

# MODULO III - MODELOS

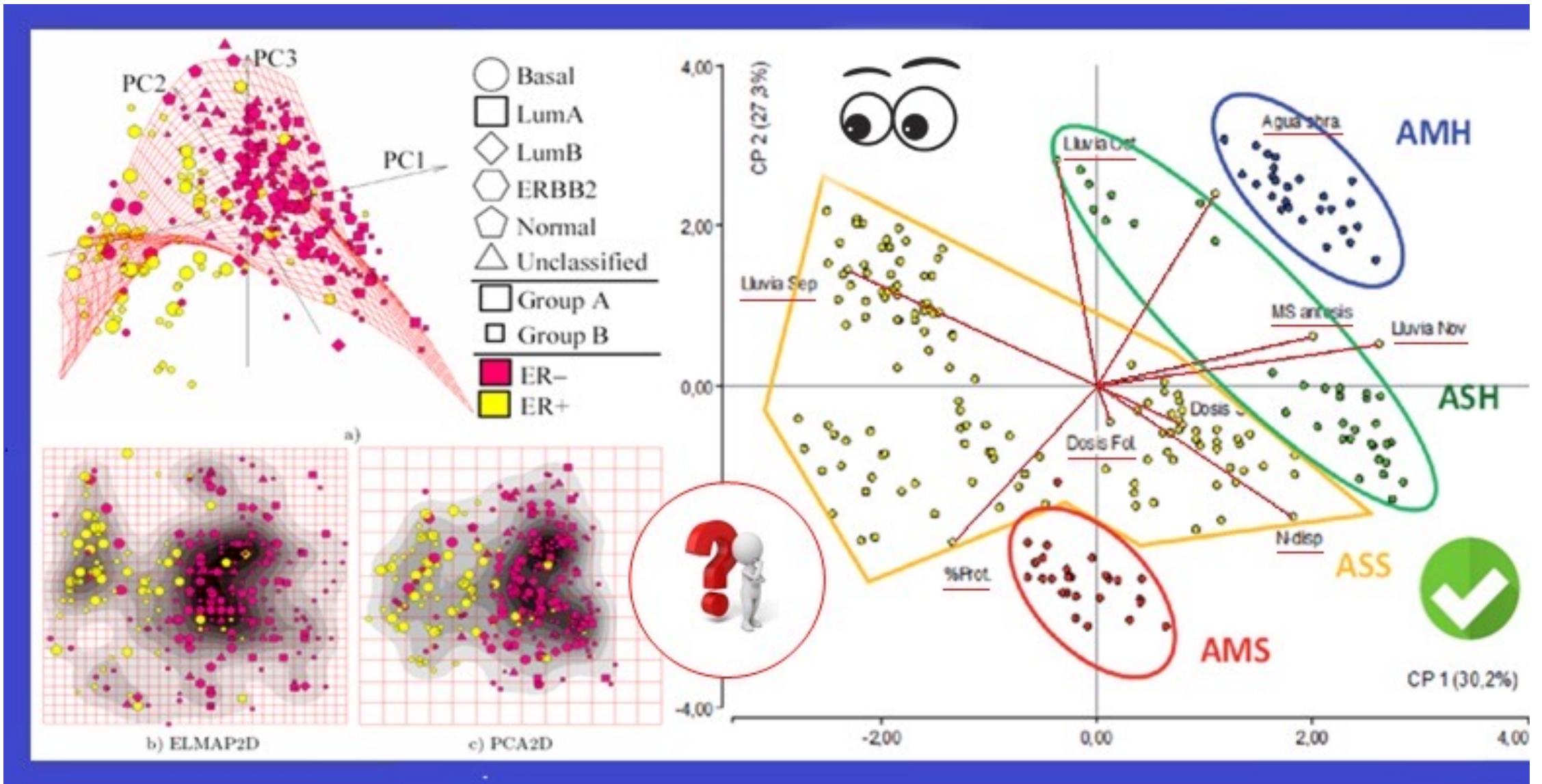
# DE APRENDIZAJE NO

# SUPERVISADO

DATA SCIENCE CON PYTHON

# Análisis de Componentes Principales (PCA)

- El Análisis de Componentes Principales (PCA) es **una técnica de reducción de dimensionalidad** que se utiliza para transformar un conjunto de datos de alta dimensión en un conjunto de datos de baja dimensión, manteniendo la mayor parte de la información original
- El PCA se basa en la idea de que **los datos pueden ser representados como una combinación lineal de un conjunto de vectores ortogonales**, llamados componentes principales.
- Los componentes principales **se ordenan de acuerdo a su varianza**, de modo que el primer componente principal captura la mayor cantidad de varianza en los datos, seguido del segundo componente principal, y así sucesivamente.



# Análisis de Componentes Principales (PCA)

- Se puede realizar de la siguiente manera:
  1. Se calcula la matriz de covarianza de los datos.
  2. Se calculan los autovalores y autovectores de la matriz de covarianza.
  3. Los autovectores se utilizan para transformar los datos a un nuevo conjunto de datos de baja dimensión.

Ejemplo:

```
import numpy as np
from sklearn.decomposition import PCA
# Generamos un conjunto de datos de alta dimensión
X = np.random.rand(100, 100)
# Realizamos el PCA
pca = PCA(n_components=5)
X_pca = pca.fit_transform(X)
# Visualizamos los componentes principales
plt.scatter(X_pca[:, 0], X_pca[:, 1])
```

# Análisis de Componentes Principales (PCA)

- Se utiliza en una amplia gama de aplicaciones, incluyendo:
  - **Reducción de dimensionalidad:** se puede utilizar para reducir la dimensionalidad de un conjunto de datos, lo que puede facilitar el análisis y la visualización de los datos.
  - **Inferencia:** se puede utilizar para realizar inferencias sobre los datos. Por ejemplo, se puede utilizar para estimar la distribución subyacente de los datos.
  - **Clasificación:** se puede utilizar para mejorar el rendimiento de los algoritmos de clasificación.

# Análisis de Componentes Principales (PCA)

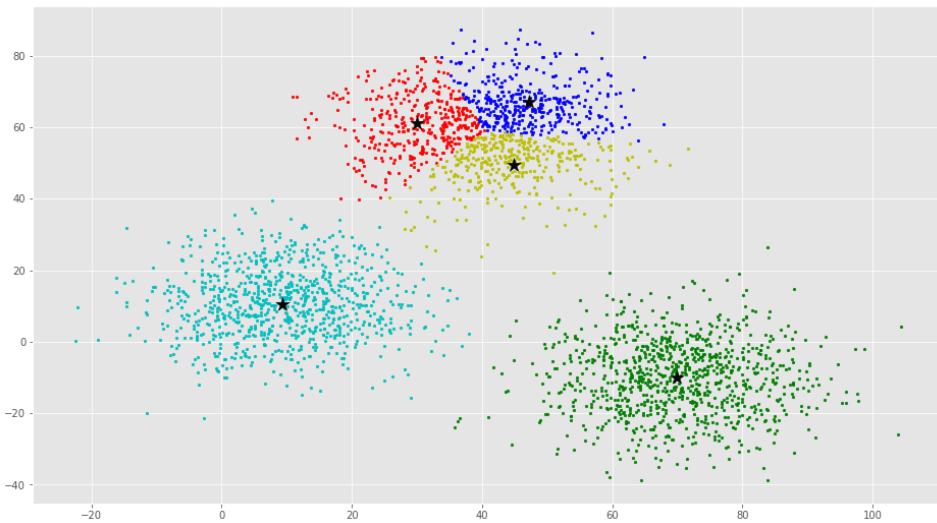
- Cuando aplicar PCA:
  - Cuando se trabaja con un **conjunto de datos de alta dimensión**. El PCA puede ser una herramienta útil para reducir la dimensionalidad de un conjunto de datos de alta dimensión, lo que puede facilitar el análisis y la visualización de los datos.
  - Cuando **se desea realizar inferencias sobre los datos**. El PCA se puede utilizar para realizar inferencias sobre los datos, como estimar la distribución subyacente de los datos.
  - Cuando **se desea mejorar el rendimiento de los algoritmos de clasificación**. El PCA se puede utilizar para mejorar el rendimiento de los algoritmos de clasificación, como los árboles de decisión y los modelos de aprendizaje automático.

# Análisis de Componentes Principales (PCA)

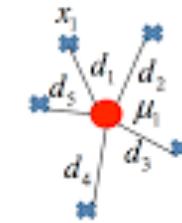
- Cuando no aplicar PCA:
  - Cuando **los datos no son linealmente correlacionados**. El PCA asume que los datos son linealmente correlacionados. Si los datos no son linealmente correlacionados, el PCA puede no ser efectivo.
  - Cuando **se desea preservar la información original**. El PCA puede perder información original al reducir la dimensionalidad de los datos. Si se desea preservar la información original, el PCA puede no ser la mejor opción.
  - Cuando **se desea identificar outliers**. El PCA puede ser sensible a outliers. Si se desea identificar outliers, el PCA puede no ser la mejor opción.

# Clustering basado en distancia (K-medias)

- El clustering es una técnica de aprendizaje no supervisado que se utiliza para agrupar datos similares. El clustering basado en distancia es un tipo de clustering que utiliza la distancia entre los datos para asignarlos a clusters.
- K-medias es un algoritmo de clustering basado en distancia que asigna cada punto de datos a uno de los K clusters. El algoritmo funciona de la siguiente manera:
  1. Se seleccionan aleatoriamente K puntos de datos como centros de los clusters.
  2. Cada punto de datos se asigna al cluster más cercano al centro de ese cluster.
  3. Se calculan los nuevos centros de los clusters.
  4. Se repiten los pasos 2 y 3 hasta que los centros de los clusters no cambien.

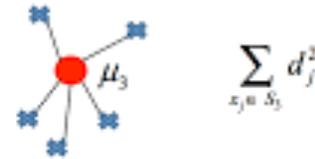


● Centroid  
❀ Sample



$$\sum_{x_j \in S_i} d_j^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2$$

$$\sum_{x_j \in S_1} d_j^2$$



$$\sum_{x_j \in S_3} d_j^2$$

$$\min_S E(\mu_i) = \sum_{x_j \in S_i} d_j^2 + \sum_{x_j \in S_1} d_j^2 + \sum_{x_j \in S_3} d_j^2$$

```
import numpy as np
from sklearn.cluster import KMeans

# Generamos un conjunto de datos
X = np.random.rand(100, 2)

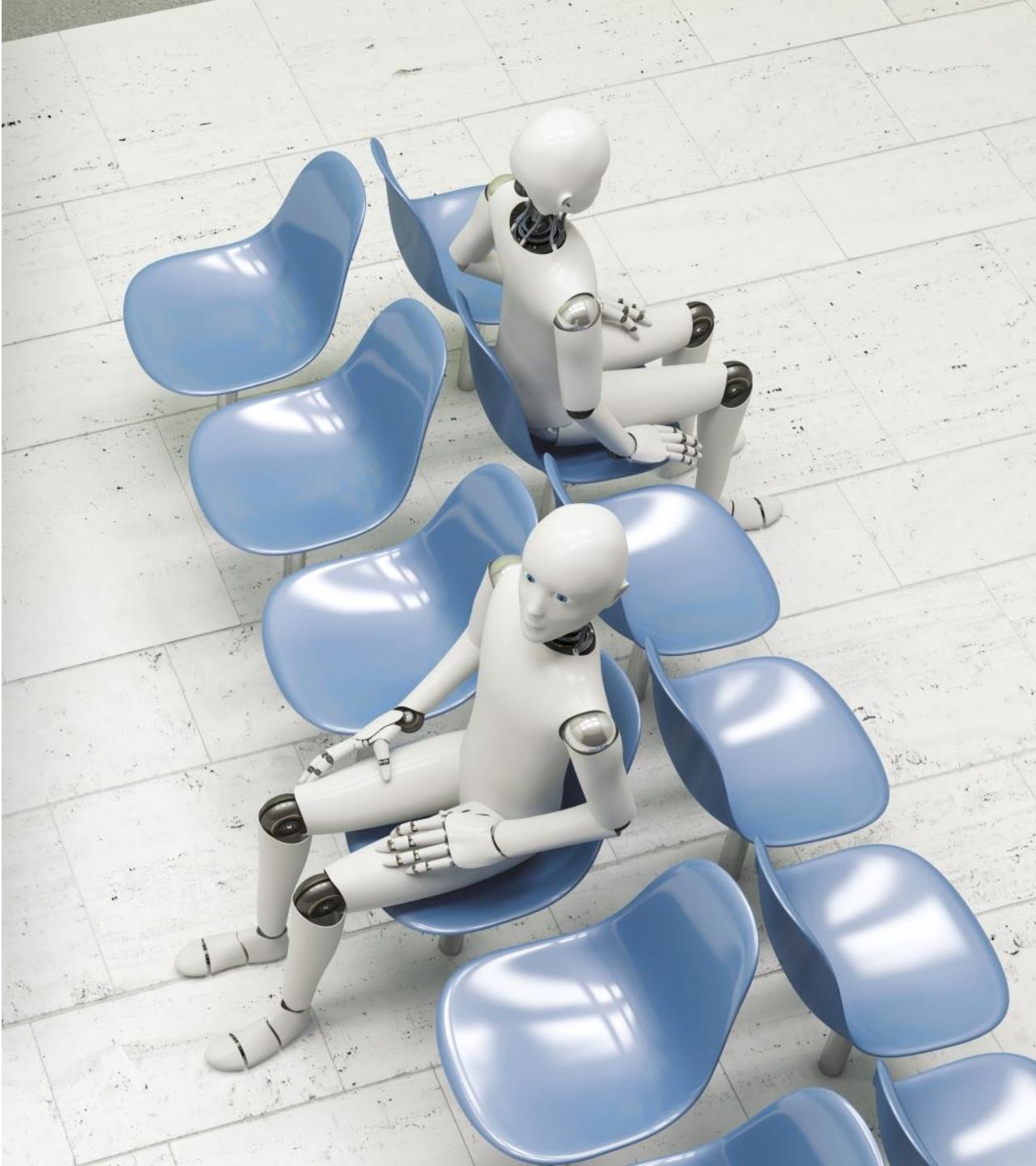
# Realizamos el clustering K-medias
kmeans = KMeans(n_clusters=3)
y_pred = kmeans.fit_predict(X)

# Visualizamos los clusters
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
```

Clustering basado en distancia (K-medias)

# Clustering basado en distancia (K-medias)

- Se utiliza en una amplia gama de aplicaciones, incluyendo:
  - Segmentación de clientes: El clustering K-medias se puede utilizar para segmentar clientes en grupos homogéneos.
  - Clasificación: El clustering K-medias se puede utilizar como preprocesamiento para mejorar el rendimiento de los algoritmos de clasificación.
  - Visualización: El clustering K-medias se puede utilizar para visualizar datos de alta dimensión.



# Clustering basado en distancia K-medias

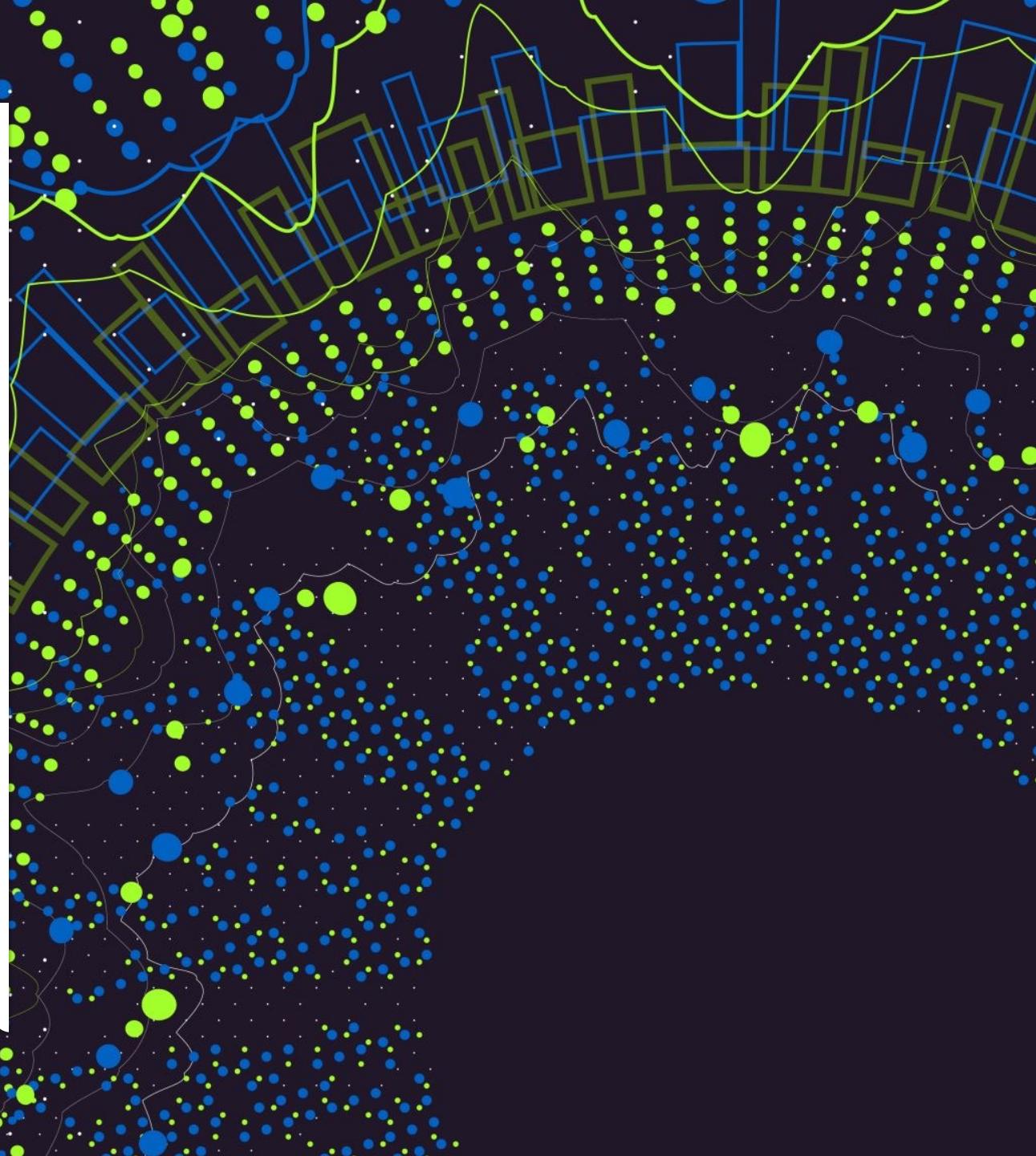
- Aplicar K-medias:
  - Cuando se desea agrupar datos similares. El K-medias es un algoritmo de clustering basado en distancia que se basa en la idea de que los puntos de datos que están más cerca entre sí pertenecen al mismo grupo.
  - Cuando se trabaja con un conjunto de datos de baja dimensión. El K-medias puede ser una herramienta útil para agrupar datos de baja dimensión, ya que no requiere mucha memoria ni tiempo de cálculo.
  - Cuando se desea una solución rápida. El K-medias es un algoritmo relativamente rápido, lo que lo hace adecuado para aplicaciones que requieren una solución rápida.

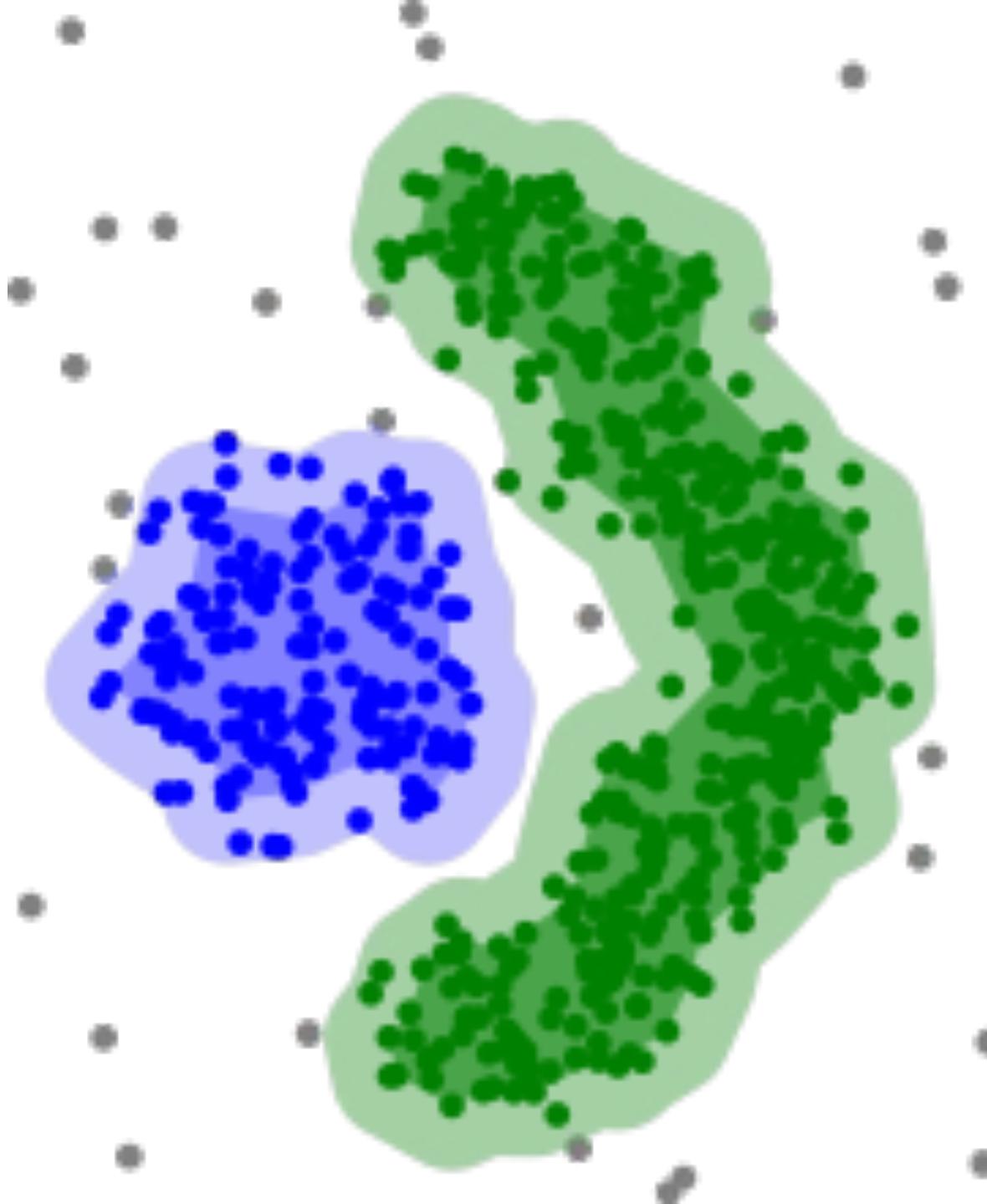
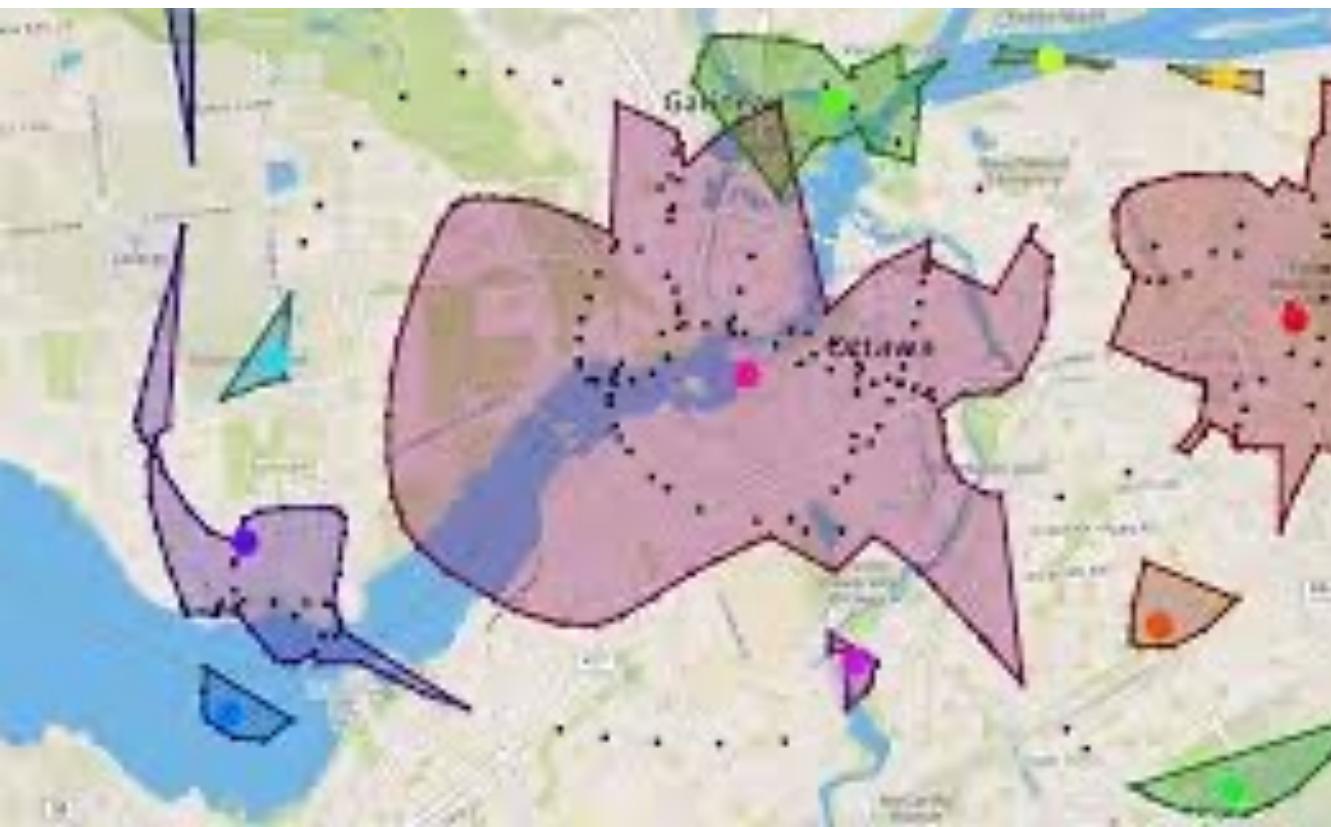
# Clustering basado en distancia (K-medias)

- No aplicar K-medias:
  - Cuando los datos no son linealmente separables. El K-medias asume que los datos son linealmente separables, lo que significa que se pueden dividir en grupos que están separados por un plano. Si los datos no son linealmente separables, el K-medias puede no ser capaz de agruparlos correctamente.
  - Cuando se desea preservar la información original. El K-medias puede perder información original al agrupar los datos. Si se desea preservar la información original, el K-medias puede no ser la mejor opción.
  - Cuando se desea identificar outliers. El K-medias puede ser sensible a outliers, lo que significa que puede agrupar los outliers con los datos normales. Si se desea identificar outliers, el K-medias puede no ser la mejor opción.

# Clusterización basado en densidad (DBSCAN)

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) es un algoritmo de clustering basado en densidad que funciona de la siguiente manera:
  1. Se define un radio de vecindad para cada punto de datos.
  2. Los puntos de datos que están dentro del radio de vecindad de otro punto de datos se consideran vecinos.
  3. Los puntos de datos que tienen un número mínimo de vecinos se consideran puntos de núcleo.
  4. Los puntos de datos que están conectados a un punto de núcleo se consideran puntos de borde.
  5. Los puntos de datos que no están conectados a ningún punto de núcleo se consideran ruido.





# Clusterización basado en densidad (DBSCAN)

```
import numpy as np
from sklearn.cluster import DBSCAN

# Generamos un conjunto de datos
X = np.random.rand(100, 2)

# Realizamos el clustering DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=10)
y_pred = dbscan.fit_predict(X)

# Visualizamos los clusters
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
```

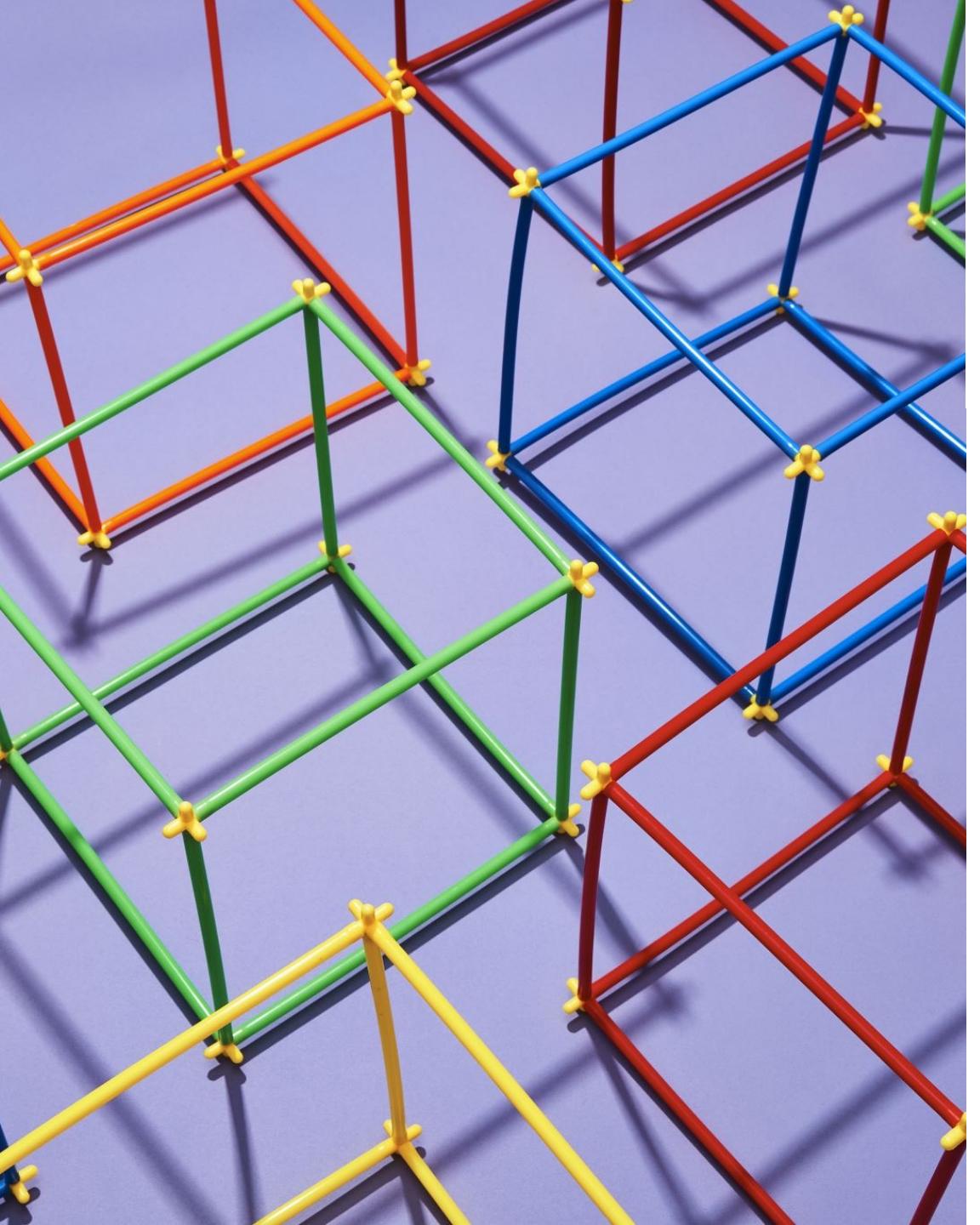
# Clusterización basado en densidad (DBSCAN)

- El clustering DBSCAN se utiliza en una amplia gama de aplicaciones, incluyendo:
  - Segmentación de clientes: El clustering DBSCAN se puede utilizar para segmentar clientes en grupos homogéneos.
  - Clasificación: El clustering DBSCAN se puede utilizar como preprocesamiento para mejorar el rendimiento de los algoritmos de clasificación.
  - Visualización: El clustering DBSCAN se puede utilizar para visualizar datos de alta dimensión.
- ¿Cuál es la diferencia entre el clustering K-medias y el clustering DBSCAN?
  - El clustering K-medias se basa en la distancia entre los puntos de datos para asignarlos a clusters. El clustering DBSCAN se basa en la densidad de los datos para asignarlos a clusters.
- ¿En qué casos es mejor utilizar el clustering DBSCAN?
  - El clustering DBSCAN es una buena opción cuando los datos no son linealmente separables o cuando se desea identificar outliers.

# Clusterización basado en densidad (DBSCAN)

- Aplicar DBSCAN:
  - Cuando los datos no son linealmente separables. El DBSCAN es una buena opción cuando los datos no se pueden dividir en grupos que están separados por un plano.
  - Cuando se desea identificar outliers. El DBSCAN puede identificar outliers, que son puntos de datos que se encuentran lejos de la mayoría de los demás datos.
  - Cuando se desea agrupar datos de alta dimensión. El DBSCAN es una buena opción para agrupar datos de alta dimensión, ya que no requiere mucha memoria ni tiempo de cálculo.





# Clusterización basado en densidad (DBSCAN)

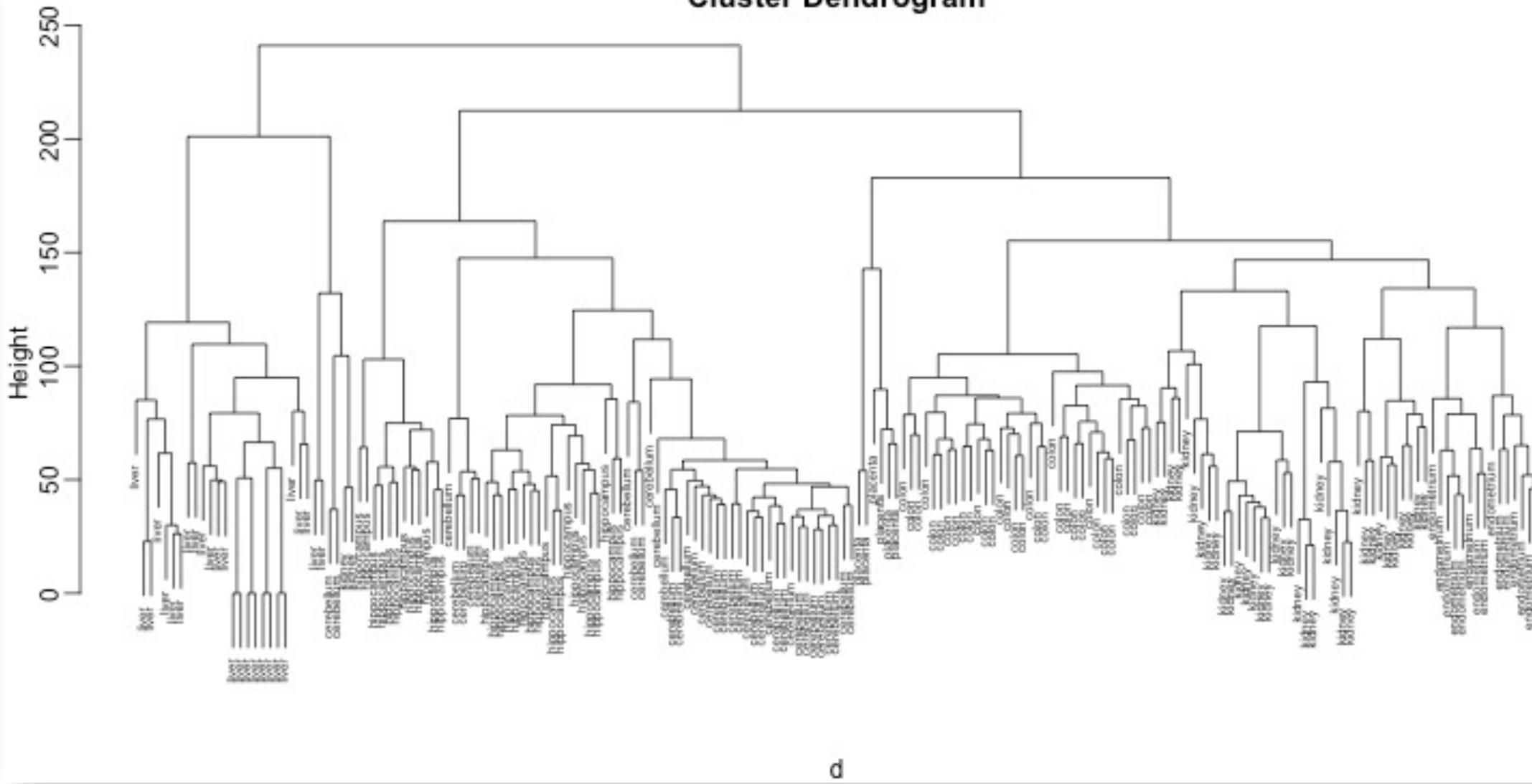
- No aplicar DBSCAN:
  - Cuando los datos son linealmente separables. Si los datos son linealmente separables, el K-medias puede ser una mejor opción que el DBSCAN.
  - Cuando se desea preservar la información original. El DBSCAN puede perder información original al agrupar los datos.
  - Cuando se desea agrupar datos con formas irregulares. El DBSCAN puede tener dificultades para agrupar datos con formas irregulares.



# Clustering jerárquico

- El clustering jerárquico es una técnica de aprendizaje no supervisado que se utiliza para agrupar datos similares. El clustering jerárquico construye una jerarquía de clusters, en la que cada cluster es un subconjunto de un cluster más grande.
- Hay dos tipos principales de clustering jerárquico:
  - Clustering aglomerativo: En el clustering aglomerativo, se comienzan con cada punto de datos como un cluster individual. Luego, los clusters se van fusionando gradualmente hasta que se obtiene un solo cluster.
  - Clustering divisive: En el clustering divisive, se comienza con un solo cluster que contiene todos los puntos de datos. Luego, los clusters se van dividiendo gradualmente hasta que se obtienen clusters individuales.

## Cluster Dendrogram



# Clustering jerárquico

```
import numpy as np
from sklearn.cluster import AgglomerativeClustering

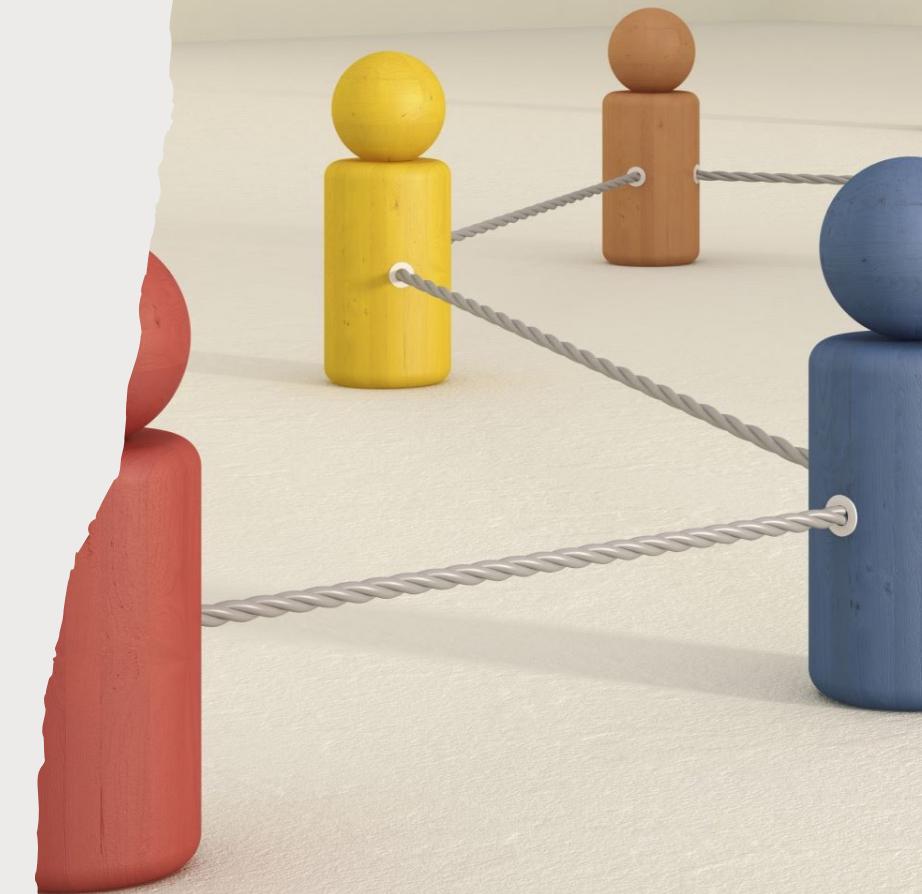
# Generamos un conjunto de datos
X = np.random.rand(100, 2)

# Realizamos el clustering jerárquico
agg = AgglomerativeClustering(n_clusters=3)
y_pred = agg.fit_predict(X)

# Visualizamos los clusters
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
```

# Clustering jerárquico

- El clustering jerárquico se utiliza en una amplia gama de aplicaciones, incluyendo:
  - Segmentación de clientes: El clustering jerárquico se puede utilizar para segmentar clientes en grupos homogéneos.
  - Clasificación: El clustering jerárquico se puede utilizar como preprocesamiento para mejorar el rendimiento de los algoritmos de clasificación.
  - Visualización: El clustering jerárquico se puede utilizar para visualizar datos de alta dimensión.

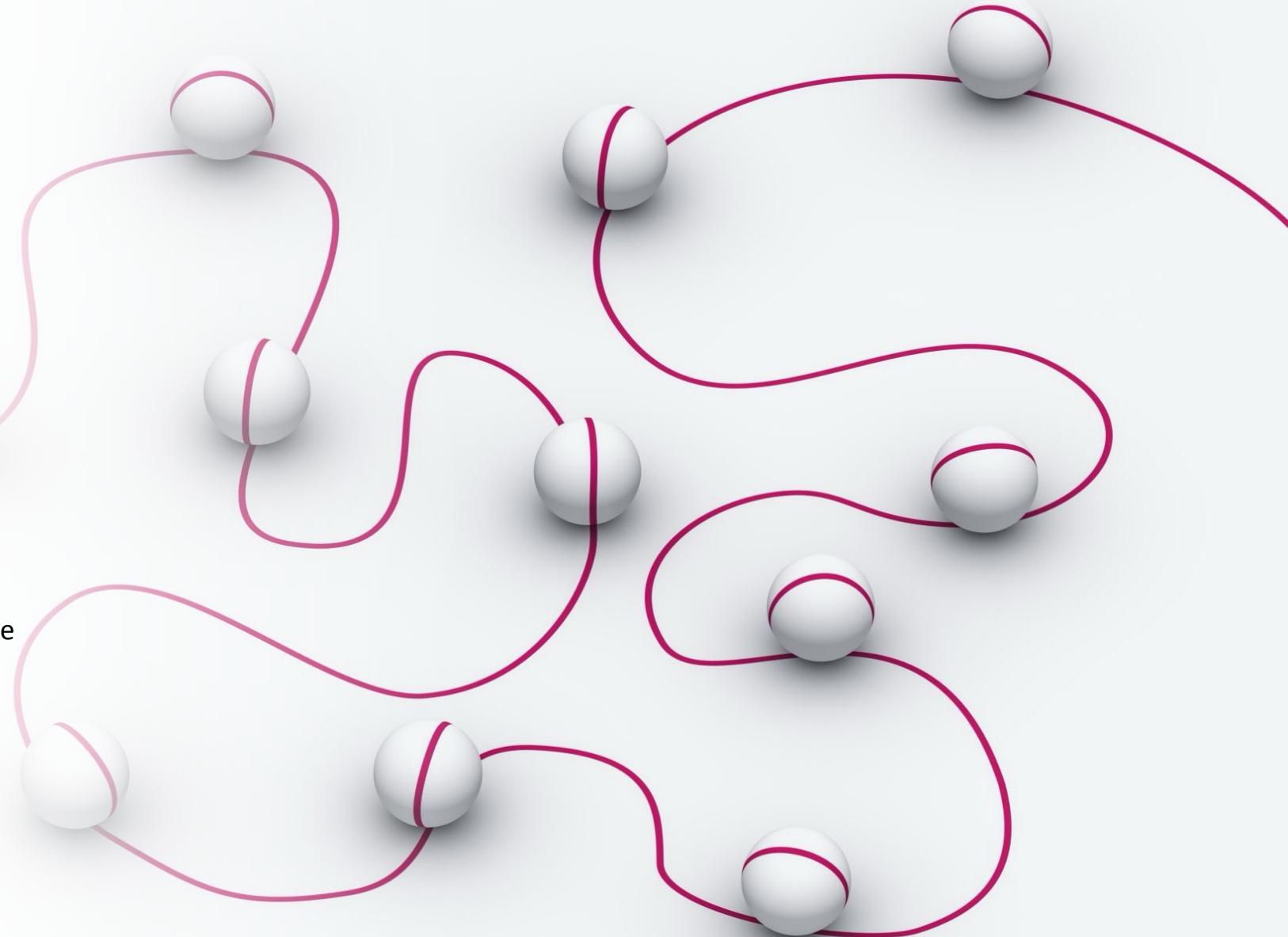


# Clustering jerárquico

- Cuál es la diferencia entre el clustering jerárquico y el clustering K-medias?
  - El clustering jerárquico construye una jerarquía de clusters, mientras que el clustering K-medias asigna cada punto de datos a uno de los K clusters.
- ¿En qué casos es mejor utilizar el clustering jerárquico?
  - El clustering jerárquico es una buena opción cuando se desea visualizar la jerarquía de clusters. También puede ser una buena opción cuando se desea identificar outliers.

# Clustering jerárquico

- Aplicar clustering jerárquico:
  - Cuando se desea visualizar la jerarquía de clusters. El clustering jerárquico proporciona una representación visual de la relación entre los clusters.
  - Cuando se desea identificar outliers. El clustering jerárquico puede identificar outliers, que son puntos de datos que se encuentran lejos de la mayoría de los demás datos.
  - Cuando se desea realizar un análisis exploratorio de datos. El clustering jerárquico puede ayudar a identificar patrones y tendencias en los datos.



# Clustering jerárquico

- No aplicar clustering jerárquico:
  - Cuando se desea un número específico de clusters. El clustering jerárquico no asigna automáticamente un número específico de clusters.
  - Cuando se desea un algoritmo robusto a los outliers. El clustering jerárquico puede ser sensible a outliers.
  - Cuando se desea un algoritmo eficiente en términos de tiempo de cómputo. El clustering jerárquico puede ser lento para conjuntos de datos grandes.

