

Andrea Valentina Delgado Ruiz – 2240048

Nelson Andres Urrea Calvo - 2221115

Librería Bigtree en python

bigtree es una librería de Python que permite trabajar con estructuras jerárquicas de manera eficiente y flexible. Permite la implementación de:

- Árboles de manera general
- Árboles binarios
- Gráficos Acíclicos Dirigidos (DAG)

Esta librería proporciona una API sencilla para construir, modificar, visualizar y analizar jerarquías, siendo ideal para aplicaciones como estructuras de carpetas, árboles genealógicos, organizaciones empresariales, entre otras.

Algunas de sus funcionalidades son:

- Crear y manejar árboles personalizados con nodos que pueden contener múltiples atributos.
- Mover, copiar y reemplazar nodos, tanto dentro de un mismo árbol como entre diferentes árboles.
- Reconstrucción de árboles a partir de rutas (/padre/hijo) o estructuras como listas, diccionarios o pandas.DataFrame.
- Buscar nodos por nombre, atributos u otras condiciones.
- Visualizar árboles en texto o integrarlos con herramientas gráficas (como Graphviz).
- Modificación y manipulación de nodos.

Esta librería además contiene métodos avanzados para modificar árboles de forma dinámica como:

- Copiar nodos
- Desplazar nodos (cambiarlos de ubicación)
- Reemplazar nodos

- Copiar nodos de un árbol a otro
- Copiar y reemplazar nodos entre árboles distintos

Es muy útil porque permite gestionar tareas complejas como la edición, el traslado y la reorganización de estructuras jerárquicas. Es muy eficiente al manejar jerarquías complejas, facilita el desarrollo de sistemas estructurados por niveles y es compatible con análisis de datos debido a su integración con pandas.

Qué son los árboles de Huffman

La codificación Huffman es un algoritmo de compresión de datos sin pérdida que permite reducir el tamaño de un archivo sin eliminar información. Esto lo hace especialmente útil en situaciones donde se necesita optimizar el almacenamiento o la transmisión de datos, pero sin sacrificar su integridad. Se utiliza ampliamente en formatos como ZIP o JPEG, lo que demuestra su eficacia y adaptabilidad en diferentes contextos.

Este algoritmo funciona asignando códigos binarios más cortos a los caracteres que aparecen con mayor frecuencia, y códigos más largos a los que aparecen con menos frecuencia. Para hacerlo, construye lo que se conoce como un árbol de Huffman, una estructura binaria que organiza los caracteres según su frecuencia de aparición. En este árbol, cada carácter se representa como una hoja, y su código binario corresponde al camino que hay que seguir desde la raíz hasta esa hoja (ir a la izquierda significa un 0, e ir a la derecha un 1).

La construcción del árbol comienza calculando cuántas veces aparece cada carácter. Luego, se crea un nodo por cada uno y se utilizan estructuras como colas de prioridad para ir combinando los nodos menos frecuentes, formando subárboles hasta tener un único árbol que incluye a todos los caracteres. Gracias a esta organización, cada carácter tiene una única ruta de acceso, lo que permite una codificación y decodificación precisas, sin que se presenten ambigüedades.

La clave de la compresión está en que no todos los caracteres usan la misma cantidad de bits: los más comunes tienen códigos más cortos y los menos comunes, más largos. Esto hace que el tamaño total del archivo disminuya

considerablemente en comparación con otras formas de codificación que usan una longitud fija.