ITESO, Universidad
Jesuita de Guadalajara

Extracción de datos de diferentes fuentes

Dr. Gaddiel Desirena López

Verano 2025

Contenido

Extracción de datos de diferentes fuentes

Muchos datos son de carácter confidencial, es por eso que no se pueden hacer públicos o no se puede tener acceso a ellos desde un servidor, por ello se verán algunas formas de obtener los datos a través de archivos.

Texto a *DataFrame*

El formato más simple para almacenar datos.
Para leer el archivo únicamente se debe conocer el separador

- Datos separados por comas
- Datos separados por espacios
- Datos alineados por tabular

Archivos de texto

- read_table: Lee un archivo genérico delimitado.
  sep = '$\backslash t$'.
- read_fwf: Lee una tabla de líneas con formato de ancho fijo.
  Debe ser una tabla alineada.
- read_csv: Lee un archivo de valores separados por coma.
  sep = ','.

Texto a *regular expression*

Para obtener un texto regular para procesamiento del lenguaje, se necesitan los datos como cadena de texto

- Abrir y leer el archivo
- Una librería para el tratamiento de cadenas de texto es 're'. Algúnas funciones son
  - search(pattern,string).
  - match(pattern,string).
  - split(pattern,string).
  - findall(pattern,string).
  - sub(pattern,repl,string).

Archivos CSV

Archivos separados por comas, éstos se leen, como ya se vio con la función 'read_csv' del paquete *pandas* y regresa un objeto *DataFrame*.

Archivos XLS o XLSX

Archivos generados por el software Excel o algún otro creador de hojas de cálculo. El paquete *pandas* ofrece la función 'read_excel('str')', donde 'str' es el nombre del archivo escrito entre comillas. La función regresa un *DataFrame* y soporta los siguientes formatos

- XLS
- XLSX
- XLSM
- XLSB
- ODF
- ODS
- ODT

Archivos XLS o XLSX

Otra forma de leer hojas de cálculo es creando un objeto de clase *pandas.ExcelFile*, éste cuenta con el método 'parse()'.

En ambos casos se puede especificar si cuentan o no con encabezado, el nombre o nombres de las hojas a leer, el nombre o nombres de las columnas, entre otras cosas.

Archivos JSON

Archivos para almacenar y compartir información de *Java Script*. Se pueden leer con la función de *pandas* 'read_json('str')', donde 'str' puede ser el nombre del archivo entre comillas, una dirección web u objetos de tipo *file*.

Archivos XML

Archivos con lenguaje *MarkUp*, *pandas* tiene la función 'read_html('str')', donde 'str', al igual que en el caso de los archivos JSON, puede ser el nombre del archivo entre comillas, una dirección url o un texto que contenga HTML.

Archivos XML

Los archivos XML se leen a través del método 'parse' del objeto *xml.etree.ElementTree()*, del elemento resultante se extrae la "raíz" con el método 'getroot()'. Otra opción es a través del método 'fromstring('str')' del mismo objeto, la diferencia ahora es que 'str' es el objeto de tipo *file* con formato XML.

Los elementos raíz son iterables, donde cada iteración, igual que un diccionario cuenta con *tag* y *attrib*, este último cuenta con la estructura de un diccionario.

Archivos SHP

Se utiliza para almacenar la ubicación geométrica y la información de atributos de estas entidades.
Es un formato multiarchivo. El número mínimo requerido es de tres y tienen las extensiones siguientes:

- SHP: es el archivo que almacena las entidades geométricas de los objetos.
- SHX: es el archivo que almacena el índice de las entidades geométricas.
- DBF: es la base de datos, en formato dBASE, donde se almacena la información de los atributos de los objetos.

Archivos SHP

Opcionalmente se pueden utilizar otros para mejorar el funcionamiento en las operaciones de consulta a la base de datos:

- PRJ: Es el archivo que guarda la información referida al sistema de coordenadas en formato WKT.
- SBN y SBX: Almacena el índice espacial de las entidades.
- FBN y FBX: Almacena el índice espacial de las entidades para los shapefiles que son inalterables (solo lectura).
- AIN y AIH: Almacena el índice de atributo de los campos activos en una tabla o el tema de la tabla de atributos. XML: Almacena los metadatos del shapefile.

Archivos SHP

Para importar estos datos en un *DataFrame* de *pandas*, instalamos e importamos el paquete 'geopandas', en él la función 'read_file('archivo')' lee los archivos disponibles con las extinciones descritas anteriormente.

De imágenes

Si se desea hacer un procesamiento de imágenes, como identificación de números, reconocimiento de patrones en firmas o un análisis del estado de salud de una planta con IA; la obtención de datos a partir de una imagen, ya sea en escala de de grises o a color, es imperativo. Estos datos se obtienen con la función 'imread' de *matplotlib.pyplot* y regresa una matriz en el caso de imágenes en escala de grises o un arreglo de tres o cuatro matrices para imágenes a color.