*Algoritmos de Agrupacion I*

***Recap | Clustering***

También conocidas como agrupamiento o segmentación tienen como principal función:

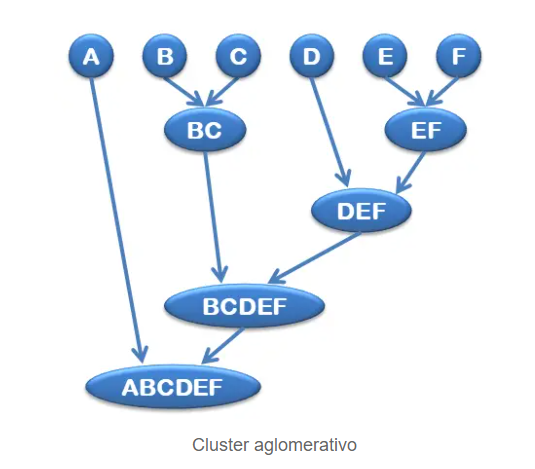
🎯 Encontrar una estructura o un patrón en una colección de datos no clasificados.

Intentan encontrar grupos en los datos que compartan atributos en común 🧐

***Cluster Jerárquicos - Tipos***

**Jerárquicos aglomerativos (bottom-up)**

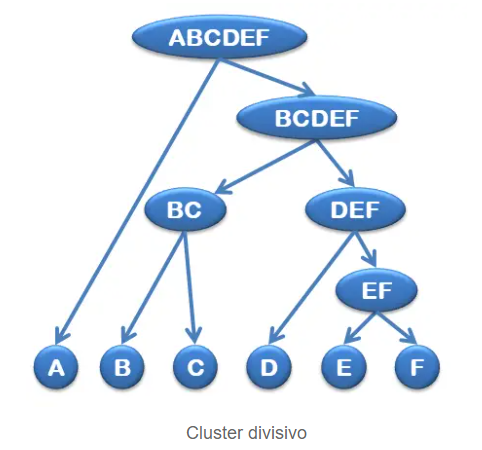
Inicialmente cada instancia es un clúster. Las estrategias aglomerativas, parten de un conjunto de elementos individuales y **van “juntando” los elementos que más se parezcan hasta quedarse con un número de clusters que se considere óptimo**.



***Tipos de Cluster Jerárquicos***

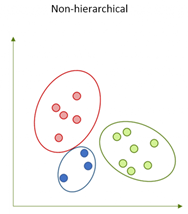
**Jerárquicos divisivos (top-down)**

Inicialmente todas las instancias están en un solo clúster y luego se van dividiendo, tal cual su nombre lo indica. Las estrategias divisivas, parten del conjunto de elementos completos y se van separando en grupos diferentes entre sí, hasta quedarse con un número de clusters que se considere óptimo.

****

***Clúster No Jerárquico***

Volviendo a este tipo de algoritmo, la cantidad de clústeres óptima se define de antemano, y los registros se asignan a los clústeres según su cercanía. Existen múltiples algoritmos de Tipo No Jerárquico, como ser por ejemplo: K – Means o HDBSCAN.



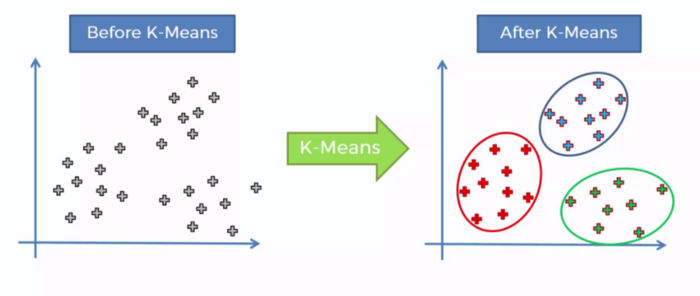
***Otros tipos de Clustering***

👉 También existen varios tipos más específicos de clústeres.

Por ejemplo overlapping o fuzzy, donde los registros pueden pertenecer a varios clusters simultáneamente; o los probabilísticos, que usan una distribución de probabilidad para asignar la membresía al clúster.

***K - Means***

El objetivo de este algoritmo es encontrar grupos en los datos, con el número de grupos representados por la variable K. El algoritmo funciona de manera iterativa para asignar cada punto de datos a uno de los grupos K en función de las características que se proporcionan.



Los puntos de datos se agrupan en función de la similitud de las características. Los resultados del algoritmo de agrupamiento K Means son:

* Los centroides de los clústeres K, que pueden ser usados para etiquetar nuevos datos.
* Etiquetas para los datos de formación, cada punto de datos se asigna a un único clúster.

***El problema del K - Means***

Puede pasar a veces que si no se escalan/normalizan los datos, no anda nada bien porque asume que los datos son “esféricos”.

Como solución se suele utilizar algoritmos de clustering basado en densidades, como el HDBSCAN.

**Gráfico de dispersión

Descripción generada automáticamente**

**HDBSCAN**

Es un algoritmo de clúster o agrupamiento basado en la densidad que puede ser utilizado para identificar clústeres de cualquier forma en un conjunto de datos que contiene ruido y valores atípicos. La idea básica detrás del enfoque de agrupamiento basado en la densidad se deriva de un método intuitivo de agrupamiento humano. Por ejemplo, al mirar la figura, uno puede identificar fácilmente tres grupos junto con varios puntos de ruido, debido a las diferencias en la densidad de puntos.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Por lo tanto, los clústeres son regiones densas en el espacio de datos, separadas por regiones de menor densidad de puntos. El algoritmo HDBSCAN se basa en esta noción intuitiva de clústeres y ruido.

***K-means vs HDBSCAN***

| **K-means** | **Hdbscan** |
| --- | --- |
| La agrupación de K-means es sensible a la cantidad de agrupaciones especificadas. | No es necesario especificar el número de conglomerados. |
| La agrupación en clústeres de K-means no funciona bien con valores atípicos y conjuntos de datos ruidosos. | La agrupación en clústeres de HDBScan maneja de manera eficiente los valores atípicos y los conjuntos de datos ruidosos. |
| Las densidades variables de los puntos de datos no afectan el algoritmo de agrupación de K-means. | La agrupación en clústeres de HDBScan no funciona muy bien para conjuntos de datos dispersos o para puntos de datos con densidad variable. |
| La agrupación en clústeres de K-means es más eficiente para grandes conjuntos de datos. | HDBSCan Clustering no puede manejar de manera eficiente conjuntos de datos de gran dimensión. |