Módulo 5: Aprendizaje no supervisado

Clustering 2da parte

Diplomatura Cs. de Datos - FaCENA-UNNE

Docentes: Magdalena Lucini, Luis Duarte, Griselda Bóbeda

Agrupamiento

Encuentro anterior vimos algoritmos divisivos, para los cuales debía especificarse el número de clusters a encontrar. Todos los individuos quedan clasificados

- K-means: hiper-esferas en el centro de cada cluster, con radios definidos por el punto más distante al centroide del cluster.
- GMM: asume que cada observación fue generada por una mezcla de gaussianas. Clusters son hiper-elipsoides.

¿Qué pasa si hay datos atípicos?, ¿si hay ruido?, ¿si clusters tienen formas arbitrarias?, ¿si hay regiones de mayor y menor densidad de puntos?

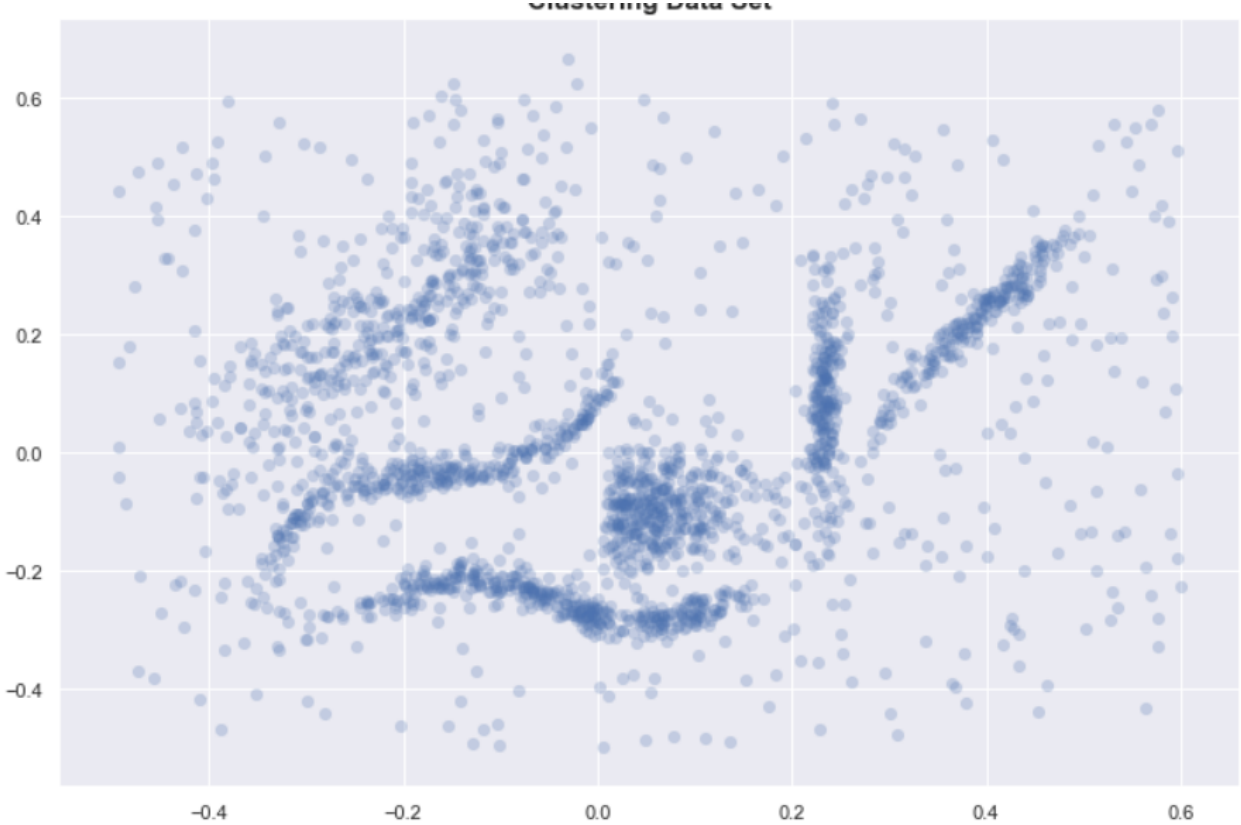
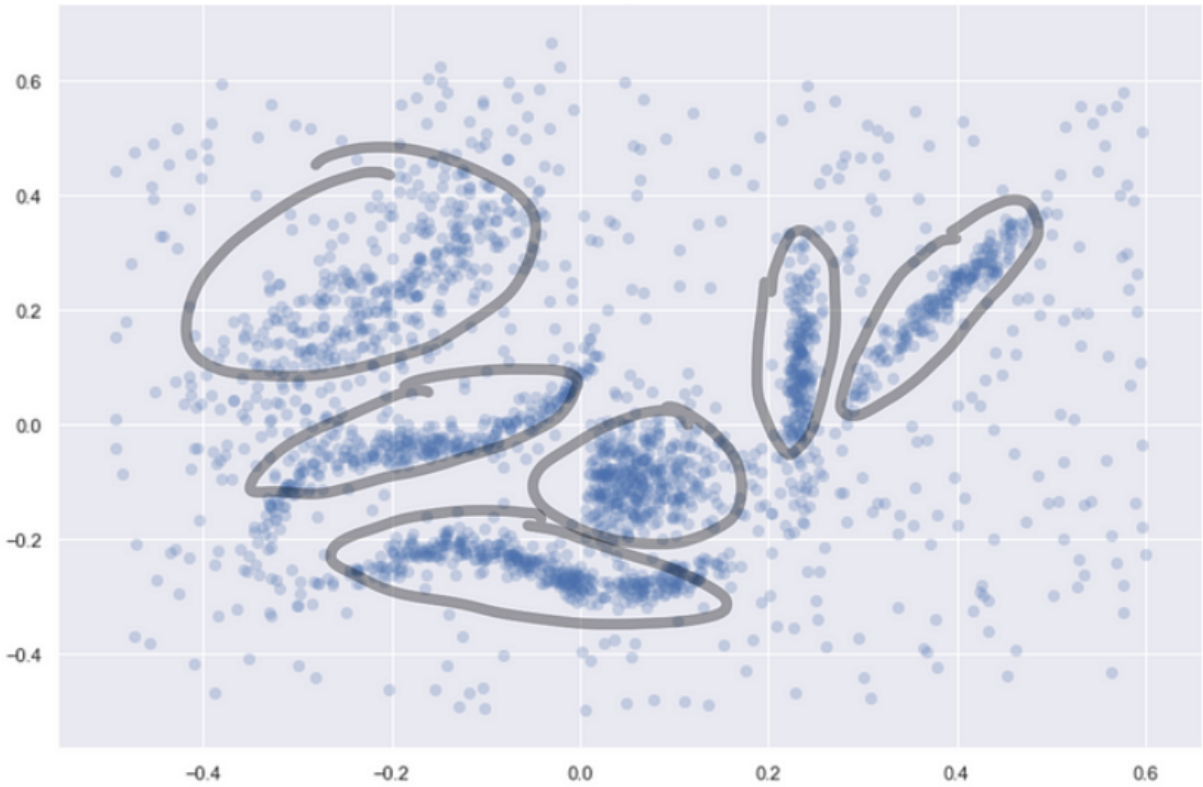


Figure: Agrupamiento por kmeans

Métodos basados en densidad (Ej DBSCAN, HDBSCAN)

- identifica distintos grupos en los datos basándose en la idea que un cluster es una región de alta densidad de puntos
- regiones de alta densidad de puntos están separadas por regiones de baja densidad de puntos
- Útil para descubrir agrupaciones de formas variadas y detectar outliers.

Métodos de agrupamiento basados en **densidades**



Métodos basados en densidades

DBSCAN

Density Based Spatial Clusterig of Applications with Noise (DBSCAN)

- Clusters: regiones densas en el espacio de datos, separadas por regiones de baja densidad de puntos.
- Idea básica: Para que un punto pertenezca a un cluster , un entorno de un radio dado, centrado en ese punto, tiene que contener una cantidad mínima de puntos
- Puede encontrar clusters de diferentes formas y tamaños , detecta ruido y puntos outliers.

[1] Ester, M., H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231. 1996

DBSCAN

Parámetros

- ϵ (eps) = define el entorno (vecindad) de un punto: Si $d(\mathrm{x}_i, \mathrm{x}_j) < \epsilon$ $\Rightarrow$ x_i y x_j son "vecinos". Un ϵ-entorno (o vecindad) de x_i se define como:

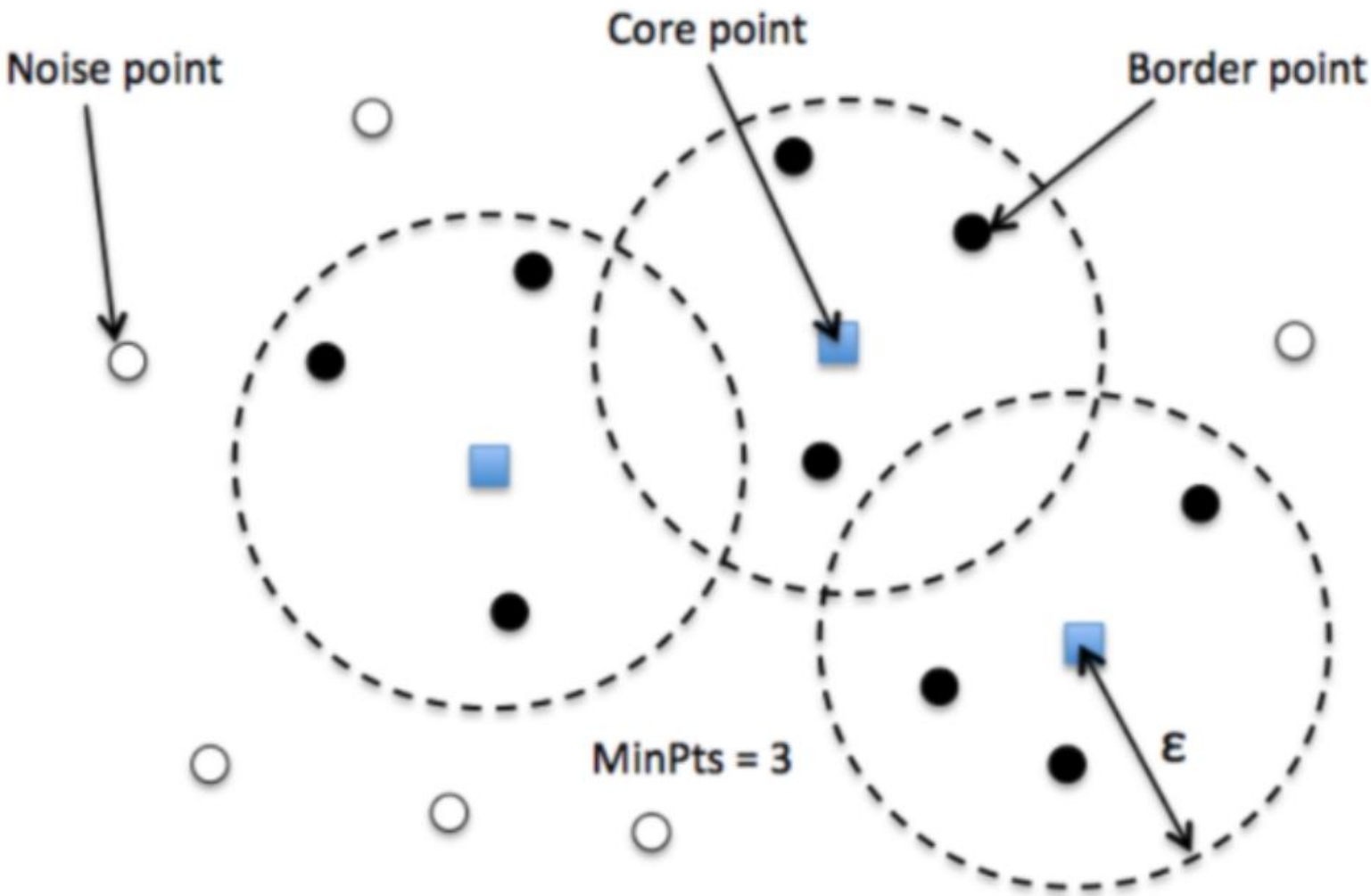
$$N_\epsilon(\mathrm{x_i}) := \{\mathrm{x} \in \mathrm{X} : d(\mathrm{x}, \mathrm{x}_i) \leq \epsilon\}$$

- min Pts: mínimo número de vecinos (puntos) en el entorno $N_\epsilon(\mathrm{x})$.

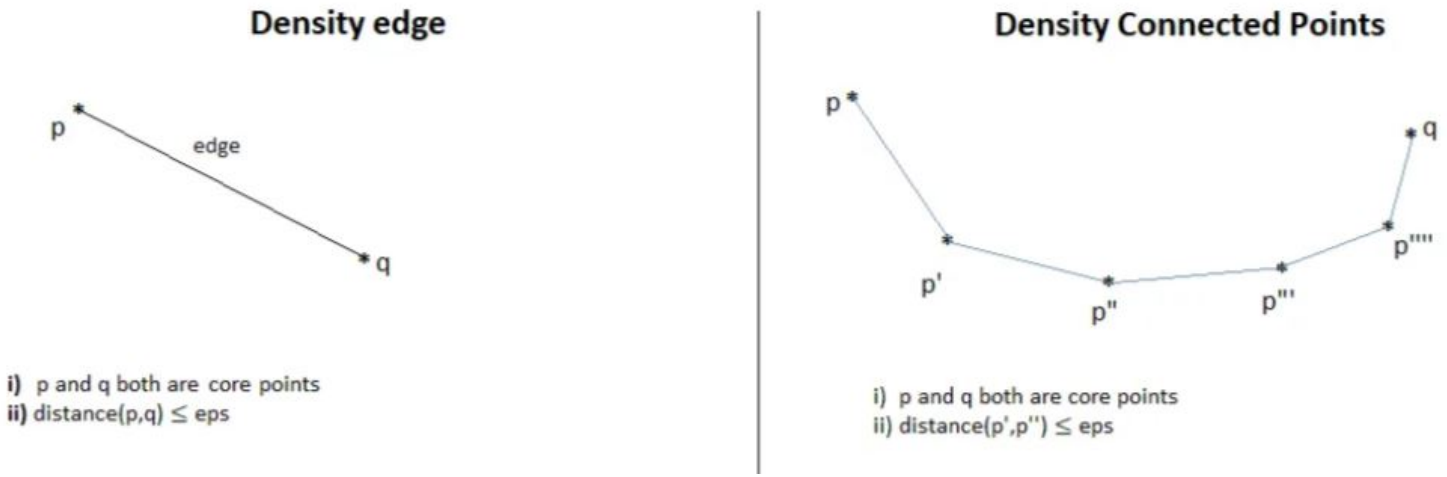
Etiquetado de los puntos:

- punto de núcleo (core): la cantidad de puntos a un radio ϵ de ese punto es mayor a min Pts. $\#N_\epsilon(\mathrm{x_i}) \geq minPts$
- punto de borde: la cantidad de puntos a un radio ϵ de ese punto es menor a min Pts, pero está en el entorno de un punto de núcleo. $\#N_\epsilon(\mathrm{x_i}) < minPts$ y $\mathrm{x}_i \in N_\epsilon(\mathrm{x_j})$, con x_j punto core.
- ruido o outlier: punto que no es punto de núcleo ni punto de borde.

DBSCAN



DBSCAN



borde de densidad: si la distancia entre dos núcleos es menor a ϵ, se pueden unir esos puntos por un segmento denominado "borde de densidad"

puntos conectados por densidad: Se dice que dos puntos p y q son puntos conectados por densidades si ambos son puntos de núcleo y existe un camino formado por bordes de densidades que conectan el punto p con el punto q

DBSCAN- Algoritmo

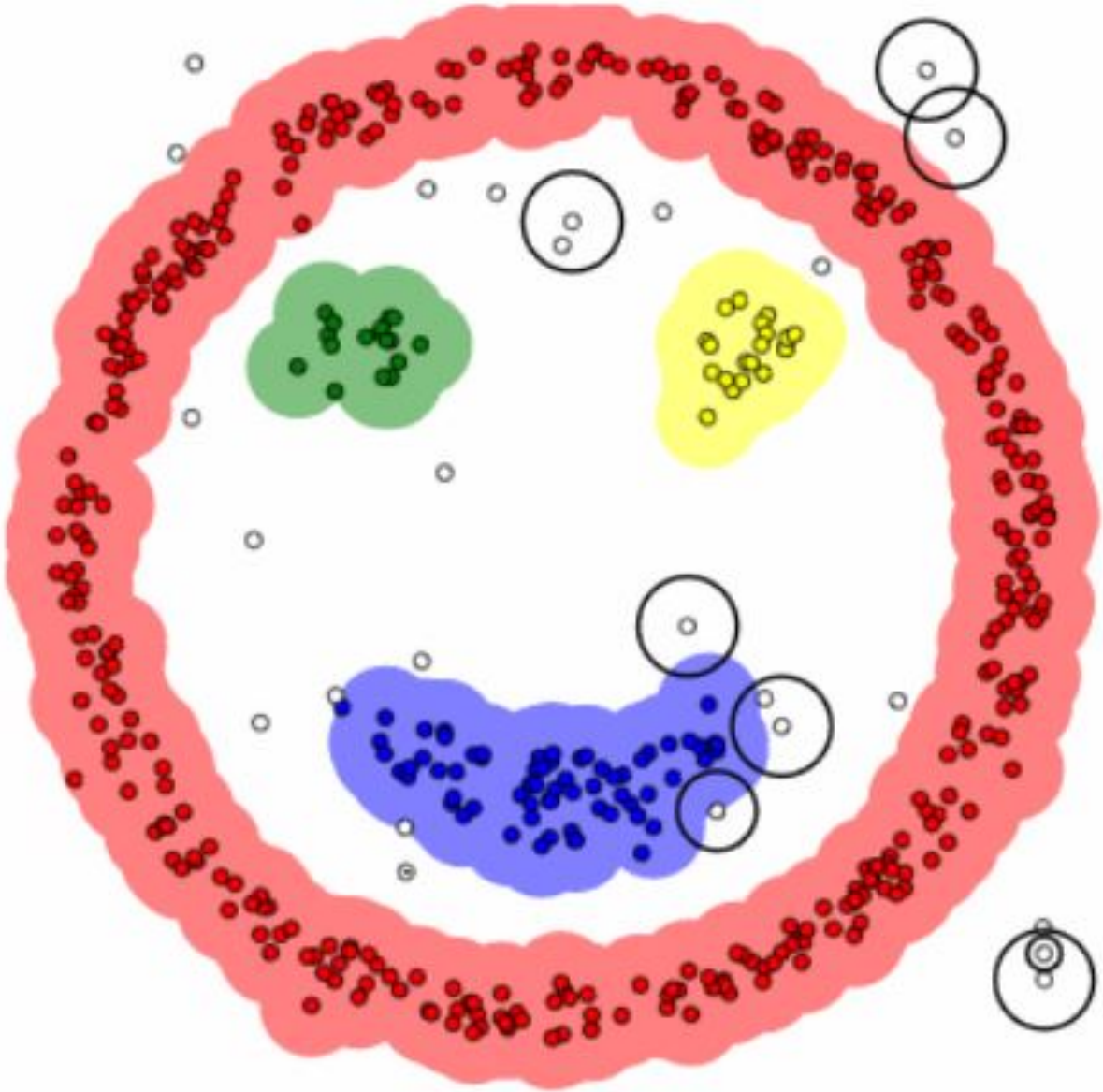
Dado un dataset y fijados los parámetros ϵ y min Pts:

1. Toma un punto arbitrario y encuentra todos los puntos en el entorno de radio ϵ
2. Se etiqueta el punto. Si es un punto de núcleo, comienza la formación de grupos, si no lo es, se etiqueta al punto como ruido (esta etiqueta puede modificarse más tarde, pues ese punto puede estar en el entorno de otro punto, o ser punto de frontera)
3. Se buscan todos los puntos conectados por densidad a ese punto núcleo y se los asigna al mismo grupo.
4. Se itera sobre todos los puntos que no fueron visitados, formándose los distintos grupos.

Los puntos que no pertenecen a ningún grupo son ruido o outliers.

DBSCAN

DBSCAN



DBSCAN - Datos Mall

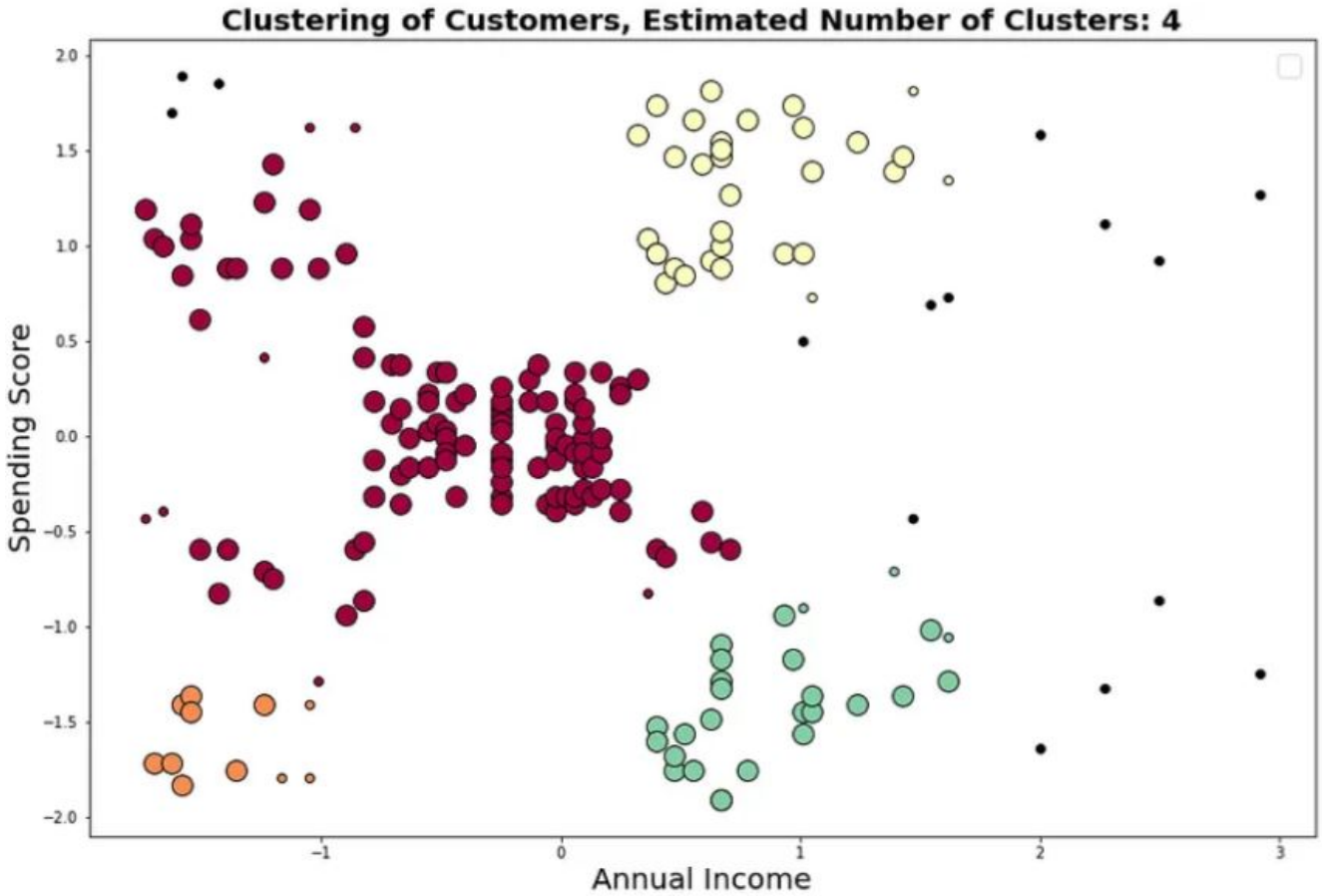


Figure: DBSCAN, $\epsilon = 0.4$, min Pts $= 5$

DBSCAN

```
Cluster 0, Avg Annual Income: 48, Avg Spending Score: 52, Count: 114
Cluster 1, Avg Annual Income: 24, Avg Spending Score: 9, Count: 11
Cluster 2, Avg Annual Income: 81, Avg Spending Score: 84, Count: 32
Cluster 3, Avg Annual Income: 84, Avg Spending Score: 14, Count: 27
```

Figure: DBSCAN, $\epsilon = 0.4$, min Pts $= 5$

```
Cluster 0, Avg Annual Income: 21, Avg Spending Score: 75, Count: 10
Cluster 1, Avg Annual Income: 25, Avg Spending Score: 32, Count: 5
Cluster 2, Avg Annual Income: 55, Avg Spending Score: 49, Count: 87
Cluster 3, Avg Annual Income: 79, Avg Spending Score: 84, Count: 27
Cluster 4, Avg Annual Income: 76, Avg Spending Score: 10, Count: 14
Cluster 5, Avg Annual Income: 90, Avg Spending Score: 14, Count: 7
```

Figure: DBSCAN, $\epsilon = 0.25$, min Pts $= 5$

```
Cluster 0, Avg Annual Income: 23, Avg Spending Score: 75, Count: 11
Cluster 1, Avg Annual Income: 55, Avg Spending Score: 49, Count: 87
Cluster 2, Avg Annual Income: 79, Avg Spending Score: 84, Count: 29
Cluster 3, Avg Annual Income: 80, Avg Spending Score: 13, Count: 22
```

Figure: DBSCAN, $\epsilon = 0.4$, min Pts $= 10$

```
Cluster 0, Avg Annual Income: 61, Avg Spending Score: 50, Count: 199
```

Figure: DBSCAN, $\epsilon = 0.75$, min Pts $= 5$

DBSCAN en sklearn

DBSCAN

```
class sklearn.cluster.DBSCAN(eps=0.5, *, min_samples=5,
metric='euclidean', metric_params=None, algorithm='auto',
leaf_size=30, p=None, n_jobs=None)
```

Los resultados de DBSCAN dependen fuertemente de la elección de sus dos parámetros principales: ϵ y MinPts (min_samples)

La elección de la distancia (metric) tiene cierto impacto en los resultados. Es importante identificar una medida de similaridad que sea apropiada para el data set

DBSCAN: elección de <Min Pts

Min Pts

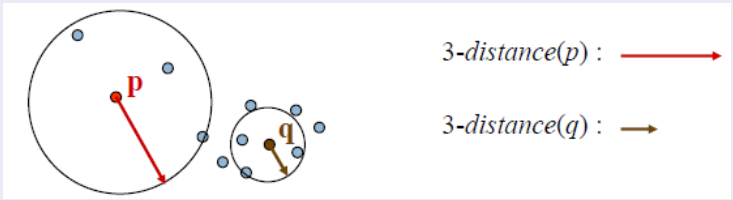
- Min Pts $\geq p + 1$, con p= cantidad de variables.
- Una aproximación inicial es MinPts $= 2p$, con p= número de características (dimensiones, variables).
- Si el data set es muy ruidoso, o se desea identificar clusters pequeños, Min Pts puede ser menor
- Si el conjunto es muy voluminoso, Min Pts podría ser mayor.
- Se requiere bastante conocimiento del dominio

DBSCAN: elección de ϵ

ϵ

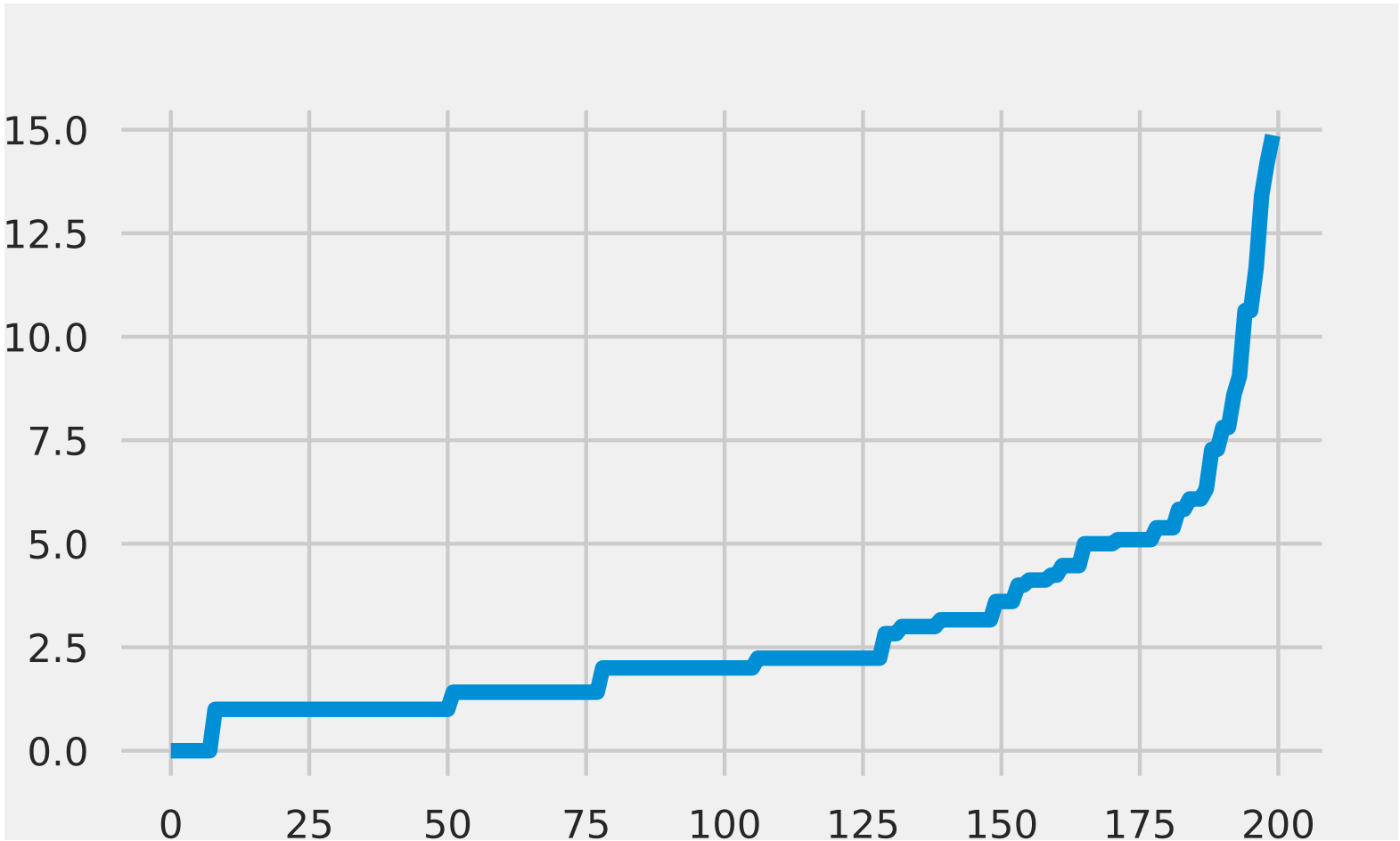
Determina la máxima distancia para la cual dos puntos pueden ser considerados vecinos. ϵ pequeño $\rightarrow$ muchos datos sin agrupar; ϵ muy grande $\rightarrow$ la mayoría de los datos en un mismo grupo.

- **Conocimiento del dominio**: si se sabe qué distancia es significativa para el problema específico, se la usa como punto de partida.
- **Gráfico de k-distancias** (gráfico de codo, knee graph):
  - Para cada punto x, se calcula la distancia d al k-ésimo vecino más cercano (k=MinPts). Esta d-vecindad de x tiene $k+1$ puntos para casi todo punto p.

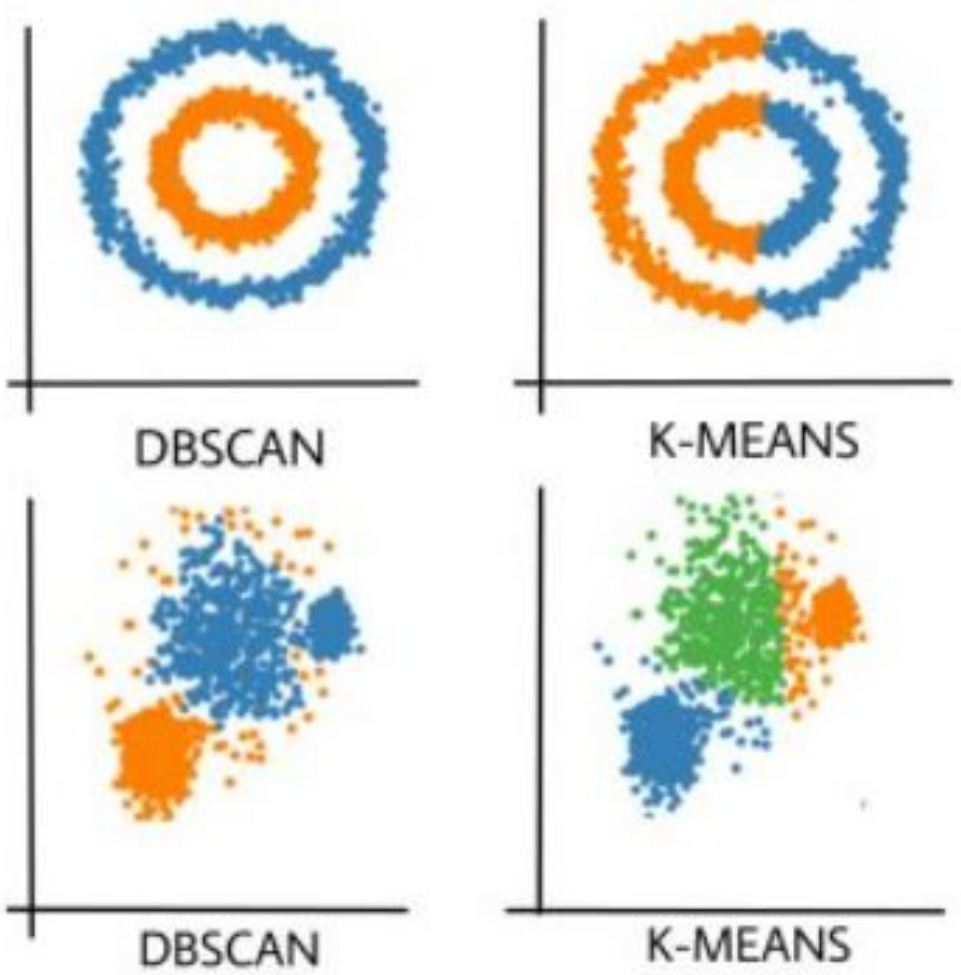


  - Se define la función k-dist:$\mathrm{X} \rightarrow \mathbb{R}$ que a cada punto x le asigna la distancia a su k-ésimo vecino más cercano.
  - Se grafican estas distancias en orden ascendente.
  - Se elige ϵ donde se observa un codo.

kNN



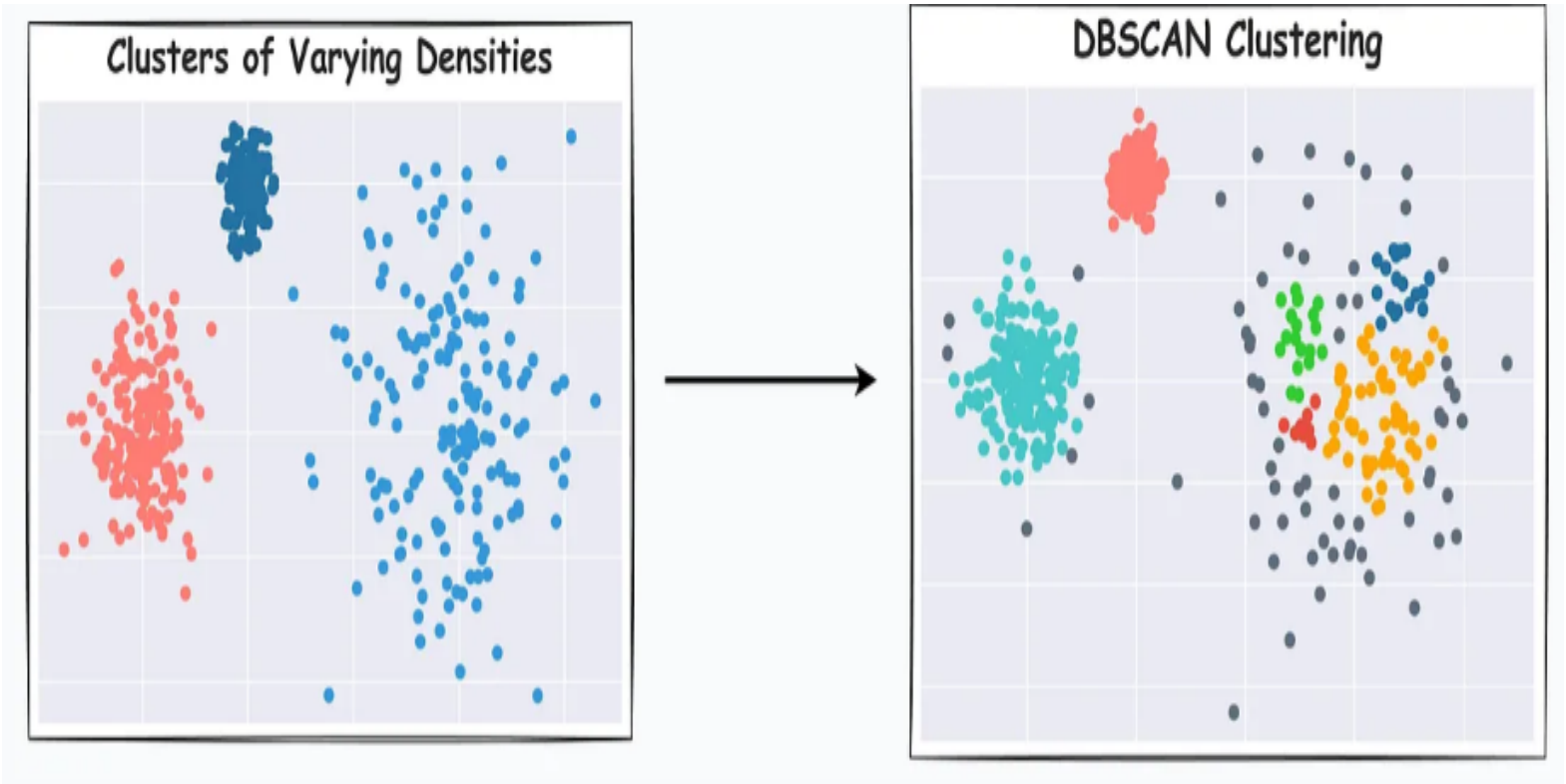
Kmeans vs DBSCAN



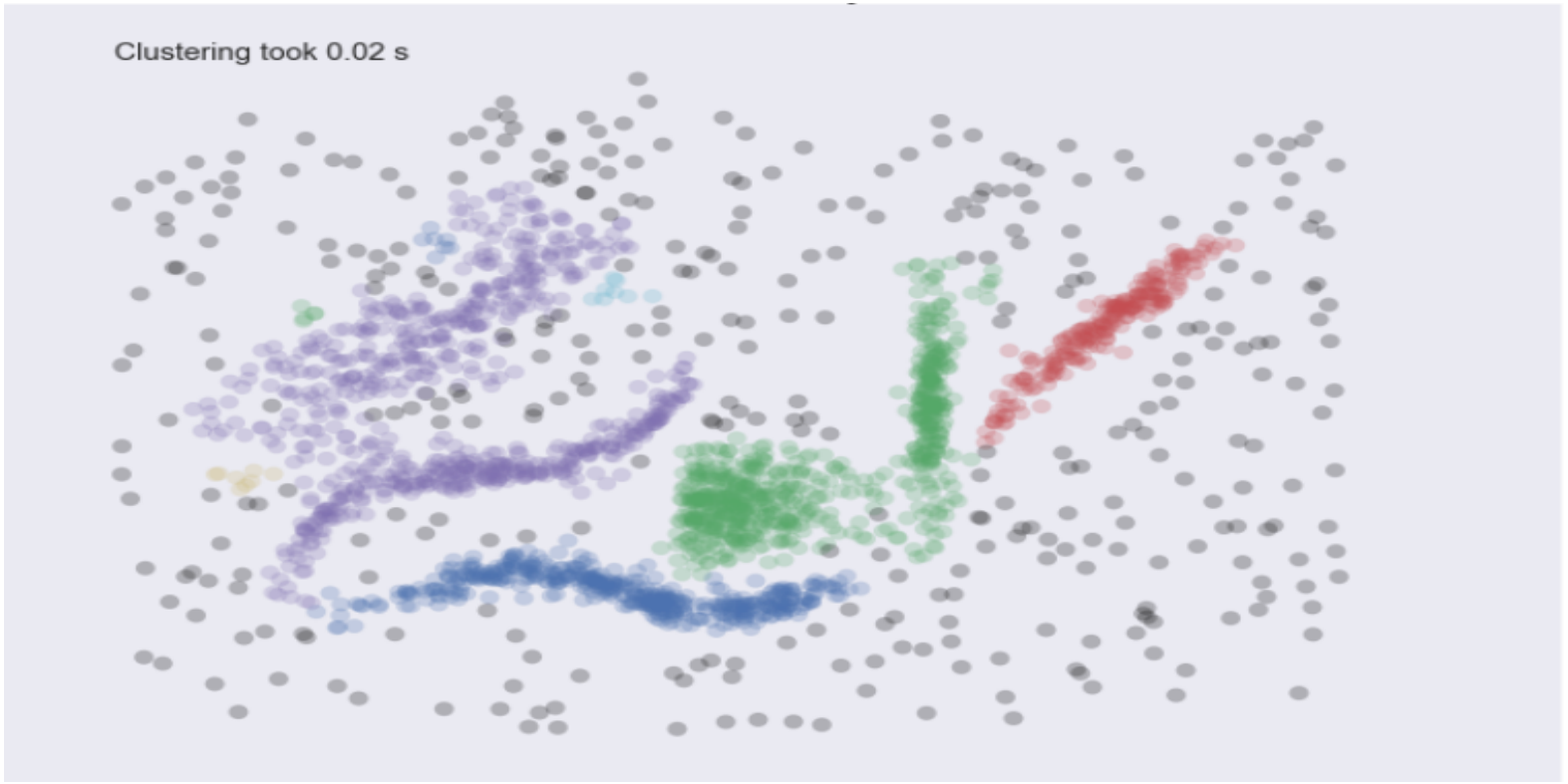
Algunas consideraciones sobre DBSCAN

- Puede identificar clusters de formas arbitrarias y de distintas densidades
- No es necesario especificar número de clusters.
- Identifica outliers como "puntos de ruido"
- Sensible a la elección de los parámetros ϵ y MinPts
- Caro computacionalmente para grandes volúmenes de datos, especialmente en situaciones de alta dimensionalidad.

¿Qué pasa si regiones muy densas y regiones poco densas?



¿Qué pasa si regiones muy densas y regiones poco densas?



Agrupamiento por DBSCAN

HDBSCAN

HDBSCAN: Hierarchical Density-Based Spatial Clustering of Applications with Noise [2]

- Extensión del algoritmo DBSCAN: lo convierte un algorimo jerárquico.
- Examina las densidades de los puntos en el dataset, crea clusters con puntos conectados por densidades.
- Construye una jerarquía de clusters basada en densidades: comienza con clusters pequeños y densos y los va agrupando en clusters más grandes a medida que la densidad decrece.
- Construida esta jerarquía, HDBSCAN busca los clusters más estables:son los que permanecen a lo largo de un rango de niveles de densidad (o a lo largo de distintas jerarquías)
- Diferencia clusters de ruido..
- Al igual que DBSCAN, no debe especificarse el número de clusters de antemano.
- Basándose en el dataset, automáticamente encuentra el valor de ϵ óptimo.

[2] Campello, R.J.G.B., Moulavi, D., Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. In Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science(), vol 7819.

HDBSCAN

```
class
sklearn.cluster.HDBSCAN(min_cluster_size=5,min_samples=None,
cluster_selection_epsilon=0.0, max_cluster_size=None,
metric='euclidean', metric_params=None, alpha=1.0,
algorithm='auto', leaf_size=40, n_jobs=None,
cluster_selection_method='eom', allow_single_cluster=False,
store_centers=None, copy=False)
```

Parámetros:

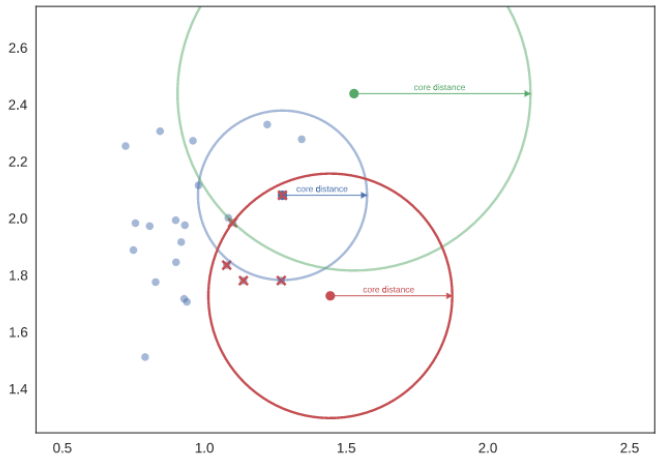
- **min_cluster_size**: Número mínimo de muestras para considerar un grupo como cluster.
- **min_samples**: Número mínimo de muestras en un vecindario para considerar un punto como central.

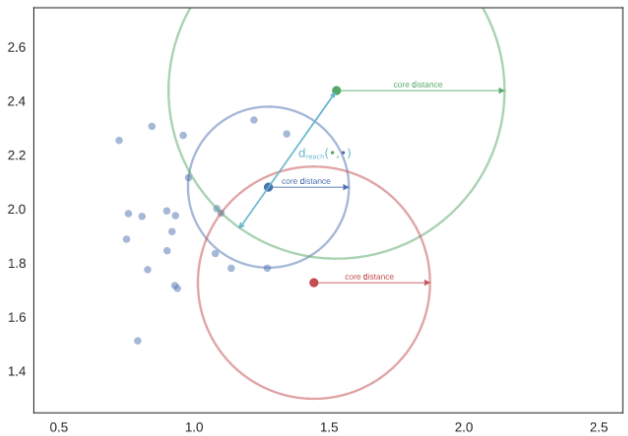
Pasos HDBSCAN

1. Estimar densidades
2. Construir una jerarquía de clusters de componentes conectadas
3. Condensar la jerarquía de clusters
4. Extaer los clusters estables

Paso 1: estimar densidades

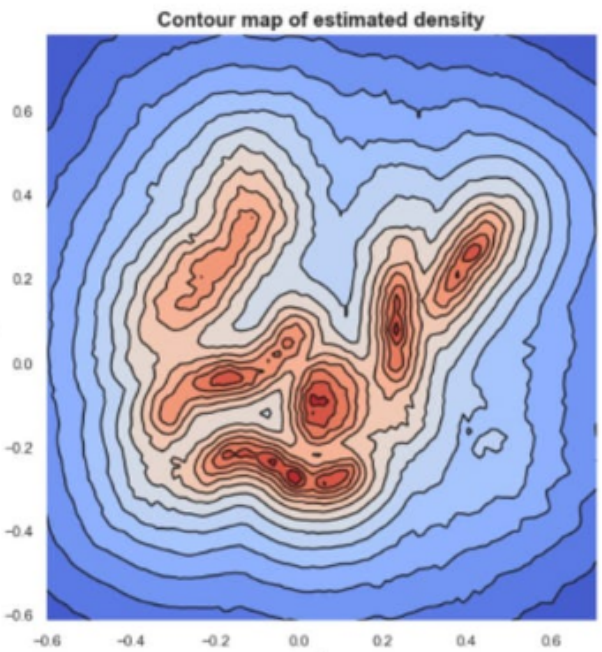
- Distinguir zonas "densas" de zonas ralas
- Distancias:
  a) k-ésimo vecino más cercano
  b) Distancia de alcance mutuo
  $d_{mreach-k}(a, b)$
- puntos en regiones densas tienen menores distancias que puntos en regiones esparsas.
- Estimador de densidad $\lambda = \frac{1}{\text{distancia}}$
- Con estas densidades se construye un escenario de densidades estimadas

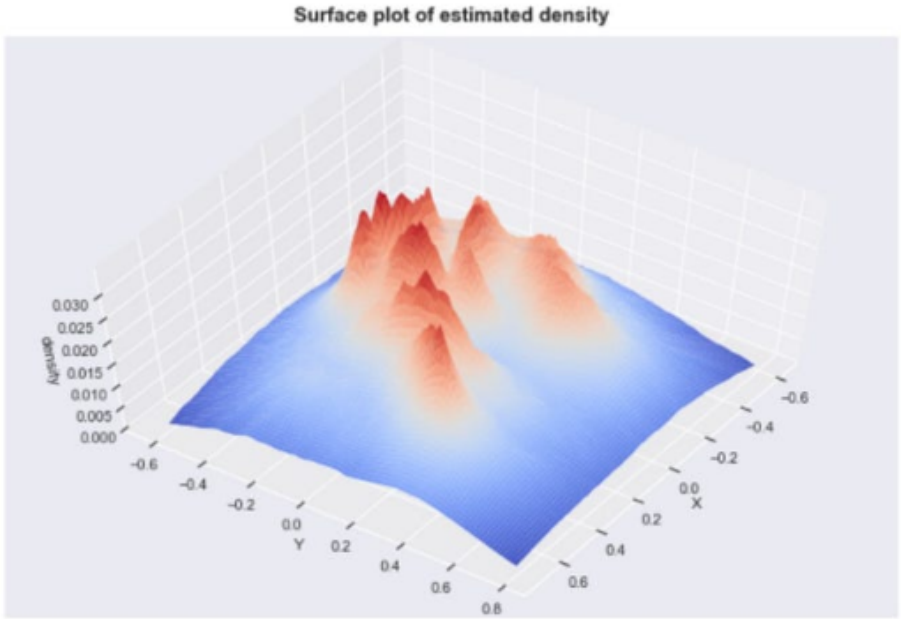




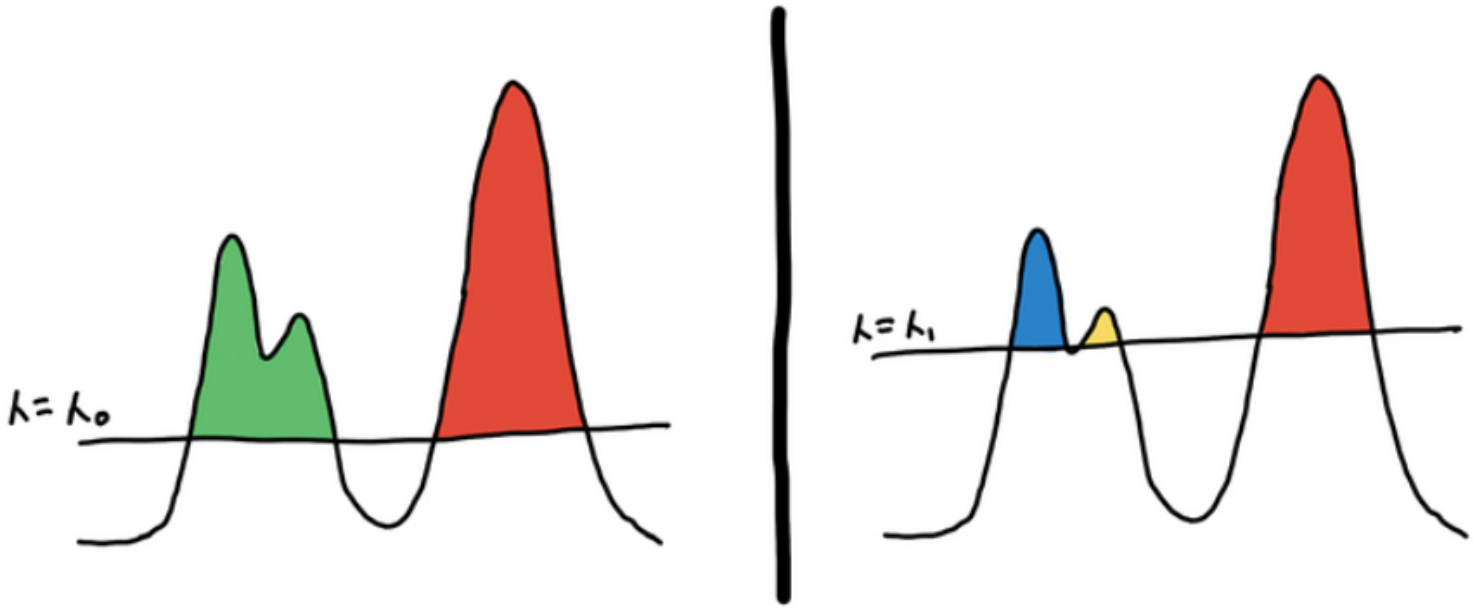
$$d_{mreach-k}(a, b) = \max\{\text{dnucleo}_k(a), \text{dnucleo}_k(b), d(a, b)\}$$

Paso 1 HDBSCAN





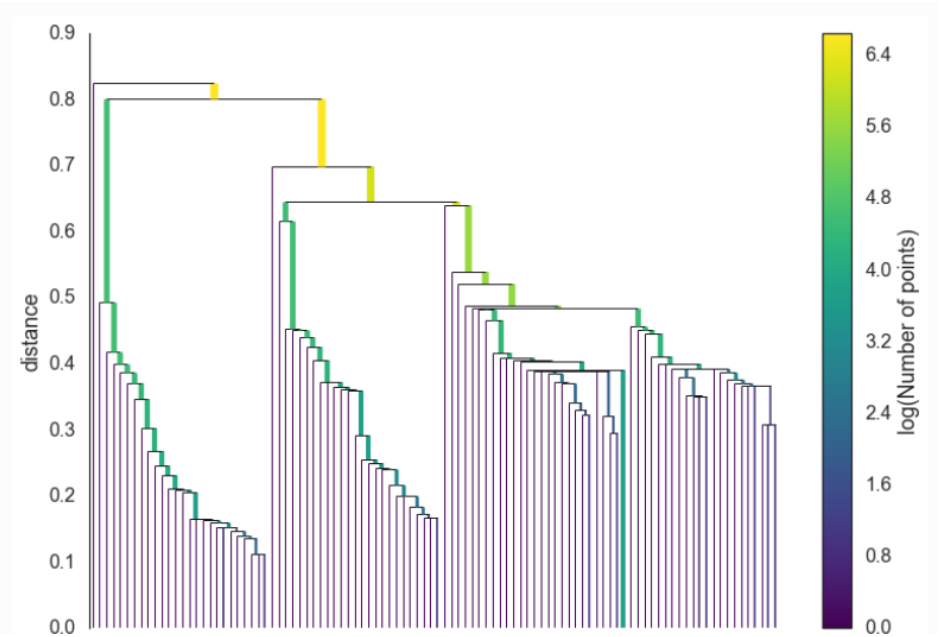
Las montañas son los clusters. Una manera de seleccionarlos es eligiendo un umbral global. Se toman los puntos con densidades por encima de ese umbral, agrupándolos se forman los clusters.



Corte de la superficie anterior, distintos umbrales para λ generan distintos clusters

Paso 2: Construir una jerarquía de clusters de componentes conectadas

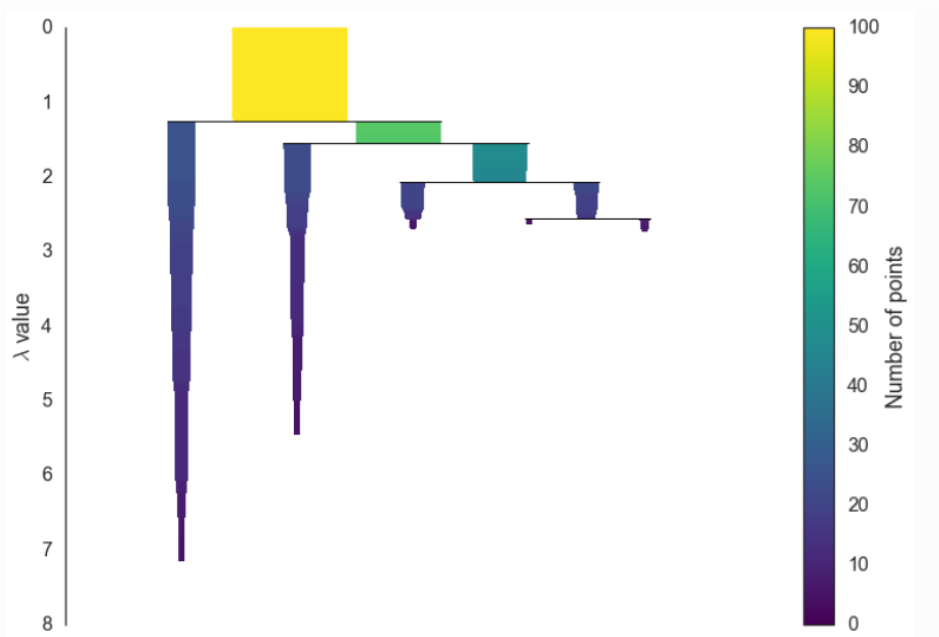
- Se construye un árbol de puntos conectados por distancias (usa $d_{mreach-k}$
- Se convierte ese árbol en una jerarquía de componentes conectadas.
- Se construye un dendograma *



* ver clustering jerárquico

Paso 3: Condensar la jerarquía de clusters

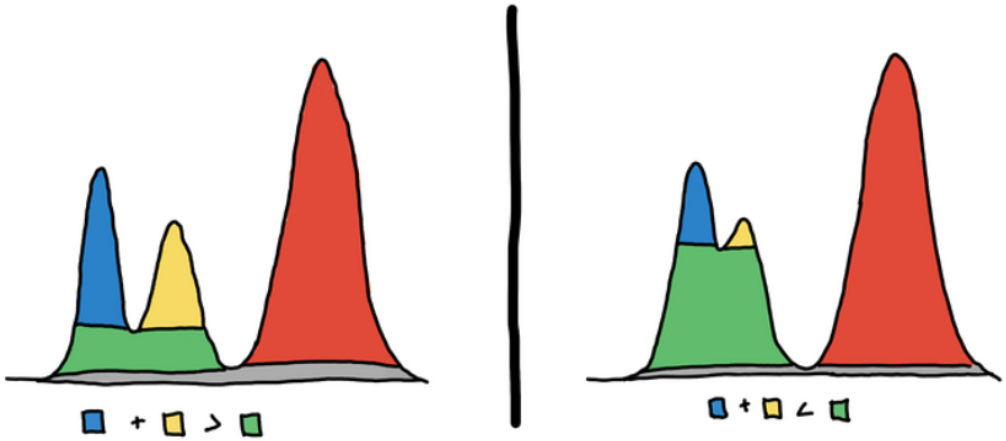
- Condensar la estructura jerárquica larga y complicada en un árbol mas pequeño.
- Se necesita el parámetro **min cluster size**: los clusters deben tener al menos esa cantidad de puntos.
- Este proceso termina con un gráfico similar a un dendograma, con un número más pequeño de nodos, el ancho de la línea representa la cantidad de puntos en el cluster.

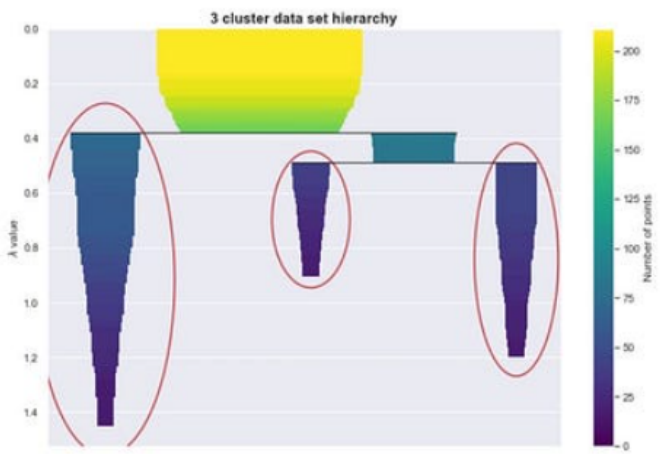


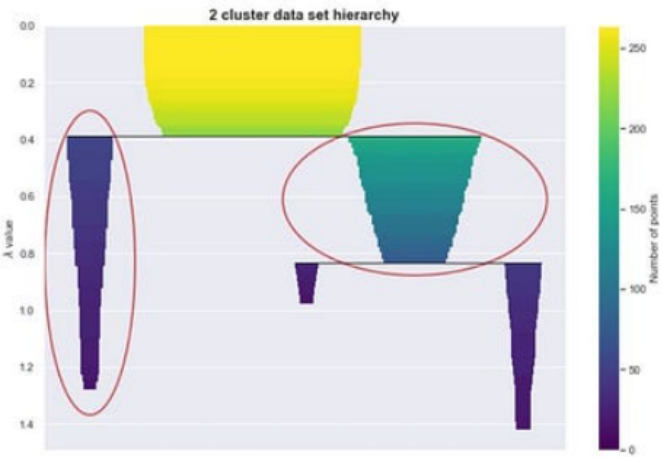
Paso 4: Extracción de clusters estables

- Elegir los clusters que más "persisten": los que tienen mayor área.
- Si se elige un nivel de corte, no se pueden elegir los "hijos" de ese cluster.
- Para un cluster dado se define λ_{birth}: valor de λ para el cual ese cluster se creó, y λ_{death} valor de λ (si existe) para el cual el cluster se dividió en clusters más pequeños.
- Para un cluster dado, para cada punto p en ese cluster, se define λ_p como el valor de λ para el cual ese punto en particular se "cayó" del cluster
- La estabilidad del cluster es $\sum_{p \in cluster}(\lambda_p - \lambda_{birth})$
- Se eligen los clusters usando **eom**: excess of mass

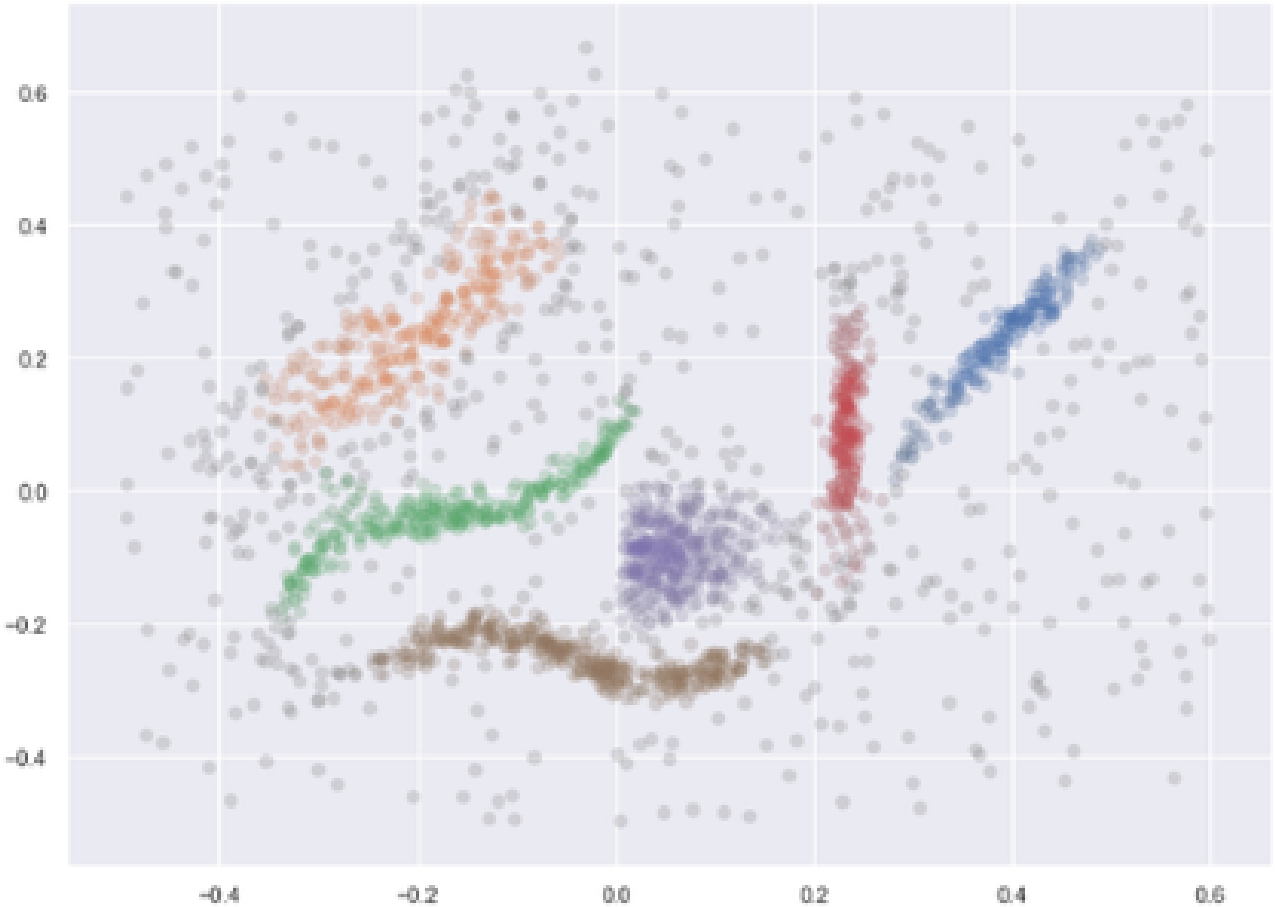
Paso 4: extracción de clusters



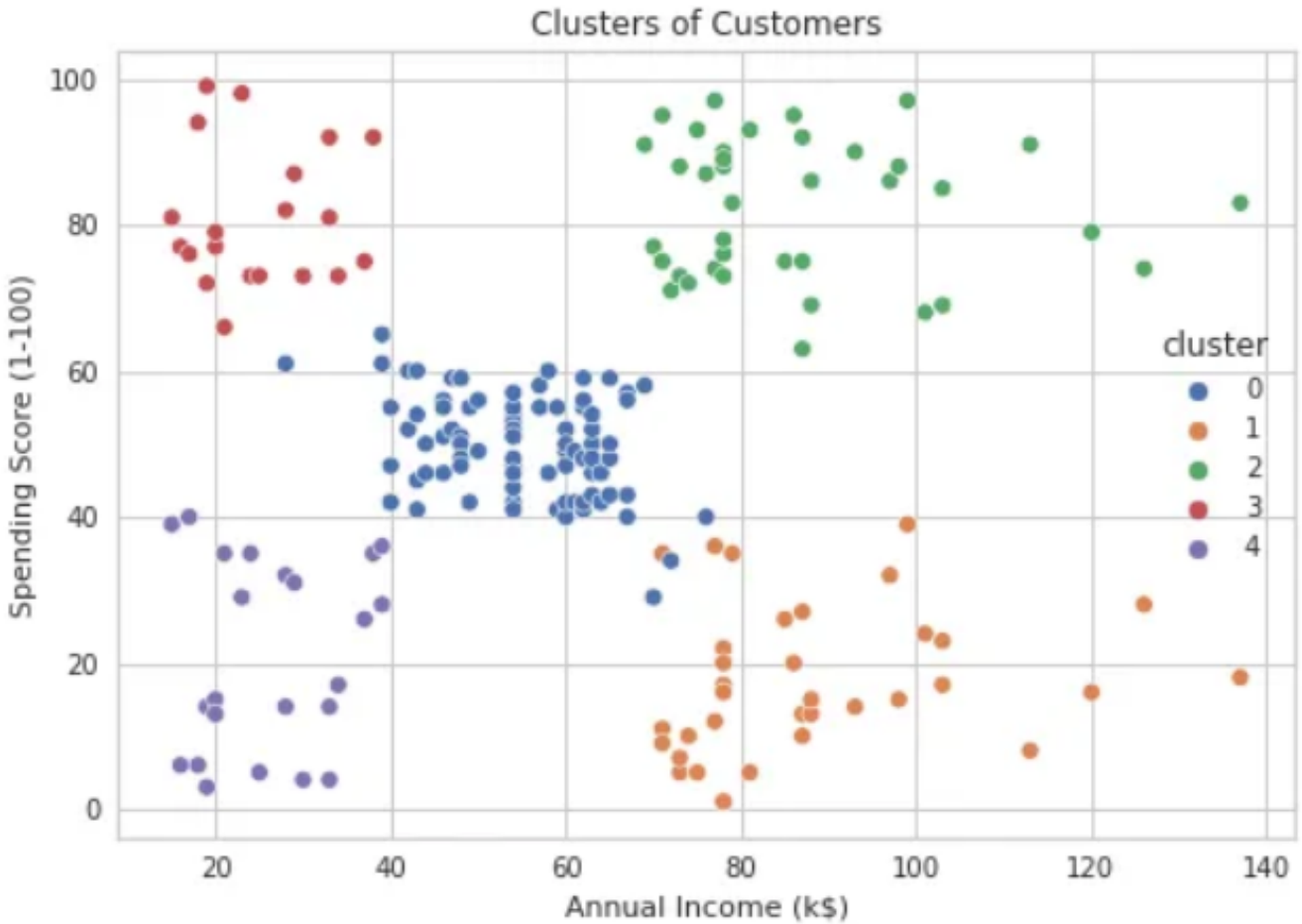




Ejemplo

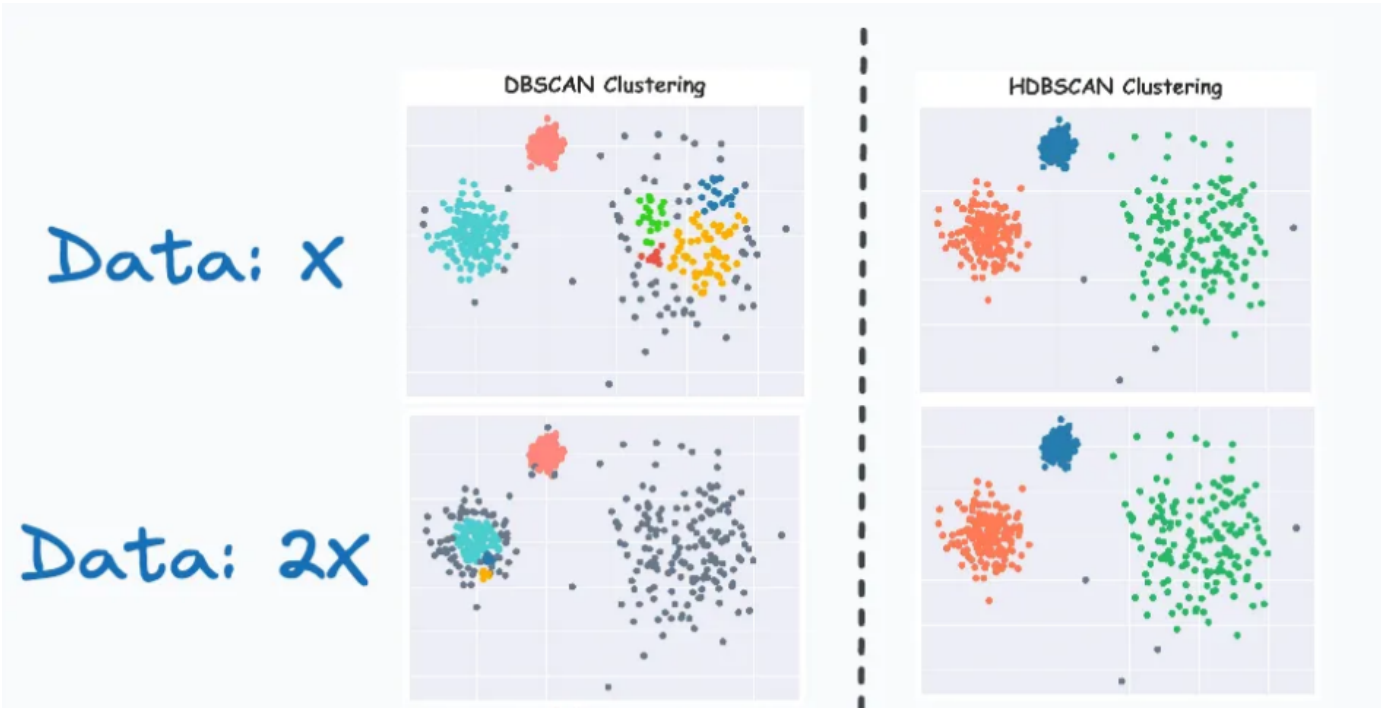


Ejemplo datos Mall



Comentarios

HDBSCAN es invariante a escalas, DBSCAN NO lo es.

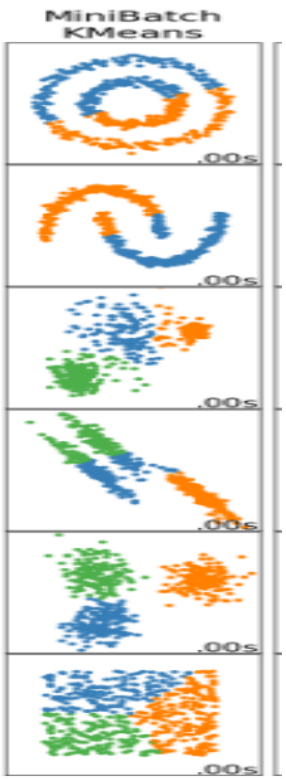


Comentarios

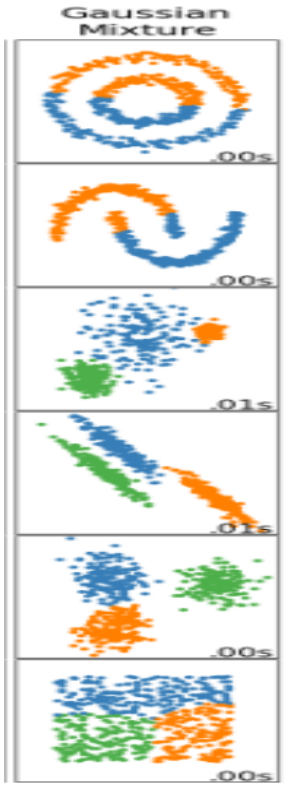
HDBSCAN:

- Puede lidiar con densidades variables
- Elige los mejores clusters automáticamente
- Si se sabe que la estructura del dataset es compleja, se sugiere usar HDBSCAN
- Es computacionalmente caro comparado con otros métodos de agrupamiento
- No es simple interpretar jerarquías.

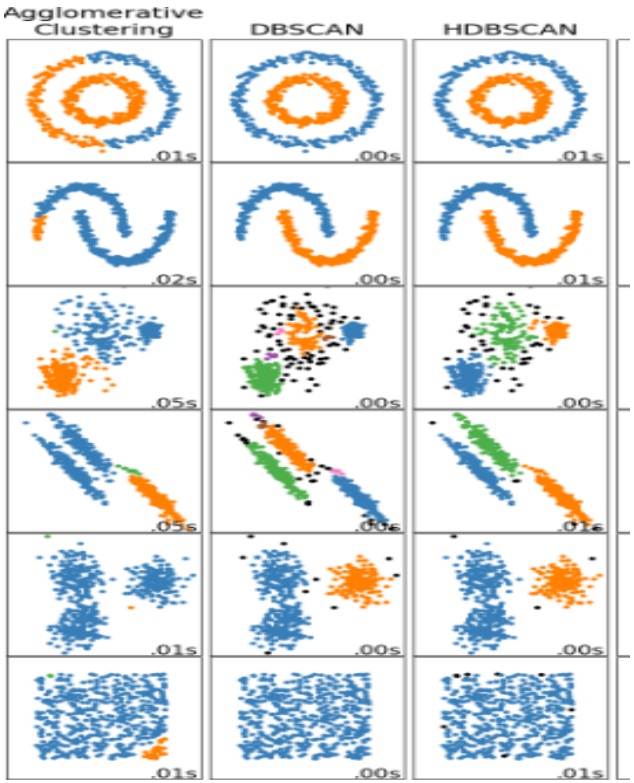
Desempeño de distintos algoritmos con “toy datasets”



Kmeans



GMM



Jerárquico, DBSCAN, HDBSCAN

Bibliografía

[1] Ester, M., H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231. 1996

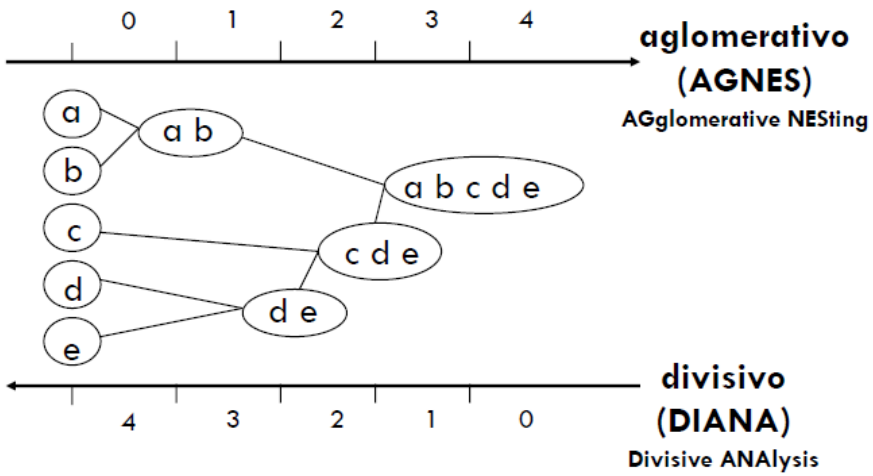
[2] Campello, R.J.G.B., Moulavi, D., Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. In Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science(), vol 7819.
https://doi.org/10.1007/978-3-642-37456-2_14

[3]https://pberba.github.io/stats/2020/01/17/hdbscan/

Esquema de agrupamiento jerárquico

Agrupamiento jerárquico

- Se generan sucesiones ordenadas de grupos:
  - juntando clusters pequeños en otros más grandes (aglomerativo)
  - diviendo grandes clusters en otros más pequeños (divisivo)
- La estructura jerárquica se representa en forma de árbol (dendograma).
- El análisis e interpretación del dendograma puede ayudar a determinar el número de clusters.



Esquema de agrupamiento jerárquico

Agrupamiento jerárquico aglomerativo

- Inicialmente cada punto es un cluster
- Se calcula la matriz de Distancias (similaridades) entre puntos.
- En cada iteración, y a partir de la matriz de distancias, los dos clusters más próximos se acuerdo al criterio seleccionado se **aglomeran** en un nuevo cluster.
- Se repite este procedimiento hasta obtener un único cluster.
- El dendograma es la represenación gráfica que muestra los pasos del procedimiento.

Criterios de aglomeración (distancias **entre** grupos): vecino más próximo, vecino más lejano, centroide, etc.

Criterios de aglomeración

A, B clusters con n_a y n_b elementos respectivamente

- Vecino más próximo: $D(A,B) = \min\{d(\mathrm{x}_i, \mathrm{x}_j), \mathrm{x}_i enA, \mathrm{x}_j enB\}$.
- Vecino más lejano: $D(A,B) = \max\{d(\mathrm{x}_i, \mathrm{x}_j), \mathrm{x}_i enA, \mathrm{x}_j enB\}$.
- Media de grupos (centroide): $D(A,B) = \frac{1}{n_A n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} d(\mathrm{x}_i, \mathrm{x}_j)$
- Método del centroide: $D(A,B) = d(\bar{\mathrm{x}}_A, \bar{\mathrm{x}}_B)$, con $\bar{\mathrm{x}}_A$ y $\bar{\mathrm{x}}_B$ vectores de medias de los vectores en A y B respectivamente. Una vez aglomerados dos clusters A y B se computa el centroide del nuevo cluster haciendo: $\bar{\mathrm{x}}_{AB} = \frac{n_A \bar{\mathrm{x}}_A + n_B \bar{\mathrm{y}}_B}{n_A + n_B}$

Criterios de aglomeración: Método de Ward

- Utiliza las distancias intra cluster e inter clusters.
- Aglomera los clusters que minimizan

$$I_{AB} = SSE_{AB} - (SSE_A + SSE_B)$$

  - $SSE_A = \sum_{i=1}^{n_A} (\mathrm{x}_i - \bar{\mathrm{x}}_A)'(\mathrm{x}_i - \bar{\mathrm{x}}_A)$
  - $SSE_B = \sum_{i=1}^{n_B} (\mathrm{x}_i - \bar{\mathrm{x}}_B)'(\mathrm{x}_i - \bar{\mathrm{x}}_B)$
  - $SSE_{AB} = \sum_{i=1}^{n_{AB}} (\mathrm{x}_i - \bar{\mathrm{x}}_{AB})'(\mathrm{x}_i - \bar{\mathrm{x}}_{AB})$
- Puede verse que $I_{AB} = \frac{n_A n_B}{n_A + n_B} (\bar{\mathrm{x}}_A - \bar{\mathrm{x}}_B)'(\bar{\mathrm{x}}_A - \bar{\mathrm{x}}_B)$ (Minimizar I_{AB} es equivalente a minimizar la distancia inter clusters)

Dendograma (Arbol Jerárquico)

Representación gráfica del resultado del proceso de agrupamiento en forma de árbol.

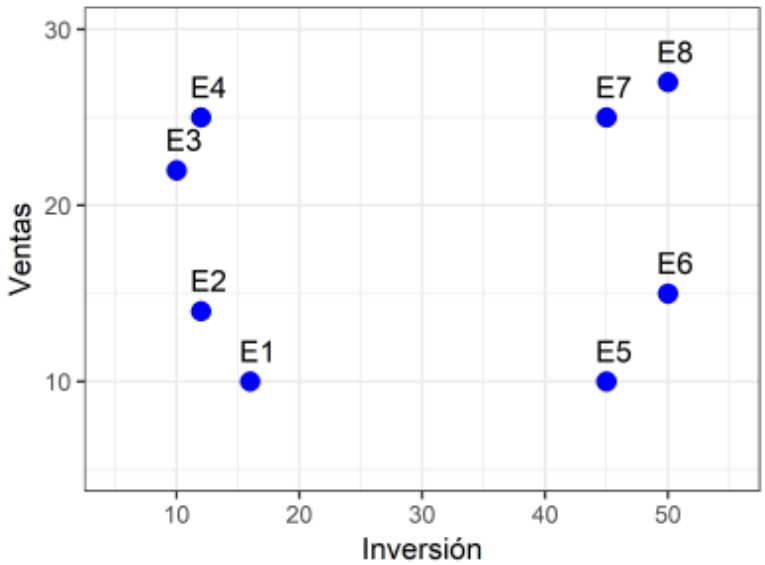
dendograma

1. En la parte inferior del gráfico se disponen los n elementos iniciales.
2. Se unen elementos por tres líneas rectas:
   - dos dirigidas a los elementos que se unen y que son perpendiculares al eje de los elementos
   - una paralela a este eje que ubica al nivel en que se unen.
3. El proceso se repite hasta que todos los elementos estén conectados por líneas rectas.

Cortando el dendrograma a un nivel de distancia dado, se obtiene una clasificación del número de grupos existentes a ese nivel y los elementos que los forman.

Ejemplo - Paso 1

Empresa	Inversión	Ventas
E1	16	10
E2	12	14
E3	10	22
E4	12	25
E5	45	10
E6	50	15
E7	45	25
E8	50	27



Inicialmente: cada dato es un grupo

Ejemplo: 1^{er} aglomerado

Grupo	Inversión	Ventas
E1	16	10
E2	12	14
E3	10	22
E4	12	25
E5	45	10
E6	50	15
E7	45	25
E8	50	27

Centroide del grupo E34:

$$\text{Inversión}(E34) = \frac{10+12}{2}$$
$$\text{Ventas}(E34) = \frac{22+25}{2}$$

Datos para paso 2:

Grupo	Inversión	Ventas
E1	16	10
E2	12	14
E34	11	23.5
E5	45	10
E6	50	15
E7	45	25
E8	50	27

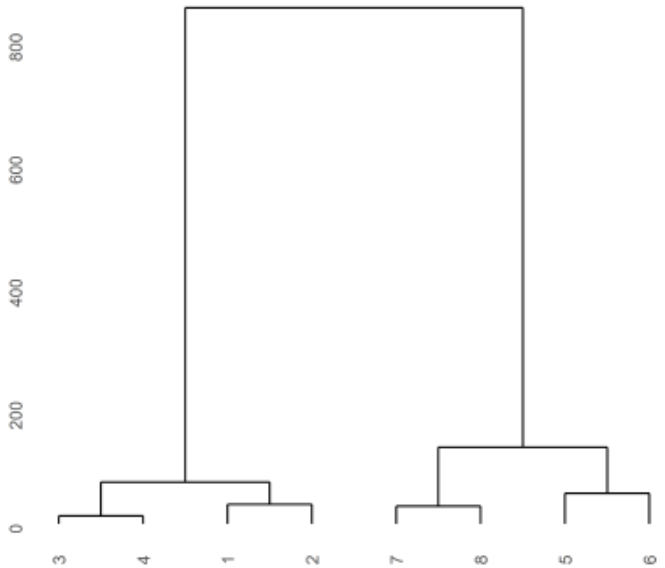
Matriz distancias euclídeas al cuadrado

	E1	E2	E34	E5	E6	E7	E8
E1	0						
E2	32	0					
E34	207	91	0				
E5	841	1105	1338	0			
E6	1181	1445	1593	50	0		
E7	1066	1210	1158	225	125	0	
E8	1445	1613	1533	314	144	29	0

Ejemplo - Pasos subsiguientes

Se sigue con el mismo procedimiento hasta llegar a un único cluster. Y se construye el dendograma con el historial de conglomeración

	height	merge.1	merge.2
1	13	-3	-4
2	29	-7	-8
3	32	-1	-2
4	50	-5	-6
5	68	1	3
6	125	2	4
7	841	5	6



- height: distancia a la que se fueron generando las respectivas agrupaciones
- merge.1 y merge.2 grupos que se unieron en cada etapa.

Ejemplo - Alturas usando vecino más cercano

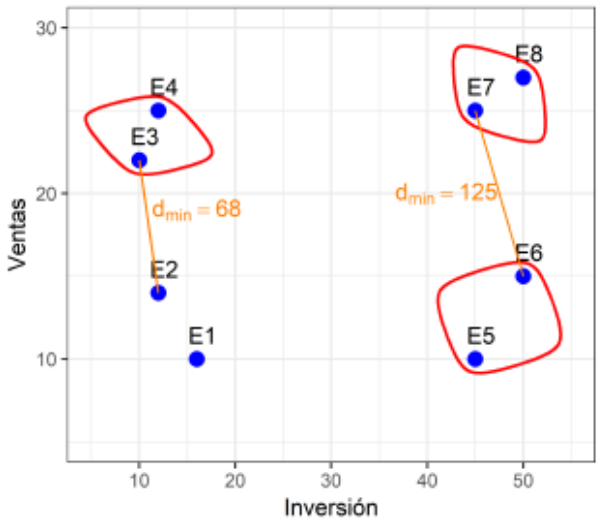


Figura: Ejemplo 8 empresas

Matriz de distancias euclídeas al cuadrado

	E1	E2	E3	E4	E5	E6	E7	E8
E1	0							
E2	32	0						
E3	180	68	0					
E4	241	121	13	0				
E5	841	1105	1369	1314	0			
E6	1181	1445	1649	1544	50	0		
E7	1066	1210	1234	1089	225	125	0	
E8	1445	1613	1625	1448	314	144	29	0

$$d_{(2,3-4)} = min\{d_{(2,3)}, d_{(2,4)}\} = 68$$

$$d_{(5-6,7-8)} = min\{d_{(5,7)}, d_{(5,8)}, d_{(6,7)}, d_{(6,8)}\} = 125$$