

*Ejercicio de Laboratorio 6: ETL – P2*

Instituto Politécnico Nacional.

Escuela Superior de Cómputo.

Licenciatura en ciencia de datos.

Bases de Datos Avanzadas

Emiliano López Méndez.

**Introduccion**

En esta práctica, trabajaremos con un archivo de datos en formato CSV que contiene información sobre ventas, y nuestro objetivo es aplicar técnicas de limpieza y manipulación de datos utilizando la biblioteca pandas en Python. En particular, nos enfocaremos en la eliminación de filas con valores nulos en la columna state y en el almacenamiento del DataFrame procesado en un archivo XML. Este ejercicio es fundamental para comprender los procesos de transformación de datos que se utilizan en proyectos de ciencia de datos y análisis de datos, donde la integridad y la organización de la información son claves para obtener resultados precisos y confiables.

A lo largo de esta práctica, no solo aplicaremos habilidades técnicas de manejo de archivos CSV y XML, sino que también comprenderemos la importancia de los procesos de ETL (Extract, Transform, Load), específicamente en la fase de transformación, en la que se asegura que los datos estén en el formato adecuado antes de ser utilizados en análisis posteriores o cargados en sistemas de almacenamiento de datos.

.

**Desarrollo de la Actividad:**

**Carga de los Datos**: En primer lugar, importamos la biblioteca pandas y utilizamos la función read\_csv para cargar el archivo sales\_data.csv en un DataFrame. Esta función nos permite leer el archivo CSV y tener la información en memoria en una estructura tabular, lo que facilita su manipulación.

**Limpieza de Datos**: Identificamos y eliminamos las filas en las que la columna state contiene valores nulos o vacíos. Para ello, utilizamos el método dropna, que elimina las filas con valores faltantes en una columna específica. Este paso es crucial, ya que los valores nulos pueden afectar el análisis de los datos y llevar a resultados incorrectos si no se manejan adecuadamente.

**Almacenamiento en Formato XML**: Una vez que los datos están limpios, guardamos el DataFrame resultante en un archivo XML utilizando el método to\_xml. Este archivo se almacena en una ubicación diferente a la del archivo original para mantener una organización clara de los datos procesados y para que el archivo limpio esté disponible para futuras consultas o análisis.

**Parte 1 Extract and load data**

En el Paso 1, recuperará datos de un archivo plano CSV y utilizará el paso Entrada de archivo de texto para conectarse a un repositorio, ver el esquema del archivo y recuperar el contenido de los datos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

**Parte 2 Filter for missing codes**

Después de completar el Paso 1: Extraer y cargar datos, estará listo para agregar un componente de transformación a su canalización de datos. El archivo fuente contiene varios registros a los que les faltan códigos postales. Esta sección del tutorial filtra aquellos registros a los que les faltan códigos postales, donde el POSTALCODE no es nulo (la condición verdadera) y garantiza que solo se carguen registros completos en la tabla de la base de datos.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente Diagrama

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Parte 3 Resolve missing codes**

Después de completar el Paso 2: Filtrar los códigos faltantes, estará listo para resolver los códigos postales faltantes. En esta sección, aprenderá a utilizar un segundo archivo de texto que contiene una lista de ciudades, estados y códigos postales, para buscar los códigos postales de aquellos registros en los que faltan campos, que es la rama falsa de su Filtro. paso de filas.

Primero, utilizará un paso de entrada de archivo de texto para leer el archivo fuente. Luego, utilizará un paso de búsqueda de secuencia para incorporar los códigos postales resueltos a la secuencia. Por último, utilizará el paso Seleccionar valores para cambiar el nombre de los campos en la secuencia, eliminar campos innecesarios y más.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente Tabla

Descripción generada automáticamente Diagrama

Descripción generada automáticamente Diagrama

Descripción generada automáticamente con confianza media Interfaz de usuario gráfica, Texto, Aplicación, Tabla

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente Texto

Descripción generada automáticamente Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media Escala de tiempo

Descripción generada automáticamente

**Parte 4 Clean the data**

Después de completar el Paso 3: Resolver los datos faltantes, puede limpiar y categorizar aún más los datos en depósitos antes de cargarlos en una base de datos relacional. En esta sección, limpiará los datos del campo PAÍS asignando Estados Unidos a EE. UU. mediante el paso del asignador de valores. La limpieza de los datos garantiza que solo haya una versión de EE. UU.

Además, aprenderá a utilizar depósitos para categorizar los datos de VENTAS en categorías pequeñas, medianas y grandes mediante el paso Rango de números. Aprenderá cómo insertar estas funciones de limpieza y categorización en su transformación justo antes del paso Escribir en la base de datos en el lienzo.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Parte 5 Run the transformation**

Pentaho Data Integration proporciona varias opciones de implementación. La sección Ejecutar una transformación en el documento Integración de datos de Pentaho explica estas y otras opciones disponibles para su ejecución. En esta sección del tutorial, creará una transformación utilizando la opción Ejecución local.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja Tabla

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Parte 6 Orchestrate with jobs**

Los trabajos se utilizan para coordinar actividades de ETL como:

Definir el flujo y las dependencias que controlan el orden lineal para que se ejecuten las transformaciones.

Preparación para la ejecución comprobando condiciones como "¿Está disponible mi archivo fuente?" o "¿Existe una tabla?"

Realizar operaciones de carga masiva de bases de datos.

Ayudar a la gestión de archivos, como publicar o recuperar archivos mediante FTP, copiar archivos y eliminarlos.

Envío de notificaciones de éxito o fracaso a través de correo electrónico.

Para esta parte del tutorial, imagine que un sistema externo es responsable de colocar su entrada sales\_data.csv en su ubicación de origen todos los sábados por la noche a las 9 p. m. Quiere crear un trabajo que verifique que el archivo haya llegado y luego ejecute la transformación para cargar los registros en la base de datos. En un ejercicio posterior, programará el trabajo para que se ejecute todos los domingos por la mañana a las 9 a. m.

Imagen que contiene Diagrama

Descripción generada automáticamente Diagrama

Descripción generada automáticamente Texto

Descripción generada automáticamente

**Conclusión**

En esta práctica, logramos procesar y limpiar un conjunto de datos utilizando herramientas fundamentales de pandas en Python, enfocándonos en la eliminación de valores nulos y el almacenamiento en distintos formatos. A través de la eliminación de las filas con valores nulos en la columna state, comprendimos la importancia de asegurar la calidad de los datos, ya que los valores faltantes pueden introducir errores en análisis y decisiones posteriores.

Además, el proceso de conversión del DataFrame a formato XML destacó la versatilidad de pandas y nos brindó una experiencia práctica en la manipulación y transformación de datos para prepararlos para almacenamiento o integración en otros sistemas. Resolver los problemas técnicos, como la codificación y la instalación de bibliotecas adicionales, nos permitió afianzar habilidades para manejar situaciones comunes al trabajar con archivos de distintos orígenes y formatos.