**Clustering**

En este post me gustaría hablar sobre diferentes estrategias de clustering de datos para problemas supervisados y no supervisados.

Previo a un análisis de casos especiales, me parece pertinente hacer una breve introducción al concepto de clustering. Supongamos que se tiene un conjunto de datos estructurado o tabular como por ejemplo un archivo .csv. En él habrá una gran cantidad de filas o ejemplos y otra cantidad de columnas o atributos. ¿Cómo podríamos hacer para extraer información relevante de este conjunto de datos? Si nuestro dataset contiene información de clientes de un supermercado y de los artículos que este compra, algo que podríamos querer hacer es agrupar a diferentes tipos de clientes para poder realizar ofertas direccionadas. El proceso de agrupar o armar clusters es lo que se conoce como clustering y tiene el propósito de extraer información mediante diferentes formas de agrupación.

**Hierarchical clustering**

El clustering jerárquico es un método de análisis de clusters que busca construir una jerarquía de clusters. Este tipo de clustering se basa en la idea que explica que un objeto está más relacionado con un ejemplo cercano que con uno que está a una distancia mayor. Por lo tanto, el clustering jerárquico conecta objetos o ejemplos basándose en la distancia entre ambos. A distancias diferentes, se formarán distintos clusters, que se pueden representar utilizando un dendograma.

Existen 2 estrategias para clustering jerárquico:

Agglomerative: Cada ejemplo comienza en su cluster propio y a medida que se aumenta la distancia (a medida que nos movemos hacia arriba en la jerarquía), se irán formando las agrupaciones.

Divisive: Los ejemplos comienzan agrupadops en un único cluster y se van separando recursivamente a medida que bajamos en la jerarquía.

Existen diferentes modos de medir las distancias entre clusters. Cada una tiene sus casos de aplicación particular, pero en el siguiente ejercicio, analizaremos el propósito de 2 de estos modos (complete link y single link).

**K-Means**

K-means es un tipo de clustering en el que el número de clusters a generar es un parámetro del algoritmo (k). Este algoritmo iniciará escogiendo k puntos aleatorios del dataset (centroides) y asignando un cluster a cada punto dependiendo de qué centroide esté más cerca de él. Luego, de forma iterativa, se vuelven a calcular los centroides tal que estos centroides (no necesariamente pertenecientes al dataset) minimicen la suma de las distancias a todos los puntos del cluster. Luego, se reclusterizan los puntos dependiendo de la proximidad de los centroides recalculados. La heurística termina cuando los centroides se hayan estabilizado o luego de un número máximo de iteraciones. Este algoritmo no siempre devuelve los mismos agrupamientos debido a la inicialización randómica de los centroides, por lo que es una buena práctica correr el algoritmo múltiples veces.

**Ejercicio**

Se desea agrupar los ejemplos del dataset ripley provisto por rapidminer en clusters.

Nota: Es de crucial importancia normalizar los atributos previo a realizar clustering, pues las formas de clustering discutidas utilizan la distancia euclídea para medir la similaridad entre ejemplos o clusters.

Diagrama

Descripción generada automáticamente

**Agglomerative clustering**

Para el agglomerative clustering, se decide hacer una comparación entre la utilización entre complete link y single link.

Complete link y distancia euclídea

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Single link y distancia euclídea

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Como se puede observar en las imágenes anteriores, la utilización de comlpete link obtiene clusters más parejos (uniformes) y distancias intraclusters medias menores que el single link. Aquí ya podemos entender el propósito de la utilización de cada uno.

Si se desea obtener clusters uniformes y parejos, la mejor forma de medir distancias entre clusters es complete link.

Si, por el contrario se desea localizar datos extremos o outliers, single link es la forma de distancia a elegir.

**Resultados** para agglomerative clustering con complete link:

Avg. within cluster distance: -31.972

Avg. within cluster distance for cluster 0: -37.551

Avg. within cluster distance for cluster 1: -13.095

Avg. within cluster distance for cluster 2: -34.942

**Top-Down Clustering**

En cuanto a top-down clustering, se decide utilizar k-means con Max depth=5 y

Max leaf size=1.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

**Resultados:**

kmeans k=2

Avg. within cluster distance: -36.971

Avg. within cluster distance for cluster 0: -55.705

Avg. within cluster distance for cluster 1: -23.788

Avg. within cluster distance for cluster 2: -13.534

**Conclusiones**

Se puede observar que la estrategia de aglomerative clustering obtuvo una menor distancia promedio intercluster que la estrategia utilizando top-down clustering con kmeans. Debido a esto, podemos argumentar que para este dataset, la mejor estrategia es la de agglomerative clustering

**Referencias:**

https://docs.rapidminer.com/latest/studio/operators/modeling/segmentation/agglomerative\_clustering.html