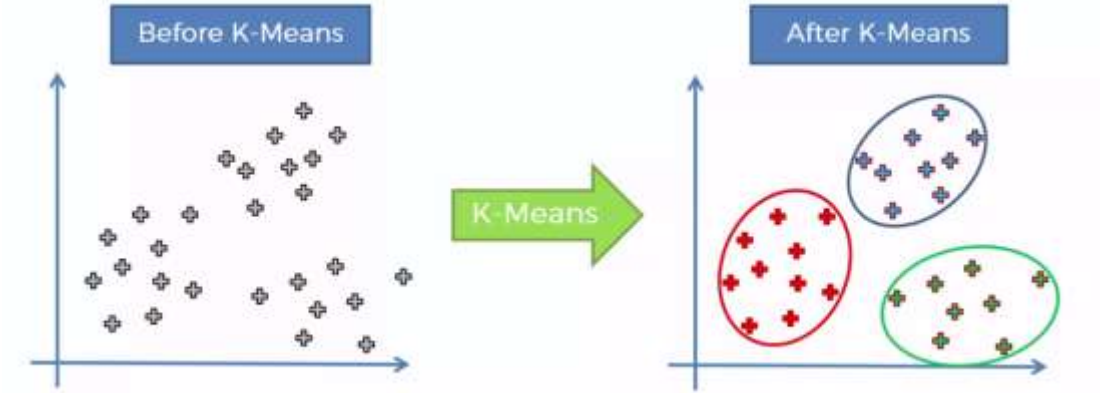

# Algoritmos de agrupación

# Clustering

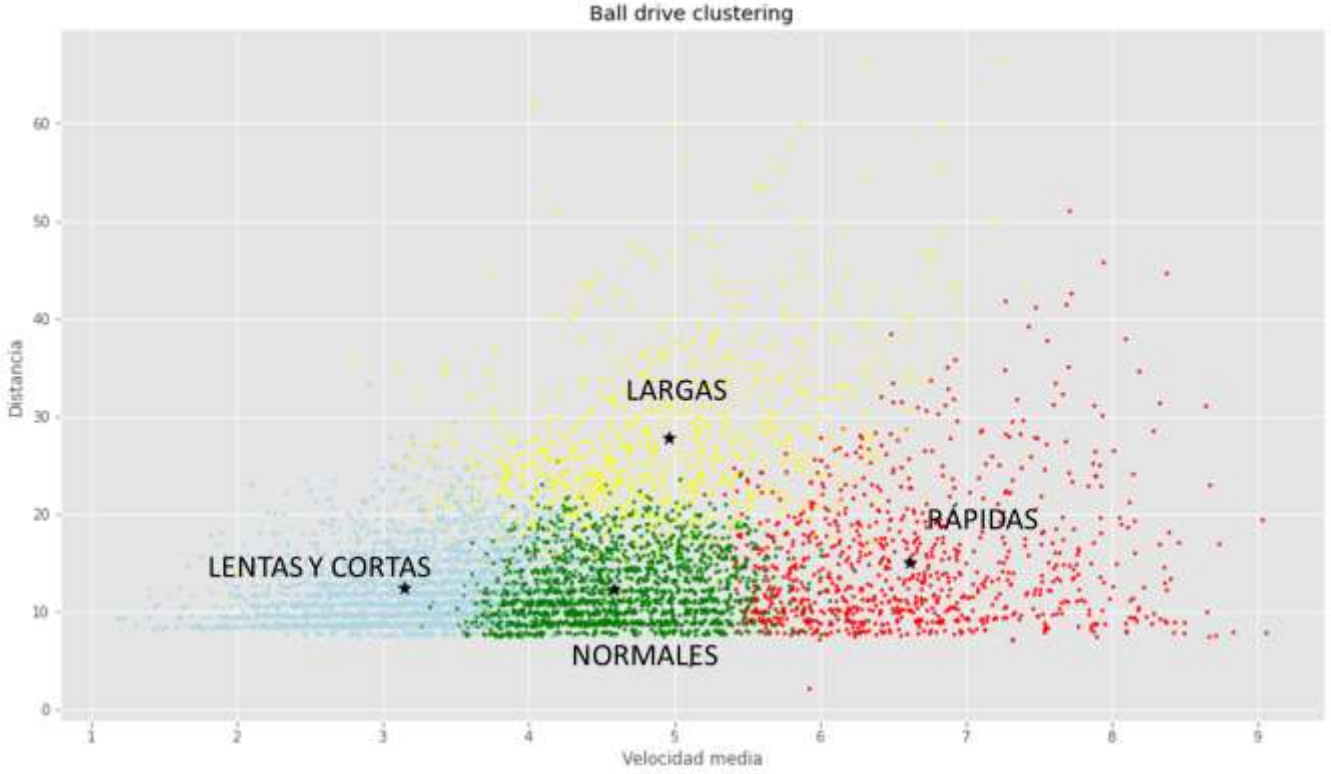
- Los algoritmos de clustering forman parte de las técnicas de aprendizaje no supervisado, en las que no hay resultado que predecir y el algoritmo solo trata de encontrar patrones en los datos.

- El algoritmo de agrupación más utilizado es el denominado k-means



# Ejemplo: Conducciones de balón

Tipos de conducción



# Clustering K-means

En K-means tenemos que definir de antemano:

1) El número de clusters que queremos obtener
2) Cuál es la medida de similaridad que usaremos

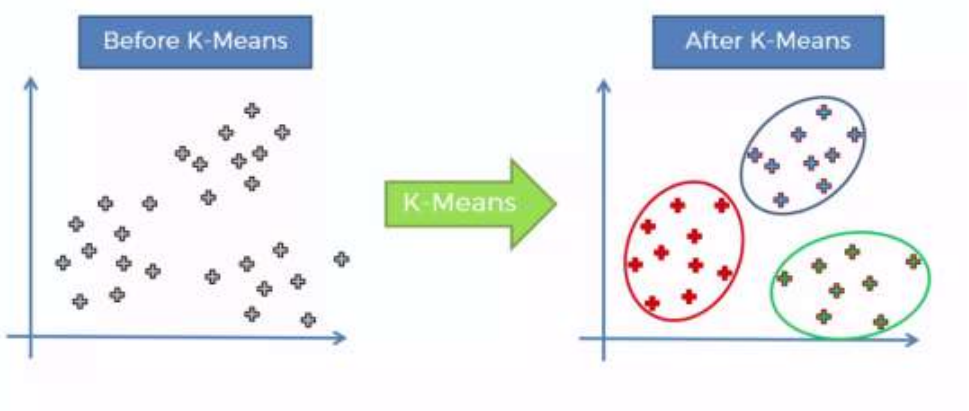
El algoritmo asignará cada observación únicamente a un cluster, de manera que las observaciones de cada cluster tengan:

- La máxima similitud (mínima distancia) entre ellos
- La mínima similitud (máxima distancia) con las observaciones de otros clusters

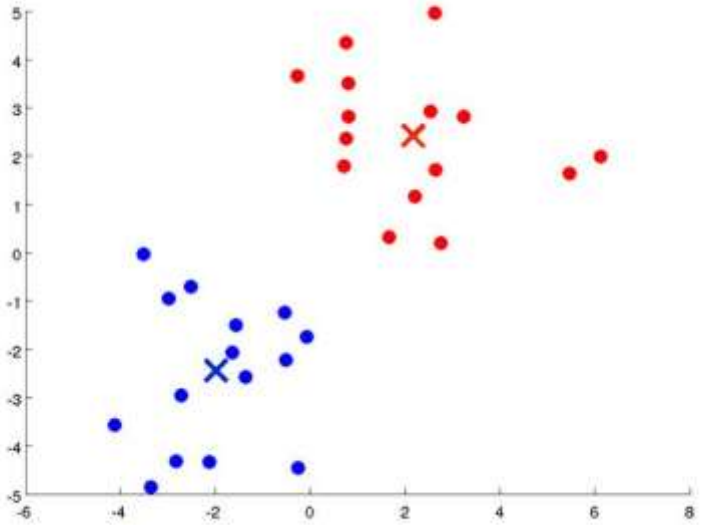
En este ejemplo hemos ejecutado kmeans para:

- Obtención de tres clusters
- Usando la distancia euclídea

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$



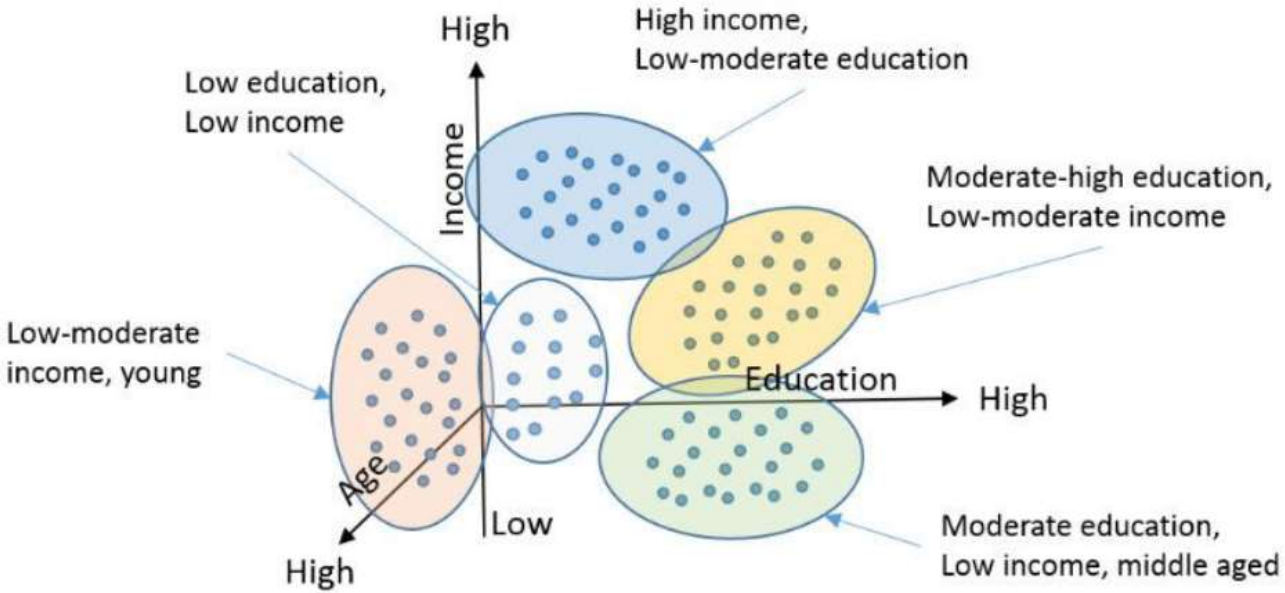
# Clustering K-means




https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

# Clustering K-means

Una vez finalizado el entrenamiento y generado los clusters, debemos saber interpretar los resultados



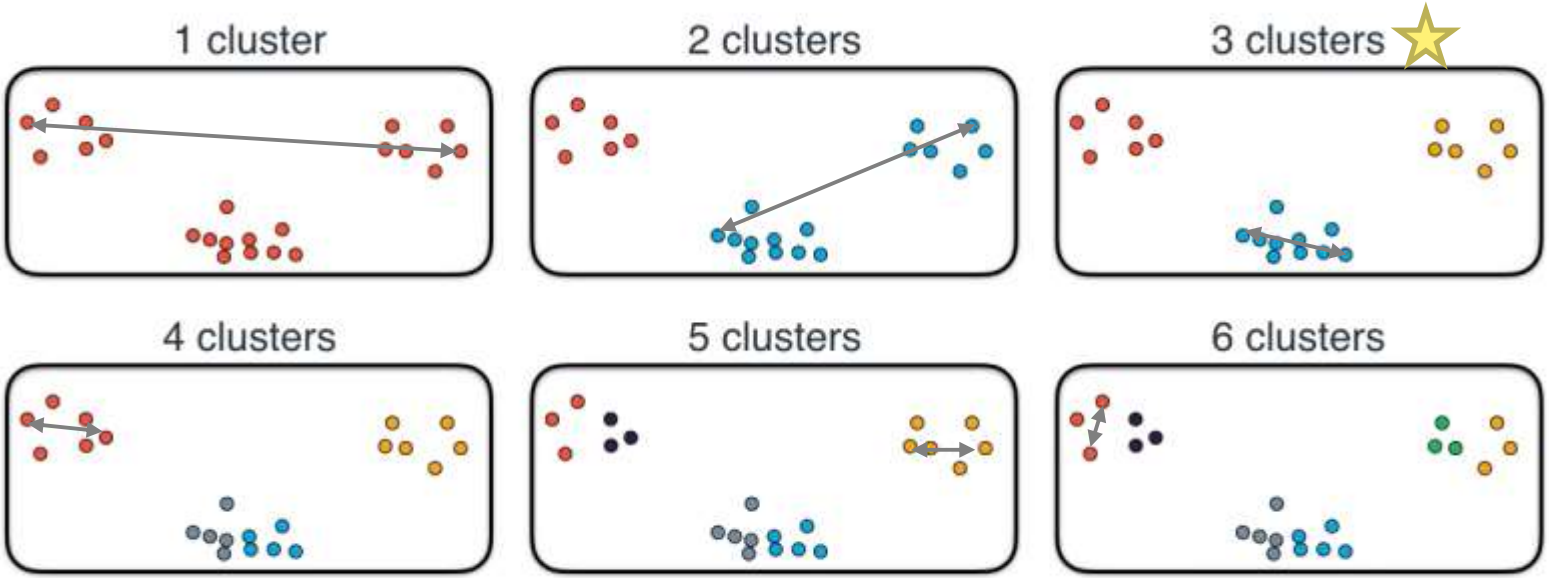
# Clustering K-means: Medir desempeño

- No hay un estándar sobre cómo medir el desempeño de k-means
- Tenemos la problemática de la ausencia de etiquetas
- Algunas ideas:
  - Ver si los clusters responden a la idea inicial que teníamos
  - Analizar algunas muestras de cada cluster y comprobar si son homogéneas y tienen sentido
  - Utilizar herramientas de visualización multidimensional

- Problemas:
  - No está asegurado que se encuentre una solución óptima
    - Probar varias ejecuciones y elegir la que proporcione mejor desempeño
    - Inicialización no aleatoria (Kmeans++)

# Método del codo

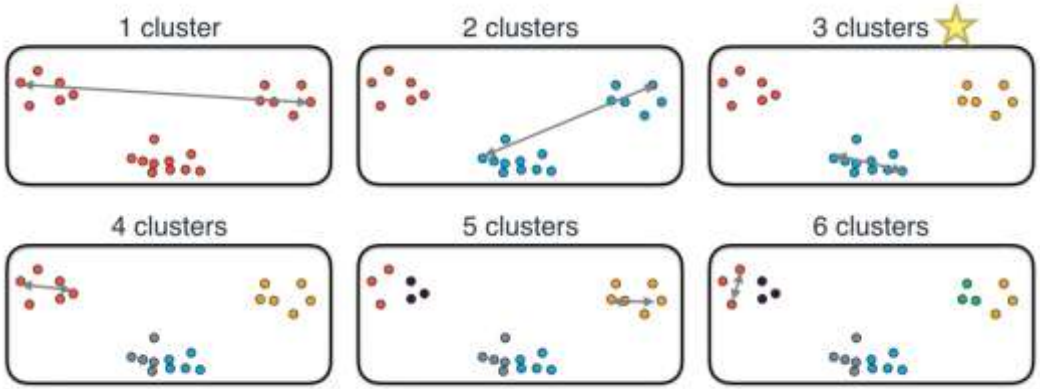
- ¿Cuántos clusters elegimos?

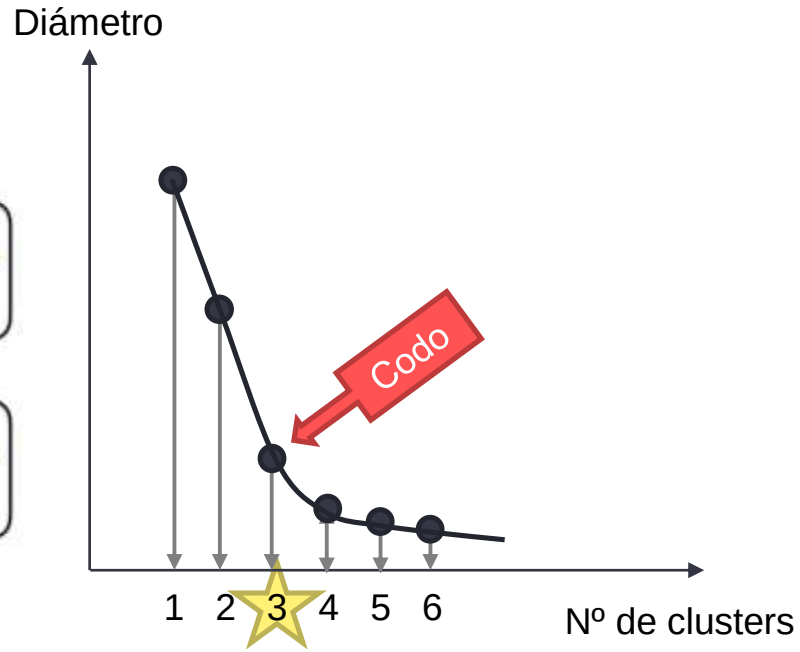



- Medimos la máxima distancia entre dos puntos del mismo cluster

# Método del codo

- ¿Cuántos clusters elegimos?



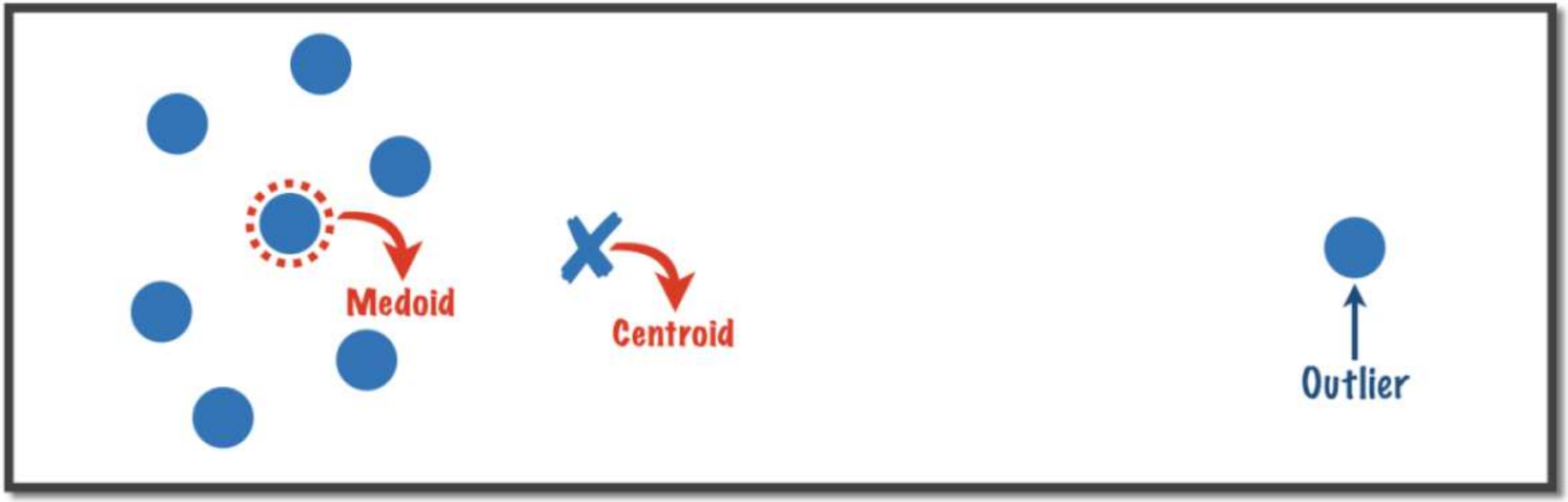




Una forma equivalente de hacer la gráfica es mediante la suma de las distancias de todos los puntos a su centroide (inercia)

# El efecto de los outliers

Existen algoritmos similares, como k-medoids, el cual:

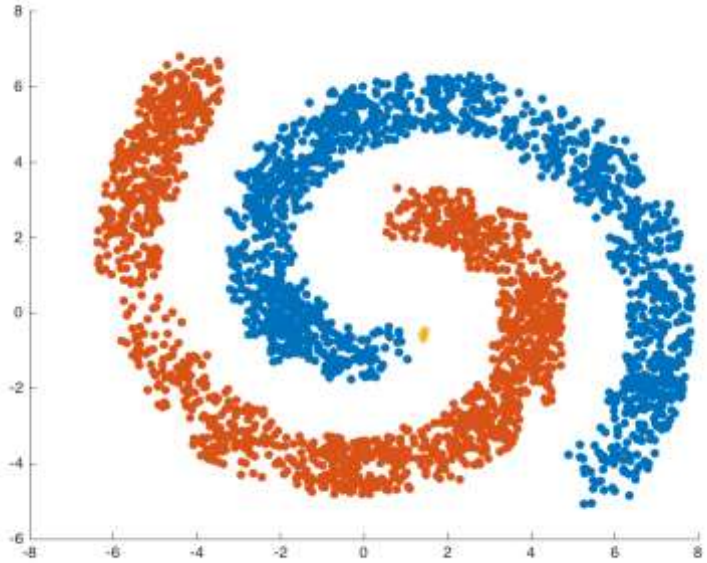
- Usa puntos de datos como centroides
- Usa la mediana en lugar de la media para ser menos sensible a outliers



# DBSCAN

El algoritmo K-means no es muy eficaz cuando los clusters tienen formas complejas

Para solucionarlo, el algoritmo DBSCAN busca agrupaciones basadas en densidad de puntos



# DBSCAN

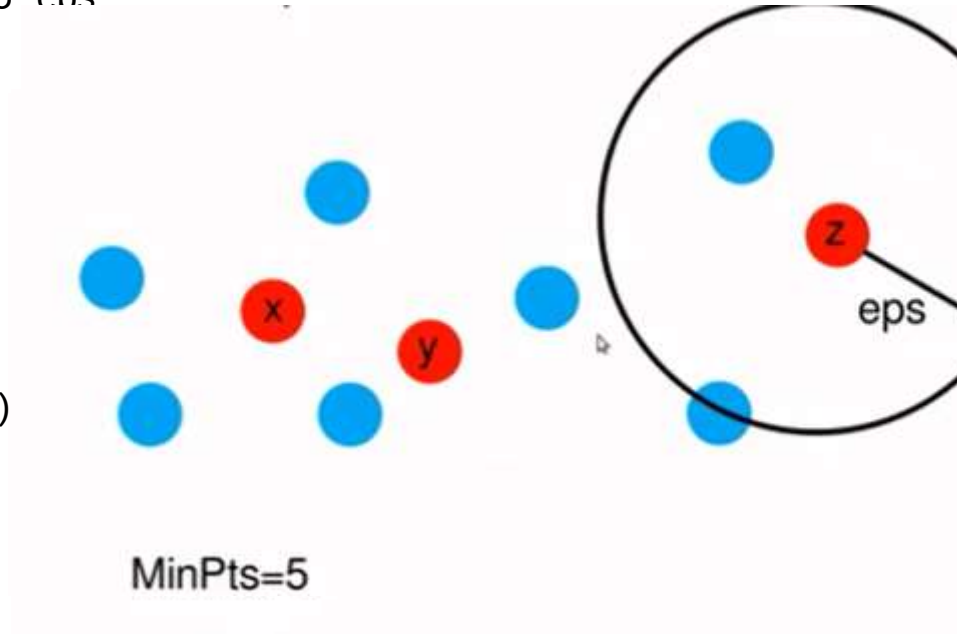
DBSCAN tiene dos parámetros:

1) eps(ilon): radio para considerar puntos como vecinos
2) min_samples: número mínimo de puntos en el radio “eps”

x es un punto CORE
(tiene al menos min_samples en un radio eps)

y es un punto BORDER
(menos de min_samples pero un core dentro del radio)

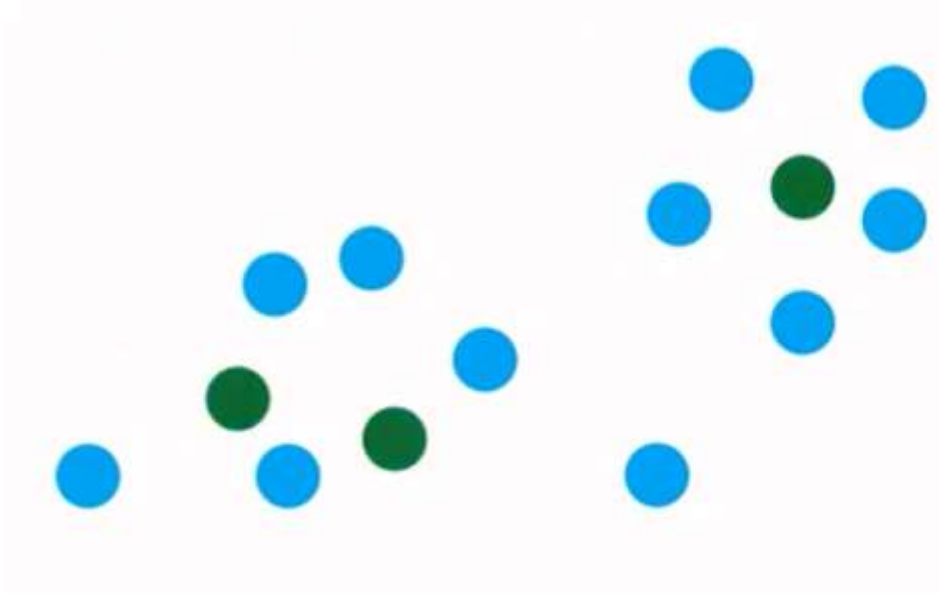
z es un punto NOISE
(no es ni core ni border)



# DBSCAN

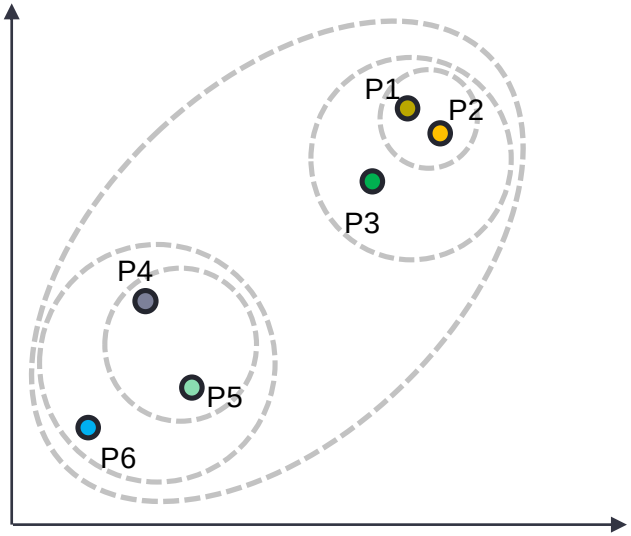
Etapas del algoritmo:

1. Busca todos los puntos CORE
2. Para cada CORE, se crea un cluster (si no se le ha asignado ya)
3. Buscar recursivamente los puntos en el radio eps y asignarlos al mismo cluster
4. Los puntos sin asignar son considerados outliers o ruido

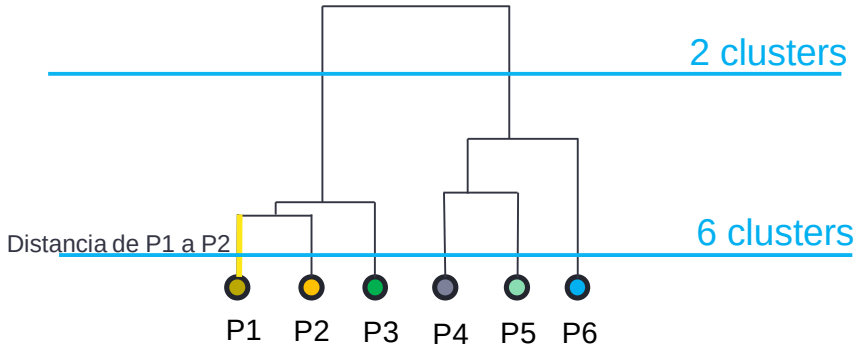


https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

# Clustering jerárquico




Dendrograma




Representa todos los posibles clusters que puede haber en los datos

Al algoritmo podemos especificarle el número de clusters que queremos o la distancia máxima para agrupar conjuntos. Generalmente, el umbral se sitúa de manera que corte el mayor segmento

# Clustering jerárquico

Ejemplo: Agrupar restaurantes en clusters con distancia entre ellos inferior a 1Km