Aprendizaje Automático No Supervisado

Alberto Barbado González

Tema 2 – Fundamentos y aplicaciones del agrupamiento K-Means

UNIVERSIDAD INTERNACIONAL DE LA RIOJA

La Universidad en Internet

La pregunta del día

¿Cómo podemos agrupar un conjunto de datos de clientes de una tienda en diferentes segmentos sin conocer previamente su comportamiento de compras?

Encuesta previa

- ¿Cómo explicarías de manera intuitiva el objetivo de las técnicas de clustering?

- ¿Ejemplos de aplicación de clustering en el día a día?

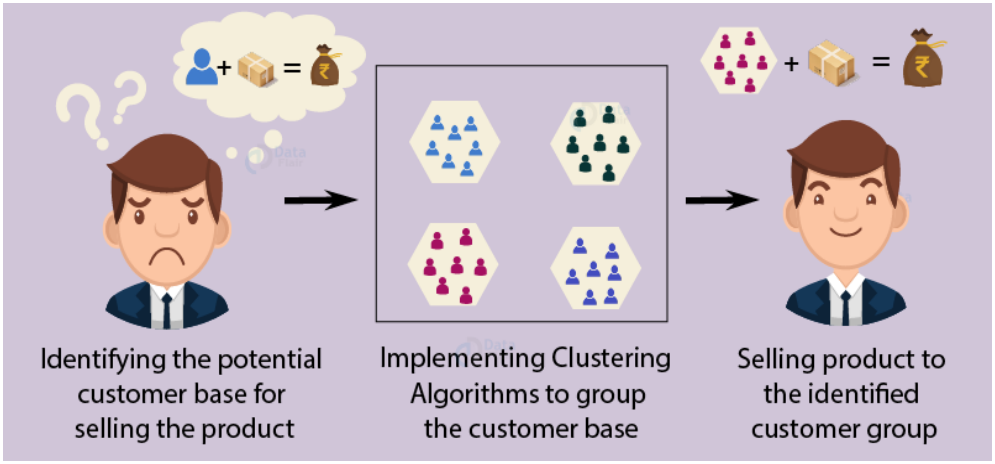
- ¿Cómo podemos saber que los resultados son buenos si no disponemos de etiquetas?

En el día de hoy

- Clustering: objetivos e intuición inicial
- Métricas de distancia y su papel en los algoritmos de clustering
- Algoritmo K-Means: Implementación y ajuste de hiperparámetros
- Algoritmo K-Means: Ventajes, desventajas y consideraciones
- Ejemplos de aplicaciones de Clustering

Clustering: Objetivo

- **Clustering:** Agrupar **datos similares** sin que existan etiquetas o clasificaciones previas. Su objetivo es identificar patrones o estructuras ocultas en los datos.

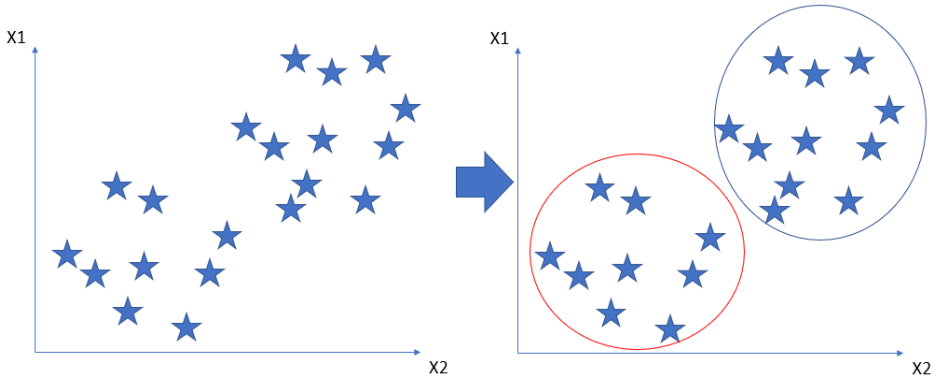


Perfiles de clientes como

- Clientes jóvenes con ingresos bajos y poca frecuencia de compra.
- Clientes de mediana edad con ingresos medios y compras moderadas.
- Clientes mayores con altos ingresos y compras frecuentes.

https://github.com/NelakurthiSudheer/Mall-Customers-Segmentation

Clustering: Intuición inicial



Matriz de rasgos (*features*): MR

Cliente	Edad	Ingresos (USD)	Frecuencia de Compra (compras/mes)	Total Gastado (USD)
1	25	3000	2	500
2	40	7000	5	3000
3	35	4500	3	1200
4	28	3200	1	200
5	50	10000	7	8000

$MR_{n\,x\,p}$ *con* $\boldsymbol{n}$ *el nº de datos y* $\boldsymbol{p}$ *el nº de features de entrada*

- Con ello, se busca encontrar vectores de datos (***data points***) que sean **similares entre sí.**
- Estos *data points* se **agruparían** dentro de un **mismo conjunto**.
- **Entrada**: Los datos de entrada son **matrices de *features***, similar a otros modelos de ML (e.j., clientes x atributos clientes).
- **Salida**: Cada fila (e.j., cliente) se vinculará con un cluster en función de sus columnas (rasgos/*features*)

Métricas de distancias

- Recordatorio: Distancia Euclídea

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

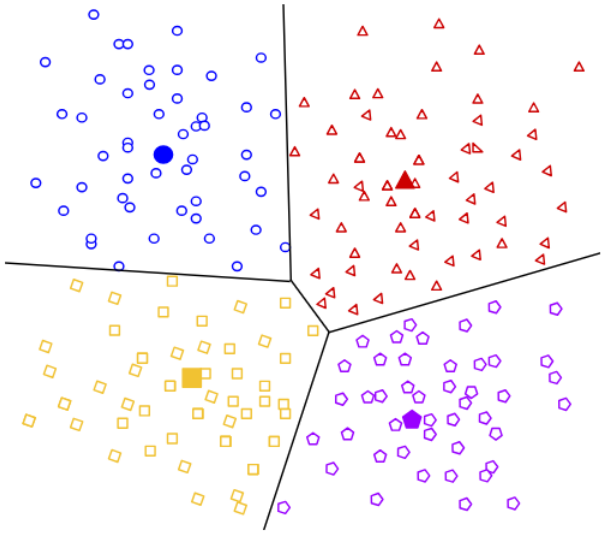
Cliente	Edad	Ingresos (USD)	Frecuencia de Compra (compras/mes)	Total Gastado (USD)
1	25	3000	2	500
2	40	7000	5	3000

$$d(cliente_2, cliente_1) = \sqrt{(40-25)^2 + (7000-3000)^2 + (5-2)^2 + (3000-500)^2} \approx 4717$$

- Las métricas de distancia sirven para dar similitud entre vectores.
- En el enfoque de clustering, dan **similitud entre vectores de *features*** (e.j., entre clientes).
- Un ejemplo de estas métricas es la **distancia Euclídea**.
- Las distancias cumplen ciertas propiedades:
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$

Clustering: Conceptos iniciales

- **Centroide**: Punto central del cluster. Se calcula como el promedio de las observaciones que pertenecen a ese cluster.



Cliente	Edad	Ingresos (USD)	Frecuencia de Compra (compras/mes)	Total Gastado (USD)
1	25	3000	2	500
2	28	3500	3	1000
3	30	3200	1	800

Cluster

$$C_{k,j} = \frac{\sum_i^n x_i}{n}$$

Edad	Ingresos (USD)	Frecuencia de Compra (compras/mes)	Total Gastado (USD)
27.67	3233.33	2.00	766.67

https://developers.google.com/machine-learning/clustering/clustering-algorithms

Métricas de distancias: Aplicación a clustering

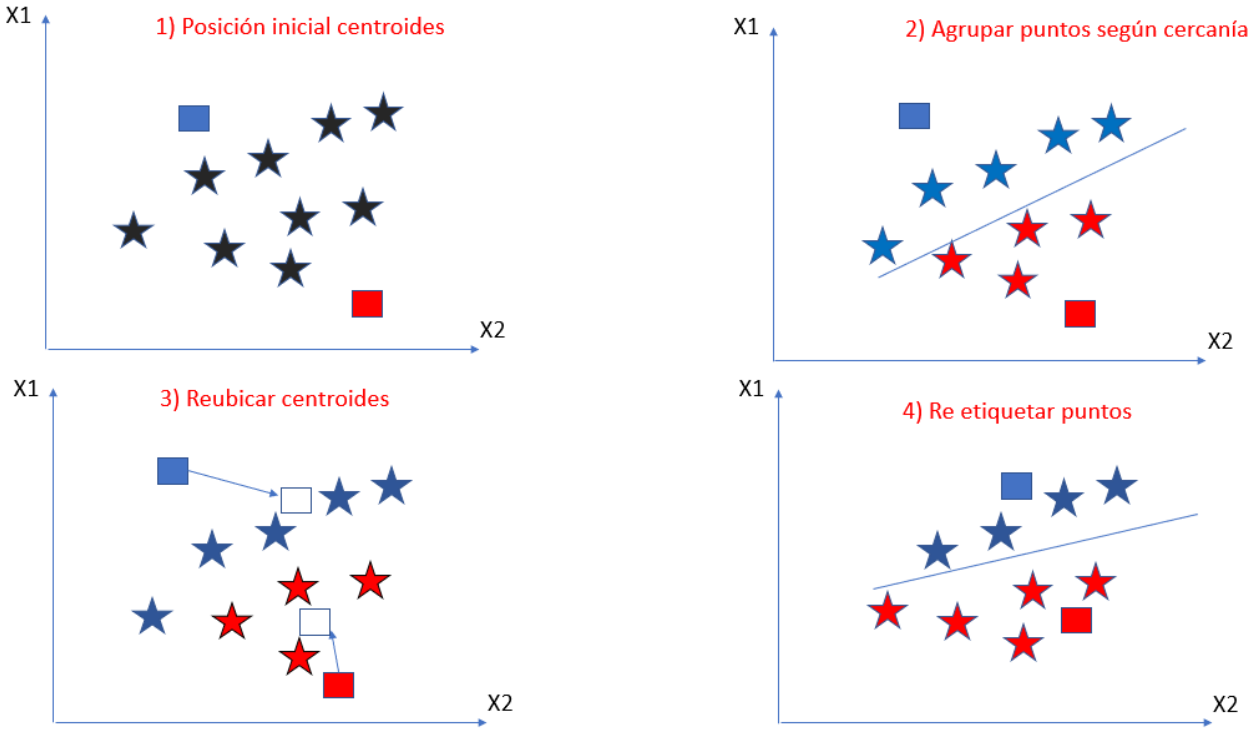
- Matriz de distancias (MD)

$$MD = \begin{bmatrix} 0 & d(1,2) & \cdots & d(1,n-1) & d(1,n) \\ d(2,1) & 0 & \cdots & d(2,n-1) & d(2,n) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d(n,-1\ 1) & d(n-1,2) & \cdots & 0 & d(n,-1n) \\ d(n,1) & d(n,2) & \cdots & d(n,n-1) & 0 \end{bmatrix}$$

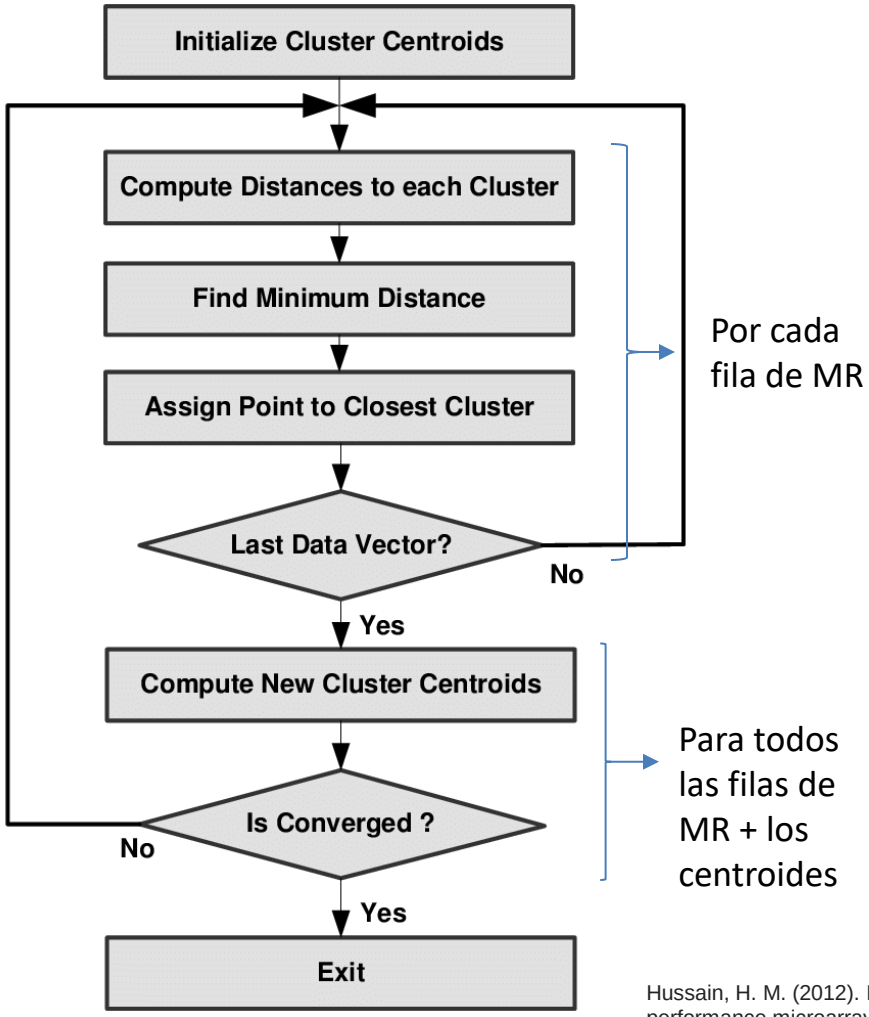
- El concepto de distancia aplicado a las técnicas de clustering
- Necesitaremos **conocer todas las distancias** entre cada punto y cada centroide

Algoritmo K-Means

- **K-Means**: Agrupa datos en **K clústeres asignando** cada dato al **centroide más cercano**. **Recalcula** los centroides hasta que se **estabilizan**, buscando **minimizar la distancia** entre los datos y sus centroides para formar clústeres compactos.



Algoritmo K-Means



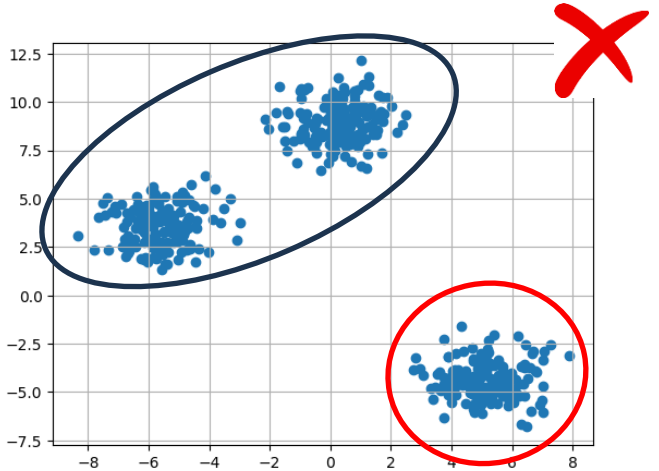
Se eligen los **parámetros del modelo** (e.j., número de clusters, K)

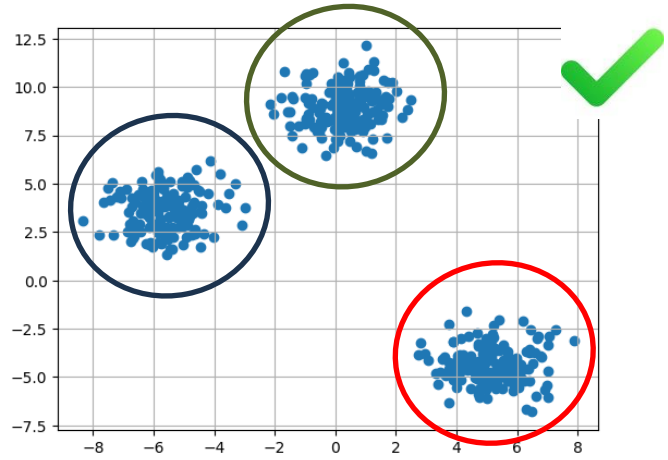
1. **Inicialización:** Se eligen **K puntos aleatorios** como centroides (valores aleatorios, no pertenecientes a MR).
2. **Asignación:** Cada fila de MR se **asigna al centroide k más cercano** (menor distancia). Se tienen así K clusters.
3. **Reubicación:** Se **calcula el nuevo centroide** en función de los datos de cada cluster.
4. **Repetición:** Se analiza **convergencia**. Por ejemplo:
   - Si "cambian mucho" los centroides -> No converge -> Se vuelve al **punto 2**.
   - Si "no cambian mucho" los centroides, se termina.

Hussain, H. M. (2012). Dynamically and partially reconfigurable hardware architectures for high performance microarray bioinformatics data analysis.

Hiperparámetros en K-Means: ¿Qué hacer?

- K-Means tiene distintos **hiperparámetros** (como otros modelos de ML).
- Uno de ellos es K: hay que **elegir el número de clústers** que se va a generar.
- ¿Cómo elegirlo si no **conozco los datos**?
- ¿Cómo elegirlo si **no hay una etiqueta** vs la que minimizar las predicciones? (e.j. ajuste hiperparametros en datos validación)
- Intuitivamente vemos en el ejemplo: 3 clusters mejor que 2 (en el primer caso hay datos no tan parecidos).





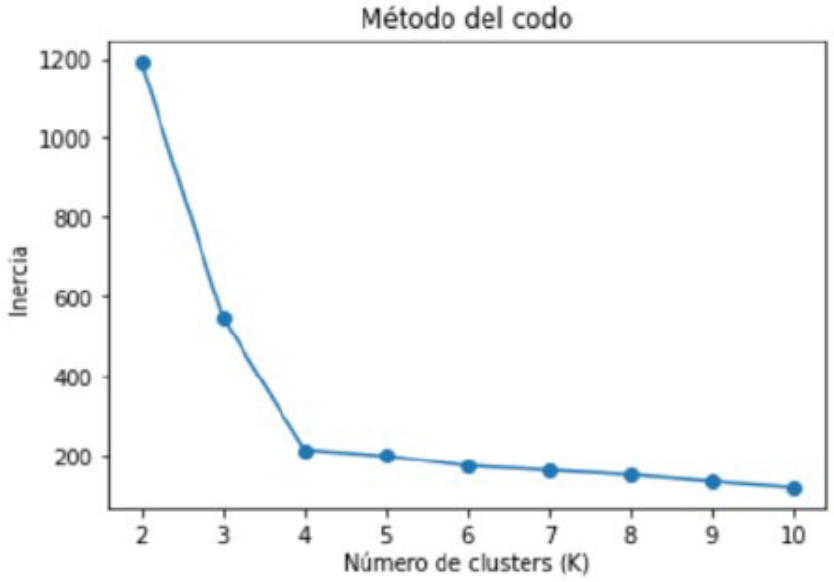
https://www.geeksforgeeks.org/k-means-clustering-introduction/

Selección de K: Método del codo

Inercia

$$I = \sum_{i=1}^{n} \|x_i - \mu_{c(i)}\|^2$$

- n es el número total de puntos.
- x_i es el i-ésimo punto de datos.
- $\mu_{c(i)}$ es el centroide del clúster al que pertenece el punto x_i .
- $\|x_i - \mu_{c(i)}\|^2$ es la distancia euclidiana cuadrada entre el punto x_i y su centroide correspondiente $\mu_{c(i)}$.



- Inercia como métrica cuantitativa que mide en qué casos se tiene un mejor clustering que en otros.
- Refleja la **heterogeneidad** de los clusters (un cluster con puntos muy lejanos del centroide, mayor heterogeneidad e inercia, peor clustering).
- Para distintos valores de K se muestra la inercia de cara a ver qué valor de K da mejores resultados (método del codo).
- El concepto de Inercia lo podemos ver también como **WCSS (within-cluster sum of squares)**

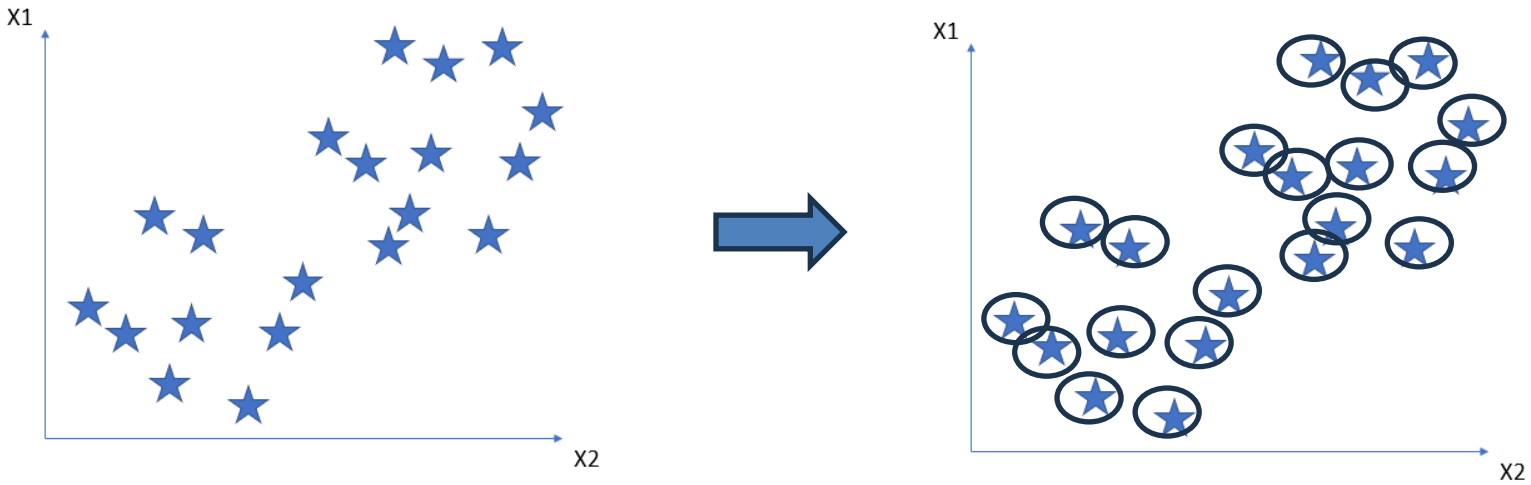
Método del codo: Consideraciones

- Elegir sólo en base a minimizar heterogeneidad tiene limitaciones…

¿Agrupación con menor Inercia?

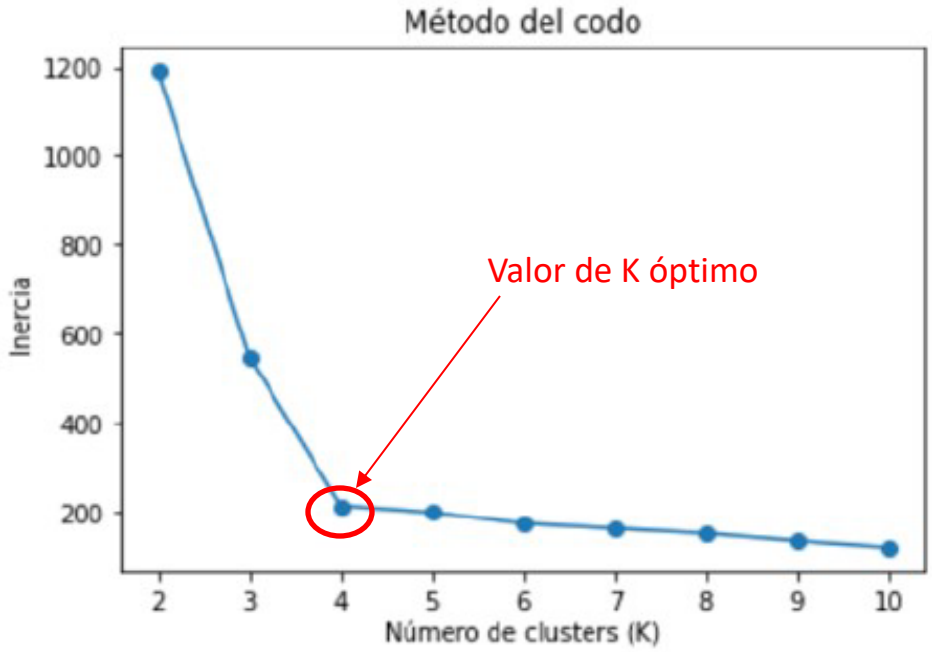
Método del codo: Consideraciones

- Tantos clusters como puntos…



- Resultado nada útil… tendríamos 1 cluster por fila de MR…
- Aquí la Inercia es, de hecho, 0.

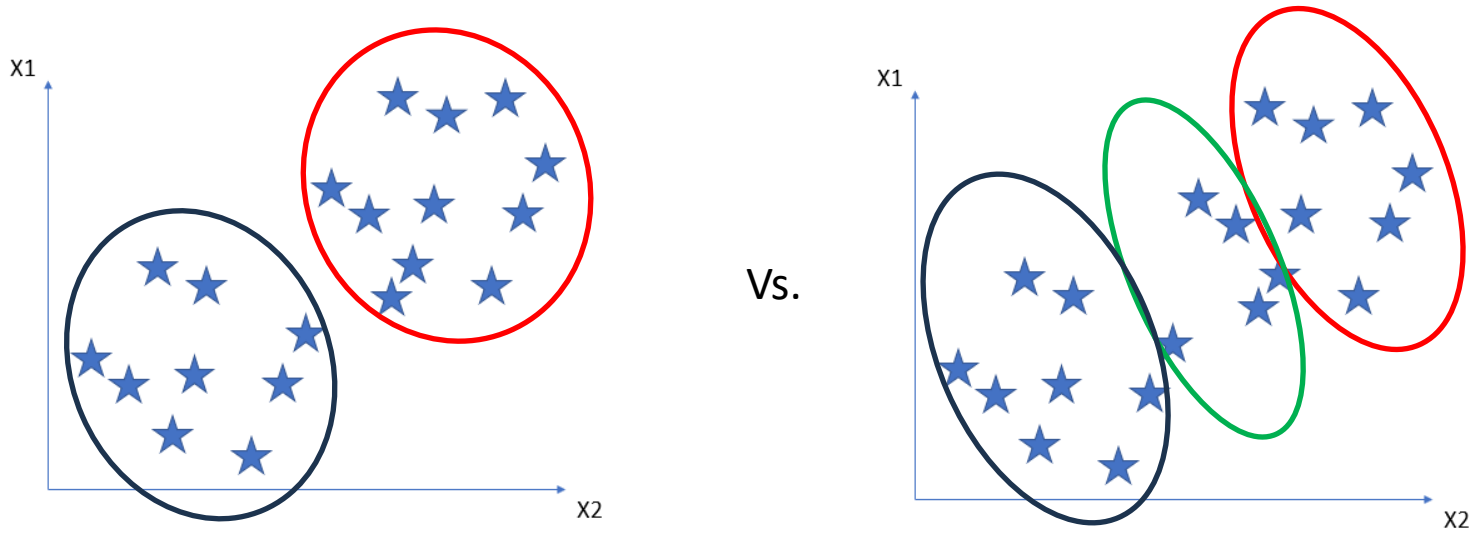
Método del codo: Consideraciones



- Para evitar esto, se elige el valor de K a partir del cual “no hay una reducción significativa de la Inercia”.
- Así, se tienen clusters representativos, con un buen valor de Inercia.

Clustering: Intuición inicial (2)

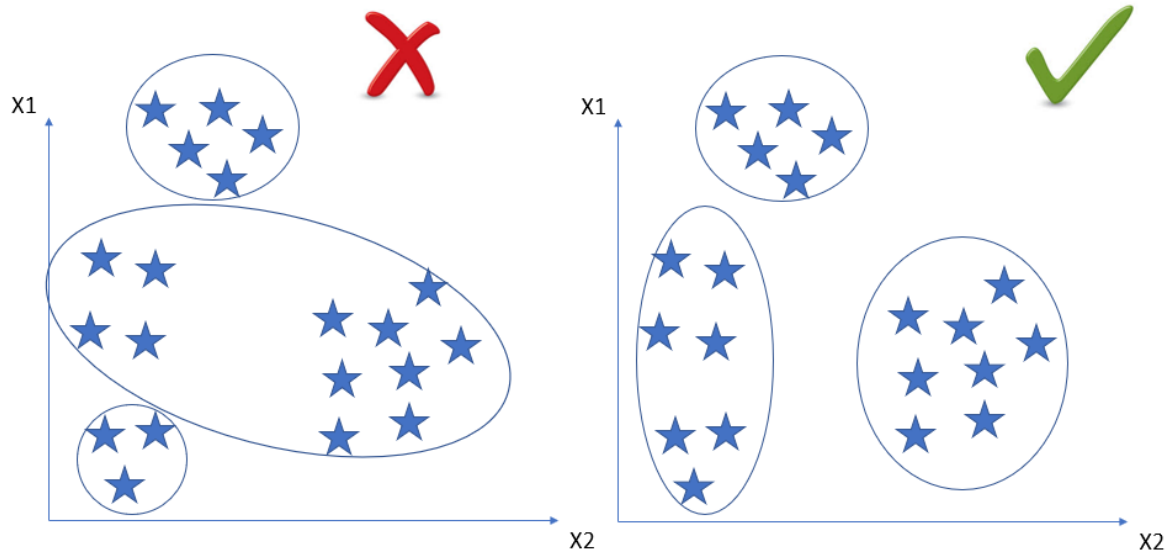
- No todo es heterogeneidad



- Además de buscar que los datos en los clusters sean parecidos entre sí (**poca heterogeneidad**) también interesa que sean distintos a los de otros clusters (**alta separabilidad**).
- En el ejemplo previo, 3 clusters pueden dar menos Inercia, pero peores resultados que usar 2.

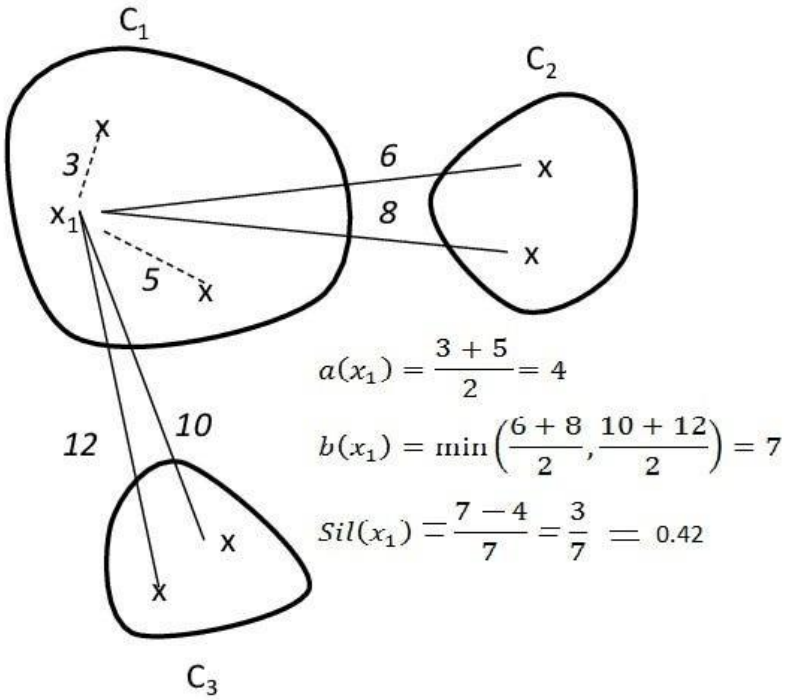
Heterogeneidad y separación de los clusters

- Visión combinada: poca heterogeneidad y alta separabilidad.



Selección de K: Coeficiente de la Silueta

- Coeficiente de la Silueta como forma de **cuantificar la separación** entre clusters y la **heterogeneidad** de los clusters.

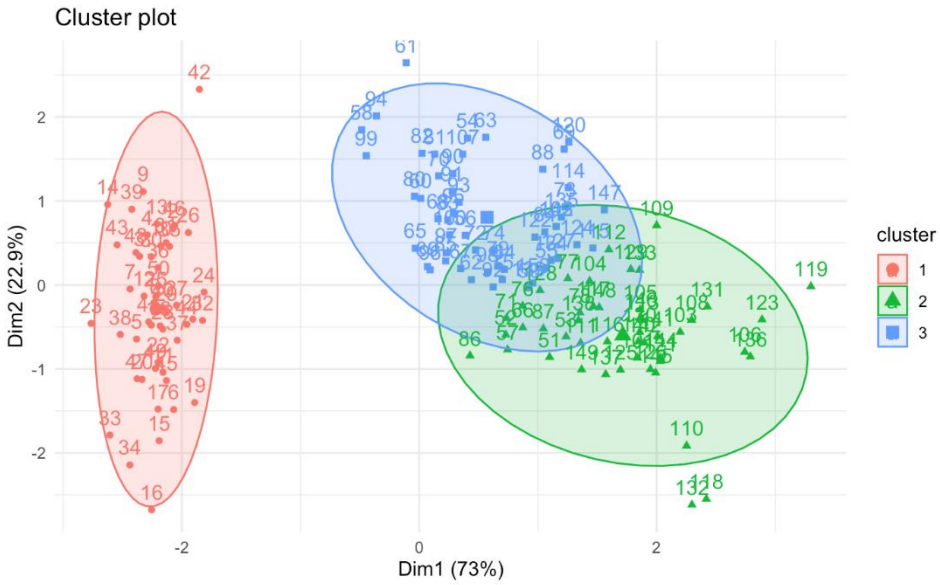


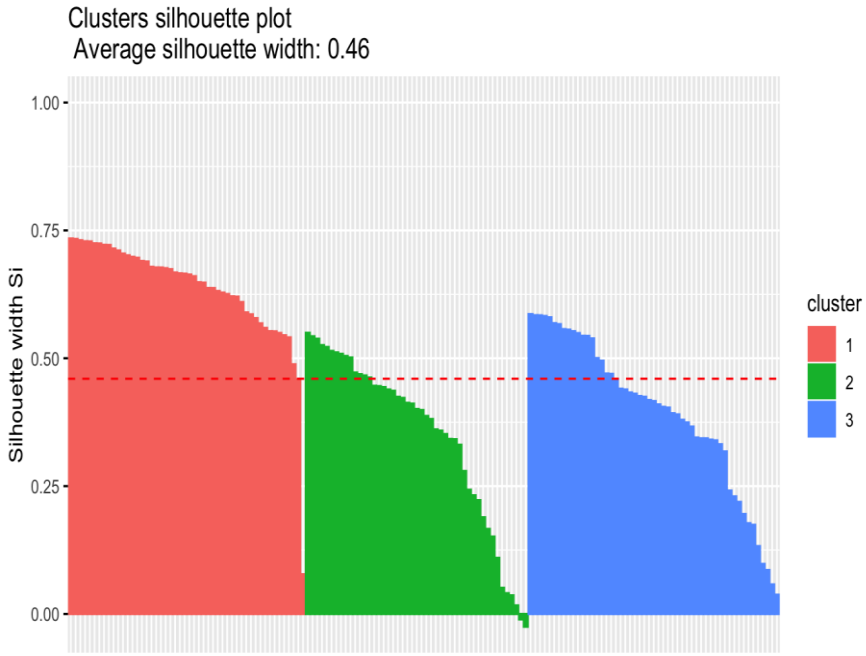
https://medium.com/@MrBam44/how-to-evaluate-the-performance-of-clustering-algorithms-3ba29cad8c03

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

- $a(i)$: Distancia promedio del punto i a todos los demás puntos de su propio clúster [**Heterogeneidad**]
- b(i): Es la distancia promedio del punto i a los puntos de otro clúster (no el suyo) más cercano [**Separación**]
- Se tendrá un valor entre -1 (el peor caso) y +1 (el mejor).
- Después, se calcula el **coeficiente de la silueta promedio** para todos los puntos.

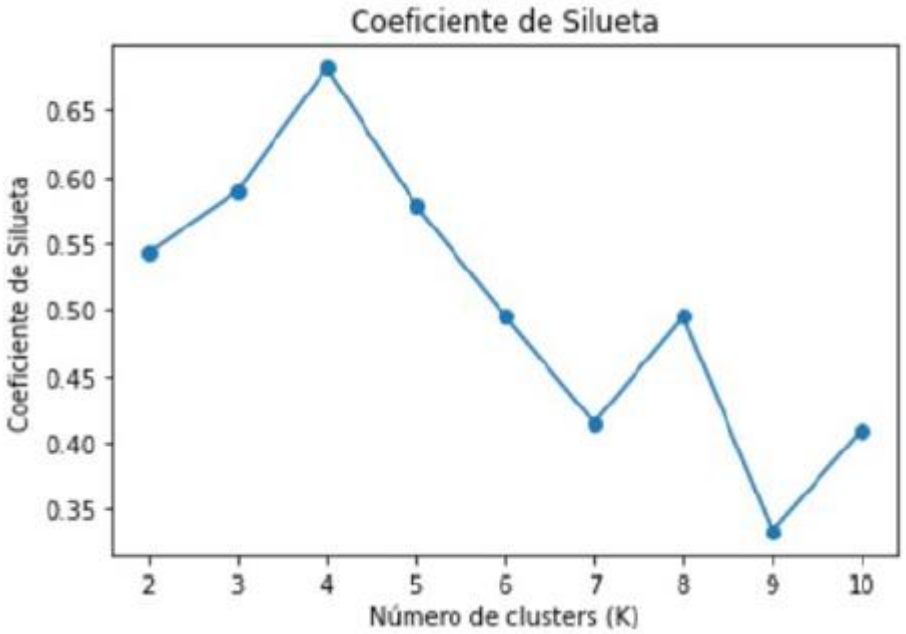
Selección de K: Coeficiente de la Silueta





https://rpkgs.datanovia.com/factoextra/reference/fviz_silhouette.html

Selección de K: Coeficiente de la Silueta



- Análogamente, se puede usar el coeficiente de la Silueta para determinar el número óptimo de clusters.
- En este caso, directamente, corresponde a K=4 ya que es el caso de mayor valor del coeficiente.

K-Means: Ventajas y desventajas

Ventajas:

- Si se define un número máximo de iteraciones, el algoritmo es bastante eficiente computacionalmente: O(t*k*n*d) con t el número de iteraciones, k el número de clústers, n el número de registros y d la dimensionalidad de las features.

Desventajas:

- Puede quedar atrapado en **mínimos locales**, lo que podría llevar a soluciones subóptimas, especialmente **influenciado por la posición inicial de los centroides**.
- Requiere **conocer el valor de k** (número de clústeres) de antemano.
- Es sensible a los valores **atípicos**, dado que su funcionamiento se basa en cálculos de distancias.
- Existe la posibilidad de que un **clúster quede vacío** si alguno de los centroides termina sin puntos asignados.
- Basado en distancias euclídeas -> sólo válido para **variables numéricas**

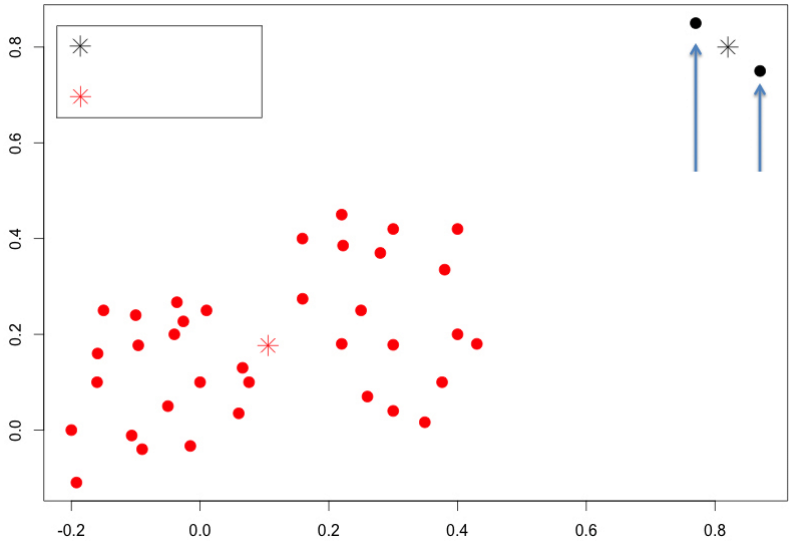
K-Means: Desventajas

Sensibles a distancias

Cliente	Edad	Ingresos (USD)	Frecuencia de Compra (compras/mes)	Total Gastado (USD)
1	25	3000	2	500
2	40	7000	5	3000

$$d(cliente_2, cliente_1) = \sqrt{(40-25)^2 + (7000-3000)^2 + (5-2)^2 + (3000-500)^2} \approx 4717$$

Sensibles a outliers



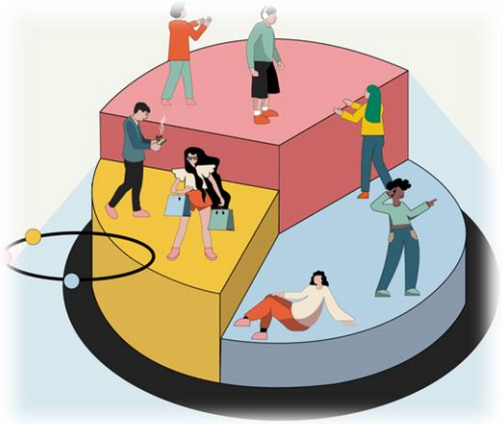
https://www.researchgate.net/figure/Outliers-effect-k-means-clustering-left-vs-k-medoids-clustering-right_fig14_309762404

Estandarizar/Normalizar datos

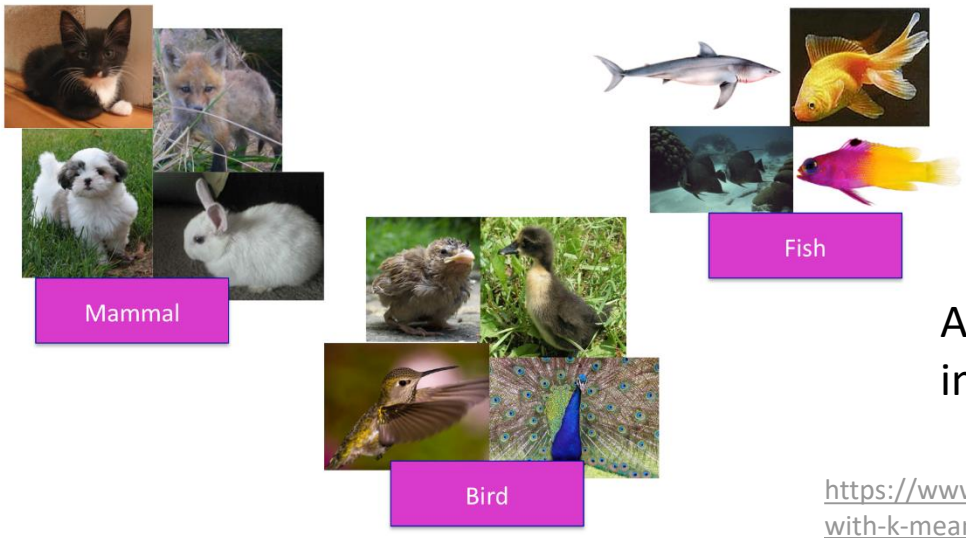
Eliminar outliers

K-Means: Algunas aplicaciones

Segmentación de clientes



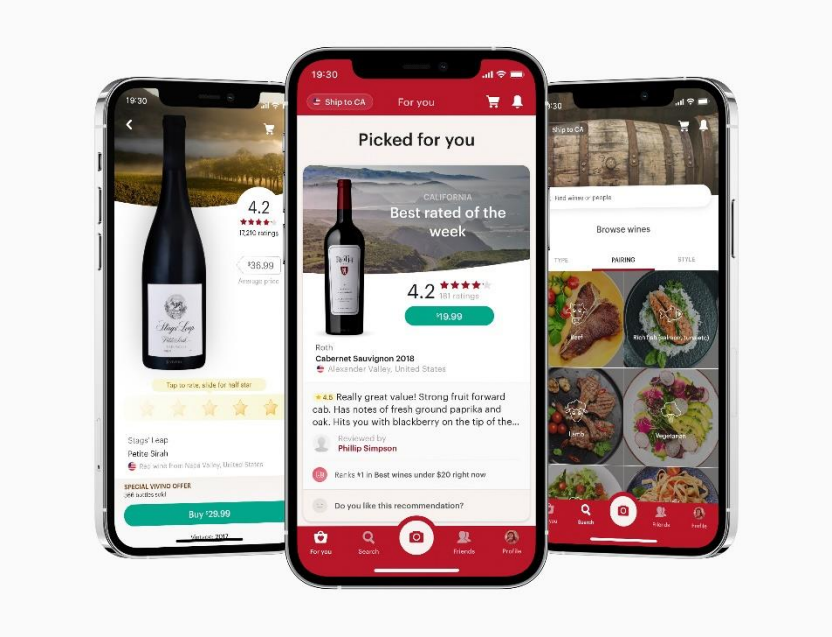
Segmentación de productos



Agrupación de imagenes similares

https://www.kaggle.com/code/mesofianeyou/customer-segmentation-with-k-means
https://genomicsclass.github.io/book/pages/distance.html

Demo: K-Means



tema 2 - ClusteringKMeans.ipynb

https://www.vivino.com/app

En resumen

- **Clustering** como aproximación para agrupar de manera no supervisada datos con patrones similares
- Agrupaciones basadas en **distancias**: sensible a **magnitudes** y **outliers** al usar distancias Euclideas
- Mejor clustering si **menor heterogeneidad** y **mayor separabilidad**
- Implementación de clustering: algoritmo **K-Means**
- Selección de **hiperparámetros** en clustering: Método de la silueta y del codo

En la próxima semana

¿Cómo se comparan y qué diferencias hay en las diversas implementaciones de K-Means y cómo pueden influir en la calidad e interpretación de resultados?

www.unir.net