

Machine Learning

Clustering

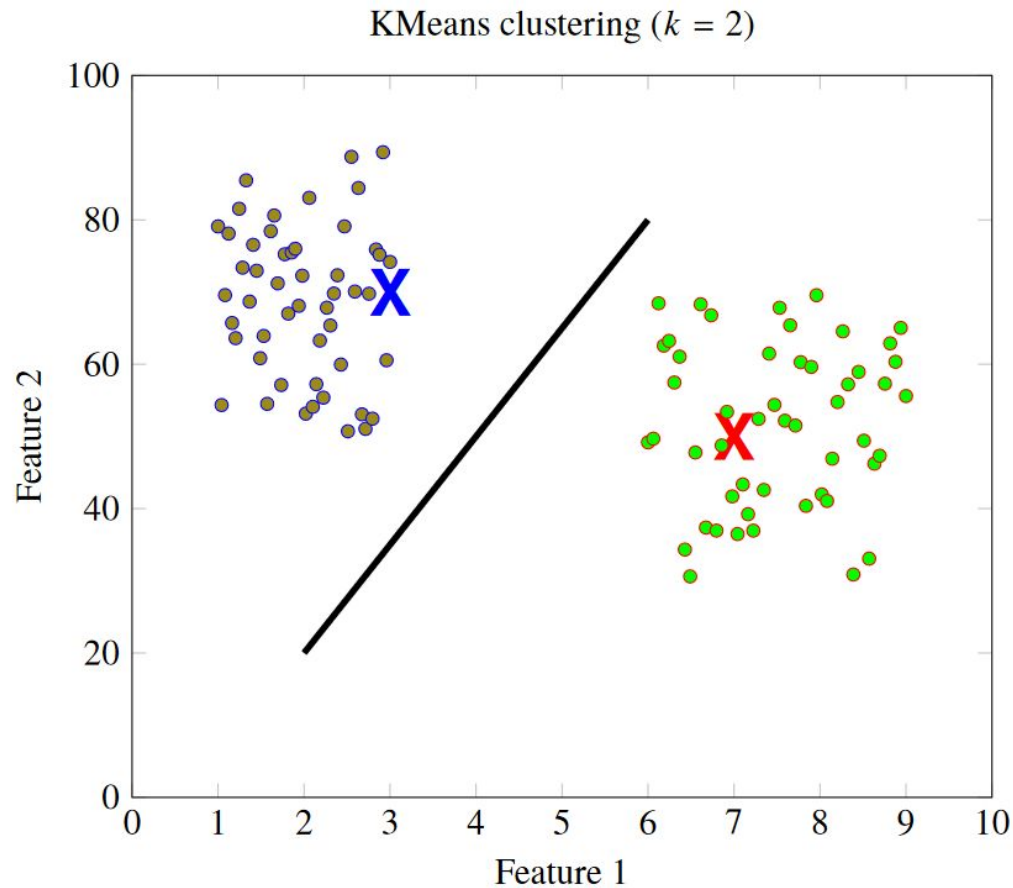
Christian Oliva Moya
Pedro Ramón Ventura Gómez

Introducción

- ¿Qué es clustering?
- ¿Diferencias entre aprendizaje supervisado y no supervisado?
- ¿Cuál es el objetivo de los algoritmos de clustering?
- ¿Qué significa proximidad o similitud?

Introducción

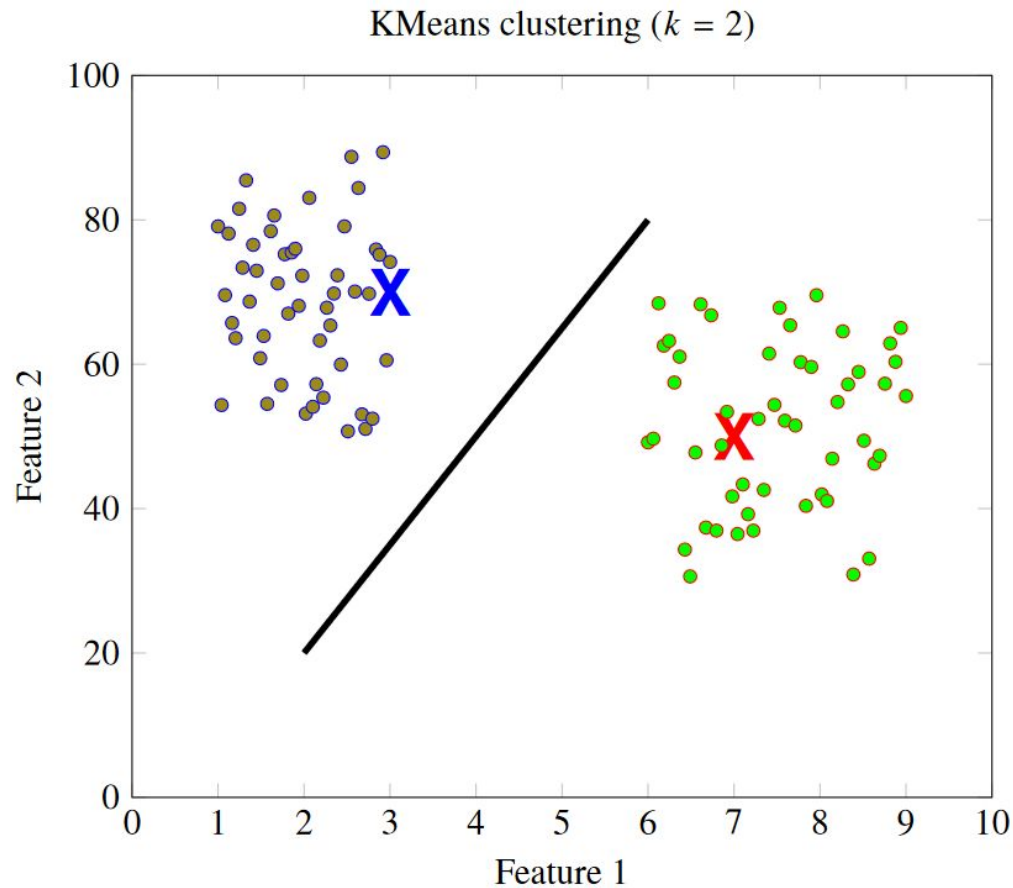
¿Esto es clustering?



Introducción

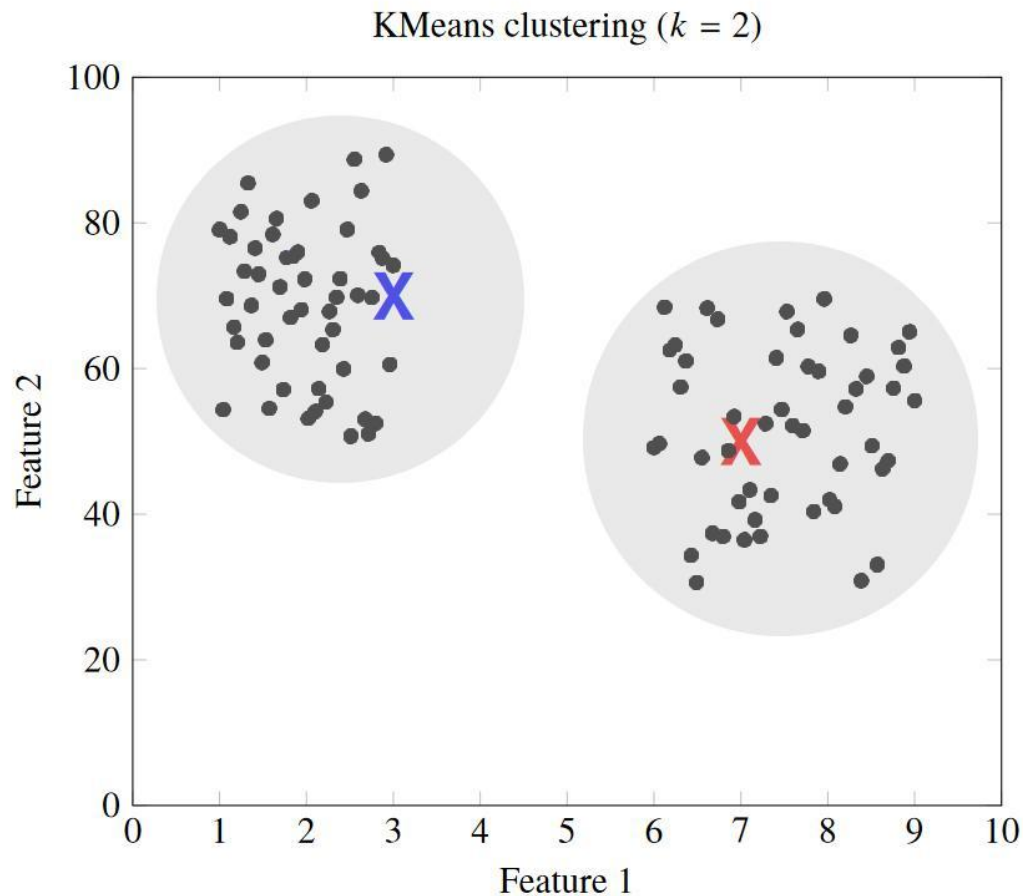
¿Esto es clustering?

NO!! Es un clasificador lineal



Introducción

Esto Sí es clustering



Introducción - Recordatorio

Clasificación

Los datos están etiquetados (supervisado)

Objetivo:

Etiquetar correctamente datos no vistos

Clustering

No hay etiquetas (no supervisado)

Los nuevos datos se agrupan según una
métrica de proximidad

Objetivo:

Identificar estructuras/patrones en los datos

Introducción - Objetivos del clustering

- Escalabilidad

El algoritmo debe ser escalable incluso con grandes cantidades de datos

- Robustez

Los outliers deben detectarse con precisión

- Independencia del orden

Diferente orden en la entrada no debe conducir a diferentes resultados finales

- Mínima elección de hiperparámetros definidos por el usuario

Introducción - Distancia

- Distancia Minkowski

$$d(x_i, x_j) = (|x_{i,1} - x_{j,1}|^g + |x_{i,2} - x_{j,2}|^g + \dots + |x_{i,p} - x_{j,p}|^g)^{1/g}$$

Si $g = 2 \Rightarrow$ Euclidean

Si $g = 1 \Rightarrow$ Manhattan

Introducción - ¿Qué vamos a ver?

- Clustering **jerárquico** (o de conectividad)
 - Aglomerativo
- Clustering basado en **centroides**
 - K-means y K-medoids
- Clustering basado en **mezcla de gaussianas**
 - Expectation Maximization
- Clustering basado en **densidades**
 - DBSCAN

Clustering Aglomerativo

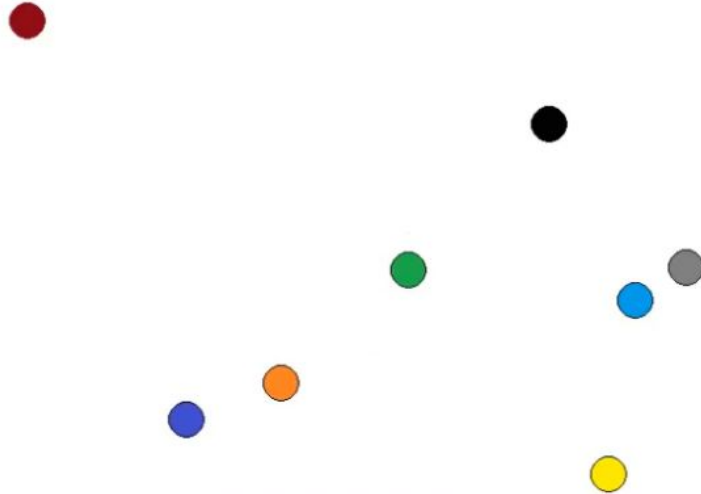
Jerárquico

Clustering jerárquico

- Clustering jerárquico (o de conectividad):

Los datos se van conectando (agrupando) de forma escalonada según los más próximos

Veamos un ejemplo:

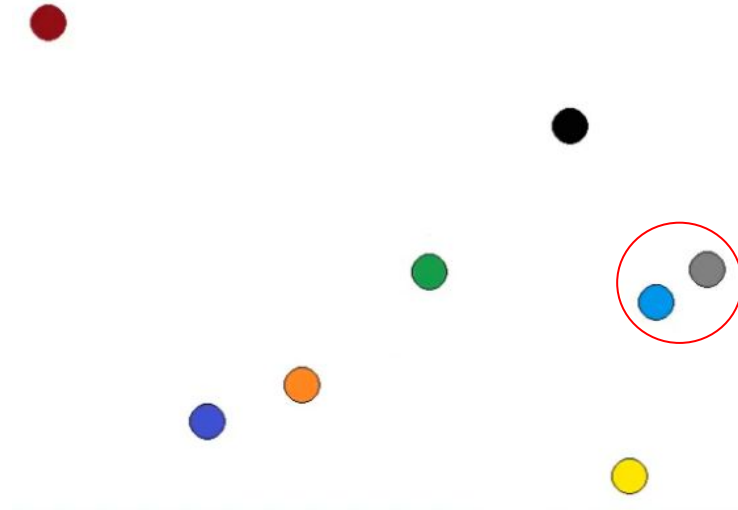


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

Los datos se van conectando (agrupando) de forma escalonada según los más próximos

Paso 1:

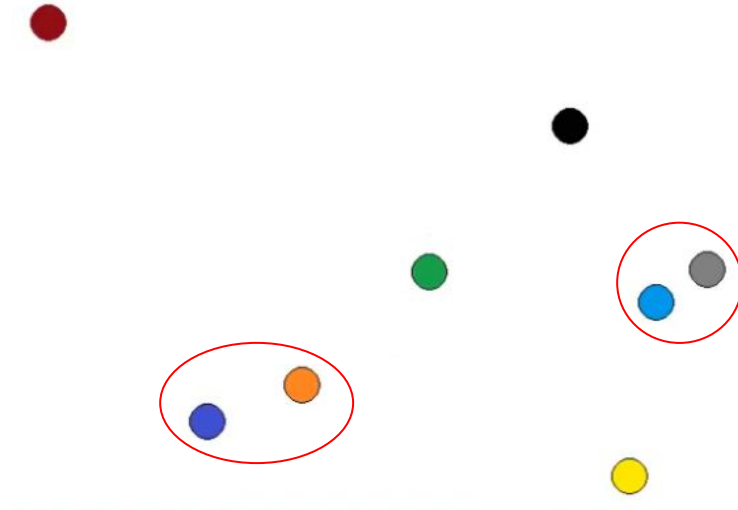


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

Los datos se van conectando (agrupando) de forma escalonada según los más próximos

Paso 2:

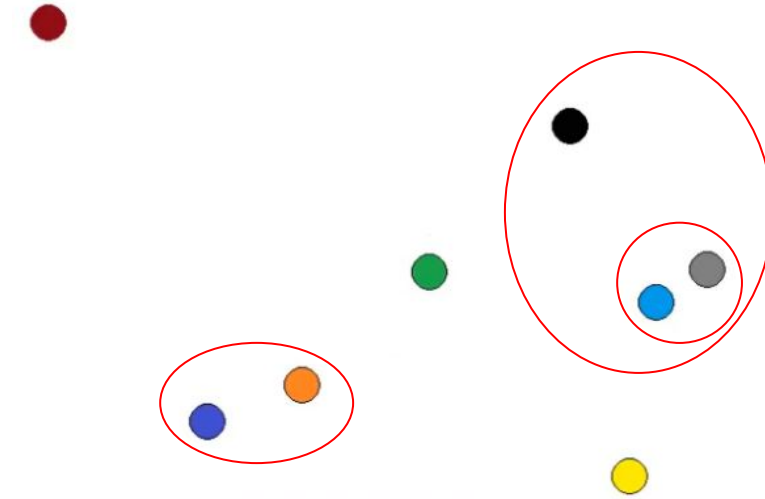


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

Los datos se van conectando (agrupando) de forma escalonada según los más próximos

Paso 3:

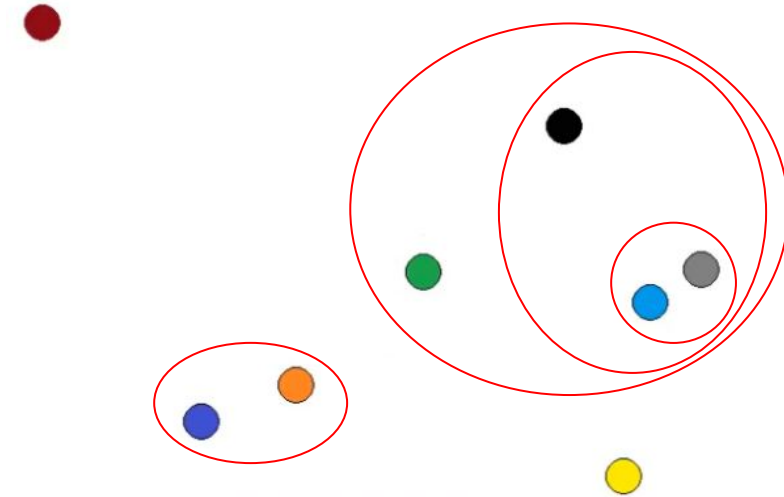


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

Los datos se van conectando (agrupando) de forma escalonada según los más próximos

Paso 4:

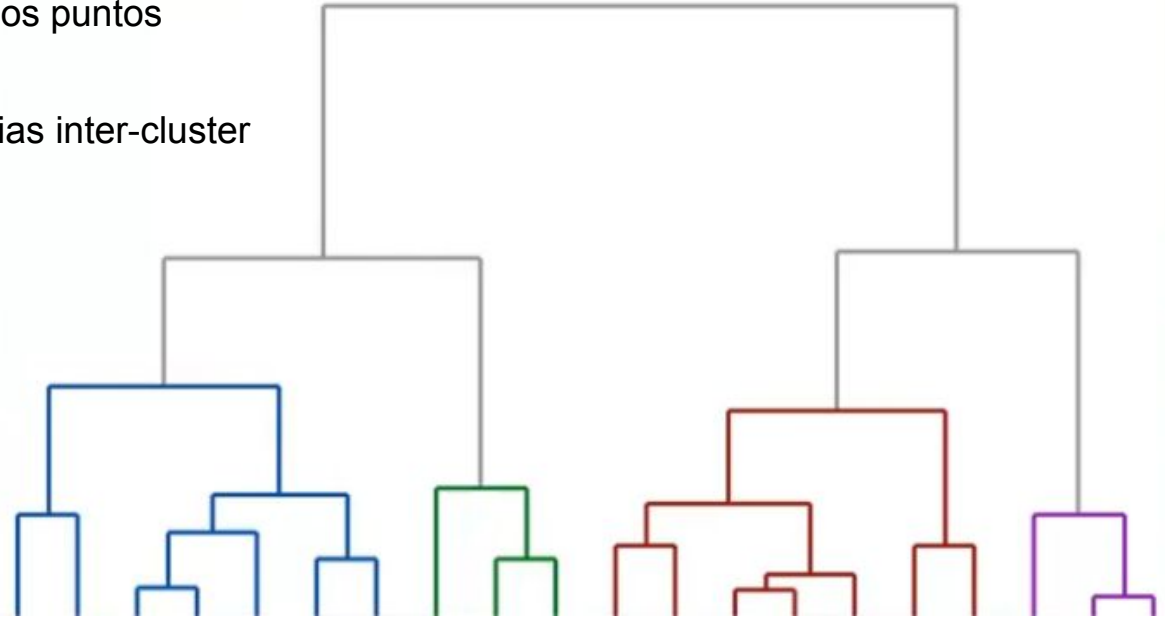


Clustering jerárquico - Dendrograma

El resultado de los algoritmos jerárquicos es un dendrograma:

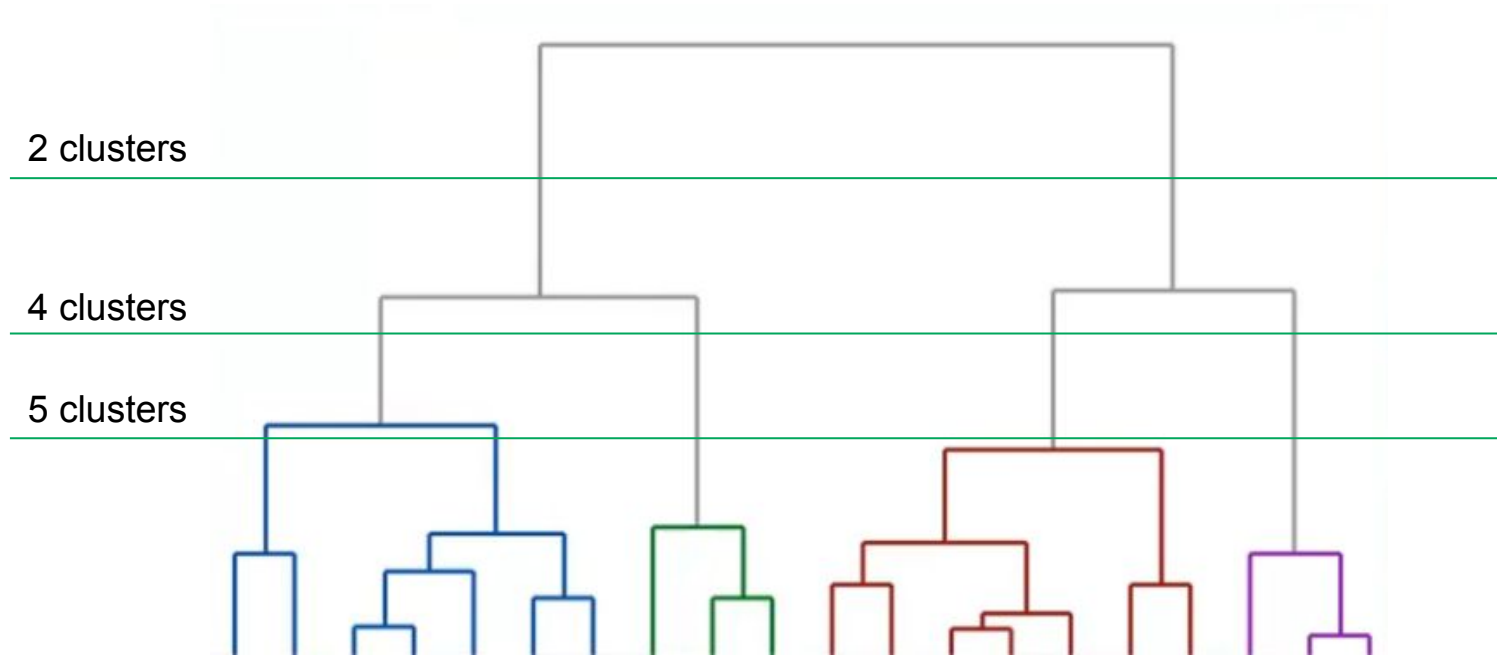
En el eje X se sitúa cada uno de los puntos

En el eje Y se marcan las distancias inter-cluster



Clustering jerárquico - Dendrograma

Se obtiene el clustering cortando el dendrograma a nivel deseado



Clustering jerárquico - Dendrograma

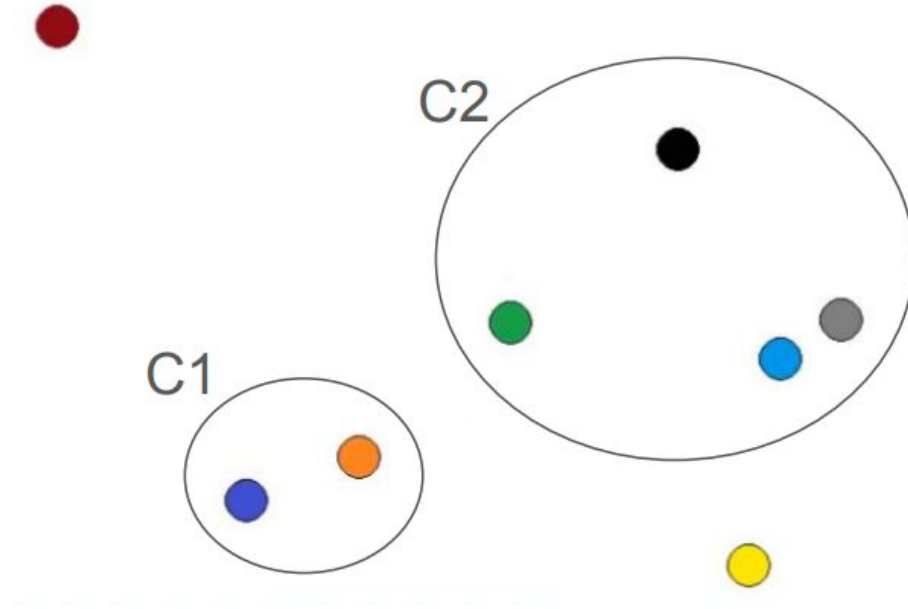
- Ventajas
 - Particiones múltiples: no es necesario especificar el número de clusters, puedo elegir el número de particiones a posteriori partiendo el dendrograma
 - El dendrograma da mucha información para interpretar los datos
- Inconvenientes
 - Es un algoritmo Greedy. No se pueden deshacer los pasos previos
- Es necesario definir cómo medimos las distancias inter-cluster

Clustering jerárquico - Algoritmo

- Inicialmente, cada punto forma un **singleton cluster**
- El objetivo es ir conectando clusters cuya **distancia sea mínima**
- Estrategias de enlazado (Linkage):
 - **Single-link**: distancia mínima entre puntos
 - **Average-link**: distancia media entre puntos
 - **Complete-link**: distancia máxima entre puntos
 - **Centroid-link**: distancia entre los centroides de los clusters
 - Otros

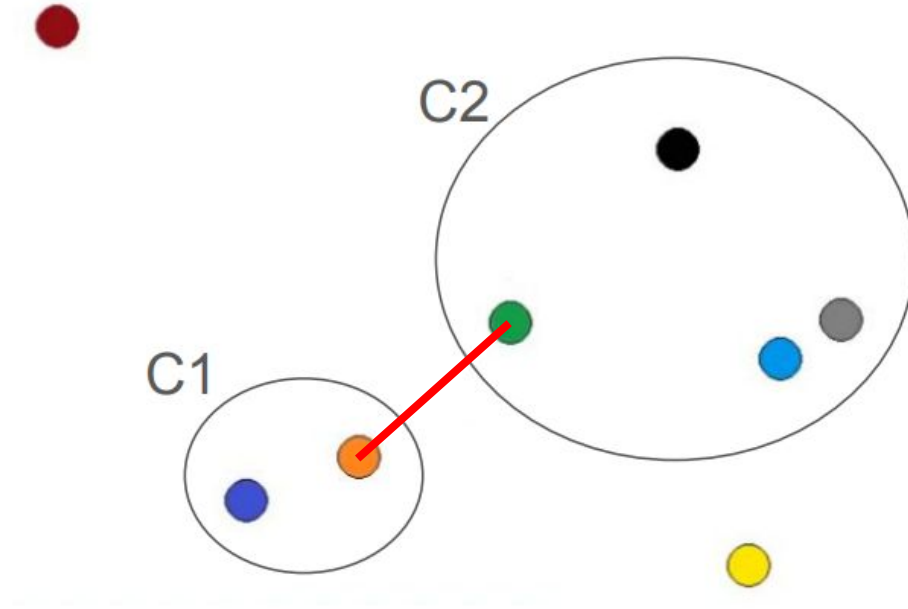
Clustering jerárquico - Linkage

- ¿Cómo defines la distancia entre los clusters C1 y C2?



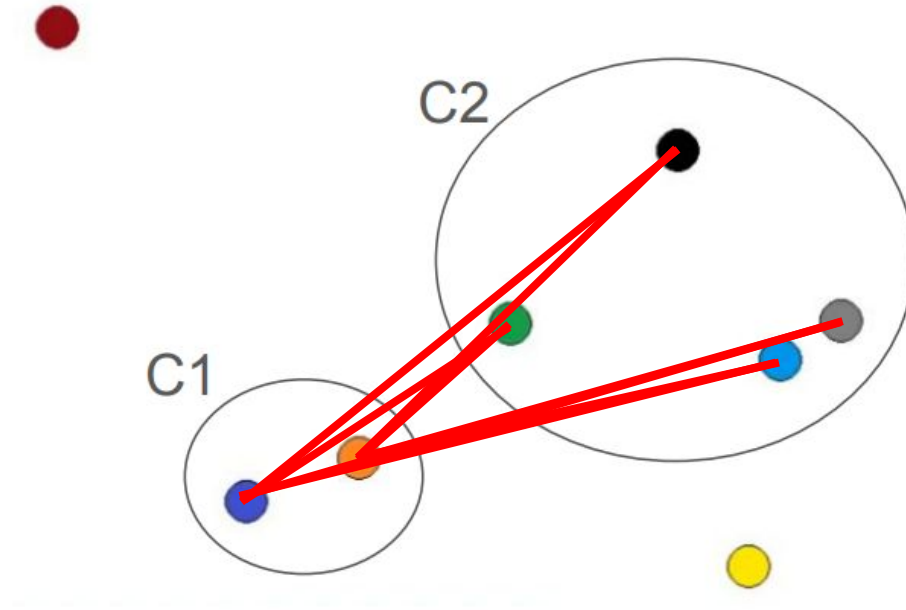
Clustering jerárquico - Linkage

- **Single Linkage**: distancia mínima: $\text{dist}(C1, C2).min()$



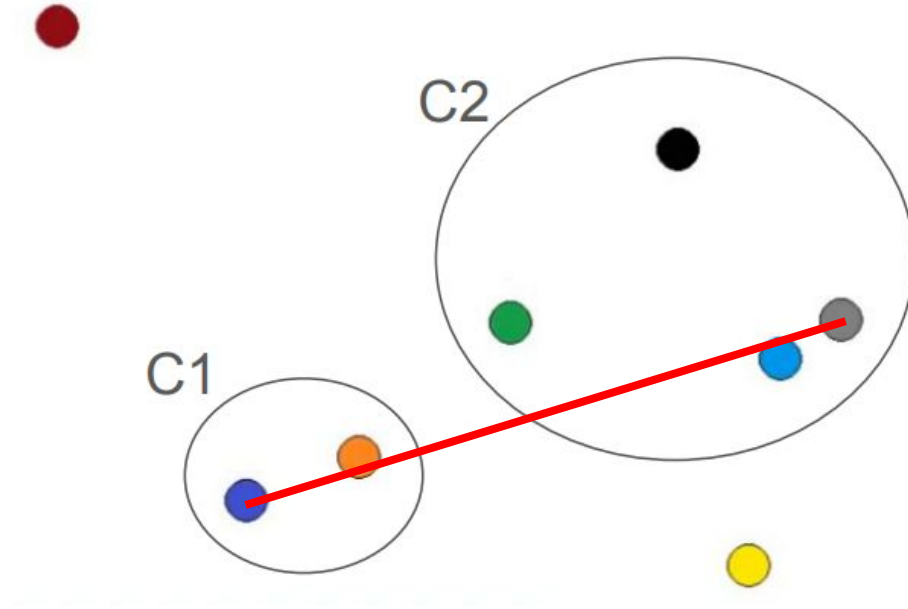
Clustering jerárquico - Linkage

- **Average Linkage**: distancia media: $\text{dist}(C1, C2).mean()$



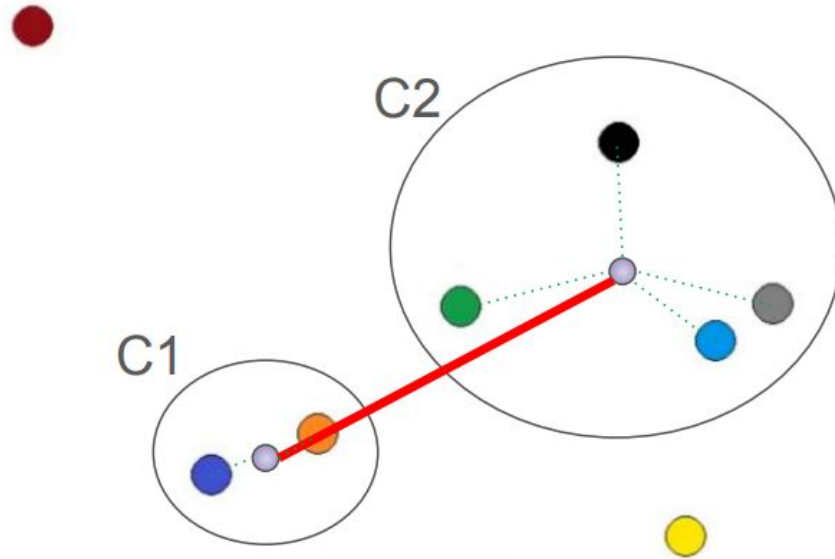
Clustering jerárquico - Linkage

- **Complete Linkage**: distancia máxima: $\text{dist}(C1, C2).max()$



Clustering jerárquico - Linkage

- **Centroid Linkage**: distancia entre centroides: `dist(C1.mean(), C2.mean())`



Clustering jerárquico - Algoritmo aglomerativo

- En el notebook [11_1_clustering_aglomerativo.ipynb](#) tenéis:
 - Una implementación manual del algoritmo aglomerativo
 - La explicación del uso de las librerías `linkage` y `fcluster` de `scipy.cluster.hierarchy`
 - Un ejemplo de clustering sobre los datos de MNIST

Clustering K-Means

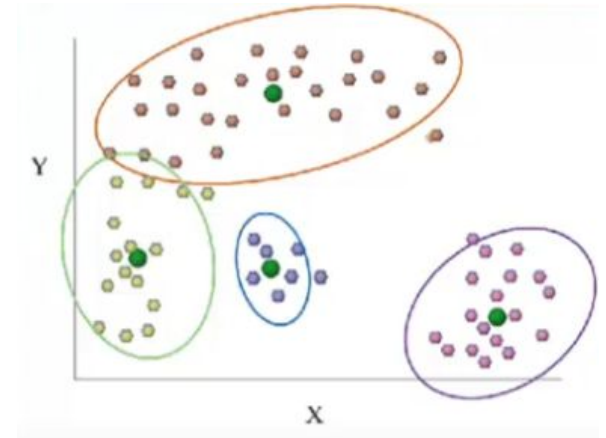
Basado en centroides

Clustering basado en centroides

- Es necesario definir cuántos grupos se van a formar (hiperparámetro K)
- Estos grupos se basan en la similitud a los **centroides**

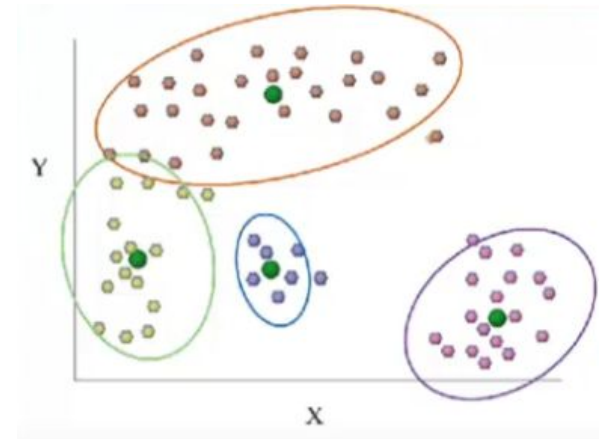
Como ya hemos visto, un centroide es el punto promedio de los puntos de un cluster

- El centroide puede representar al grupo correspondiente



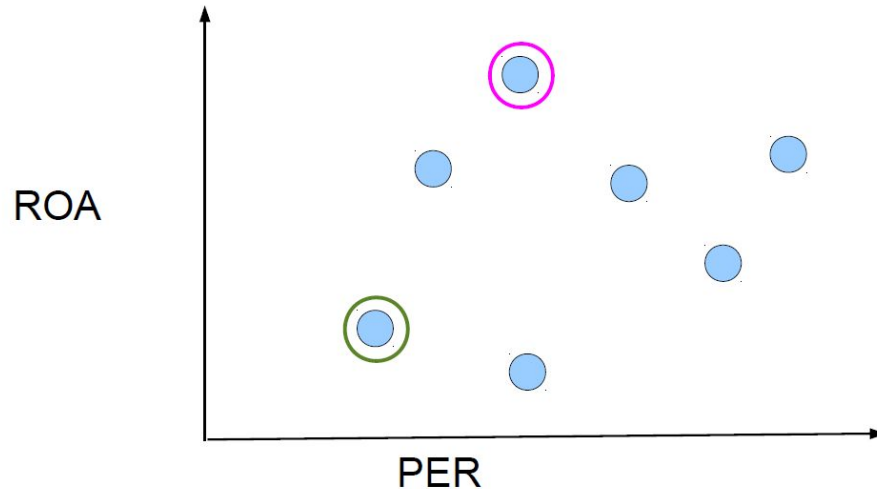
Clustering basado en centroides - K-means

- Se define previamente el valor K
- Se inicializan los centroides en K instancias del conjunto de entrenamiento
- Proceso iterativo: Mientras los centroides se vayan modificando
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters



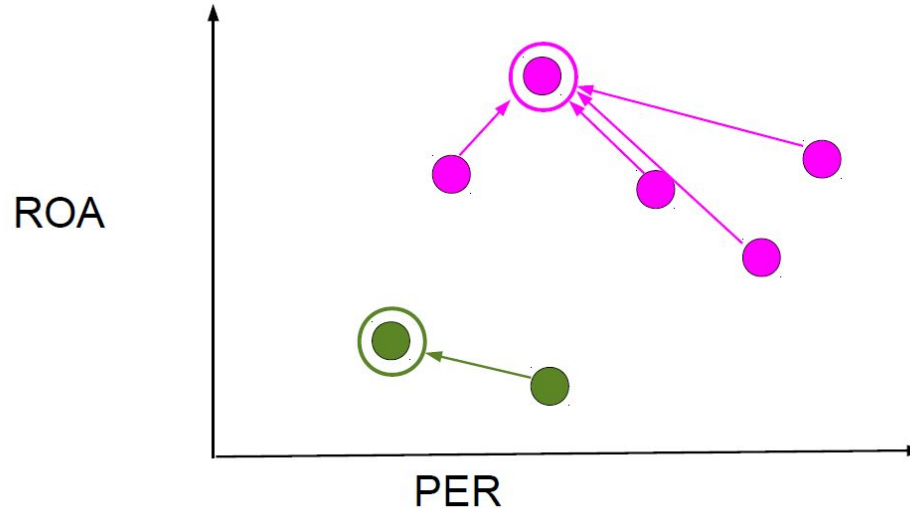
Clustering basado en centroides - K-means

- Se muestra un ejemplo:
 - Se define $K = 2$ como hiperparámetro
 - Se inicializan los centroides en $K=2$ instancias aleatorias del conjunto de entrenamiento



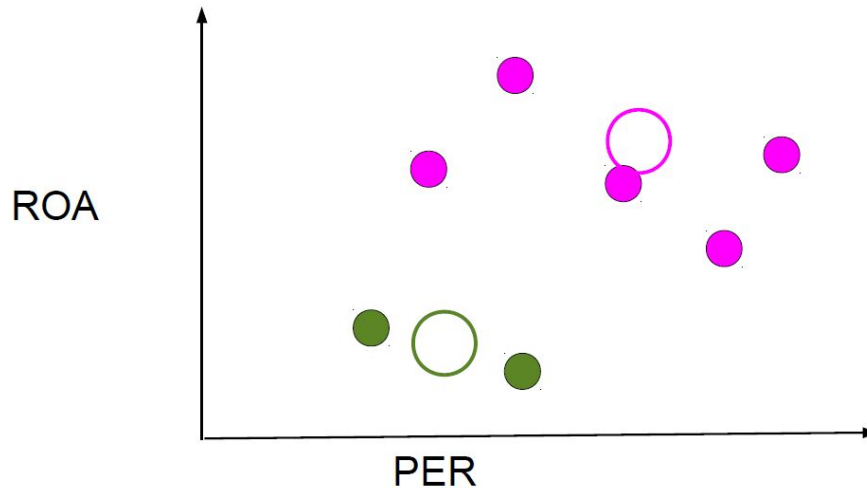
Clustering basado en centroides - K-means

- Mientras los centroides se vayan modificando:
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters



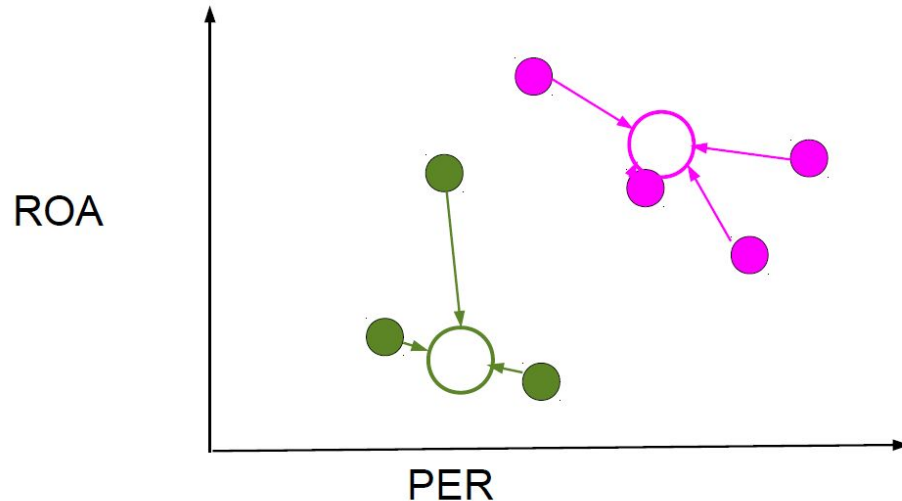
Clustering basado en centroides - K-means

- Mientras los centroides se vayan modificando:
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters



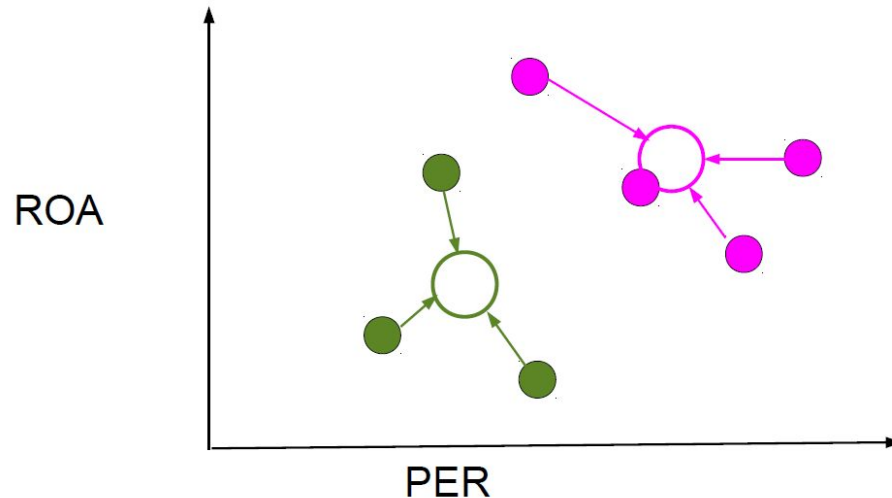
Clustering basado en centroides - K-means

- Mientras los centroides se vayan modificando:
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters



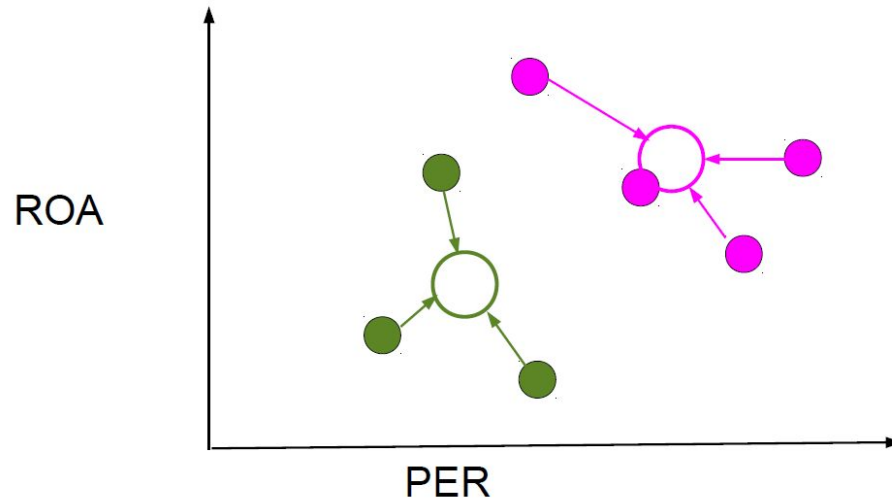
Clustering basado en centroides - K-means

- Mientras los centroides se vayan modificando:
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters



Clustering basado en centroides - K-means

- Mientras los centroides se vayan modificando:
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters

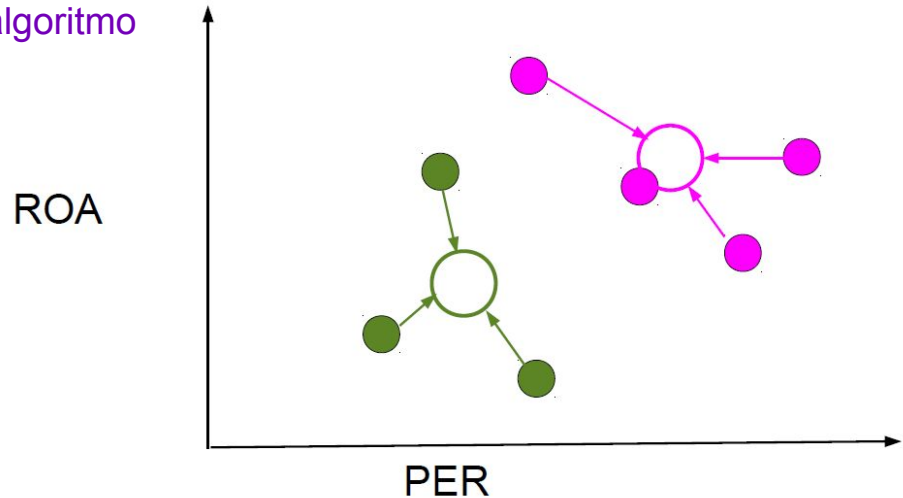


Clustering basado en centroides - K-means

- Mientras los centroides se vayan modificando:
 - **Asignación:** Asignar a cada instancia el cluster más cercano
 - **Actualización:** Calcular los centroides de los K clusters



Ya no se mueven los centroides. Fin del algoritmo



Clustering basado en centroides - K-means

- Ventajas:
 - Baja complejidad y fácil interpretación
 - Adaptabilidad a datos dispersos
- Inconvenientes:
 - Alta sensibilidad a la partición inicial
 - Alta sensibilidad a datos ruidosos y outliers
 - Válido solo para datos numéricos (hay que definir la media)
 - Necesario definir el valor K. No es trivial

Clustering basado en centroides - K-means

- Vamos al notebook [11_2_kmeans.ipynb](#)

En este notebook vamos a ver también un poco sobre proyección a 2D para visualizar

Clustering basado en centroides - K-medoids

- K-medoids es similar a K-means, pero los centroides se reemplazan por medoids

Un medoid es el ejemplo más cercano al centroide del cluster

- El algoritmo PAM (Partitioning Around Medoids) consiste en lo siguiente:
 - Se define previamente el valor K
 - Se inicializan los medoids en K instancias del conjunto de entrenamiento
 - Proceso iterativo: Mientras se mejore la función de coste durante la iteración
 - **Asignación:** Asignar a cada instancia el medoid más cercano
 - **Actualización:** Calcular el coste de intercambiar cada instancia con su medoid. El nuevo medoid será la instancia cuyo intercambio genera menor coste.

Clustering EM

Basado en mezcla de Gaussianas

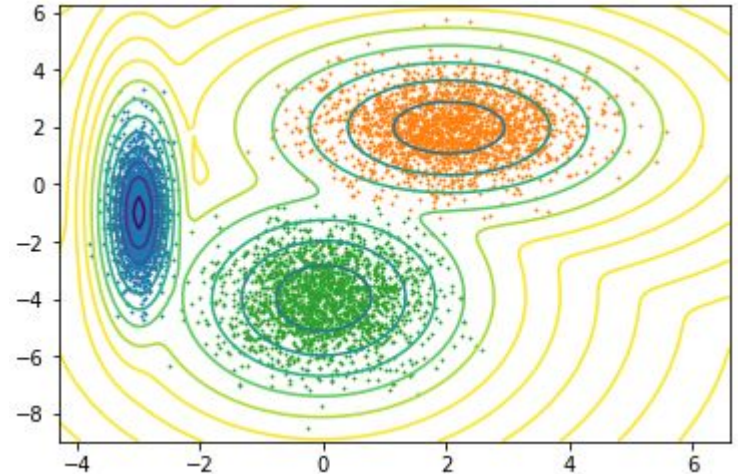
Clustering basado en mezcla de Gaussianas - EM

- Expectation-Maximization (EM) describe probabilísticamente los grupos

K-means asume que los datos se organizan en K grupos según los centroides ([medias](#))

EM asume K distribuciones estadísticas (Gaussianas: [medias y varianzas](#))

- Es un algoritmo SOFT, donde un punto pertenece a varios clusters con una probabilidad



Clustering basado en mezcla de Gaussianas - EM

- Teorema de Bayes

donde:

$$P(G_j|x_i) = \frac{P(x_i|G_j)P(G_j)}{P(x_i)}$$

- $P(G_j|x_i)$ es la probabilidad de que x_i pertenezca a $G_j \rightarrow$ Probabilidad a posteriori
- $P(G_j)$ es la probabilidad de pertenecer a la gaussiana $G_j \rightarrow$ Probabilidad a priori
- $P(x_i)$ es la probabilidad de seleccionar el elemento $x_i \rightarrow$ Evidencia
- $P(x_i|G_j)$ es la verosimilitud (likelihood), que sigue una distribución gaussiana:

$$P(x_i|G_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

Clustering basado en mezcla de Gaussianas - EM

$$P(G_j|x_i) = \frac{P(x_i|G_j)P(G_j)}{P(x_i)}$$

$$P(x_i|G_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

- Asumimos que:
 - $P(G_j)$ es la probabilidad de pertenecer a la gaussiana $G_j \rightarrow$ Probabilidad a priori
No conocemos los prioris $\rightarrow P(G_j) = 1 / K$ para todas las gaussianas.
 - $P(x_i)$ es la probabilidad de seleccionar el elemento $x_i \rightarrow$ Evidencia
La probabilidad de cada punto es igual para todos $\rightarrow P(x_i) = 1 / n$ para todos los puntos.

Clustering basado en mezcla de Gaussianas - EM

- Ejemplo:

$X = (x_1 = 3, x_2 = 2, x_3 = -3, x_4 = -2)$ y $K = 2$

$G_1 = (\mu_1 = 2.0, \sigma_1 = 1.0)$ y $G_2 = (\mu_2 = -2.0, \sigma_2 = 1.0)$

Calculamos los prioris y las evidencias:

$$P(G_j|x_i) = \frac{P(x_i|G_j)P(G_j)}{P(x_i)}$$

$$P(x_i|G_i) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

Clustering basado en mezcla de Gaussianas - EM

- Ejemplo:

$X = (x_1 = 3, x_2 = 2, x_3 = -3, x_4 = -2)$ y $K = 2$

$G_1 = (\mu_1 = 2.0, \sigma_1 = 1.0)$ y $G_2 = (\mu_2 = -2.0, \sigma_2 = 1.0)$

Calculamos los prioris y las evidencias:

$$P(G_1) = P(G_2) = 1 / 2 = 0.5$$

$$P(x_1) = P(x_2) = P(x_3) = P(x_4) = 1 / 4 = 0.25$$

Calculamos verosimilitudes G1:

Calculamos verosimilitudes G2:

$$P(G_j|x_i) = \frac{P(x_i|G_j)P(G_j)}{P(x_i)}$$

$$P(x_i|G_i) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

Clustering basado en mezcla de Gaussianas - EM

- Ejemplo:

$X = (x_1 = 3, x_2 = 2, x_3 = -3, x_4 = -2)$ y $K = 2$

$G_1 = (\mu_1 = 2.0, \sigma_1 = 1.0)$ y $G_2 = (\mu_2 = -2.0, \sigma_2 = 1.0)$

Calculamos los prioris y las evidencias:

$$P(G_1) = P(G_2) = 1 / 2 = 0.5$$

$$P(x_1) = P(x_2) = P(x_3) = P(x_4) = 1 / 4 = 0.25$$

Calculamos verosimilitudes G1:

$$P(x_1 | G_1) = \dots = 0.241971$$

$$P(x_2 | G_1) = \dots = 0.398942$$

$$P(x_3 | G_1) = \dots = 0.000001$$

$$P(x_4 | G_1) = \dots = 0.000134$$

Calculamos verosimilitudes G2:

$$P(x_1 | G_2) = \dots = 0.000001$$

$$P(x_2 | G_2) = \dots = 0.000134$$

$$P(x_3 | G_2) = \dots = 0.241971$$

$$P(x_4 | G_2) = \dots = 0.398942$$

$$P(G_j | x_i) = \frac{P(x_i | G_j)P(G_j)}{P(x_i)}$$

$$P(x_i | G_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

Calculamos las probabilidades:

Clustering basado en mezcla de Gaussianas - EM

- Ejemplo:

$X = (x_1 = 3, x_2 = 2, x_3 = -3, x_4 = -2)$ y $K = 2$

$G_1 = (\mu_1 = 2.0, \sigma_1 = 1.0)$ y $G_2 = (\mu_2 = -2.0, \sigma_2 = 1.0)$

Calculamos los prioris y las evidencias:

$$P(G_1) = P(G_2) = 1 / 2 = 0.5$$

$$P(x_1) = P(x_2) = P(x_3) = P(x_4) = 1 / 4 = 0.25$$

Calculamos verosimilitudes G1:

$$P(x_1 | G_1) = \dots = 0.241971$$

$$P(x_2 | G_1) = \dots = 0.398942$$

$$P(x_3 | G_1) = \dots = 0.000001$$

$$P(x_4 | G_1) = \dots = 0.000134$$

Calculamos verosimilitudes G2:

$$P(x_1 | G_2) = \dots = 0.000001$$

$$P(x_2 | G_2) = \dots = 0.000134$$

$$P(x_3 | G_2) = \dots = 0.241971$$

$$P(x_4 | G_2) = \dots = 0.398942$$

$$P(G_j | x_i) = \frac{P(x_i | G_j)P(G_j)}{P(x_i)}$$

$$P(x_i | G_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

Calculamos las probabilidades:

$$P(G_1 | x_1) = \dots = 0.483941$$

$$P(G_1 | x_2) = \dots = 0.797884$$

$$P(G_1 | x_3) = \dots = 0.000003$$

$$P(G_1 | x_4) = \dots = 0.000268$$

$$P(G_2 | x_1) = \dots = 0.000003$$

$$P(G_2 | x_2) = \dots = 0.000268$$

$$P(G_2 | x_3) = \dots = 0.483941$$

$$P(G_2 | x_4) = \dots = 0.797884$$

Clustering basado en mezcla de Gaussianas - EM

- Ejemplo:
 - Asignamos al cluster con mayor probabilidad:

$$P(G_1 | x_1) = \dots = 0.483941 \rightarrow 0.999999$$

$$P(G_2 | x_1) = \dots = 0.000003 \rightarrow 0.000001$$

$$P(G_1 | x_2) = \dots = 0.797884 \rightarrow 0.999664$$

$$P(G_2 | x_2) = \dots = 0.000268 \rightarrow 0.000336$$

$$P(G_1 | x_3) = \dots = 0.000003 \rightarrow 0.000001$$

$$P(G_2 | x_3) = \dots = 0.483941 \rightarrow 0.999999$$

$$P(G_1 | x_4) = \dots = 0.000268 \rightarrow 0.000336$$

$$P(G_2 | x_4) = \dots = 0.797884 \rightarrow 0.999664$$

Clustering basado en mezcla de Gaussianas - EM

- Algoritmo EM:
 - Inicializar de forma aleatoria las gaussianas. Por ejemplo:
Los K puntos de partida son puntos del conjunto de datos de entrenamiento
Se calculan las K medias y varianzas de los datos más cercanos a cada punto de partida
 - Mientras no converja el algoritmo:
 - Paso 1: Expectation (asignación a grupos)
 - Paso 2: Maximization (actualización de medias y varianzas)

Clustering basado en mezcla de Gaussianas - EM

- **Paso 1: Expectation:** Asignar cada elemento x_i al cluster S_j cuya verosimilitud sea máxima:

$$P(x_i|G_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right)$$

$$\theta_{ij} = P(G_j|x_i) = \frac{P(x_i|G_j)P(G_j)}{P(x_i)}$$

$$S_j = S_j \cup \{ \argMax(\theta_{ij}) \text{ for each } j \text{ in range}(K) \}$$

Clustering basado en mezcla de Gaussianas - EM

- **Paso 2: Maximization:** Recalcular los parámetros para cada Gaussiana G_j

$$\mu_j^{(t+1)} = \frac{\sum \theta_{ij}^{(t)} x_i}{\sum \theta_{ij}^{(t)}} \quad \sigma_j^{2:(t+1)} = \frac{\sum \theta_{ij}^{(t)} (x_i - \mu_j^{(t+1)})^2}{\sum \theta_{ij}^{(t)}}$$

Clustering basado en mezcla de Gaussianas - EM

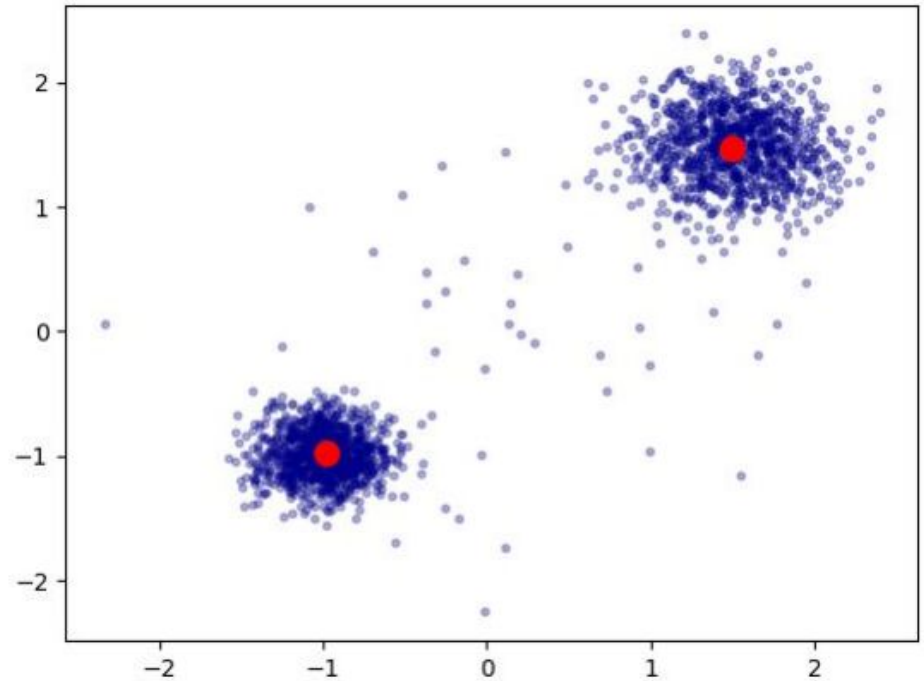
- Notebook [11_3_clustering_em.ipynb](#)

Clustering DBScan

Basado en densidades

Particional vs Densidad

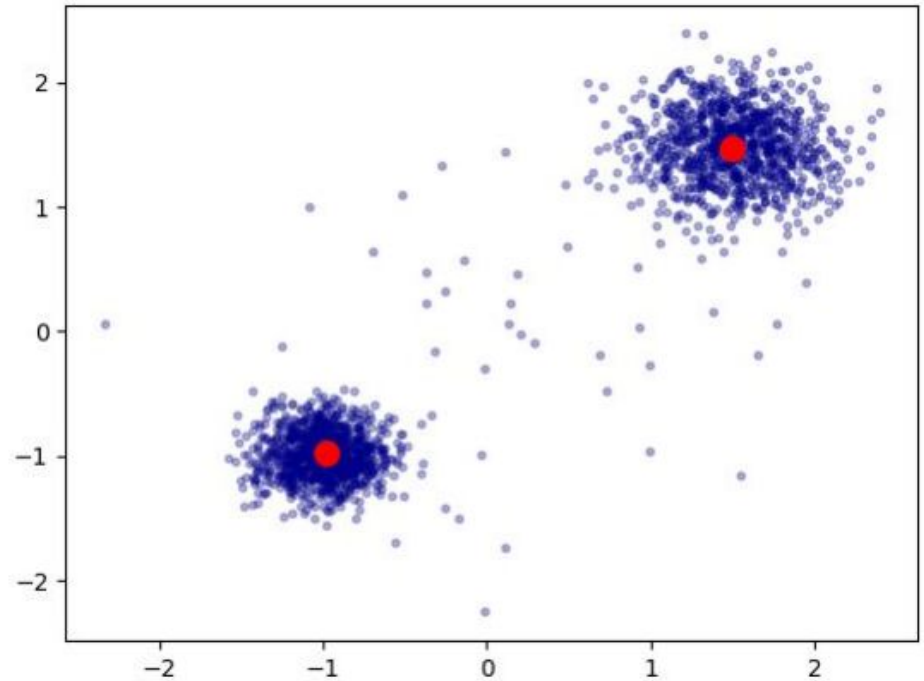
- ¿Qué es clustering?
- ¿Esto es clustering?



Particional vs Densidad

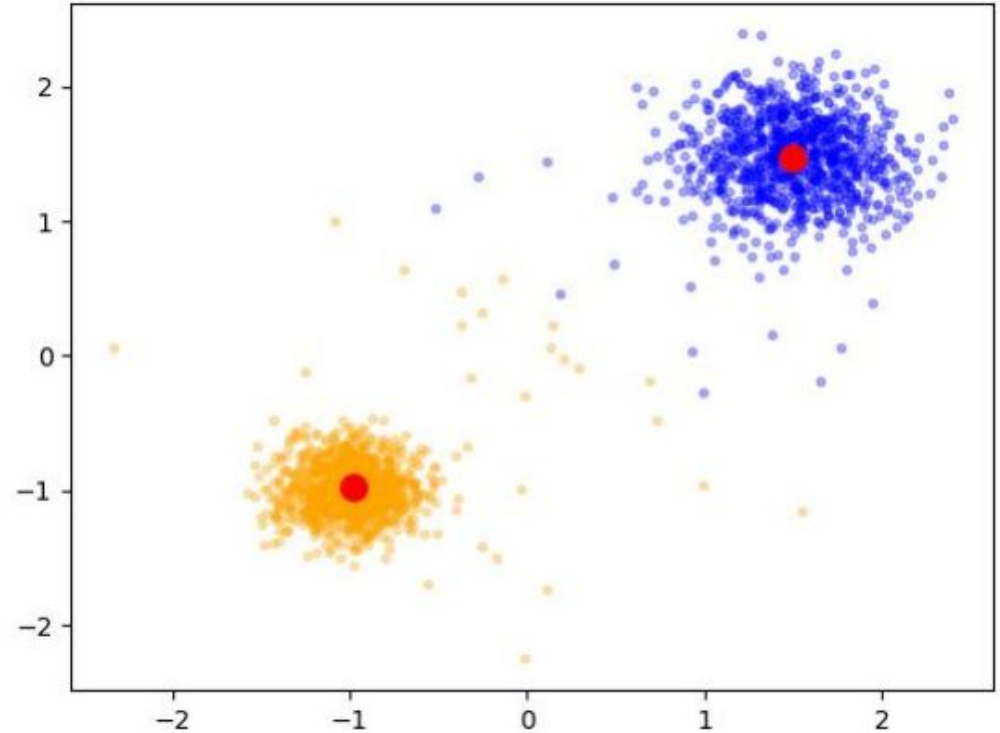
- ¿Qué es clustering?
- ¿Esto es clustering?

Sí, ya que encuentro dos grupos



Particional vs Densidad

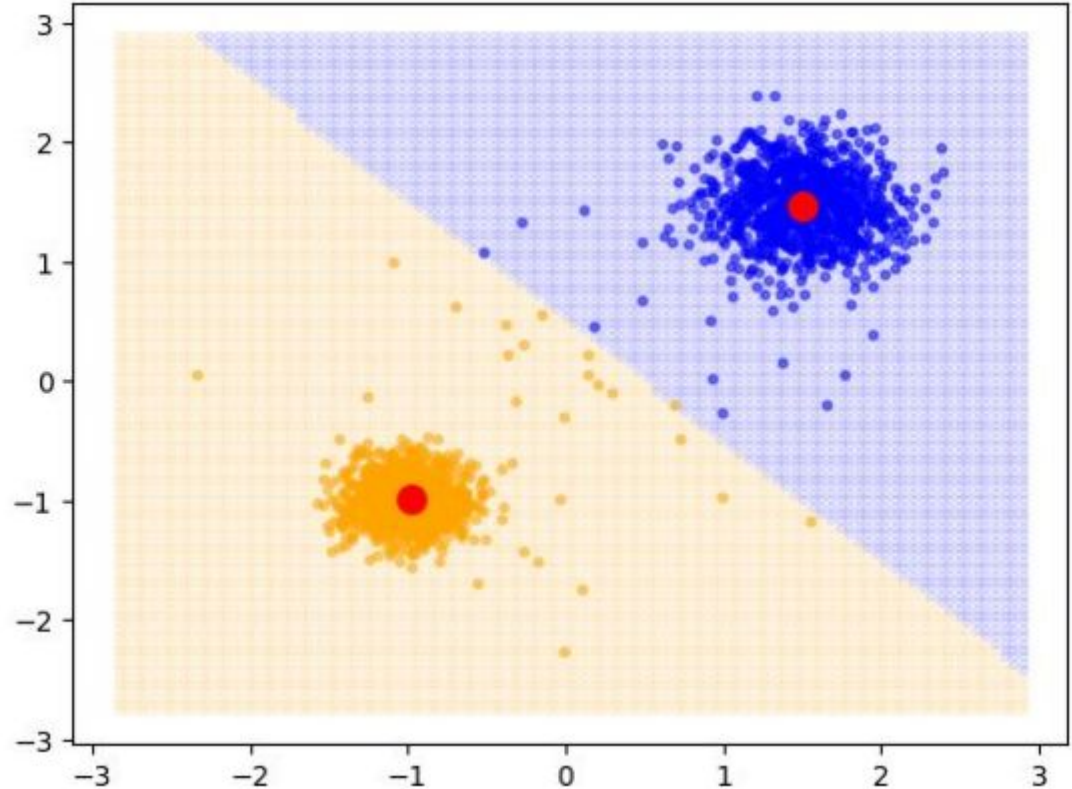
- ¿Qué es clustering?
- ¿Estoy haciendo clustering?



Particional vs Densidad

- ¿Qué es clustering?
- ¿Estoy haciendo clustering?

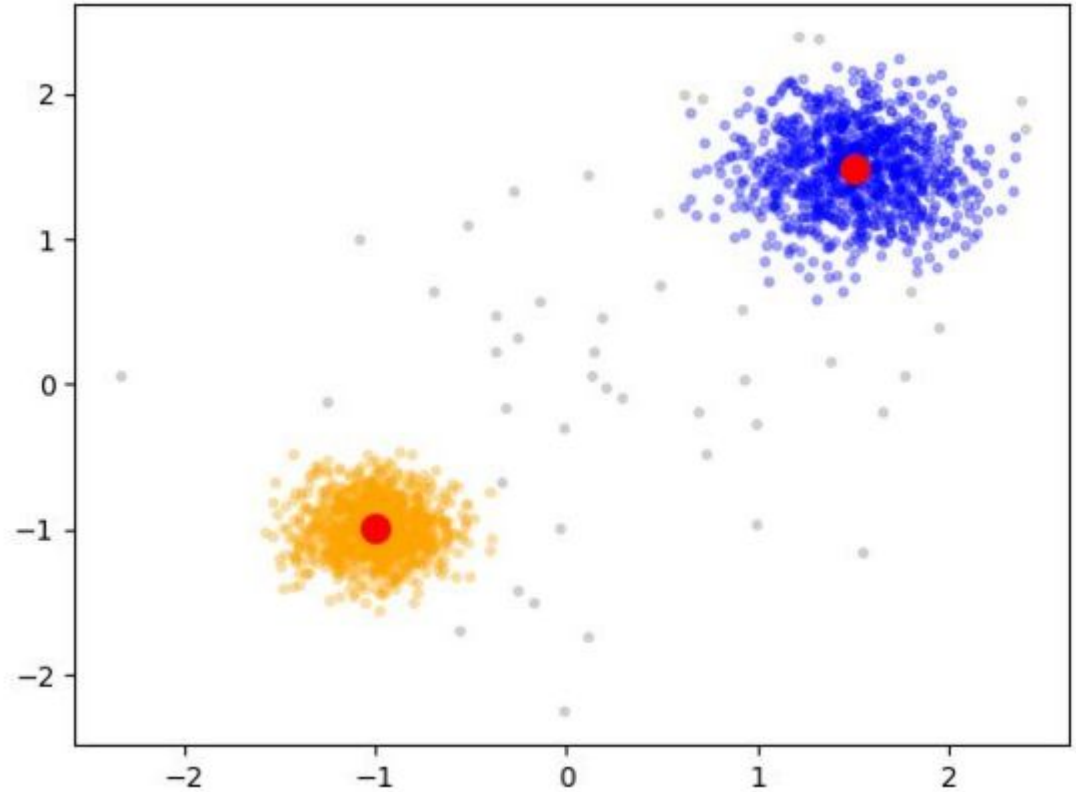
DEPENDE



Particional vs Densidad

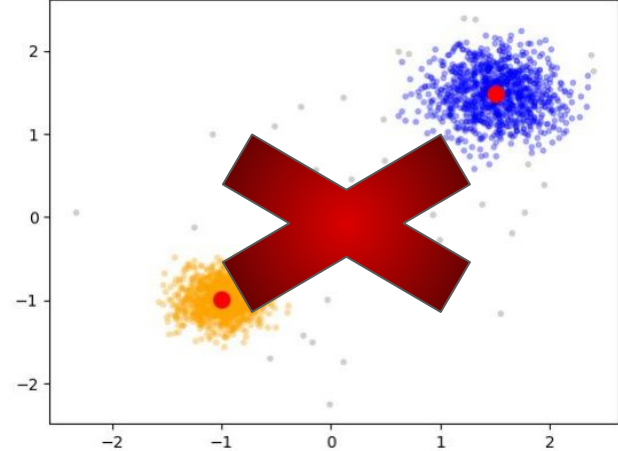
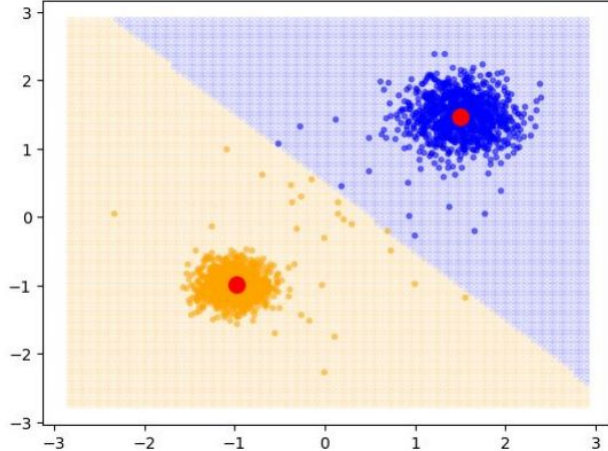
- ¿Qué es clustering?
- ¿Estoy haciendo clustering?

DEPENDE



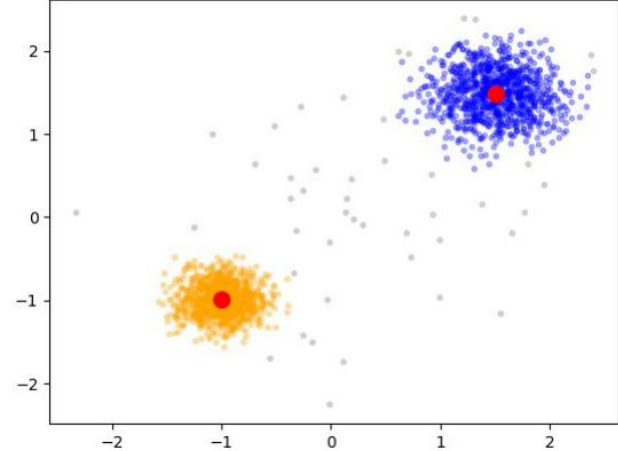
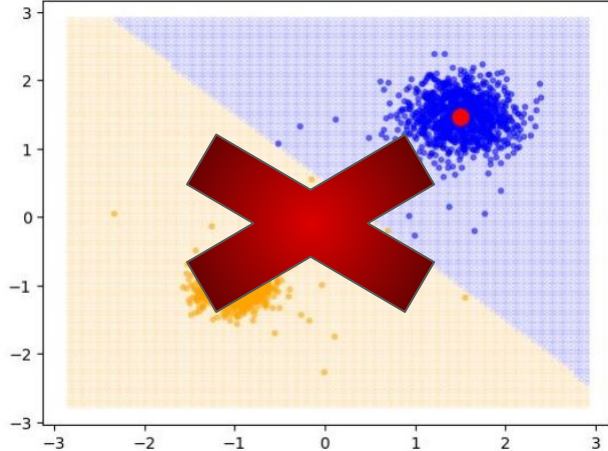
Particional vs Densidad

- ¿Qué es clustering?
 - Si es una colección de patrones basados en una similitud
 - Si es una región en el espacio con una alta densidad de puntos



Particional vs Densidad

- ¿Qué es clustering?
 - Si es una colección de patrones basados en una similitud
 - Si es una **región en el espacio con una alta densidad de puntos**



Particional vs Densidad

- Los algoritmos particionales (K-means, EM) dividen el espacio en regiones separadas
 - Definiendo K clusters
 - Definiendo una métrica de similitud

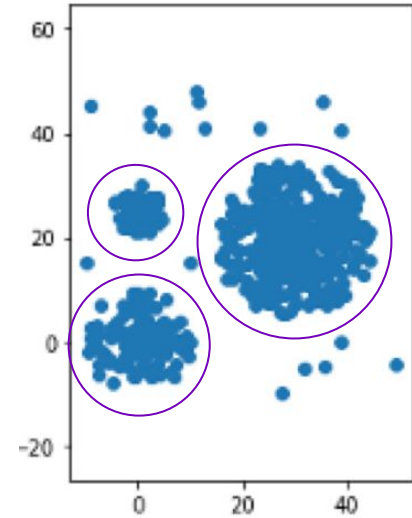
Asumen que todos los puntos pertenecen al cluster más cercano

- Los algoritmos basados en densidades (DBSCAN) identifica zonas muy pobladas
 - Definiendo una métrica de similitud

Puede haber puntos que no pertenecen a ningún cluster

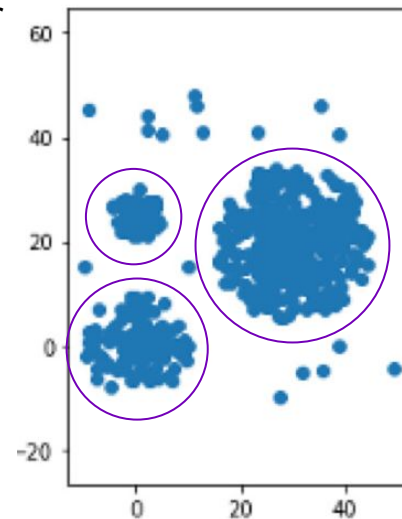
Clustering basado en densidades - DBSCAN

- DBSCAN surge de un método intuitivo de agrupamiento humano
- Al mirar la figura, cualquiera identifica fácilmente tres grupos junto con varios puntos ruidosos
- ¿Qué estamos observando?
 - La densidad de los puntos, es decir, cuán agrupados están los puntos



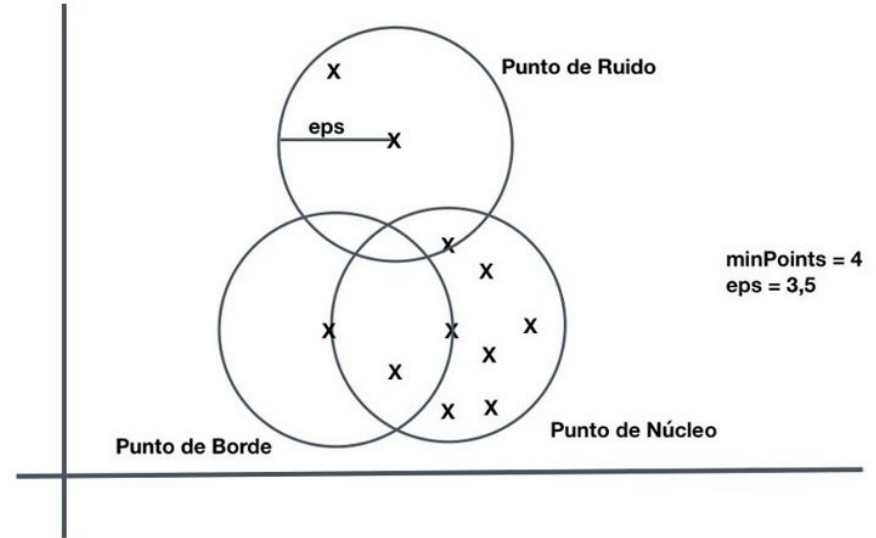
Clustering basado en densidades - DBSCAN

- DBSCAN no necesita definir el hiperparámetro K, en contraposición necesitamos definir:
 - **Épsilon (eps)**: especifica lo cerca que deben estar los puntos entre sí para ser considerados parte de un cluster. Si la distancia entre dos puntos es menor o igual que eps, esos dos puntos son vecinos.
 - **Puntos mínimos (minPts)**: el número de puntos necesarios para formar una región densa.



Clustering basado en densidades - DBSCAN

- Con los valores de eps y minPts, tenemos tres tipologías de puntos:
 - **Puntos de núcleo:** Tiene más de un número especificado de puntos minPts en su radio.
 - **Puntos de borde:** Tiene menos de un número especificado de puntos minPts pero es vecino de un punto de núcleo.
 - **Puntos de ruido:** Todos los demás.

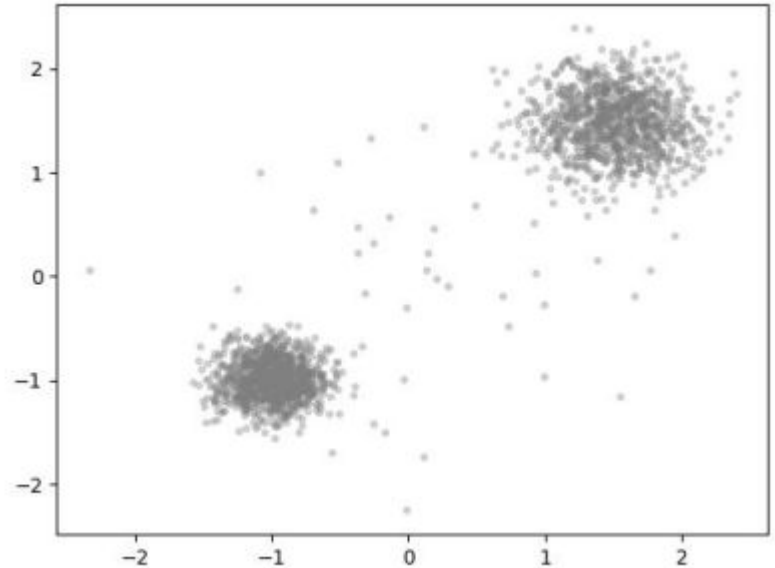


Clustering basado en densidades - DBSCAN

- Algoritmo DBSCAN:
 - **Selecciona un punto arbitrario** que no haya sido previamente visitado. Se calculan las distancias a todos y la información de su vecindario se recupera según el valor de ϵ .
 - **Si es punto de núcleo** (contiene minPts o más puntos en el vecindario):
 - Todos los puntos del vecindario **forman un cluster**, además de todos los vecinos de aquellos vecinos que son también puntos de núcleo.
 - **Si no:**
 - Se etiqueta como **punto de ruido**. Este punto podrá ser modificado más adelante si se encuentra en el vecindario de otro punto de núcleo (era punto de borde).

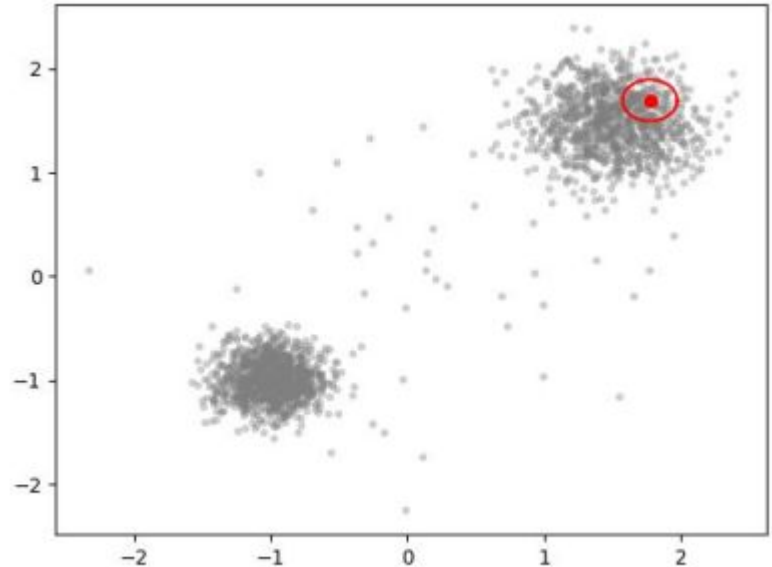
Clustering basado en densidades - DBSCAN

- Ejemplo de ejecución
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



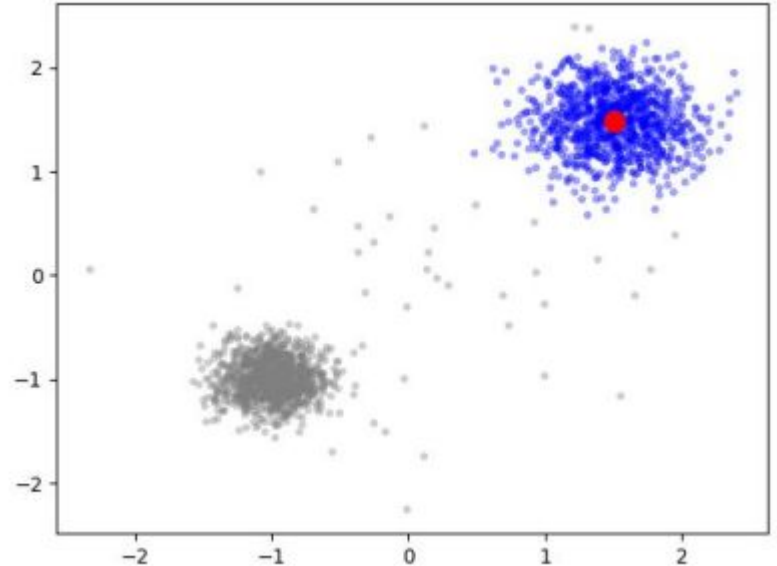
Clustering basado en densidades - DBSCAN

- Selecciona un punto al azar y traza el círculo con radio eps:
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



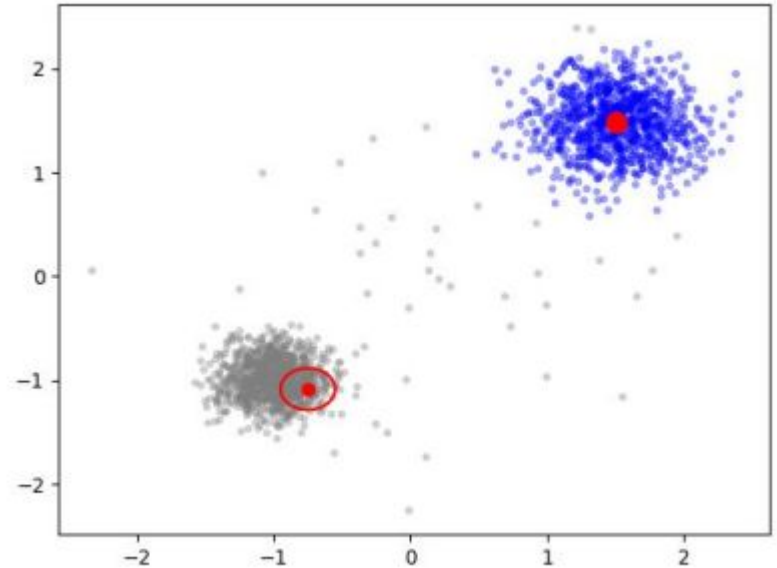
Clustering basado en densidades - DBSCAN

- Como es un **punto de núcleo**, es decir, hay más de minPts dentro del radio → **De forma recursiva, todos los vecinos forman un cluster**
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



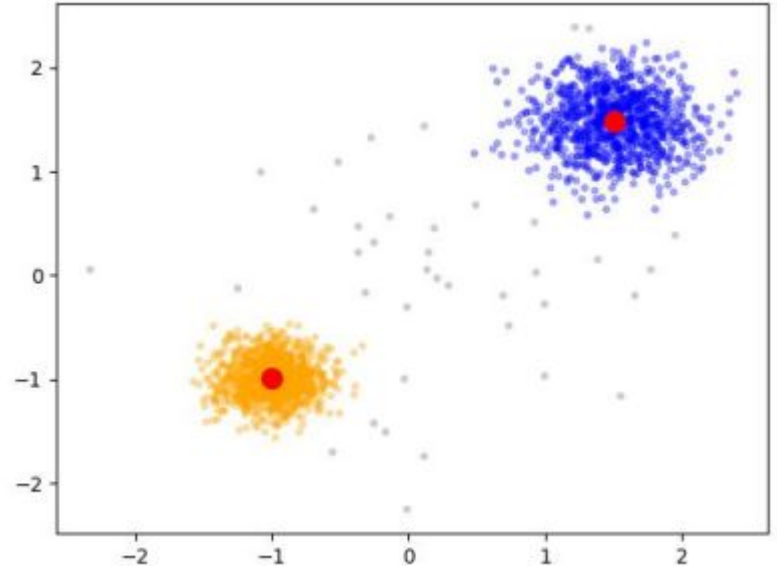
Clustering basado en densidades - DBSCAN

- Selecciona un punto al azar y traza el círculo con radio eps:
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



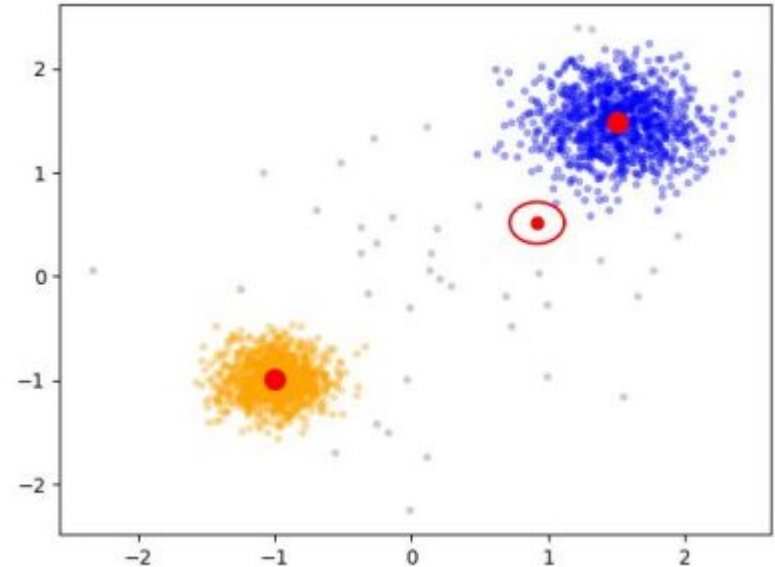
Clustering basado en densidades - DBSCAN

- Como es un **punto de núcleo**, es decir, hay más de minPts dentro del radio → **De forma recursiva, todos los vecinos forman un cluster**
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



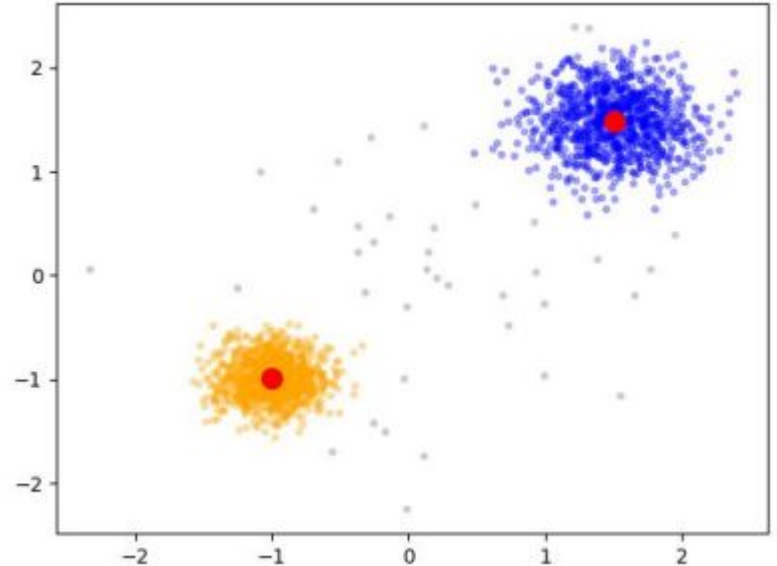
Clustering basado en densidades - DBSCAN

- Selecciona un punto al azar y traza el círculo con radio eps:
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



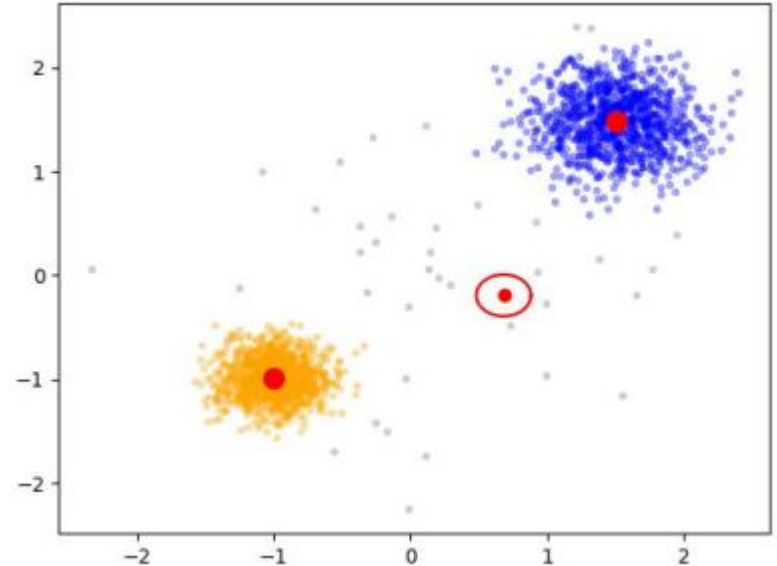
Clustering basado en densidades - DBSCAN

- Como es un **punto de ruido**, es decir, hay menos de minPts dentro del radio → **no hago nada**
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



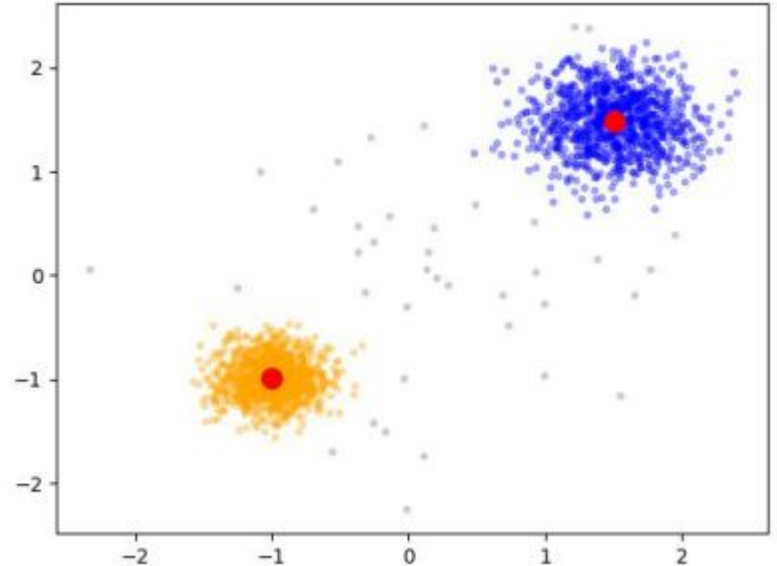
Clustering basado en densidades - DBSCAN

- Selecciona un punto al azar y traza el círculo con radio eps:
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



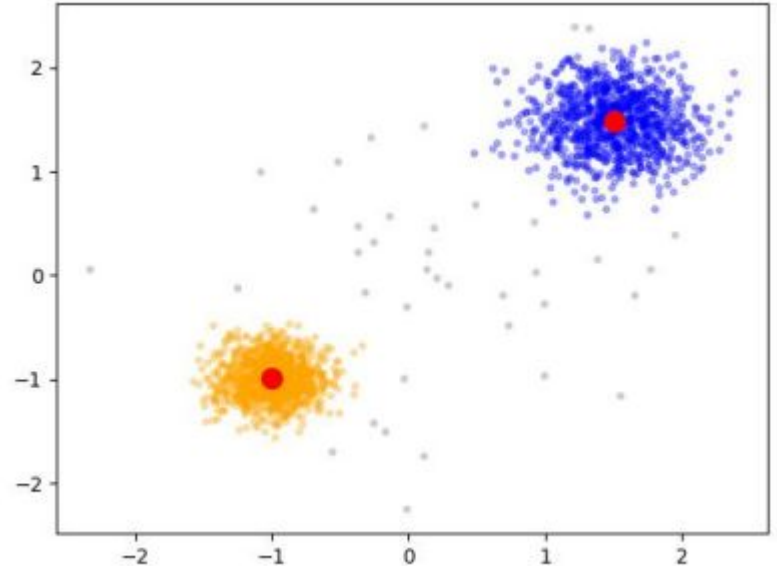
Clustering basado en densidades - DBSCAN

- Como es un **punto de ruido**, es decir, hay menos de minPts dentro del radio → **no hago nada**
 - $\text{eps} = 0.2$
 - $\text{minPts} = 10$



Clustering basado en densidades - DBSCAN

- Así sucesivamente hasta que todos los puntos sin color han sido visitados.
- Se acaba el algoritmo y tenemos 2 clusters



Clustering basado en densidades - DBSCAN

- Ventajas:
 - No requiere el valor K
 - Puede encontrar cualquier forma de cluster
 - Discrimina muy bien el ruido y los valores atípicos
 - Es excelente para separar clusters de alta densidad frente a otros
 - Es visualmente atractivo e intuitivo
- Inconvenientes
 - Es sensible a los hiperparámetros ϵ y minPts

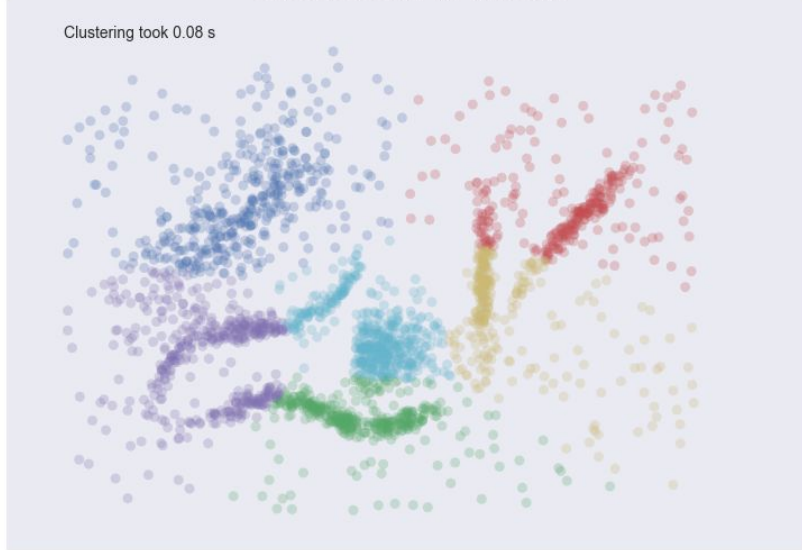
Clustering basado en densidades - DBSCAN

- En el notebook [11_4_DBSCAN.ipynb](#) tenéis:
 - Ejemplo manual de DBSCAN utilizado en las diapositivas
 - Ejemplo de uso de DBSCAN con MNIST
 - Ejemplo de implementación “más compleja” con MNIST

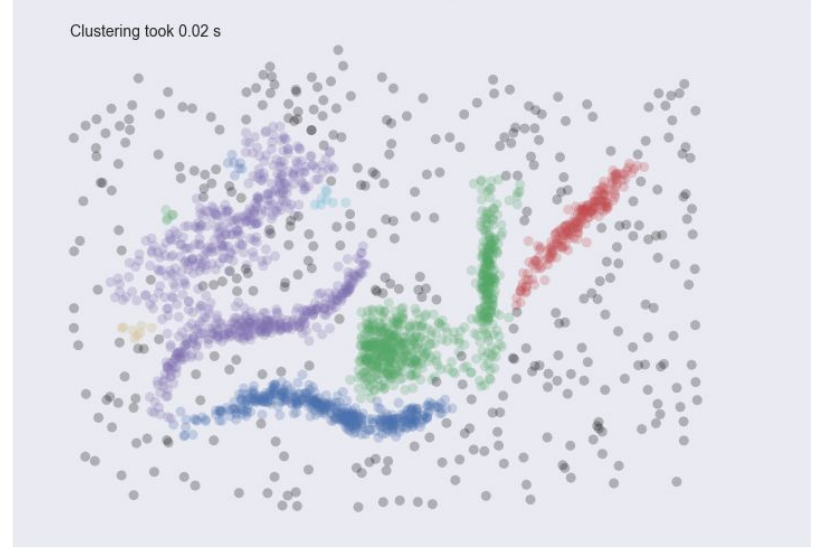
Clustering basado en densidades - K-means vs DBSCAN

- Comparación K-means vs DBSCAN

Clusters found by KMeans



Clusters found by DBSCAN



Clustering - Consideraciones finales

- Cuidado con la alta dimensionalidad en clustering. Cada nuevo atributo hace que los elementos estén más alejados. Las medidas de distancias acaban siendo inútiles
- Alternativas:
 - Aplicar un algoritmo de reducción de dimensionalidad
 - Selección de características (Correlación, Permutación, etc)
 - Transformación de características (PCA, Autoencoders, etc)

Clustering - Consideraciones finales

- ¿Cómo comparar el resultado de diferentes algoritmos de clustering?
- Podemos medir las siguientes propiedades:
 - **Cohesión intra-cluster**: cómo de cerca están los puntos de un mismo cluster
 - **Separación inter-cluster**: cómo de lejos están unos clusters de otros
- Ejemplo con K-means:
 - ¿Cómo de cerca están los puntos de un cluster a su centroide?
 - ¿Cómo de lejos están los centroides unos de otros?

Clustering - Consideraciones finales

- Validación Silhouette: medimos la similitud entre los miembros de un cluster comparado con la similitud a otros clusters:
 - $a(i)$: promedio de la distancia de x_i a los miembros de su propio cluster
 - $b(i)$: promedio de la distancia de x_i a los miembros del cluster diferente más cercano

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

- **Coeficiente de Silhouette**: Media de $s(i)$ para todas las instancias

Clustering - Consideraciones finales

- En el notebook [11_5_comparacion_clustering.ipynb](#) tenéis:
 - Dos ejercicios para hacer en clase probando dos problemas sintéticos
 - Selección del mejor modelo con coeficiente de silhouette

Ejemplos de clustering en problemas financieros

- Clustering de activos
 - Mediante ventanas de momentum
 - ¿Y si...?
- Clustering de fondos de inversión mediante atributos cuantitativos
 - Incluye una sección de XAI usando árboles para explicar la clusterización

No-Code Data Analytics

- Graphext es una herramienta online (con versión gratuita) No-Code para data análisis.
- Incluye lo siguiente:
 - Visualización de atributos
 - Implica la aplicación de un algoritmo de reducción de dimensionalidad no lineal (UMAP)
 - Implica la aplicación de un algoritmo de clustering (algoritmo Louvain)
 - Implica la aplicación de un meta-modelo para regresión y clasificación (CatBoost)

<https://www.graphext.com/>