PANDAS 1: EXPLORACIÓN DE DATOS

PYTHON PROGRAMMING

nuclio digital school

Learn [to be] the future

Anna Perea Navarro

Senior Data Scientist

Graduada en Matemáticas por la Universidad de Barcelona y máster en Ingeniería Computacional y Matemática en la Universidad Rovira y Virgili.

CONTENIDO

01 Importación y preview del dataset (1/2)

Un dataset es un conjunto de datos ordenados, estructurados en formato tabular. Cuándo importamos esta tabla o dataset en un entorno Python mediante Pandas lo convertimos en un DataFrame, la estructura propia de Pandas para tratar con tablas de datos.

Importar un DataFrame de un fichero csv

Para obtener un DataFrame de un fichero csv, debemos usar la siguiente función: `pd.read_csv('ruta/nombre_archivo.csv')` e igualarla al nombre que queremos que tenga nuestro DataFrame, por ejemplo, df

Primera visualización de los datos

Sea df nuestro DataFrame ya importado a entorno Pyhton, podemos observar sus 5 primeras filas mediante el método `df.head()` y sus 5 últimas filas mediante `df.tail()`. Si indicamos un número n en el paréntesis obtendremos las n primeras o últimas filas respectivamente.

01 Importación y preview del dataset (1/2)

Una vez hemos importado nuestro dataset, convirtiéndose ahora en un DataFrame y nos hemos hecho una idea de los datos que contiene mediante los métodos .head() y .tail(), sigamos aumentando el conocimiento sobre el DataFrame mediante los siguientes atributos:

Forma

Sea df un DataFrame de Pandas:

- df.shape devuelve la forma del DataFrame, es decir su número de filas y de columnas, de la siguiente manera: (num filas, num columnas)

Tipo de datos

- En un mismo dataset podemos encontrar distintos tipos de datos, estos serán los mismos que hemos visto en el primer módulo de python (int, float, string, booleana) con la diferencia de que el tipo string se llamara object en pandas. Para conocer de que tipo son los datos de cada columna del DataFrame usaremos el atributo .dtypes
- En el caso de que queramos modificar el tipo del dato de alguna de las columnas podemos usar el método .astype de la siguiente manera: Sea df nuestro dataframe y col la columna que queremos modificar, por ejemplo de int64 a object, escribiremos df['col']=df['col'].astype('object')

02 Selección de columnas

Si solamente queremos seleccionar una columna para realizar algún cálculo o análisis, o bien si queremos seleccionar varias, deberemos realizar el acceso a los datos de la siguiente forma:

Sea df un DataFrame de Pandas y col_name la columna que queremos seleccionar:

- Mediante col = df['col_name'], obtendremos una Serie de Pandas almacenada en la variable col
- Mediante col = df[['col_name']] obtendremos un DataFrame almacenado en la variable col, ya que habremos pasado una lista entre los corchetes. Podríamos pasar más de una columna y obtener un subdataset: subdf = df[['col1', 'col2']]

Importante tener en cuenta el doble corchete para la selección de múltiples columnas: Uno para el acceso de las columnas y otro porque le estamos pasando una lista con las columnas que queremos seleccionar.

03 Distribución de los datos

Conociendo y entendiendo más nuestros datos podemos llegar a resultados directos o a información para seguir transformando nuestro dataset.

Estadísticos descriptivos

- df.describe() devuelve un resumen de la distribución de las variables numéricas continuas en el dataset, podemos aplicarlo a todo el dataframe df o a una columna numérica concreta llamada 'num_col' de la siguiente forma df['num_col'].describe()
- df['cat_col'].describe() también podemos aplicar la función describe a una variable categórica, obteniendo así unos descriptivos de la distribución distintos a los que obtenemos en variables numéricas, debido a la naturaleza de los datos categóricos.

Frecuencia

- df['col'].value_counts() nos dirá la frecuencia de cada uno de los valores encontrados en la columna en números absolutos
- Por defecto, se ordenará del elemento que aparezca con más frecuencia al que menos, si queremos que el orden sea inverso podemos usar el parámetro ascending=True en el paréntesis.
- Si queremos saber en porcentaje cuánta presencia tiene cada valor en una determinada columna, en vez del número absoluto de veces que aparece, deberemos usar el parámetro normalize=True en el paréntesis.