

INTRODUCCIÓN A LA CIENCIA DE DATOS



UNSUPERVISED LEARNING

Unsupervised learning

El aprendizaje no supervisado (unsupervised learning) es una de las tres tareas más comunes de machine learning junto con supervised learning y reinforcement learning.

En un problema de unsupervised learning no tenemos un atributo objetivo que predecir, quedando a merced del algoritmo encontrar la estructura de los datos. El objetivo principal de este tipo de problemas es generalmente **encontrar patrones ocultos en los datos o ser un medio para un fin**, como el aprendizaje de atributos.

Unsupervised learning

Con frecuencia, el objetivo principal es encontrar patrones en los datos, siendo nuestros datos de entrenamiento una serie de atributos sin ninguna columna objetivo correspondiente. Es decir, no tendremos una “verdad absoluta” que predecir. Se suele decir que el aprendizaje no supervisado es “aprender sin un profesor”.

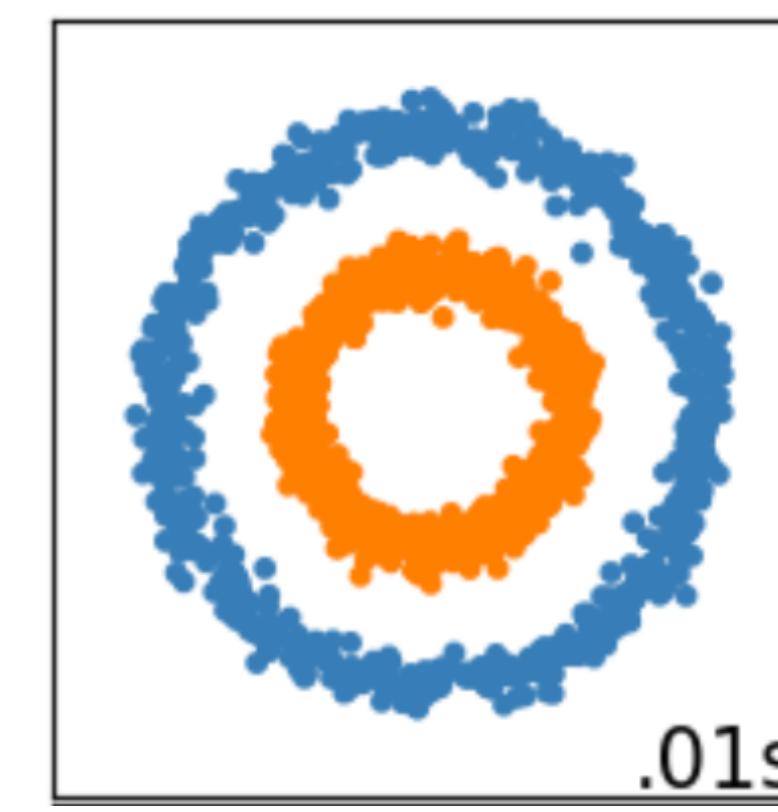
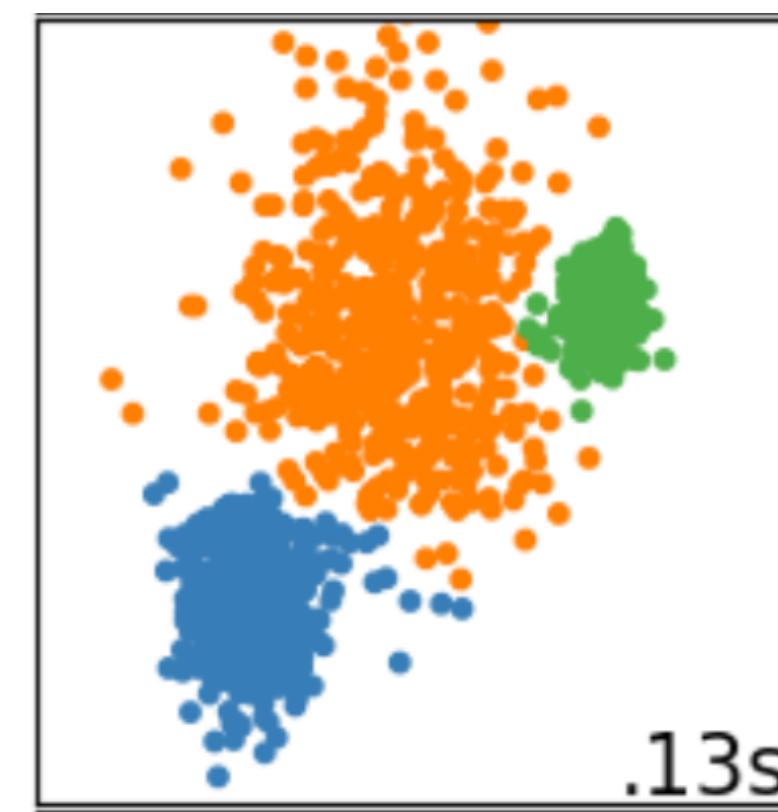
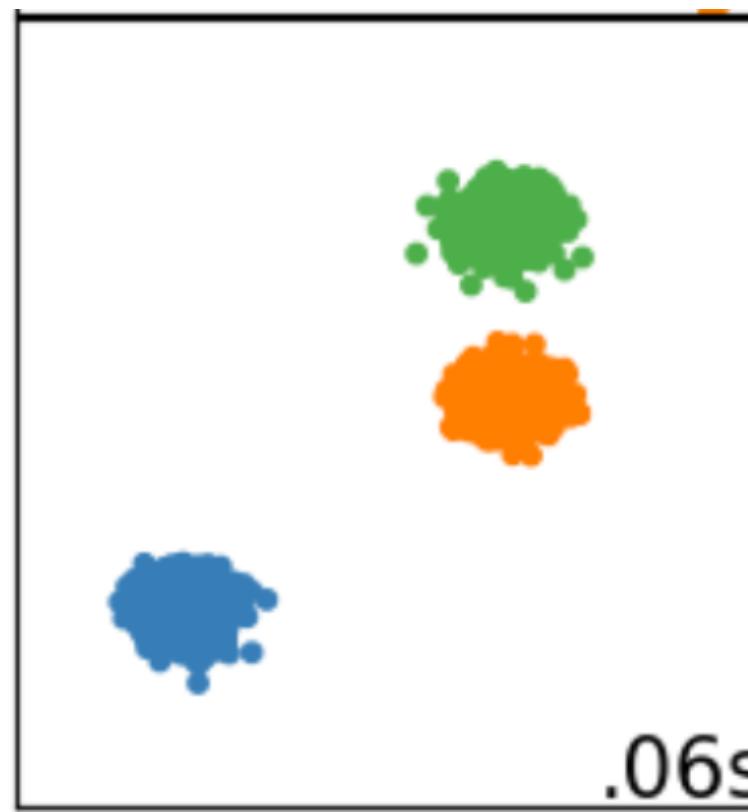
Existen dos grandes problemas dentro de unsupervised learning:

- 1. Aprendizaje no supervisado paramétrico:** En este escenario se asume que la distribución de los datos es paramétrica, es decir, se asume que los datos provienen de una distribución de probabilidades conocida y basada en un conjunto de parámetros fijos. Por ejemplo, una distribución paramétrica conocida es la distribución normal, cuyos parámetros son la media y la desviación estándar, conocidos estos dos parámetros conocemos la forma de la distribución.

- 2. Aprendizaje no supervisado no paramétrico:** En este escenario los datos son agrupados en grupos, también conocidos como **clusters**. El objetivo principal es que cada cluster forme un grupo bien definido a partir de las categorías presentes en los datos. Una gran diferencia con el aprendizaje no supervisado paramétrico es que no hace ninguna suposición de la distribución de los datos. Por esta razón es que es conocido como un método libre de distribución.

Clustering

En este curso nos enfocaremos principalmente en **clustering** ya que es el problema más frecuente e importante dentro de unsupervised learning. Cuando hacemos clusters de los datos buscamos organizar los objetos/registros en grupos cuyos miembros sean lo más símiles entre si y los más disímiles posibles a los miembros de otros grupos.



Distintos tipos de clusters

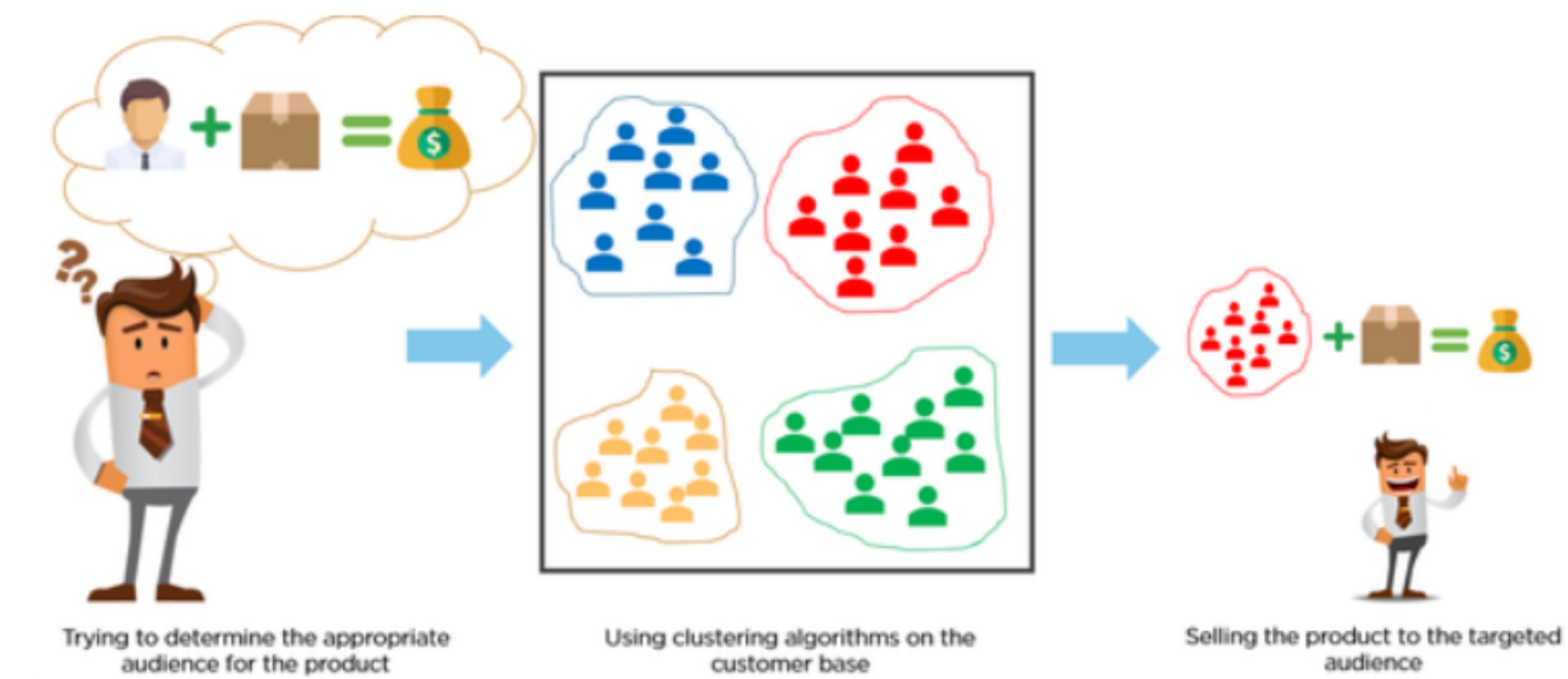
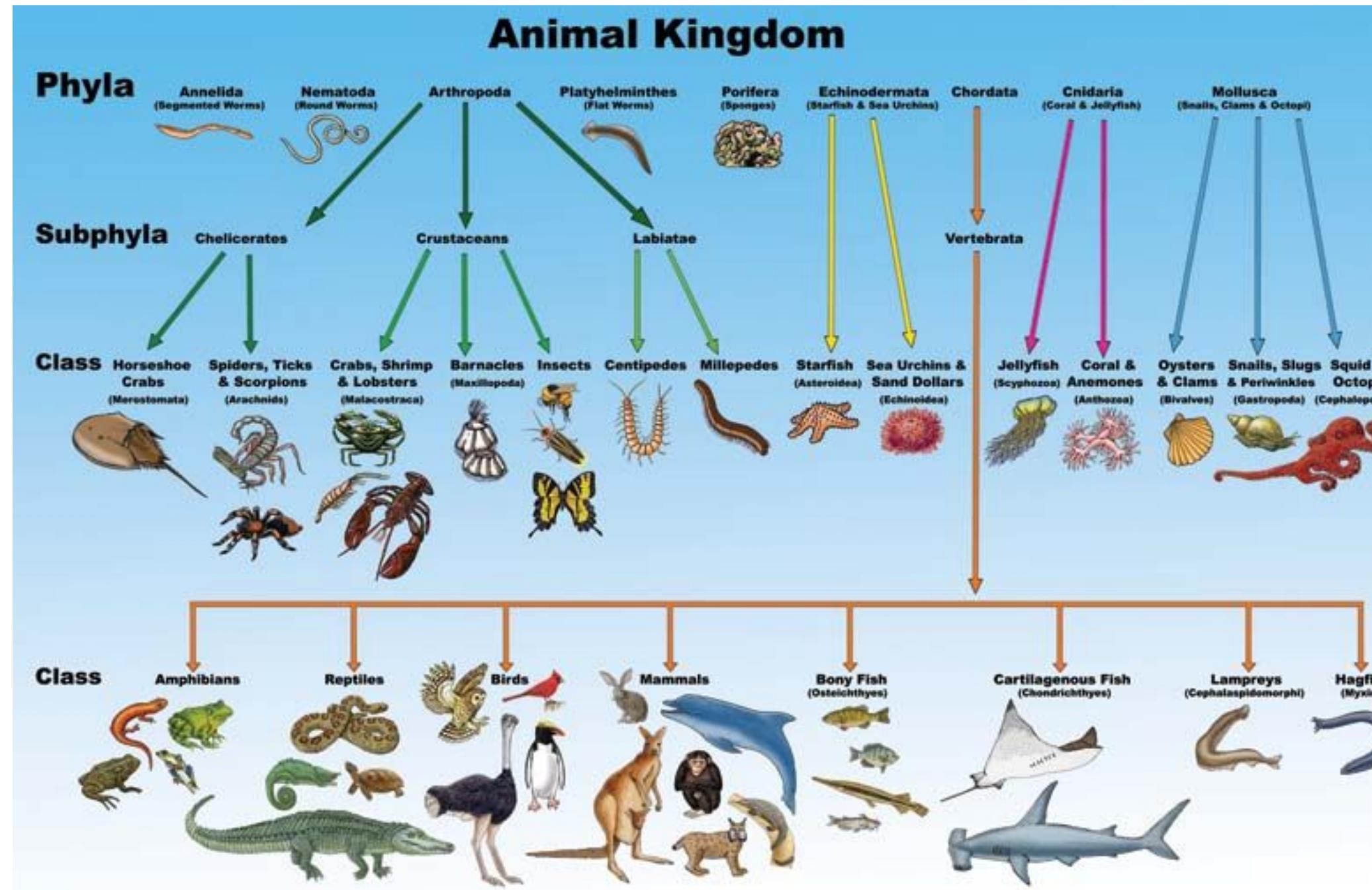
Clustering

Clustering divide los datos en grupos, los cuales deben ser significativos, útiles o ambos.

Las técnicas de clustering se utilizan generalmente con dos fines:

1. Entendimiento: los clusters capturan la naturaleza de la estructura de los datos:

- **Biología:** taxonomía (jerárquica) de los seres vivos, analizar grandes volúmenes de información genética.
- **Information retrieval:** los clusters capturan aspectos particulares de una query (ejemplo: la búsqueda “película” puede agrupar búsquedas relacionadas como “actor”, “trailers”, “cines”, etc).
- **Negocios:** agrupar clientes actuales y potenciales para un análisis más profundo y enfocar así el marketing.



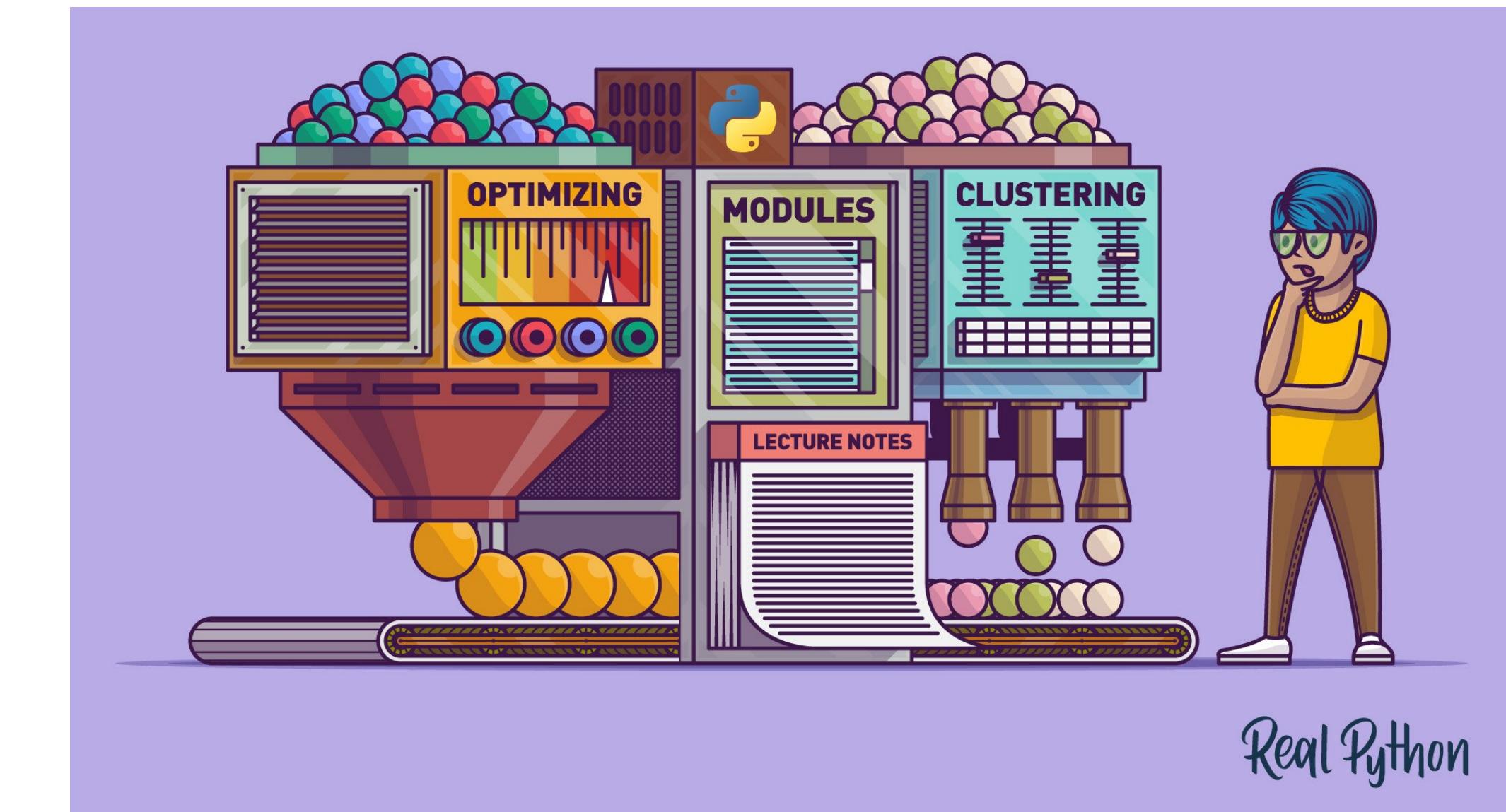
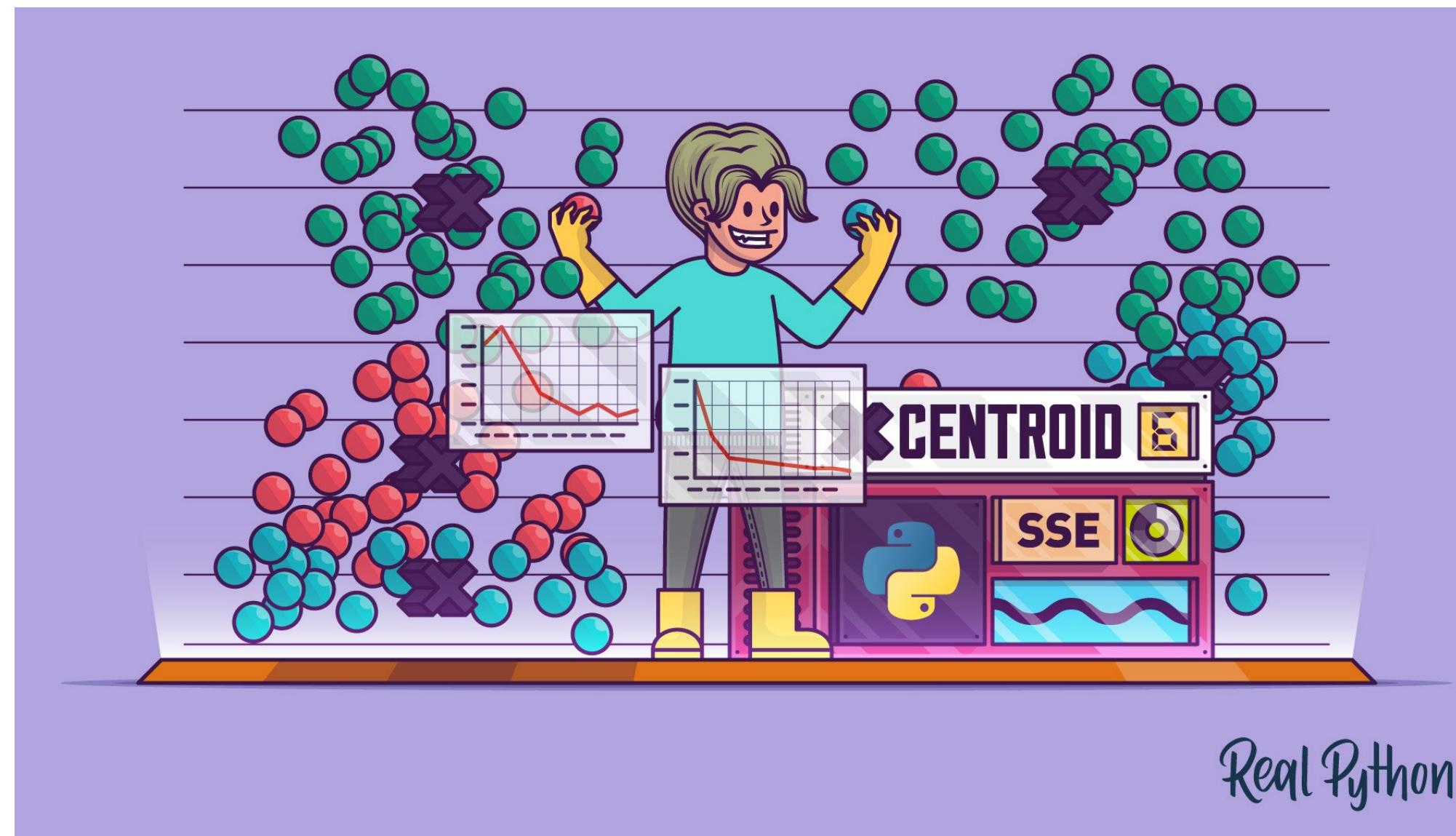
Clustering

Clustering divide los datos en grupos, los cuales deben ser significativos, útiles o ambos.

Las técnicas de clustering se utilizan generalmente con dos fines:

2. Utilidad: obtener una abstracción de los objetos individuales utilizando clusters.

- **Resumir:** existen modelos que son muy complejos y requieren mucho tiempo de cómputo ($O(n^2)$), no siendo práctico para datasets muy grandes. Una buena estrategia es resumir los datos en clusters, utilizando las características en común del grupo como una unidad.
- **Comprimir:** usualmente aplicado a datos multimedia como audios, imágenes y videos. Es aceptable si hay muchos objetos similares, es aceptable perder información y si se requiere una reducción.



Clustering

En general, los clusters no están bien definidos:



Distintos:

- Animales.
- Colores.
- Poses.
- Emociones.
- Fondos de imagen.

Clustering

En general, los clusters no están bien definidos:

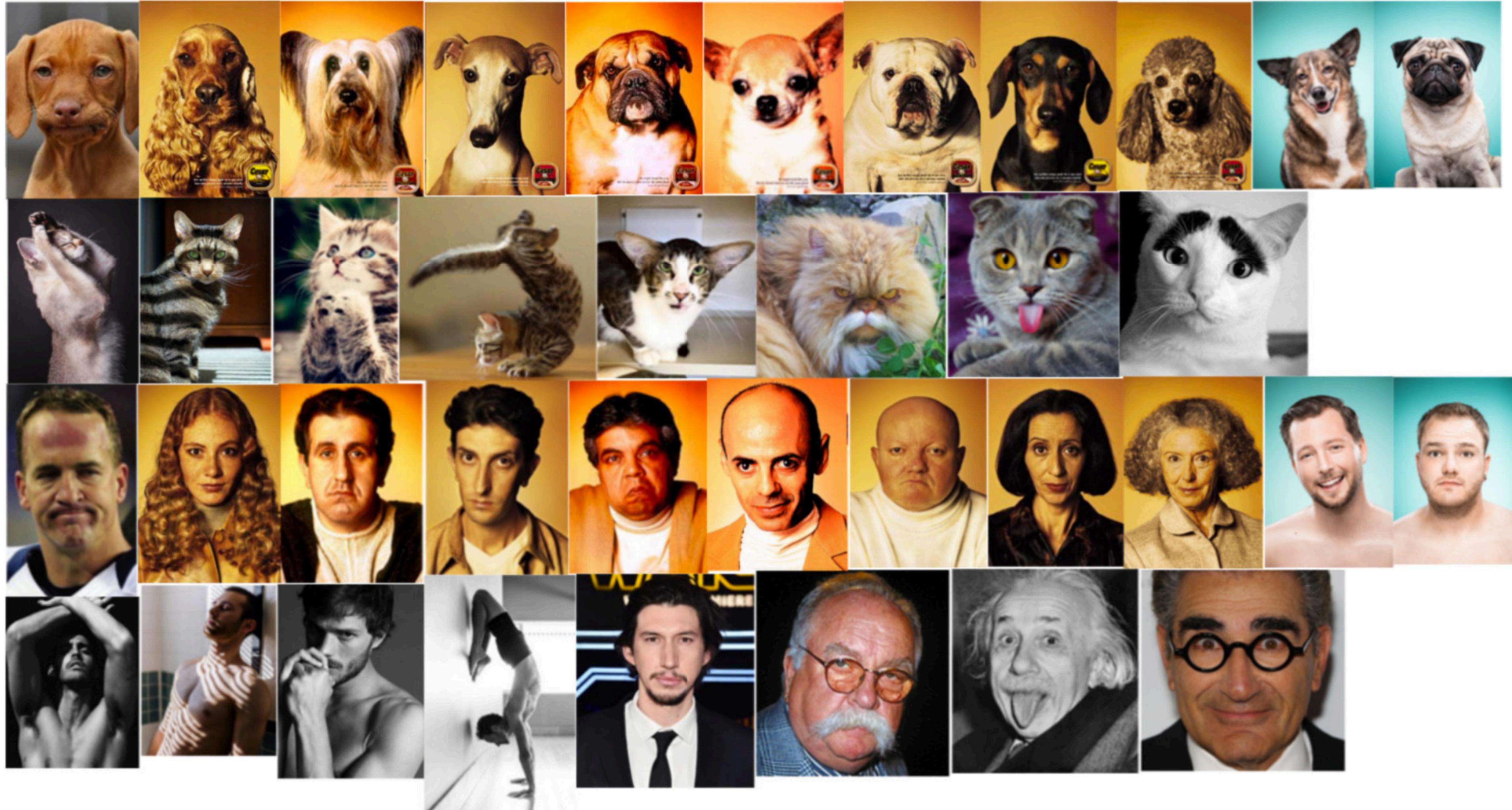


Distintos:

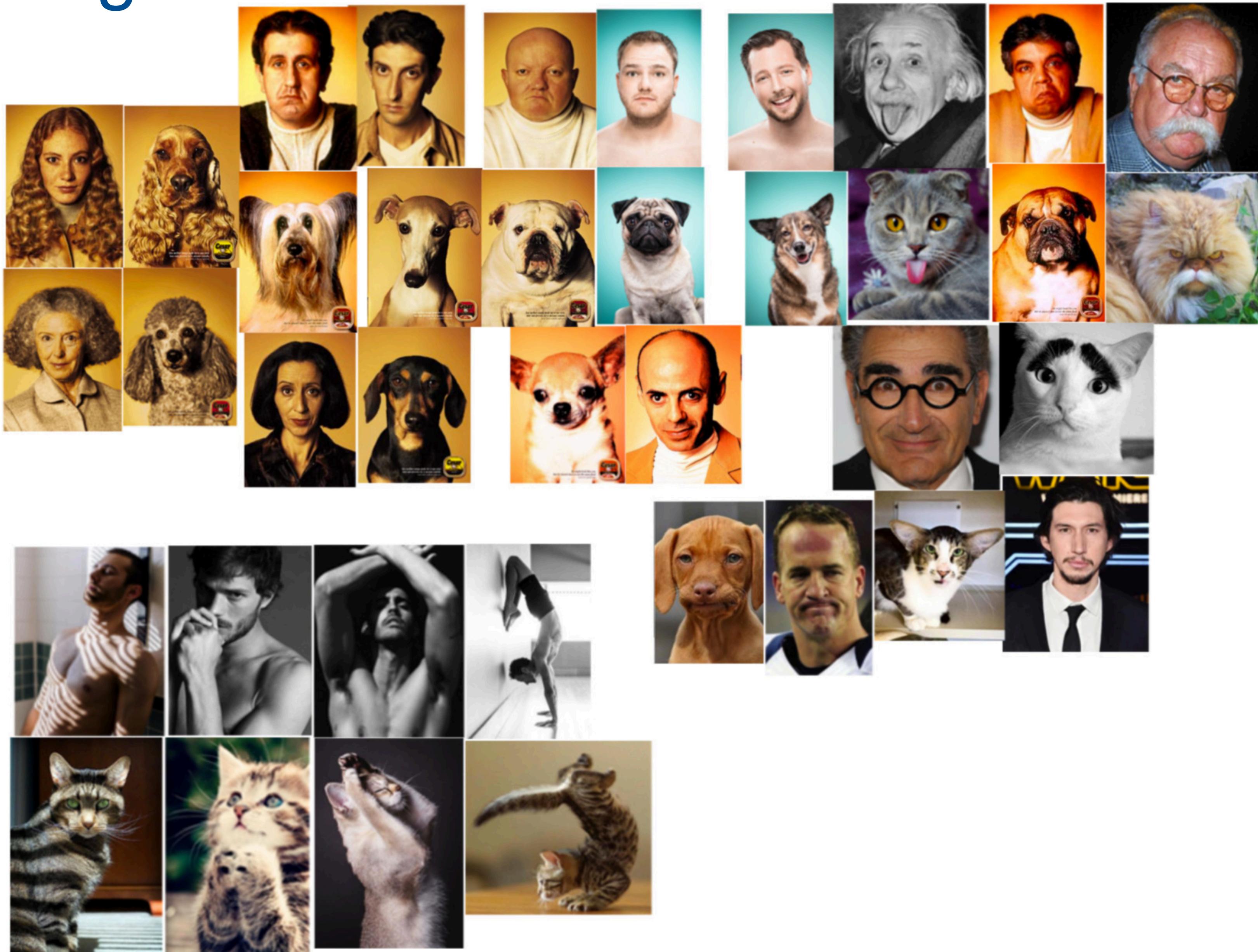
- Animales.
- Colores.
- Poses.
- Emociones.
- Fondos de imagen.

Clustering

En general, los clusters no estan bien definidos:



Clustering



Tipos de clustering

1. **Hierarchical (nested) vs. Partitional (unnested):**

- *Herarchical: clusters anidados que se organizan como un árbol.*
- *Partitional: se dividen los datos en conjuntos disjuntos.*

2. **Exclusive vs. Overlapping vs. Fuzzy:**

- *Exclusive: cada objeto es asignado a un único cluster.*
- *Overlapping: los objetos pueden pertenecer a varios clusters a la vez.*
- *Fuzzy: cada objeto tiene un “peso” asociado a cada cluster.*

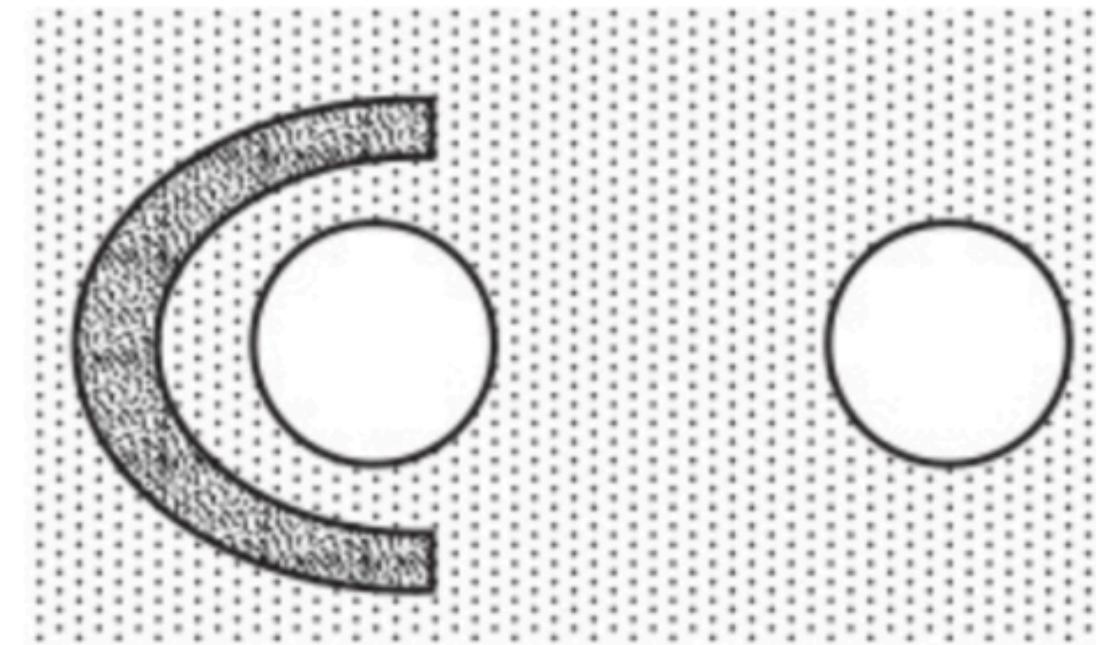
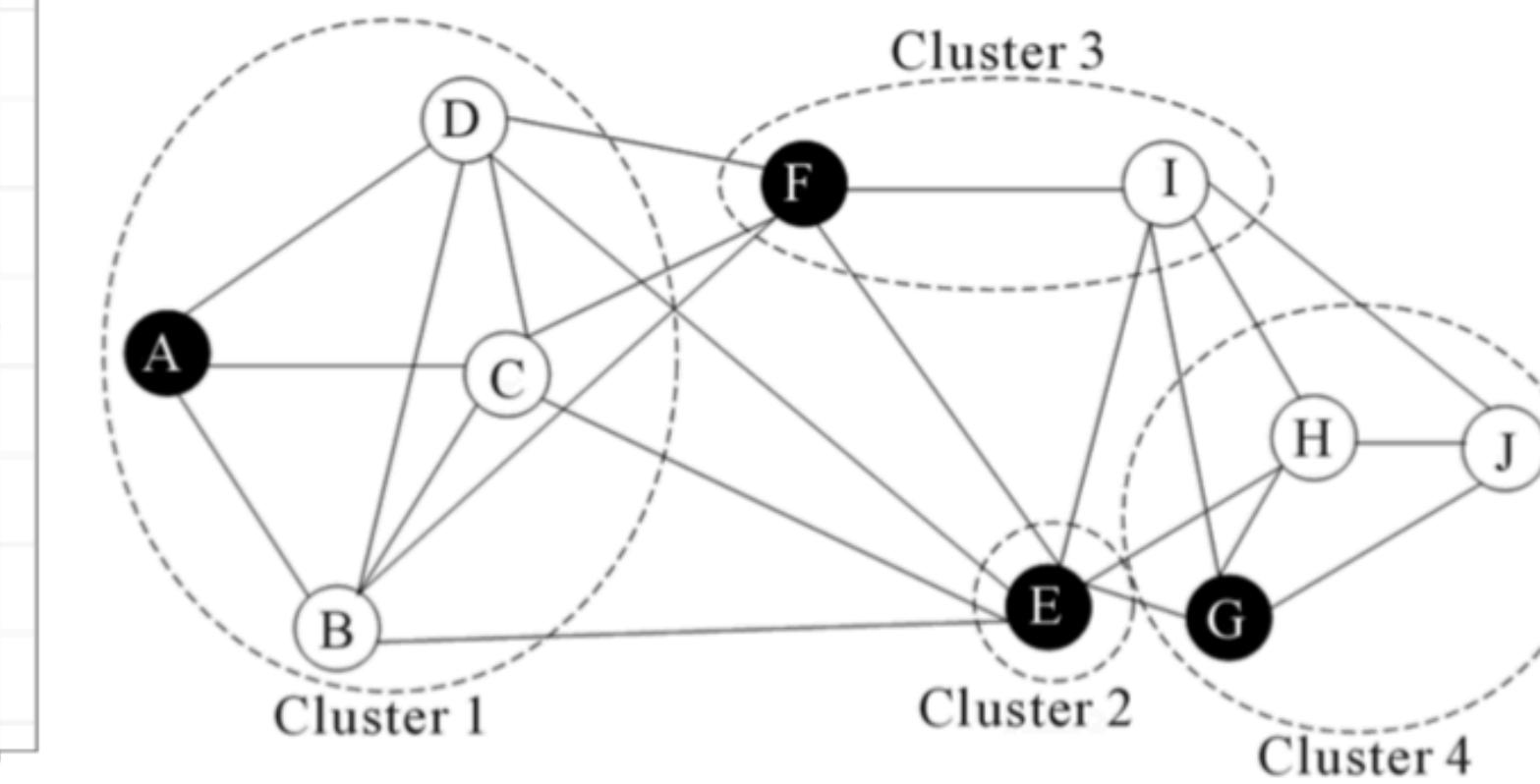
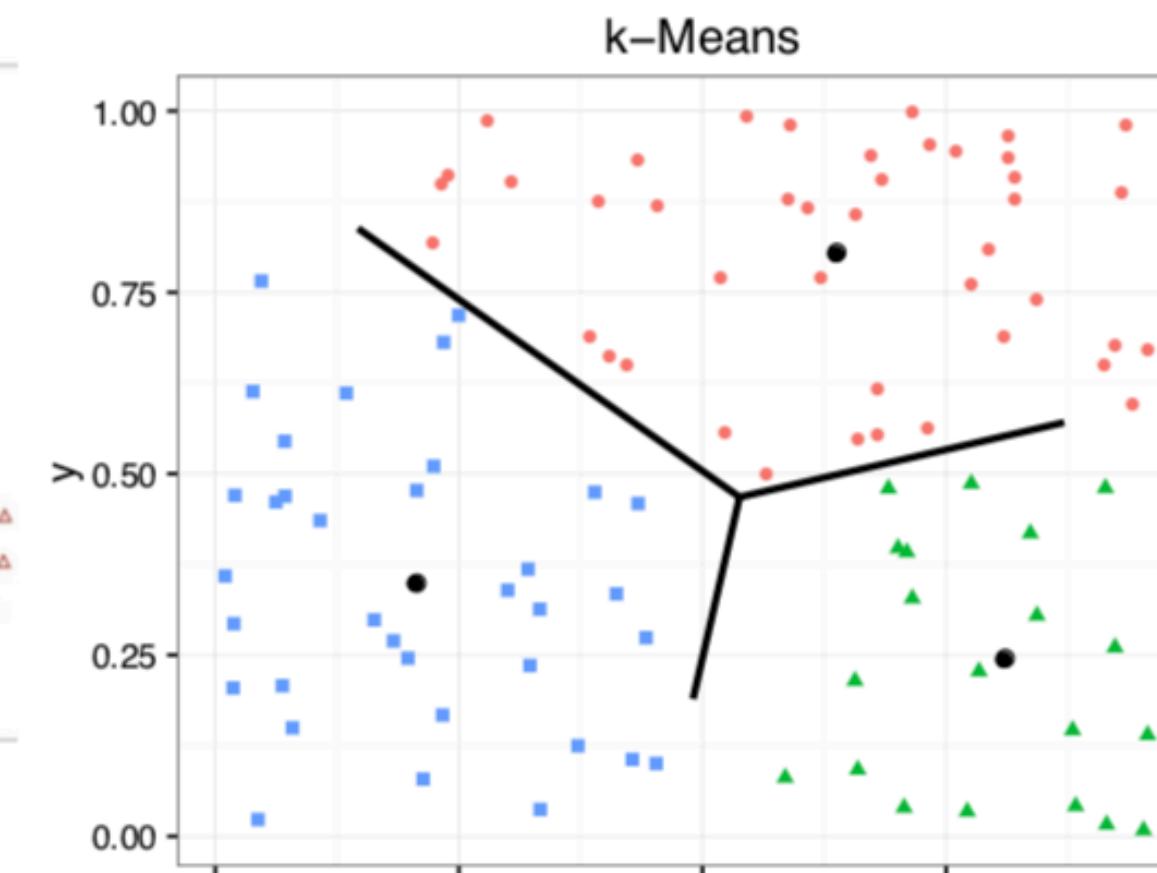
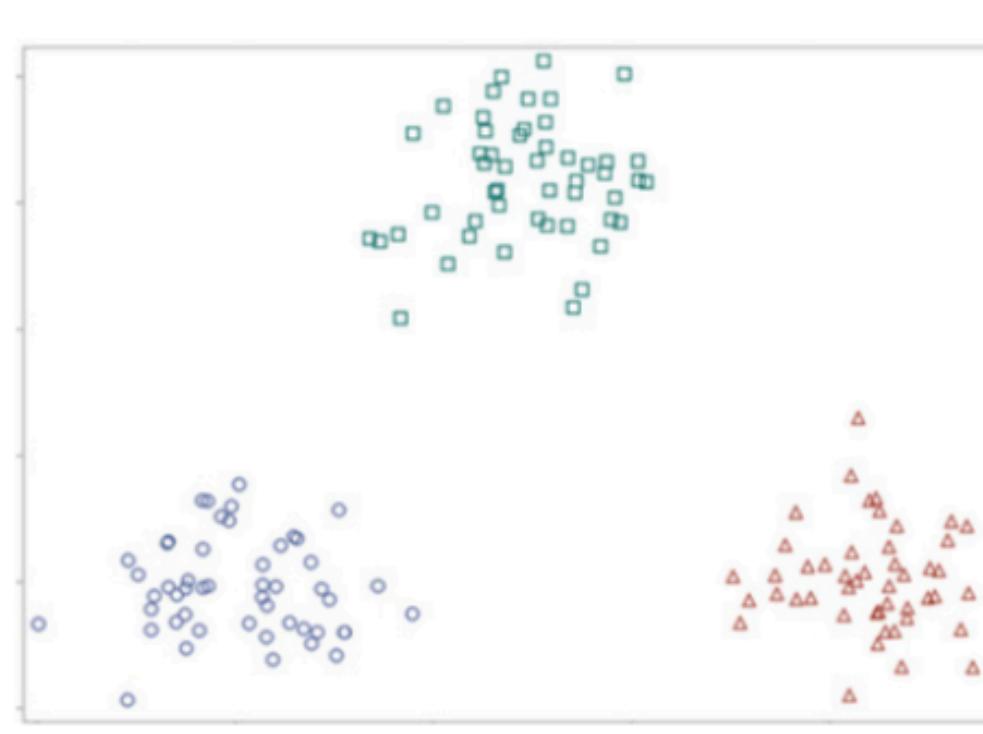
3. **Complete vs. Partial:**

- *Complete: el algoritmo de clustering le asigna un cluster a cada objeto.*
- *Partial: algunos objetos en los datos pueden no pertenecer a ningún grupo (por ejemplo, outliers o ruido).*

Tipos de clustering

Los distintos tipos de clusterings están asociados al fin que buscamos:

- **Bien definidos:** agrupar objetos en grupos bien definidos.
- **Prototype-based:** agrupar objetos respecto a un cluster prototipo.
- **Graph-based:** los nodos son objetos y los links representan las conexiones entre objetos. El cluster puede ser definido como un componente conexo.
- **Density-based:** los clusters son definidos por su densidad, quedando regiones de mayor densidad separadas de regiones de menor densidad.



Hierarchical Clustering

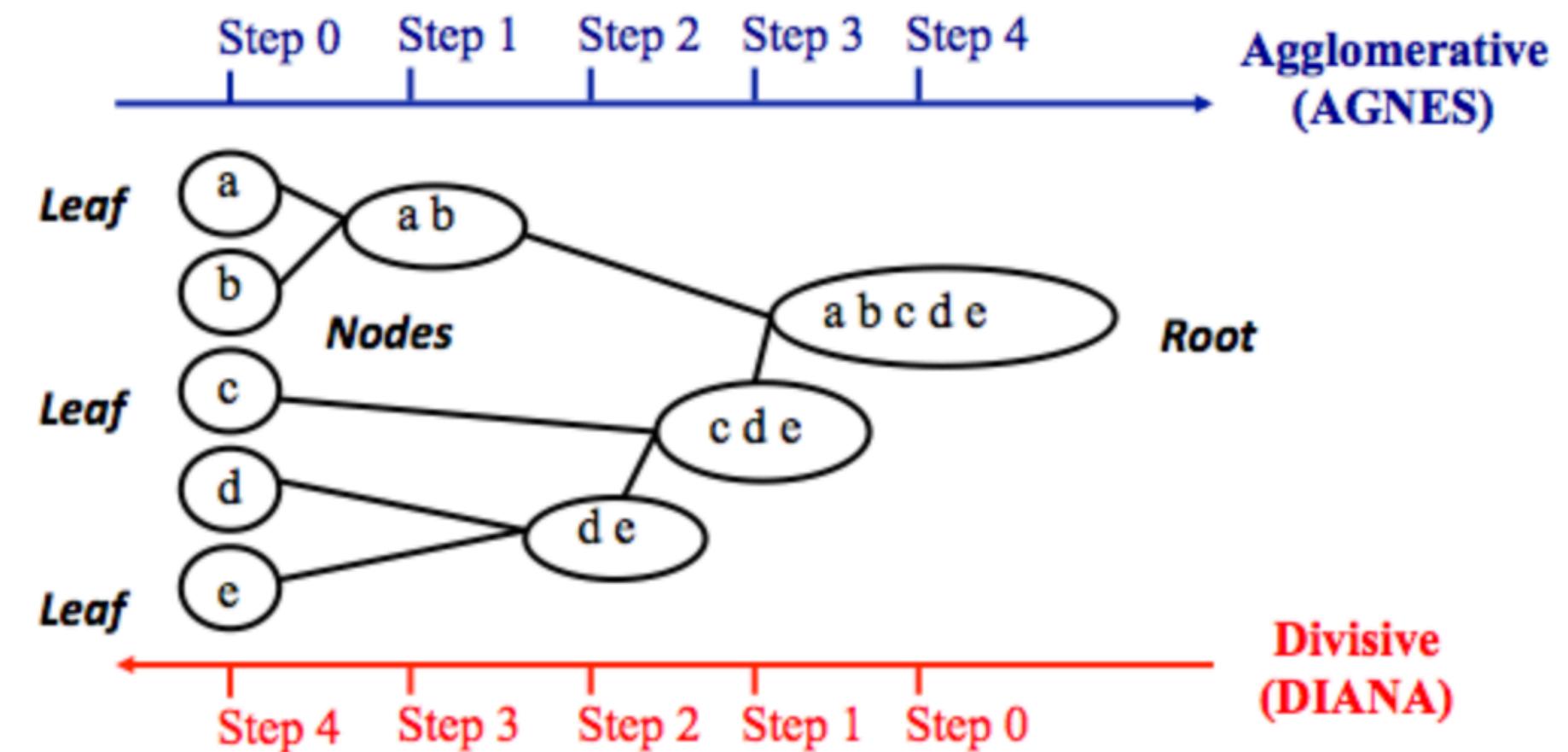
Dos tipos:

1. **Divisorios:** se comienza por un único cluster y se divide hasta que permanezcan solo puntos individuales.
2. **Aglomerativos:** se comienza por un punto y se procede uniendo puntos.

Algoritmo básico de clustering aglomerativo:

Data: $x \in \mathbb{R}^d$

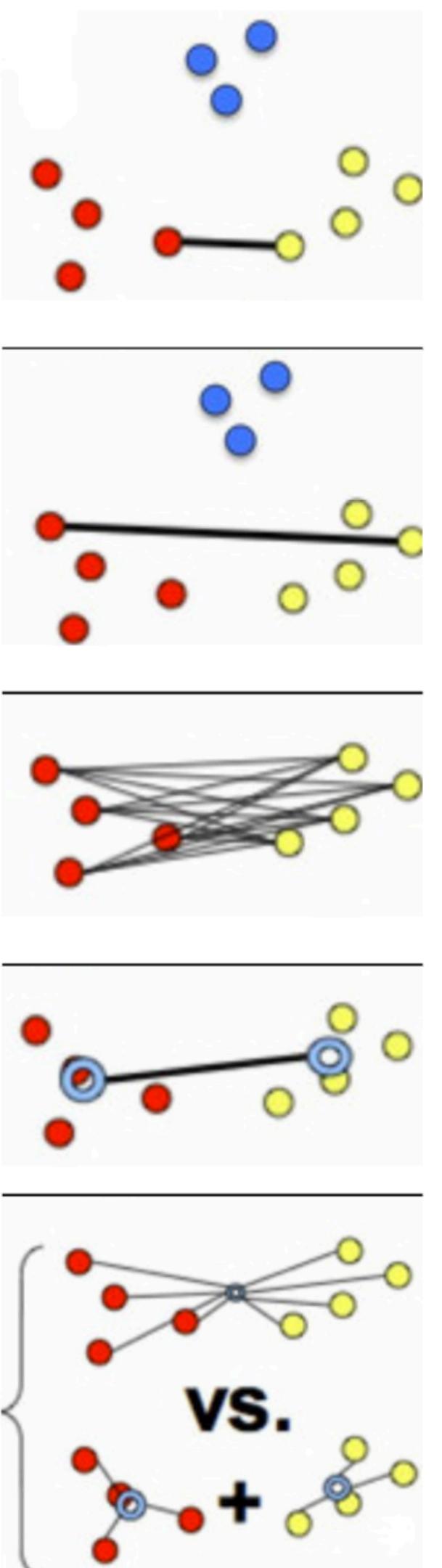
1. Calcular la matriz de proximidad D
2. **Repetir 3 y 4:**
3. Unir los dos cluster más próximos.
4. Actualizar la matriz de proximidad D para reflejar los clusters nuevos y los clusters originales
5. **Hasta** que solo permanezca un único cluster.



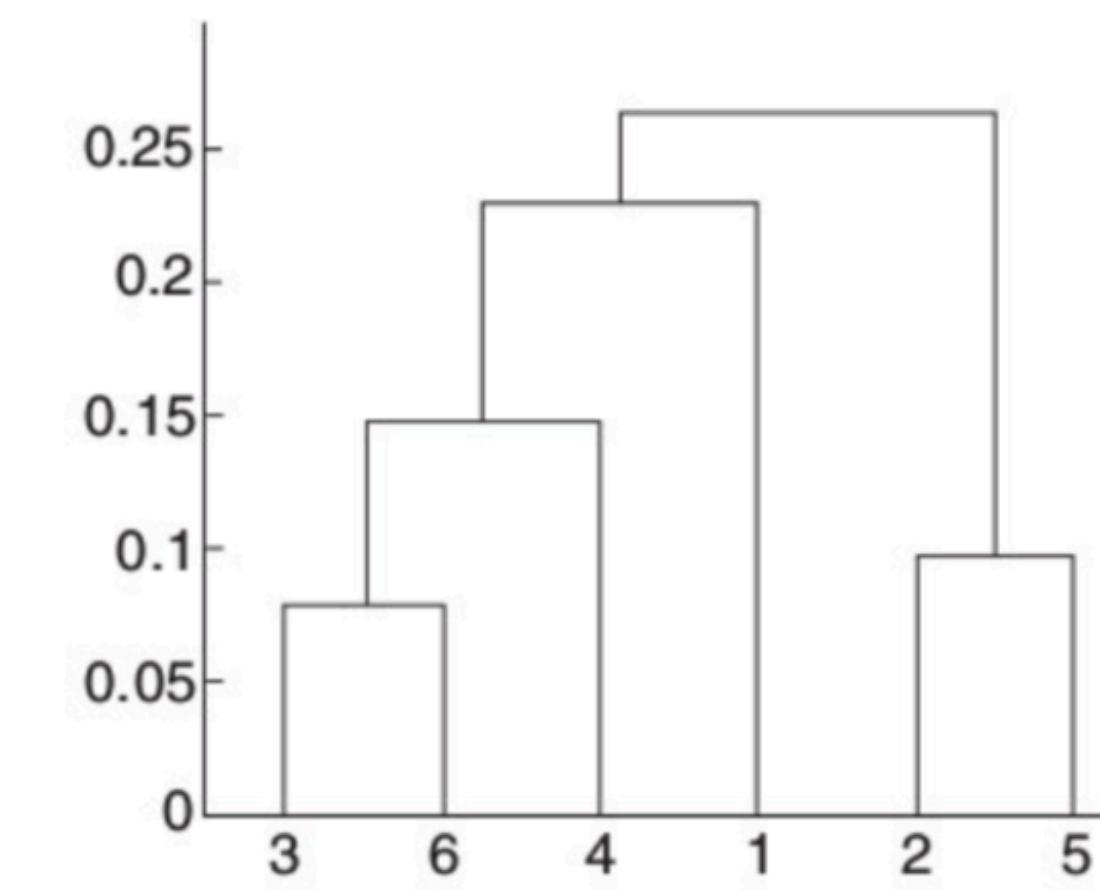
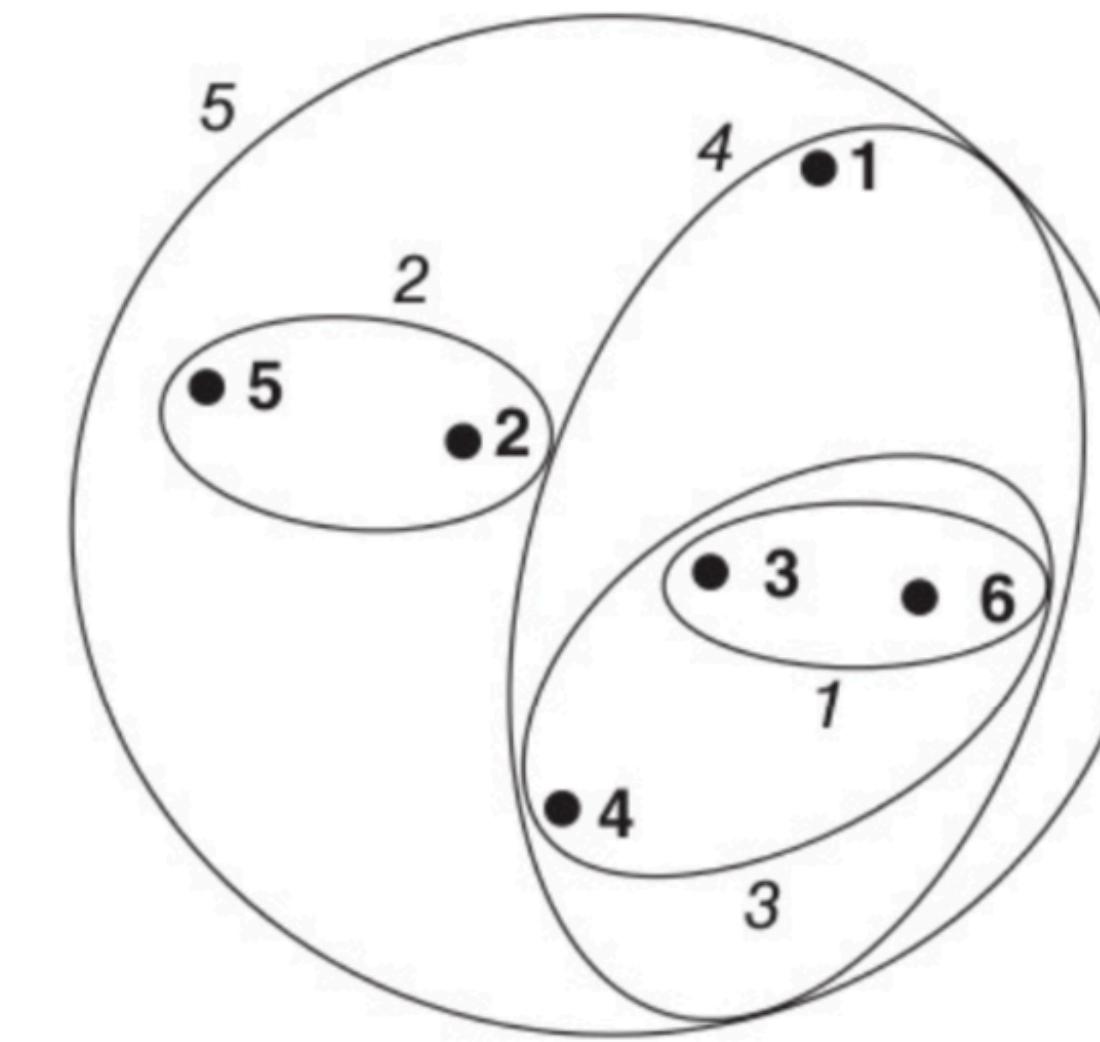
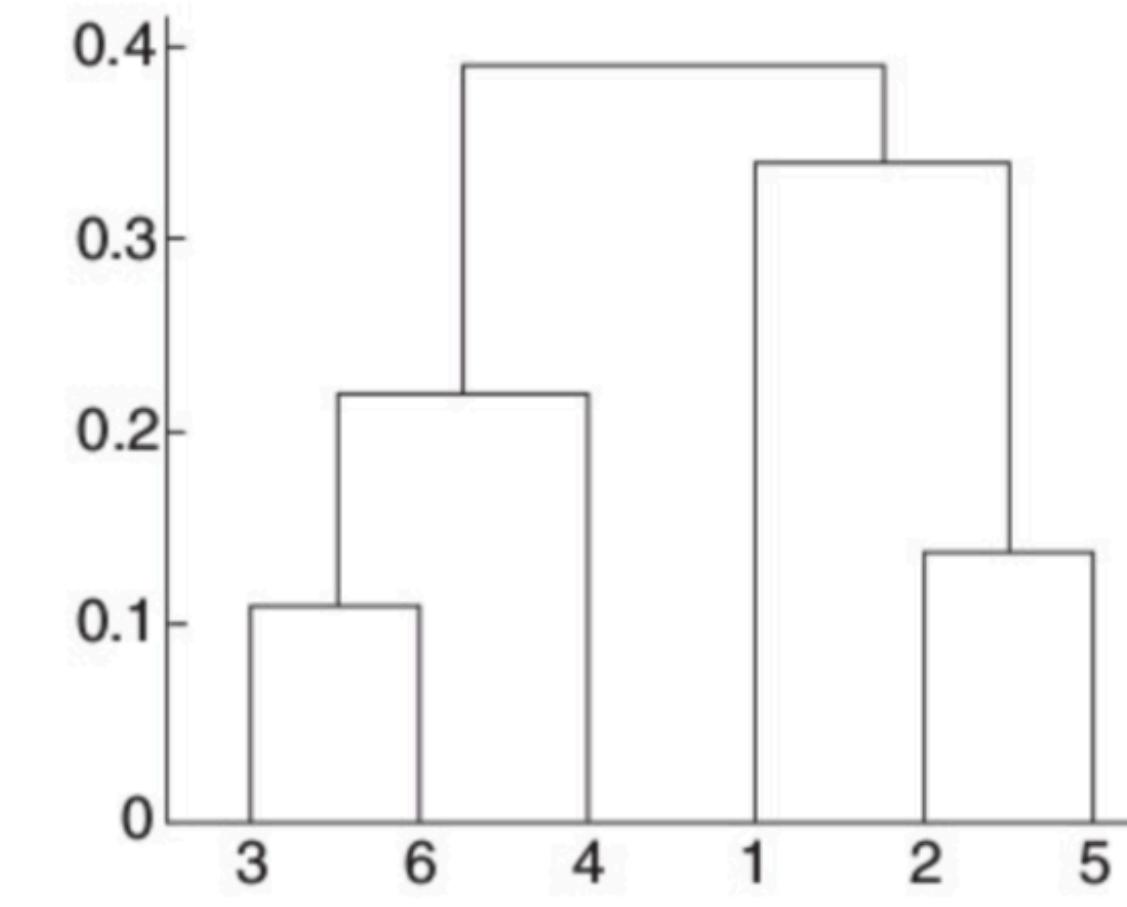
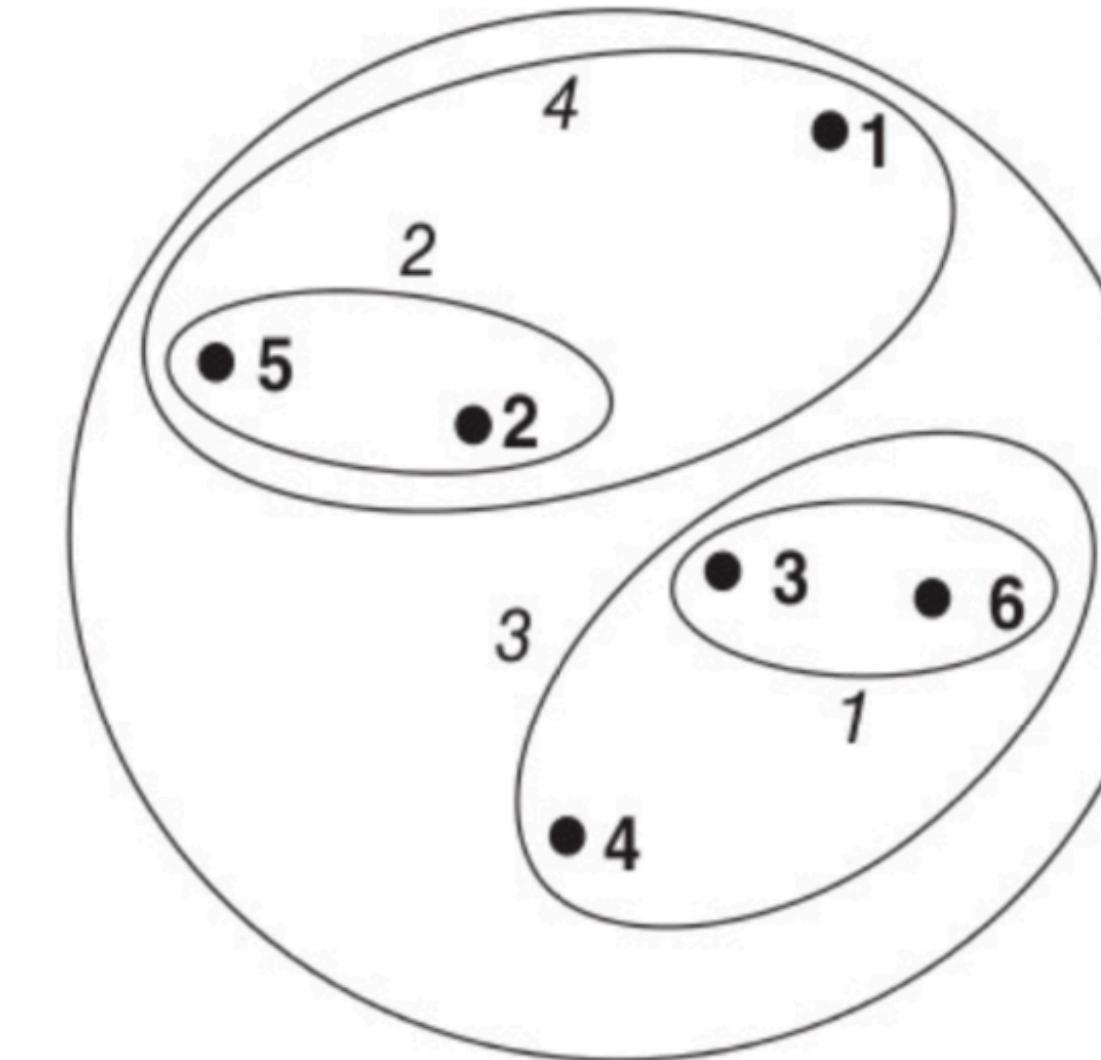
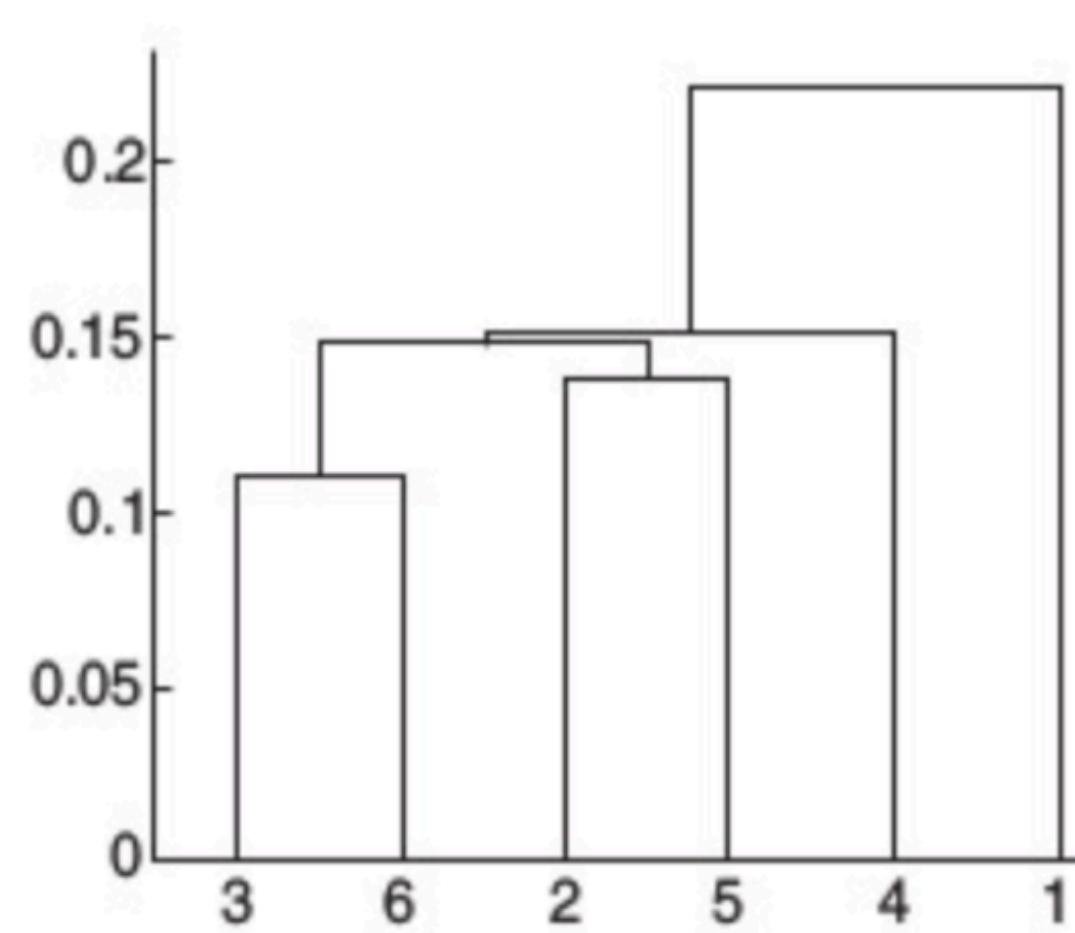
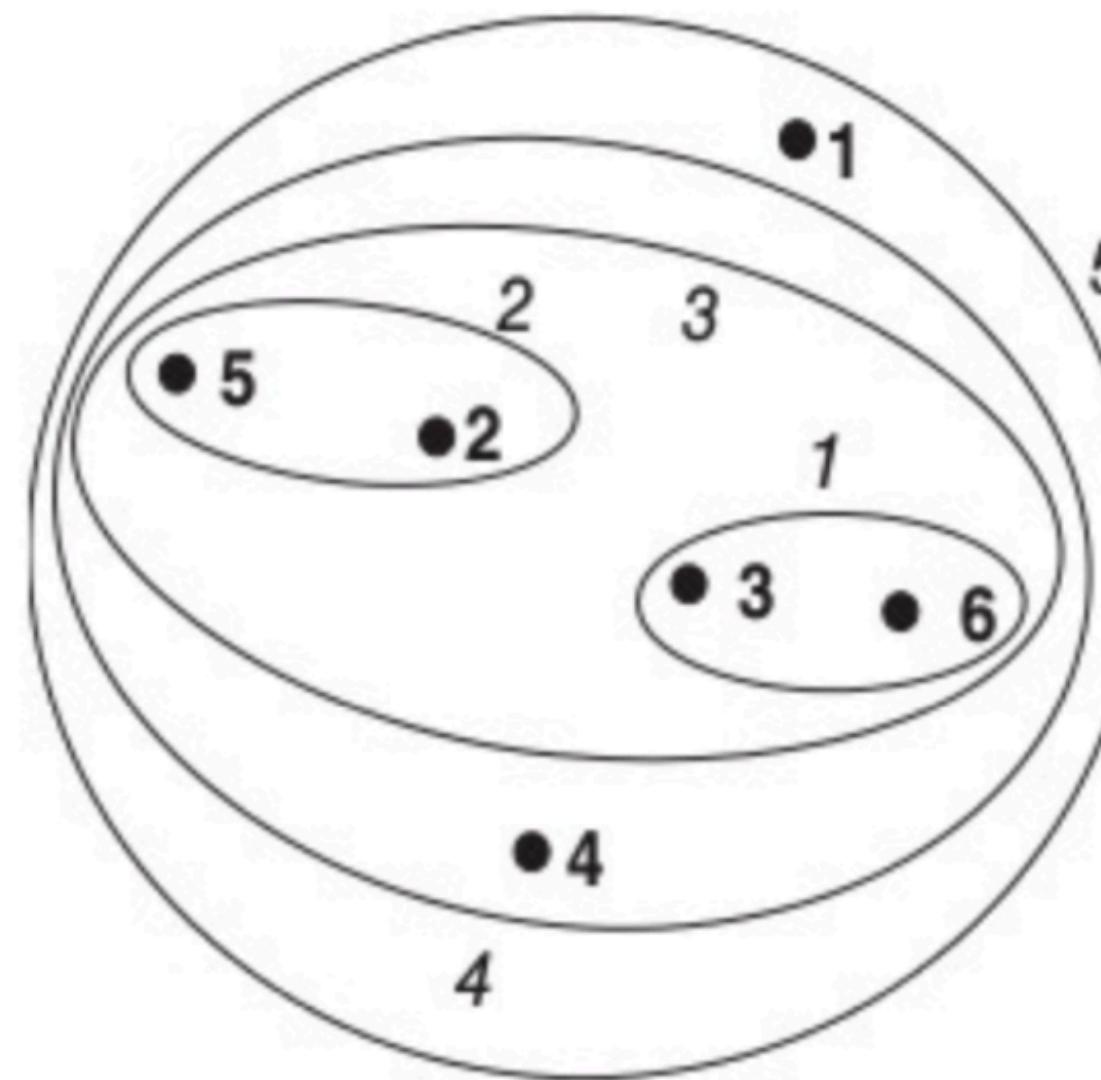
Hierarchical Clustering

Tipos de distancias:

- **Single link:** $D(c_1, c_2) = \min D(x_1, x_2)$ donde $x_1 \in c_1$; donde $x_2 \in c_2$
- La distancia es la distancia entre los puntos más cercanos de dos clusters.
- Produce largas cadenas
- **Complete link:** $D(c_1, c_2) = \max D(x_1, x_2)$ donde $x_1 \in c_1$; donde $x_2 \in c_2$
- La distancia es la distancia entre los puntos más lejanos de dos clusters.
- Fuerza a que existan clusters “esféricos” (diámetro constante).
- **Average link:** $D(c_1, c_2) = \frac{1}{|c_1||c_2|} \sum_{x_1 \in c_1} \sum_{x_2 \in c_2} D(x_1, x_2)$
- Promedia todas las distancias entre los puntos de dos clusters.
- Se ve menos afectado por la presencia de outliers.
- **Centroides:** $D(c_1, c_2) = D\left(\frac{1}{|c_1|} \sum_{x \in c_1} x, \frac{1}{|c_2|} \sum_{x \in c_2} x\right)$
- Toma en cuenta la distancia entre los centros de dos clusters.
- **Ward:** $TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x, \mu_{c_1 \cup c_2})^2$
- Considera unir dos clusters y se “pregunta”: Qué tanto cambia la distancia total (TD) de los centroides?



Hierarchical Clustering



Ventajas:

- No se necesita ninguna información previa sobre el número de clusters requerido.
- No se necesita inicialización o mínimo local.
- Fácil de implementar.

Desventajas:

- Falta de Función Objetivo Global. Se fija localmente en cada iteración que clusters deben ser unidos.
- Puede sufrir de ruido u outliers, rompiendo grandes clusters.
- Complejidad. Ej: $O(n^2)$, $O(n^3)$, $O(n^2 \log n)$

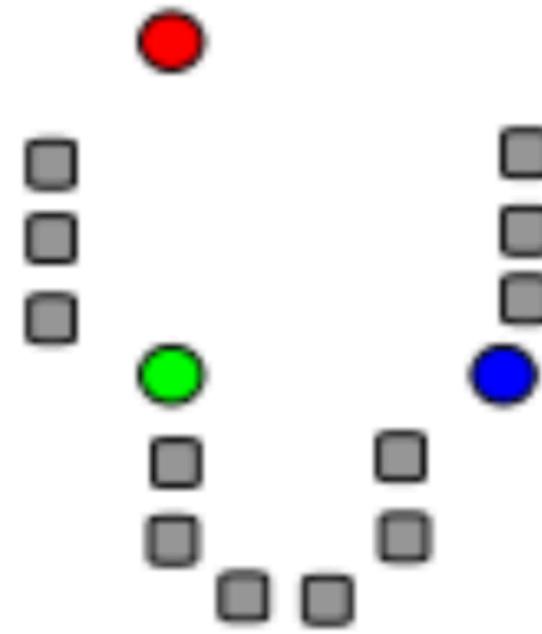
Prototype-based Clustering

El algoritmo más popular es K-means clustering, el cual partitiona N puntos en K clusters donde cada observación pertenece al cluster con la media más cercana. Se dice que K-means es un prototype-based clustering porque los centroides del cluster actúan como prototipos del cluster.

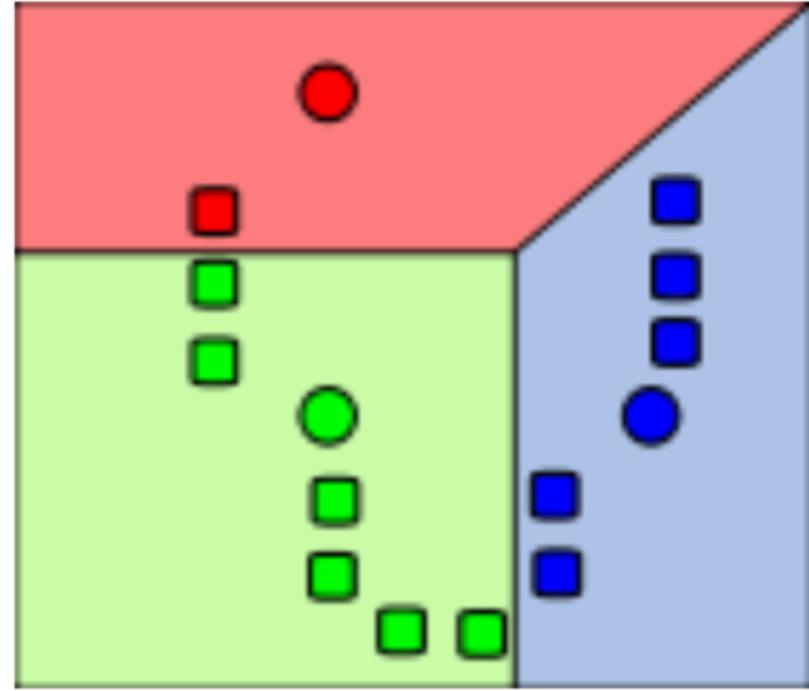
Algoritmo básico de K-means:

Data: $x \in \mathbb{R}^d$, número de clusters K.

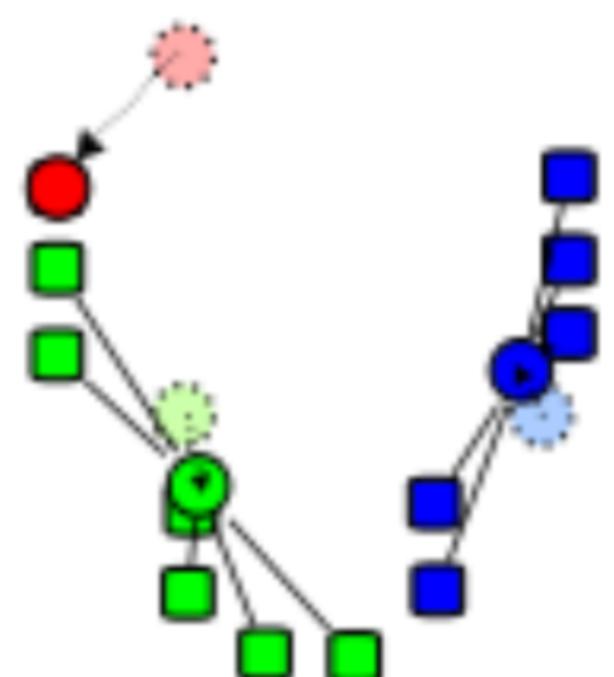
1. Inicializar K centros de manera aleatoria: $\mu_1, \mu_2, \dots, \mu_n$
2. **Repetir 3, y 4:**
3. **Para cada punto x_j :** asignar cada punto x_j al centro más cercano. Es decir, para cada punto x_j se calcula la distancia Euclídea a los K centros y se le asigna el centro con la menor distancia: $\arg \min D(x_j, \mu_k)$
4. **Para cada centroide μ_k :** recalculiar la posición de los centroides. Es decir, se toman todos los puntos que pertenecen a ese cluster y se promedian. El promedio de estos puntos es el nuevo centroide: $\mu_k = \frac{1}{n} \sum x_j$ para todos los puntos x_j pertenecientes al cluster K.
5. **Hasta** que los puntos no cambien de cluster (no garantiza encontrar el óptimo).



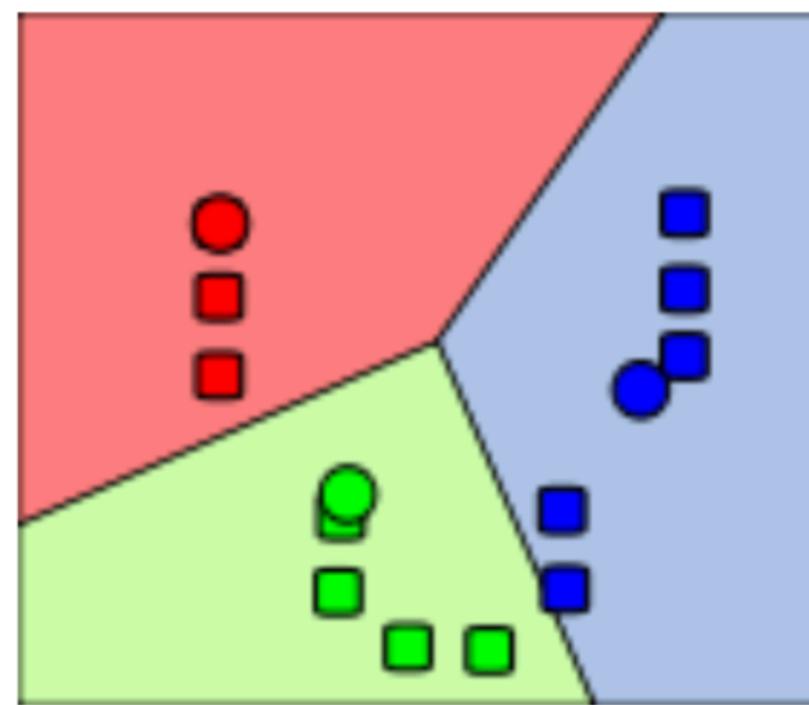
1) Inicializar centros



2) Asignar centro más cercano



3) Recalcular centros



4) Reasignar centros

- Si se toma la distancia Euclídea entonces K-means minimiza la suma de los errores cuadrados (Sum of Squared Errors SSE):

$$F(\mu, c) = \sum_{i=1}^k \sum_{x \in c_i} \|\mu_i - x\|^2$$

Puede ser demostrado por Gradient Descent.

- Encuentra mínimos locales respecto a SSE, ya que esta basado en decisiones específicas de la posición de los centroides y los clusters más que en todas las combinaciones posibles.
- Complejidad computacional $O(n^{dk+1} \log n)$

Existen problemas al elegir los centroides de manera aleatoria:

- Muchas iteraciones.
- Pueden tomarse puntos muestrales y agruparlos mediante hierarchical clustering

Clusters vacíos (ningún punto es asignado en la asignación en el paso de inicialización).

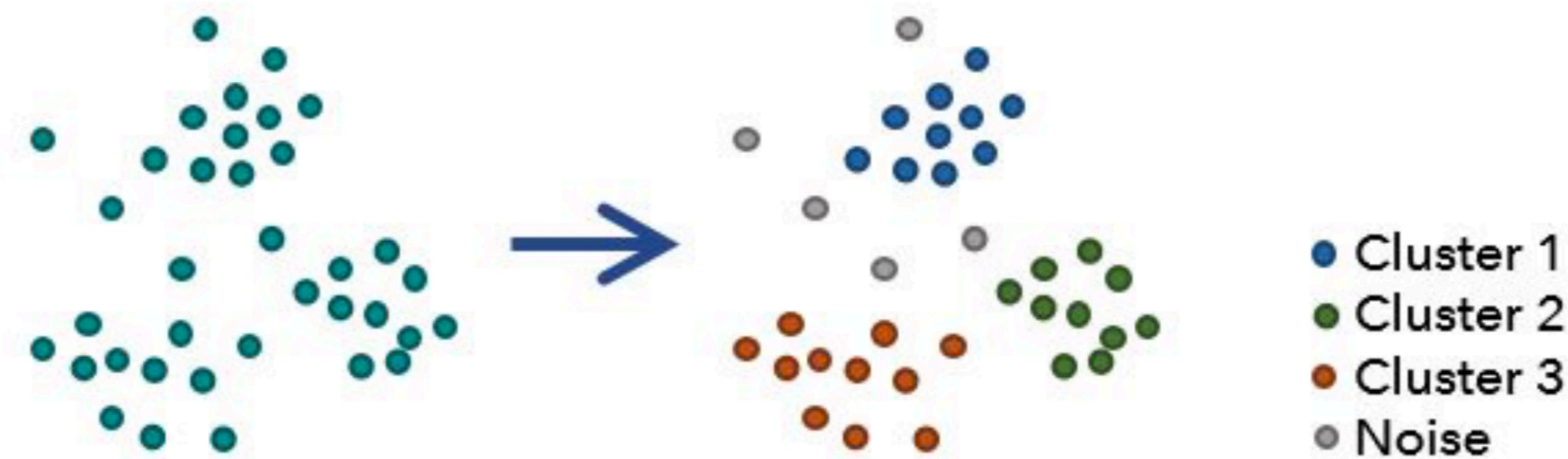
- Elegir el punto más lejano de algún centroide actual.
- Elegir el punto con mayor SSE.

Outliers

- Los clusters resultantes pueden no ser muy representativos.
- Pueden distorsionar la estructura del cluster.
- Pueden particionar clusters en regiones no representativas.

DBSCAN (Density Based Spatial Clustering Algorithm with Noise)

“The main concept of DBSCAN algorithm is to locate regions of high density that are separate from one other by regions of low density”.



Razones por las cuales usar DBSCAN:

1. No requiere mucho conocimiento del campo.
2. Puede descubrir clusters de formas arbitrarias.
3. Eficiente para grandes volúmenes de datos.
4. Pocos hiperparámetros que tunnear (radio épsilon y mínimo número de puntos).
5. Maneja eficientemente la presencia de outliers y ruido. Se ve poco afectado por éstos.

Cómo medimos la densidad y qué es una región densa?

- **Densidad en un punto P:** número de puntos en un círculo de radio ϵ desde un punto P.
- **Región densa:** para cada punto en un cluster, el círculo de radio ϵ contiene al menos el mínimo número de puntos (MinPts).

Definiciones:

- **Vecindad épsilon:** la vecindad épsilon de un punto P en un conjunto de datos D se define como:

$$N(p) = \{q \in D \mid dist(p, q) \leq \epsilon\}$$

La vecindad épsilon de un punto P es entonces el grupo de puntos que caen en un círculo de radio ϵ siendo P el centro de dicho círculo.

- **Core points:** puntos que satisfacen $|N(p)| \geq \text{MinPts}$.

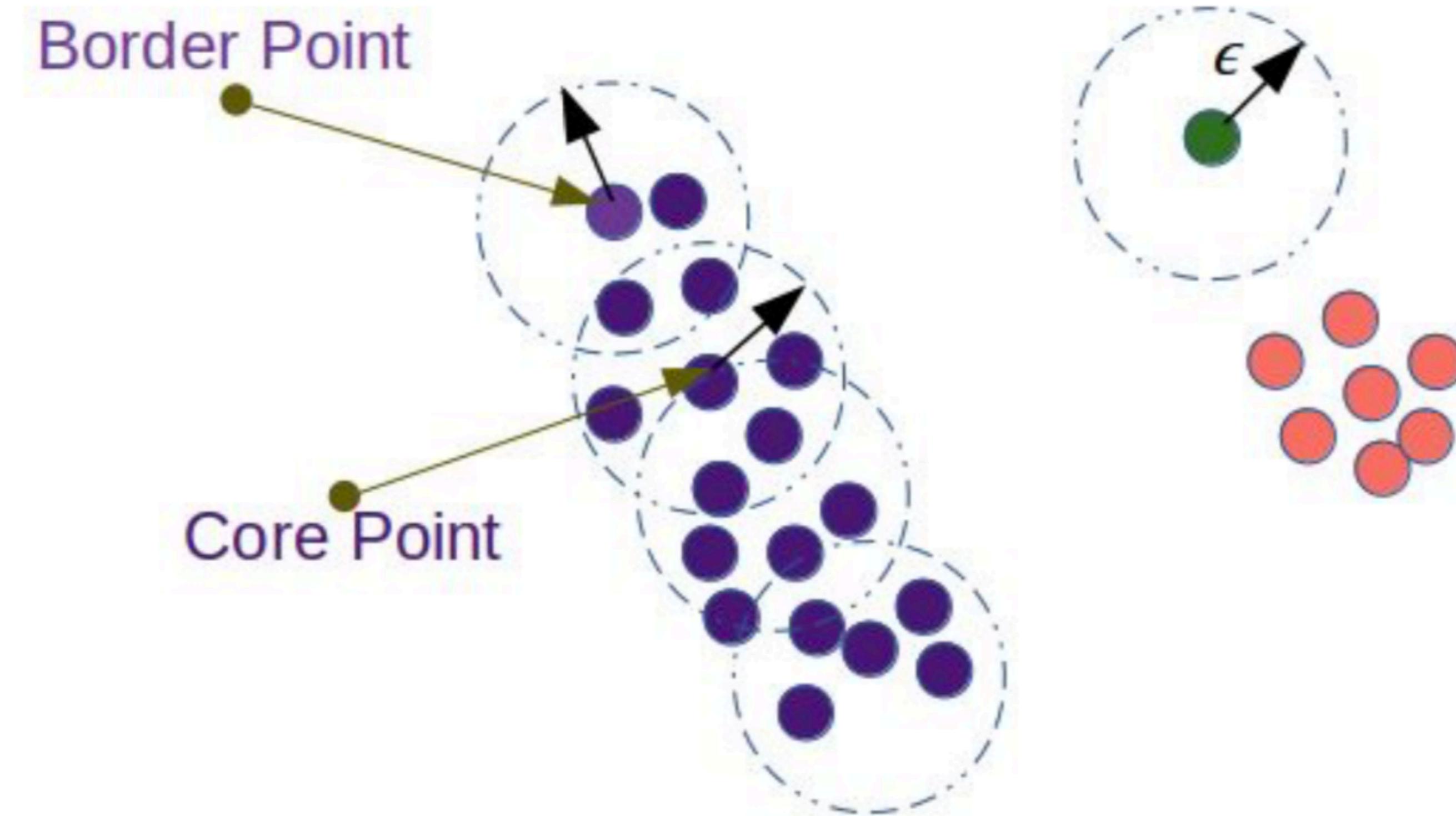
Los core points son puntos que si dibujamos un círculo de radio ϵ , con el core point como su centro, tendrán dentro de este círculo la cantidad mínima de puntos requeridos o más.

Definiciones:

- **Border point:** puntos que satisfacen $|N(p)| < \text{MinPts}$.

Los puntos bordes son puntos que si trazamos un círculo de radio épsilon, tomando el punto como centro, no contienen el mínimo número de puntos requeridos.

- **Ruido:** puntos que no son core ni border.



$$N_{Eps}(p) = \{q \in D \text{ such that } dist(p, q) \leq \epsilon\}$$

$\epsilon = 1$ unit, MinPts = 7

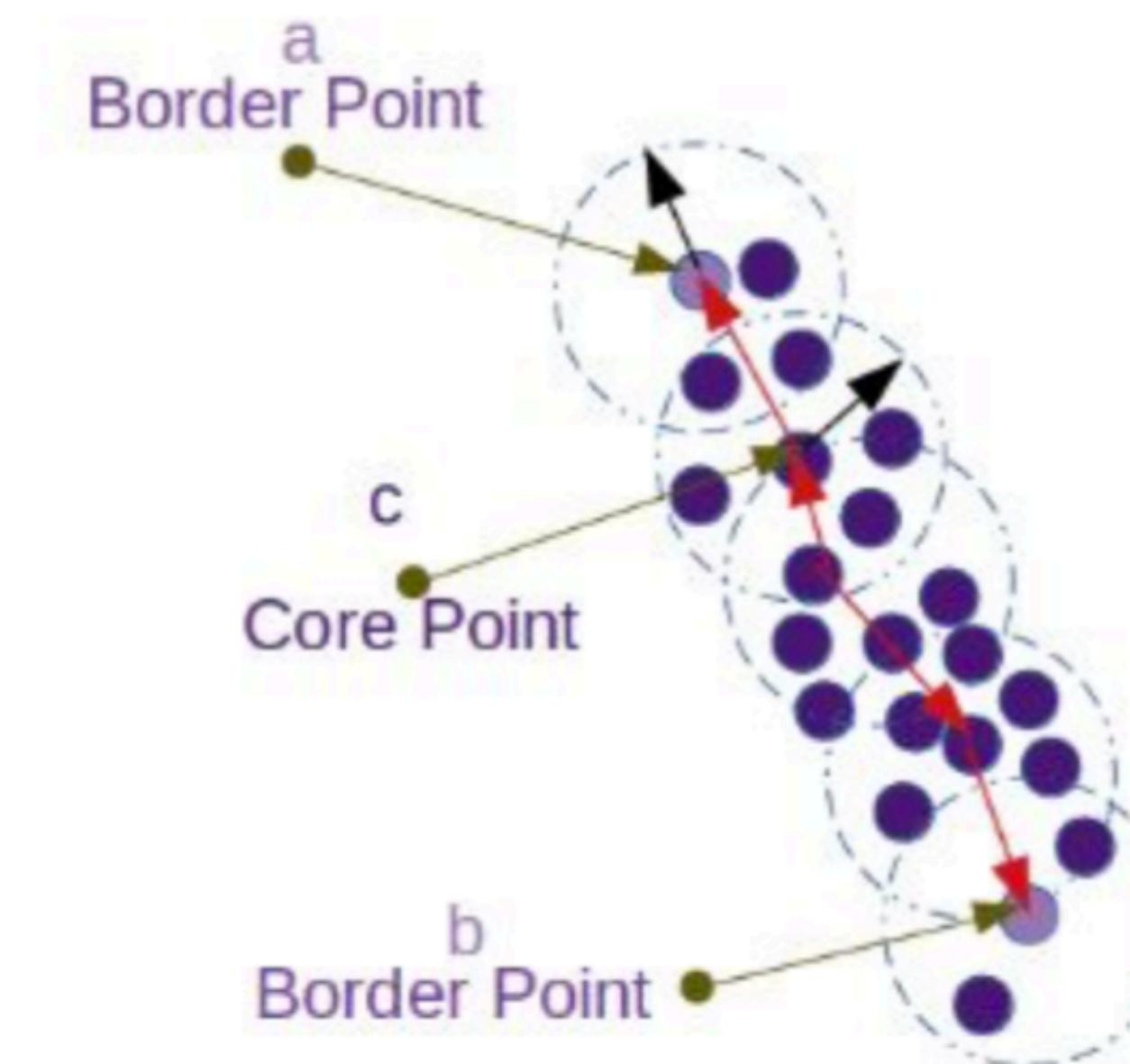
Directamente alcanzable por densidad: un punto A es directamente alcanzable por densidad (directly density reachable) por un punto B si:

1. **B es un core point** $|N(b)| \geq MinPts$
2. **A pertenece a la vecindad épsilon de B** $a \in N(b)$

Alcanzable por densidad: un punto A es alcanzable por densidad (density reachable) por un punto B con respecto a un ε y un número mínimo de puntos si:

“Para una cadena de puntos P_1, P_2, \dots, P_n donde $P_1 = A$ y $P_n = B$, los puntos P_{i+1} son directamente alcanzables por densidad de los puntos P_i .”

Densamente conectados: un punto A esta densamente conectado con un punto B, con respecto a ε y MinPts., si existe un punto C tal que A y B son alcanzables por densidad por C.



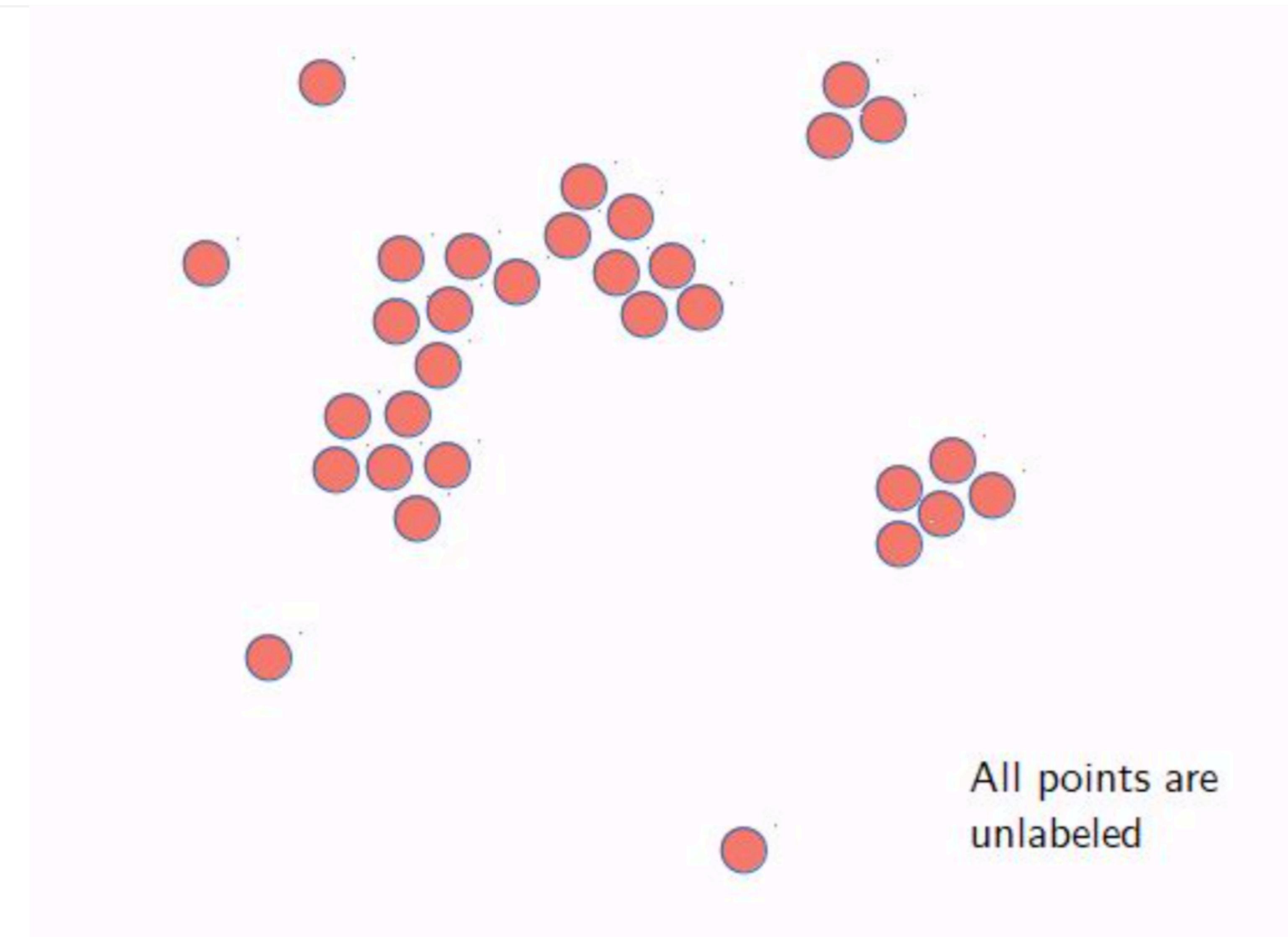
a, b are Density Reachable from a core point c.

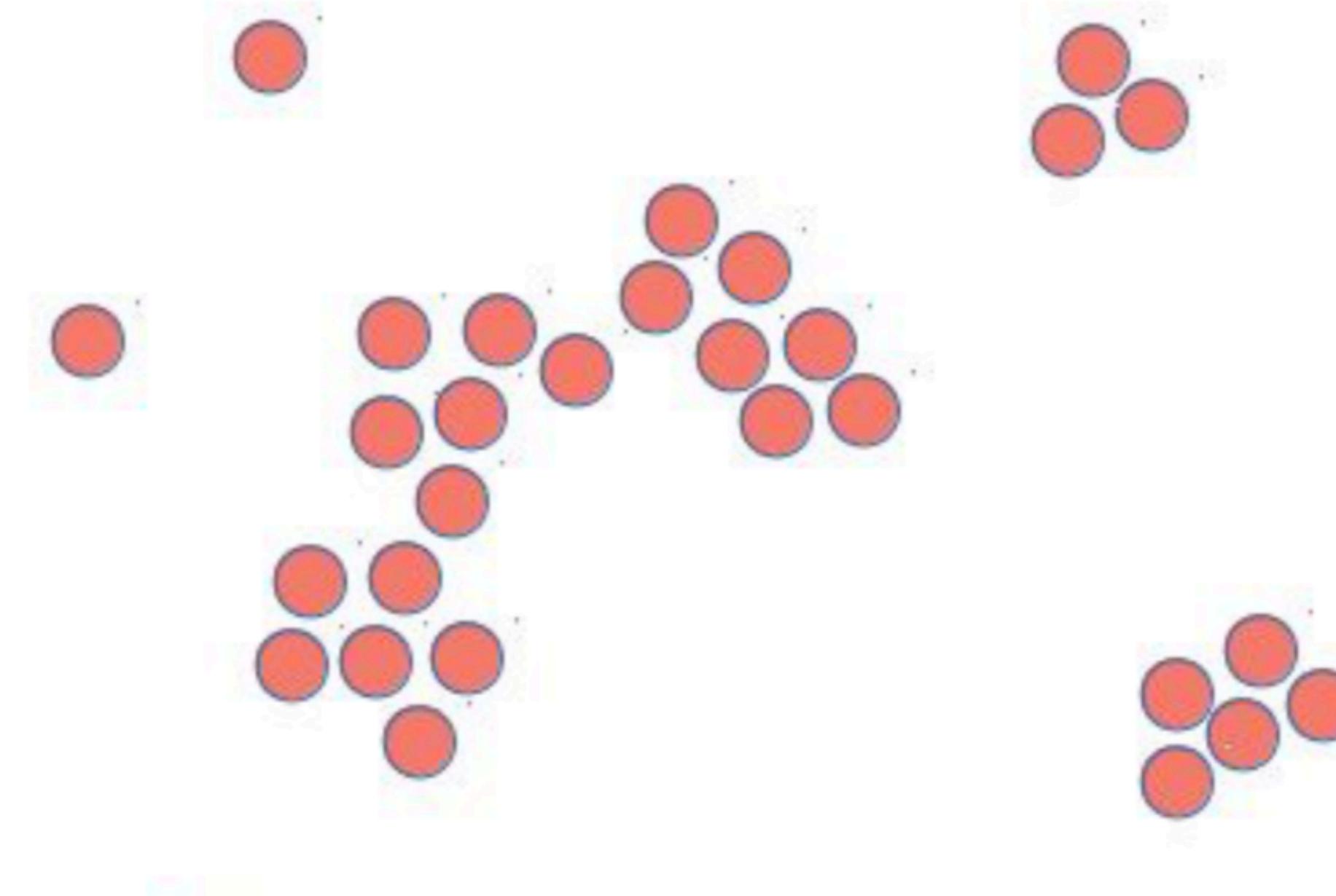
a, b are called Density Connected points.

Two border points a, b are density connected through the core point c.

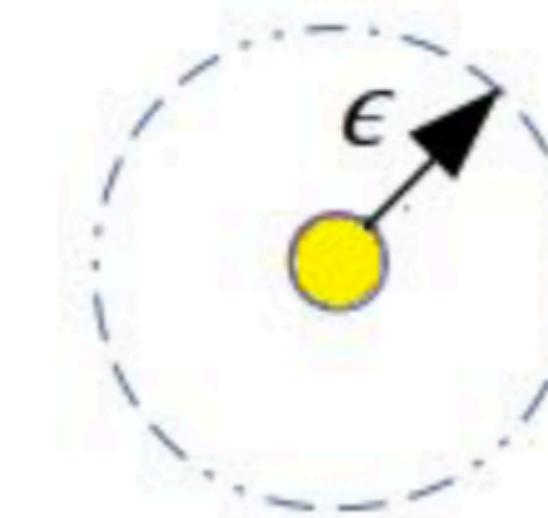
Pasos del algoritmo DBSCAN:

1. El algoritmo comienza con un punto arbitrario que no ha sido visitado y su información de vecindad es obtenida a partir de épsilon.
2. Si este punto contiene MinPts. en su vecindad épsilon se forma un cluster. De lo contrario, es catalogado como ruido. Este punto puede ser más tarde encontrado en la vecindad épsilon de otro punto y por ende pertenecer a otro cluster.
3. Si se determina que un punto es core entonces los puntos de su vecindad épsilon son parte de su cluster. Entonces, todos los puntos de encontrados en la vecindad épsilon son añadidos al igual que toda su vecindad si son puntos core.
4. El proceso anterior se repite hasta que todo el cluster densamente conectado es hallado.
5. El proceso se reinicia con un nuevo punto que puede ser parte de un nuevo cluster o ruido.

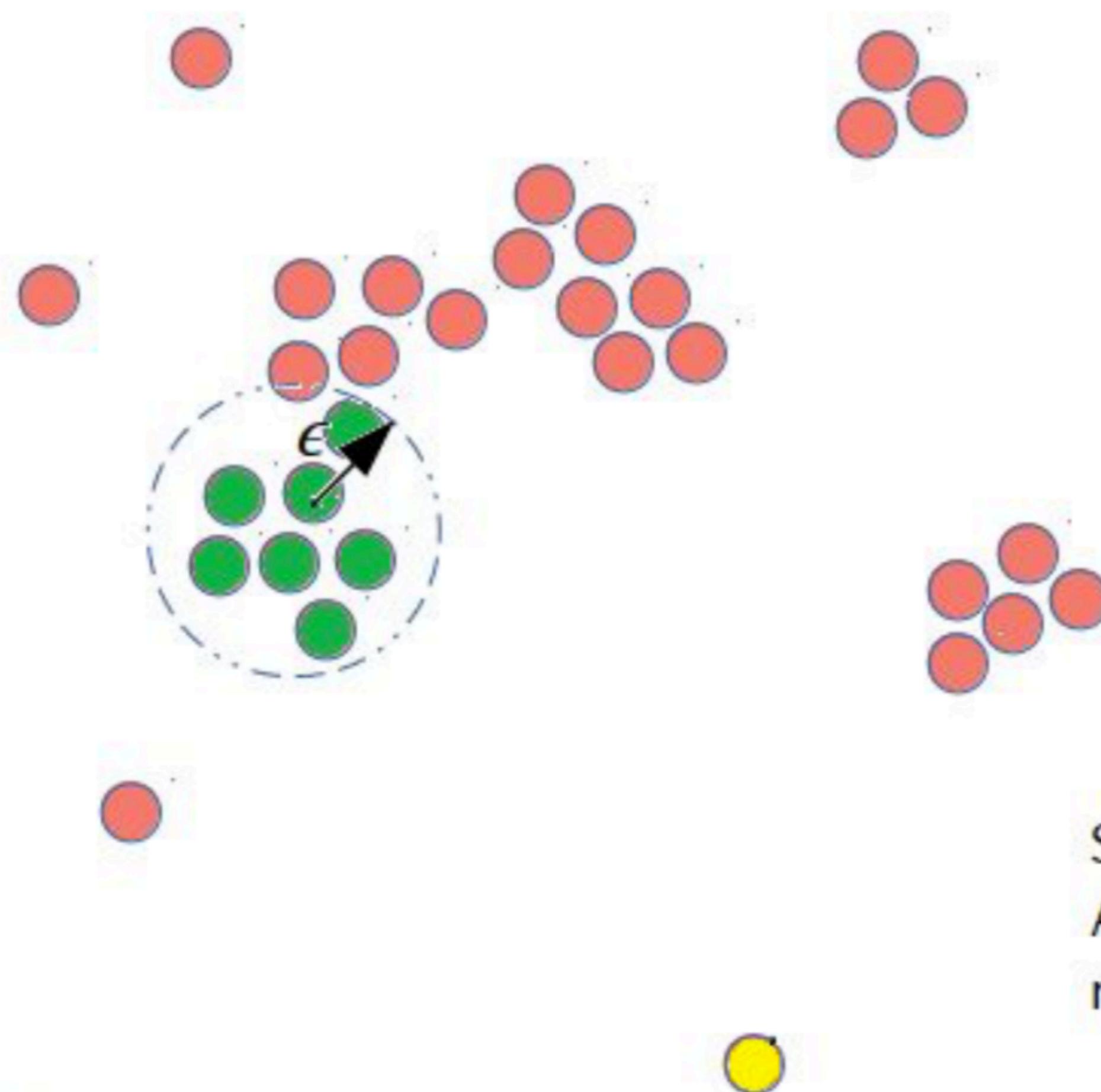




$\text{epsilon} = \epsilon$
 $\text{MinPts} = 5$

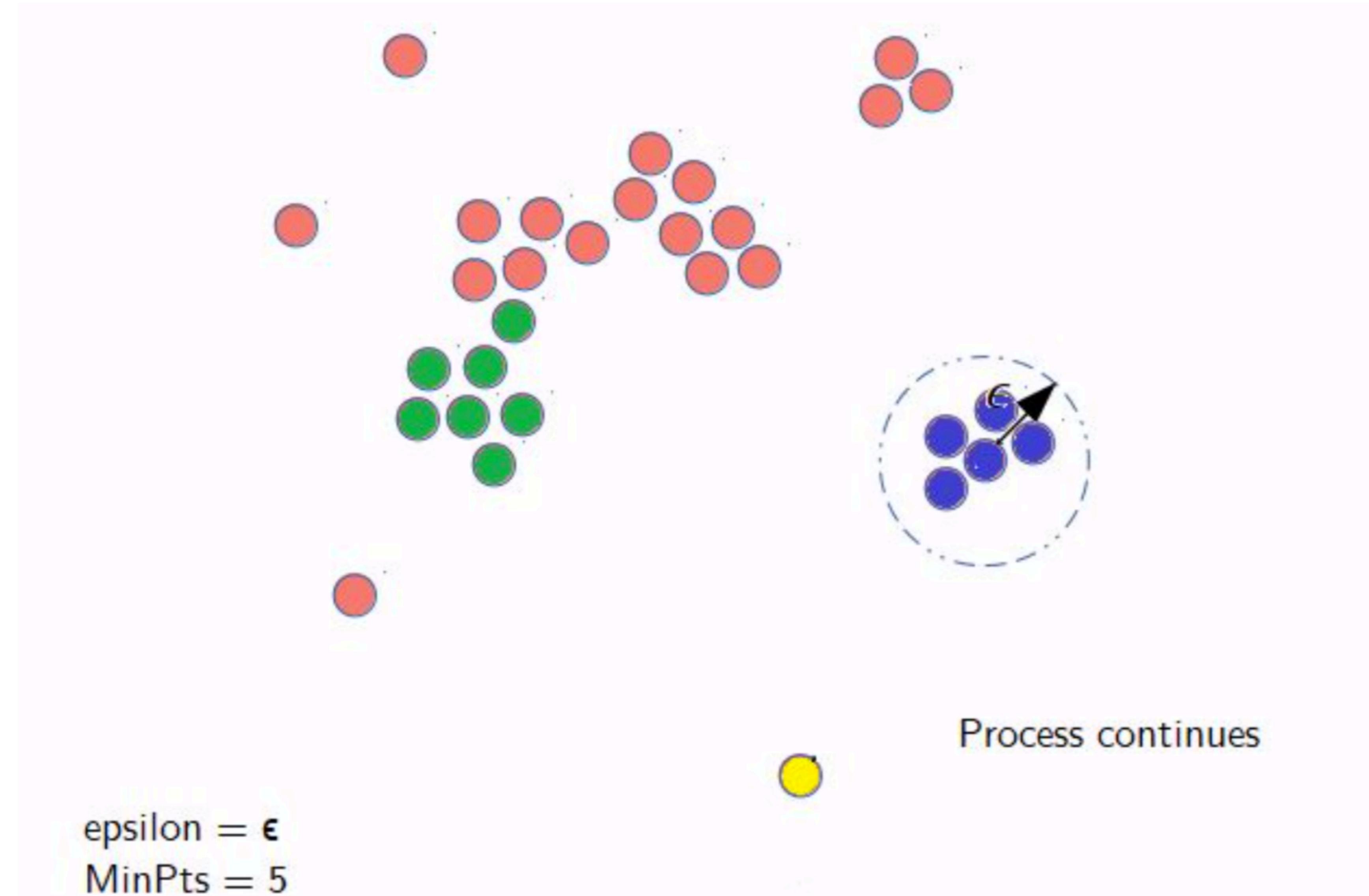


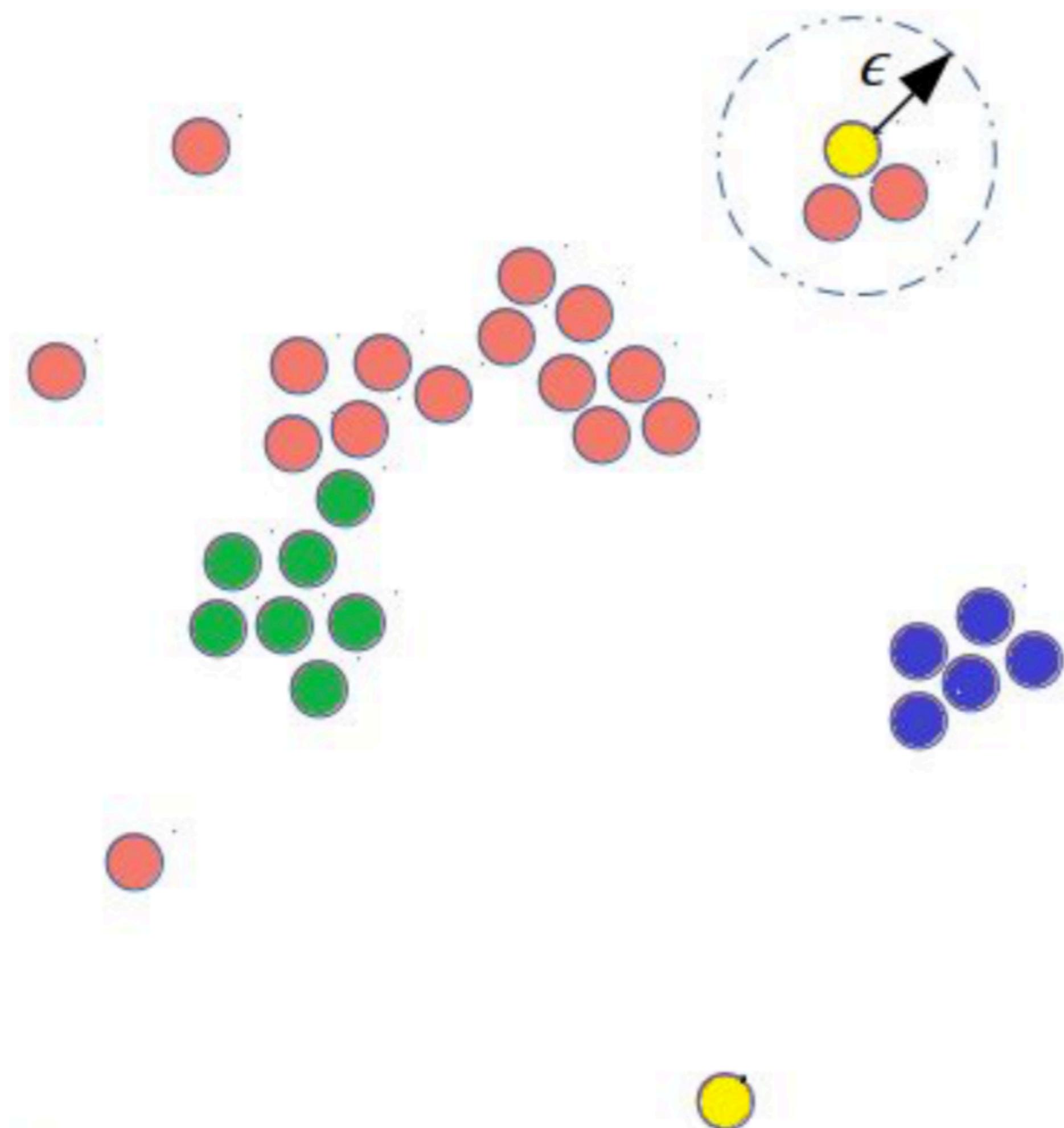
Algorithm starts with
arbitrary point.
Point is not core.
Point is labeled as noise



epsilon = ϵ
MinPts = 5

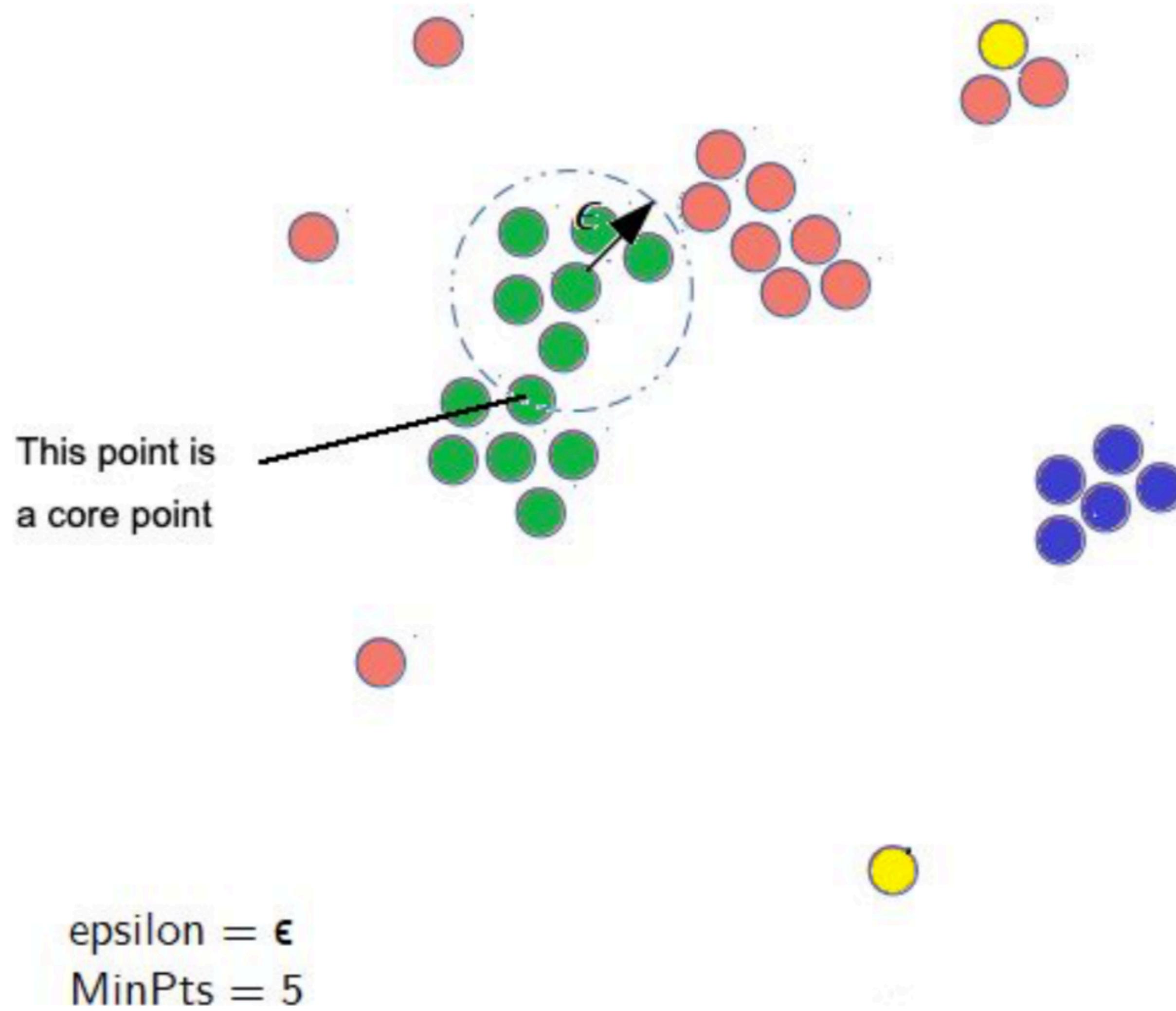
Second arbitrary point is core
All points in its
neighbourhood are labeled

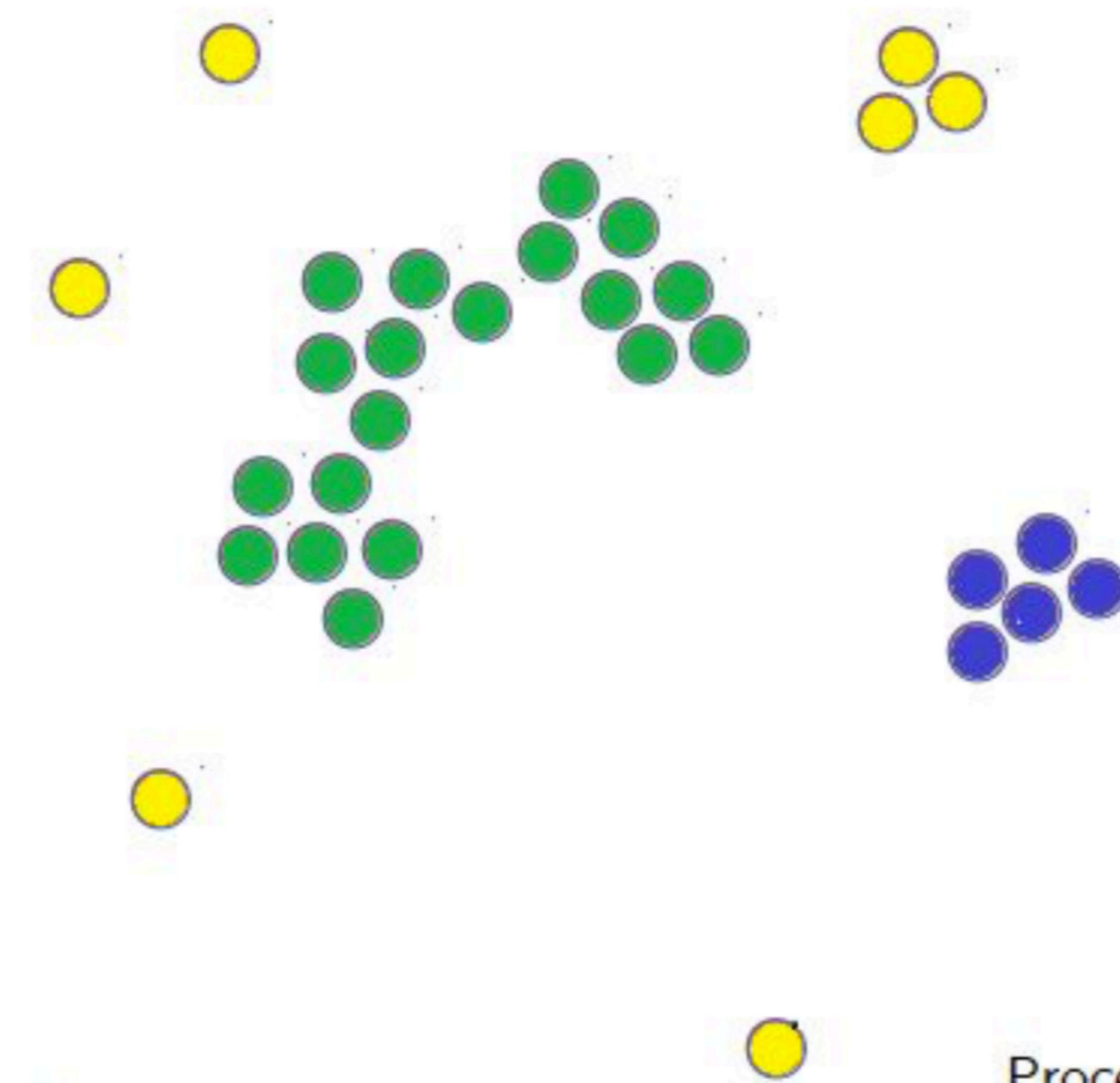




$\text{epsilon} = \epsilon$
 $\text{MinPts} = 5$

Process continues





epsilon = ϵ
MinPts = 5

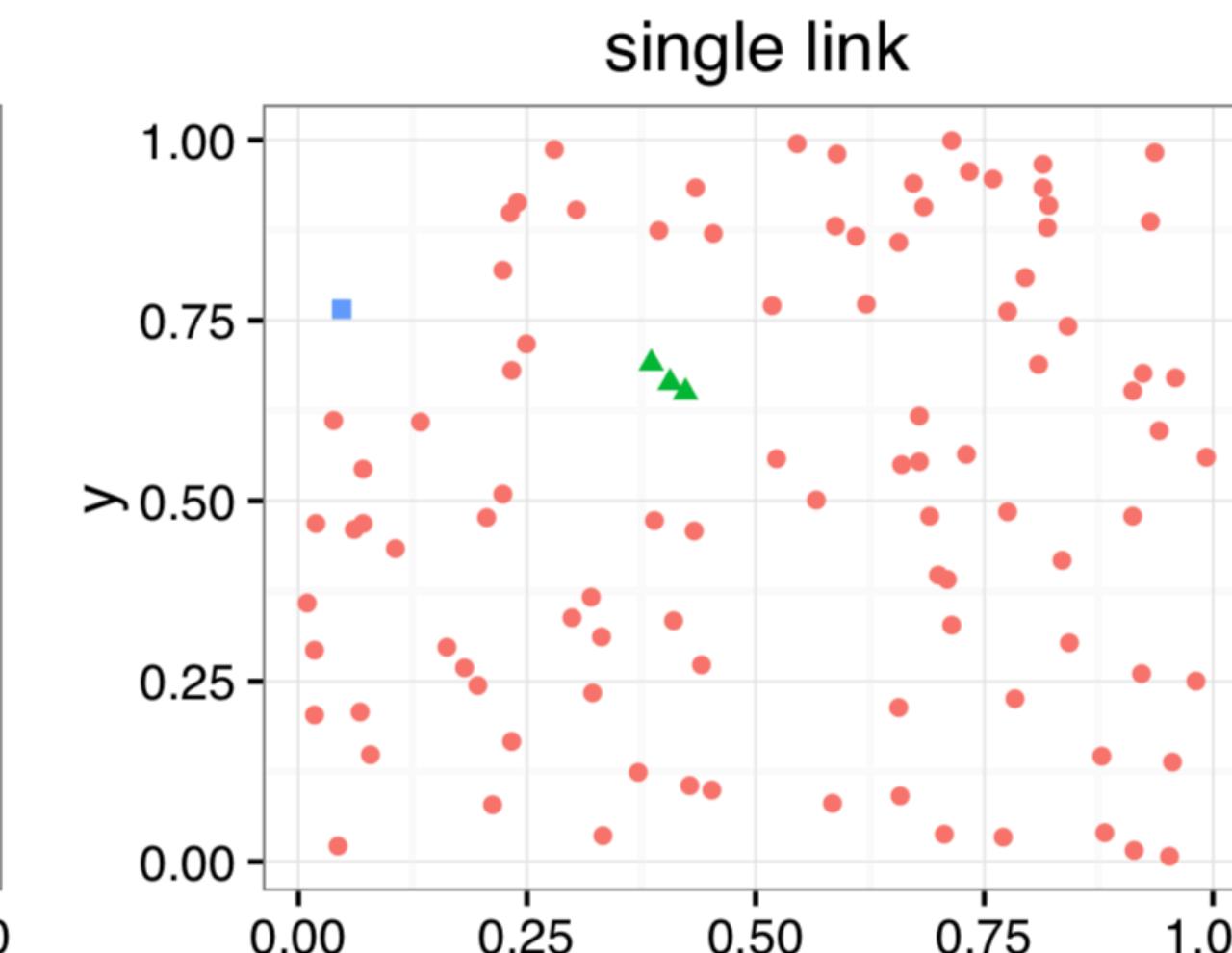
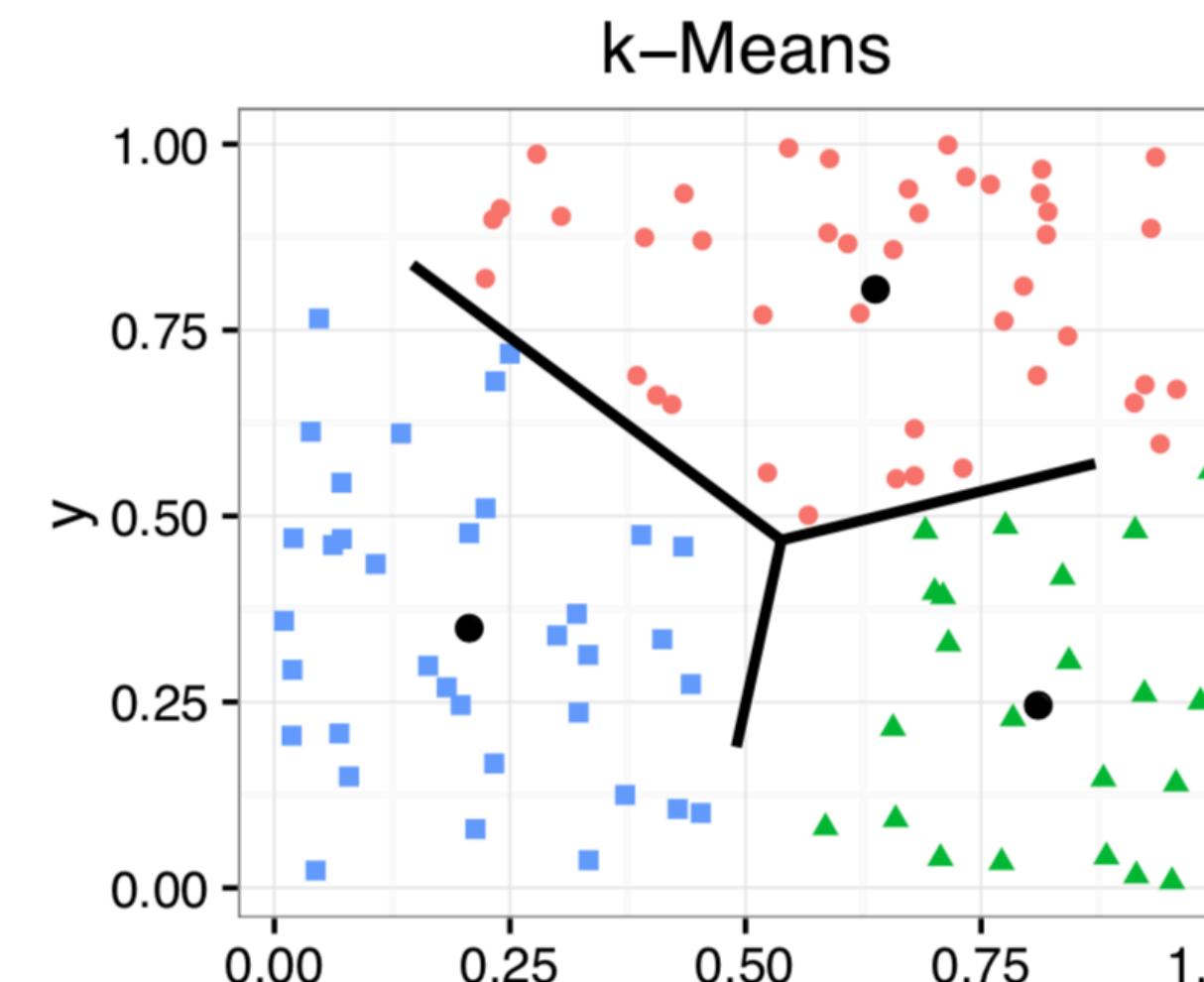
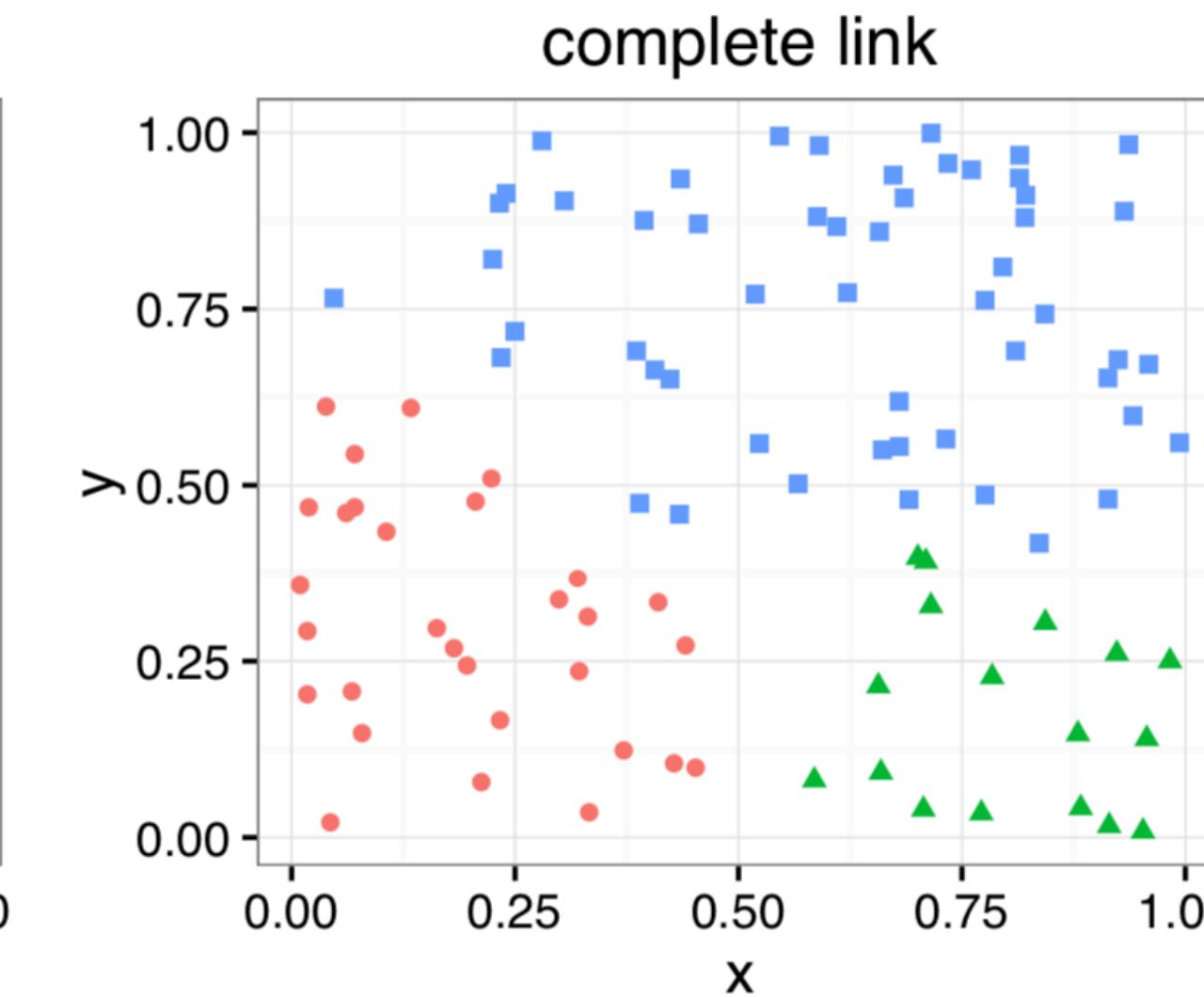
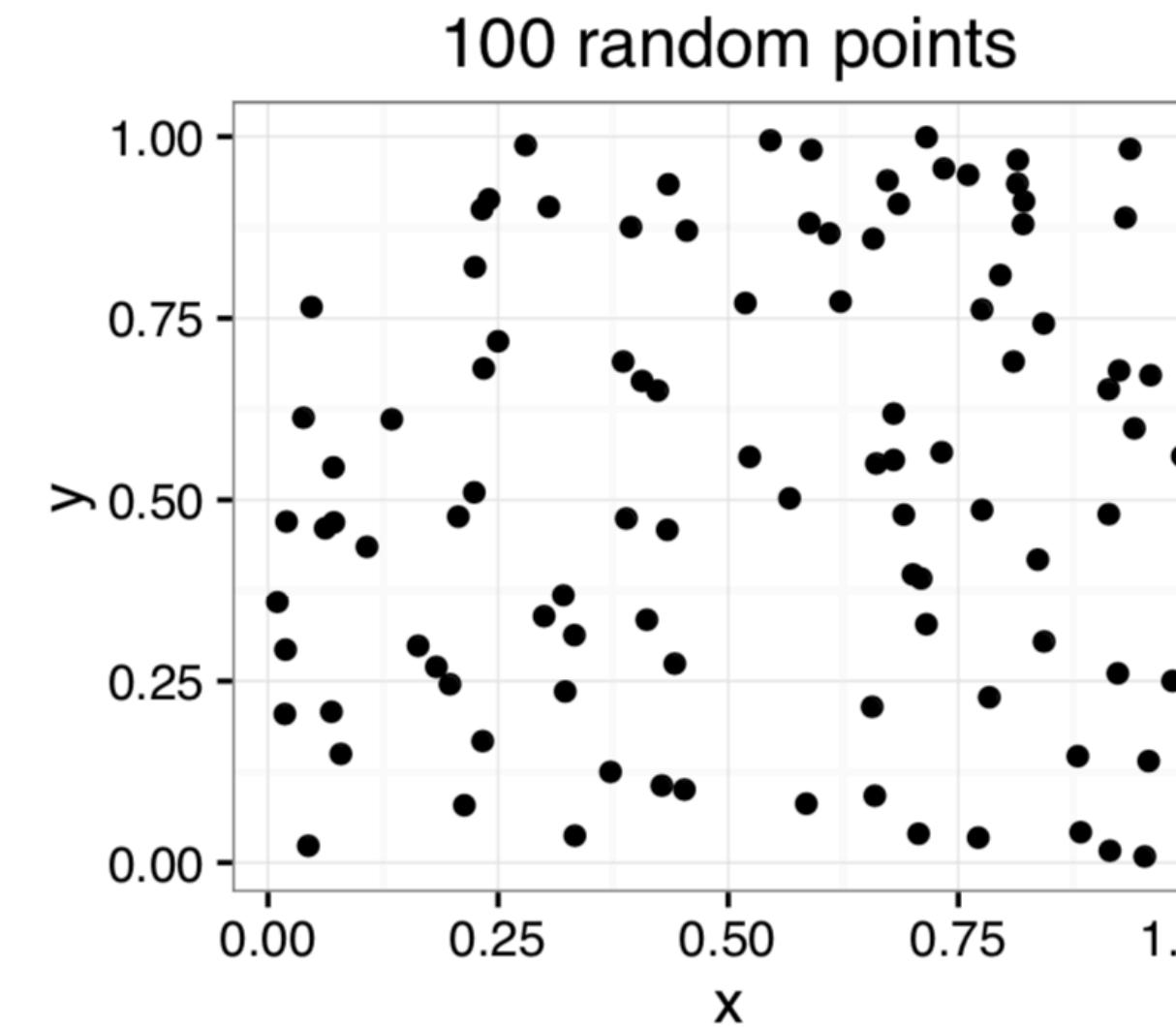
Process continues
until every point is
labeled

Desventajas:

- Si el conjunto de datos presenta clusters de densidad variable entonces DBSCAN falla en agrupar correctamente los datos ya que el algoritmo depende de épsilon y MinPts. No pueden ser elegidos de manera separada para cada cluster.
- Si los datos y los atributos no son bien entendidos por un experto en el campo, entonces establecer épsilon y MinPts. puede ser difícil. Puede requerir múltiples iteraciones con distintos valores de estos hiperparámetros.

Evaluación de clustering

Por qué es necesario evaluar nuestro clustering?



Evaluación de clustering

Por qué es necesario evaluar nuestro clustering?

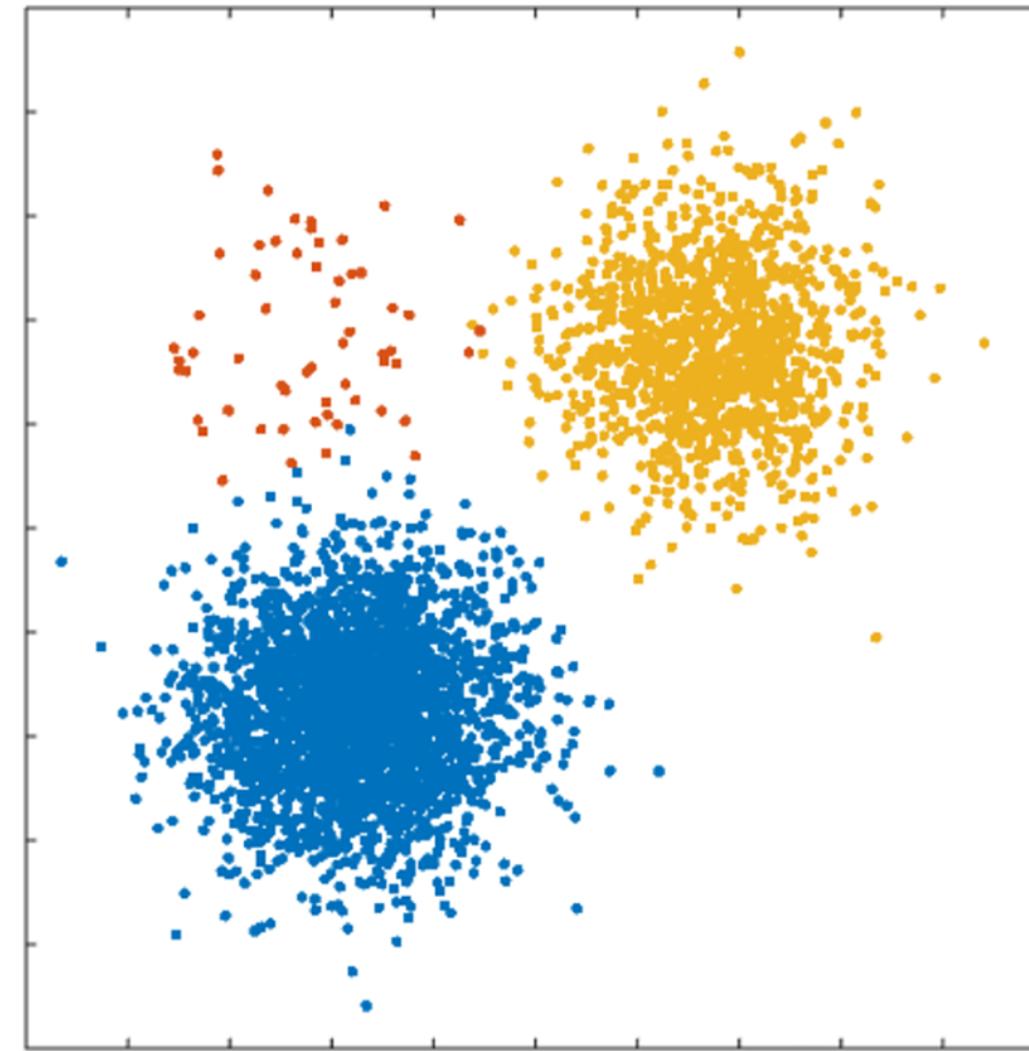
Cuestiones importantes para la evaluación de clustering:

1. Determinar la tendencia de los clusters de nuestros datos.
2. Determinar el número correcto de clusters.
3. Evaluar que tan bien los resultados de un análisis de clusters se ajustan a nuestros datos sin referencia a información externa.
4. Comparar los resultados de nuestro análisis de clusters con información externa conocida.
5. Comparar dos conjuntos de agrupamientos para ver cuál es mejor.

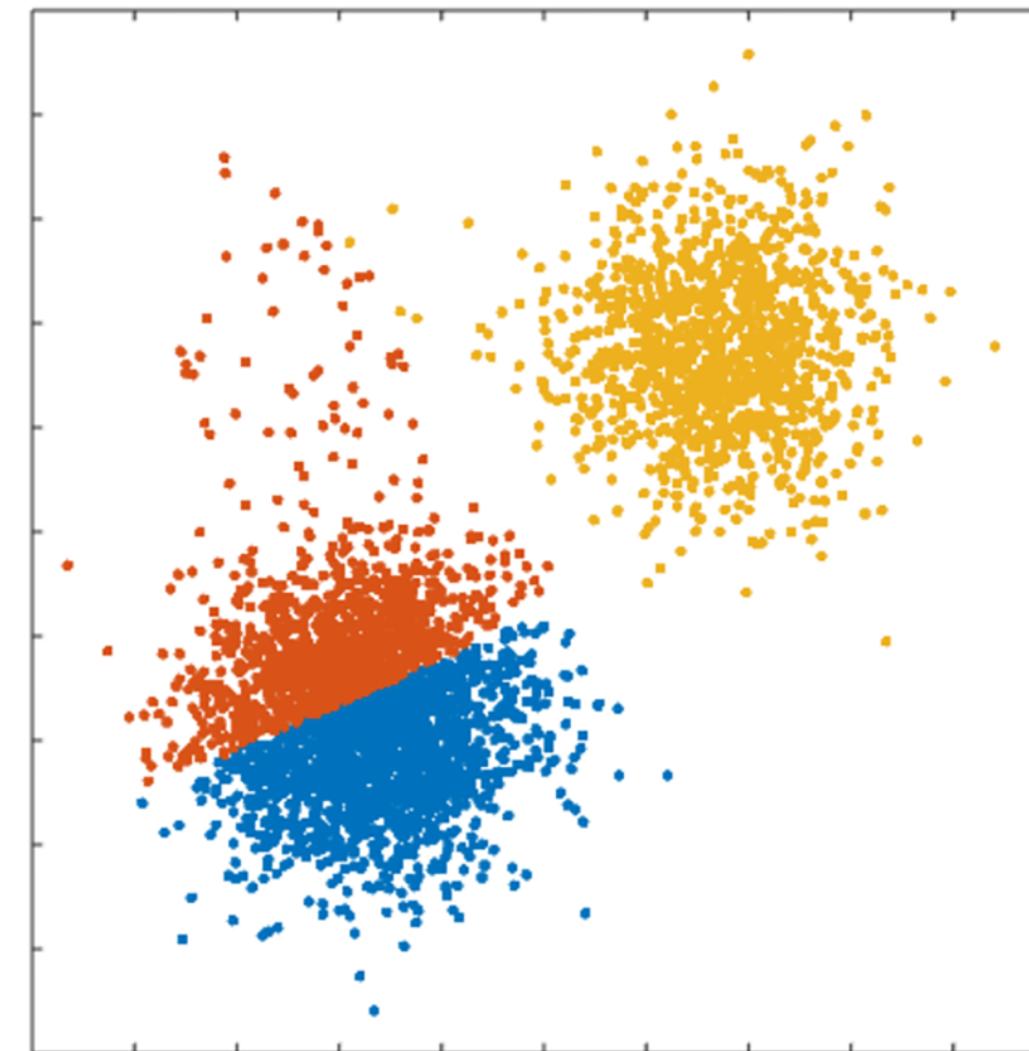
Evaluación de clustering

Por qué es necesario evaluar nuestro clustering?

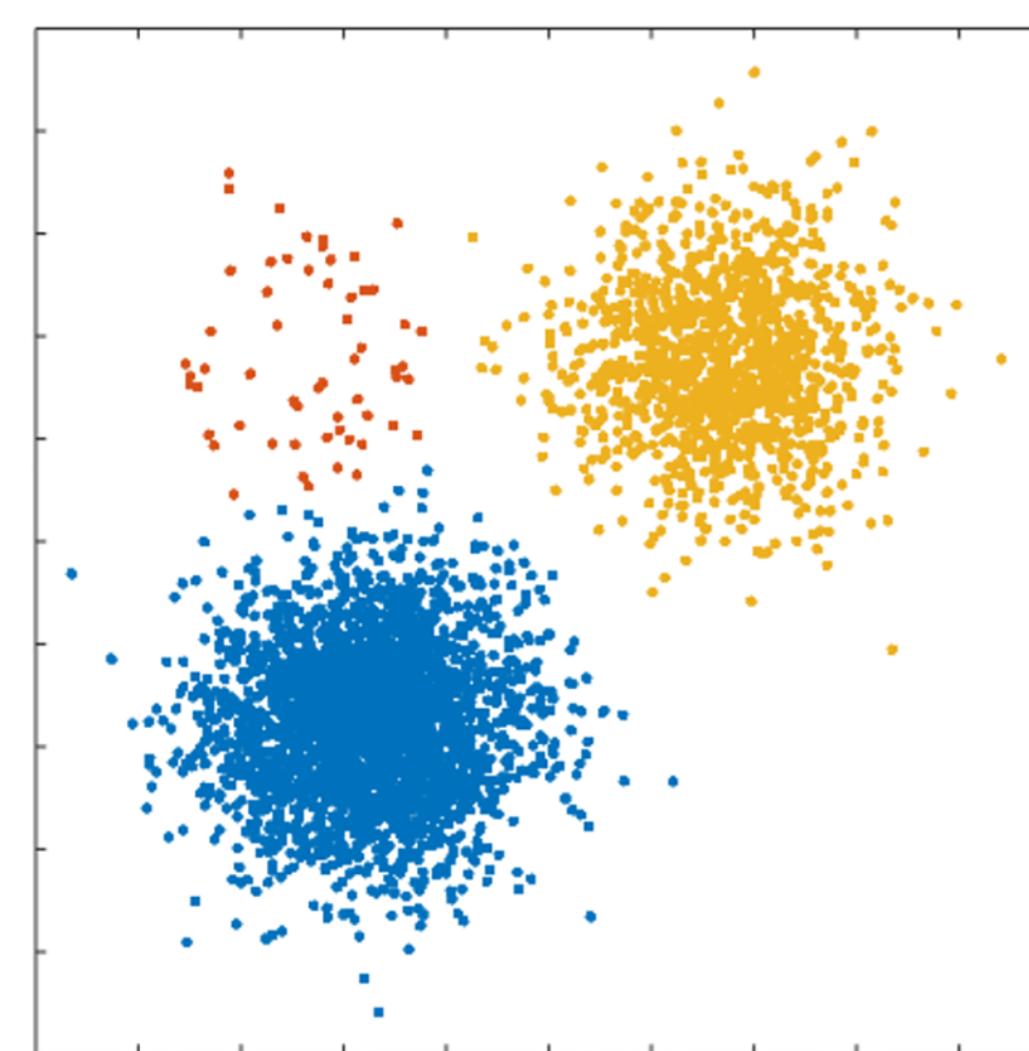
- Nuestros datos presentan clusters si al menos existen algunos clusters de buena calidad.
- Correr múltiples métodos.
- Evaluar la calidad de los clusters encontrados.
- Si nuestros clusters son uniformemente pobres, entonces nos puede indicar de que no hay presencia de clusters en nuestros datos.



(a) Generated synthetic data



(b) K -means



(c) MAP-DP

Evaluación de clustering

Testeo sin clustering:

- **Espacio euclídeo:** empleo de test estadísticos para aleatoriedad espacial. Elegir un modelo, estimar parámetros y evaluar la significancia estadística de la hipótesis de que los datos no fueron generados de manera no aleatoria. Muy desafiante.
- **Estadístico de Hopkins:**
 1. Muestreo uniforme de m puntos p_1, p_2, \dots, p_m de un conjunto de datos D.
 2. $\forall p_i \in D$ encontrar el vecino más cercano p_j con distancia $d_i = \text{dist}(p_i, p_j)$.
 3. Generar datos simulados q_1, q_2, \dots, q_m de una distribución aleatoria uniforme con la misma varianza que el dataset original D.
 4. $\forall q_i \in D$ encontrar el vecino más cercano q_j con distancia $\hat{d}_i = \text{dist}(q_i, q_j)$
 5. Calcular el estadístico de Hopkins:

$$H = \frac{\sum_{i=1}^m \hat{d}_i}{\sum_{i=1}^m d_i + \sum_{i=1}^m \hat{d}_i}$$

Evaluación de clustering

Como interpretar el estadístico de Hopkins?

Si D fuese uniformemente distribuido entonces: $\sum_{i=1}^m d_i \cong \sum_{i=1}^m \hat{d}_i$

Por lo tanto, $H \approx 0.5$

Sin embargo, si hay clusters presentes en nuestros datos la distancia artificial sería substancialmente más grande que nuestra distancia real y el valor de H incrementaría.

- $H \approx 0.5$ si puntos aleatorios y nuestros datos son aproximadamente parecidos.
- $H \approx [0.7 – 1]$ los datos presentan clusters evidentes.

$$H = \frac{\sum_{i=1}^m \hat{d}_i}{\sum_{i=1}^m d_i + \sum_{i=1}^m \hat{d}_i}$$

Evaluación de clustering

Evaluación interna:

- **Coeficiente de Silhouette:** mide qué tan similar es un objeto a su propio cluster (cohesión) comparado con otros clusters (separación).
- Para cada muestra x_i definimos:
 1. $a(i)$ = disimilitud promedio a otros puntos del mismo cluster.
 2. $d(i, c)$ = disimilitud promedio con todos los objetos en otro cluster c .
 3. $b(i) = \min(d(i, c))$ $c \neq A$

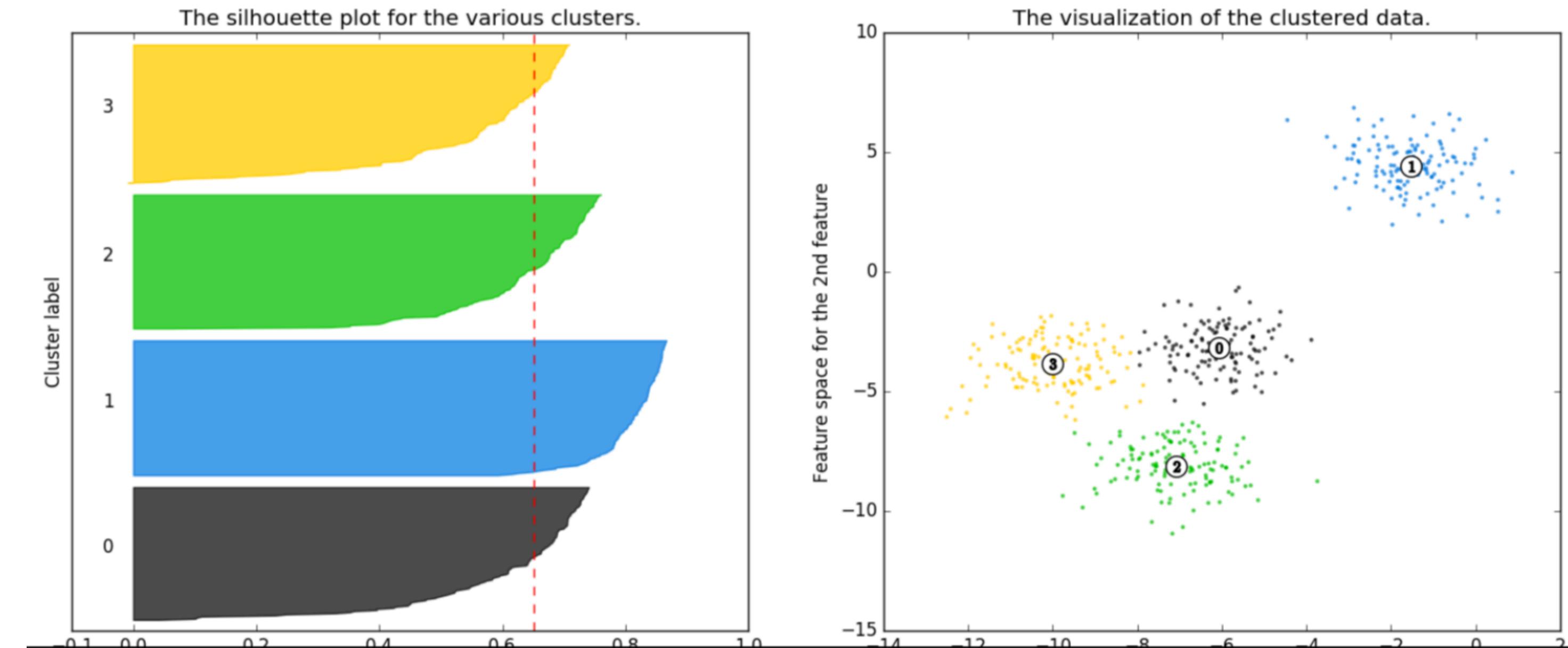
$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad -1 \leq s(i) \leq 1$$

Evaluación de clustering

- El coeficiente de Silhouette varía entre -1 y 1. Un valor cercano a 1 indica que el objeto tiene una buena correspondencia con su propio cluster y poca correspondencia con los clusters vecinos.
- Si la mayoría de los objetos posee un valor alto entonces la configuración de clusters es buena. Por lo contrario, si la mayoría de los objetos poseen un valor bajo o negativo la configuración de clusters posee muchos o muy pocos clusters.



Evaluación de clustering

Existen otras medidas de evaluación de clusters:

1. Davies-Bouldin.
2. Calinski-Harabasz.
3. Índice de Dunn.
4. Índice R cuadrado.
5. Hubert-Levin (Índice C).
6. Krzanowski-Lai.
7. Hartigan.
8. Root mean squared standard deviation index (RMSSTD).
9. Semi-partial R-squared.
10. Distancia entre dos clusters (CD).
11. Weighted inter-intra index.
12. Índice de homogeneidad
13. Índice de separación.

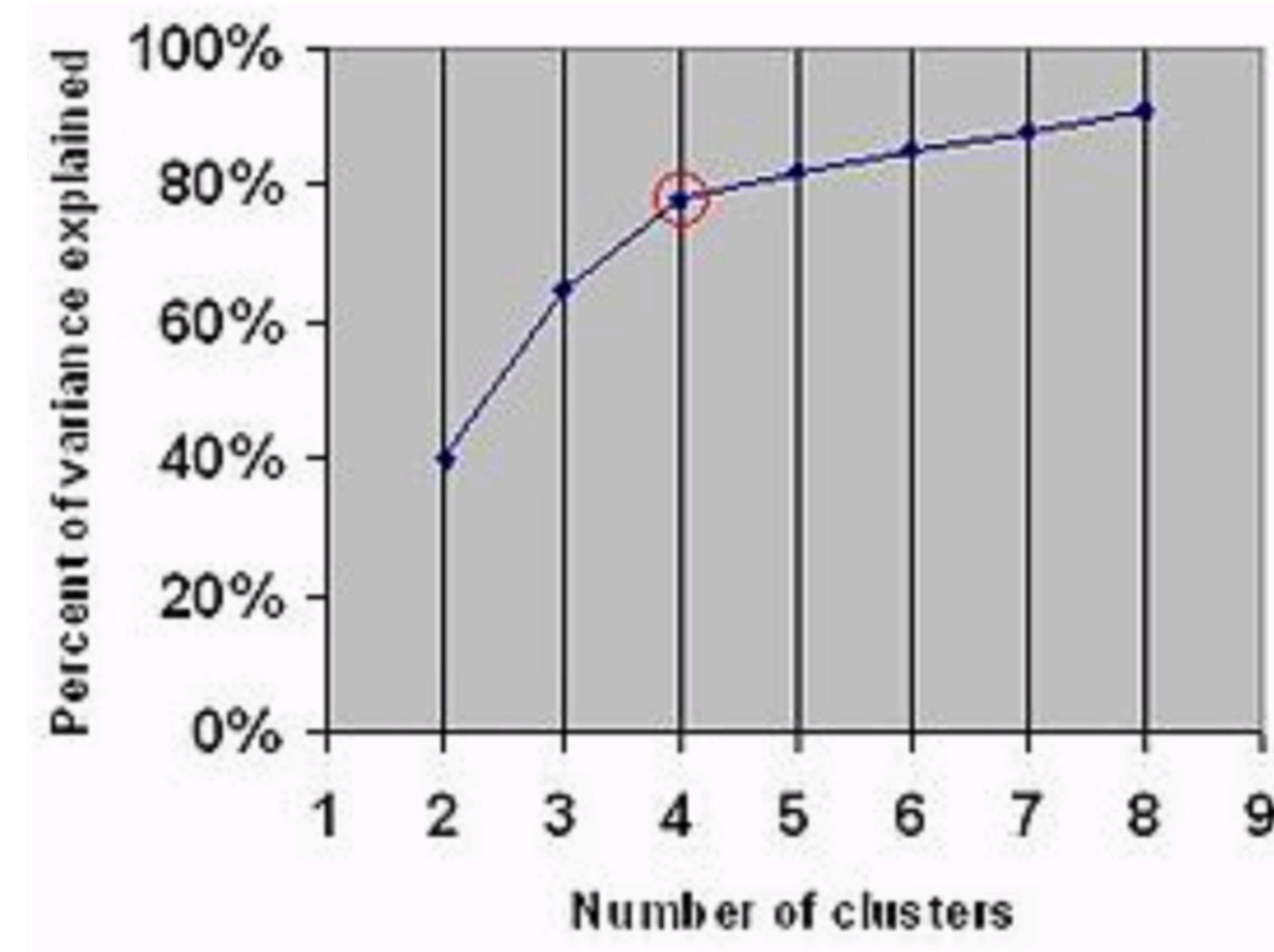
Determinar el número correcto de clusters

- Para ciertos algoritmos de clustering el parámetro K indica el número de clusters presentes a detectar.
- Hierarchical clustering evita este problema.
- El número correcto de clusters es frecuentemente ambigüo dependiendo de la interpretabilidad de la forma de la distribución y el nivel de resolución que se quiere alcanzar.

Estrategias:

- Buscar el número de clusters donde se produce un “codo” o “pico” en la gráfica “Medida de evaluación vs. Número de clusters”.
- Cross-validation: agrupar en $v-1$ training sets y calcular una función objetivo en el restante v test set y promediar para cada número de clusters. Elegir el que minimiza el error del test set.
- Elbow method: porcentaje de la varianza explicada (F-test) en función del número de clusters.

Determinar el número correcto de clusters



Problemas asociados a clustering

1. Lidiar con volúmenes grandes de datos y un gran número de dimensiones puede ser problemático por la complejidad del tiempo de cómputo.
2. La efectividad del método depende de la definición de distancia y la métrica a evaluar.
3. El resultado de un análisis de clustering puede ser interpretado de distintas maneras pudiendo llegar a ser tanto ambigüo como arbitrario.