Machine Learning

Clustering

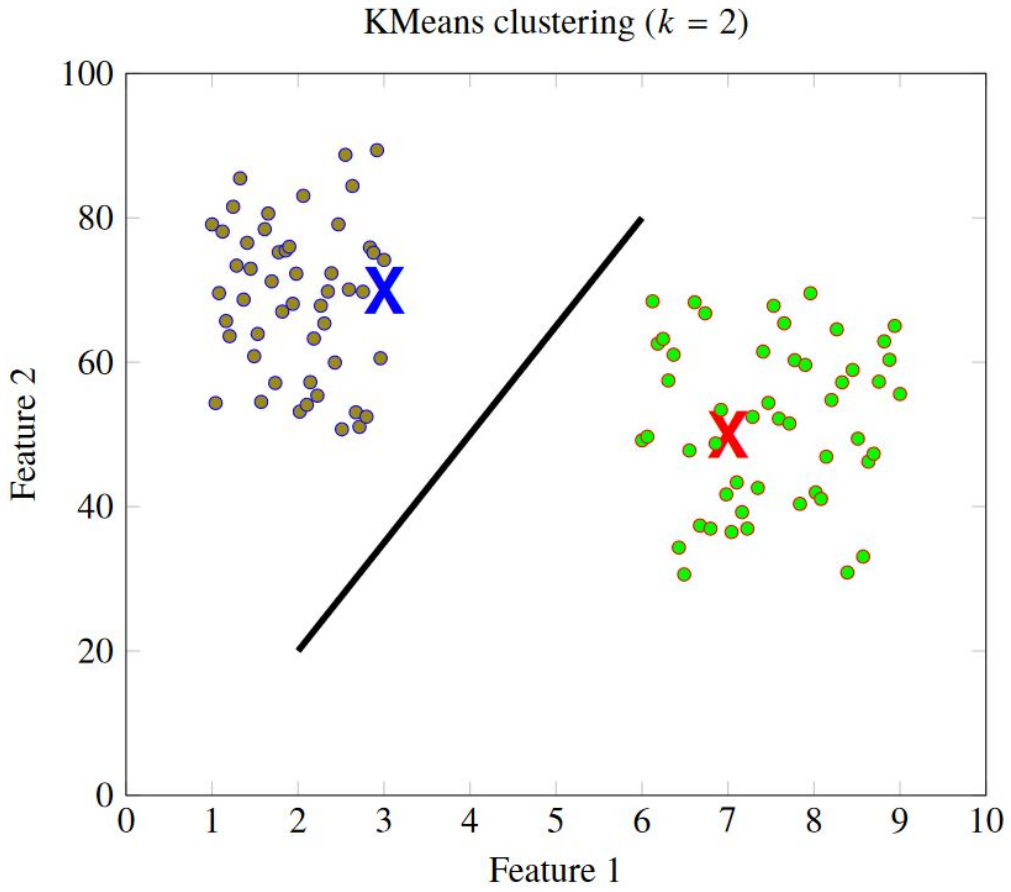
Christian Oliva Moya
Luis Fernando Lago Fernández

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

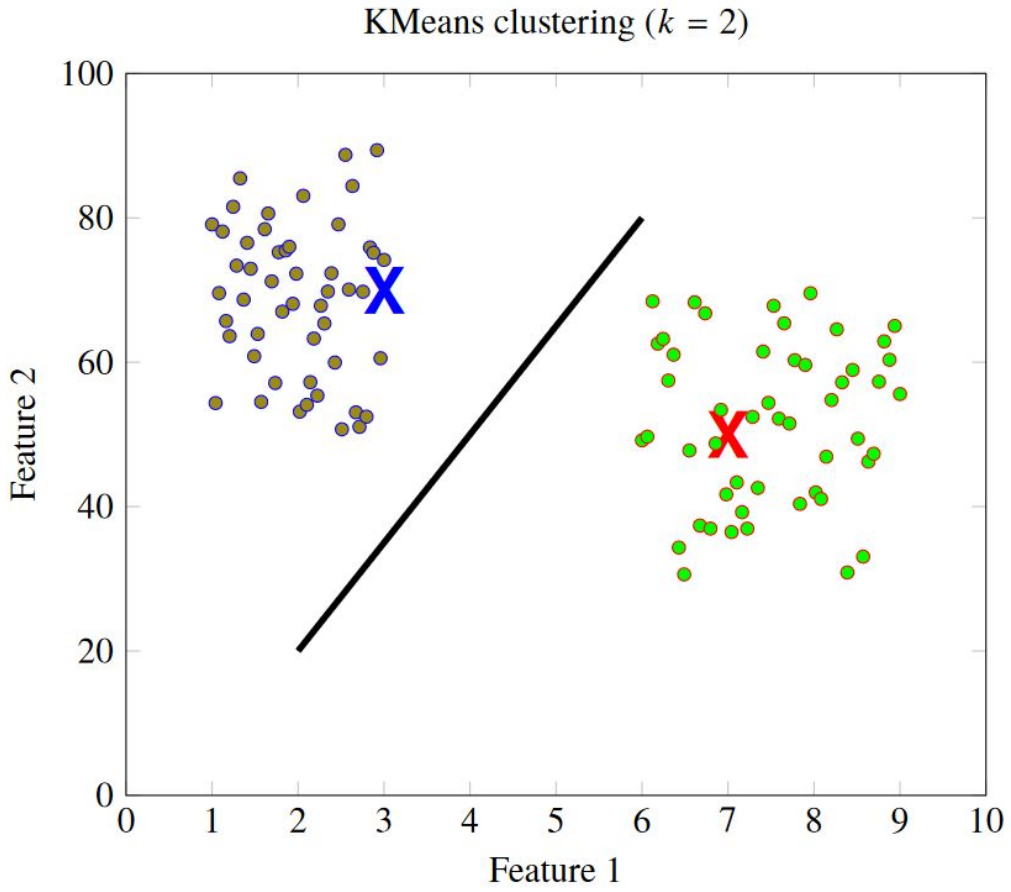
¿Esto es clustering?



Introducción

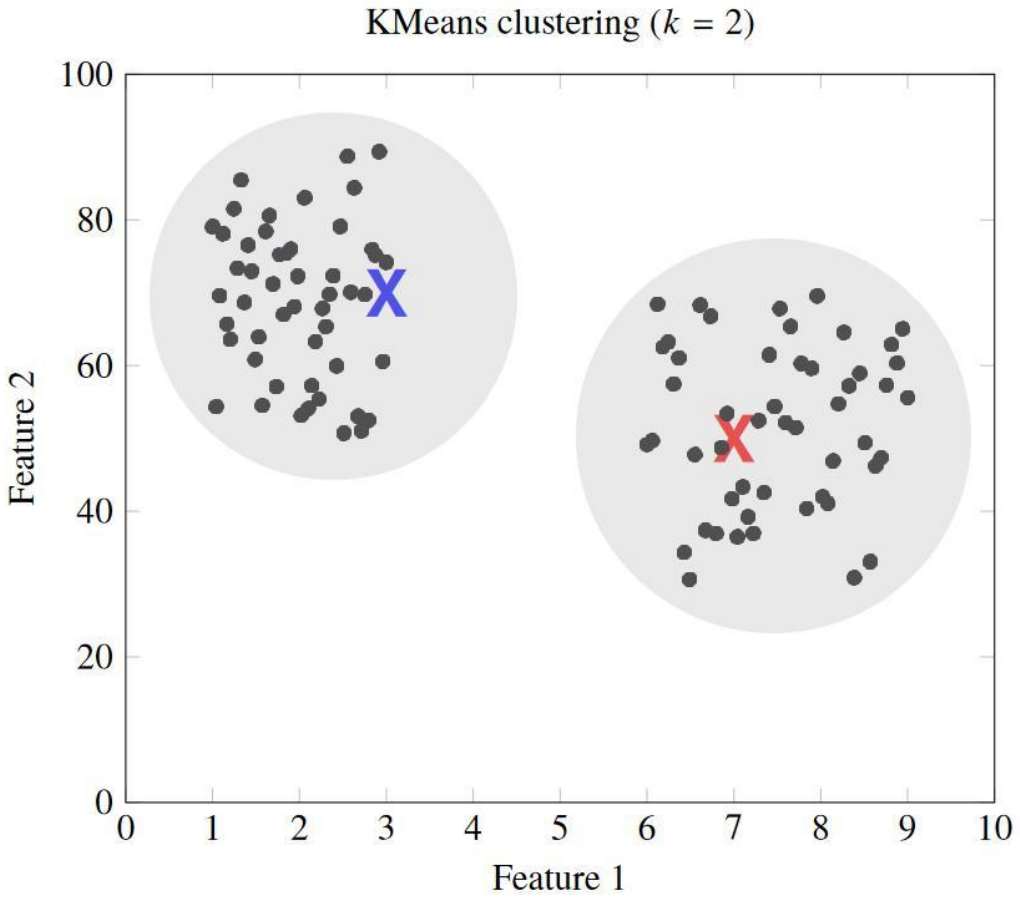
¿Esto es clustering?

NO!! Es un clasificador lineal



Introducción

Esto SÍ es clustering



Introducción - Recordatorio

Clasificación	**Clustering**
Los datos están etiquetados (supervisado)	No hay etiquetas (no supervisado)
	Los nuevos datos se agrupan según una métrica de proximidad
Objetivo:	Objetivo:
Etiquetar correctamente datos no vistos	Identificar estructuras/patrones en los datos

Introducción - Aplicaciones

- Construcción de carteras:

  Se identifican grupos de activos distintos

- Réplica de un índice bursátil:

  Agrupar componentes según una serie de indicadores de rentabilidad y volatilidad

  Elegir un representante de cada cluster

- Identificación de cambios estructurales del mercado:

  Agrupar la evolución de ratios de cotización entre activos

  Salir del cluster se interpreta como un posible cambio a estudiar

Introducción - Objetivos del clustering

- Escalabilidad

  La complejidad temporal y espacial debe estar acotada incluso con grandes cantidades de datos

- Robustez

  Los outliers deben detectarse con precisión. El modelo no debe verse afectado por datos ruidosos

- 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

Definition ($d(x_i, x_j)$)

The distance between two instances x_i and x_j, which is a metric distance measure if it satifies the following properties:

- ✓ *Triangle inequality*

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j), \ \forall x_i, x_j, x_k \in \mathcal{S}$$

- ✓ $d(x_i, x_j) = 0 \rightarrow x_i = x_j \ \forall x_i, x_j \in \mathcal{S}$

Introducción - Distancia

- Distancia Minkowski

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

Si g = 2 ⇒ Euclidean

Si g = 1 ⇒ Manhattan

Introducción - ¿Qué vamos a ver?

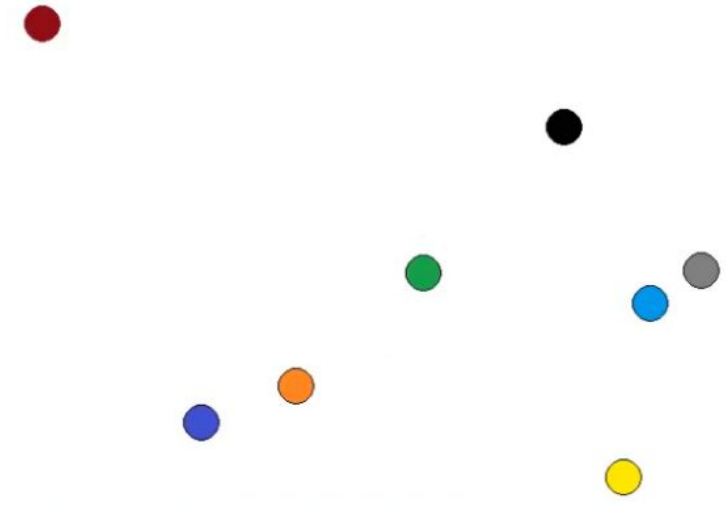
- Clustering jerárquico (o de conectividad)
  - Aglomerativo
- Clustering basado en centroides
  - K-means y K-medoids
- Clustering basado en mezcla de gaussianas
  - Expectation Maximization
- Clustering basado en densidades
  - DBSCAN

Clustering jerárquico

- Clustering jerárquico (o de conectividad):

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

  Veamos un ejemplo:

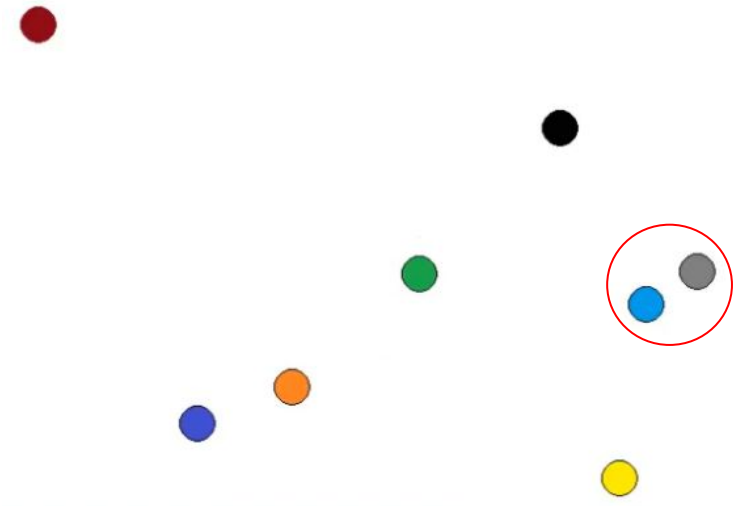


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

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

  Paso 1:

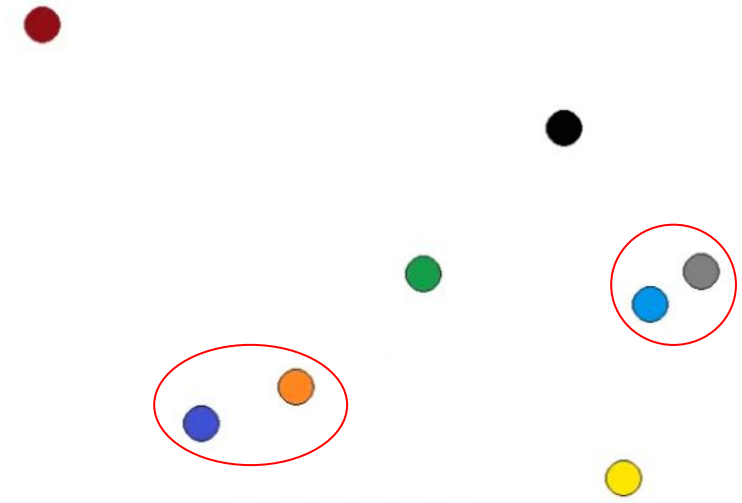


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

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

  Paso 2:

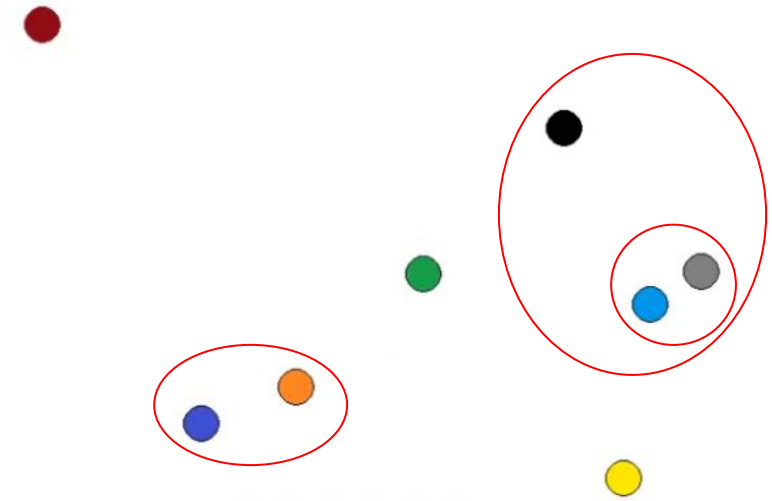


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

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

  Paso 3:

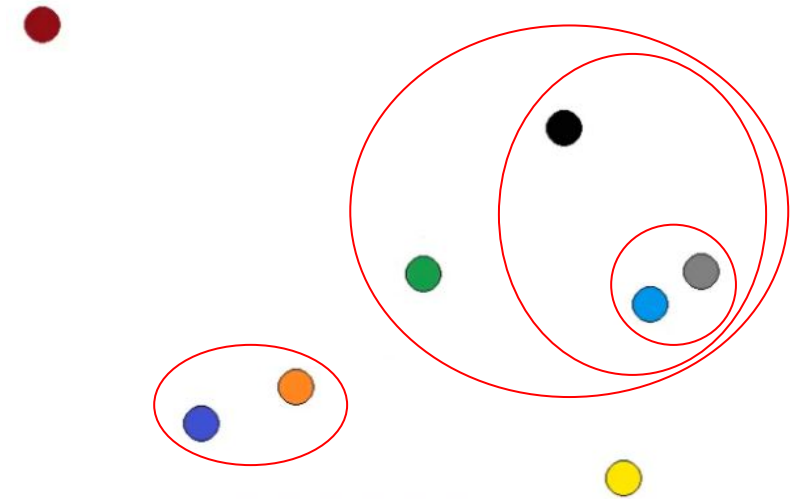


Clustering jerárquico

- Clustering jerárquico (o de conectividad):

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

  Paso 4:

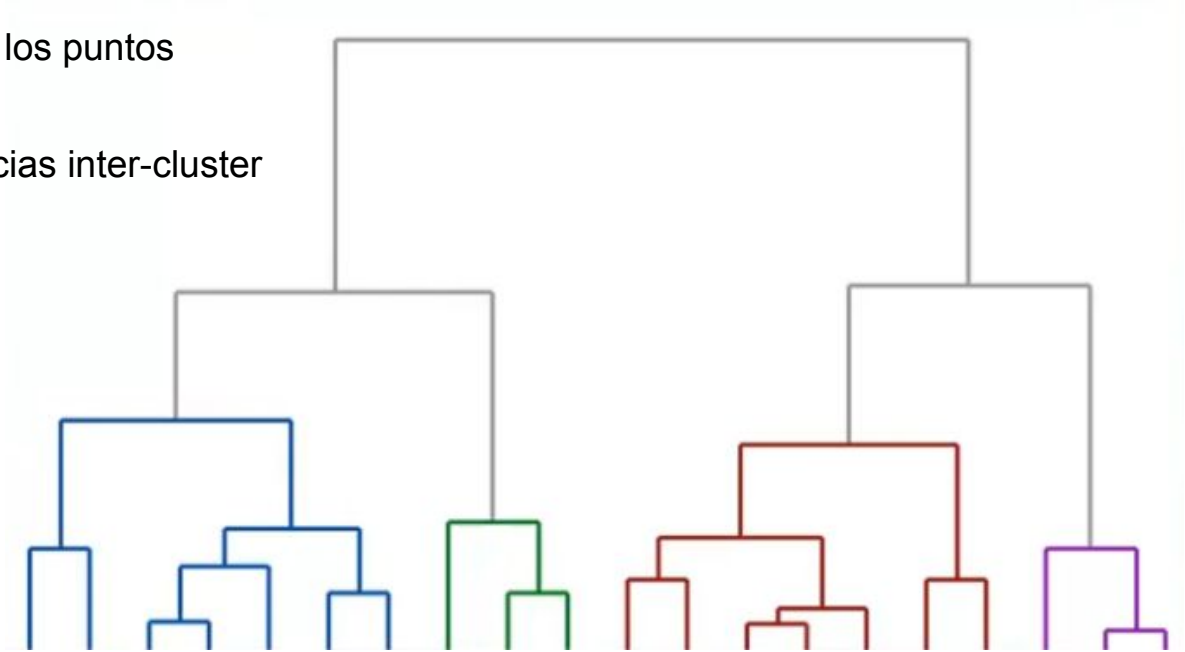


Clustering jerárquico - Dendrograma

El resultado de los algoritmos jerárquicos es un dendrograma:

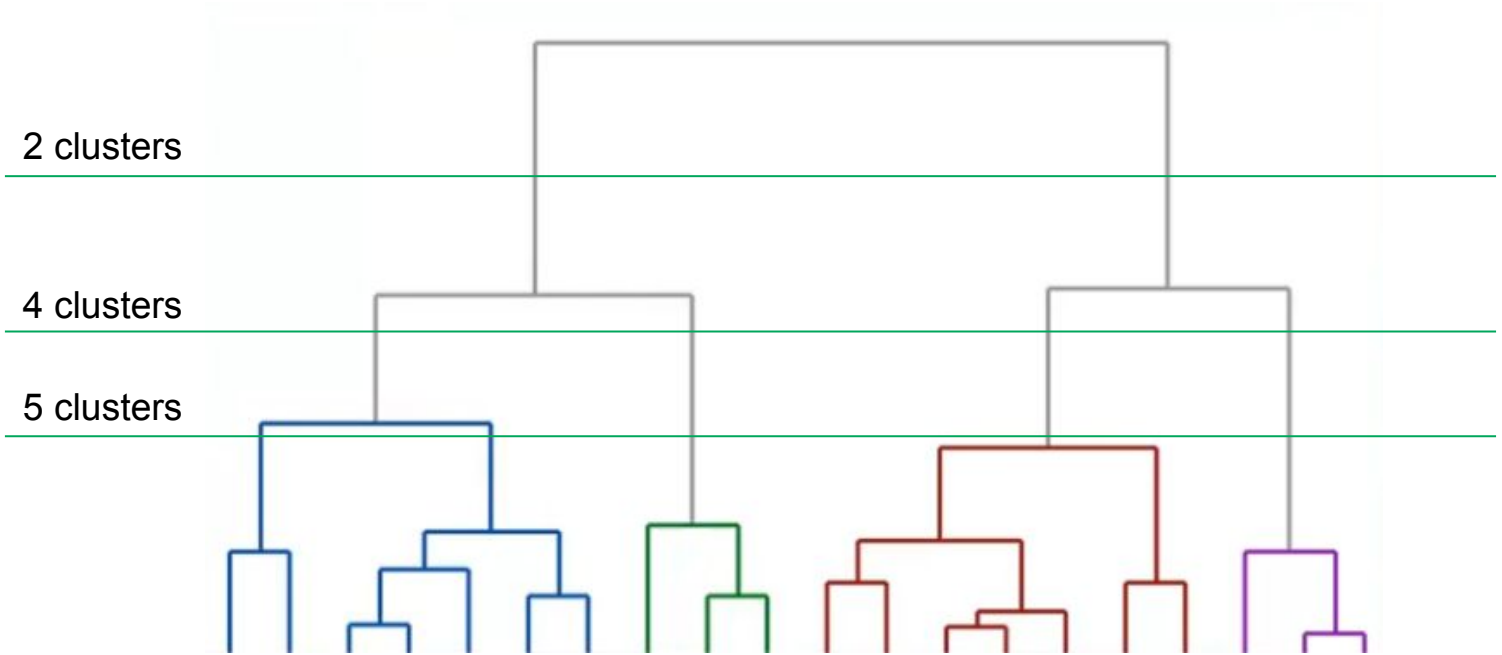
En el eje X se sitúa cada uno de los puntos

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 - Distancia inter-cluster

- Single-link clustering

  La menor distancia posible entre puntos de cada cluster.

- Complete-link clustering

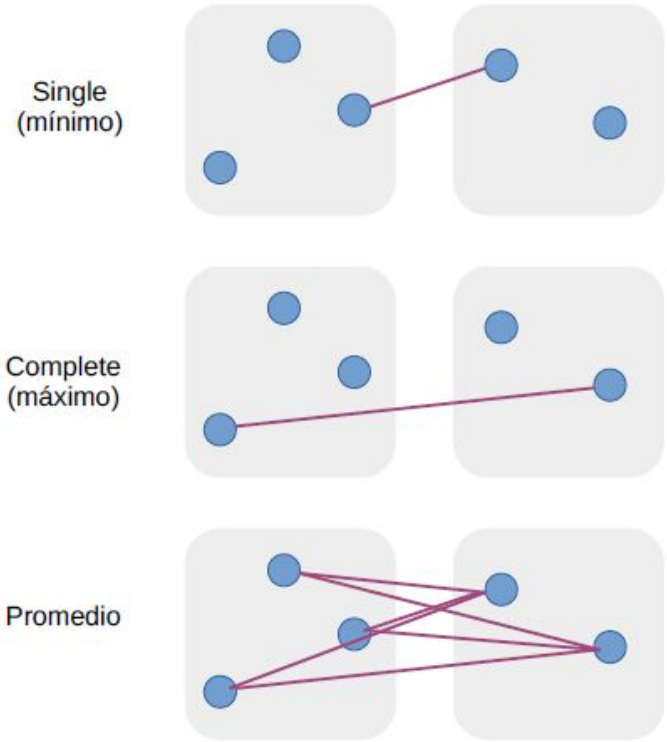
  La mayor distancia posible entre puntos de cada cluster.

- Average-link clustering

  La distancia promedio entre puntos de cada cluster.

- Centroid-link clustering

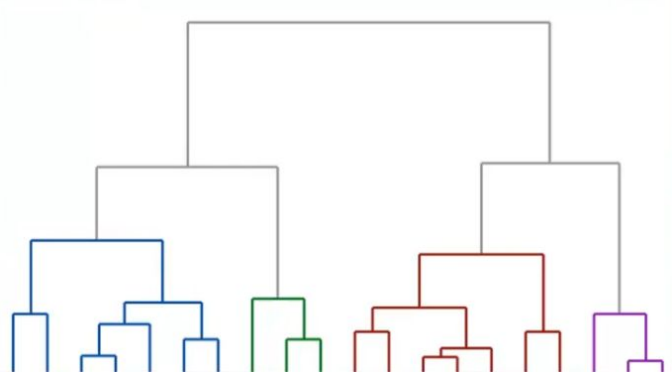
  La distancia al centroide de cada cluster.



Clustering jerárquico - Algoritmo aglomerativo

- Inicialmente, cada punto forma un singleton cluster
- El objetivo es ir conectando clusters de forma iterativa cuya distancia es mínima
- El algoritmo acaba cuando tenemos un único cluster con todos los puntos

Con este proceso se habrá generado un dendrograma con las distancias y, por tanto, podemos partir por donde queramos para tener el número de clusters que se necesiten



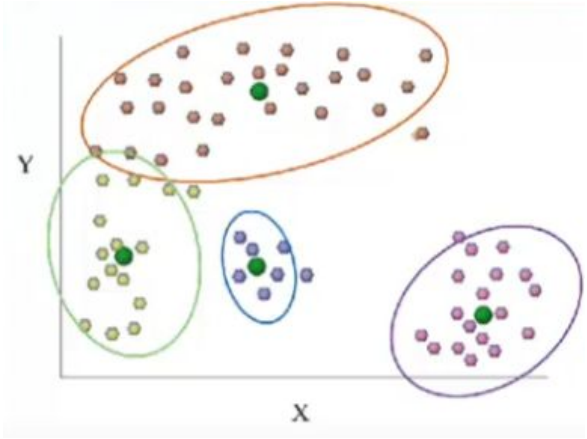
Clustering jerárquico - Algoritmo aglomerativo

- En el notebook 4_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

Clustering basado en centroides

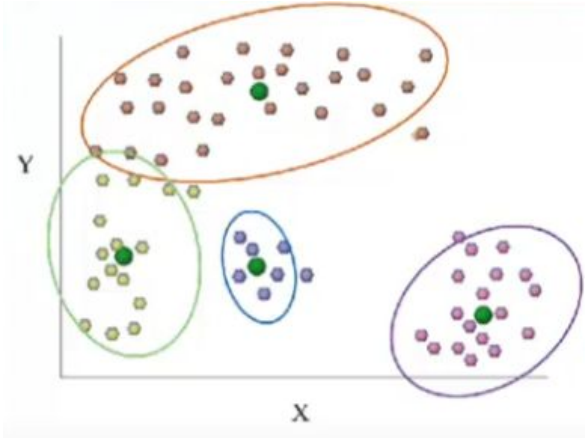
- Es necesario definir cuántos grupos se van a formar (hiperparámetro K)
- Estos grupos se basan en la similitud a los centroides

  Como ya hemos visto, un centroide es el punto promedio de los puntos de un cluster
- El centroide puede representar al grupo correspondiente



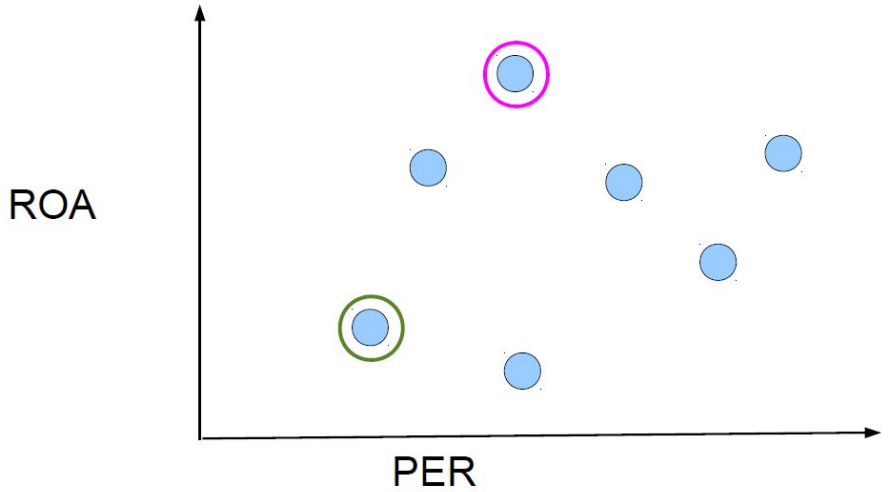
Clustering basado en centroides - K-means

- 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



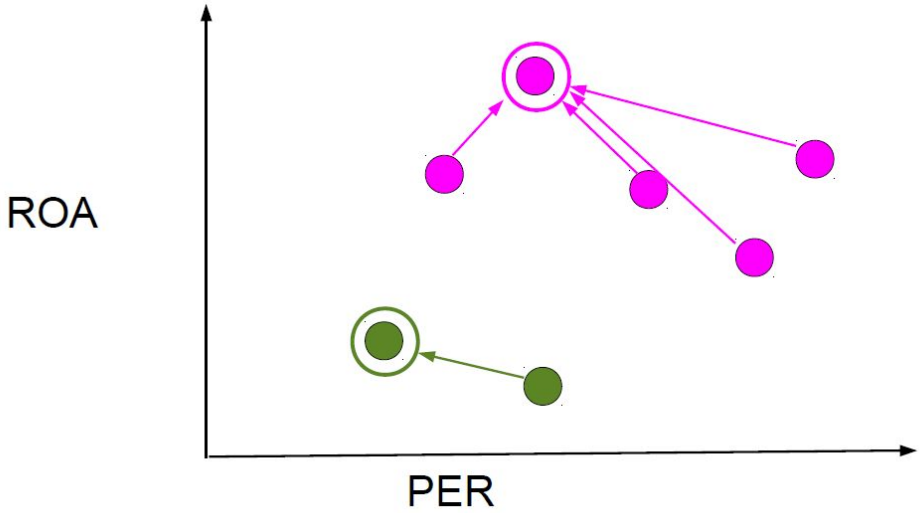
Clustering basado en centroides - K-means

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



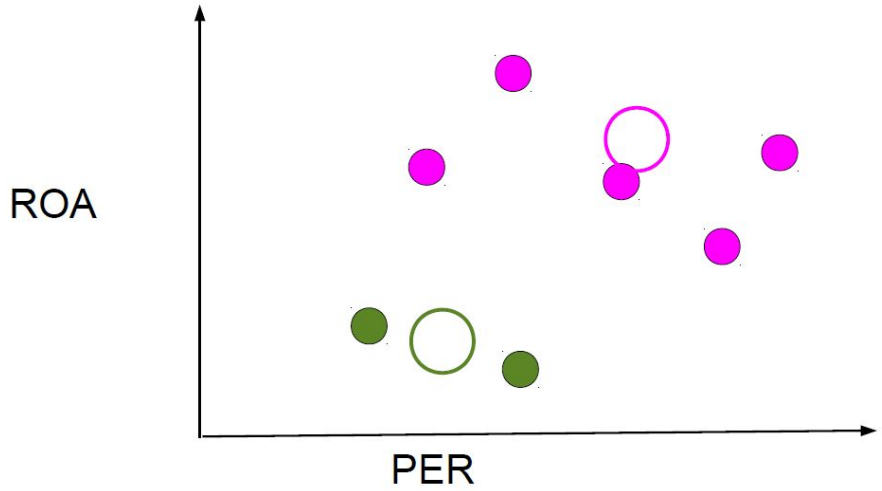
Clustering basado en centroides - K-means

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



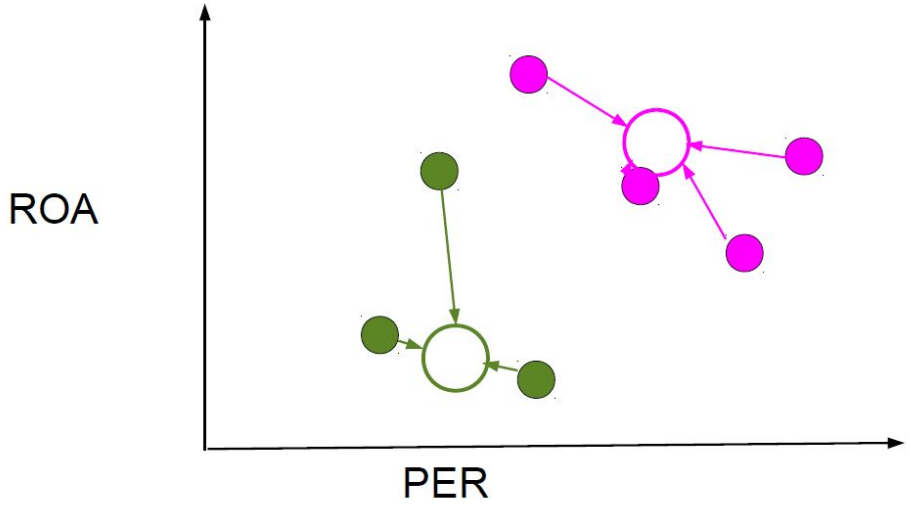
Clustering basado en centroides - K-means

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



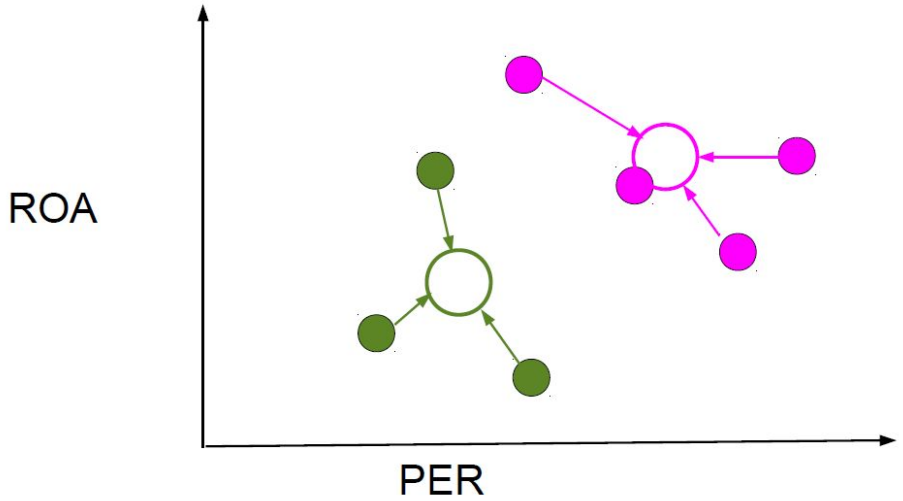
Clustering basado en centroides - K-means

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



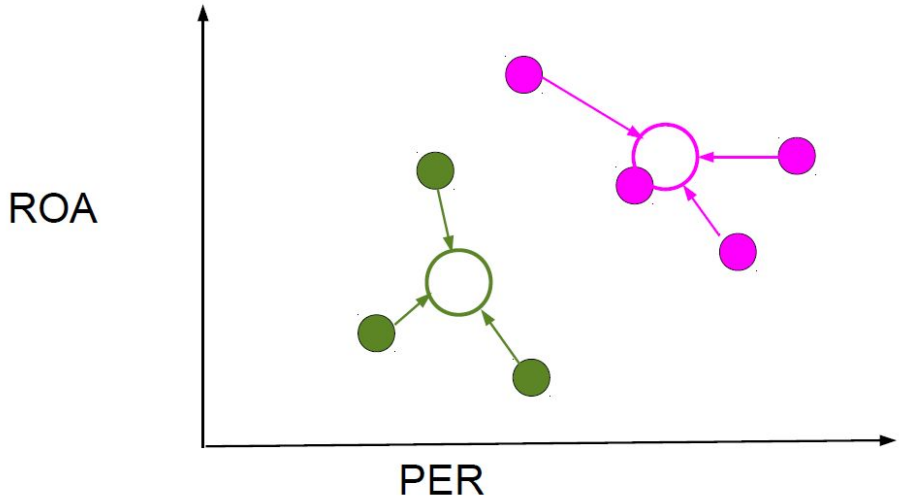
Clustering basado en centroides - K-means

- 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

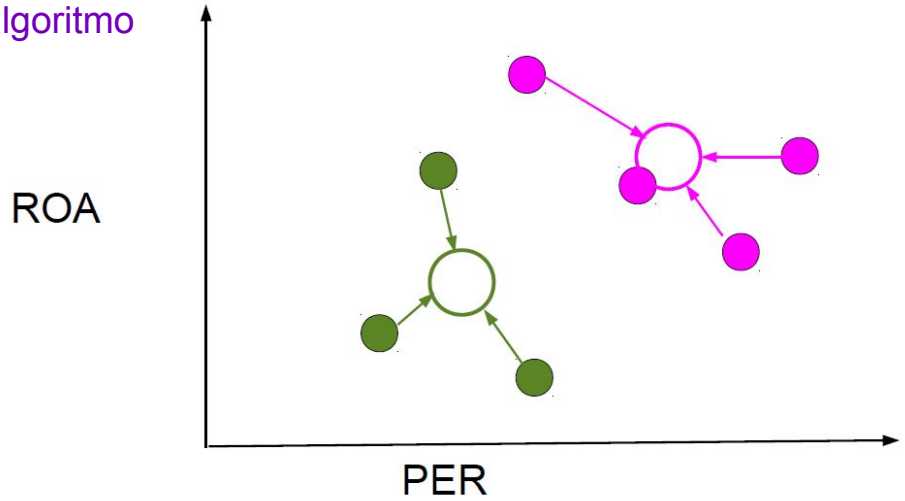
- Mientras los centroides se vayan modificando:
  - Asignación: Asignar a cada instancia el cluster más cercano
  - Actualización: Calcular los centroides de los K clusters



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 para datos numéricos (hay que definir la media)
  - Necesario definir el valor K. No es trivial sin conocimiento previo

Clustering basado en centroides - K-means

- Vamos al notebook 4_2_clustering_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 basado en mezcla de Gaussianas - EM

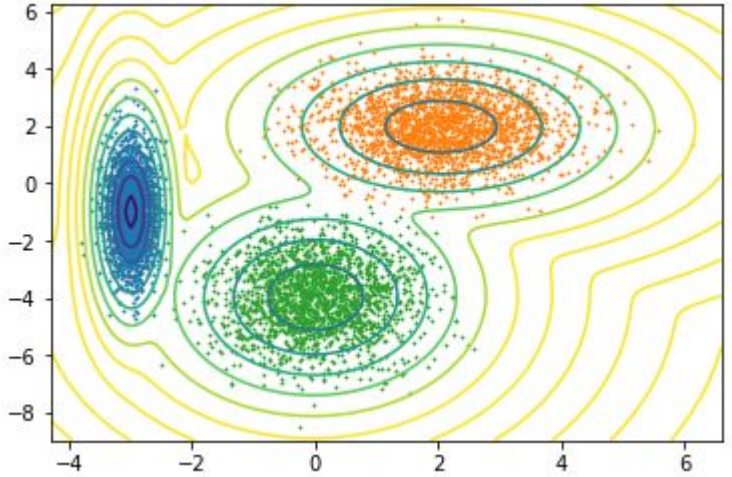
- Expectation-Maximization (EM) describe probabilísticamente los grupos

  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

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

donde:

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

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

Clustering basado en mezcla de Gaussianas - EM

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

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

- Asumimos que:
  - $P(G_j)$ es la probabilidad de pertenecer a la gaussiana G_j → Probabilidad a priori

    No conocemos los prioris → $P(G_j) = 1 / K$ para todas las gaussianas.
  - $P(x_i)$ es la probabilidad de seleccionar el elemento x_i → Evidencia

    La probabilidad de cada punto es igual para todos → $P(x_i) = 1 / n$ para todos los puntos.

Clustering basado en mezcla de Gaussianas - EM

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

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

- Ejemplo:

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

  G_1 = (μ_1= 2.0, σ_1=1.0) y G_2 = (μ_2= -2.0, σ_2=1.0)

  Calculamos los prioris y las evidencias:

Clustering basado en mezcla de Gaussianas - EM

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

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

- Ejemplo:

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

  G_1 = (μ_1= 2.0, σ_1=1.0) y G_2 = (μ_2= -2.0, σ_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:

Clustering basado en mezcla de Gaussianas - EM

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

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

- Ejemplo:

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

  G_1 = (μ_1= 2.0, σ_1=1.0) y G_2 = (μ_2= -2.0, σ_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 las probabilidades:

  Calculamos verosimilitudes G1:

  $P(x1 \mid G1)$ = … = 0.241971

  $P(x2 \mid G1)$ = … = 0.398942

  $P(x3 \mid G1)$ = … = 0.000001

  $P(x4 \mid G1)$ = … = 0.000134

  Calculamos verosimilitudes G2:

  $P(x1 \mid G2)$ = … = 0.000001

  $P(x2 \mid G2)$ = … = 0.000134

  $P(x3 \mid G2)$ = … = 0.241971

  $P(x4 \mid G2)$ = … = 0.398942

Clustering basado en mezcla de Gaussianas - EM

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

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

- Ejemplo:

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

  G_1 = (μ_1= 2.0, σ_1=1.0) y G_2 = (μ_2= -2.0, σ_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:

$P(x1 \mid G1)$ = … = 0.241971

$P(x2 \mid G1)$ = … = 0.398942

$P(x3 \mid G1)$ = … = 0.000001

$P(x4 \mid G1)$ = … = 0.000134

Calculamos verosimilitudes G2:

$P(x1 \mid G2)$ = … = 0.000001

$P(x2 \mid G2)$ = … = 0.000134

$P(x3 \mid G2)$ = … = 0.241971

$P(x4 \mid G2)$ = … = 0.398942

Calculamos las probabilidades:

$P(G1 \mid x1)$ = … = 0.483941

$P(G1 \mid x2)$ = … = 0.797884

$P(G1 \mid x3)$ = … = 0.000003

$P(G1 \mid x4)$ = … = 0.000268

$P(G2 \mid x1)$ = … = 0.000003

$P(G2 \mid x2)$ = … = 0.000268

$P(G2 \mid x3)$ = … = 0.483941

$P(G2 \mid x4)$ = … = 0.797884

Clustering basado en mezcla de Gaussianas - EM

- Ejemplo:
  - Asignamos al cluster con mayor probabilidad:

$P(G_1 \mid x_1) = \ldots = 0.483941 \rightarrow 0.999999$

$P(G_2 \mid x_1) = \ldots = 0.000003 \rightarrow 0.000001$

$P(G_1 \mid x_2) = \ldots = 0.797884 \rightarrow 0.999664$

$P(G_2 \mid x_2) = \ldots = 0.000268 \rightarrow 0.000336$

$P(G_1 \mid x_3) = \ldots = 0.000003 \rightarrow 0.000001$

$P(G_2 \mid x_3) = \ldots = 0.483941 \rightarrow 0.999999$

$P(G_1 \mid x_4) = \ldots = 0.000268 \rightarrow 0.000336$

$P(G_2 \mid x_4) = \ldots = 0.797884 \rightarrow 0.999664$

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:

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

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

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

Clustering basado en mezcla de Gaussianas - EM

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

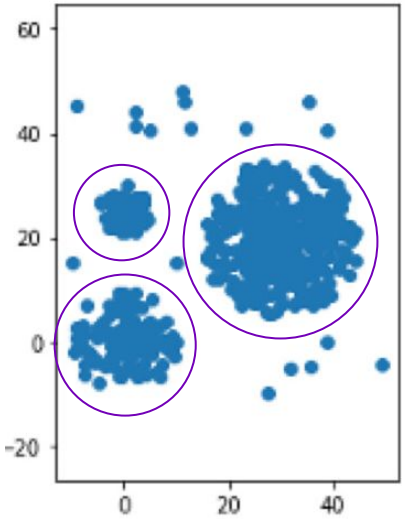
$$\mu_j^{(t+1)} = \frac{\sum \theta_{ij}^{(t)} x_i}{\sum \theta_{ij}^{(t)}} \qquad \sigma_j^{2:(t+1)} = \frac{\sum \theta_{ij}^{(t)} (x_i - \mu_j^{(t+1)})^2}{\sum \theta_{ij}^{(t)}}$$

Clustering basado en mezcla de Gaussianas - EM

- Notebook 8_3_clustering_em.ipynb

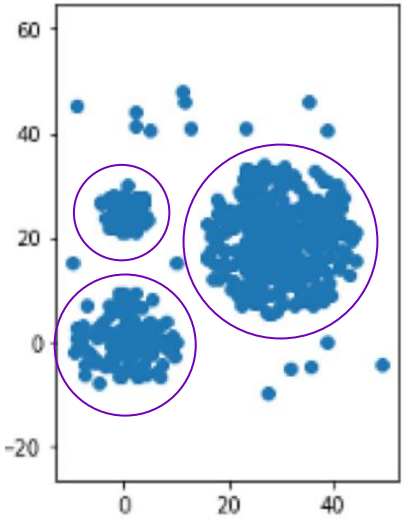
Clustering basado en densidades - DBSCAN

- 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



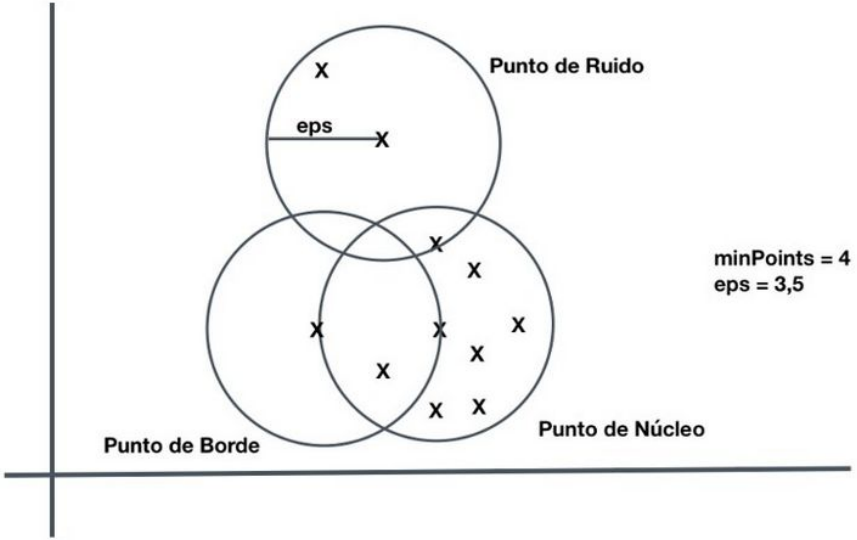
Clustering basado en densidades - DBSCAN

- 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

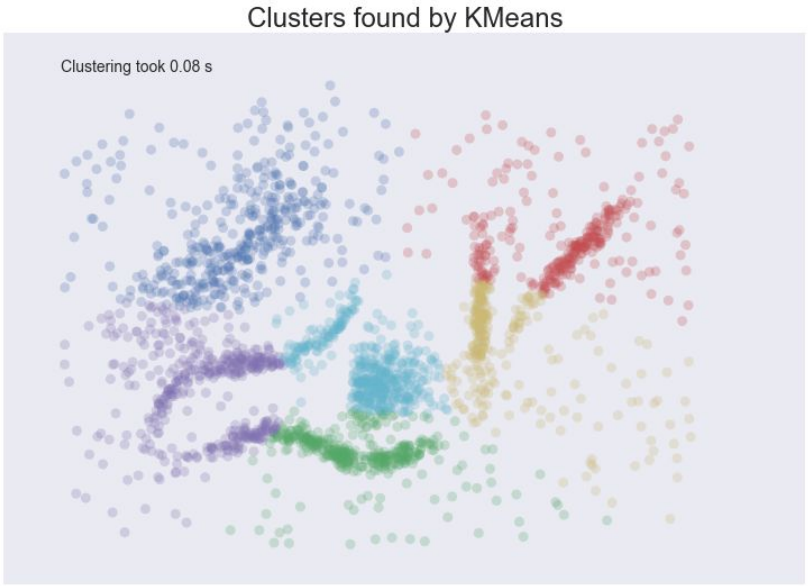
- Algoritmo DBSCAN:
  - Selecciona un punto arbitrario que no haya sido previamente visitado. Se calculan las distancias a todos y la información de su vecindario se recupera según el valor de eps.
  - Si es punto de núcleo (contiene minPts o más puntos en el vecindario):
    - Todos los puntos del vecindario 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).

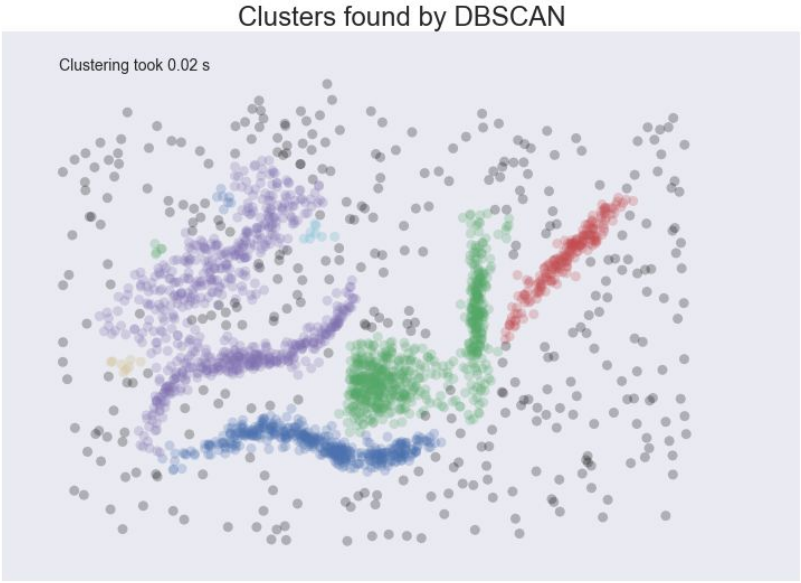
Clustering basado en densidades - DBSCAN

- 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 eps y minPts

Clustering basado en densidades - K-means vs DBSCAN

- Comparación K-means vs 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 (PCA)
  - Selección de características
  - Clustering en sub-espacios: combinación de características

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 4_4_comparacion_clustering.ipynb tenéis:
  - Dos ejercicios para hacer en clase probando dos problemas sintéticos
  - Selección del mejor modelo con coeficiente de silhouette

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/