Práctica 13: Agrupamiento jerárquico

Unidad 2: Modelos avanzados de clustering

Tokio.

Ejercicio: Agrupamiento jerárquico

¿Qué vamos hacer?

En este tutorial, aprenderemos a realizar un agrupamiento jerárquico aglomerativo utilizando Python. Trabajaremos con datos de clientes de un centro comercial que incluyen sus ingresos anuales e índices de gastos. Nuestro objetivo es agrupar a los clientes en diferentes clústeres basándonos en su comportamiento de compra. Para lograr esto, utilizaremos un dendrograma para determinar el número óptimo de grupos y luego visualizaremos los resultados con un diagrama de dispersión.

Objetivos

- Cargar los datos de clientes y seleccionar las variables relevantes (ingresos anuales e índice de gastos).
- Generar un dendrograma para visualizar cómo se agrupan los clientes y determinar el número óptimo de clústeres.
- Realizar el agrupamiento jerárquico aglomerativo utilizando la distancia euclidiana y el criterio de enlace Ward.
- Visualizar los clústeres resultantes en un diagrama de dispersión, diferenciando a los clientes según su agrupación..

Pasos a seguir

- Debes cargar los datos desde un archivo CSV que contiene la información de los clientes del centro comercial. Selecciona únicamente las columnas de ingresos anuales e índice de gastos, que serán las variables clave para el análisis.
- Usa la función dendrogram de la biblioteca scipy para generar un gráfico que muestra cómo se agrupan los clientes de manera jerárquica. El dendrograma permitirá visualizar la estructura de los datos y determinar el número óptimo de clústeres.
- Al observar el dendrograma, debes identificar la línea paralela que intercepte el mayor número de líneas verticales sin cruzar ninguna línea horizontal. Esto te indicará el número óptimo de grupos para tu análisis.
- Con el número de clústeres decidido, realiza el agrupamiento utilizando el algoritmo de agrupamiento jerárquico aglomerativo. Este algoritmo utiliza la distancia euclidiana y el criterio de enlace Ward para minimizar las diferencias dentro de los grupos.
- Una vez que el algoritmo ha sido ejecutado, cada cliente será asignado a uno de los clústeres. Estas asignaciones permitirán agrupar a los clientes según su comportamiento.
- Finalmente, genera un diagrama de dispersión para visualizar los diferentes grupos de clientes. Cada grupo estará representado con un color distinto, lo que facilitará la interpretación de cómo los clientes están distribuidos en función de sus ingresos y su índice de gastos.

Entrega

Incluye el código Python utilizado, las gráficas generadas, conclusiones, etc