Aprendizaje de Máquina

No Supervisado

Sesión 4

¿Qué es el Agrupamiento Jerárquico?

Organización Jerárquica

El agrupamiento jerárquico organiza los datos en una estructura de árbol, permitiendo analizar las relaciones entre grupos a diferentes niveles. A diferencia de K-Means, no requiere definir el número de clusters de antemano.

Proceso Progresivo

Los clusters se forman de manera progresiva mediante fusión o división. Los resultados se representan con un dendrograma, mostrando la similitud entre los datos.

Dendrogramas

Un dendrograma es un diagrama en forma de árbol donde los datos más similares se agrupan en ramas cercanas y los más diferentes en ramas más separadas.

Tipos de Agrupamiento Jerárquico

1 Aglomerativo

El algoritmo aglomerativo, o Hierarchical Agglomerative Clustering (HAC), es el más utilizado y se basa en una estrategia "bottom-up". Comienza con cada punto como un cluster individual y fusiona los más similares en cada paso hasta que todos pertenecen a un solo cluster.

2 Divisivo

El algoritmo divisivo, o Divisive Hierarchical Clustering (DHC), es menos común y sigue una estrategia "top-down". Comienza con todos los datos en un solo cluster grande y lo divide en subgrupos basándose en diferencias entre los datos hasta que cada punto es su propio cluster.

Algoritmo Aglomerativo y Vivisivo

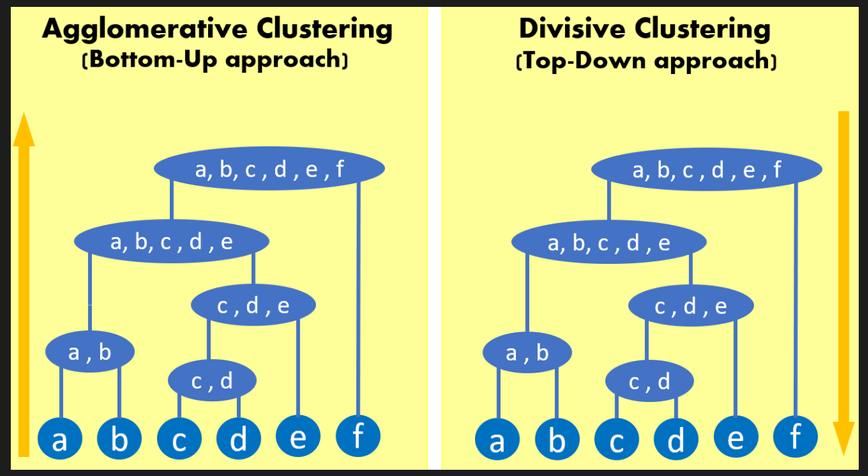


Imagen de Reddit: Algoritmo de Clustering Jerárquico

¿Cuál es mejor?

Depende del caso:

⇒**Aglomerativo** es más usado porque es más eficiente computacionalmente.

⇒**Divisivo** puede ser mejor cuando hay grandes diferencias en los datos y queremos dividirlos de forma clara.

Ambos producen un **dendrograma**, que permite elegir el número de clusters analizando la estructura del árbol.

Dendogramas: Visualización e Interpretación

Visualización de la Jerarquía

Un dendrograma es un diagrama en forma de árbol que muestra cómo se agrupan los datos en un proceso de agrupamiento jerárquico. Los datos más similares están conectados en ramas cercanas, mientras que los más distintos están en ramas separadas.

Determinación del Número Óptimo

Al cortar el dendrograma en un determinado nivel de altura, se pueden identificar los grupos más significativos. Se observa dónde hay grandes saltos en la distancia de fusión, indicando que unir más clusters haría los grupos menos homogéneos.

Agrupación Jerárquica: Etapas Clave

1. **Calcular la Matriz de Distancias**
   - Se mide la similitud entre los datos utilizando una métrica de distancia, como la distancia euclidiana o Manhattan.
2. **Aplicar un Método de Enlace**
   - Esto define cómo se combinan los clusters en cada iteración: enlace simple, completo, promedio o por centroide.
3. **Construcción del Dendrograma**
   - Se representa gráficamente la jerarquía de clusters para analizar cómo se agrupan los datos.
4. **Elección del Número de Clusters**
   - Se analiza el dendrograma y se elige un punto de corte para definir cuántos clusters habrá.
5. **Asignación final de clusters**
   - Se agrupan los datos en los clusters definidos.

Ventajas y Desventajas de esta técnica

Ventajas:

✓ No requiere especificar el número de clusters de antemano, a diferencia de K-Means.

✓ Proporciona una estructura jerárquica útil para interpretar relaciones entre datos.

✓ Puede manejar datos con formas de clusters no esféricos, a diferencia de K-Means.

Desventajas:

✗ Es computacionalmente costoso, especialmente con grandes volúmenes de datos.

✗ Sensibilidad a los valores atípicos, lo que puede afectar la formación de clusters.

✗ No permite re-asignación de puntos una vez que un dato ha sido asignado a un cluster.

Implementando el Agrupamiento Jerárquico

Objetivo: Implementar el agrupamiento jerárquico utilizando la librería scipy para generar un dendograma y skleam para aplicar el clustering.

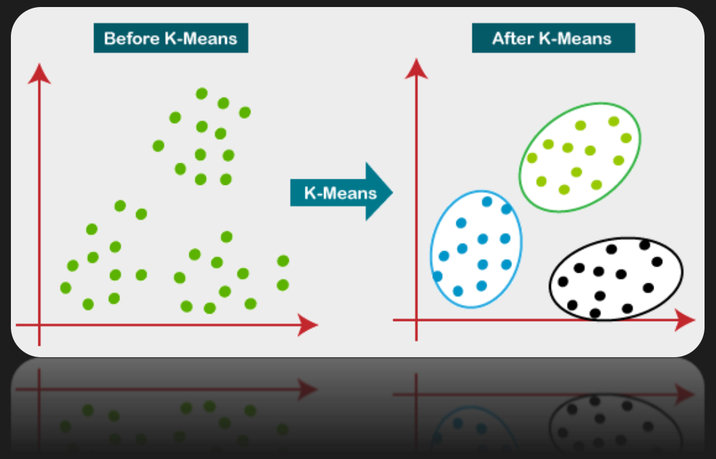
Requisitos:

1. **Obtener un conjunto de datos** (usaremos el dataset *Iris*).
2. **Calcular la matriz de distancias** y representar los clusters en un dendrograma.
3. **Elegir el número de clusters** óptimo con base en el dendrograma.
4. **Aplicar el algoritmo de clustering jerárquico** para asignar cada punto a un cluster.
5. **Visualizar los resultados**.

Utilizaremos el **conjunto de datos *Iris***, que es un dataset clásico de Machine Learning con características de flores. Usaremos solo dos columnas (sepal length y sepal width) para visualizar mejor los clusters.

El detalle de la actividad se encuentra en la guía de estudio de la sesión.

Algoritm K-Means: Fundamentos



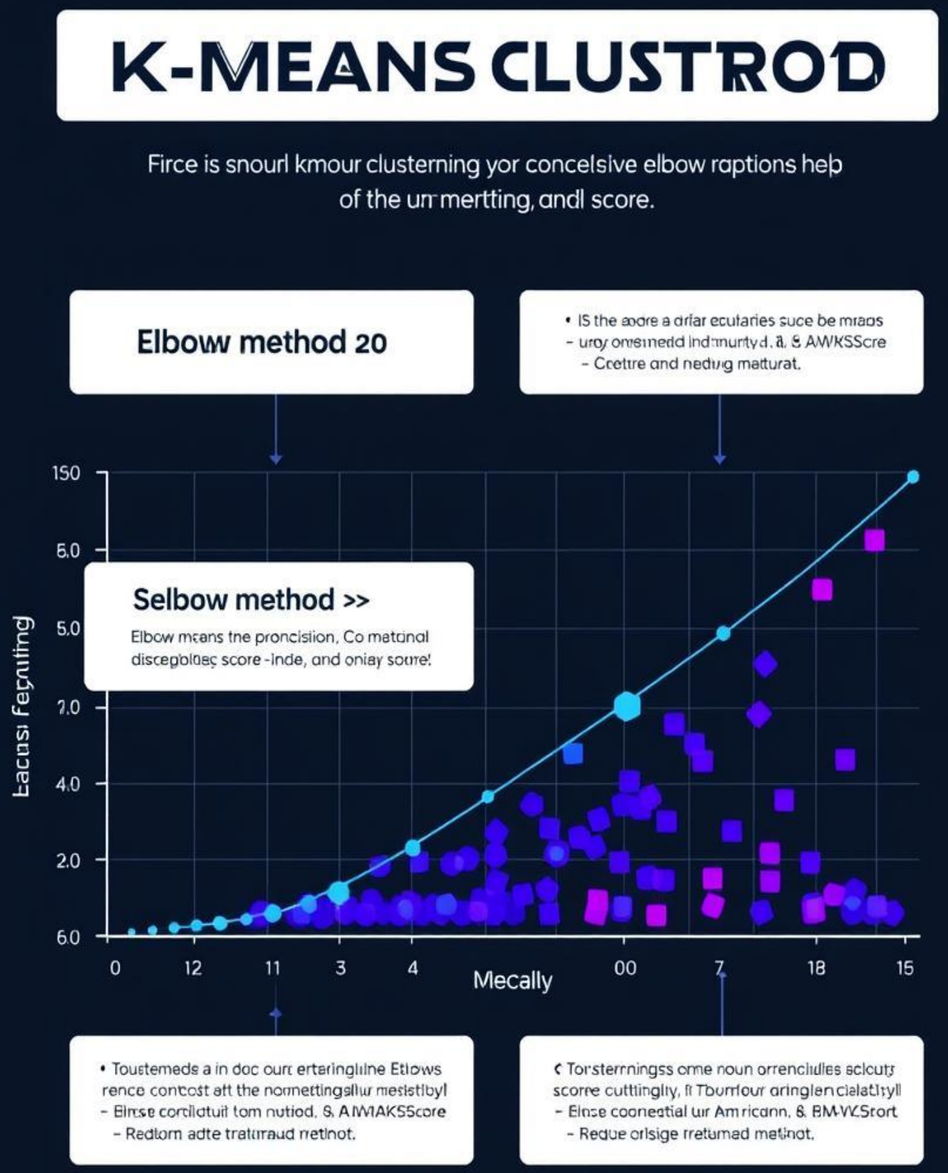
Ejemplo antes y después de K-Means. Lea Setruk

División en K Grupos

El algoritmo K-Means divide un conjunto de datos en K grupos (clusters), donde cada dato pertenece al cluster con el centroide más cercano. Es un algoritmo no supervisado que minimiza la distancia entre los puntos y el centroide del cluster al que pertenecen.

Pasos Fundamentales

El algoritmo K-Means sigue pasos como elegir el número de clusters, inicializar centroides aleatorios, asignar cada punto al cluster más cercano y recalcular los centroides hasta converger.



Elección del Valor de K en K-Means

Método del Codo
El método del codo calcula la suma de los errores cuadrados (WCSS) para diferentes valores de K. Se grafica el WCSS vs. K y se elige el "punto de inflexión" donde la disminución de WCSS se hace menos pronunciada.

Coeficiente de Silueta
El coeficiente de silueta mide qué tan bien separados están los clusters y qué tan coherentes son internamente. Se calcula el coeficiente de silueta promedio para diferentes valores de K y se elige el que maximiza esta métrica.

Ventajas y Desventajas de K-Means

Ventajas	Desventajas
•Eficiencia computacional. •Fácil de entender e implementar. •Escalabilidad. •Funciona bien en datos con clusters bien definidos.	•Sensibilidad al número de clusters. •No maneja bien clusters de diferentes formas y densidades. •Sensibilidad a valores atípicos (outliers). •No garantiza el óptimo global. •No detecta ruido o puntos sin cluster.

Implementando K-Means

Requisitos:

1. Cargaremos un conjunto de datos (Iris) y seleccionaremos solo dos características para poder visualizar los clusters.
2. Aplicaremos el **algoritmo K-Means** con un valor de K elegido previamente.
3. Graficaremos los clusters resultantes y sus centroides.

Usaremos el conjunto de datos **Iris**, disponible en la librería **sklearn.datasets**. Este dataset contiene información sobre tres tipos de flores **(Setosa, Versicolor, Virginica)** y sus características (largo del pétalo, ancho del pétalo, etc.).

El detalle de la actividad se encuentra en la guía de estudio de la sesión.

Algoritmo DBSCAN: Fundamentos

Clustering Basado en Densidad DBSCAN (Density-Based Spatial Clustering of Applications with Noise) es un algoritmo de clustering basado en densidad, ideal para identificar estructuras en conjuntos de datos con formas irregulares y para detectar outliers.	**Clasificación de Puntos** DBSCAN clasifica los puntos en centrales, frontera y ruido (outliers). No requiere especificar la cantidad de clusters como K-Means, ya que los clusters emergen de la estructura de los datos.

Ventajas y Desventajas de DBSCAN

1

Ventajas

- No requiere especificar el número de clusters.
- Identifica clusters de diferentes formas y densidades.
- Detecta puntos ruido y outliers automáticamente.
- Robusto ante valores atípicos.

2

Desventajas

- Sensibilidad a los parámetros ε y MinPts.
- Manejo ineficiente de datos con variaciones en densidad.
- Requiere cálculos de distancia entre todos los puntos.

Implementando DBSCAN

Requisitos:

1. Generaremos datos de prueba con **make_moons().**
2. Aplicaremos **DBSCAN** con diferentes valores de **eps**.
3. Compararemos los clusters resultantes con el algoritmo **K-Means**.

El detalle de la actividad se encuentra en la guía de estudio de la sesión.

Preguntas

Sección de preguntas

Aprendizaje de Máquina

No Supervisado

Continúe con las actividades