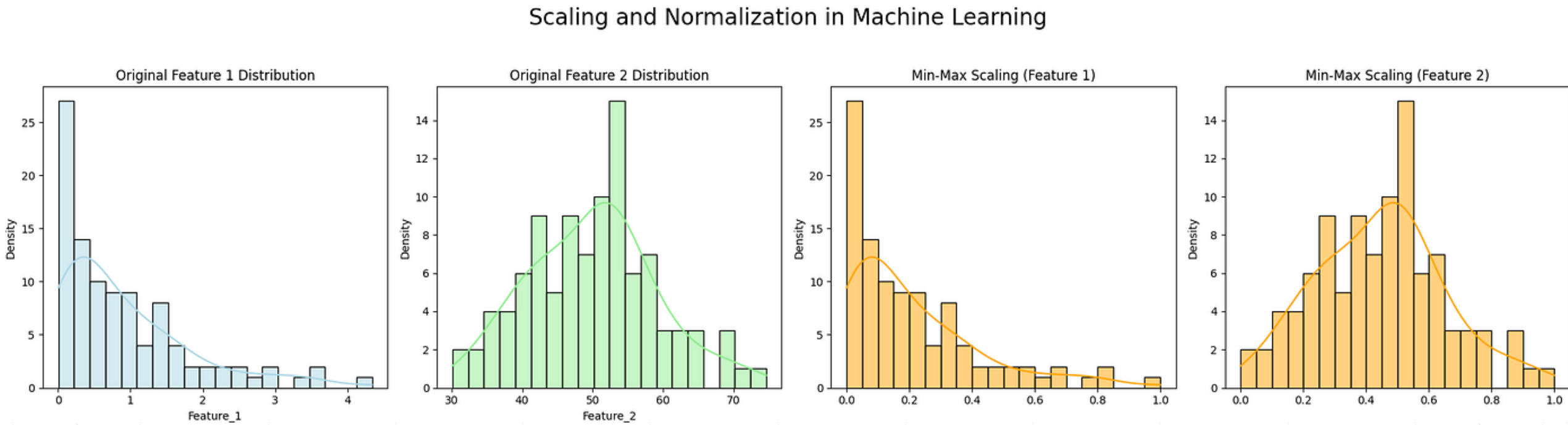
Problema de escalas

Si mezclás columnas con números gigantes (Salario) y números chicos (Edad), la columna gigante tapa a las demás. Las distancias y los modelos se quedan con “lo más fuerte”.

- Distancias y gradientes **sí** miran la escala.
- Ejemplo: 35 años vs **$80.000** → el $80.000 manda.
- Solución general: **escalar/normalizar** antes de modelar.

Problema de escalas



Normalización

Normalizar = poner todas las columnas en la misma escala para comparar manzanas con manzanas.

- Hace que las variables hablen **el mismo idioma**.
- Ayuda a los algoritmos a **aprender más parejo**.
- No crea información nueva ni arregla sesgos.

Min-Max vs StandardScaler

Dos formas rápidas de “ajustar el volumen”.

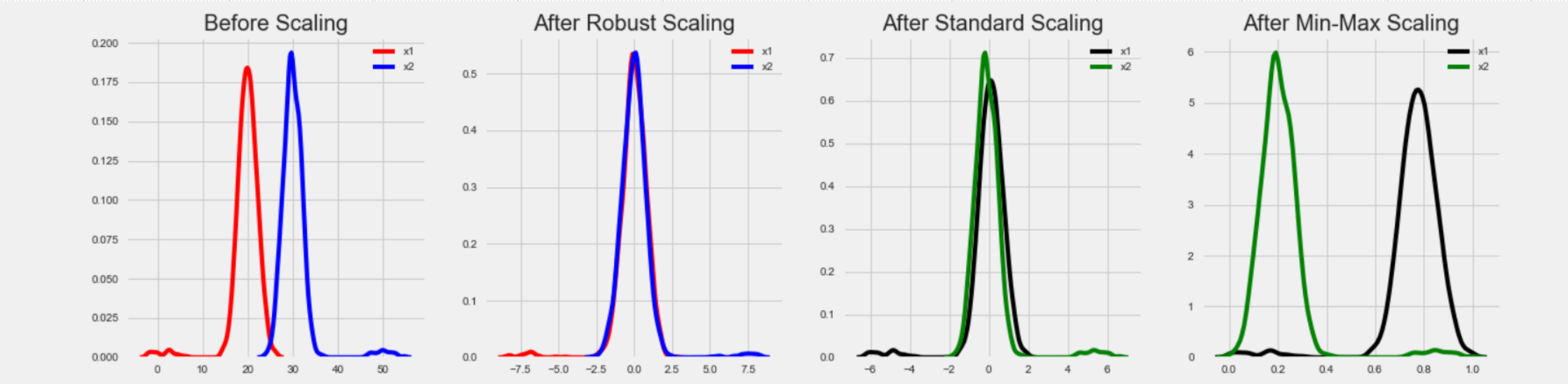
Min-Max (0–1) → *“aprieta y estira”* para que el más chico sea 0 y el más grande 1.

- Datos limpios y acotados.
- Con **outliers** se rompe (todo queda aplastado).

StandardScaler (Z-score) → *“centra en 0 y ajusta el tamaño”*.

- Mejor cuando hay valores raros.
- Suele ayudar a optimizadores basados en gradiente.

Min-Max vs StandardScaler



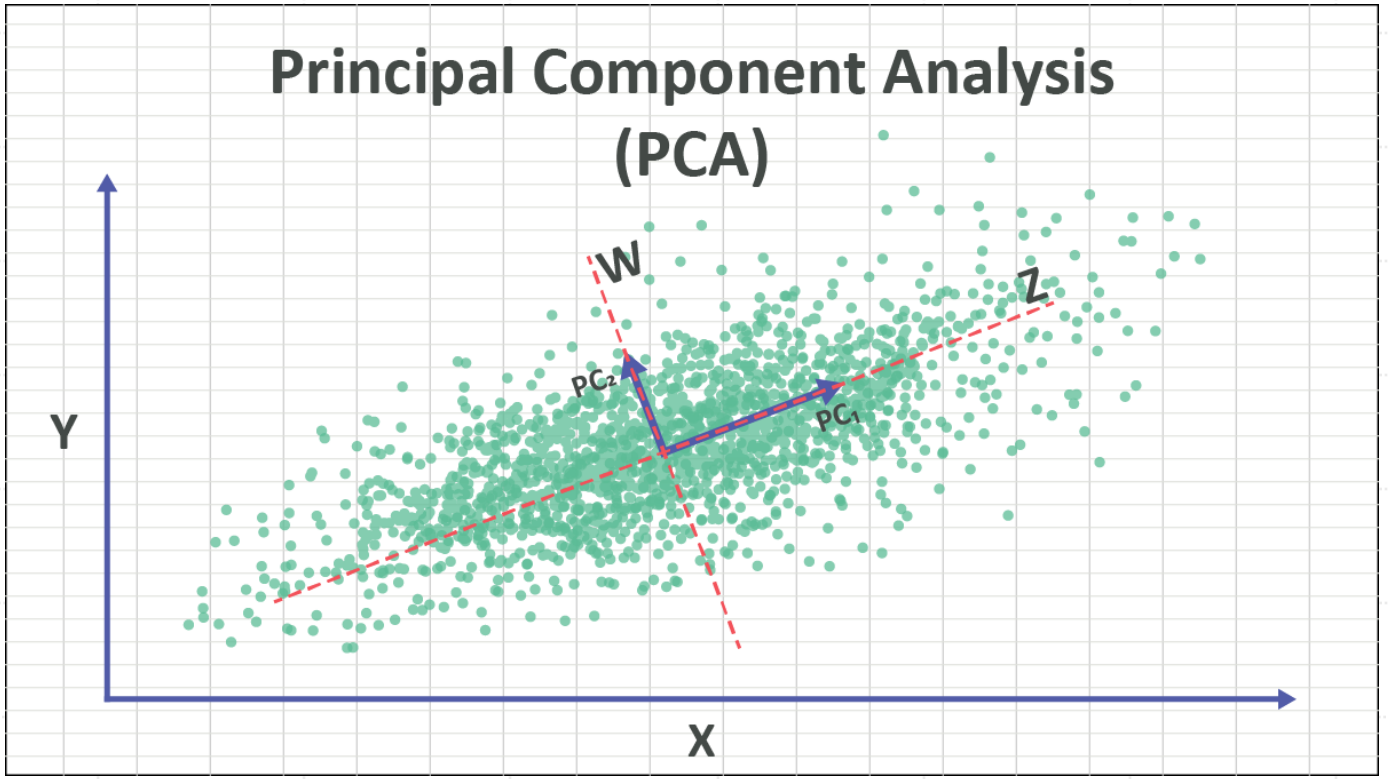
Principal Component Analysis (PCA)

PCA es un **resumen inteligente**: te deja mirar muchos datos a la vez desde el **mejor ángulo**.

- **Visualizar** en 2D/3D datos que tienen decenas de columnas.
- **Acelerar** modelos (menos columnas = menos cuentas).
- **Quitar ruido** y **duplicaciones** (variables muy parecidas entre sí).

Principal Component Analysis (PCA)

Imaginá una **nube de puntos**. Giramos el plano para mirar por la dirección donde la nube **cambia más** (PC1). La segunda mejor dirección, **en 90°**, es PC2.



Principal Component Analysis (PCA)

Cómo se calcula **cada PC**

- **Preparar** los datos: escalá y **centrá** (media 0) cada columna **usando solo TRAIN**.
- **Buscar la dirección con más variación**: el algoritmo mira la nube y encuentra la **flecha** por donde los puntos están más **esparcidos** → esa es **PC1**.
- **Forzar 90°**: ahora busca otra flecha **perpendicular** (90°) que capture la mayor variación restante → **PC2**.
- Repite para **PC3, PC4, …** siempre en **90°** con las anteriores (por eso las PCs son independientes entre sí).

PCA: Varianza explicada

"varianza explicada" = **qué porcentaje del dibujo original** conserva tu resumen.

¿Contra qué datos se calcula? Contra los **datos usados para entrenar el PCA** (normalizados/centrados). **No** contra todo el dataset: primero **fit en TRAIN**, luego **transform** en VALID/TEST (para evitar *leakage*).

Cómo se mide (idea simple):

- Cada PC trae un **valor de varianza** (cuánto "cuenta").
- El **ratio** de cada PC = varianza de esa PC / varianza total.
- La **acumulada** te dice cuánta info llevás con las primeras *k* PCs.

¿Con cuántos componentes? Reglas prácticas

- **Visualizar**: 2–3 PCs (solo para mirar).
- **Compactar/Acelerar**: elegí *k* tal que la **acumulada** sea **90–95%**

Selección de atributos

PCA hace un resumen mezclando columnas; la selección de atributos elige las mejores columnas y descarta el resto.

- **Filtro (rápidos):** quitá columnas casi constantes (**variance threshold**) y columnas **duplicadas/muy correlacionadas**.
- **Wrappers (prueba y error):** el modelo te dice qué conviene.
  - **Forward:** empezás con 0 y **vas sumando** columnas.
  - **Backward:** empezás con todas y **vas sacando**.
  - Usá una **métrica** para decidir
- **Embebidos (lo decide el modelo):**
  - **L1/Lasso:** deja varias **en cero** (se “apagan”).
  - **Árboles/Random Forest:** traen **importancia de variables**.
  - **Permutation importance:** medí cuánto **empeora** si desordenás una columna.

Clustering ≠ Clasificación

- **Supervisado** = examen **con respuestas** (entrenás con etiquetas).
- **No supervisado** = **explorás** y buscás **grupos** sin respuestas.

- En clustering **no hay "verdad"**: hay **segmentos útiles**.
- Métricas internas (Silhouette) ayudan, pero **negocio manda**.

K-Means: espacio, posición y distancia

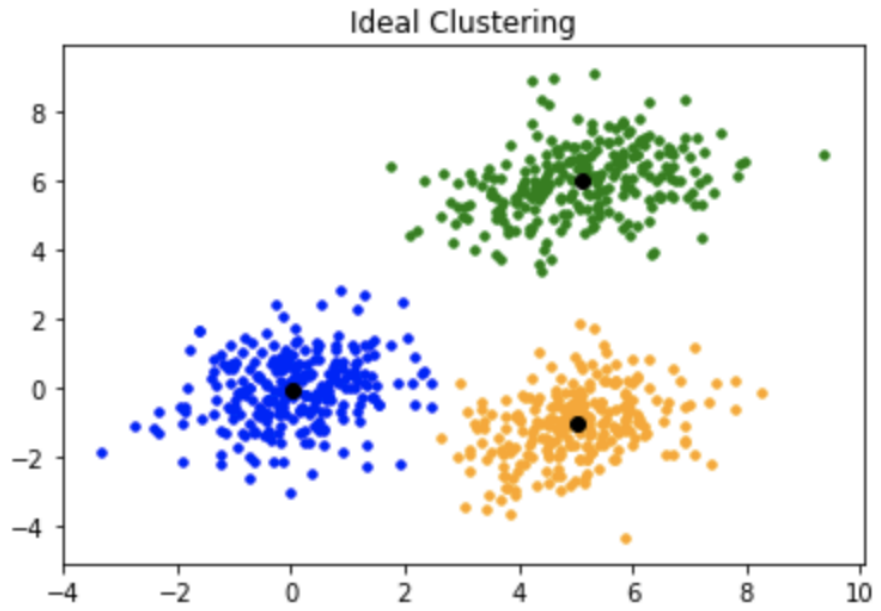
¿Qué es el *espacio*?

- Pensá que **cada columna** (ya **escalada**) es un **eje**.
- Si tenés 3 columnas (Edad, Ingresos, Tiempo), vivís en un **espacio 3D**. Con 20 columnas, es 20D (no lo vemos, pero **existe matemáticamente**).

K-Means: espacio, posición y distancia

¿Qué significa la *posición* de un punto?

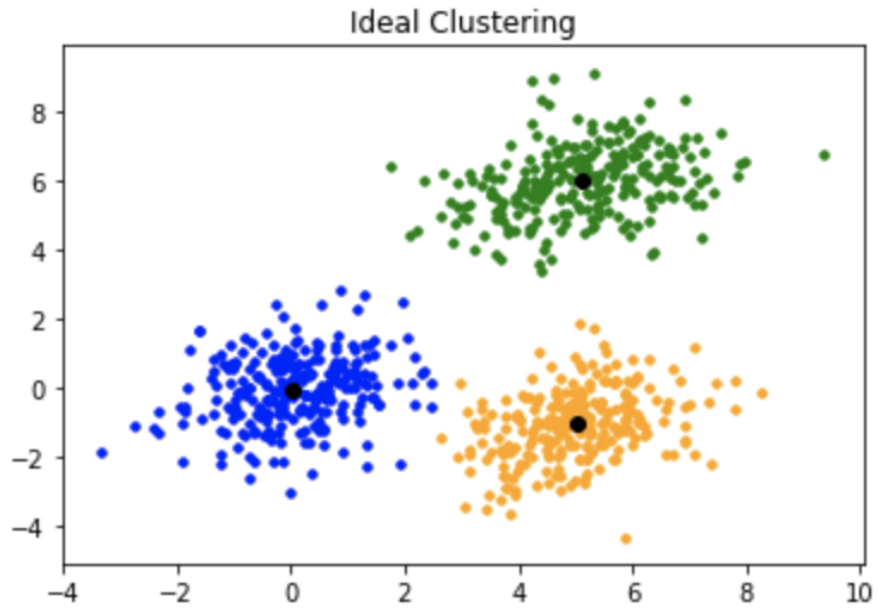
- Es simplemente su **vector de valores** en esos ejes:
  x = [edad_est, ingresos_est, tiempo_est, …].
- Por eso **escalar** es clave: si un eje tiene números gigantes, **domina** la posición y las distancias.



K-Means: espacio, posición y distancia

¿Qué es el *centroide*?

- Es el **promedio componente a componente** de los puntos del cluster: μ = mean(x_1, x_2, …, x_n).
- Intuición: el **punto "típico"** del grupo.



K-Means: espacio, posición y distancia

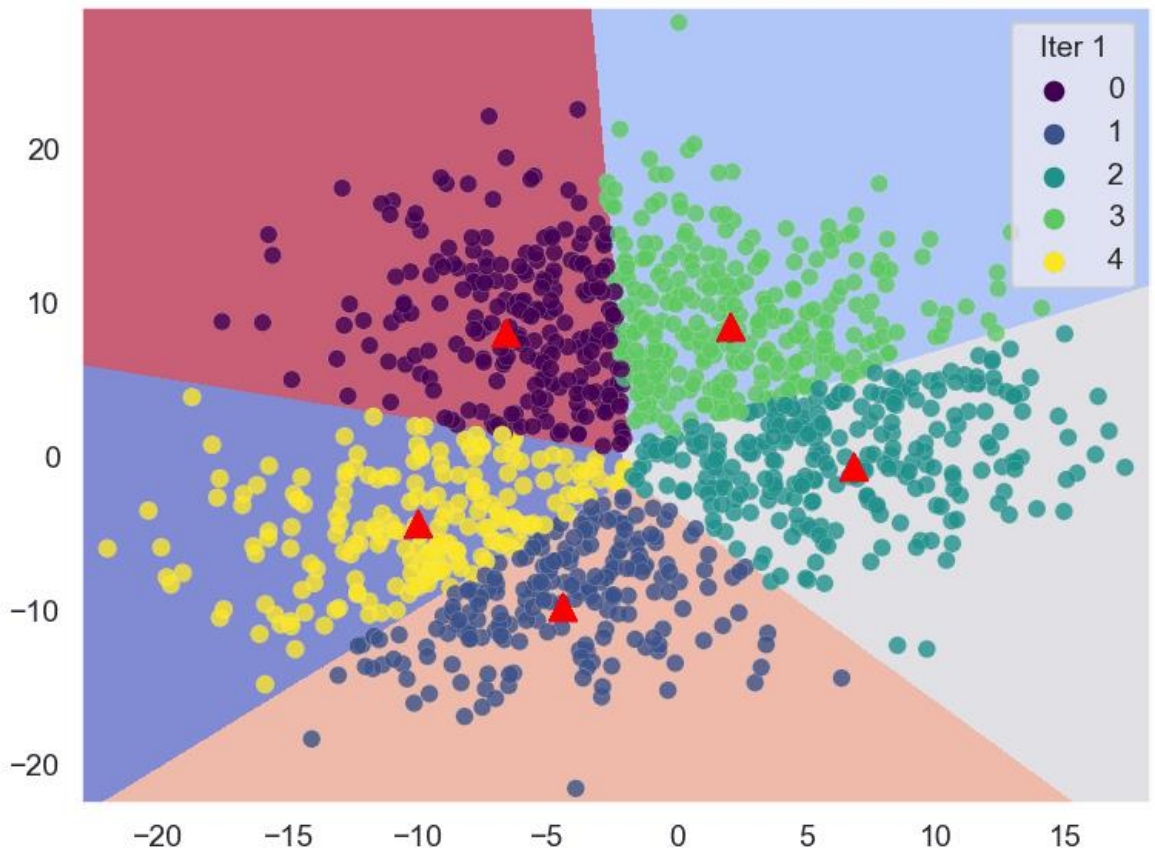
¿Cómo mide la *distancia*?

- K-Means clásico usa **distancia euclídea** (la de regla):
  "qué tan lejos" está el punto del centroide **sumando cuadraditos en cada eje**.
- El objetivo que minimiza es la **suma de distancias al cuadrado** (SSE).
  Por eso el mejor centroide resulta ser el **promedio** (no la mediana)

K-Means

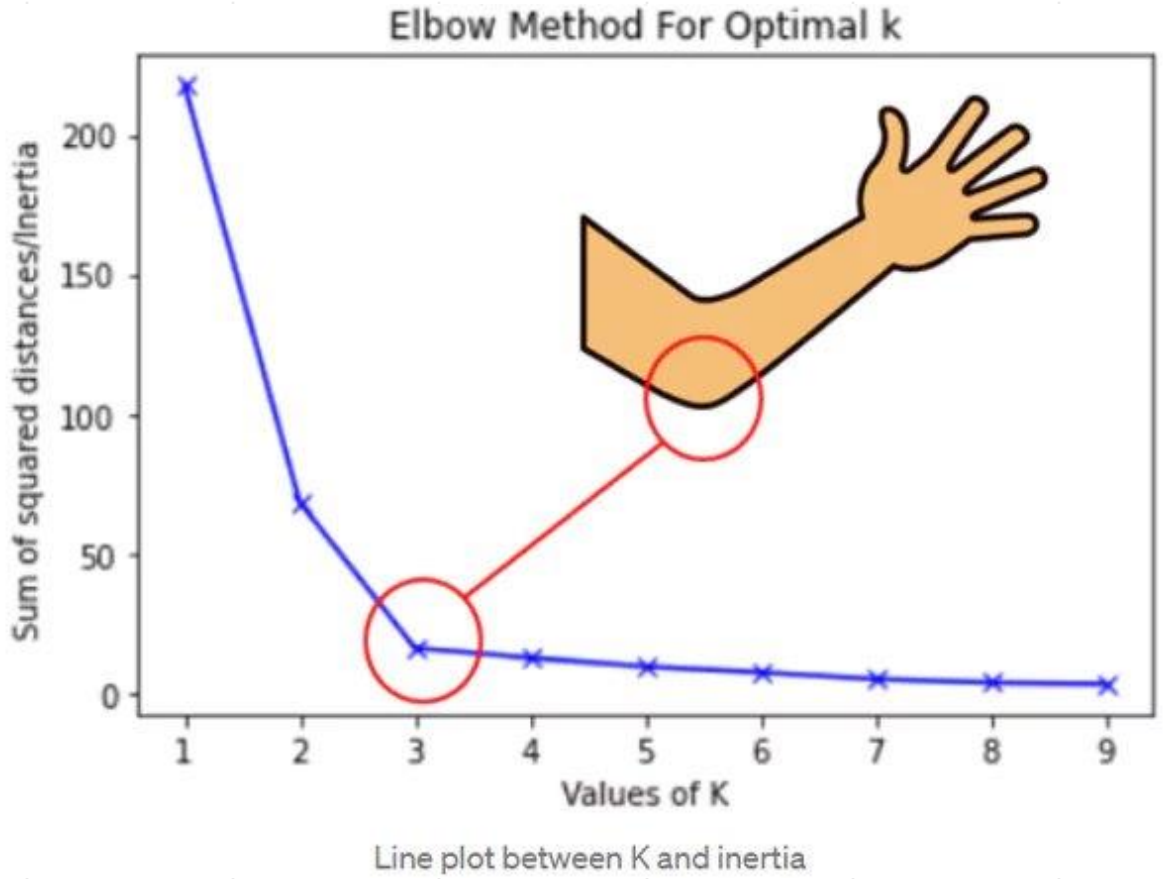
Juego de **imanes** y **puntos**.

- Poné **K imanes** al azar (centros).
- Cada punto se pega al imán más cercano.
- Mové cada imán al **promedio** de sus puntos.
- Repetí 2–3 hasta que todo se **quede quieto**.
- Resultado: K grupos compactos.



Elegir K: Codo

- **Codo**: mirá la curva; cuando **agregas más** y ya casi no mejora, ahí está el **codo**.



Line plot between K and inertia