijamos a:

- Centroide 1: $\mu_1^{(0)} = A = (1,1)$
- Centroide 2: $\mu_2^{(0)} = B = (8,8)$

Si incluimos los centroides en la Figura 2 obtenemos lo siguiente:

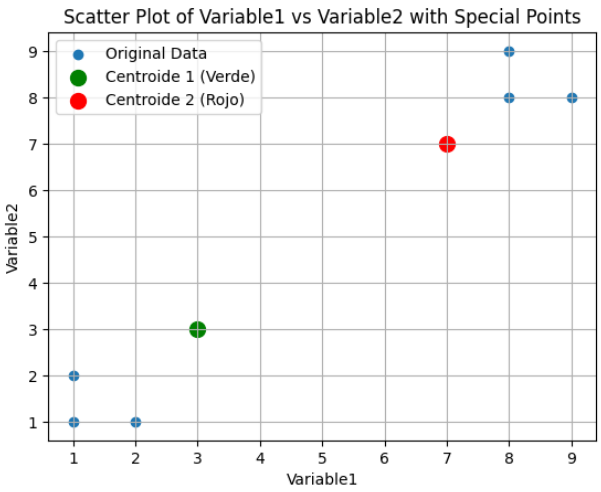

Figura 3. Gráfica del conjunto de datos y centroides

El segundo paso es la asignación de cada punto al centroide más cercano. Calculemos la distancia del punto A(1,1), a los dos centroides:

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{(1-3)^2 + (1-3)^2}$$

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{(-2)^2 + (-2)^2}$$

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{4+4}$$

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{8}$$

$$d\left(A, \mu_1^{(0)}\right) = 2.82$$

$$d\left(A, \mu_2^{(0)}\right) = \sqrt{(1-7)^2 + (1-7)^2}$$

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{(-6)^2 + (-6)^2}$$

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{36 + 36}$$

$$d\left(A, \mu_1^{(0)}\right) = \sqrt{72}$$

$$d\left(A, \mu_1^{(0)}\right) = 8.48$$

Hacemos lo mismo con los demás puntos. En la siguiente Figura se puede observar la distancia entre cada punto y los dos centroides:

Punto	Variable1	Variable2	Distancia_Centroide1	Distancia_Centroide2
A	1	1	2.828427	8.485281
B	1	2	2.236068	7.810250
C	2	1	2.236068	7.810250
D	8	8	7.071068	1.414214
E	9	8	7.810250	2.236068
F	8	9	7.810250	2.236068

Figura 4. Resultado de las distancias entre cada punto y los centroides.

Una vez que se calculan las distancias, se procede a asignar cada punto con su centroide más cercano. El conjunto de datos se vería de la siguiente manera:

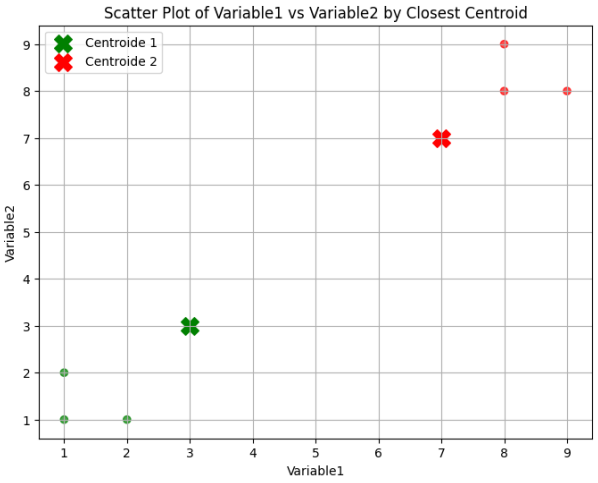

Figura 5. Puntos asignados a su centroide más cercano

El paso 3 consiste en la actualización de los centroides de tal manera que la distancia promedio entre cada uno de sus puntos asignados sea la menor. Para hacerlo, utilizaremos la ecuación 3:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

$$\mu_1 = \frac{1}{3} \sum_{x \in C_i} [(1,1,2),(1,2,1)]$$

$$\mu_1 = \frac{1}{3}(4,4)$$

$$\mu_1 = (1.33, 1.33)$$

Haciendo lo mismo para el centroide 2, tenemos que:

$$\mu_1 = \frac{1}{3} \sum_{x \in C_i} [(8,9,8),(8,8,9)]$$

$$\mu_1 = \frac{1}{3}(25,25)$$

$$\mu_1 = (8.33, 8.33)$$

Graficando los nuevos centroides, podemos observar (Figura 6) que éstos se han movido

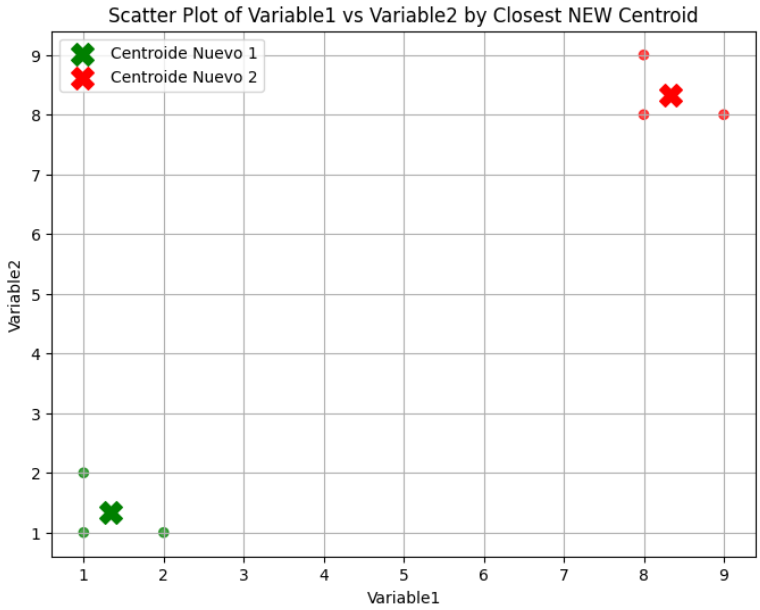

Figura 6. Agrupamiento del conjunto de datos

En este pequeño ejemplo se alcanzó la convergencia en la segunda iteración ya que el conjunto de datos fue muy sencillo.

K-means es un algoritmo ampliamente utilizado debido a su simplicidad, rapidez y facilidad de implementación, lo que lo convierte en una excelente herramienta para el análisis exploratorio de datos y para trabajar con conjuntos de datos grandes. Además, es computacionalmente eficiente, fácil de interpretar y suele ofrecer buenos resultados cuando los clusters son compactos, bien separados y de tamaño similar. Sin embargo, presenta varias limitaciones importantes: requiere definir previamente el número de clusters K, es sensible a la inicialización de los centroides, puede converger a mínimos locales, y su desempeño se ve afectado por outliers y por variables en diferentes escalas si no se normalizan adecuadamente. Asimismo, asume que los clusters tienen formas aproximadamente esféricas y utiliza típicamente la distancia euclidiana, lo que lo hace poco adecuado para datos con estructuras complejas o distribuciones no uniformes