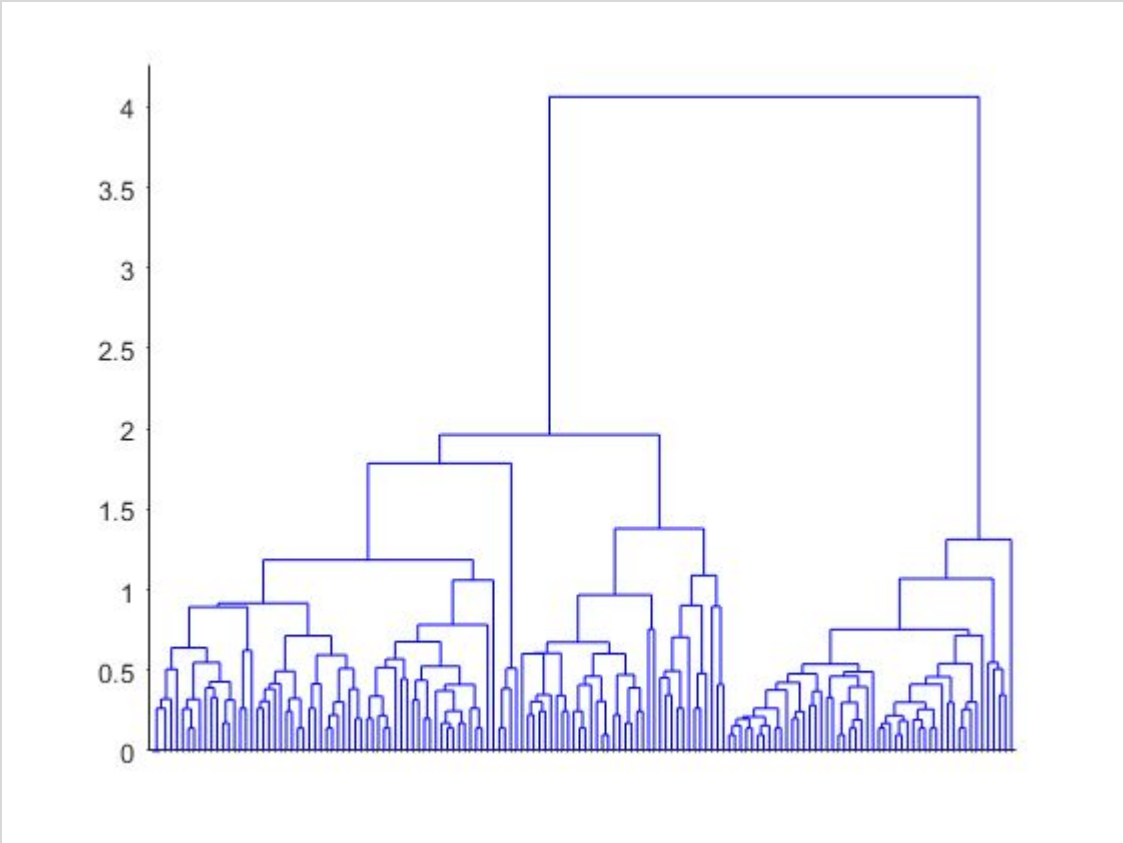
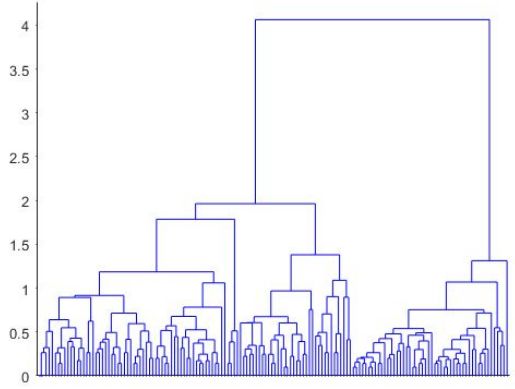


DigitalHouse > Coding School

DATA SCIENCE

Introducción al clustering jerárquico y DBSCAN

1. Introducir conceptos vinculados al clustering jerárquico
2. Explicar qué es un dendrograma y aspectos vinculados a las diferentes medidas de disimilitud
3. Introducir el algoritmo de clustering DBSCAN
4. Enunciar las diferencias y similitudes con el método de K-Means



Clustering Jerárquico

- Como ya vimos, K-Means requiere que especifiquemos de antemano cuál es el número de clusters que vamos a construir (k). Esto puede constituir una desventaja.

- El **clustering jerárquico** es una técnica que no requiere esa definición de antemano: **no es necesario que definamos previamente el k**.

- Además ofrece una **representación visual** de las observaciones **en forma de árbol** que nos permite observar en una mirada los clusters obtenidos para cada uno de los posibles k (desde 1 hasta n).

- Vamos ver los métodos de clustering llamados **“aglomerativos” o “bottom-up”**, de los más comunes. La denominación se refiere al hecho de que el dendograma es construido comenzando desde las hojas (los “puntos”) hasta el tronco.

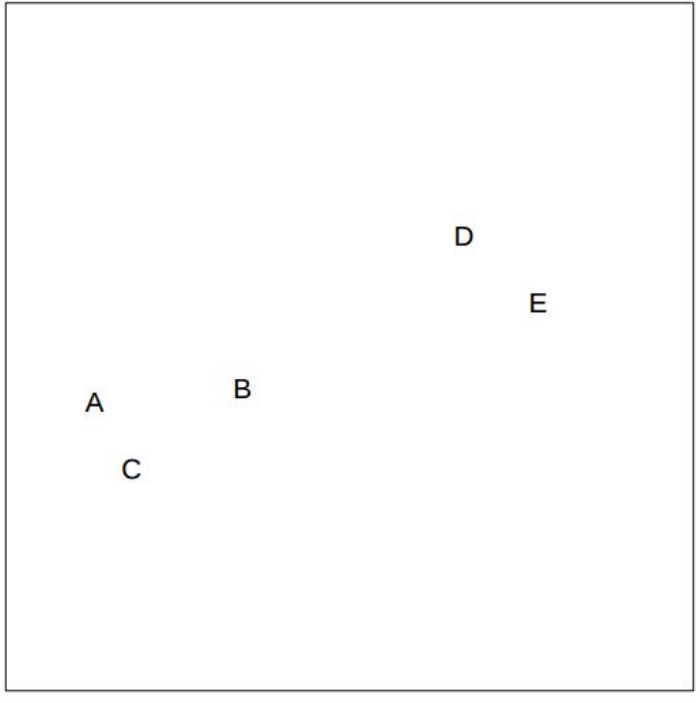
Clustering Jerárquico

Visión general del algoritmo

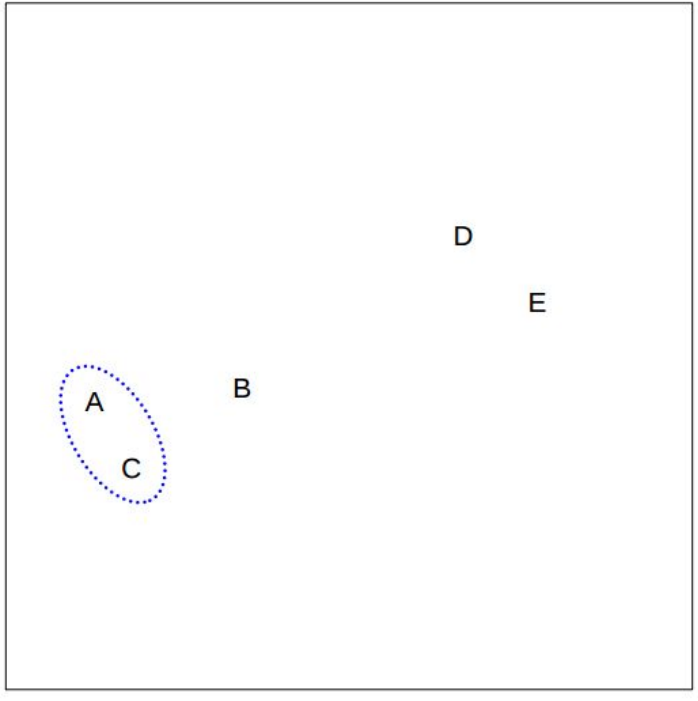
El enfoque de **clustering jerárquico** funciona de la siguiente forma:

- Comienza con **cada punto como un cluster**.
- Identifica los **dos clusters más cercanos y conforma un cluster** aglomerando esos dos:
    - Para esto computa alguna **medida de disimilaridad** entre todos los clusters.
- Se **repite** hasta que todos los puntos conforman un **único cluster**.

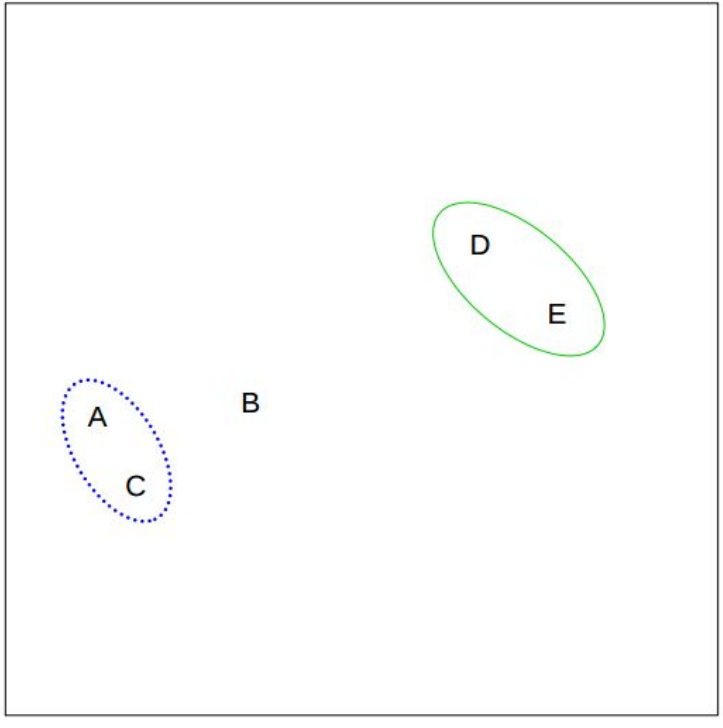
Ejemplo del proceso de clustering aglomerativo:



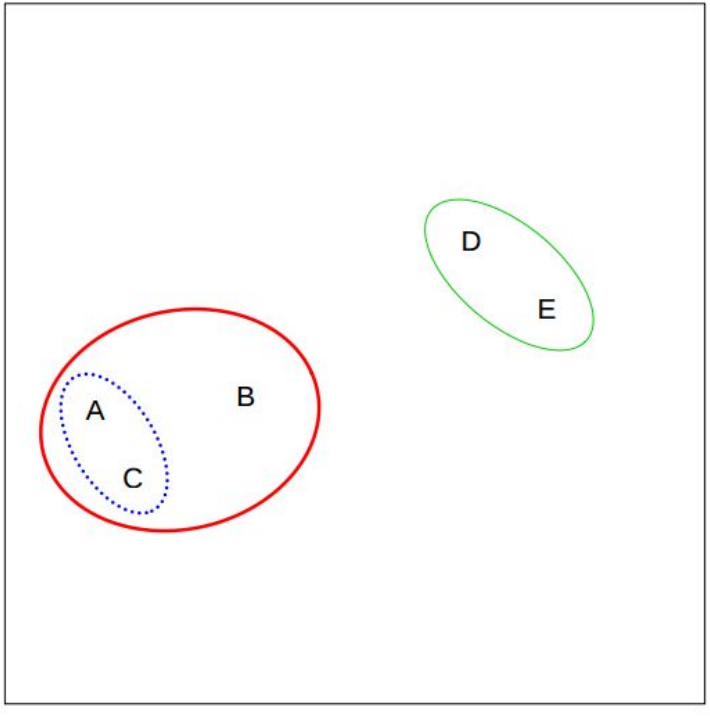
Ejemplo del proceso de clustering aglomerativo:



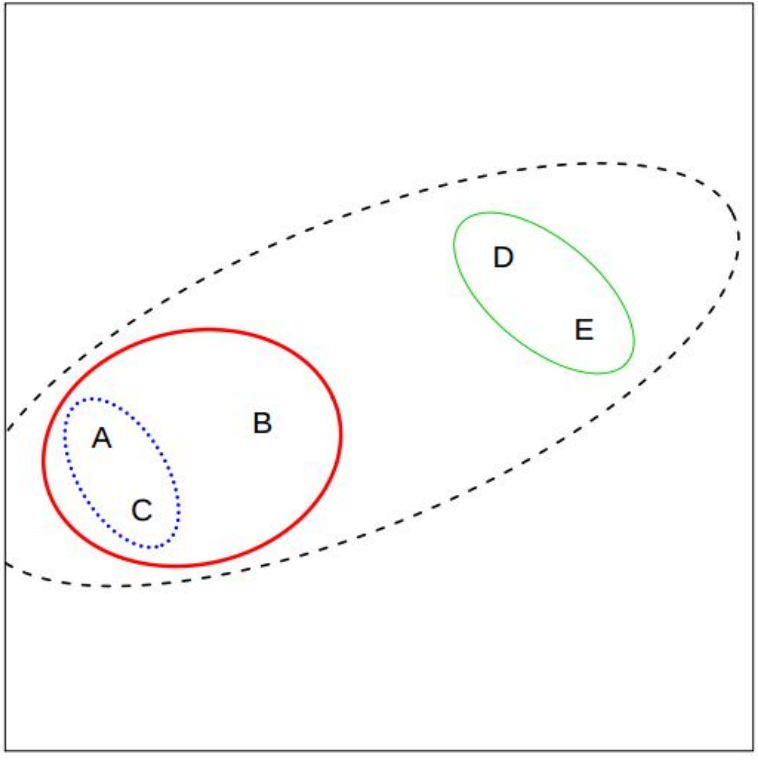
Ejemplo del proceso de clustering aglomerativo:



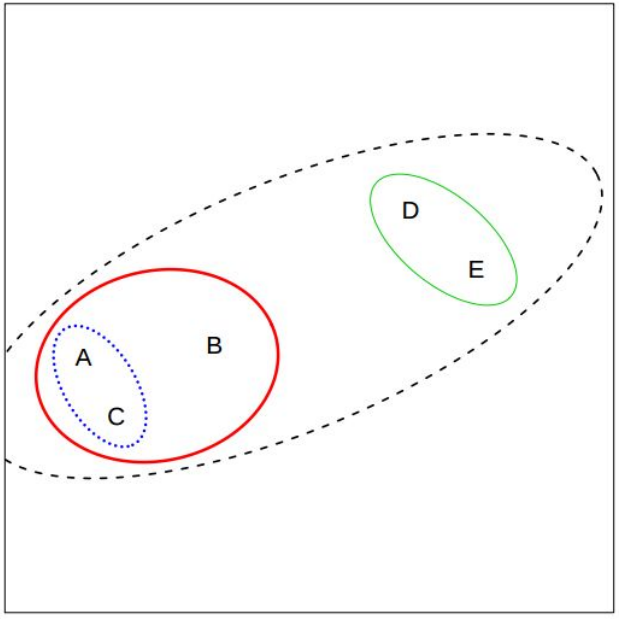
Ejemplo del proceso de clustering aglomerativo:

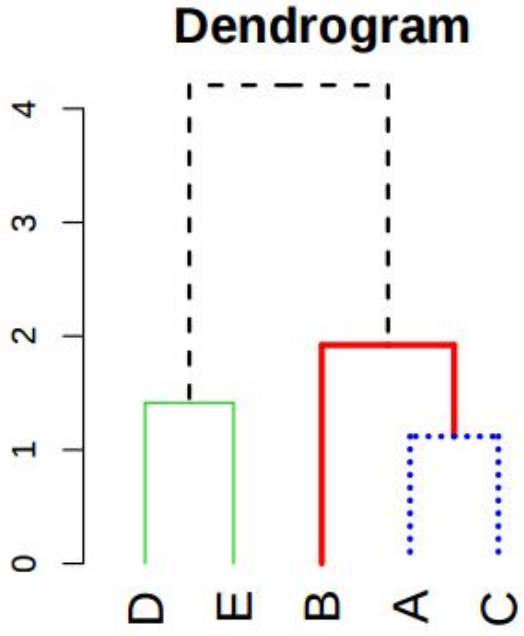


Ejemplo del proceso de clustering aglomerativo:



Ejemplo del proceso de clustering aglomerativo:

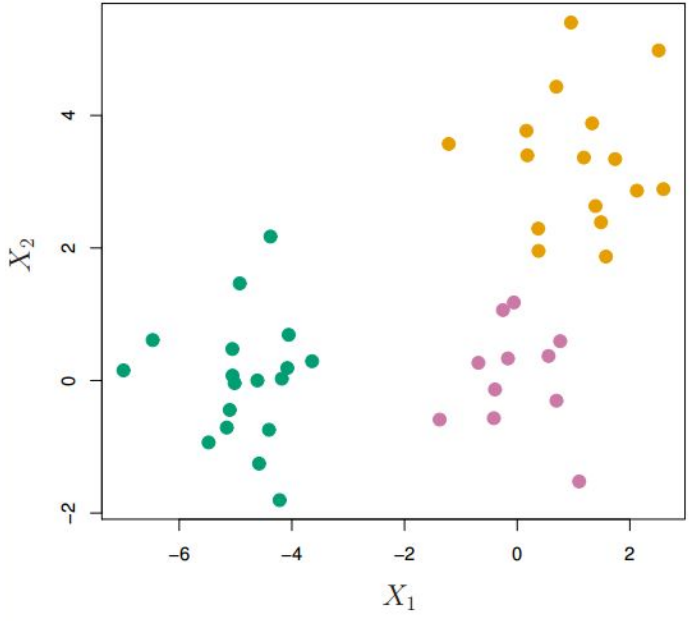


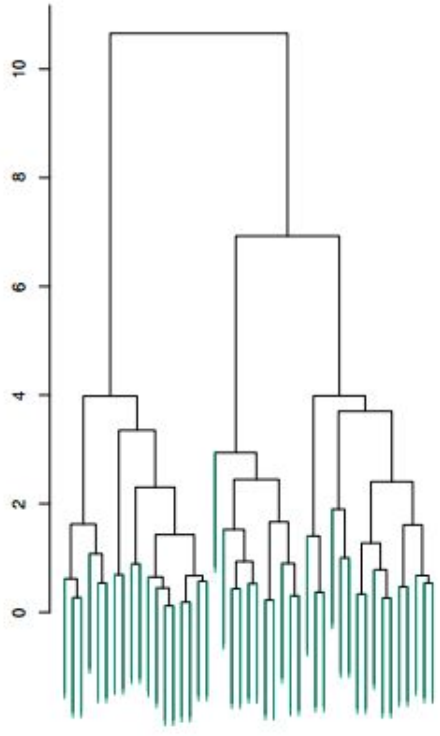


¿Cómo interpretar el dendrograma?

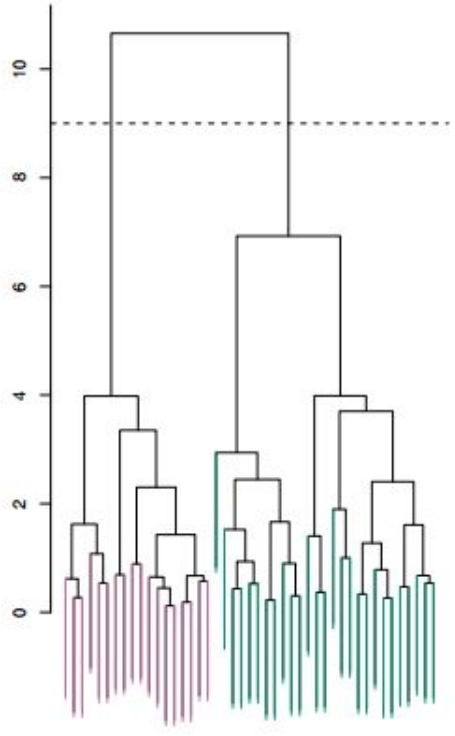
Para interpretar el **dendrograma** veamos un ejemplo un poco más complejo

- Comenzamos con 45 observaciones generadas en un espacio de dos dimensiones.
- Al final del dendrograma, cada observación es un solo cluster.
- **A medida que nos movemos hacia arriba del árbol, algunas hojas se empiezan a fusionar**: las que corresponden a observaciones muy similares.
- A medida que avanzamos más arriba del árbol, un número creciente de observaciones se han fusionado. **Cuanto más temprano (más bajo en el árbol) dos observaciones se funden, más similares son entre sí**.
- Las observaciones que se unen más arriba son las más diferentes

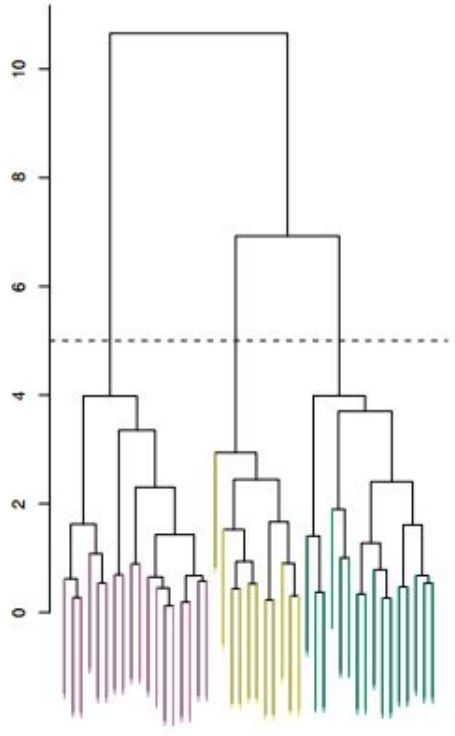




Un cluster



Dos clusters



Tres clusters

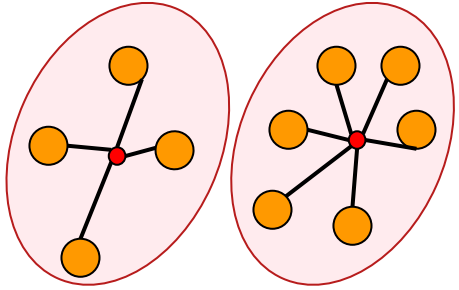
¿Cómo formar los clusters?

Problema a resolver

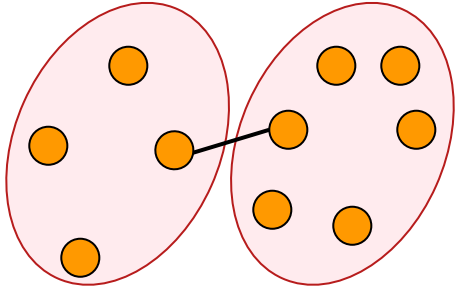
¿Cómo definimos la forma en que dos clusters se unen? ¿Qué medida usar? Algunas de las más utilizadas son:

TIpo de linkage	Descripción
Single	Calcula todos los pares de distancias entre los miembros del cluster A y el cluster B y utiliza la **mínima**.
Completo	Calcula todas los pares de distancias entre los miembros del cluster A y el cluster B y utiliza la **máxima**.
Average	Calcula todas los pares de distancias entre los miembros del cluster A y el cluster B y utiliza el **promedio de todas.**
Ward	Calcula la diferencia en la varianza total generada al aglomerar los diferentes clusters y busca la mínima

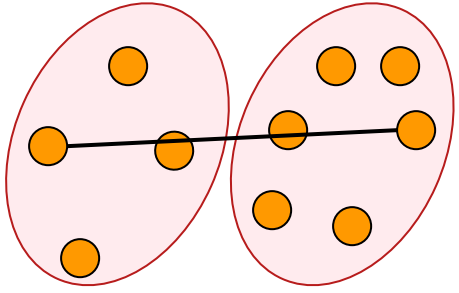
Ward Linkage



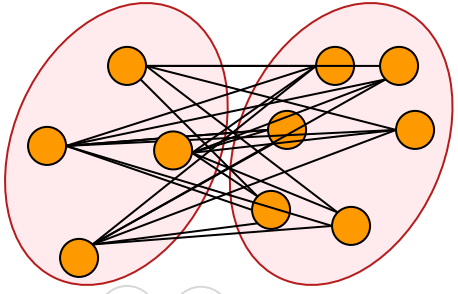
Single Linkage



Complete Linkage



Average Linkage



El tipo de **linkage** puede generar resultados diferentes:

- ***Ward***: es la opción por default. Tiende a generar clusters de dimensiones similares. Funciona bien para gran parte de los casos.

- Si tenemos clusters con diferentes cantidades de miembros, ***complete*** y ***average*** son buenas opciones.

- En cambio, ***single linkage*** tiende a generar clusters extendidos en los que las hojas se van fusionando una por una.

¿Cómo podemos **evaluar** un clustering jerárquico?

Una forma es a través del **coeficiente Cophenético** (Cophenetic Coefficient)

- Compara las distancias originales entre los puntos con las distancias que surgen del agrupamiento generado por el proceso de clustering
  - La idea es que valores cercanos a 1 del coeficiente indica que las dos distancias están muy correlacionadas.

$$c = \frac{\sum_{i<j}(x(i,j) - \bar{x})(t(i,j) - \bar{t})}{\sqrt{[\sum_{i<j}(x(i,j) - \bar{x})^2][\sum_{i<j}(t(i,j) - \bar{t})^2]}}.$$

  - x(i,j) son las distancias entre los puntos
  - t(i,j) son las distancias entre los clusters (el punto en el que se unen en el dendograma)

Clustering Jerárquico: conclusiones

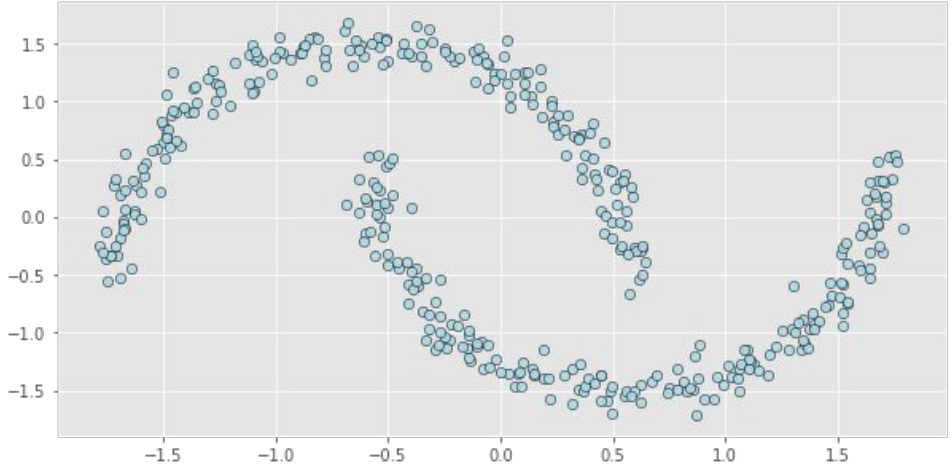
Conclusiones:

- Ofrecen una ventaja respecto a K-Means dado que **no requieren definir a priori la cantidad de clusters a crear**.
- Ofrecen una **representación gráfica (dendograma)** del proceso de generación de clusters.
- Hay varios **métodos** para definir la forma de unión (**linkage**) entre los diferentes clusters.
- La desventaja es que es **más costoso computacionalmente** que K-Means.

DBSCAN

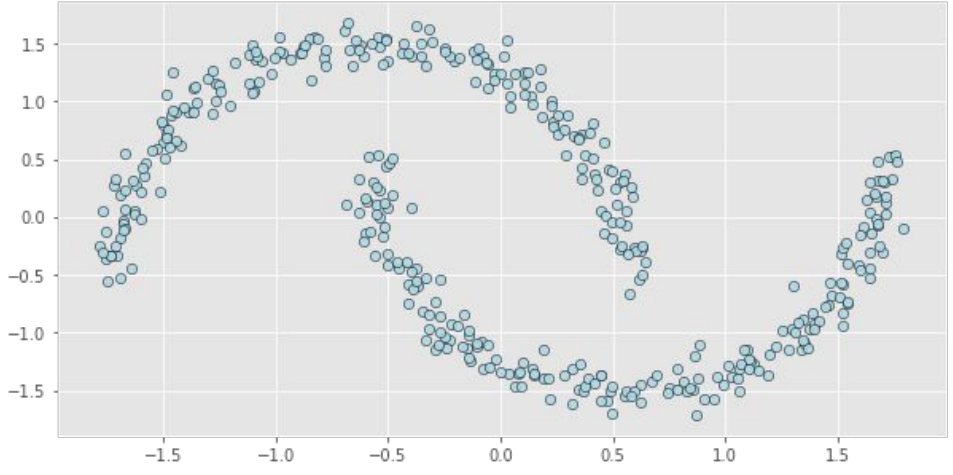
K-Means es un algoritmo de clustering basado en distancias

- Características de K-Means:
  - Tenemos que definir la cantidad de clusters
  - Sensible a la aleatoriedad de la inicialización de los centroides
  - Susceptible a outliers: asigna todos los puntos a un clúster. No puede detectar el ruido en los datos.
  - Asume que los datos están distribuidos en formas convexas.



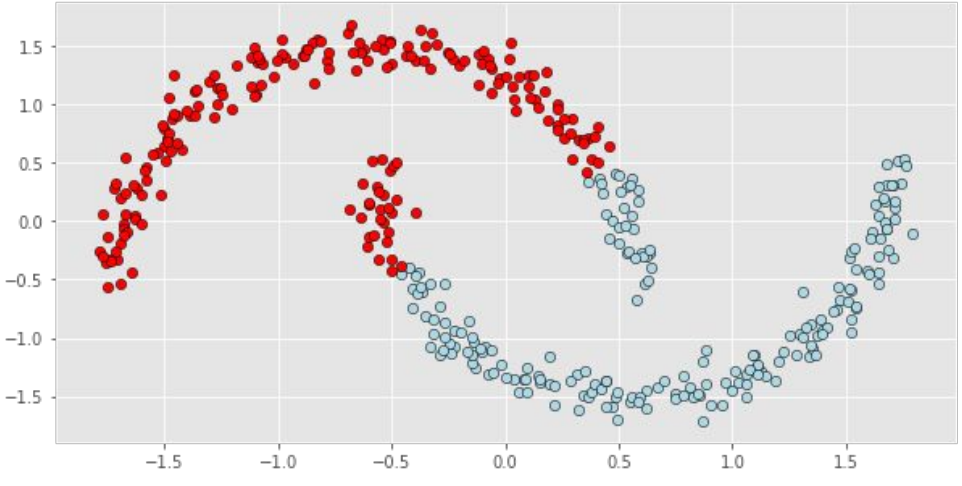
K-Means es un algoritmo de clustering basado en distancias

¿Qué pasa cuando nuestros datos están ordenados de formas complejas?



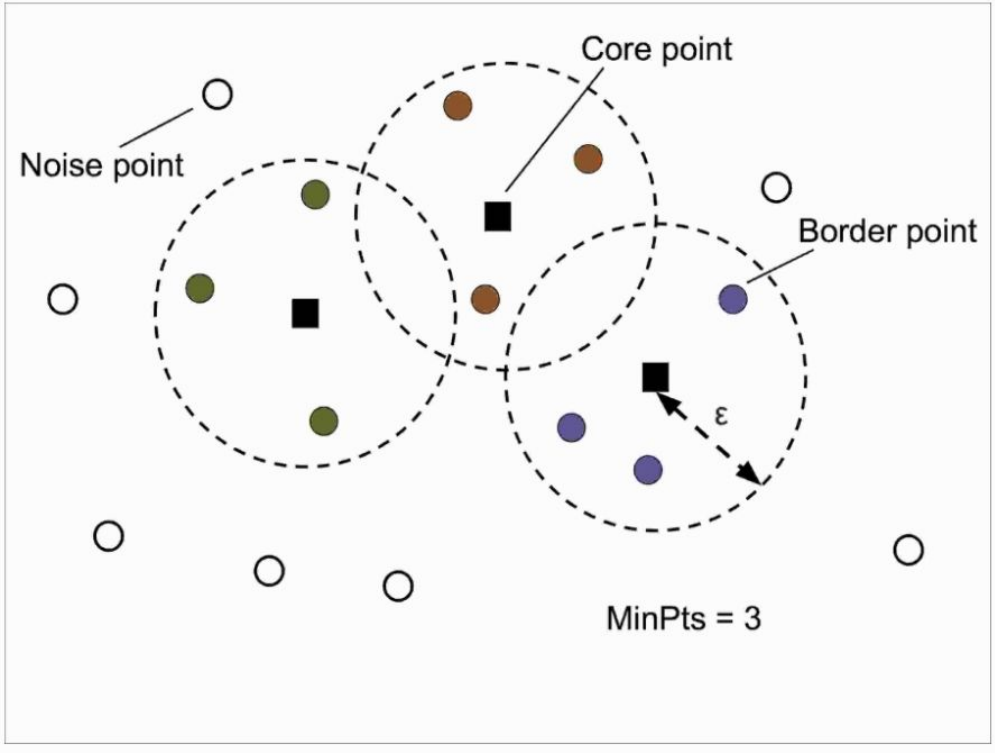
K-Means es un algoritmo de clustering basado en distancias

¿Qué pasa cuando nuestros datos están ordenados de formas complejas?



DBSCAN: ¿Cómo funciona?

DBSCAN es un algoritmo basado en densidad espacial



Dos parámetros:

- **epsilon:** máxima distancia entre dos puntos para considerarlos como pertenecientes al mismo cluster (responde a lo que entendemos por "cercanía" para un determinado problema).
- **min_points:** es decir, el mínimo de puntos necesarios para formar un cluster (la idea de este parámetro es evitar la formación de clusters demasiado pequeños).

Visualización muy útil para entender DBSCAN y K-means:

- https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Diferencias con k-means y jerárquico

DBSCAN:

- Funciona muy bien con clusters con **límites no lineales** y de **tamaños muy diferentes.**
- La diferencia fundamental con k-means es que se basa en la "**densidad**" en lugar de comenzar por la distancia a un punto central (centroide).
- No requiere la definición de un número de clusters a priori.
- También es muy útil cuando tenemos datos muy densamente distribuidos:
  - en estos casos, es probable que k-means nos arroje un solo cluster
  - DBSCAN debería poder "romper" este agrupamiento en partes menores
- Incorpora en el proceso el concepto de "ruido", lo que lo hace **robusto a outliers**.
- Es **determinístico**, una vez fijados los parámetros iniciales (salvo para los puntos de frontera).

- Una de las mayores dificultades se presenta en datasets con **zonas con densidades muy diferentes** en el espacio de puntos: en estos puede no funcionar correctamente.

- En algunos casos, si no se tiene clara la escala de los datos, **los parámetros pueden ser difíciles de encontrar**.