

Aprendizaje Automático No Supervisado

---

# Tema 4. Agrupamiento jerárquico

# Índice

## Esquema

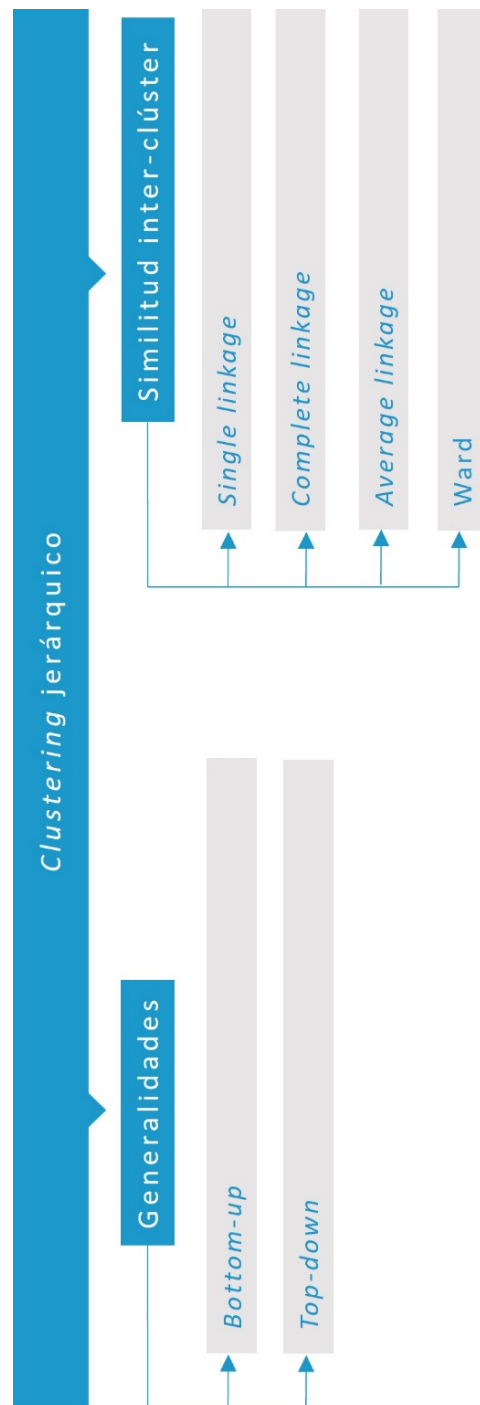
## Ideas clave

- 4.1. Introducción y objetivos
- 4.2. Tipos de clustering jerárquico
- 4.3. Implementación en Python
- 4.4. Comparación entre K-Means y clúster jerárquico
- 4.5. Cuaderno de ejercicios
- 4.6. Referencias bibliográficas

## A fondo

- Una introducción al clustering jerárquico en Python
- ¿Cómo se analiza el clúster jerárquico en la herramienta estadística SPSS?

## Test



## 4.1. Introducción y objetivos

En el universo del aprendizaje no supervisado, el **agrupamiento jerárquico** surge como alternativa al algoritmo K-Means. Se caracteriza por construir una jerarquía de clústeres que pueden ser visualizados a través de un gráfico 2D llamado dendrograma.

El **dendrograma** que es nuestro protagonista de este capítulo, es similar a un árbol que se va dividiendo con base en una secuencia de registros. A diferencia de K-Means no es necesario conocer el número de grupos que se desean crear, haciéndolo más flexible.

El *clustering* jerárquico puede ser aglomerativo o divisivo; veremos cada uno de ellos en detalle.

En este tema nos planteamos los siguientes **objetivos**:

- ▶ Explicar detalladamente qué es el *clustering* jerárquico y cuál es la diferencia con otros métodos de *clustering*.
- ▶ Definir los dos tipos de clúster jerárquico: divisivo y aglomerativo.
- ▶ Aprender a interpretar un dendrograma y cómo utilizarlo para hallar el número óptimo de clústeres.
- ▶ Estudiar los algoritmos más comunes utilizados para *clustering* jerárquico y su implementación en Python.

## 4.2. Tipos de clustering jerárquico

El *clustering* jerárquico es un algoritmo de aprendizaje no supervisado, agrupa los datos en jerarquía y de allí su nombre. La jerarquía se representa a través de un dendrograma o árbol que muestra las relaciones de inclusión de los diferentes grupos (Van Der Post y Smith, 2023).

Dentro del clúster jerárquico tenemos dos grandes enfoques que trabajaremos de manera más detallada:

- ▶ *Clustering* jerárquico aglomerativo o *bottom-up*.
- ▶ *Clustering* jerárquico divisivo o *top-down*.

### Clustering jerárquico aglomerativo o bottom-up

#### Descripción

Se denomina aglomerativo porque empieza con grupos muy pequeños hasta formar un grupo. En este caso, cada punto del conjunto de datos comienza como un clúster individual, luego se combinan repetidamente en función de su similitud hasta que todos los puntos hagan parte de un solo clúster o se alcance un criterio de parada.

#### Proceso

Asumiendo que se tienen un montón de puntos, el funcionamiento es el siguiente (Dissanayake, 2024):

- ▶ Primero, cada punto es un grupo en sí mismo. Como se puede ver en la Figura 1.

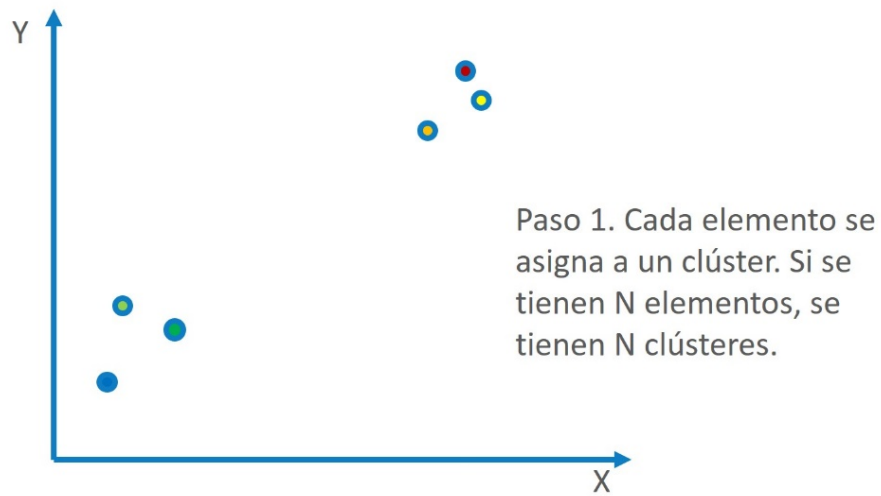


Figura 1. Cada punto es un clúster. Fuente: elaboración propia.

- Luego se agrupan o dos puntos o dos grupos o un punto y un grupo cercano.

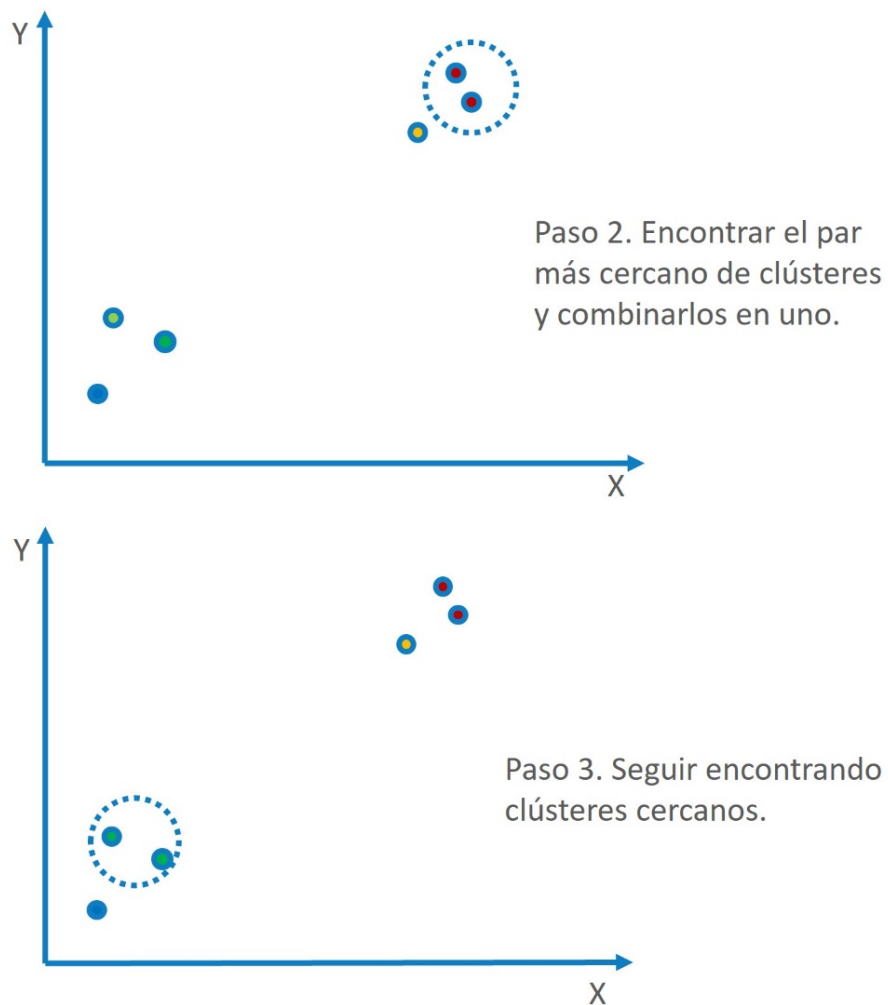


Figura 2. Creando clústeres entre puntos o grupos cercanos. Fuente: elaboración propia.

- ▶ Se repiten los dos pasos anteriores hasta que todos pertenezcan a un mismo grupo. Como se puede ver en la Figura 3.

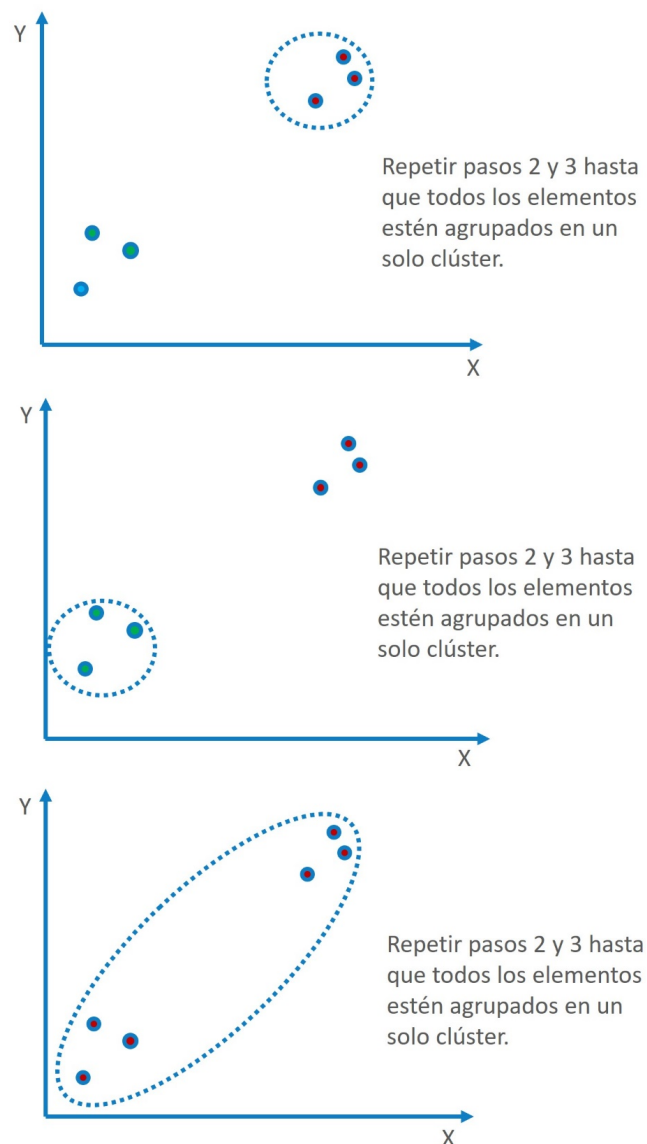


Figura 3. Creación de un clúster. Fuente: elaboración propia.

## Clustering jerárquico divisivo o top-down

### Descripción

Comienza con todos los datos en un solo clúster y se divide recursivamente en clústeres más pequeños.

### Proceso



El método divisivo funciona exactamente al revés del método aglomerativo (Dissanayake, 2024):

- ▶ Comienza guardando en un solo grupo todos los puntos.
- ▶ En la segunda iteración, divide el grupo en dos pequeños grupos, intenta dividir un grupo más grande en grupos más pequeños.
- ▶ En la tercera iteración, se divide aún más en grupos más pequeños.
- ▶ El algoritmo itera hasta que cada punto sea un clúster en sí mismo.

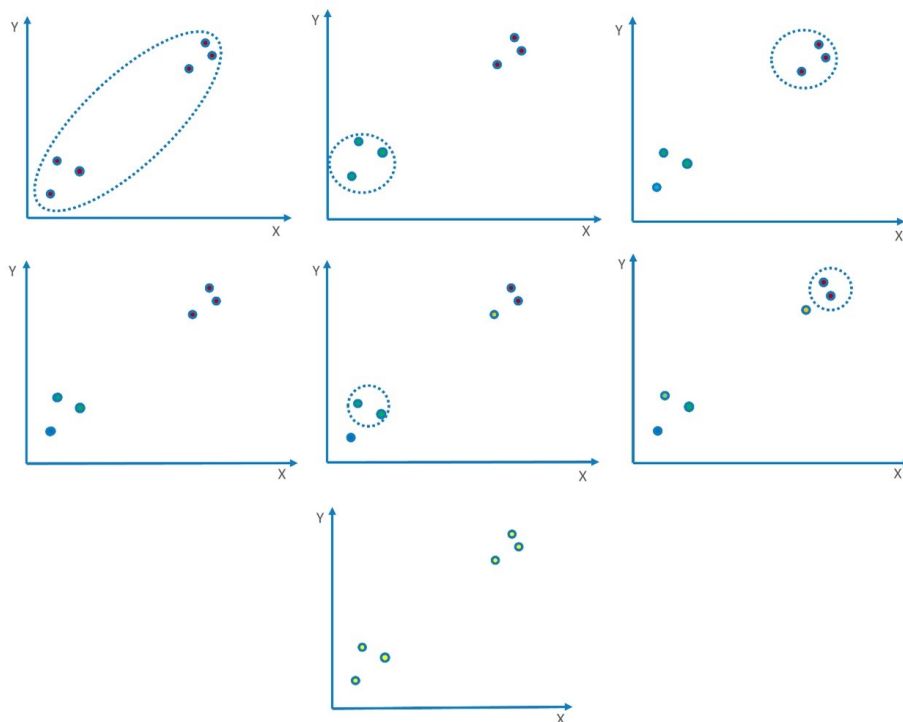


Figura 4. Paso a paso del clúster jerárquico divisivo. Fuente: elaboración propia.

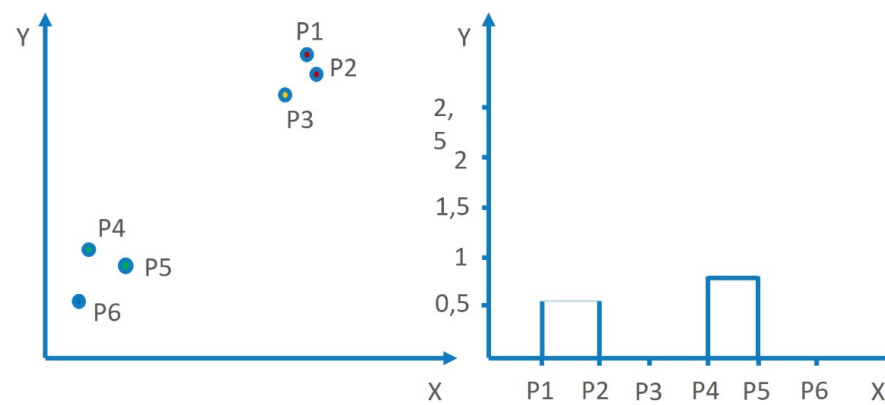
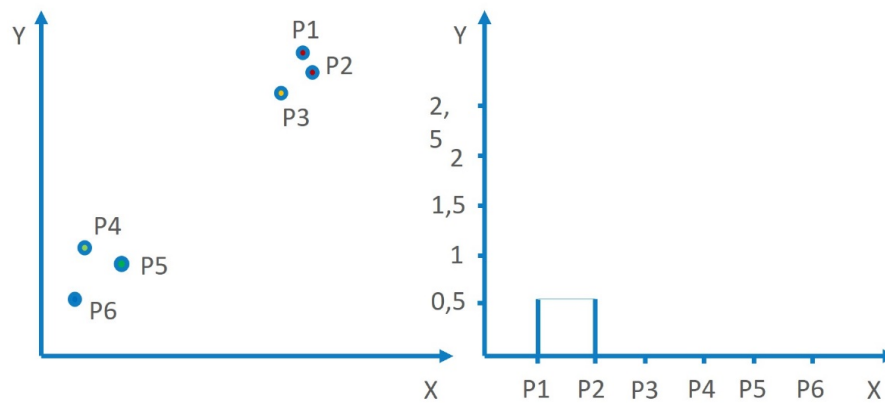
## ¿Cómo dividir los clústeres?

Rara vez se utiliza el método divisivo, normalmente se usa el aglomerativo. Es más natural intentar agrupar en función de la similitud o la distancia que intentar dividir pedazos más pequeños.

Las secuencias de los puntos en un grupo se pueden registrar como un árbol y por eso le llamamos jerarquía. El dendrograma es un árbol que registra la secuencia de fusiones en el caso del agrupamiento aglomerativo o divisiones en el caso de agrupamiento divisivo (Prakash, 2023).

Como ya hemos mencionado, este algoritmo no recibe el número de grupos como hiperparámetros. En este caso, se puede elegir el número de grupos de manera visual.

- ▶ El primer paso es calcular una matriz de proximidad, lo que significa partiendo de los puntos dados:  $P_1, P_2, P_3, \dots, P_n$ . En la matriz se registran los valores de similitud entre cada par de puntos.
- ▶ Se genera un bucle con los siguientes pasos:
  - Fusión de los grupos más cercanos y registro del valor de la distancia entre los grupos en la matriz de proximidad.
  - Se ignoran los valores de la diagonal, ya que es el cálculo de la distancia del punto consigo mismo. Es necesario registrar los valores fuera de la diagonal.
- ▶ Con las distancias de los diferentes clústeres creados se va creando el dendrograma, el detalle algorítmico se puede ver en la Figura 5.



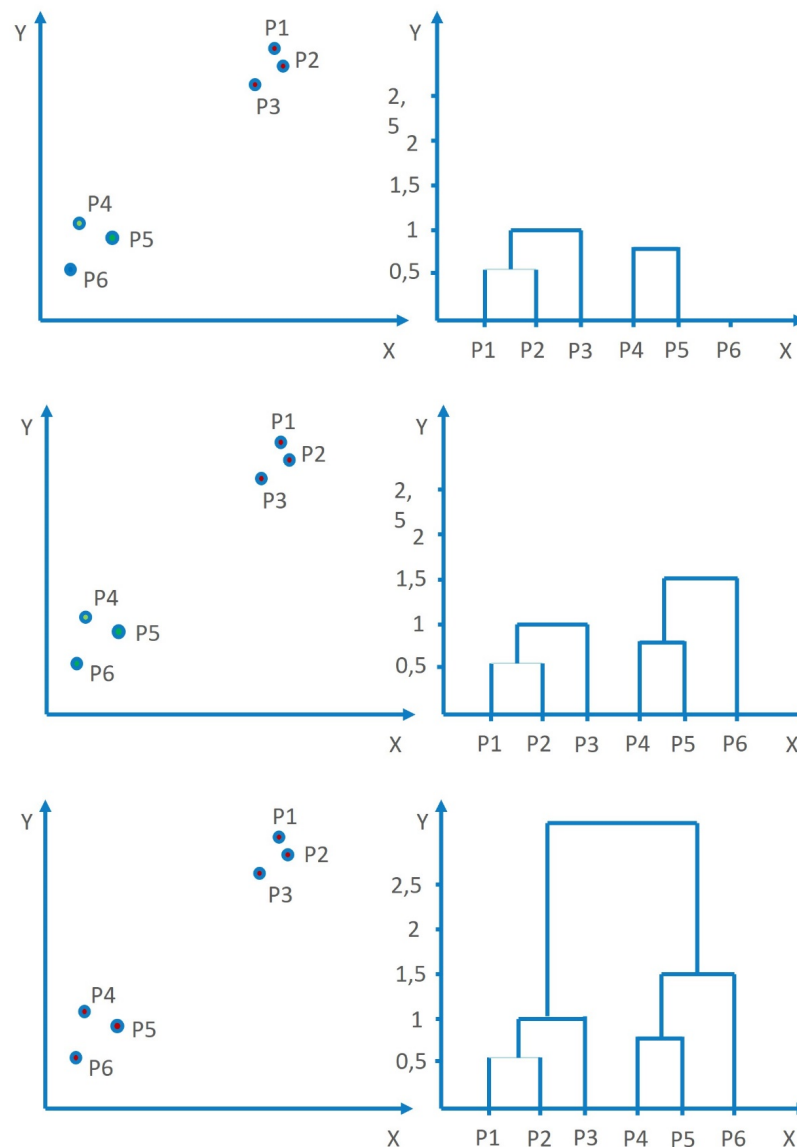


Figura 5. Paso a paso para crear el dendrograma. Fuente: elaboración propia.

- ▶ Se analizan cada una de las divisiones que se pueden realizar en el dendrograma para elegir el número de clústeres ideal.
- Vamos a trazar una línea vertical que haga un primer corte en el diagrama. En este caso se crearían 2 clúster. Uno formado por los puntos P1, P2, P3 y el segundo formado por los puntos P4, P5 y P6 (ver Figura 6).

- Hagamos otro corte más abajo. En este caso, se crean 4 clústeres o grupos (ver Figura 7).
- Y un corte más, creando en esta ocasión 6 clústeres (ver Figura 8).

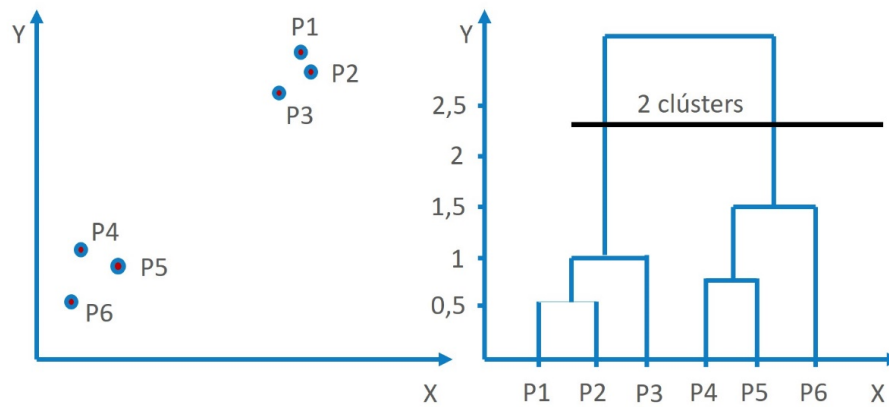


Figura 6. Corte en la parte más alta del dendrograma. Fuente: elaboración propia.

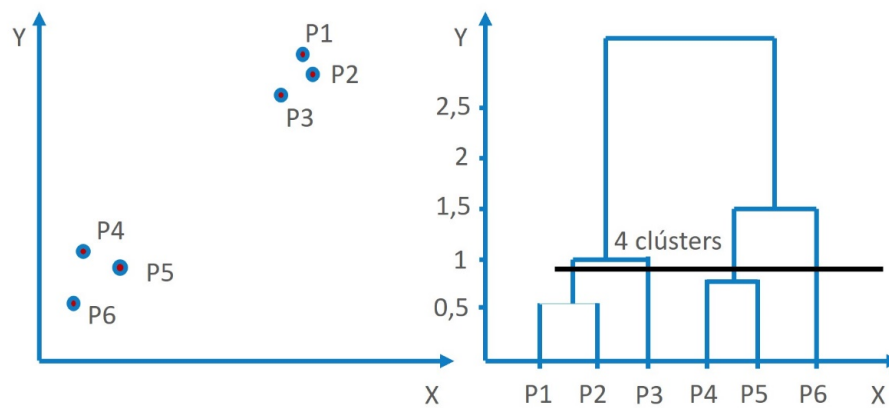


Figura 7. Corte en la parte media del dendrograma. Fuente: elaboración propia.

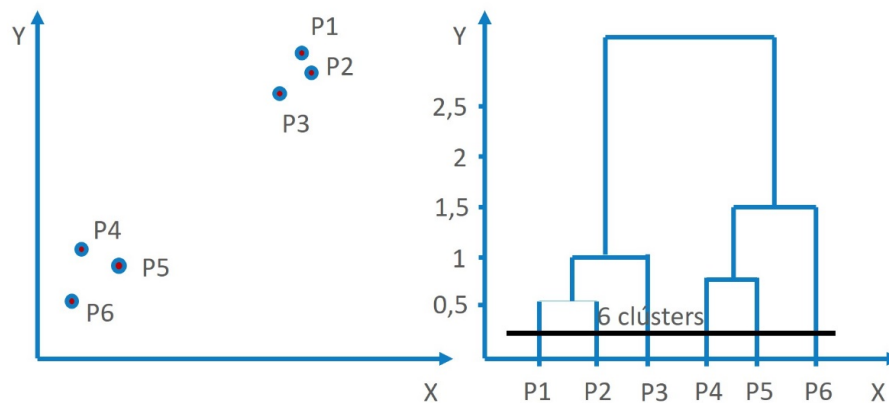


Figura 8. Corte en la parte baja del dendrograma. Fuente: elaboración propia.

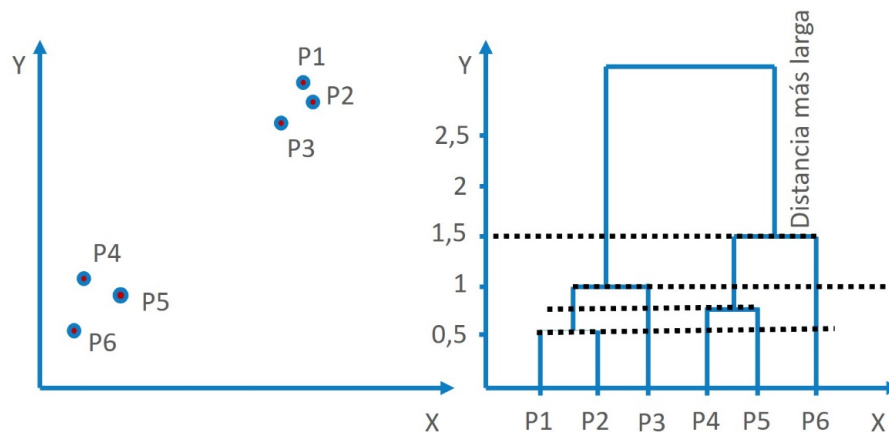


Figura 9. Análisis del mejor corte del dendrograma para decidir el mejor valor para el número de clústeres.

Fuente: elaboración propia.

Para el ejemplo anterior, el mejor corte es el de 2 clústeres. Debido a que es la distancia más larga y garantiza la suficiente distancia inter-clúster haciendo que cada grupo este bien diferenciado.

¿Cómo definimos la similitud inter-clúster? Para dar respuesta a esta pregunta existen diferentes métodos:

- **Mínimo o «single linkage algorithm»:** es la distancia más corta entre dos puntos que pertenezcan a los dos grupos. Se mide la distancia entre los puntos más

cercanos de los diferentes grupos (Figura 10). La fortaleza de este método es que puede manejar formas no elípticas. Y como limitación es que es altamente sensible a ruido y valores *outliers*.

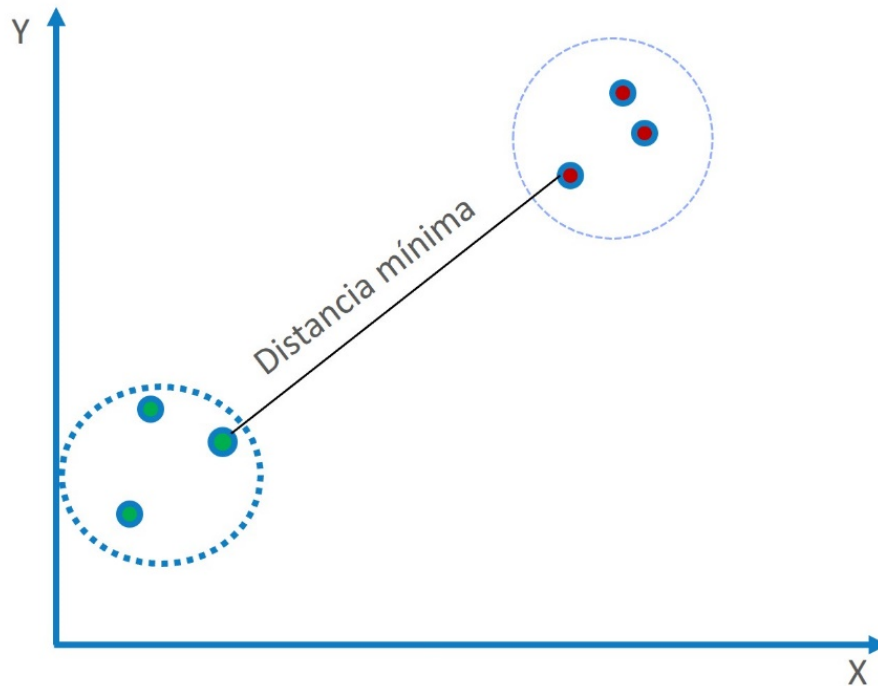


Figura 10. Distancia mínima inter-clúster. Fuente: elaboración propia.

- **Máximo o «complete linkage algorithm»:** calcula la distancia máxima entre cualquier par de puntos que pertenezcan a diferentes grupos. Representa la máxima disimilitud entre conglomerados e indica la presencia de conglomerados bien separados, como se puede observar en la Figura 11. Es menos susceptible al ruido.

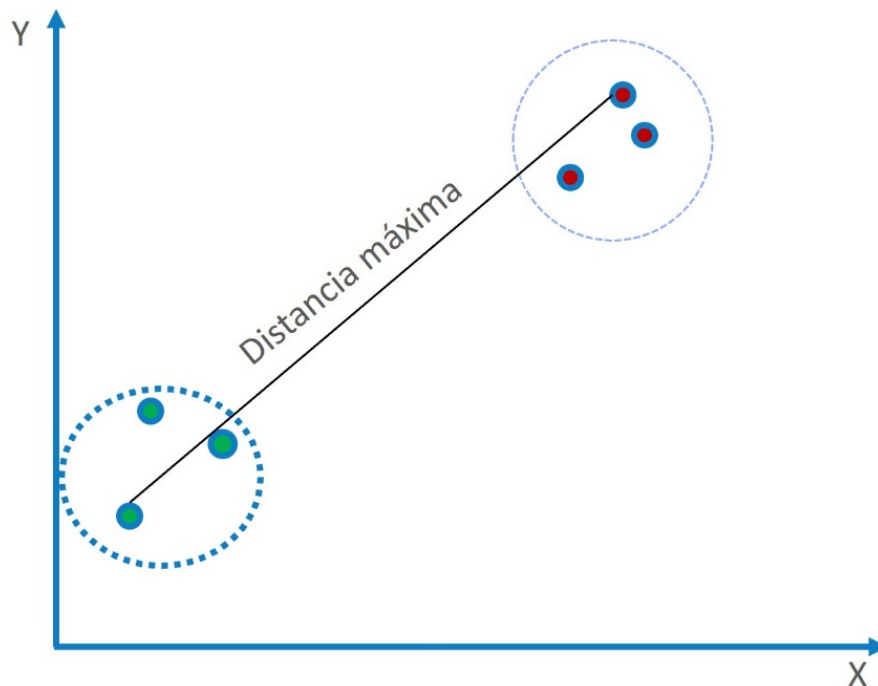


Figura 11. Distancia máxima inter-clúster. Fuente: elaboración propia.

- **Media-grupal «average linkage»:** calcula la distancia promedio entre todos los pares de puntos de datos que pertenecen a los diferentes grupos. Se obtiene una medida de disimilitud promedio entre grupos, como se puede observar en la Figura 12. Este método también es menos susceptible al ruido y a los valores atípicos.



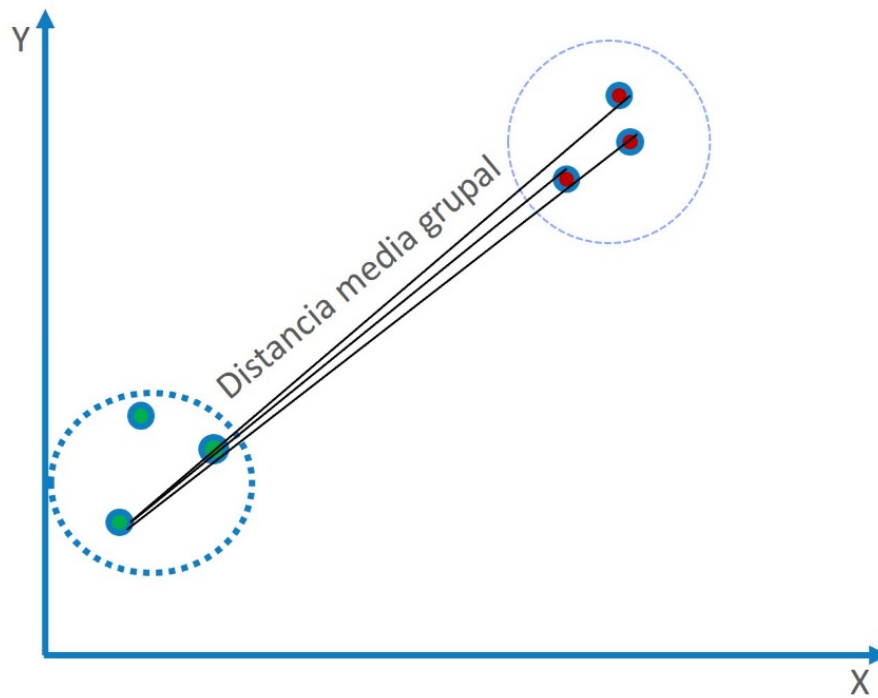


Figura 12. Distancia media grupal inter-clúster. Fuente: elaboración propia.

- **Entre-centroides:** se calcula el centroide de cada grupo y se calcula la distancia entre ellos. Como se puede observar en la Figura 13.

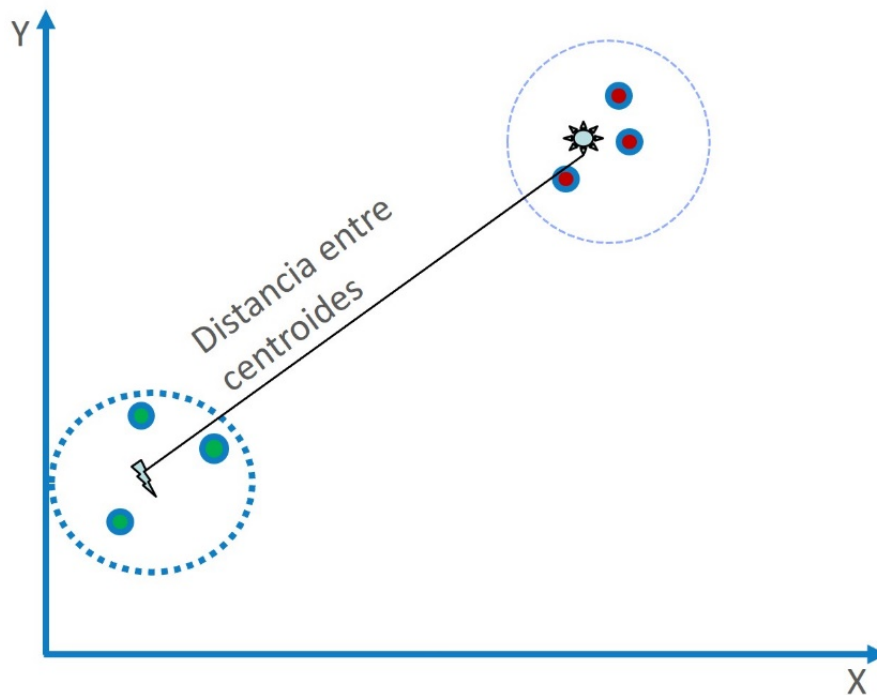


Figura 13. Distancia entre centroides. Fuente: elaboración propia.

- **Método Ward:** es una técnica de enlace utilizada en el *clustering* jerárquico aglomerativo. Su objetivo principal es minimizar la varianza dentro de cada clúster. Cada punto de datos comienza como un clúster individual. En cada paso, se seleccionan los dos clústeres cuya fusión resulta en el menor incremento en la suma total de las varianzas dentro de todos los clústeres. Esta suma se calcula como la suma de las distancias cuadráticas entre cada punto y el centroide del clúster al que pertenece. El proceso de fusión se repite hasta que todos los puntos estén en un solo clúster o se cumpla un criterio de parada específico (por ejemplo, un número predefinido de clústeres).

## 4.3. Implementación en Python

Inspirados en el ejemplo mostrado en el siguiente enlace <https://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/> construiremos el dendograma de un conjunto de datos que corresponde a un análisis psicológico de *tweets* de famosos.

El conjunto de datos tiene los siguientes atributos: grado de apertura, grado de orden, extraversión, empatía, neuroticismo, número medio de palabras usadas en los *tweets* y categoría de actividad laboral.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb

dataframe = pd.read_csv(r" analisis.csv")
dataframe.head()

#140 famosos se les hizo un test Basado en una metodología de psicología
conocida como "Ocean: The Big Five"
#OP --> grado de apertura a nuevas experiencias, curiosidad, arte.
#CO --> grado de orden, pulcritud, organización.
#EX --> Extraversión - grado de timidez, soledad o participación ante el
grupo social
#AG --> Agreeableness - grado de empatía con los demás, temperamento
#NE --> Neuroticismo, - grado de neuroticismo, nerviosismo, irritabilidad,
seguridad en sí mismo.
#Wordcount--> número medio de palabras utilizadas en sus tweets.
#Categoría - actividad laboral del usuario (actor, cantante, etc.)
```

El archivo contiene 9 categorías laborales: actor/actriz, cantante, modelo, tv, series, radio, tecnología, deportes, política y escritor.

```
print(dataframe.groupby('categoria').size())
# Como vemos tenemos 34 cantantes, 27 actores, 17 deportistas, 16
políticos,etc.

#Borramos la categoría para no limitar los clústeres.
```

```
dataframe.drop(['categoria'],1).hist()
plt.show()

def plot_dendrogram(model, **kwargs):
    # Crear una matriz de enlaces y luego se dibuja el dendrograma.

    # crear los recuentos de muestras debajo de cada nodo
    counts = np.zeros(model.children_.shape[0])
    n_samples = len(model.labels_)
    for i, merge in enumerate(model.children_):
        current_count = 0
        for child_idx in merge:
            if child_idx < n_samples:
                current_count += 1
            else:
                current_count += counts[child_idx - n_samples]
        counts[i] = current_count

    linkage_matrix = np.column_stack(
        [model.children_, model.distances_, counts]
    ).astype(float)

    # Dibujar el dendrograma
    dendrogram(linkage_matrix, **kwargs)

from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram

model = AgglomerativeClustering(distance_threshold=0, n_clusters=None,
affinity='euclidean')
# affinity: distancia entre instancias

# distance_threshold=0, El umbral de distancia de enlace o por encima del
cual los clústeres no se fusionan.
# n_clusters=None si no se tienen predefinidos el número de clúster
# linkage{'ward', 'complete', 'average', 'single'}, default='ward':
# * Enlace simple (single linkage): La distancia entre dos clusters se
define como la distancia entre los dos puntos más cercanos, uno de cada
cluster.
# * Enlace promedio (average linkage): distancia promedio entre todos los
pares de puntos, uno de cada cluster.
# * Enlace completo (complete linkage): mide la distancia entre dos
clusters basándose en la distancia máxima entre cualquier par de puntos,
tomando el par más lejano.
# * Enlace Ward: se basa en los criterios de minimización de la varianza.
Busca que la varianza total del nuevo clúster sea lo más baja posible.
```

```
model = model.fit(X)
plt.title('Clúster Jerárquico')
plot_dendrogram(model, truncate_mode="level")
plt.xlabel("Número de instancias en el nodo.")
# Cutting the dendrogram at max_d
plt.axhline(y=45, c='k')
plt.show()
```

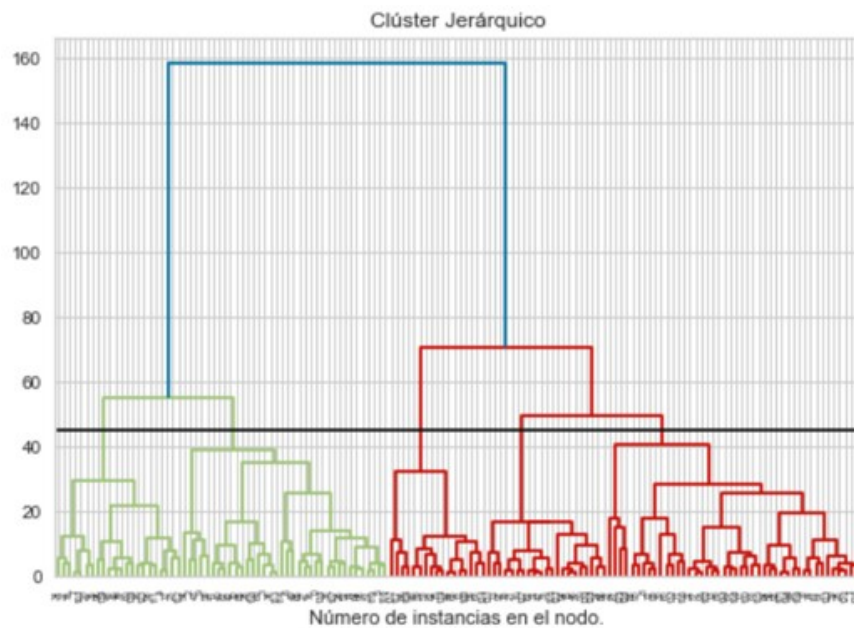


Figura 14. Dendrograma resultante. Fuente: elaboración propia.

Ahora etiquetamos cada instancia según la división que hemos realizado. Se van a generar 5 clústeres.

```
model_1 = AgglomerativeClustering(n_clusters=5, affinity='euclidean')
etiquetas = model_1.fit_predict(X)
# affinity: distancia entre instancias
```

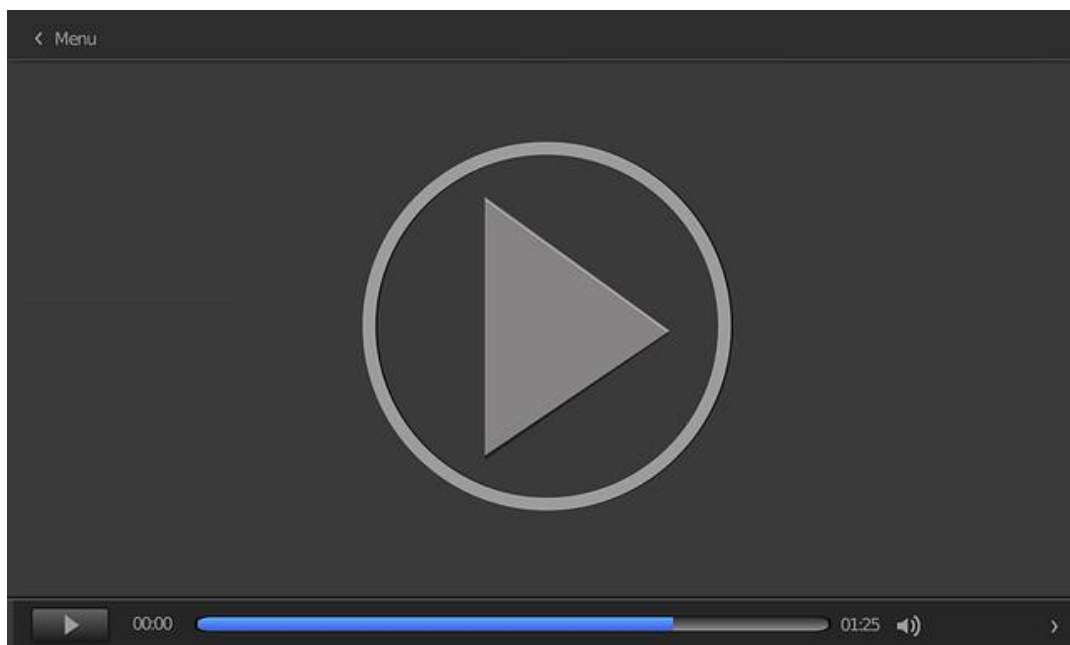
Generándose etiquetas desde la 0 hasta la 4. Como se puede ver en la Figura 15.

etiquetas

```
array([1, 0, 1, 1, 3, 1, 1, 3, 0, 1, 4, 0, 2, 0, 1, 4, 3, 0, 4, 3, 3, 4,  
       0, 3, 4, 0, 1, 1, 1, 0, 0, 4, 0, 4, 3, 3, 1, 3, 1, 0, 1, 1, 1, 1,  
       2, 2, 4, 4, 3, 1, 4, 1, 1, 2, 1, 2, 0, 0, 1, 2, 0, 0, 4, 2, 1, 1,  
       2, 0, 2, 1, 1, 0, 0, 4, 0, 2, 1, 3, 3, 3, 3, 0, 2, 0, 0, 1, 4, 1,  
       0, 0, 0, 0, 0, 0, 4, 1, 3, 4, 3, 0, 2, 0, 0, 3, 2, 4, 1, 1, 0, 0,  
       0, 3, 3, 1, 1, 0, 0, 2, 1, 1, 0, 3, 4, 0, 2, 0, 0, 4, 4, 0, 4, 4,  
       0, 0, 2, 1, 3, 2, 0, 3], dtype=int64)
```

Figura 15. Etiquetas a partir del análisis gráfico del dendrograma. Fuente: elaboración propia.

A continuación, veremos el vídeo **Implementación de clustering jerárquico en Python**.



Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=de6618d3-211d-4070-a01e-b1be010c35ad>

## 4.4. Comparación entre K-Means y clúster jerárquico

Las principales diferencias entre estos dos algoritmos son a nivel de metodología, número de clústeres, forma de los clústeres, interpretabilidad, escalabilidad y sensibilización a la inicialización.

Aspecto	K-Means	Clúster jerárquico
<b>Metodología</b>	Algoritmo de partición iterativo. Inicializa aleatoriamente los k centroides y asigna iterativamente los puntos de datos al centroide más cercano, actualiza centroides y repite el proceso hasta convergencia.	Construye jerarquías de clústeres fusionando o dividiendo clústeres de forma iterativa. Se puede realizar de arriba hacia abajo o de abajo hacia arriba.
<b>Número de clústeres</b>	Requiere que se especifique de antemano el número de clústeres.	No requiere que el número de clústeres previamente. Crea una jerarquía permitiendo exploración a diferentes niveles de granularidad.
<b>Forma de los clústeres</b>	Supone que los grupos son esféricos y tienen tamaños similares. Intenta minimizar la suma de cuadrados dentro del grupo.	Puede manejar grupos con formas arbitrarias, permitiendo mayor flexibilidad a la hora de capturar estructuras de grupos complejas.
<b>Interpretabilidad</b>	Los grupos resultantes son fáciles de interpretar y analizar.	Proporciona una representación visual de las relaciones de los grupos y permite la exploración a diferentes niveles de granularidad.
<b>Escalabilidad</b>	Computacionalmente eficiente y puede manejar grandes conjuntos de datos con una cantidad razonable de clústeres.	Puede ser costoso computacionalmente, especialmente para grandes conjuntos de datos. Los requerimientos de memoria aumentan a medida que crece el número de puntos.
<b>Sensibilización a la inicialización</b>	Es sensible a la ubicación inicial de los centroides. Diferentes inicializaciones pueden conducir a diferentes resultados o converger a soluciones no óptimas.	No depende de la inicialización aleatoria. Los resultados de la agrupación están determinados por las decisiones de fusión basadas en proximidad de las agrupaciones.

Tabla 1. Principales diferencias entre K-Means y clúster jerárquico. Fuente: elaboración propia.

### ¿Cuándo usar K-Means y cuándo agrupación jerárquica?

- K-Means funciona muy bien cuando la separación entre los grupos está bien

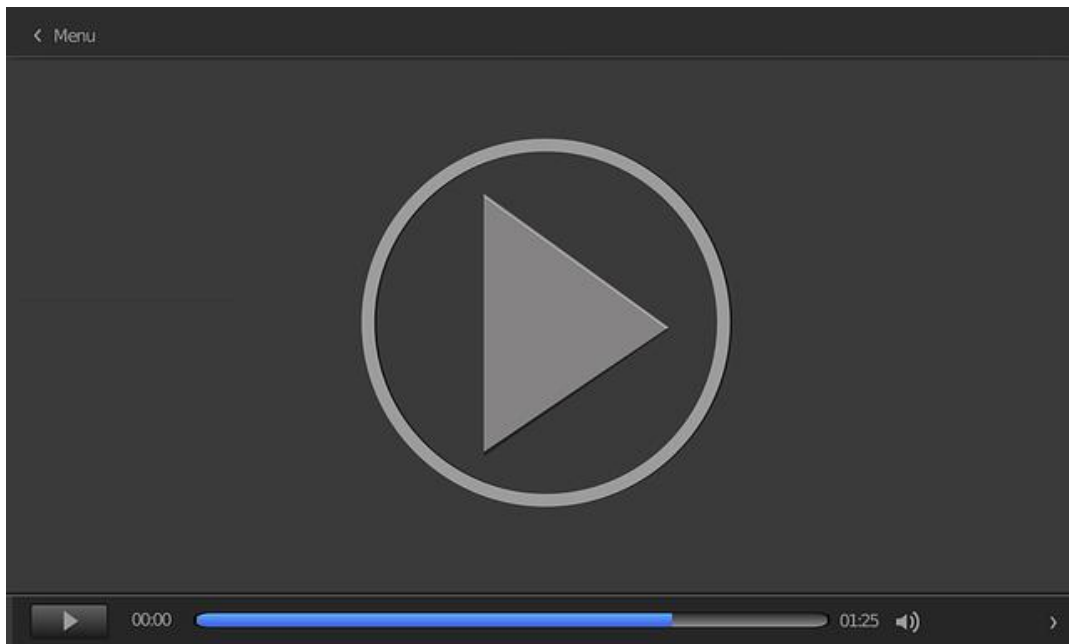
definida. Como el algoritmo busca minimizar la suma de cuadrados dentro del grupo; es muy eficaz identificando grupos compactos o esféricos.

- ▶ El clúster jerárquico permite explorar las relaciones dentro de los datos en diferentes niveles de granularidad.
- ▶ K-Means es computacionalmente más eficiente y puede manejar gran cantidad de datos. Es ideal para aplicaciones donde la eficiencia es importante.
- ▶ El clúster jerárquico es adecuado para conjuntos de datos pequeños o medianos.
- ▶ K-Means requiere el número de clústeres previamente, si se conoce este valor K-Means puede ser una opción adecuada.
- ▶ El clúster jerárquico no requiere que el número este predeterminado.
- ▶ K-Means funciona muy bien con datos numéricos, gracias al manejo de distancias entre puntos. Es adecuado para variables continuas o atributos que se puedan representar numéricamente.
- ▶ El clúster jerárquico es ideal para datos donde existe una estructura jerárquica natural, como taxonomías biológicas, jerarquías organizacionales o relaciones genealógicas.
- ▶ K-Means ayuda a crear prototipos y a realizar análisis exploratorio de datos. Proporciona información inicial sobre la estructura de los datos y ayuda a formular una hipótesis y realizar observaciones iniciales.
- ▶ El clúster jerárquico es visualmente atractivo e informativo. Es útil cuando se desea explorar y visualizar cómo se agrupan los datos en diferentes niveles de similitud, lo que puede proporcionar una visión inicial valiosa antes de aplicar otros métodos de *clustering*.
- ▶ Si tiene una idea de donde se pueden inicializar los clústeres, K-Means puede converger más rápido y producir resultados más precisos.



- ▶ En el clúster jerárquico, a diferencia de K-Means, no es necesario especificar el número de clústeres al principio. El dendrograma permite visualizar y decidir el número óptimo de clústeres.

A continuación, veremos el vídeo ***Comparación entre K-Means y clustering jerárquico***.



Accede al vídeo:

<https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=b63b42fc-95ba-4426-9da1-b1be014b0f73>

## 4.5. Cuaderno de ejercicios

### Ejercicio 1

#### Comparación de métodos

Describe las principales diferencias entre el *clustering* jerárquico aglomerativo y divisivo. ¿En qué situaciones crees que es uno mejor que el otro?

#### Solución

El método aglomerativo comienza con cada punto como un clúster individual y los fusiona hasta formar un único clúster, mientras que el divisivo comienza con todos los puntos en un solo clúster y los divide recursivamente. Se puede discutir la eficiencia computacional y la facilidad de interpretación en diferentes contextos de datos.

### Ejercicio 2

#### Interpretación de dendrogramas

Dibuja un dendrograma con base en los siguientes puntos dados {2,3,5,8,10,12}. Describe el proceso de agrupamiento y determina el número óptimo de clústeres a partir del dendrograma.

#### Solución

Mostrar un dendrograma donde se fusionan los puntos más cercanos primero (2 y 3, luego 5, y así sucesivamente) y explicar cómo el corte del dendrograma puede determinar el número de clústeres óptimos.

### Ejercicio 3

#### Aplicación práctica

Implementa un algoritmo de *clustering* jerárquico aglomerativo en Python utilizando un conjunto de datos de tu elección. Describe cada paso del proceso, desde la importación hasta la visualización del dendrograma.

## Solución

Un ejemplo sencillo para este punto podría ser el siguiente:

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets

# Import iris data
iris = datasets.load_iris()

iris_data = pd.DataFrame(iris.data)
iris_data.columns = iris.feature_names
iris_data['flower_type'] = iris.target
iris_data.head()

import scipy.cluster.hierarchy as sc
# Plot dendrogram
plt.figure(figsize=(20, 7))
plt.title("Dendrograms")

# Create dendrogram
sc.dendrogram(sc.linkage(iris_X, method='ward'))

plt.title('Dendrogram')
plt.xlabel('Sample index')
plt.ylabel('Euclidean distance')
```

## Ejercicio 4

### Evaluación de similitud

Explica cómo se puede definir la similitud inter-clúster utilizando métodos *single linkage*, *complete linkage* y *average linkage* y proporciona un ejemplo.

## Solución

## **Single linkage**

Definición: la distancia entre dos clústeres se define como la distancia entre los dos puntos más cercanos, uno de cada clúster.

Ejemplo:

- ▶ Clúster A: puntos (1, 2) y (2, 3).
- ▶ Clúster B: puntos (4, 5) y (6, 7).

Cálculo de distancias entre todos los puntos:

- ▶ Distancia entre (1, 2) y (4, 5):  $\sqrt{(4-1)^2 + (5-2)^2} = \sqrt{18} = 4.24$ .
- ▶ Distancia entre (1, 2) y (6, 7):  $\sqrt{(6-1)^2 + (7-2)^2} = \sqrt{50} = 7.07$ .
- ▶ Distancia entre (2, 3) y (4, 5):  $\sqrt{(4-2)^2 + (5-3)^2} = \sqrt{8} = 2.83$ .
- ▶ Distancia entre (2, 3) y (6, 7):  $\sqrt{(6-2)^2 + (7-3)^2} = \sqrt{32} = 5.66$ .

*Single linkage*: la distancia mínima es 2.83.

## **Complete linkage**

Definición: la distancia entre dos clústeres se define como la distancia entre los dos puntos más alejados, uno de cada clúster.

Ejemplo:

- ▶ Clúster A: puntos (1, 2) y (2, 3).
- ▶ Clúster B: puntos (4, 5) y (6, 7).

Cálculo de distancias entre todos los puntos:

- ▶ Distancia entre (1, 2) y (4, 5):  $\sqrt{(4-1)^2 + (5-2)^2} = \sqrt{18} = 4.24$ .

- ▶ Distancia entre (1, 2) y (6, 7):  $\sqrt{(6-1)^2 + (7-2)^2} = \sqrt{50} = 7.07$ .
- ▶ Distancia entre (2, 3) y (4, 5):  $\sqrt{(4-2)^2 + (5-3)^2} = \sqrt{8} = 2.83$ .
- ▶ Distancia entre (2, 3) y (6, 7):  $\sqrt{(6-2)^2 + (7-3)^2} = \sqrt{32} = 5.66$ .

*Complete linkage*: la distancia máxima es 7.07.

### **Average linkage**

Definición: la distancia entre dos clústeres se define como el promedio de todas las distancias entre pares de puntos, uno de cada clúster.

Ejemplo:

- ▶ Clúster A: puntos (1, 2) y (2, 3).
- ▶ Clúster B: puntos (4, 5) y (6, 7).

Cálculo de distancias entre todos los puntos:

- ▶ Distancia entre (1, 2) y (4, 5):  $\sqrt{(4-1)^2 + (5-2)^2} = \sqrt{18} = 4.24$ .
- ▶ Distancia entre (1, 2) y (6, 7):  $\sqrt{(6-1)^2 + (7-2)^2} = \sqrt{50} = 7.07$ .
- ▶ Distancia entre (2, 3) y (4, 5):  $\sqrt{(4-2)^2 + (5-3)^2} = \sqrt{8} = 2.83$ .
- ▶ Distancia entre (2, 3) y (6, 7):  $\sqrt{(6-2)^2 + (7-3)^2} = \sqrt{32} = 5.66$ .

*Average linkage*: el promedio de estas distancias es  $(4.24 + 7.07 + 2.83 + 5.66) / 4 = 4.95$ .

## 4.6. Referencias bibliográficas

Dissanayake, K. (2024, junio 6). *Machine Learning Algorithms(14)— K-Means Clustering and Hierarchical Clustering*. Medium. <https://towardsdev.com/machine-learning-algorithms-14-k-means-clustering-and-hierarchical-clustering-46acc005057d>

Prakash, C. (2023, junio 25). *Everything to know about Hierarchical Clustering: Agglomerative Clustering & Divisive Clustering*. Medium. <https://medium.com/@chandu.bathula16/everything-to-know-about-hierarchical-clustering-agglomerative-clustering-divisive-clustering-badf31ae047>

Van Der Post, H. y Smith, M. (2023). *Unsupervised Machine Learning: with Python*. Reactive Publishing.

### Una introducción al clustering jerárquico en Python

Keita, Z. (2023, enero). *An introduction to Hierarchical clustering in Python*. Datacamp. <https://www.datacamp.com/es/tutorial/introduction-hierarchical-clustering-python>

Este artículo proporciona los detalles de la agrupación jerárquica y su implementación en Python.

### ¿Cómo se analiza el clúster jerárquico en la herramienta estadística SPSS?

Elías Alvarado. (2023, diciembre 31). *Análisis clúster jerárquico en SPSS* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=jUAVEHXdjUI&t=4s>

Elías a través de un ejemplo práctico enseña cómo utilizar el SPSS para crear y analizar clústeres jerárquicos.



1. ¿Qué método de *clustering* jerárquico comienza con todos los puntos en un solo clúster y los divide recursivamente en clústeres más pequeños?
  - A. *Clustering* aglomerativo.
  - B. *Clustering* divisivo.
  - C. K-Means.
  - D. DBSCAN.
  
2. ¿Cuál de las siguientes técnicas de enlace calcula la distancia entre dos clústeres basándose en la distancia mínima entre puntos?
  - A. *Complete linkage*.
  - B. *Average linkage*.
  - C. *Single linkage*.
  - D. *Centroid linkage*.
  
3. ¿Qué representa un dendrograma en el contexto del *clustering* jerárquico?
  - A. Un gráfico de dispersión.
  - B. Una jerarquía de clústeres.
  - C. Un histograma de frecuencias.
  - D. Un gráfico de líneas.
  
4. ¿Cuál es la principal diferencia entre el *clustering* jerárquico aglomerativo y el divisivo?
  - A. El aglomerativo es más rápido que el divisivo.
  - B. El aglomerativo requiere el número de clústeres previamente establecidos.
  - C. El divisivo usa técnicas de enlace, mientras que el aglomerativo no.
  - D. El aglomerativo fusiona los clústeres, mientras que el divisivo los divide.

5. ¿Cuál de las siguientes técnicas de enlace calcula la distancia entre dos clústeres basándose en la distancia máxima de puntos?
  - A. *Complete linkage*.
  - B. *Average linkage*.
  - C. *Single linkage*.
  - D. Método Ward.
  
6. ¿Qué tipos de datos son más adecuados para el *clustering* jerárquico?
  - A. Datos con forma esférica.
  - B. Datos categóricos.
  - C. Datos con estructuras complejas y arbitrarias.
  - D. Datos que requieren particionamiento rápido.
  
7. ¿Cuál es una desventaja del *clustering* jerárquico comparado con K-Means?
  - A. No puede manejar datos categóricos.
  - B. Es computacionalmente más costoso.
  - C. Requiere el número de clústeres previamente.
  - D. No proporciona visualización de resultados.
  
8. En el contexto del *clustering* jerárquico, ¿qué método de enlace se basa en la minimización de la varianza?
  - A. *Single linkage*.
  - B. *Complete linkage*.
  - C. *Average linkage*.
  - D. Método Ward.

9. ¿Cuál es el principal beneficio de usar el *clustering* jerárquico sobre K-Means cuando no se conoce el número de clústeres?

- A. Es más rápido.
- B. Ofrece una jerarquía de clústeres.
- C. Es menos sensible a la inicialización.
- D. Maneja mejor los datos categóricos.

10. En el *clustering* jerárquico, ¿qué se utiliza para representar las relaciones de inclusión entre diferentes grupos?

- A. Gráfico de barras.
- B. Diagrama de cajas.
- C. Dendrograma.
- D. Gráfico de líneas.