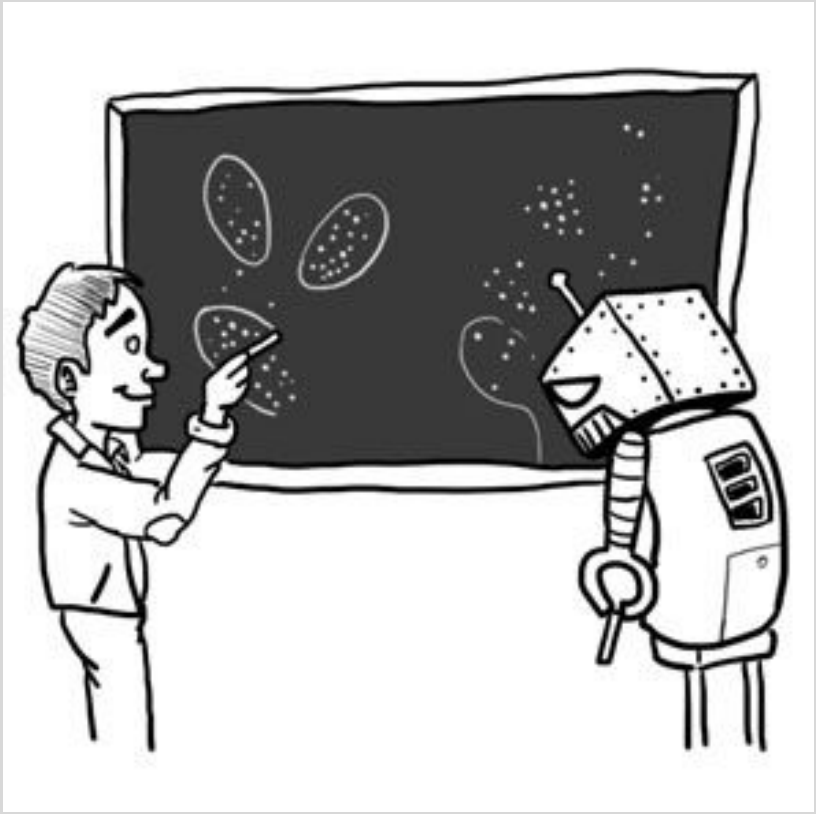


DigitalHouse > Coding School

DATA SCIENCE

Introducción al proceso de Clustering

OBJETIVOS DE LA CLASE

1

SOBRE LOS DATOS

Analizar formato y preprocesamiento para aplicar algoritmos de clusterización

2

SOBRE ALGORITMOS

Conocer los algoritmos más usados. Realizar un análisis de cluster con K-Means

3

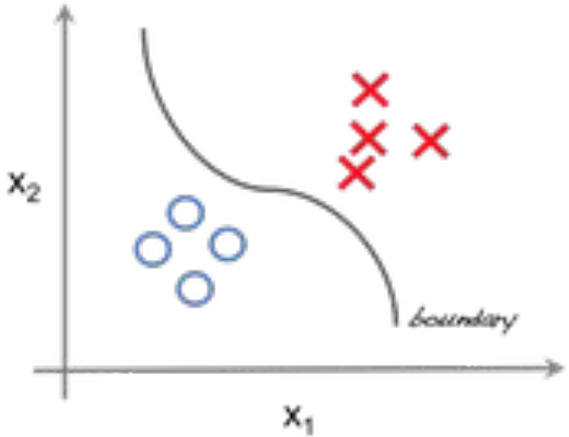
SOBRE EVALUACIÓN DE AJUSTE

Conocer conceptos de evaluación para el ajuste de clusterización

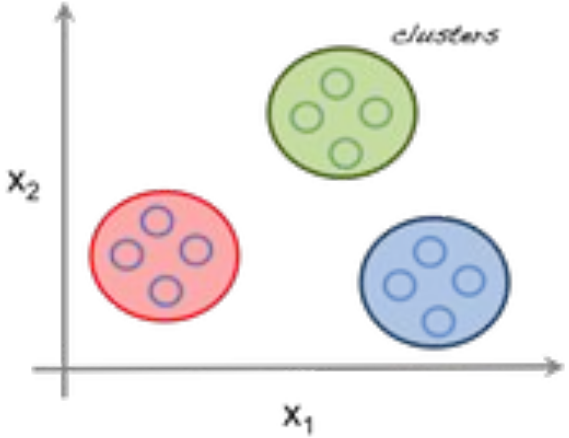
Aprendizaje No Supervisado

¿Cuál es la principal diferencia entre los problemas de **aprendizaje supervisado** y **aprendizaje no supervisado**?

Aprendizaje Supervisado

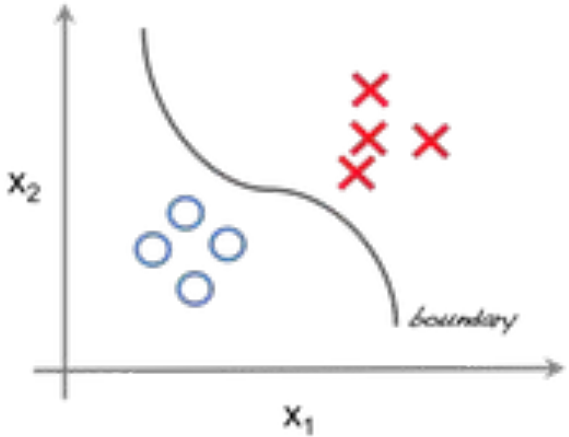


Aprendizaje no Supervisado

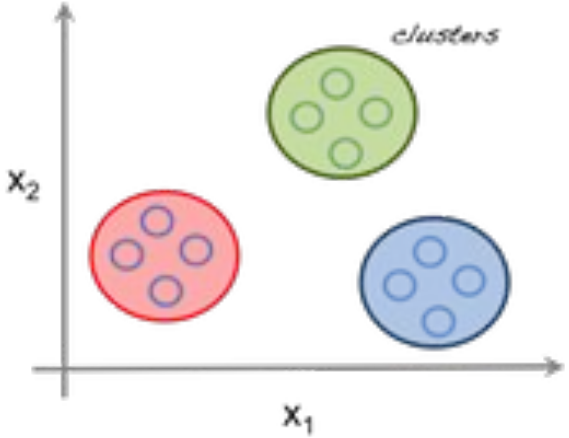


Aprendizaje No Supervisado: solamente disponemos de la matriz de predictores **X** → **No hay una variable target**.

Aprendizaje Supervisado



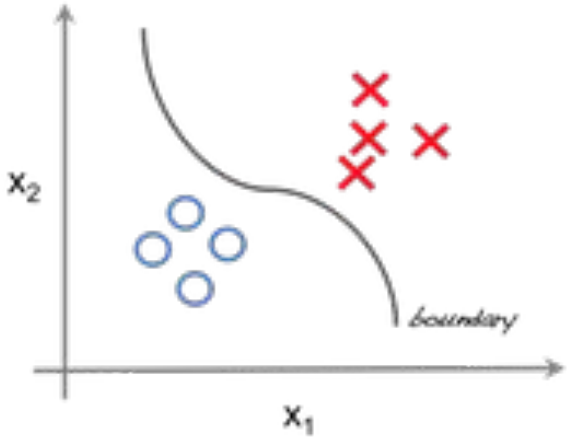
Aprendizaje no Supervisado



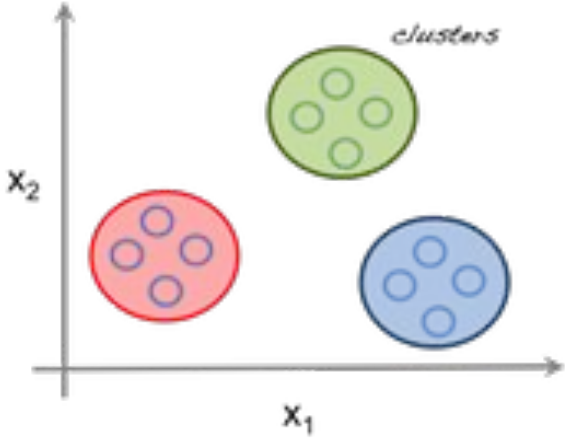
El objetivo en un problema de **Aprendizaje No Supervisado** es descubrir alguna **estructura-patrón interesante en los datos**.

- ¿Es posible visualizar los datos de manera informativa?
- ¿Se pueden identificar grupos de casos o variables similares?

Aprendizaje Supervisado



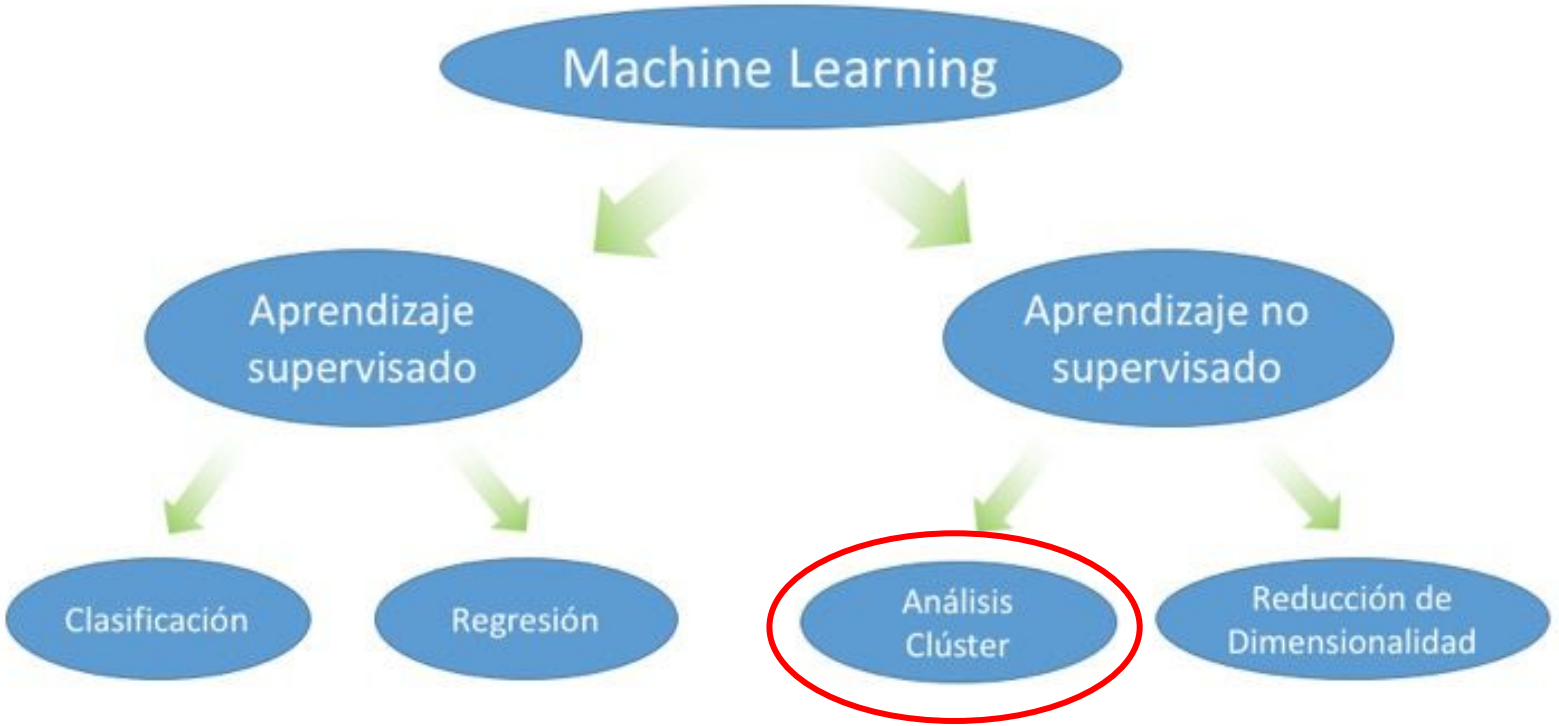
Aprendizaje no Supervisado



- El **Aprendizaje No Supervisado** tiene un carácter un poco más subjetivo que el Supervisado, dado que no hay un objetivo "simple" en el análisis. No hay una variable target contra la cual evaluar los resultados.

- No obstante, la utilización de técnicas de Aprendizaje No Supervisado está creciendo bastante en varios campos:
  - identificación de **grupos de pacientes** con cáncer, agrupados según mediciones de sus expresiones genéticas.
  - identificación de **grupos de compradores** caracterizados por sus historias de compra previas
  - identificación de **grupos de películas** de acuerdo a las calificaciones asignadas por los espectadores

Introducción al proceso de clustering

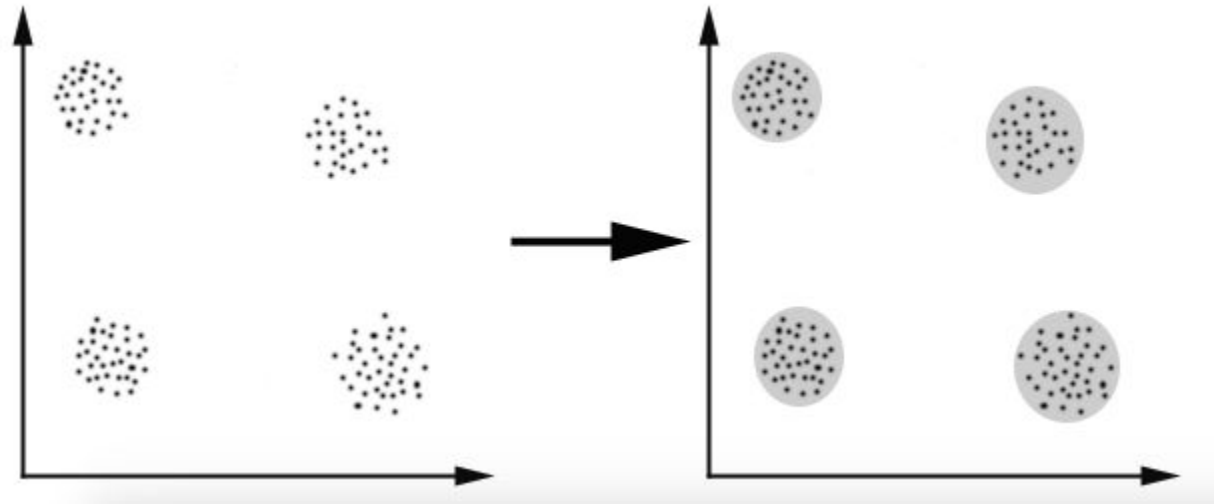

Machine Learning
Aprendizaje supervisado
Aprendizaje no supervisado
Clasificación
Regresión
Análisis Clúster
Reducción de Dimensionalidad

- El proceso de **Clusterización** o **Clustering** es uno de los métodos más usados para comprender cierta estructura en un conjunto de datos. Probablemente sea uno de los métodos de aprendizaje no supervisado más importantes.

- Una definición informal de Clustering podría ser "**el proceso de organizar los objetos en grupos cuyos miembros son similares de alguna manera**".

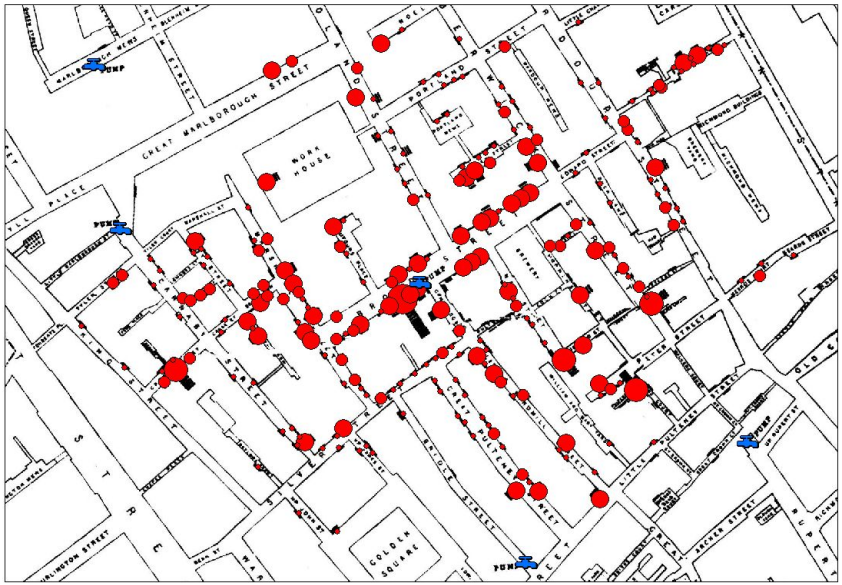
- Un **cluster** es por lo tanto una colección de objetos que son "similares" entre ellos y son "disímiles" a los objetos pertenecientes a otros clusters.

El criterio de similitud es una medida de proximidad: dos o más objetos pertenecen al mismo grupo si son "similares", si la medida de similaridad es grande entonces las observaciones son similares, o si la medida de no similaridad (o distancia) es pequeña.



Un poco de historia...

- **John Snow**, un médico londinense, graficó en un mapa la ubicación de las **muertes por cólera durante una epidemia** ocurrida en la década de **1850**.
- Las ubicaciones indican que **los casos se *clusterizaban* alrededor de ciertas esquinas** donde había **bombas de agua contaminadas**, exponiendo así el problema y la solución.



Posibles aplicaciones:

- **Marketing:** encontrar **grupos de clientes** con un comportamiento similar dado una gran base de datos de clientes que contienen sus propiedades y registros de compras.
- **Biología: clasificación de plantas y animales** dados sus características.
- **Seguros:** identificación de **grupos de titulares de pólizas de seguros** de automóviles con un costo alto en promedio en reclamos.
- **Urbanismo:** identificación de **grupos de casas según su tipo de vivienda, valor y ubicación** geográfica.
- **Estudios de terremotos:** agrupar los epicentros de terremotos observados para identificar **zonas peligrosas**.
- **WWW:** clasificación de documentos; Agrupación de datos de weblog para descubrir **grupos de patrones de acceso similares**.

¿Cuál es la diferencia entre Clasificación y Clustering?

Si sólo estamos creando grupos, ¿no son los dos el mismo proceso?

Existe una importante **distinción entre clasificación y agrupación**: En la **clasificación**, estamos agrupando los datos de acuerdo con un **conjunto de grupos predefinidos**. *(Supervisado)*

"Sabemos cuáles son las características de un mamífero, y los humanos tienen las características de ese grupo predefinido."

En la agrupación (o **clustering**), sin embargo, nos propusimos averiguar si los puntos de nuestro conjunto de datos tienen relaciones entre sí, y **agrupamos aquellos con características similares en un grupo**. En otras palabras, **tenemos que descubrir las propias clases**. *(No Supervisado)*

Propiedades deseables de los algoritmos de clustering:

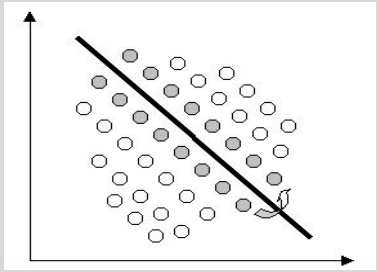
- Escalabilidad
- Tratar con diferentes tipos de atributos
- Capacidad de descubrir clusters con forma arbitraria
- Capacidad para hacer frente al ruido y los valores extremos
- Insensibilidad al orden de los registros de entrada
- Alta dimensionalidad
- Interpretabilidad y usabilidad

Hay una serie de problemas en Clustering. Entre ellos:

- Las técnicas actuales de Clustering no abordan todos los requisitos adecuadamente (y simultáneamente).
- Tratar con un **gran número de dimensiones** y un gran número de elementos de datos puede ser **problemático debido a la complejidad del tiempo de cómputo**.
- La eficacia del método depende de la **definición de "distancia"** (para el agrupamiento basado en la distancia).
- Si no existe una medida obvia de distancia, debemos "definirla", lo que no siempre es fácil, especialmente en espacios multidimensionales.
- El resultado del algoritmo de clustering (que en muchos casos puede ser arbitrario) puede interpretarse de diferentes maneras.

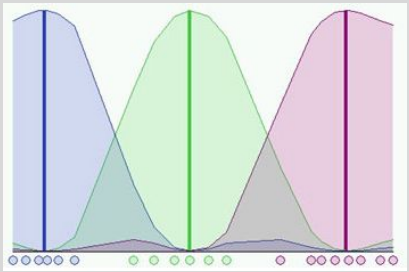
Los algoritmos de Clustering pueden clasificarse como:

Si un determinado dato pertenece a un grupo definido, entonces no puede ser incluido en otro.



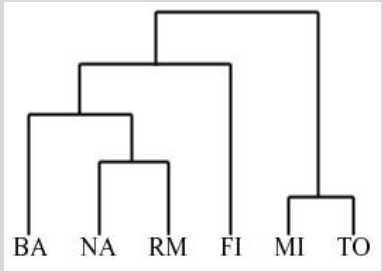
Clustering exclusivo (ej: K-means)

Utilizan conjuntos difusos para agrupar datos, de modo que cada punto puede pertenecer a dos o más grupos con diferentes grados de pertenencia.



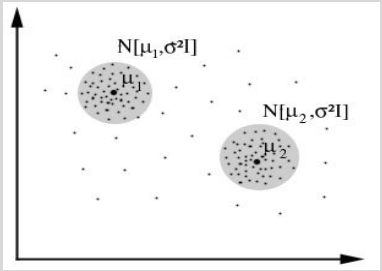
Clustering solapado (ej: Fuzzy C-means)

Se basan en la unión entre los dos clusters más cercanos. La condición inicial se realiza estableciendo cada dato como un cluster. Después de algunas iteraciones llega a los clusters finales deseados.



Clustering jerárquico

Utilizan un enfoque completamente probabilístico.



Clustering probabilístico (ej: Mixture of Gaussians)

Métricas de distancia

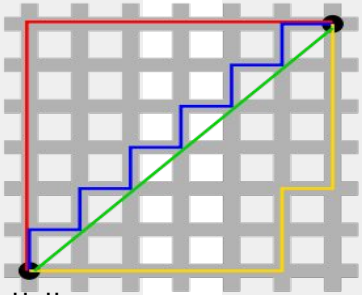
Una medida muy popular es la distancia **de Minkowski**:

$$d_p(x_i, x_j) = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}$$

donde *d* es la dimensionalidad de los datos y p es un entero positivo . Algunos casos particulares son:

Si p=1 obtenemos la **distancia de Manhattan:**

$$d_1(x_i, x_j) = \sum_{k=1}^{d} |x_{ik} - x_{jk}|$$



Es una aproximación a la distancia euclidiana de menor costo computacional

Si p=2 obtenemos la **distancia euclídea:**

$$d_2(x_i, x_j) = \sqrt{\sum_{k=1}^{d} |x_{ik} - x_{jk}|^2}$$

¡Esta distancia es invariante a traslaciones!

ALGORITMO K-MEANS

(MacQueen,1967)

- El procedimiento sigue una manera sencilla y fácil de clasificar un determinado conjunto de datos a través de un cierto **número de clusters** (suponga k clusters) **fijado a priori**. Es un algoritmo del tipo particional.

- El **usuario determina el número de clusters k**. Cada cluster tiene un centro llamado centroide.

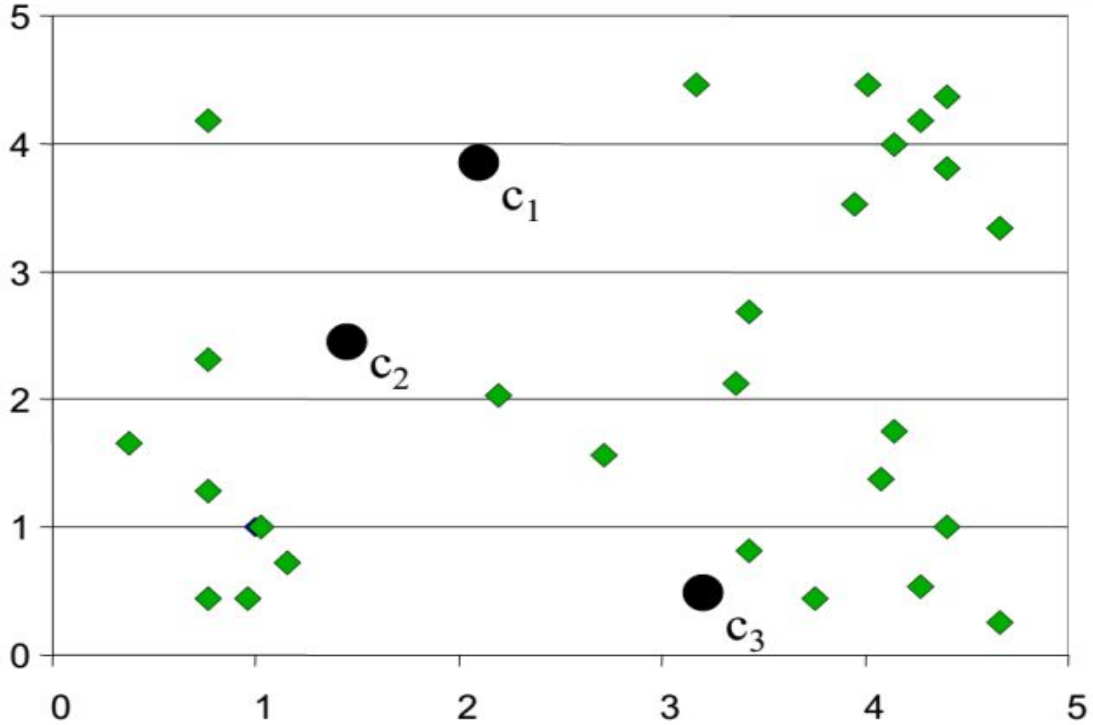
    - Paso 1: *Se eligen al azar K puntos para ser los centroides iniciales.*

    - Paso 2: *Se forman K cluster asociando los puntos a los centroides más cercanos. Las particiones aquí representan el diagrama de Voronoi generado por los centroides.*

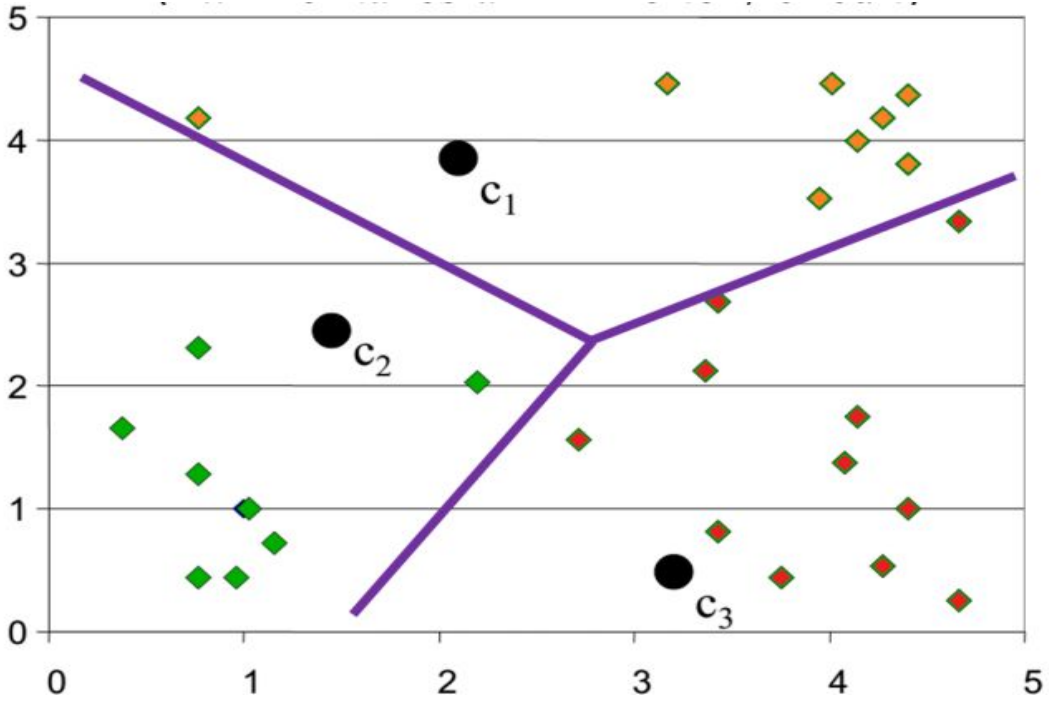
    - Paso 3: *Se recalcula el centroide de cada uno de los K clusters usando la asignación actual de puntos a cada cluster.*

    - Paso 4: *Se repite el paso 2 y 3 hasta que se cumple algún criterio de convergencia*

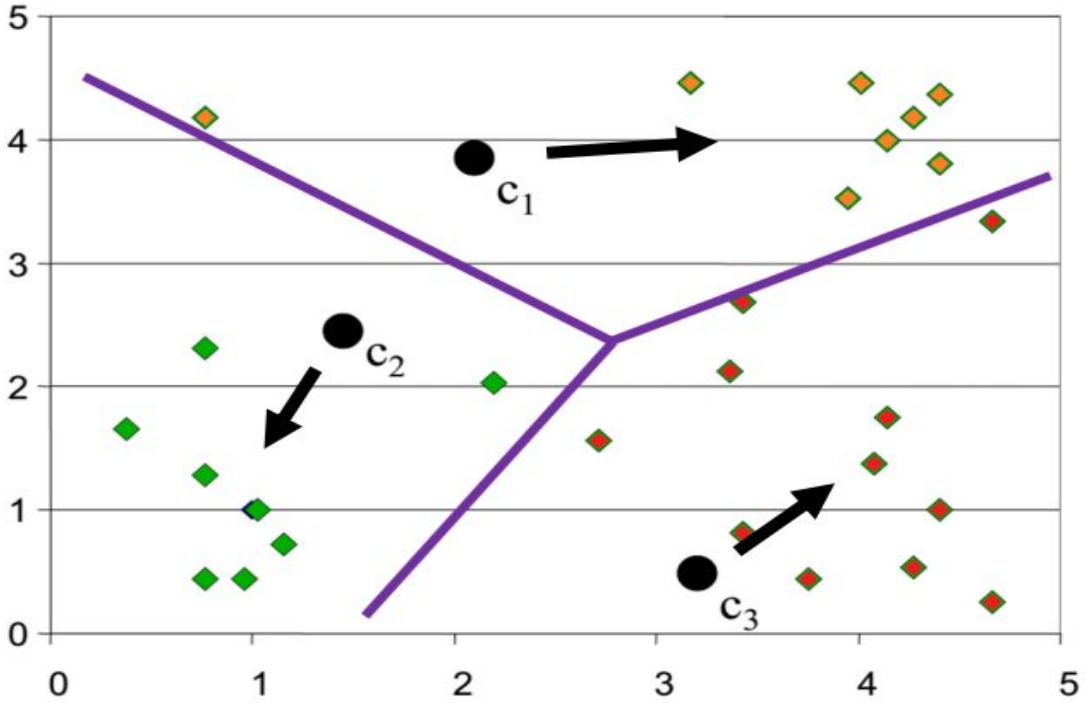
Paso 1: *Se eligen al azar K puntos para ser los centroides iniciales.*



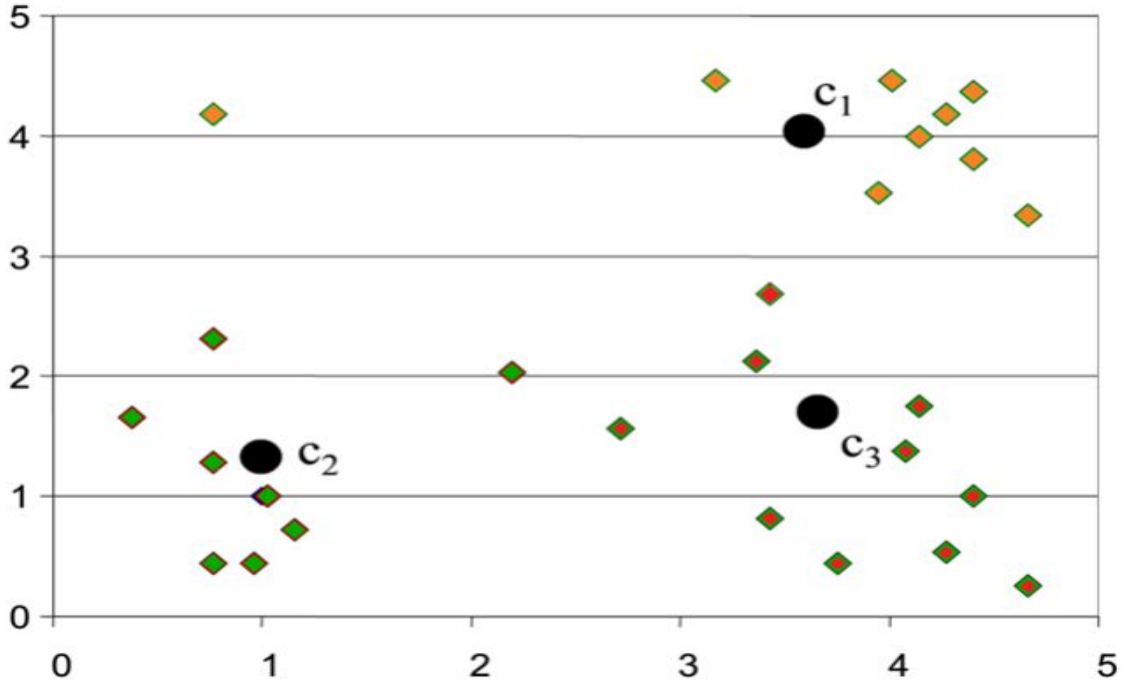
Paso 2: *Se forman K cluster asociando los puntos a los centroides más cercanos. Las particiones aquí representan el diagrama de Voronoi generado por los centroides.*



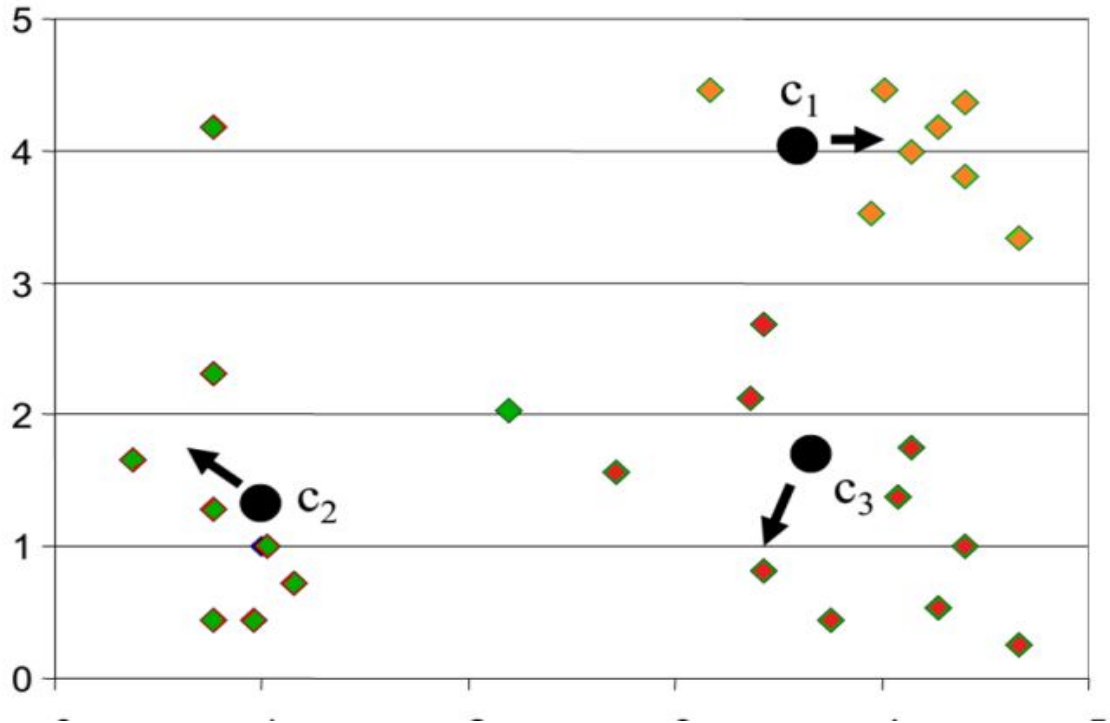
Paso 3: *Se recalcula el centroide de cada uno de los K clusters usando la asignación actual de puntos a cada cluster.*



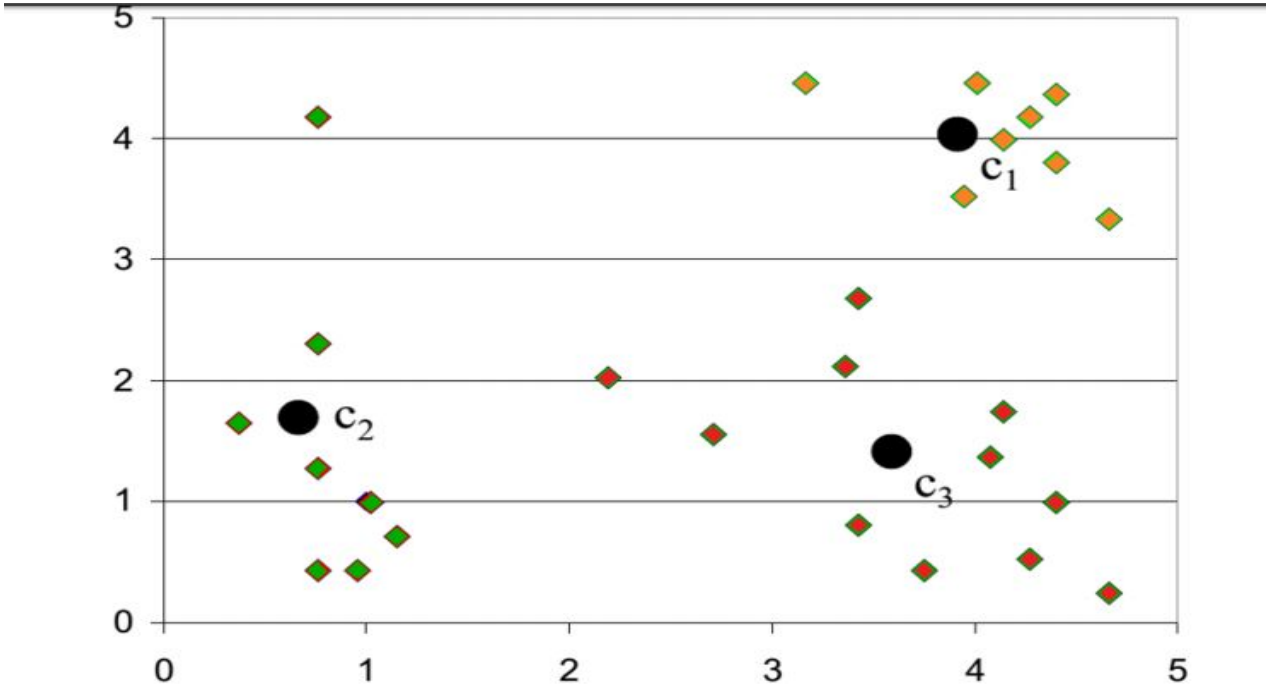
Resultado de la primer iteración.



Paso 4: *Se repite el paso 2 y 3 hasta que se cumple algún criterio de convergencia.*



Paso 4: *Se repite el paso 2 y 3 hasta que se cumple algún criterio de convergencia.*



Función Objetivo de K-Means

Este algoritmo tiene como objetivo **minimizar una función objetivo**, en este caso una función de error cuadrático. La función objetivo:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

donde $\left\| x_i^{(j)} - c_j \right\|^2$ es la distancia del *punto-i* al centroide del *cluster-j.*

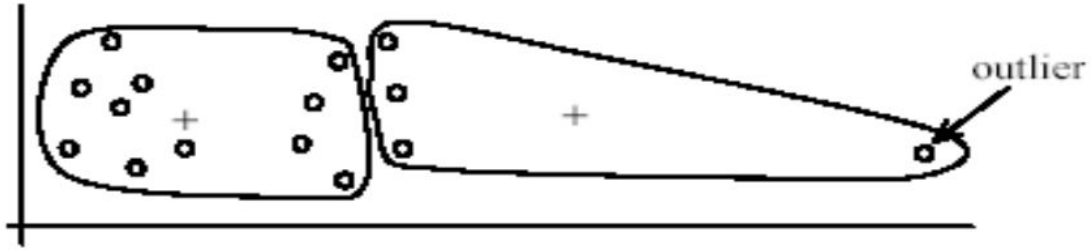
¿Por qué usar K-means?

- **Simplicidad**: fácil de entender y de implementar.
- **Eficiencia**: la complejidad tiempo de cómputo es O(tkn) donde n es el tamaño de la muestra , k el número de clusters y t es el número de iteraciones.
- Como tanto k como t son pequeñas k-means es considerado un algoritmo lineal.

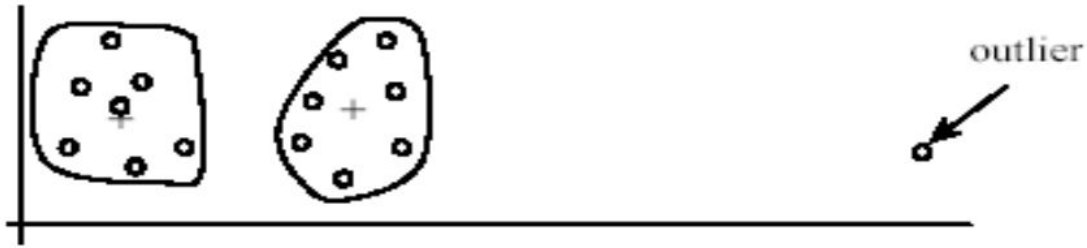
Algunos **problemas del algoritmo K-Means**:

- Dado que comienza con un asignación de clusters aleatoria puede suceder que los **resultados varíen en diferentes corridas.**
- Es necesario **“tunear” el parámetro K** (cantidad de clusters).
- Por las características del algoritmo, el mínimo global puede no ser logrado. Termina en un óptimo local si se usa SSE. El óptimo global es difícil de encontrar debido a la complejidad del problema.
- El algoritmo **sólo es aplicable si las medias están definidas** (para datos categóricos, k-mode el centroide se representa por los valores más frecuentes).

El algoritmo es **sensible a valores atípicos**.



(A): Undesirable clusters

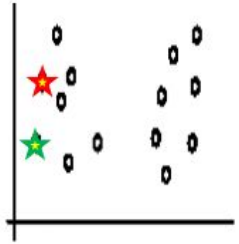


(B): Ideal clusters

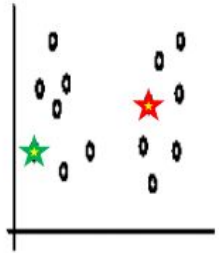
- Una forma de trabajar con los **outliers** es **eliminar algunos puntos** de datos que están mucho más lejos de los centroides que otros puntos de datos.

- Otra posibilidad es realizar un **muestreo aleatorio**: eligiendo un pequeño subconjunto de los puntos de datos, la posibilidad de seleccionar un valor atípico es mucho más pequeña.

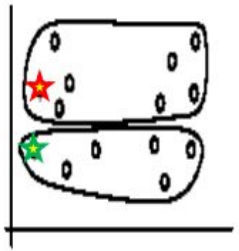
Sensibilidad a las condiciones iniciales aleatorias: dado que comienza con un asignación de clusters aleatoria puede suceder que los **resultados varíen en diferentes corridas.**



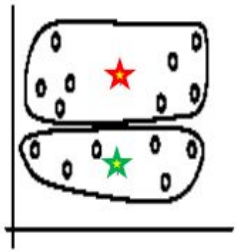
Random selection of seeds (centroids)



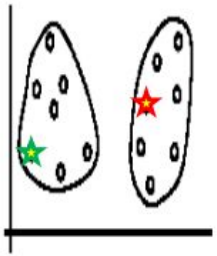
Random selection of seeds (centroids)



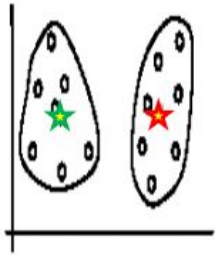
Iteration 1



Iteration 2

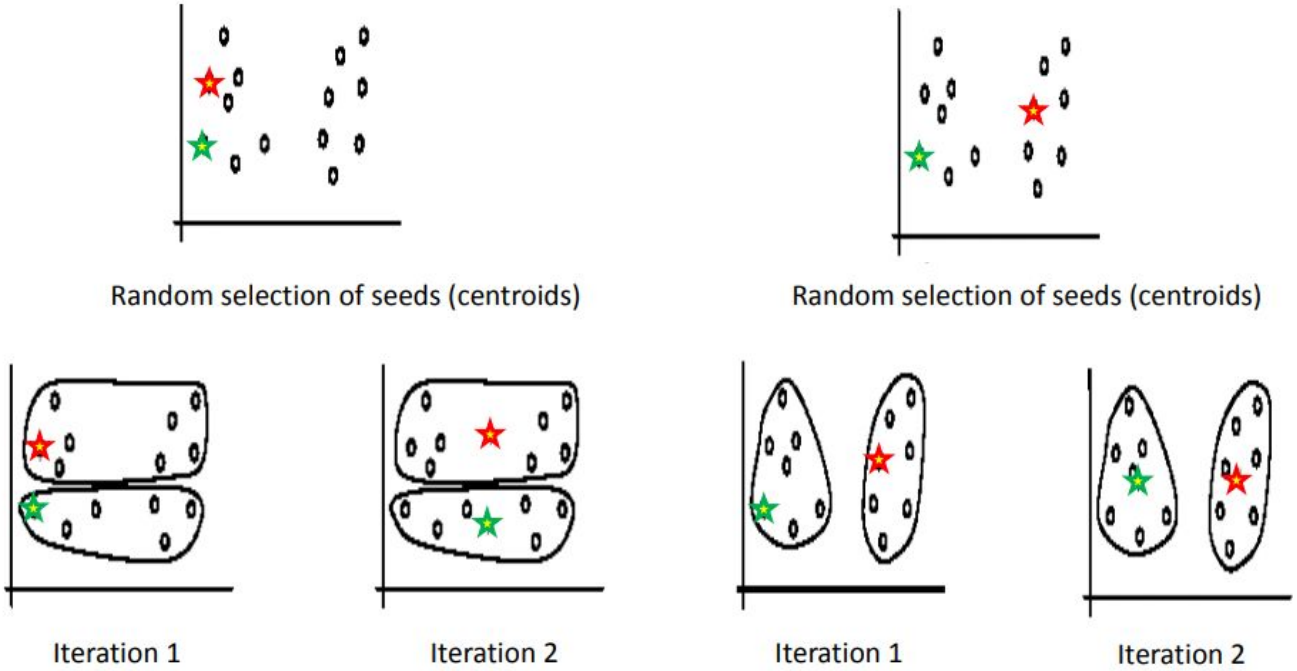


Iteration 1



Iteration 2

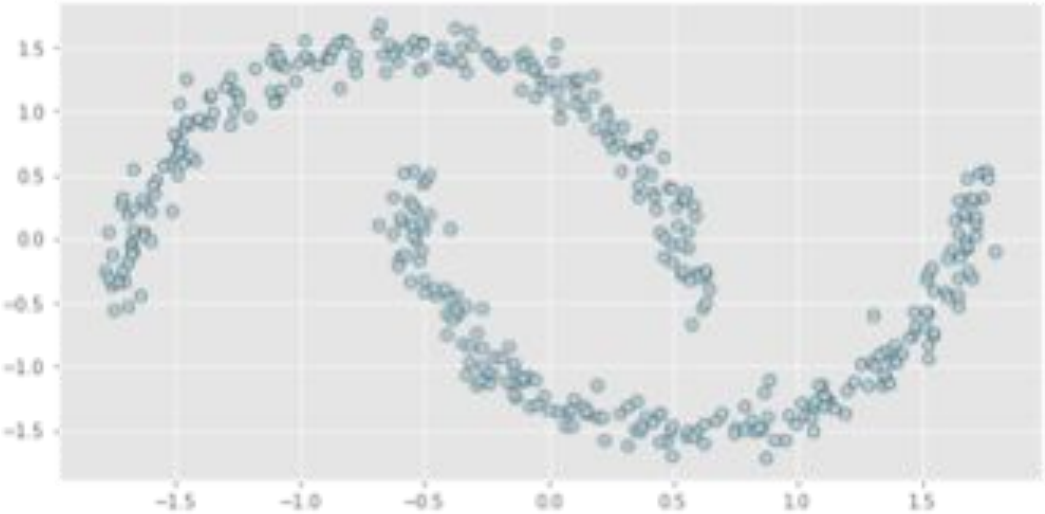
Sensibilidad a las condiciones iniciales aleatorias: dado que comienza con un asignación de clusters aleatoria puede suceder que los **resultados varíen en diferentes corridas.**



¡Solución: correr el algoritmo muchas veces con diferentes inicializaciones y quedarse con la agrupación más frecuente!

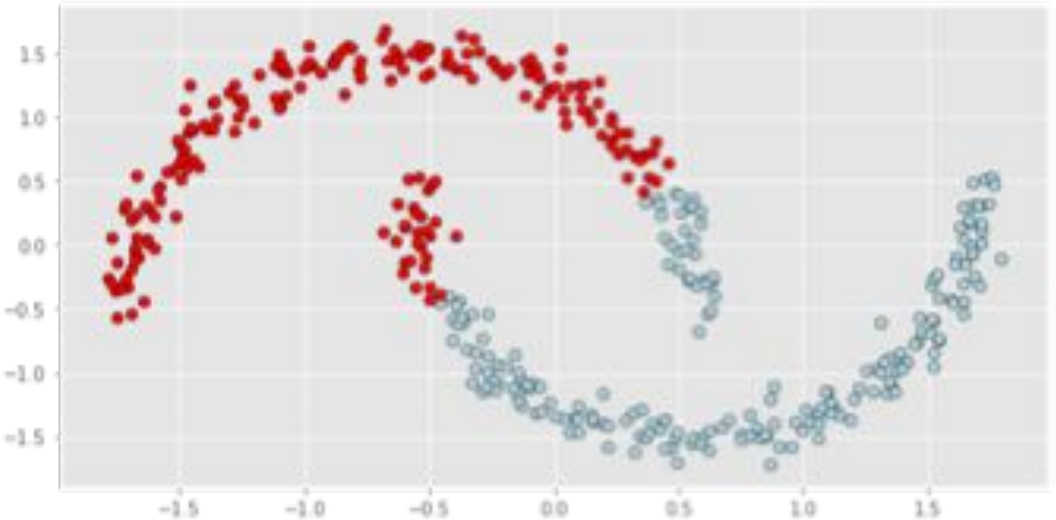
El algoritmo k-means **asume que los datos están distribuidos en formas convexas**, como híper-elipsoides o híper-esferas.

¿Qué pasa cuando los datos están ordenados de formas más complejas?



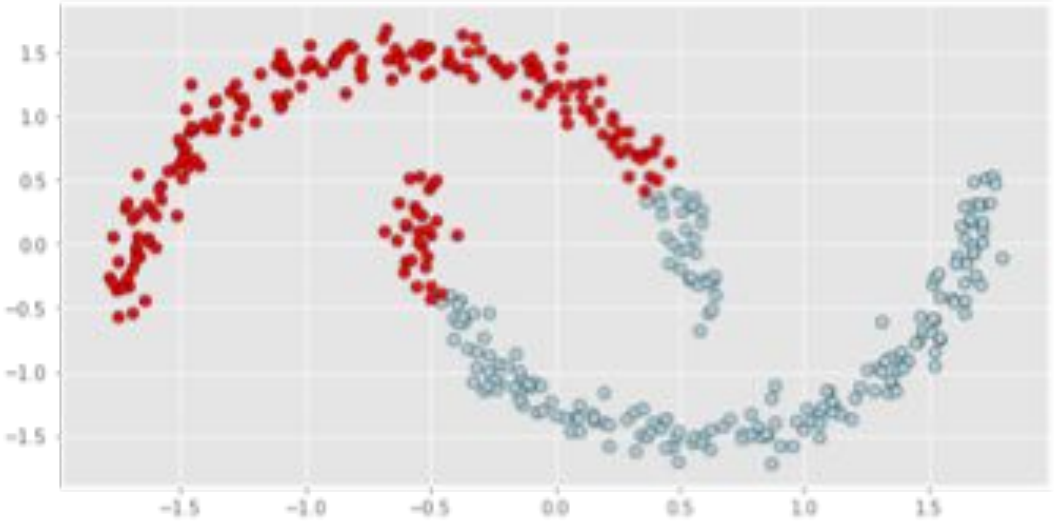
El algoritmo k-means **asume que los datos están distribuidos en formas convexas**, como híper-elipsoides o híper-esferas.

¿Qué pasa cuando los datos están ordenados de formas más complejas?



El algoritmo k-means **asume que los datos están distribuidos en formas convexas**, como híper-elipsoides o híper-esferas.

¿Qué pasa cuando los datos están ordenados de formas más complejas?



Para este tipo de estructuras es conveniente usar **DBSCAN**, un **algoritmo de clustering basado en densidad**.

En resumen...

- A pesar de las debilidades, k-means sigue siendo el algoritmo más **popular debido a su simplicidad y eficiencia**.
- No hay evidencia clara de que cualquier otro agrupamiento algoritmo tenga un mejor rendimiento en general.
- Comparar diferentes algoritmos de agrupamiento es una tarea difícil. ¡**Nadie sabe el clustering correcto**!

VARIABLES CATEGÓRICAS
ALGORITMO K-MODES
(Huang, 1999)

- **K-means funciona muy bien para variables cuantitativas**. ¿Pero qué pasa si queremos clusterizar **variables categóricas**?

- Una opción directa sería clusterizar en base a las variables dummies para cada categoría de las variables originales y luego utilizar k-means.
  - Problema: para datasets de alta dimensionalidad este enfoque se hace impracticable

- Una opción es el **algoritmo K-modes**: este enfoque reemplaza las métricas de distancia euclidiana por una medida de "disimilaridad" y **usa las modas de cada variable para representar los centros de los clusters**.

La distancia es definida de la siguiente forma:

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \qquad \delta(x_j, y_j) = \begin{cases} 0, & x_j = y_j \\ 1, & x_j \neq y_j . \end{cases}$$

Es decir, **se computa la cantidad de "diferencias"** o "no coincidencias" que existen entre dos elementos Xi e Yi.

En el caso específico del algoritmo k-modes, las distancias se calcularán para cada individuo que presenta un vector de categorías de cada una de las variables y el vector de centroides de un cluster.

Algoritmo:

- Paso 1. Se seleccionan aleatoriamente k casos únicos como los centros de los clusters (modas).
- Paso 2. Se calculan las distancias entre cada objeto y entre los centros. El objeto es asignado al centro con menor distancia. Se repite este proceso hasta que todos los objetos han sido asignados.
- Se selecciona una nueva moda para cada cluster y se compara con la moda anterior. Si es diferente se vuelve al paso 2. Si es igual, se termina el proceso.

El proceso de clustering minimiza la siguiente función

$$F(U,Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} u_{i,l} d(x_{i,j}, z_{l,j})$$

donde U es la matriz de partición (que indica para cada observación del dataset a qué cluster pertenece). Z es el set de vectores de modas.

Ejemplo: dataset de secuencias genotípicas

	Locus									
Indiv	1	2	3	4	5	6	7	8	9	10
1	BB	AB	AB	AB	AB	AA	AB	BB	AB	BB
2	AB	BB	BB	AB	BB	BB	AB	AB	BB	AB
3	BB	BB	AA	AA	AB	AB	AA	AB	BB	AB
4	AB	BB	AB	AB	AB	AB	BB	AB	AA	AB
5	BB	AB	AA	AB	AA	AB	AA	AB	AA	BB
6	BB	AB	AB	AB	BB	BB	AB	AA	AB	AB
7	BB	BB	BB	BB	AB	AB	AA	AB	BB	AB
8	AB	BB	AB	AB	AA	AA	AB	BB	AB	BB
9	BB	AA	AB	AB	BB	AB	AB	AA	AB	AB
10	AB	BB	AB	BB	AB	AB	BB	AB	AB	AA
11	AA	BB	AA	AA	AA	AB	AA	AB	AB	AB
12	BB	AB	BB	BB	AB	BB	AB	BB	AA	AB
13	AB	BB	AB	AA	AB	AB	BB	AB	AA	AA
14	BB	AA	AB	AB	BB	BB	AB	AA	AB	AB
15	AB	BB	BB	BB	AB	AA	AB	BB	AB	AA

- Dataset de secuencias genotípicas.
  - Cada fila es un individuo.
  - Cada columna es un secuencia.

- Objetivo: encontrar grupos de individuos con secuencias genotípicas similares => **K-modes**.

Ejemplo: dataset de secuencias genotípicas

(a)

	Locus									
Cluster	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1 (1)	BB	AB	AB	AB	AB	AA	AB	BB	AB	BB
2 (5)	BB	AB	AA	AB	AA	AB	AA	AB	AA	BB
3 (12)	BB	AB	BB	BB	AB	BB	AB	BB	AA	AB
4 (15)	AB	BB	BB	BB	AB	AA	AB	BB	AB	AA

- Se seleccionan los individuos 1, 5, 12 y 15 como los centros de los clusters.
- Se calculan las distancias de cada individuo a los 4 clusters como el número de "no coincidencias" respecto al centroide del cluster.
- Se asigna a cada individuo al cluster con menos disimilaridad.

Ejemplo: dataset de secuencias genotípicas

(b)

	Locus										Cluster Distance			
Indiv	1	2	3	4	5	6	7	8	9	10	1	2	3	4
1	BB	AB	AB	AB	AB	AA	AB	BB	AB	BB	0	6	5	5
2	AB	BB	BB	AB	BB	BB	AB	AB	BB	AB	8	8	6	6
3	BB	BB	AA	AA	AB	AB	AA	AB	BB	AB	8	5	7	8
4	AB	BB	AB	AB	AB	AB	BB	AB	AA	AB	7	6	7	7
5	BB	AB	AA	AB	AA	AB	AA	AB	AA	BB	6	0	5	10
6	BB	AB	AB	AB	BB	BB	AB	AA	AB	AB	4	7	5	8
7	BB	BB	BB	BB	AB	AB	AA	AB	BB	AB	8	6	5	6
8	AB	BB	AB	AB	AA	AA	AB	BB	AB	BB	4	7	8	4
9	BB	AA	AB	AB	BB	AB	AB	AA	AB	AB	5	7	8	8
10	AB	BB	AB	BB	AB	AB	BB	AB	AB	AA	7	8	8	4
11	AA	BB	AA	AA	AA	AB	AA	AB	AB	AB	9	5	9	8
12	BB	AB	BB	BB	AB	BB	AB	BB	AA	AB	5	7	0	5
13	AB	BB	AB	AA	AB	AB	BB	AB	AA	AA	8	7	8	6
14	BB	AA	AB	AB	BB	BB	AB	AA	AB	AB	5	7	6	8
15	AB	BB	BB	BB	AB	AA	AB	BB	AB	AA	5	10	5	0

(c)

	Locus									
Cluster	1	2	3	4	5	6	7	8	9	10
1	BB	**AA**	AB	AB	**BB**	AA	AB	**AA**	AB	**AB**
2	BB	**BB**	AA	**AA**	AA	AB	AA	AB	AA	**AB**
3	BB	**BB**	BB	BB	AB	BB	AB	**AB**	**BB**	AB
4	AB	BB	**AB**	BB	AB	**AB**	**BB**	**AB**	AB	AA

- El siguiente paso es actualizar los centroides del clúster en función de las personas ahora asignadas a los clústeres.
- El genotipo modal entre los individuos asignados a un grupo se convierte en el genotipo centroide en ese locus. Los genotipos que cambiaron de la inicialización a la actualización se muestran en negrita.
- Se vuelven a calcular las disimilaridades de cada individuo con los nuevos centroides y se vuelven a asignar.
- Se repite hasta que ningún individuo cambia la pertenencia a su cluster.

(d)

	Locus										Cluster Distance			
Indiv	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**1**	**2**	**3**	**4**
1	BB	AB	AB	AB	AB	AA	AB	BB	AB	BB	**4**	9	7	7
2	AB	BB	BB	AB	BB	BB	AB	AB	BB	AB	6	7	**3**	7
3	BB	BB	AA	AA	AB	AB	AA	AB	BB	AB	8	**3**	4	6
4	AB	BB	AB	AB	AB	AB	BB	AB	AA	AB	7	6	6	**3**
5	BB	AB	AA	AB	AA	AB	AA	AB	AA	BB	8	**3**	8	8
6	BB	AB	AB	AB	BB	BB	AB	AA	AB	AB	**2**	8	6	8
7	BB	BB	BB	BB	AB	AB	AA	AB	BB	AB	8	4	**2**	5
8	AB	BB	AB	AB	AA	AA	AB	BB	AB	BB	**5**	8	8	6
9	BB	AA	AB	AB	BB	AB	AB	AA	AB	AB	**1**	7	7	7
10	AB	BB	AB	BB	AB	AB	BB	AB	AB	AA	8	7	6	**0**
11	AA	BB	AA	AA	AA	AB	AA	AB	AB	AB	8	**2**	7	6
12	BB	AB	BB	BB	AB	BB	AB	BB	AA	AB	7	7	**3**	8
13	AB	BB	AB	AA	AB	AB	BB	AB	AA	AA	9	5	7	**2**
14	BB	AA	AB	AB	BB	BB	AB	AA	AB	AB	**1**	8	6	8
15	AB	BB	BB	BB	AB	AA	AB	BB	AB	AA	7	9	5	**4**

Técnicas para evaluar clusters

Dos grandes tipos de formas de evaluación:

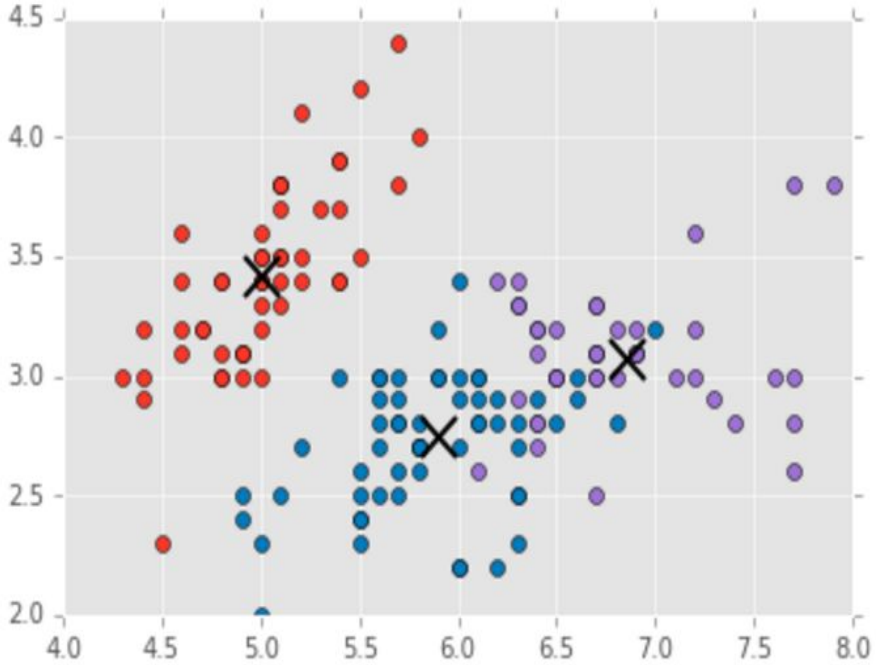
- **Cohesión Interna**
  - Buscan evaluar qué tan parecidos entre sí son los miembros de un cluster (**homogeneidad**) y qué tan diferentes son con respecto al resto de los clusters.
  - Es decir, se basan en la información intrínseca que posee el dataset del que disponemos.
  - Por ejemplo, midiendo **qué tan cerca se encuentran los puntos del clúster respecto al centroide**, la suma de errores al cuadrado es una medida usual.
  - Además, pediríamos una buena separación, es decir, que los **diferentes centroides estén lejos uno de otro**.

Dos grandes tipos de formas de evaluación:

- **Externas:**
  - Utilizan **alguna variable que se asuma "correlacionada"** a la clusterización para evaluar la performance de los clusters.
  - La idea sería que el **clustering separe "correctamente" estas clases**.
  - Es decir, se basan en buena medida en el **conocimiento previo** sobre el problema en cuestión.

VISUALIZACIÓN

- Corroborar visualmente.
- Después de ejecutar el algoritmo y calcular los centroides como lo hicimos en la clase anterior, podemos trazar los clusters resultantes para ver dónde se posan los centroides y cómo se agrupan los clusters.
- Tiene aplicabilidad limitada.



DigitalHouse > Coding School

SILHOUTTE SCORE

- El coeficiente de silueta, es la medida de **cuán estrechamente relacionado** está un punto con miembros de su grupo en lugar de con miembros de otros grupos.
- Si s de un punto es grande, la distancia media del punto dentro del clúster es menor que la distancia promedio a los puntos en el cluster vecino => por lo que el punto está bien clasificado.
- Si es pequeña, la distancia media del punto dentro del grupo es mayor que la distancia promedio al objetos en el clúster vecino, por lo que el punto se ha clasificado erróneamente.

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

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

$$-1 \leq s(i) \leq 1$$

Donde:

"a" es la distancia media entre el punto y todos los demás puntos del mismo cluster.

"b" es la distancia media entre el punto y todos los puntos del cluster vecino más cercano.

SILHOUTTE SCORE

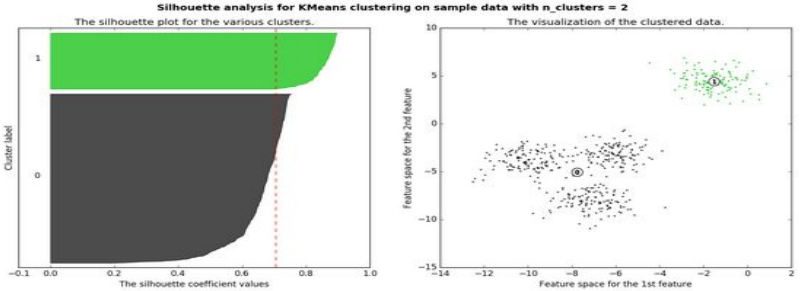
- Esta métrica permite evaluar qué tan bueno es el proceso de clusterización tanto en un solo elemento como en el agregado.
- El coeficiente de silueta de todo el set es dado por la media de los coeficientes calculados para cada punto.

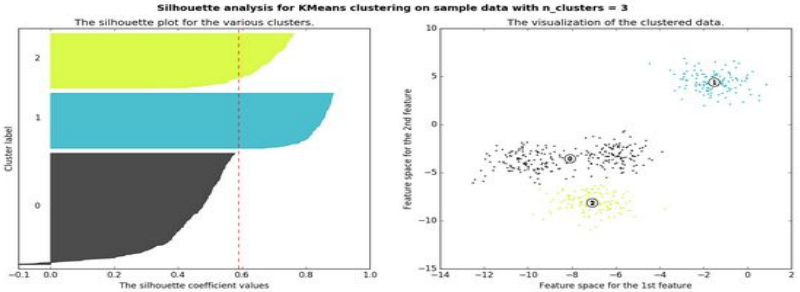
$$sil(C) = \overline{sil(k)} = \frac{1}{k}\sum_{i=1}^{k} sil(C_i)$$

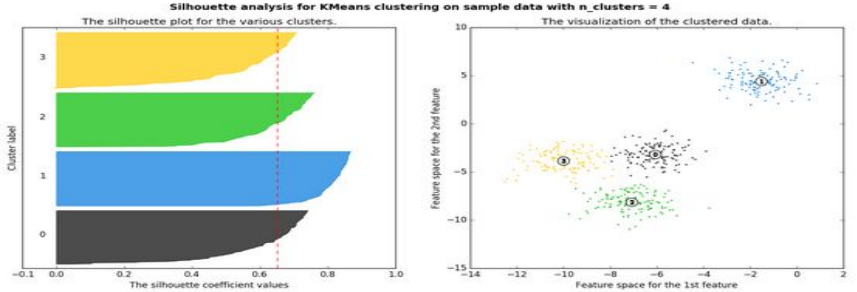
- Donde C_i es el la media para cada cluster

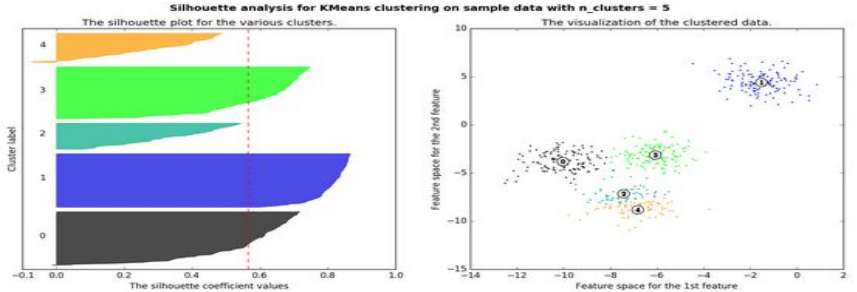
DigitalHouse > Coding School

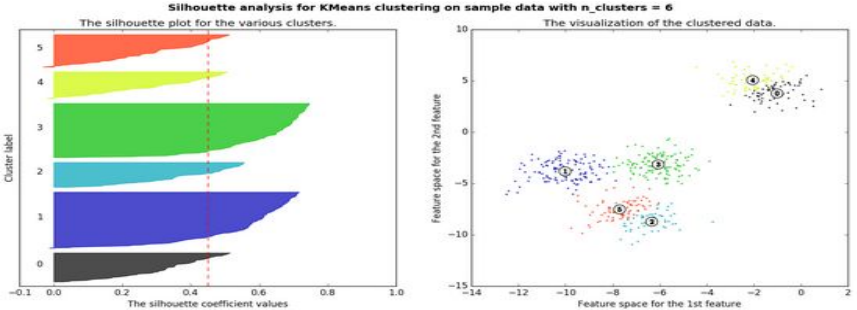
Ejemplo de Análisis de SC para distintos números de k-cluster











Observar y comparar **tamaño de los clusters** y valores de **sc por encima de la media.**
Para k= 3,5 y 6 es una mala elección. Entre 2 y 4 es más ambiguo. k=4 parece ser la correcta.

Calinski-Harabaz Index

- Se define como la razón entre la dispersión entre clusters y la dispersión al interior de los clusters.
- Cuanto mayor sea el score, mejor es el modelo de clustering: la dispersión entre clusters es mayor que la dispersión al interior de los mismos.
- Esto es bueno porque el cálculo tiene relación directa con el concepto de cluster.

$$s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1}$$

Donde:

"Bk" es la dispersión entre clusters

"Wk" es la dispersión al interior de los clusters.

Conclusiones

- Existen **diferentes métodos** para evaluar la calidad de nuestro análisis (incluyendo visualización, silhouette scores, F-metrics, y matrices de confusión.

- Después de analizar los Cluster, es posible que haya que redefinir la cantidad "k" de clusters buscados.

- Siempre es **conveniente examinar múltiples métricas** para entender la calidad del análisis realizado.