
Análisis de frecuencias a través de tablas y archivos XML

202201947 – Pablo Andres Rodriguez Lima

Resumen

El programa analiza frecuencias de sonido en un archivo XML y crea tablas con estos datos, simplificando su interpretación.

Estas tablas se organizan en conjuntos y se fusionan para una representación gráfica. Es esencial para quienes enfrentan grandes volúmenes de información en archivos XML. Al estructurar los datos, facilita la visualización y análisis, revelando patrones.

La característica de agrupar y fusionar tablas permite discernimiento en los datos, revelando tendencias. En un entorno de análisis de información compleja, esta aplicación es esencial, dotando a los usuarios con herramientas para obtener conocimiento de datos aparentemente caóticos.

Al estructurar estos datos en tablas, la aplicación simplifica la visualización y el análisis, ampliando su comprensión.

Palabras clave

Frecuencias

Interpretación

Amalgamar

Comprensión

Distintiva

Abstract

The program analyzes sound frequencies in an XML file and creates tables with this data, simplifying its interpretation. These tables are organized into sets and merged to create graphical representations. It is essential for those dealing with substantial volumes of information in XML files. By structuring the data, the program facilitates visualization and analysis, thereby revealing patterns. The feature of grouping and merging tables allows for a deeper understanding of the data, uncovering trends. In a complex information analysis environment, this application is indispensable, equipping users with tools to derive knowledge from seemingly chaotic data. Notably, the program proves to be an asset for those requiring extensive analysis of XML-contained data. The capacity to group and merge tables further enhances discernment, potentially exposing underlying patterns and trends.

Keywords

Frequencies

Interpretation

Merge

Comprehension

Distinctive

Introducción

En la era digital actual, la capacidad para procesar y analizar grandes volúmenes de datos se ha convertido en una habilidad esencial en una variedad de campos, desde la ciencia de datos hasta la ingeniería y más allá. En este contexto, el procesamiento de datos estructurados y su transformación en información significativa desempeñan un papel crucial en la toma de decisiones informadas y la generación de conocimiento.

Este proyecto se centra en la manipulación y análisis de datos contenidos en archivos XML, un formato ampliamente utilizado para representar información estructurada. Específicamente, se aborda la tarea de cargar datos desde un archivo XML que contiene información relacionada con el tiempo y la amplitud, analizar estos datos para identificar patrones significativos y agrupar las filas.

1. ¿Qué es Python?

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

1.1) ¿Cómo se utiliza Python?

El lenguaje Python se aplica a varios casos de uso en el desarrollo de aplicaciones, incluidos los ejemplos siguientes:

Desarrollo web del lado del servidor

El desarrollo web del lado del servidor incluye las funciones complejas de backend que los sitios web llevan a cabo para mostrar información al usuario.

Por ejemplo, los sitios web deben interactuar con las bases de datos, comunicarse con otros sitios web y proteger los datos cuando se los envía a través de la red.

Python es útil para escribir código del lado del servidor debido a que ofrece muchas bibliotecas que constan de código prescrito para crear funciones de backend complejas. Los desarrolladores también utilizan un amplio rango de marcos de Python que proporcionan todas las herramientas necesarias para crear aplicaciones web con mayor rapidez y facilidad. Por ejemplo, los desarrolladores pueden crear la aplicación web esqueleto en segundos porque no deben escribirla desde cero. Pueden probarla por medio de las herramientas de prueba del marco, sin depender de herramientas de prueba externas.

1.2) Automatización con scripts de Python

Un lenguaje de scripting es un lenguaje de programación que automatiza las tareas que suelen llevar a cabo las personas. Los programadores utilizan ampliamente los scripts de Python para automatizar muchas tareas diarias, como las siguientes:

- Cambiar el nombre de una gran cantidad de archivos a la vez
- Convertir un archivo en otro tipo de archivo
- Eliminar palabras duplicadas de un archivo de texto
- Llevar a cabo operaciones matemáticas básicas
- Enviar mensajes por email
- Descargar contenido
- Efectuar análisis básicos de registros
- Encontrar errores en varios archivos

- Realizar tareas de ciencia de datos y machine learning

La consiste en extraer conocimientos valiosos a partir de los datos, mientras que el enseña a las computadoras a aprender automáticamente de los datos y a efectuar predicciones precisas. Los científicos de datos utilizan Python para realizar tareas de ciencia de datos, como las que se indican a continuación:

- Corregir y eliminar datos incorrectos, lo que se conoce como limpieza de datos
- Extraer y seleccionar varias características de los datos
- , que consiste en agregar nombres significativos a los datos
- Buscar diferentes estadísticas a partir de los datos
- Visualizar los datos mediante el uso de tablas y gráficos, como los gráficos de líneas, los de barras, los circulares y los histogramas

1.3) XML tree and elements

XML es un formato de datos inherentemente jerárquico, y la forma más natural de representarlo es con un árbol. ET tiene dos clases para este propósito: ElementTree representa todo el documento XML como un árbol, y Element representa un solo nodo en este árbol. Las interacciones con todo el documento (lectura y escritura desde/hacia archivos) generalmente se realizan en el nivel de ElementTree. Las interacciones con un solo elemento XML y sus subelementos se realizan en el nivel de Element.

Implementaciones para resolución del proyecto:

- a) Clases con métodos y atributos.
- b) Clases del tipo nodo.
- c) Clases del tipo listas para almacenar datos.

a) Clases con métodos y atributos

Una clase define una plantilla o molde para crear objeto, los cuales son instancias de esa clase. Los objetos creados a partir de una clase tienen las mismas propiedades y comportamientos definidos por la clase, pero pueden tener valores diferentes para los atributos que se definen en la clase.

En Python, una clase se define mediante la palabra clave «class», seguida del nombre de la clase y dos puntos (:) y luego el cuerpo de la clase. El cuerpo de la clase contiene definiciones de métodos y atributos, que pueden ser públicos o privados según su acceso.

Un atributo es una variable que se define dentro de una clase, la cual almacena datos que pertenecen a un objeto de esa clase. Los atributos se utilizan para representar características o propiedades de un objeto, como su estado actual, su identificador, su tamaño, su color, etc.

Los atributos pueden ser de diferentes tipos de datos, como enteros, flotantes, cadenas, listas, diccionarios, entre otros. Además, los atributos pueden tener distintos niveles de visibilidad, que se especifican mediante los modificadores de acceso en la definición de la clase. Por defecto, los atributos son públicos en Python, lo que significa que puede accederse a ellos desde cualquier lugar del programa.

En la definición de una clase, los atributos se definen como variables que se inicializan en el método especial `__init__`. Por ejemplo, en la clase Persona

que definimos anteriormente, los atributos «nombre» y «edad» se definen de la siguiente manera:

b) Clases del tipo nodo.

Que es un nodo:

El nodo es un elemento de la lista el cual contiene: el elemento que queremos guardar, una referencia hacia el nodo anterior y una referencia al nodo siguiente.

Nodo de una lista.

Comenzaremos con construir un nodo utilizando una clase class Nodo: el constructor toma de valores a self y a elemento el cual será el elemento que se encuentre dentro del nodo, además, podemos ver que se encuentran otros dos atributos los cuales nos van a indicar las conexiones entre los nodos. Inicializamos siguiente y anterior a None ya al crear un Nodo queremos que no apunten a ningún otro nodo. Y por último tenemos al atributo elemento el cual será el dato que vamos a guardar en el nodo.

Clase y constructor de un Nodo.

Constructor de la lista doblemente ligada.

A continuación, crearemos una clase para la lista doblemente ligada class doubleList: la cual contiene únicamente el nodo inicial o el root de la lista el cual le da un inicio a la lista. Inicializamos root hacia None ya que al crear una lista queremos que este vacía. Durante el artículo crearemos varios métodos para interactuar con la lista, asegúrate de agregar esos métodos a esta clase.

c) Clases del tipo listas para almacenar datos.

Clase y constructor de una lista doblemente ligada.

¿Como se ve una lista cuyo root apunta al nodo?

La lista doblemente ligada inicia siempre con un root el cual va a apuntar al primer nodo de la lista.

Root referenciando al primer nodo de una lista.

Funciones para insertar un elemento a la lista doblemente ligada.

Insertando un elemento a una lista doblemente ligada vacía.

Para esta función tomaremos como parámetro a self y dato. Dato será el valor que va a tomar el elemento del nodo que vamos a insertar. Nuestra primera función será la siguiente: la cual va a recibir como parámetro a self y a dato el cual será el elemento que queremos guardar en el nodo. Al insertar nuestro primer elemento a la lista primero queremos revisar que la lista este vacía, si está vacía vamos a crear el primer nodo y pasaremos como parámetro el dato de la clase. En caso contrario vamos a desplegar un mensaje al usuario de que la lista no está vacía.

2. ¿Qué es Graphviz?

Es un conjunto de herramientas open-source realizado inicialmente en los laboratorios de investigación de AT&T para el dibujo de gráficos especificados en lenguaje de scripts DOT. Provee librerías para ser usadas por otras aplicaciones. Graphviz es software libre licenciado bajo CPL (Common Public License).

- Aplicaciones
- Estructuras de datos.
- Estructuras de árbol.
- Representación de análisis social de redes.
- Diagramas entidad relación.
- Diagramas de redes.
- Diagramas de flujo.
- Diagramas de procesos.

2.1) Grafos

Un grafo es un conjunto de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas (arcs en inglés) que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).

campos. Desde la ciencia de datos hasta la ingeniería y más allá, la capacidad de manipular datos estructurados y extraer información significativa puede ser una habilidad valiosa para profesionales en diversas disciplinas.

Conclusiones

- Eficiencia en el Procesamiento de Datos XML: El proyecto ha demostrado la eficiencia de las herramientas de programación, como Python, en el procesamiento de datos en formato XML. La capacidad de cargar, analizar y transformar datos estructurados de tiempo y amplitud en tablas y grupos ha facilitado significativamente la manipulación de información compleja y la generación de conocimiento a partir de ella.
- Potencial de Visualización: Las visualizaciones generadas a partir de los datos procesados han destacado el potencial de las representaciones gráficas en la comprensión de patrones y tendencias en los datos. La combinación de herramientas como Graphviz ha permitido presentar de manera efectiva resultados complejos, lo que puede ser valioso para la toma de decisiones informadas y la comunicación de hallazgos.
- Amplias Aplicaciones: Además de abordar la tarea específica de procesar datos de tiempo y amplitud en formato XML, este proyecto ha demostrado cómo las técnicas y herramientas utilizadas pueden aplicarse en una variedad de

Referencias bibliográficas

Gomez, O. A. (s/f). Visualizando Grafos usando Graphviz. OSiUX. Recuperado el 4 de septiembre de 2023, de <https://osiux.com/visualizando-grafos-graphviz.org>

Londoño, P. (2023, abril 26). Qué son las clases en Python, para qué sirven y cómo funcionan. Hubspot.es. <https://blog.hubspot.es/website/clases-python>

Pérez, J. R. C. (2020, noviembre 21). Implementando una lista doblemente ligada en Python. Medium. <https://a01153884.medium.com/implementando-una-lista-doblemente-ligada-en-python-b8def2b8df73>

xml.etree.ElementTree — The ElementTree XML API. (s/f). Python documentation. Recuperado el 4 de septiembre de 2023, de <https://docs.python.org/3/library/xml.etree.elementtree.html>

(S/f). Amazon.com. Recuperado el 4 de septiembre de 2023, de <https://aws.amazon.com/es/what-is/python/>

Apéndices

