

Algoritmos de Clustering:
K-Means, Jerárquico y DBSCAN
HE2: Consultoría Económica con IA Responsable

Santiago Neira & Catalina Bernal

Universidad de los Andes
Departamento de Economía

Febrero 2026

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

De PCA a Clustering: El Siguiente Paso

Clase pasada: PCA nos permitió *reducir dimensiones* y visualizar estructura.

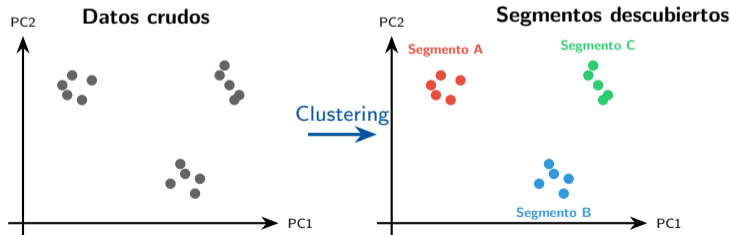
Hoy: Queremos *encontrar grupos* en los datos — sin etiquetas.

PCA respondía:

“¿Cuáles son las *direcciones* más importantes en los datos?”

Clustering responde:

“¿Existen *grupos naturales* en los datos?”



Sector público:

- Segmentar municipios por necesidades para focalización de programas sociales
- Tipologías de hogares para políticas de subsidios
- Perfiles de contribuyentes para estrategias de recaudo

Sector privado:

- Segmentación de clientes para marketing diferenciado
- Perfiles de riesgo crediticio
- Tipologías de zonas de carga eléctrica

Pregunta clave de IA responsable

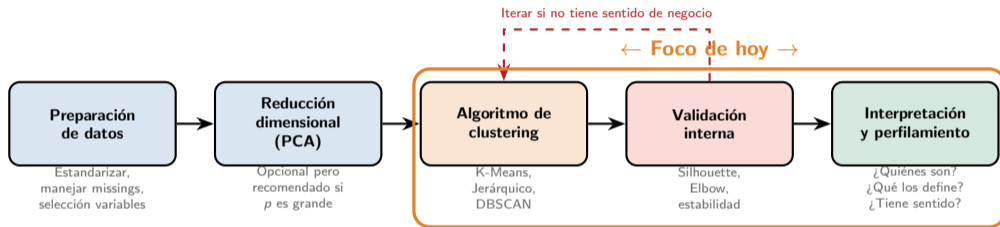
Cuando segmentamos personas, **¿quién define qué es un “grupo natural”?**

Los clusters no son verdad objetiva — son una **decisión de modelado** que tiene consecuencias reales.

Hoy aprenderemos

A elegir el algoritmo correcto, entender sus supuestos, y — críticamente — a **validar e interpretar** los segmentos resultantes.

El Pipeline Completo de Segmentación



Mensaje central

El algoritmo es solo una pieza. El valor real está en la **interpretación** y en la **accionabilidad** de los segmentos. Un clustering técnicamente perfecto pero que no se puede explicar al cliente es inútil.

- 1 ¿Por Qué Segmentar?
- 2 **K-Means**
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

Objetivo: Particionar n observaciones en K grupos minimizando la varianza intra-cluster.

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (1)$$

donde $\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$ es el **centroide** del cluster k .

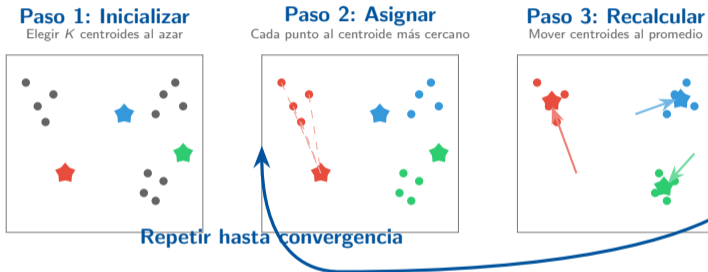
En palabras:

- Cada punto pertenece al cluster cuyo centro está más cerca
- Cada centro es el promedio de los puntos en su cluster
- “Más cerca” = distancia euclidiana

Observación importante

El problema exacto es NP-hard. K-Means usa un algoritmo **iterativo** (Lloyd's) que converge a un *óptimo local*.

K-Means: El Algoritmo Paso a Paso



Convergencia: Cuando las asignaciones no cambian (o el cambio en la función objetivo es $< \epsilon$). En la práctica: pocas iteraciones ($\sim 10-50$). <https://machinelearningcoban.com/2017/01/01/kmeans/>

K-Means: Sensibilidad a la Inicialización

Problema: Diferentes inicializaciones \rightarrow diferentes soluciones.

Buena inicialización



WCSS = 2.1

Mala inicialización



WCSS = 8.7 (óptimo local)

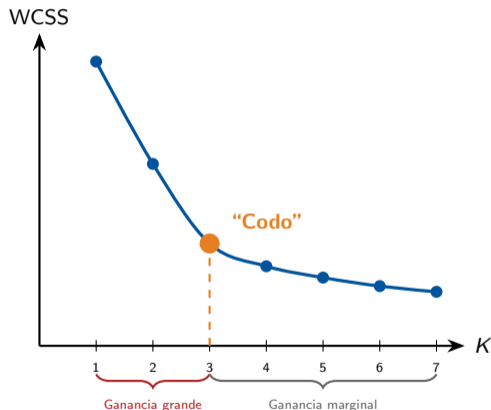
Solución: K-Means++

1. Primer centroide: aleatorio
 2. Sigüientes centroides: probabilidad proporcional a $d(x, \text{centroide más cercano})^2$
 3. Puntos lejanos tienen más probabilidad de ser elegidos
- En sklearn:**
`init='k-means++'`
(es el default)

Buena práctica

Correr K-Means **múltiples veces** (`n_init=10` por default en sklearn) y quedarse con la solución de menor WCSS.

Elegir K : El Método del Codo (Elbow)



WCSS (Within-Cluster Sum of Squares):

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

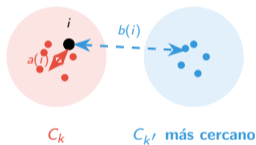
- WCSS *siempre* decrece con K
- Si $K = n$: WCSS = 0 (trivial)
- Buscamos el punto donde la ganancia marginal "se aplana"

Advertencia

El codo es **subjetivo**. En datos reales, rara vez hay un codo claro. Combinar con Silhouette y con sentido de negocio.

Para cada observación i en el cluster C_k :

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \in [-1, 1] \quad (2)$$



$a(i)$ = dist. promedio a puntos de **su** cluster
 $b(i)$ = dist. promedio al cluster **vecino más cercano**

Interpretación:

$s(i)$	Significado
≈ 1	Bien asignado
≈ 0	En la frontera
< 0	Prob. mal asignado

Promedio global: $\bar{s} = \frac{1}{n} \sum_{i=1}^n s(i)$

Elegir K que maximice \bar{s} .

Ventaja: No solo da el K óptimo, sino que identifica *qué* observaciones están mal clasificadas.

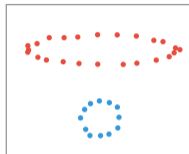
Supuestos implícitos:

- Clusters son **esféricos** (isotropía)
- Clusters tienen **tamaño similar**
- Clusters tienen **densidad similar**
- K se conoce *a priori*

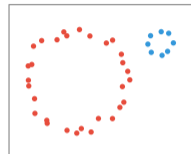
Fortalezas:

- Rápido: $O(nKd \cdot \text{iter})$
- Escala a millones de observaciones
- Fácil de interpretar y explicar
- Determinístico dado la inicialización

K-Means falla aquí



Tamaños desiguales



Izquierda: Clusters elongados — K-Means los partirá al medio.

Derecha: Un cluster grande “absorbe” puntos del pequeño.

Para el consultor

K-Means es el **punto de partida** por su velocidad e interpretabilidad. Pero si los clusters no son aproximadamente esféricos o tienen densidades muy distintas, necesitamos otras herramientas.

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 **Clustering Jerárquico**
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

En lugar de fijar K de antemano, construimos una **jerarquía completa** de agrupaciones.

Aglomerativo (bottom-up):

- 1 Cada observación es su propio cluster
- 2 En cada paso, fusionar los dos clusters *más cercanos*
- 3 Repetir hasta tener un solo cluster

Es el más usado en la práctica.

Divisivo (top-down):

- 1 Todos en un solo cluster
- 2 En cada paso, dividir el cluster más heterogéneo
- 3 Repetir hasta que cada punto sea un cluster

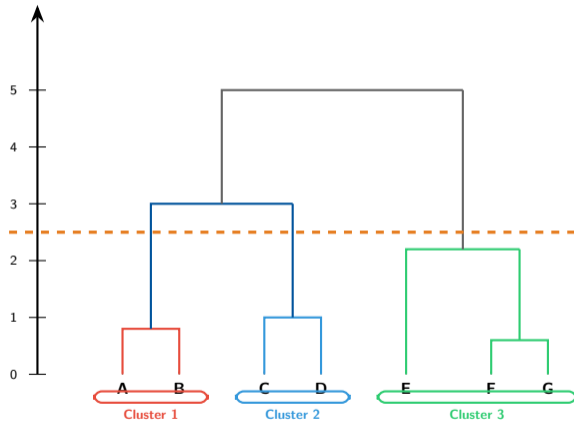
Menos común, computacionalmente costoso.

Ventaja fundamental

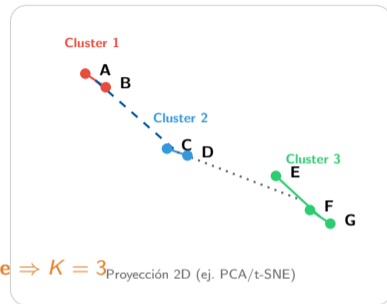
No hay que elegir K antes de correr el algoritmo. La jerarquía nos permite explorar múltiples niveles de granularidad con una sola corrida.

El Dendrograma: Leyendo la Jerarquía - Veamos un ejemplo en el tablero

Distancia



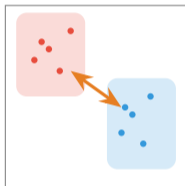
Corte $\Rightarrow K = 3$ Proyección 2D (ej. PCA/t-SNE)



Lectura: La altura de cada fusión indica la *disimilitud* entre los clusters fusionados. Cortar a diferentes alturas \Rightarrow diferentes números de clusters.

Pregunta clave: ¿Cómo medimos la distancia *entre clusters*?

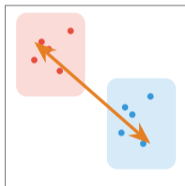
Single (mínimo)



$$d(A, B) = \min_{a \in A, b \in B} d(a, b)$$

Tiende a crear cadenas

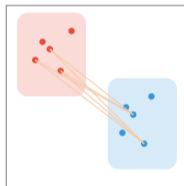
Complete (máximo)



$$d(A, B) = \max_{a \in A, b \in B} d(a, b)$$

Clusters compactos

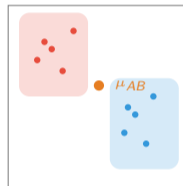
Average (promedio)



$$d(A, B) = \frac{1}{|A||B|} \sum_{a, b} d(a, b)$$

Buen balance, más robusto

Ward



$\Delta WCSS$ al fusionar A y B

Minimiza varianza, como K-Means

Recomendación para consultoría

Ward = default más robusto (clusters compactos, análogo a K-Means). **Average** = buena alternativa si clusters tienen tamaños distintos.

Fortalezas:

- No requiere fijar K a priori
- Dendrograma: herramienta visual poderosa
- Puede capturar clusters no esféricos (single linkage)
- Determinístico

n	Memoria	Tiempo
1,000	~8 MB	< 1 s
10,000	~800 MB	~30 s
50,000	~20 GB	minutos
100,000	~80 GB	×

Limitaciones:

- Complejidad: $O(n^2 \log n)$ tiempo, $O(n^2)$ espacio
- **No escala:** $n > 10,000$ impracticable
- Fusiones **irreversibles** (greedy)

En la práctica

Para datasets grandes: K-Means o DBSCAN. Jerárquico es ideal para **exploración** con $n < 10,000$.

Estrategia híbrida: Jerárquico en muestra para explorar K , luego K-Means al dataset completo.

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN**
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

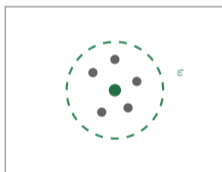
DBSCAN: Clustering Basado en Densidad

Idea: Un cluster es una zona *densa* separada de otras zonas densas por zonas de *baja densidad*.

Dos hiperparámetros:

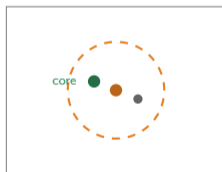
- ϵ (epsilon): radio de vecindad
- `min_samples`: mínimo de puntos para ser “denso”

Punto core



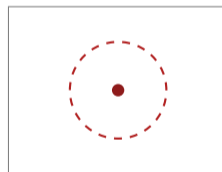
$\geq \text{min_samples}$ vecinos
dentro de ϵ

Punto frontera



$< \text{min_samples}$ vecinos
pero alcanzable desde un core

Punto ruido



No es core ni alcanzable
desde ningún core

Diferencia fundamental

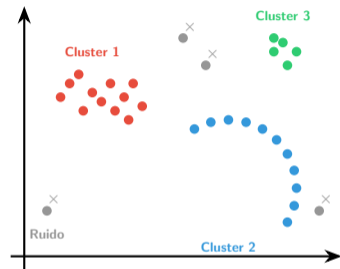
DBSCAN **no asigna todos los puntos** a un cluster. Puntos en zonas de baja densidad = **ruido** (-1). Poderoso para detectar outliers.

Procedimiento:

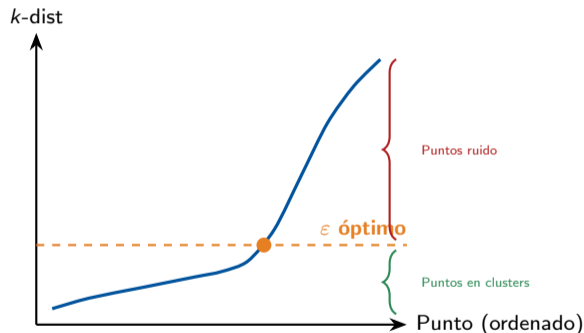
- 1 Para cada punto, contar vecinos dentro de ϵ
- 2 Clasificar en **core**, **frontera**, o **ruido**
- 3 Conectar puntos core que son vecinos entre sí
- 4 Cada componente conexa de puntos core = un cluster
- 5 Asignar puntos frontera al cluster de su core más cercano

Complejidad:

- Con KD-tree: $O(n \log n)$
- Sin índice espacial: $O(n^2)$
- En alta dimensión ($d > 20$): KD-tree pierde eficiencia.
- Ver animación



DBSCAN puede encontrar clusters de **forma arbitraria** — algo que K-Means no puede hacer.



Método:

- 1 Para cada punto, calcular la distancia a su k -ésimo vecino más cercano (donde $k = \text{min_samples}$)
- 2 Ordenar estas distancias de menor a mayor
- 3 Buscar el "codo" en la curva

Regla general: $\text{min_samples} \geq d + 1$ donde d es la dimensión.

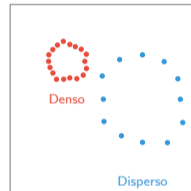
Fortalezas:

- No requiere fijar K
- Descubre clusters de **forma arbitraria**
- **Detecta outliers** automáticamente
- Robusto a ruido
- Determinístico (para puntos core)

Limitaciones:

- Sensible a ϵ y `min_samples`
- Problemas con **densidades variables**
- En alta dimensión: distancias se concentran ("curse of dimensionality")
- No produce centroides interpretables

Problema: densidades variables



Un ϵ pequeño captura el cluster denso pero fragmenta el disperso.
Un ϵ grande une el disperso pero fusiona los dos clusters.

Alternativas: HDBSCAN (adaptativo), OPTICS.

Para explorar

```
from sklearn.cluster import HDBSCAN  
Selecciona  $\epsilon$  automáticamente por zona.
```

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad**
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

Comparación de los Tres Algoritmos

	K-Means	Jerárquico	DBSCAN
Forma clusters	Esféricos	Según linkage	Arbitraria
Requiere K	Sí	No (corte post.)	No
Detecta outliers	No	No	Sí
Determinístico	No	Sí	Sí*
Centroides	Sí	No	No
Complejidad	$O(nKd)$	$O(n^2 \log n)$	$O(n \log n)^{**}$
Escala a n grande	Millones	<10K	Según d
Interpretabilidad	Alta	Media	Baja

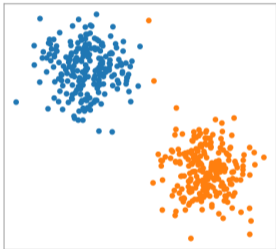
*Puntos frontera pueden variar. **Con KD-tree; $O(n^2)$ en alta dimensión.

Guía rápida de decisión

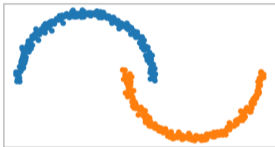
$n > 50K \Rightarrow$ K-Means | No convexos \Rightarrow DBSCAN | Explorar jerarquía \Rightarrow Jerárquico en muestra | Outliers \Rightarrow DBSCAN | Explicar al cliente \Rightarrow K-Means

¿Cuándo Cada Forma de Cluster Importa?

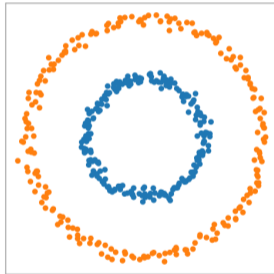
K-Means: OK



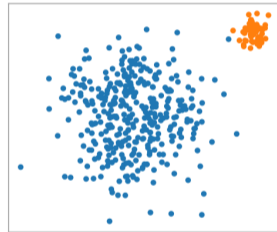
DBSCAN: OK



DBSCAN: OK



Jerarquico (single): OK



- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos**
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos

Un vector de etiquetas $\{0, 1, 2\}$ no le dice nada al cliente.

El trabajo del consultor **empieza** cuando el algoritmo termina:

- 1 **Perfilar:** ¿Qué caracteriza a cada segmento?
- 2 **Nombrar:** Darle nombres interpretables y accionables
- 3 **Validar:** ¿Los segmentos tienen sentido de negocio?
- 4 **Dimensionar:** ¿Cuántos son? ¿Cuánto valen?
- 5 **Accionar:** ¿Qué hacemos diferente con cada segmento?

Error común del data scientist junior

Entregar una tabla con “Cluster 0, Cluster 1, Cluster 2” y sus promedios. Eso no es consultoría — es un output de Python.

Paso 1: Perfilamiento con Estadísticas Descriptivas

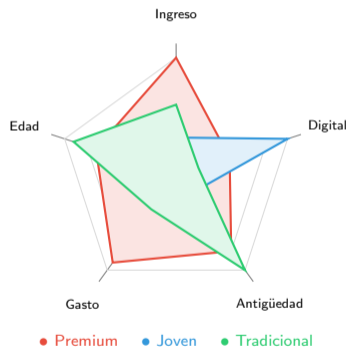
Para cada cluster, calcular estadísticas de las variables originales (no estandarizadas):

Variable	Cluster 0 (n=1,234)	Cluster 1 (n=2,567)	Cluster 2 (n=890)	Total (n=4,691)
Ingreso mensual (miles)	\$8,200	\$2,100	\$4,500	\$4,200
Edad promedio	42	28	55	38
Gasto mensual (miles)	\$5,800	\$1,900	\$2,200	\$3,100
% con crédito hipotecario	72 %	15 %	85 %	45 %
Meses como cliente	48	12	72	36
Nombre propuesto	Premium activo	Joven digital emergente	Tradicional estable	

Índice de caracterización

Para variable j en cluster k : Índice $_k^j = \bar{x}_k^j / \bar{x}_{\text{total}}^j$. Valor de 1.5 = 50 % sobre el promedio.

Radar chart / Spider plot:



Muestra el “perfil” de cada segmento de forma intuitiva para el cliente.

Heatmap de índices:

	Premium	Joven	Tradic.
Ingreso	1.95	0.50	1.07
Edad	1.11	0.74	1.45
Gasto	1.87	0.61	0.71
Antigüedad	1.33	0.33	2.00

>1.2: sobre-representado <0.8: sub-representado

¿Qué más visualizar?

- Distribuciones por cluster (boxplots)
- PCA scatter coloreado por cluster
- Mapas si hay variable geográfica

Paso 3: Validación — ¿Son Reales Estos Segmentos?

Validación interna (estadística):

- **Silhouette promedio:** $\bar{s} > 0,5$ = estructura razonable; $> 0,7$ = fuerte
- **Estabilidad:** Correr en submuestras (bootstrap) y ver si los clusters se mantienen

Validación externa (negocio):

- ¿Los segmentos se comportan *diferente* en una variable no usada para el clustering?
- ¿Un experto de dominio los reconoce?
- ¿Son *accionables*?

Test de accionabilidad

Para cada segmento, responder:

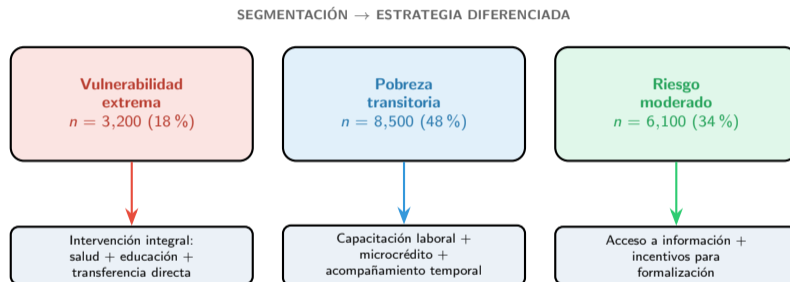
- 1 ¿Puedo *identificar* a un individuo nuevo como parte de este segmento?
- 2 ¿Puedo *diseñar una acción diferenciada* para este segmento?
- 3 ¿El segmento es lo suficientemente *grande* para justificar una estrategia diferente?

Si alguna respuesta es “no”, considerar fusionar o replantear.

Trampa de la sobre-segmentación

$K = 10$ puede dar mejor silhouette, pero ¿el equipo comercial puede manejar 10 estrategias distintas? Menos es más.

Ejemplo: Segmentación de beneficiarios para programa social



Entregable en consultoría

El output no es un .csv con labels. Es un **documento** que dice: “Identificamos 3 perfiles de beneficiarios. Para cada uno, recomendamos estrategias diferenciadas. Aquí está la evidencia y la lógica.”

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación**
- 8 Resumen y Próximos Pasos

¿Qué Puede Salir Mal?

1. Proxies de variables protegidas

Si incluimos código postal, barrio, o tipo de colegio, los clusters pueden terminar siendo proxies de raza, etnia o estrato socioeconómico.

2. Reificación de segmentos

Los clusters son artefactos estadísticos, no categorías naturales. Tratar a las personas *solo* según su cluster ignora la heterogeneidad intra-grupo.

3. Ciclos de retroalimentación

Si el segmento “alto riesgo” recibe peores condiciones crediticias, sus resultados empeoran, confirmando la etiqueta.



Pregunta obligatoria

Antes de entregar una segmentación, preguntarse: **¿Si alguien en este segmento viera su etiqueta, la consideraría justa?**

Checklist de IA Responsable para Segmentación

☐ **Variables:** ¿Incluimos alguna variable que sea proxy de raza, género, etnia, o religión?

☐ **Composición:** ¿Los clusters reflejan desproporcionadamente a un grupo demográfico?

☐ **Granularidad:** ¿El nivel de segmentación es proporcional a la acción que se tomará?

☐ **Acciones:** ¿La acción diferenciada *beneficia* o *perjudica* a algún segmento?

☐ **Dinamismo:** ¿Un individuo puede cambiar de segmento, o la etiqueta es permanente?

☐ **Transparencia:** ¿Podemos explicar los criterios de segmentación a los afectados?

☐ **Retroalimentación:** ¿Las acciones podrían reforzar desigualdades existentes?

Regla del consultor responsable

Si no puedes explicar *por qué* cada segmento existe y *qué* se hará con esa información, no entregues la segmentación.

- 1 ¿Por Qué Segmentar?
- 2 K-Means
- 3 Clustering Jerárquico
- 4 DBSCAN
- 5 Comparación y Escalabilidad
- 6 Post-Processing: Identificar y Perfilar Segmentos
- 7 IA Responsable en Segmentación
- 8 Resumen y Próximos Pasos**

K-Means

Rápido y escalable
Clusters esféricos
Centroides interpretables
Elegir K con Elbow + Silhouette

Jerárquico

No requiere fijar K
Dendrograma exploratorio
No escala ($n < 10K$)
Ward \approx K-Means en resultado

DBSCAN

Formas arbitrarias
Detecta outliers
Sensible a ϵ
Sin centroides ni K

El algoritmo es solo el 30 % del trabajo.

Perfilar, validar, nombrar, y accionar los segmentos es el **70 % que genera valor.**

Y siempre preguntarse: **¿es justo? ¿es explicable? ¿a quién afecta?**

Libros de texto:

- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (ISLR), 2nd Ed. Cap. 12.4: Clustering Methods. <https://www.statlearning.com>
- Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning* (ESL), 2nd Ed. Cap. 14.3: Cluster Analysis.
- Müller, A.C. & Guido, S. (2016). *Introduction to Machine Learning with Python*. O'Reilly. Cap. 3: Unsupervised Learning.

Artículos clave:

- Arthur, D. & Vassilvitskii, S. (2007). "k-means++: The advantages of careful seeding." *SODA '07*.
- Ester, M., Kriegel, H.P., Sander, J. & Xu, X. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." *KDD '96*.
- Rousseeuw, P.J. (1987). "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." *J. Comput. Appl. Math.*, 20, 53–65.

IA Responsable:

- Barocas, S. & Selbst, A.D. (2016). "Big Data's Disparate Impact." *California Law Review*, 104(3).

¿Preguntas?

Santiago Neira & Catalina Bernal
HE2: Consultoría Económica con IA Responsable
Universidad de los Andes — 2026-I