

# Clase 8: Aprendizaje No Supervisado - Clustering

Matías Leoni

**Aprendizaje Automático I**

*Maestría en IA - Universidad de San Andrés*

5 de Agosto de 2025

# Agenda de la Clase

- Introducción al Aprendizaje No Supervisado

# Agenda de la Clase

- Introducción al Aprendizaje No Supervisado
- K-Means: Agrupamiento por Centroides

# Agenda de la Clase

- Introducción al Aprendizaje No Supervisado
- K-Means: Agrupamiento por Centroides
- Clustering Jerárquico: Aglomerativo y Divisivo

# Agenda de la Clase

- Introducción al Aprendizaje No Supervisado
- K-Means: Agrupamiento por Centroides
- Clustering Jerárquico: Aglomerativo y Divisivo
- DBSCAN: Agrupamiento Basado en Densidad

# Agenda de la Clase

- Introducción al Aprendizaje No Supervisado
- K-Means: Agrupamiento por Centroides
- Clustering Jerárquico: Aglomerativo y Divisivo
- DBSCAN: Agrupamiento Basado en Densidad
- Comparativa y Elección del Método Adecuado

# Agenda de la Clase

- Introducción al Aprendizaje No Supervisado
- K-Means: Agrupamiento por Centroides
- Clustering Jerárquico: Aglomerativo y Divisivo
- DBSCAN: Agrupamiento Basado en Densidad
- Comparativa y Elección del Método Adecuado
- Taller Práctico en Jupyter Notebook



# Dos Paradigmas del Machine Learning

Aprendizaje Supervisado vs. No Supervisado

# Dos Paradigmas del Machine Learning

Aprendizaje Supervisado vs. No Supervisado

## **Aprendizaje Supervisado**

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).
- **Evaluación:** Métrica de error clara sobre un set de prueba (ej. Accuracy, MSE).

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).
- **Evaluación:** Métrica de error clara sobre un set de prueba (ej. Accuracy, MSE).

### Aprendizaje No Supervisado

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).
- **Evaluación:** Métrica de error clara sobre un set de prueba (ej. Accuracy, MSE).

### Aprendizaje No Supervisado

- **Objetivo:** Descubrir patrones, estructuras o subgrupos inherentes en los datos.



# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).
- **Evaluación:** Métrica de error clara sobre un set de prueba (ej. Accuracy, MSE).

### Aprendizaje No Supervisado

- **Objetivo:** Descubrir patrones, estructuras o subgrupos inherentes en los datos.
- **Datos de Entrada:** Solamente features  $X$ , sin una etiqueta  $Y$  asociada.

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).
- **Evaluación:** Métrica de error clara sobre un set de prueba (ej. Accuracy, MSE).

### Aprendizaje No Supervisado

- **Objetivo:** Descubrir patrones, estructuras o subgrupos inherentes en los datos.
- **Datos de Entrada:** Solamente features  $X$ , sin una etiqueta  $Y$  asociada.
- **Ejemplos:** Clustering, Reducción de Dimensionalidad (PCA), Detección de Anomalías.

# Dos Paradigmas del Machine Learning

## Aprendizaje Supervisado vs. No Supervisado

### Aprendizaje Supervisado

- **Objetivo:** Predecir una variable objetivo (etiqueta)  $Y$  a partir de un conjunto de features  $X$ .
- **Datos de Entrada:** Pares  $(X, Y)$ .
- **Ejemplos:** Regresión (predecir un valor continuo), Clasificación (predecir una categoría).
- **Evaluación:** Métrica de error clara sobre un set de prueba (ej. Accuracy, MSE).

### Aprendizaje No Supervisado

- **Objetivo:** Descubrir patrones, estructuras o subgrupos inherentes en los datos.
- **Datos de Entrada:** Solamente features  $X$ , sin una etiqueta  $Y$  asociada.
- **Ejemplos:** Clustering, Reducción de Dimensionalidad (PCA), Detección de Anomalías.
- **Evaluación:** Más subjetiva, a menudo exploratoria y sin una “verdad” única.

# ¿Qué es el Clustering?

## Descubriendo Grupos en los Datos

- **Definición:** Es la tarea de agrupar un conjunto de objetos de tal manera que los objetos en el mismo grupo (o **cluster**) sean más similares entre sí que con los de otros grupos.

# ¿Qué es el Clustering?

## Descubriendo Grupos en los Datos

- **Definición:** Es la tarea de agrupar un conjunto de objetos de tal manera que los objetos en el mismo grupo (o **cluster**) sean más similares entre sí que con los de otros grupos.
- **Principio Clave:** Maximizar la similitud *intra-cluster* y minimizar la similitud *inter-cluster*.

# ¿Qué es el Clustering?

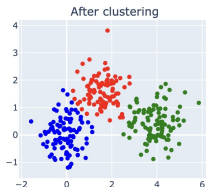
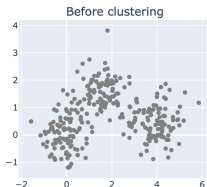
## Descubriendo Grupos en los Datos

- **Definición:** Es la tarea de agrupar un conjunto de objetos de tal manera que los objetos en el mismo grupo (o **cluster**) sean más similares entre sí que con los de otros grupos.
- **Principio Clave:** Maximizar la similitud *intra-cluster* y minimizar la similitud *inter-cluster*.
- La “similitud” se define comúnmente a través de una **métrica de distancia** (ej. distancia Euclidiana) o una medida de correlación.

# ¿Qué es el Clustering?

## Descubriendo Grupos en los Datos

- **Definición:** Es la tarea de agrupar un conjunto de objetos de tal manera que los objetos en el mismo grupo (o **cluster**) sean más similares entre sí que con los de otros grupos.
- **Principio Clave:** Maximizar la similitud *intra-cluster* y minimizar la similitud *inter-cluster*.
- La “similitud” se define comúnmente a través de una **métrica de distancia** (ej. distancia Euclidiana) o una medida de correlación.
- **Aplicaciones:** Segmentación de clientes en marketing, agrupación de documentos, bioinformática (agrupar genes con patrones de expresión similares), etc.



# K-Means: Particionamiento por Centroides

- Es un algoritmo de **particionamiento**: divide el dataset en un número  $K$  de clusters predefinido por el usuario.



# K-Means: Particionamiento por Centroides

- Es un algoritmo de **particionamiento**: divide el dataset en un número  $K$  de clusters predefinido por el usuario.
- Cada cluster es representado por su **centroide**, que es la media de todos los puntos de datos pertenecientes a ese cluster.

# K-Means: Particionamiento por Centroides

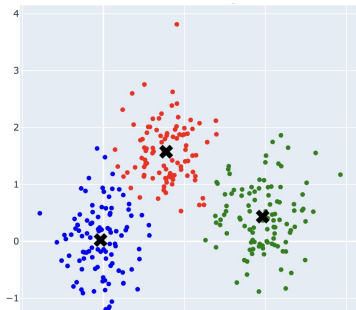
- Es un algoritmo de **particionamiento**: divide el dataset en un número  $K$  de clusters predefinido por el usuario.
- Cada cluster es representado por su **centroide**, que es la media de todos los puntos de datos pertenecientes a ese cluster.
- **Objetivo**: Minimizar la suma de las distancias cuadradas entre cada punto y el centroide de su cluster asignado.

# K-Means: Particionamiento por Centroides

- Es un algoritmo de **particionamiento**: divide el dataset en un número  $K$  de clusters predefinido por el usuario.
- Cada cluster es representado por su **centroide**, que es la media de todos los puntos de datos pertenecientes a ese cluster.
- **Objetivo**: Minimizar la suma de las distancias cuadradas entre cada punto y el centroide de su cluster asignado.
- A esta medida se le conoce como **Inercia** o *Within-Cluster Sum of Squares (WCSS)*.

# K-Means: Particionamiento por Centroides

- Es un algoritmo de **particionamiento**: divide el dataset en un número  $K$  de clusters predefinido por el usuario.
- Cada cluster es representado por su **centroide**, que es la media de todos los puntos de datos pertenecientes a ese cluster.
- **Objetivo**: Minimizar la suma de las distancias cuadradas entre cada punto y el centroide de su cluster asignado.
- A esta medida se le conoce como **Inercia** o *Within-Cluster Sum of Squares (WCSS)*.



# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .

# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .
- Asignar aleatoriamente  $K$  puntos del dataset como centroides iniciales.

# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .
- Asignar aleatoriamente  $K$  puntos del dataset como centroides iniciales.

## Paso 2: Iteración (Repetir hasta la convergencia)



# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .
- Asignar aleatoriamente  $K$  puntos del dataset como centroides iniciales.

## Paso 2: Iteración (Repetir hasta la convergencia)

- 1 **Paso de Asignación:** Asignar cada punto de dato al cluster cuyo centroide esté más cercano (usando, e.g. distancia Euclidiana).

# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .
- Asignar aleatoriamente  $K$  puntos del dataset como centroides iniciales.

## Paso 2: Iteración (Repetir hasta la convergencia)

- 1 **Paso de Asignación:** Asignar cada punto de dato al cluster cuyo centroide esté más cercano (usando, e.g. distancia Euclidiana).
- 2 **Paso de Actualización:** Recalcular el centroide de cada cluster como la media de todos los puntos asignados a él.

# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .
- Asignar aleatoriamente  $K$  puntos del dataset como centroides iniciales.

## Paso 2: Iteración (Repetir hasta la convergencia)

- 1 **Paso de Asignación:** Asignar cada punto de dato al cluster cuyo centroide esté más cercano (usando, e.g. distancia Euclidiana).
- 2 **Paso de Actualización:** Recalcular el centroide de cada cluster como la media de todos los puntos asignados a él.

**Convergencia:** El algoritmo converge cuando las asignaciones de los clusters no cambian en una iteración.

# El Algoritmo K-Means: Pasos

## Paso 1: Inicialización

- Elegir el número de clusters,  $K$ .
- Asignar aleatoriamente  $K$  puntos del dataset como centroides iniciales.

## Paso 2: Iteración (Repetir hasta la convergencia)

- 1 **Paso de Asignación:** Asignar cada punto de dato al cluster cuyo centroide esté más cercano (usando, e.g. distancia Euclidiana).
- 2 **Paso de Actualización:** Recalcular el centroide de cada cluster como la media de todos los puntos asignados a él.

**Convergencia:** El algoritmo converge cuando las asignaciones de los clusters no cambian en una iteración.

# K-Means: Formulación Matemática

# K-Means: Formulación Matemática

- Buscamos una partición de las  $n$  observaciones en  $K$  conjuntos disjuntos  $C_1, C_2, \dots, C_K$ .

# K-Means: Formulación Matemática

- Buscamos una partición de las  $n$  observaciones en  $K$  conjuntos disjuntos  $C_1, C_2, \dots, C_K$ .
- El objetivo es minimizar la **Inercia**, que es la variación total intra-cluster:

$$J(C, \mu) = \sum_{k=1}^K W(C_k); \quad \min_{C_1, \dots, C_K} J(C, \mu)$$

# K-Means: Formulación Matemática

- Buscamos una partición de las  $n$  observaciones en  $K$  conjuntos disjuntos  $C_1, C_2, \dots, C_K$ .
- El objetivo es minimizar la **Inercia**, que es la variación total intra-cluster:

$$J(C, \mu) = \sum_{k=1}^K W(C_k); \quad \min_{C_1, \dots, C_K} J(C, \mu)$$

- Donde  $W(C_k)$  es la suma de las distancias cuadradas de los puntos en el cluster  $k$  a su centroide  $\mu_k$ :

$$W(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$



# K-Means: Formulación Matemática

- Buscamos una partición de las  $n$  observaciones en  $K$  conjuntos disjuntos  $C_1, C_2, \dots, C_K$ .
- El objetivo es minimizar la **Inercia**, que es la variación total intra-cluster:

$$J(C, \mu) = \sum_{k=1}^K W(C_k); \quad \min_{C_1, \dots, C_K} J(C, \mu)$$

- Donde  $W(C_k)$  es la suma de las distancias cuadradas de los puntos en el cluster  $k$  a su centroide  $\mu_k$ :

$$W(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

- El centroide  $\mu_k$  es el vector medio de los puntos en el cluster  $C_k$ :

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

# K-Means: Formulación Matemática

- Buscamos una partición de las  $n$  observaciones en  $K$  conjuntos disjuntos  $C_1, C_2, \dots, C_K$ .
- El objetivo es minimizar la **Inercia**, que es la variación total intra-cluster:

$$J(C, \mu) = \sum_{k=1}^K W(C_k); \quad \min_{C_1, \dots, C_K} J(C, \mu)$$

- Donde  $W(C_k)$  es la suma de las distancias cuadradas de los puntos en el cluster  $k$  a su centroide  $\mu_k$ :

$$W(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

- El centroide  $\mu_k$  es el vector medio de los puntos en el cluster  $C_k$ :

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

- Un valor bajo de Inercia indica clusters más densos y compactos.

# K-Means: Garantía de Convergencia

Un Resultado Matemático Clave

# K-Means: Garantía de Convergencia

Un Resultado Matemático Clave

## Teorema

El algoritmo K-Means **siempre converge** a un óptimo local.

# K-Means: Garantía de Convergencia

## Un Resultado Matemático Clave

### Teorema

El algoritmo K-Means **siempre converge** a un óptimo local.

- **¿Por qué?** En cada paso, el valor de la función objetivo (Inercia) está garantizado a **disminuir o permanecer igual**. Nunca aumenta.

# K-Means: Garantía de Convergencia

## Un Resultado Matemático Clave

### Teorema

El algoritmo K-Means **siempre converge** a un óptimo local.

- **¿Por qué?** En cada paso, el valor de la función objetivo (Inercia) está garantizado a **disminuir o permanecer igual**. Nunca aumenta.
- **Paso de Asignación:** Al reasignar un punto a un centroide más cercano, por definición, se reduce su contribución a la suma total de distancias cuadradas.

# K-Means: Garantía de Convergencia

## Un Resultado Matemático Clave

### Teorema

El algoritmo K-Means **siempre converge** a un óptimo local.

- **¿Por qué?** En cada paso, el valor de la función objetivo (Inercia) está garantizado a **disminuir o permanecer igual**. Nunca aumenta.
- **Paso de Asignación:** Al reasignar un punto a un centroide más cercano, por definición, se reduce su contribución a la suma total de distancias cuadradas.
- **Paso de Actualización:** La media de un conjunto de puntos es el punto que minimiza la suma de distancias euclidianas al cuadrado a dichos puntos. Al recalcular el centroide, encontramos el “mejor” centro posible para los puntos recién asignados, minimizando la inercia del cluster.

# K-Means: Garantía de Convergencia

## Un Resultado Matemático Clave

### Teorema

El algoritmo K-Means **siempre converge** a un óptimo local.

- **¿Por qué?** En cada paso, el valor de la función objetivo (Inercia) está garantizado a **disminuir o permanecer igual**. Nunca aumenta.
- **Paso de Asignación:** Al reasignar un punto a un centroide más cercano, por definición, se reduce su contribución a la suma total de distancias cuadradas.
- **Paso de Actualización:** La media de un conjunto de puntos es el punto que minimiza la suma de distancias euclidianas al cuadrado a dichos puntos. Al recalcular el centroide, encontramos el “mejor” centro posible para los puntos recién asignados, minimizando la inercia del cluster.

### Para los interesados

Se demuestra formalmente este teorema utilizando que el centroide minimiza la suma de distancias al cuadrado:

$$J^{(t+1)} = J(C^{(t+1)}, \mu^{(t+1)}) \leq J(C^{(t+1)}, \mu^{(t)}) \leq J(C^{(t)}, \mu^{(t)}) = J^{(t)}$$



# Evaluación (1): Inercia y el Método del Codo

# Evaluación (1): Inercia y el Método del Codo

- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.

# Evaluación (1): Inercia y el Método del Codo

- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.
- **Problema:** La inercia siempre disminuye a medida que  $K$  aumenta. Si  $K = n$  (cada punto es un cluster), la inercia es 0. ¡Esto no es útil!

# Evaluación (1): Inercia y el Método del Codo

- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.
- **Problema:** La inercia siempre disminuye a medida que  $K$  aumenta. Si  $K = n$  (cada punto es un cluster), la inercia es 0. ¡Esto no es útil!
- **Método del Codo (Elbow Method):**

# Evaluación (1): Inercia y el Método del Codo

- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.
- **Problema:** La inercia siempre disminuye a medida que  $K$  aumenta. Si  $K = n$  (cada punto es un cluster), la inercia es 0. ¡Esto no es útil!
- **Método del Codo (Elbow Method):**
  - 1 Se ejecuta K-Means para un rango de valores de  $K$  (ej.,  $K = 1, \dots, 10$ ).

# Evaluación (1): Inercia y el Método del Codo

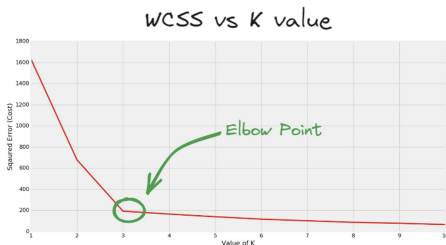
- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.
- **Problema:** La inercia siempre disminuye a medida que  $K$  aumenta. Si  $K = n$  (cada punto es un cluster), la inercia es 0. ¡Esto no es útil!
- **Método del Codo (Elbow Method):**
  - 1 Se ejecuta K-Means para un rango de valores de  $K$  (ej.,  $K = 1, \dots, 10$ ).
  - 2 Se grafica la Inercia (WCSS) en función de  $K$ .

# Evaluación (1): Inercia y el Método del Codo

- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.
- **Problema:** La inercia siempre disminuye a medida que  $K$  aumenta. Si  $K = n$  (cada punto es un cluster), la inercia es 0. ¡Esto no es útil!
- **Método del Codo (Elbow Method):**
  - 1 Se ejecuta K-Means para un rango de valores de  $K$  (ej.,  $K = 1, \dots, 10$ ).
  - 2 Se grafica la Inercia (WCSS) en función de  $K$ .
  - 3 Se busca el “codo” en la gráfica: el punto donde la tasa de disminución de la inercia se aplana significativamente. Este punto es un buen candidato para el valor óptimo de  $K$ .

# Evaluación (1): Inercia y el Método del Codo

- La **Inercia** mide la cohesión del clustering. Un valor más bajo es generalmente mejor.
- **Problema:** La inercia siempre disminuye a medida que  $K$  aumenta. Si  $K = n$  (cada punto es un cluster), la inercia es 0. ¡Esto no es útil!
- **Método del Codo (Elbow Method):**
  - 1 Se ejecuta K-Means para un rango de valores de  $K$  (ej.,  $K = 1, \dots, 10$ ).
  - 2 Se grafica la Inercia (WCSS) en función de  $K$ .
  - 3 Se busca el “codo” en la gráfica: el punto donde la tasa de disminución de la inercia se aplana significativamente. Este punto es un buen candidato para el valor óptimo de  $K$ .





# Evaluación (2): Coeficiente de Silueta

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :
  - $a(i)$ : Distancia media a otros puntos en el **mismo** cluster.

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :
  - $a(i)$ : Distancia media a otros puntos en el **mismo** cluster.
  - $b(i)$ : Distancia media a los puntos en el cluster más **cercano** (distinto al suyo).

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :
  - $a(i)$ : Distancia media a otros puntos en el **mismo** cluster.
  - $b(i)$ : Distancia media a los puntos en el cluster más **cercano** (distinto al suyo).
- **Coeficiente de Silueta** para el punto  $i$ :

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :
  - $a(i)$ : Distancia media a otros puntos en el **mismo** cluster.
  - $b(i)$ : Distancia media a los puntos en el cluster más **cercano** (distinto al suyo).
- **Coeficiente de Silueta** para el punto  $i$ :

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

- **Interpretación del valor (entre -1 y 1):**

## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :
  - $a(i)$ : Distancia media a otros puntos en el **mismo** cluster.
  - $b(i)$ : Distancia media a los puntos en el cluster más **cercano** (distinto al suyo).
- **Coeficiente de Silueta** para el punto  $i$ :

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

- **Interpretación del valor (entre -1 y 1):**
  - $s(i) \approx 1 \implies$  El punto está bien agrupado.
  - $s(i) \approx 0 \implies$  El punto está en el límite entre dos clusters.
  - $s(i) \approx -1 \implies$  El punto podría estar en el cluster incorrecto.



## Evaluación (2): Coeficiente de Silueta

- Mide qué tan bien una observación está agrupada, considerando tanto la **cohesión** (proximidad a su propio cluster) como la **separación** (lejanía de otros clusters).
- Para cada punto  $i$ :
  - $a(i)$ : Distancia media a otros puntos en el **mismo** cluster.
  - $b(i)$ : Distancia media a los puntos en el cluster más **cercano** (distinto al suyo).
- **Coeficiente de Silueta** para el punto  $i$ :

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

- **Interpretación del valor (entre -1 y 1):**
  - $s(i) \approx 1 \implies$  El punto está bien agrupado.
  - $s(i) \approx 0 \implies$  El punto está en el límite entre dos clusters.
  - $s(i) \approx -1 \implies$  El punto podría estar en el cluster incorrecto.
- El score de silueta de un clustering es el promedio de  $s(i)$  sobre todos los puntos.

# Pausa

(15 minutos)

# Clustering Jerárquico: Creando una Taxonomía

# Clustering Jerárquico: Creando una Taxonomía

- A diferencia de K-Means, **no requiere** pre-especificar el número de clusters  $K$ .

# Clustering Jerárquico: Creando una Taxonomía

- A diferencia de K-Means, **no requiere** pre-especificar el número de clusters  $K$ .
- Crea una jerarquía de clusters que se puede visualizar con un diagrama de árbol llamado **dendrograma**.

# Clustering Jerárquico: Creando una Taxonomía

- A diferencia de K-Means, **no requiere** pre-especificar el número de clusters  $K$ .
- Crea una jerarquía de clusters que se puede visualizar con un diagrama de árbol llamado **dendrograma**.
- **Dos enfoques principales:**

# Clustering Jerárquico: Creando una Taxonomía

- A diferencia de K-Means, **no requiere** pre-especificar el número de clusters  $K$ .
- Crea una jerarquía de clusters que se puede visualizar con un diagrama de árbol llamado **dendrograma**.
- **Dos enfoques principales:**
  - **Aglomerativo (Bottom-Up):** El más común. Comienza con cada punto como un cluster individual y fusiona progresivamente los pares de clusters más cercanos hasta que solo queda uno.

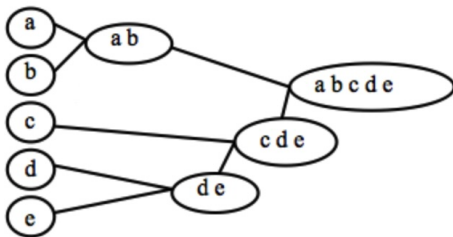
# Clustering Jerárquico: Creando una Taxonomía

- A diferencia de K-Means, **no requiere** pre-especificar el número de clusters  $K$ .
- Crea una jerarquía de clusters que se puede visualizar con un diagrama de árbol llamado **dendrograma**.
- **Dos enfoques principales:**
  - **Aglomerativo (Bottom-Up):** El más común. Comienza con cada punto como un cluster individual y fusiona progresivamente los pares de clusters más cercanos hasta que solo queda uno.
  - **Divisivo (Top-Down):** Comienza con todos los puntos en un único cluster y los divide recursivamente en sub-clusters.



# Clustering Jerárquico: Creando una Taxonomía

- A diferencia de K-Means, **no requiere** pre-especificar el número de clusters  $K$ .
- Crea una jerarquía de clusters que se puede visualizar con un diagrama de árbol llamado **dendrograma**.
- **Dos enfoques principales:**
  - **Aglomerativo (Bottom-Up):** El más común. Comienza con cada punto como un cluster individual y fusiona progresivamente los pares de clusters más cercanos hasta que solo queda uno.
  - **Divisivo (Top-Down):** Comienza con todos los puntos en un único cluster y los divide recursivamente en sub-clusters.



# Algoritmo Jerárquico Aglomerativo

## Construyendo el Dendrograma "Bottom-Up"

- 1 **Inicialización:** Empezar con  $n$  clusters, donde cada observación es su propio cluster. Calcular todas las  $\binom{n}{2}$  disimilitudes (distancias) entre los pares de observaciones.

# Algoritmo Jerárquico Aglomerativo

## Construyendo el Dendrograma “Bottom-Up”

- ❶ **Inicialización:** Empezar con  $n$  clusters, donde cada observación es su propio cluster. Calcular todas las  $\binom{n}{2}$  disimilitudes (distancias) entre los pares de observaciones.
- ❷ **Iterar**  $n - 1$  veces (mientras haya más de un cluster):
  - **a) Encontrar par más cercano:** Identificar el par de clusters (A, B) que son más similares (tienen la menor disimilitud) según el criterio de *linkage* elegido.
  - **b) Fusionar:** Unir los clusters A y B en un nuevo cluster único (AB). La disimilitud a la que se fusionaron define la altura en el dendrograma.
  - **c) Actualizar Distancias:** Calcular las disimilitudes entre el nuevo cluster (AB) y todos los demás clusters existentes.

# Algoritmo Jerárquico Aglomerativo

## Construyendo el Dendrograma “Bottom-Up”

- ❶ **Inicialización:** Empezar con  $n$  clusters, donde cada observación es su propio cluster. Calcular todas las  $\binom{n}{2}$  disimilitudes (distancias) entre los pares de observaciones.
- ❷ **Iterar**  $n - 1$  veces (mientras haya más de un cluster):
  - **a) Encontrar par más cercano:** Identificar el par de clusters (A, B) que son más similares (tienen la menor disimilitud) según el criterio de *linkage* elegido.
  - **b) Fusionar:** Unir los clusters A y B en un nuevo cluster único (AB). La disimilitud a la que se fusionaron define la altura en el dendrograma.
  - **c) Actualizar Distancias:** Calcular las disimilitudes entre el nuevo cluster (AB) y todos los demás clusters existentes.
- ❸ **Finalización:** El algoritmo termina cuando todas las observaciones pertenecen a un único cluster.

Este es un algoritmo “voraz” (greedy): en cada paso toma la decisión localmente óptima (la mejor fusión), y esa decisión es final.

# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- 1 **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.

# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- ➊ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ➋ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**

# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- ➊ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ➋ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**
  - b) **Realizar la división:** Partir el cluster seleccionado en dos sub-clusters. La división se debe hacer de forma que los dos nuevos grupos estén lo más separados posible entre sí.

# Enfoque Divisivo ( “Top-Down” ) y Comparación

## Dividiendo el Universo

- ❶ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ❷ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**
  - b) **Realizar la división:** Partir el cluster seleccionado en dos sub-clusters. La división se debe hacer de forma que los dos nuevos grupos estén lo más separados posible entre sí.
    - Este sub-problema es computacionalmente complejo, ya que hay  $2^{k-1} - 1$  formas de dividir un cluster de  $k$  puntos en dos.



# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- ➊ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ➋ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**
  - b) **Realizar la división:** Partir el cluster seleccionado en dos sub-clusters. La división se debe hacer de forma que los dos nuevos grupos estén lo más separados posible entre sí.
    - Este sub-problema es computacionalmente complejo, ya que hay  $2^{k-1} - 1$  formas de dividir un cluster de  $k$  puntos en dos.
  - c) **Repetir** el proceso recursivamente sobre los nuevos clusters hasta que cada observación forme su propio cluster.

# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- ❶ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ❷ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**
  - b) **Realizar la división:** Partir el cluster seleccionado en dos sub-clusters. La división se debe hacer de forma que los dos nuevos grupos estén lo más separados posible entre sí.
    - Este sub-problema es computacionalmente complejo, ya que hay  $2^{k-1} - 1$  formas de dividir un cluster de  $k$  puntos en dos.
  - c) **Repetir** el proceso recursivamente sobre los nuevos clusters hasta que cada observación forme su propio cluster.

**Aglomerativo vs. Divisivo:**

# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- ❶ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ❷ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**
  - b) **Realizar la división:** Partir el cluster seleccionado en dos sub-clusters. La división se debe hacer de forma que los dos nuevos grupos estén lo más separados posible entre sí.
    - Este sub-problema es computacionalmente complejo, ya que hay  $2^{k-1} - 1$  formas de dividir un cluster de  $k$  puntos en dos.
  - c) **Repetir** el proceso recursivamente sobre los nuevos clusters hasta que cada observación forme su propio cluster.

## Aglomerativo vs. Divisivo:

- **Aglomerativo:** Se enfoca en micro-estructuras (qué puntos son muy similares). Es computacionalmente más eficiente y mucho más común en la práctica.

# Enfoque Divisivo (“Top-Down”) y Comparación

## Dividiendo el Universo

- ❶ **Inicialización:** Empezar con un único cluster que contiene todas las  $n$  observaciones.
- ❷ **Iterar:**
  - a) **Seleccionar un cluster a dividir.**
  - b) **Realizar la división:** Partir el cluster seleccionado en dos sub-clusters. La división se debe hacer de forma que los dos nuevos grupos estén lo más separados posible entre sí.
    - Este sub-problema es computacionalmente complejo, ya que hay  $2^{k-1} - 1$  formas de dividir un cluster de  $k$  puntos en dos.
  - c) **Repetir** el proceso recursivamente sobre los nuevos clusters hasta que cada observación forme su propio cluster.

## Aglomerativo vs. Divisivo:

- **Aglomerativo:** Se enfoca en micro-estructuras (qué puntos son muy similares). Es computacionalmente más eficiente y mucho más común en la práctica.
- **Divisivo:** Se enfoca en macro-estructuras (cuáles son las divisiones principales en los datos). Puede ser más preciso, pero su costo computacional es prohibitivo para datasets grandes.

# Interpretando Dendrogramas

# Interpretando Dendrogramas

- El **dendrograma** muestra el orden de las fusiones. Cada hoja es una observación.

# Interpretando Dendrogramas

- El **dendrograma** muestra el orden de las fusiones. Cada hoja es una observación.
- El **eje vertical** representa la disimilitud (distancia) a la que se fusionan los clusters. Fusiones más bajas indican mayor similitud.

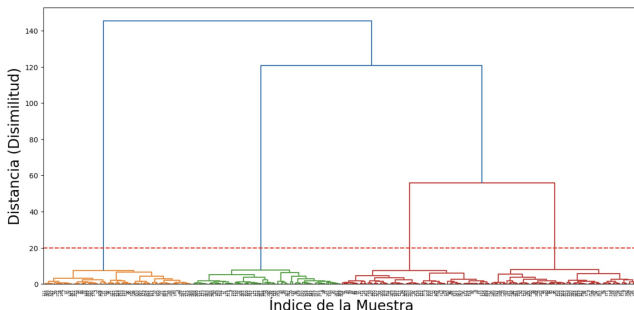
# Interpretando Dendrogramas

- El **dendrograma** muestra el orden de las fusiones. Cada hoja es una observación.
- El **eje vertical** representa la disimilitud (distancia) a la que se fusionan los clusters. Fusiones más bajas indican mayor similitud.
- Se pueden obtener  $K$  clusters haciendo un **corte horizontal** en el dendrograma.



# Interpretando Dendrogramas

- El **dendrograma** muestra el orden de las fusiones. Cada hoja es una observación.
- El **eje vertical** representa la disimilitud (distancia) a la que se fusionan los clusters. Fusiones más bajas indican mayor similitud.
- Se pueden obtener  $K$  clusters haciendo un **corte horizontal** en el dendrograma.



# Criterios de Enlace (Linkage)

# Criterios de Enlace (Linkage)

- **Criterio de Linkage:** ¿Cómo se mide la distancia entre dos clusters?

# Criterios de Enlace (Linkage)

- **Criterio de Linkage:** ¿Cómo se mide la distancia entre dos clusters?
  - **Complete (Máximo):** Distancia máxima entre un punto de un cluster y un punto del otro. Tiende a crear clusters compactos y balanceados.

# Criterios de Enlace (Linkage)

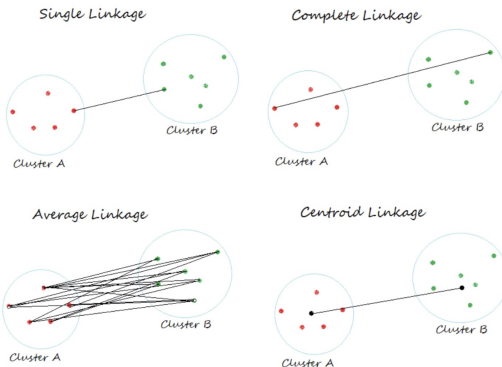
- **Criterio de Linkage:** ¿Cómo se mide la distancia entre dos clusters?
  - **Complete (Máximo):** Distancia máxima entre un punto de un cluster y un punto del otro. Tiende a crear clusters compactos y balanceados.
  - **Single (Mínimo):** Distancia mínima. Puede crear clusters alargados (“efecto cadena”).

# Criterios de Enlace (Linkage)

- **Criterio de Linkage:** ¿Cómo se mide la distancia entre dos clusters?
  - **Complete (Máximo):** Distancia máxima entre un punto de un cluster y un punto del otro. Tiende a crear clusters compactos y balanceados.
  - **Single (Mínimo):** Distancia mínima. Puede crear clusters alargados (“efecto cadena”).
  - **Average (Promedio):** Distancia promedio entre todos los pares de puntos. Un buen compromiso entre los dos anteriores.

# Criterios de Enlace (Linkage)

- **Criterio de Linkage:** ¿Cómo se mide la distancia entre dos clusters?
  - **Complete (Máximo):** Distancia máxima entre un punto de un cluster y un punto del otro. Tiende a crear clusters compactos y balanceados.
  - **Single (Mínimo):** Distancia mínima. Puede crear clusters alargados (“efecto cadena”).
  - **Average (Promedio):** Distancia promedio entre todos los pares de puntos. Un buen compromiso entre los dos anteriores.



# DBSCAN: Clustering Basado en Densidad



# DBSCAN: Clustering Basado en Densidad

- **DBSCAN:** *Density-Based Spatial Clustering of Applications with Noise.*

# DBSCAN: Clustering Basado en Densidad

- **DBSCAN:** *Density-Based Spatial Clustering of Applications with Noise.*
- **Idea Clave:** Un cluster es una región de alta densidad de puntos, separada de otras regiones similares por áreas de baja densidad.

# DBSCAN: Clustering Basado en Densidad

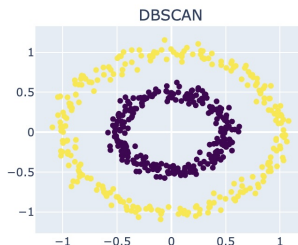
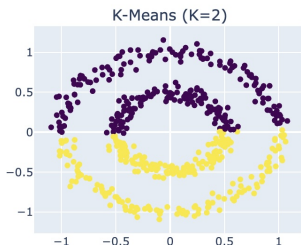
- **DBSCAN:** *Density-Based Spatial Clustering of Applications with Noise.*
- **Idea Clave:** Un cluster es una región de alta densidad de puntos, separada de otras regiones similares por áreas de baja densidad.
- **Ventajas Principales:**
  - No asume que los clusters tienen formas esféricas. Puede encontrar clusters de **formas arbitrarias**.

# DBSCAN: Clustering Basado en Densidad

- **DBSCAN:** *Density-Based Spatial Clustering of Applications with Noise.*
- **Idea Clave:** Un cluster es una región de alta densidad de puntos, separada de otras regiones similares por áreas de baja densidad.
- **Ventajas Principales:**
  - No asume que los clusters tienen formas esféricas. Puede encontrar clusters de **formas arbitrarias**.
  - Es robusto a los *outliers*, a los que identifica explícitamente como **ruido**.

# DBSCAN: Clustering Basado en Densidad

- **DBSCAN:** *Density-Based Spatial Clustering of Applications with Noise.*
- **Idea Clave:** Un cluster es una región de alta densidad de puntos, separada de otras regiones similares por áreas de baja densidad.
- **Ventajas Principales:**
  - No asume que los clusters tienen formas esféricas. Puede encontrar clusters de **formas arbitrarias**.
  - Es robusto a los *outliers*, a los que identifica explícitamente como **ruido**.
  - No requiere especificar el número de clusters de antemano.



# DBSCAN: Conceptos Clave

Puntos Núcleo, Frontera y Ruido

# DBSCAN: Conceptos Clave

Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

# DBSCAN: Conceptos Clave

Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.



# DBSCAN: Conceptos Clave

Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.
- **minPts**: El número mínimo de puntos requeridos dentro de la vecindad  $\epsilon$  para que una región sea considerada densa.

# DBSCAN: Conceptos Clave

Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.
- **minPts**: El número mínimo de puntos requeridos dentro de la vecindad  $\epsilon$  para que una región sea considerada densa.

A partir de ellos, los puntos se clasifican como:

# DBSCAN: Conceptos Clave

Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.
- **minPts**: El número mínimo de puntos requeridos dentro de la vecindad  $\epsilon$  para que una región sea considerada densa.

A partir de ellos, los puntos se clasifican como:

- **Punto Núcleo (Core Point)**: Un punto con al menos **minPts** vecinos (incluyéndose a sí mismo) dentro de su radio  $\epsilon$ .

# DBSCAN: Conceptos Clave

## Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.
- **minPts**: El número mínimo de puntos requeridos dentro de la vecindad  $\epsilon$  para que una región sea considerada densa.

A partir de ellos, los puntos se clasifican como:

- **Punto Núcleo (Core Point)**: Un punto con al menos **minPts** vecinos (incluyéndose a sí mismo) dentro de su radio  $\epsilon$ .
- **Punto Frontera (Border Point)**: Un punto que no es núcleo, pero es vecino de un punto núcleo (está en la “orilla” de un cluster).

# DBSCAN: Conceptos Clave

## Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.
- **minPts**: El número mínimo de puntos requeridos dentro de la vecindad  $\epsilon$  para que una región sea considerada densa.

A partir de ellos, los puntos se clasifican como:

- **Punto Núcleo (Core Point)**: Un punto con al menos **minPts** vecinos (incluyéndose a sí mismo) dentro de su radio  $\epsilon$ .
- **Punto Frontera (Border Point)**: Un punto que no es núcleo, pero es vecino de un punto núcleo (está en la “orilla” de un cluster).
- **Punto de Ruido (Noise Point)**: Un punto que no es ni núcleo ni frontera. Es un outlier.

# DBSCAN: Conceptos Clave

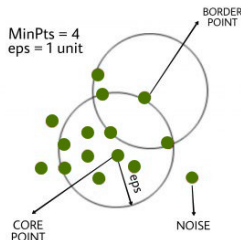
## Puntos Núcleo, Frontera y Ruido

DBSCAN se define por dos hiperparámetros:

- $\epsilon$  (**eps**): El radio de la vecindad a considerar alrededor de cada punto.
- **minPts**: El número mínimo de puntos requeridos dentro de la vecindad  $\epsilon$  para que una región sea considerada densa.

A partir de ellos, los puntos se clasifican como:

- **Punto Núcleo (Core Point)**: Un punto con al menos **minPts** vecinos (incluyéndose a sí mismo) dentro de su radio  $\epsilon$ .
- **Punto Frontera (Border Point)**: Un punto que no es núcleo, pero es vecino de un punto núcleo (está en la “orilla” de un cluster).
- **Punto de Ruido (Noise Point)**: Un punto que no es ni núcleo ni frontera. Es un outlier.



# El Algoritmo DBSCAN: Pasos

# El Algoritmo DBSCAN: Pasos

- 1 Seleccionar un punto arbitrario  $p$  que no haya sido visitado.



# El Algoritmo DBSCAN: Pasos

- 1 Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- 2 Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .

# El Algoritmo DBSCAN: Pasos

- 1 Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- 2 Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .
- 3 **Si** el número de vecinos  $\geq \text{minPts}$ :

# El Algoritmo DBSCAN: Pasos

- 1 Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- 2 Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .
- 3 **Si** el número de vecinos  $\geq \text{minPts}$ :
  - $p$  es un **punto núcleo**. Se crea un nuevo cluster y se le añade  $p$ .

# El Algoritmo DBSCAN: Pasos

- ① Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- ② Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .
- ③ **Si** el número de vecinos  $\geq \text{minPts}$ :
  - $p$  es un **punto núcleo**. Se crea un nuevo cluster y se le añade  $p$ .
  - Se examinan todos los vecinos de  $p$ . Si un vecino  $q$  también es un punto núcleo, sus vecinos se añaden al cluster. El proceso se repite, expandiendo el cluster a partir de todos los puntos núcleo alcanzables (*density-reachable*).

# El Algoritmo DBSCAN: Pasos

- ① Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- ② Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .
- ③ **Si** el número de vecinos  $\geq \text{minPts}$ :
  - $p$  es un **punto núcleo**. Se crea un nuevo cluster y se le añade  $p$ .
  - Se examinan todos los vecinos de  $p$ . Si un vecino  $q$  también es un punto núcleo, sus vecinos se añaden al cluster. El proceso se repite, expandiendo el cluster a partir de todos los puntos núcleo alcanzables (*density-reachable*).
- ④ **Si no:**

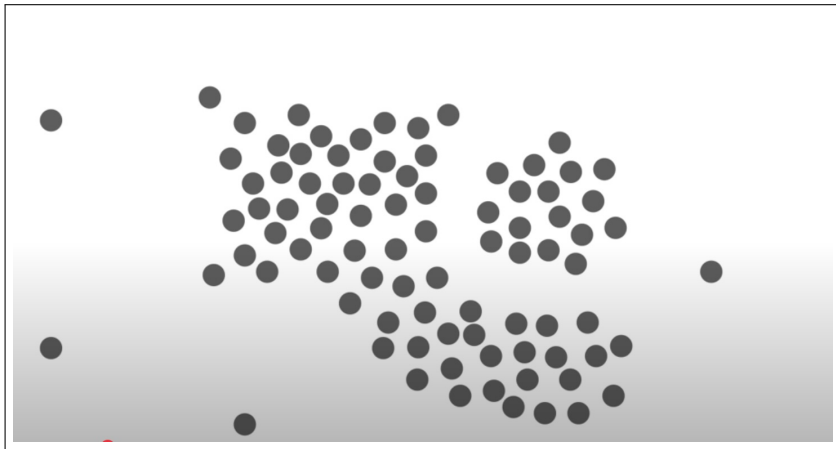
# El Algoritmo DBSCAN: Pasos

- 1 Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- 2 Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .
- 3 **Si** el número de vecinos  $\geq \text{minPts}$ :
  - $p$  es un **punto núcleo**. Se crea un nuevo cluster y se le añade  $p$ .
  - Se examinan todos los vecinos de  $p$ . Si un vecino  $q$  también es un punto núcleo, sus vecinos se añaden al cluster. El proceso se repite, expandiendo el cluster a partir de todos los puntos núcleo alcanzables (*density-reachable*).
- 4 **Si no**:
  - $p$  es marcado (temporalmente) como **ruido**. Puede ser “reclamado” más tarde por un cluster si resulta ser un punto frontera.

# El Algoritmo DBSCAN: Pasos

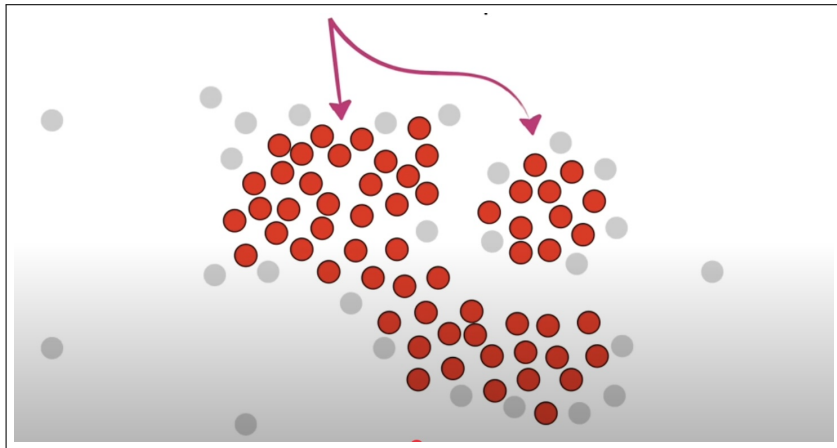
- 1 Seleccionar un punto arbitrario  $p$  que no haya sido visitado.
- 2 Marcar  $p$  como visitado y encontrar sus vecinos en el radio  $\epsilon$ .
- 3 **Si** el número de vecinos  $\geq \text{minPts}$ :
  - $p$  es un **punto núcleo**. Se crea un nuevo cluster y se le añade  $p$ .
  - Se examinan todos los vecinos de  $p$ . Si un vecino  $q$  también es un punto núcleo, sus vecinos se añaden al cluster. El proceso se repite, expandiendo el cluster a partir de todos los puntos núcleo alcanzables (*density-reachable*).
- 4 **Si no**:
  - $p$  es marcado (temporalmente) como **ruido**. Puede ser “reclamado” más tarde por un cluster si resulta ser un punto frontera.
- 5 Repetir los pasos 1-4 hasta que todos los puntos hayan sido visitados.

# DBSCAN: Ejemplo Visual del Proceso (1)

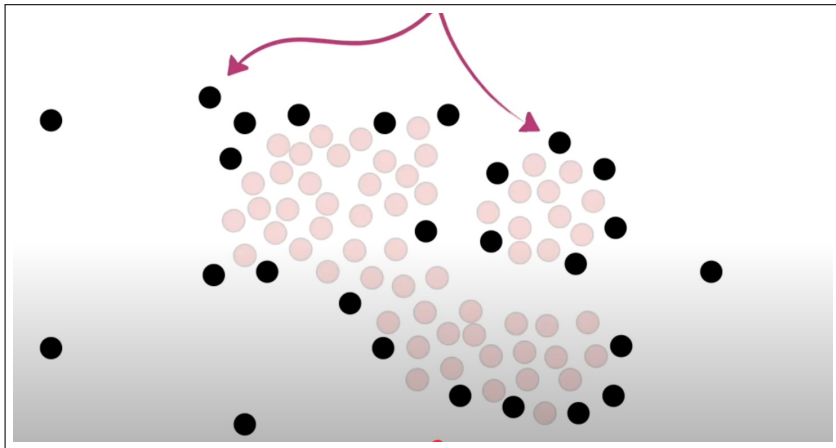




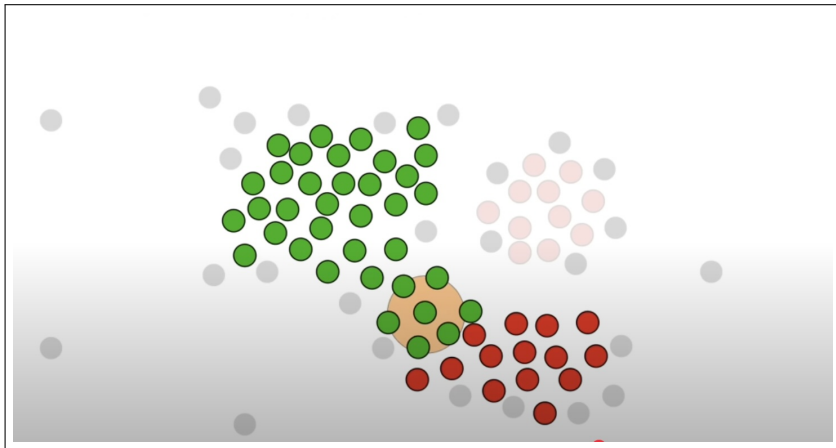
## DBSCAN: Ejemplo Visual del Proceso (2)



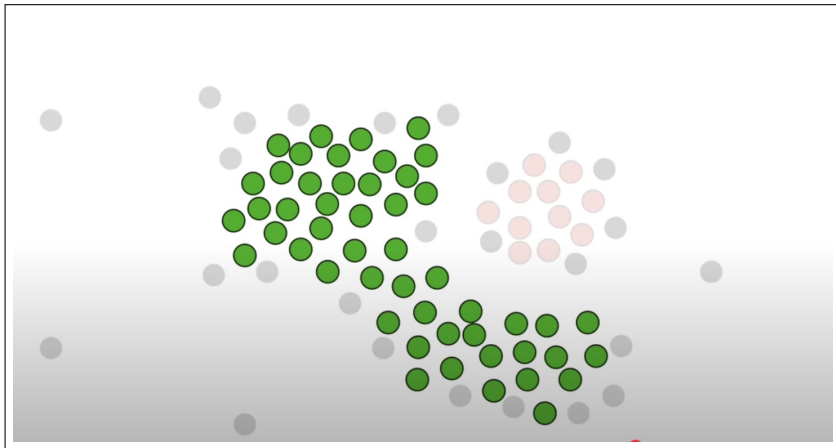
# DBSCAN: Ejemplo Visual del Proceso (3)



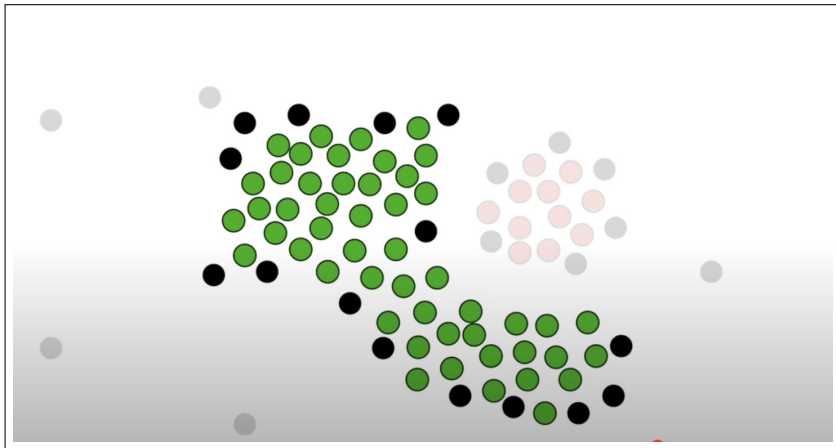
# DBSCAN: Ejemplo Visual del Proceso (4)



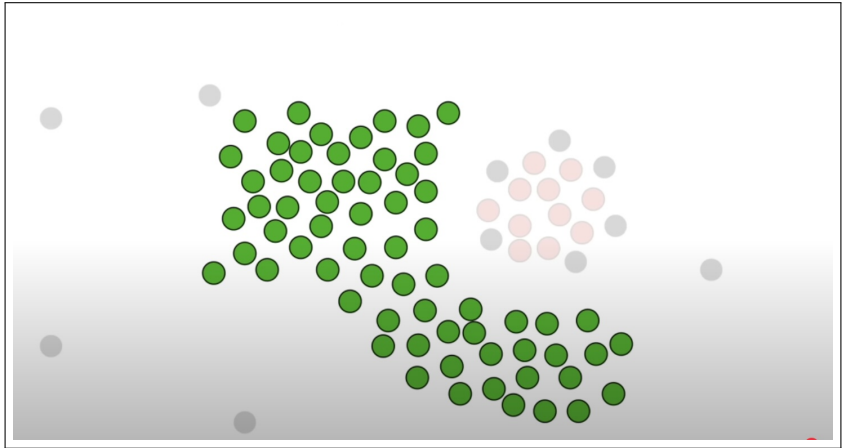
# DBSCAN: Ejemplo Visual del Proceso (5)



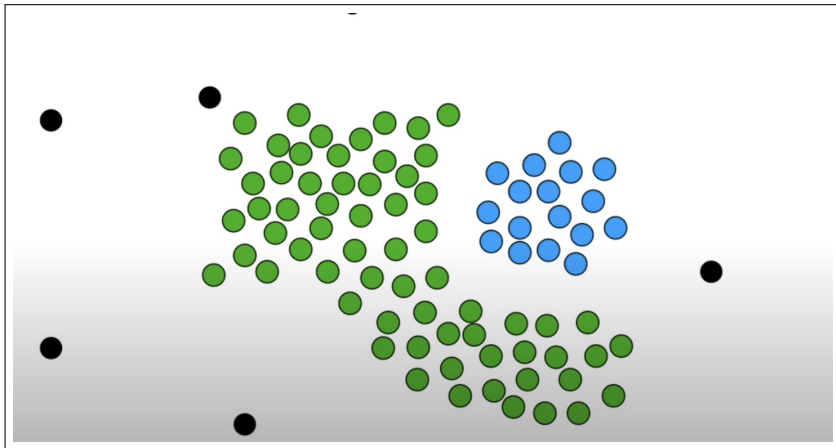
# DBSCAN: Ejemplo Visual del Proceso (6)



# DBSCAN: Ejemplo Visual del Proceso (7)



# DBSCAN: Ejemplo Visual del Proceso (8)



# Comparación de Métodos de Clustering

Método	Ventajas	Desventajas
<b>K-Means</b>	<ul style="list-style-type: none"><li>- Simple y rápido.</li><li>- Escalable a grandes datasets.</li><li>- Bueno para clusters esféricos/convexos.</li></ul>	<ul style="list-style-type: none"><li>- Requiere <math>K</math> a priori.</li><li>- Sensible a la inicialización y outliers.</li><li>- Malo para formas no convexas y densidades variables.</li></ul>
<b>Jerárquico</b>	<ul style="list-style-type: none"><li>- No requiere <math>K</math>.</li><li>- El dendrograma es muy informativo y visual.</li></ul>	<ul style="list-style-type: none"><li>- Computacionalmente costoso (<math>O(n^2)</math> o más).</li><li>- Las fusiones son irreversibles.</li><li>- Sensible al tipo de linkage.</li></ul>
<b>DBSCAN</b>	<ul style="list-style-type: none"><li>- No requiere <math>K</math>.</li><li>- Encuentra formas arbitrarias.</li><li>- Robusto a outliers (ruido).</li></ul>	<ul style="list-style-type: none"><li>- La elección de 'eps' y 'minPts' es crucial y no trivial.</li><li>- No funciona bien con clusters de densidades muy diferentes.</li></ul>



# Resumen de la Clase y Próximos Pasos

# Resumen de la Clase y Próximos Pasos

- El aprendizaje no supervisado, y en particular el clustering, es una herramienta poderosa para **explorar y encontrar estructura** en datos no etiquetados.

# Resumen de la Clase y Próximos Pasos

- El aprendizaje no supervisado, y en particular el clustering, es una herramienta poderosa para **explorar y encontrar estructura** en datos no etiquetados.
- **K-Means** agrupa por centroides. Es rápido y efectivo para clusters compactos y esféricos. Se evalúa con *Inercia* y *Coeficiente de Silueta*.

# Resumen de la Clase y Próximos Pasos

- El aprendizaje no supervisado, y en particular el clustering, es una herramienta poderosa para **explorar y encontrar estructura** en datos no etiquetados.
- **K-Means** agrupa por centroides. Es rápido y efectivo para clusters compactos y esféricos. Se evalúa con *Inercia* y *Coeficiente de Silueta*.
- **Clustering Jerárquico** construye una taxonomía de los datos, visualizada en un dendrograma, que ofrece una visión multi-escala de las agrupaciones.

# Resumen de la Clase y Próximos Pasos

- El aprendizaje no supervisado, y en particular el clustering, es una herramienta poderosa para **explorar y encontrar estructura** en datos no etiquetados.
- **K-Means** agrupa por centroides. Es rápido y efectivo para clusters compactos y esféricos. Se evalúa con *Inercia* y *Coeficiente de Silueta*.
- **Clustering Jerárquico** construye una taxonomía de los datos, visualizada en un dendrograma, que ofrece una visión multi-escala de las agrupaciones.
- **DBSCAN** agrupa por densidad. Es excelente para identificar clusters de formas complejas y para separar el ruido de la señal.

# Resumen de la Clase y Próximos Pasos

- El aprendizaje no supervisado, y en particular el clustering, es una herramienta poderosa para **explorar y encontrar estructura** en datos no etiquetados.
- **K-Means** agrupa por centroides. Es rápido y efectivo para clusters compactos y esféricos. Se evalúa con *Inercia* y *Coeficiente de Silueta*.
- **Clustering Jerárquico** construye una taxonomía de los datos, visualizada en un dendrograma, que ofrece una visión multi-escala de las agrupaciones.
- **DBSCAN** agrupa por densidad. Es excelente para identificar clusters de formas complejas y para separar el ruido de la señal.
- La elección del método **depende fundamentalmente de la estructura de los datos** y los objetivos del análisis.

# Resumen de la Clase y Próximos Pasos

- El aprendizaje no supervisado, y en particular el clustering, es una herramienta poderosa para **explorar y encontrar estructura** en datos no etiquetados.
- **K-Means** agrupa por centroides. Es rápido y efectivo para clusters compactos y esféricos. Se evalúa con *Inercia* y *Coeficiente de Silueta*.
- **Clustering Jerárquico** construye una taxonomía de los datos, visualizada en un dendrograma, que ofrece una visión multi-escala de las agrupaciones.
- **DBSCAN** agrupa por densidad. Es excelente para identificar clusters de formas complejas y para separar el ruido de la señal.
- La elección del método **depende fundamentalmente de la estructura de los datos** y los objetivos del análisis.
- **A continuación:** Taller Práctico para aplicar estos métodos en Python.