Práctica 1

Tipología y ciclo de vida de los datos

Paula de Jaime

Índice

Contexto

Los datos recogidos en respuesta a la práctica corresponden a relatos de avistamientos de OVNIS *(UFO en inglés, Unidentified Floating Object)*.

El encargado de recoger los relatos es el *National UFO Reporting Center*, centro fundado en 1974 por el investigador Robert J. Gribble. Durante las dos últimas dos décadas la función general del centro ha sido recibir, grabar, en su medida corroborar y documentar relatos de individuos que procesan haber sido testigos de experiencias inusuales, posiblemente relacionadas con el fenómeno UFO. [1]

Descripción del dataset

El título elegido para el dataset es “Relatos sobre experiencias UFO”.

Como bien el nombre indica, este dataset contendrá relatos sobre experiencias UFO reportadas por distintos individuos desde el 2019 hasta el 1800.

Contenido

Aunque el centro responsable de los datos fuera fundado en 1974, los relatos que se encuentran en este van desde el 2019 hasta antes del 1500. También existen ciertos relatos que no cuentan con una fecha determinada. En el dataset recopilado los relatos no serán más antiguos de 1800.

Por cada relato registrado se obtienen los siguientes datos:

- **Date/Time:** Será la fecha y la hora del suceso experimentado en formato *mm/dd/aa hh:mm*.
- **City:** La ciudad en la que ocurrió.
- **State:** El estado en el que se dió la experiencia UFO. Si la experiencia se dio fuera de Estados Unidos o no se ha especificado este campo estará vacío.
- **Shape:** Forma del avistamiento (círculo, cono, luz,..)
- **Duration:** Duración del avistamiento.
- **Summary:** Descripción de la experiencia.
- **Posted:** Fecha en la que se publicó el relato en la web.

El centro registra los relatos mediante llamadas telefónicas o mediante un formulario en su web.

Los datos de los relatos se muestran en páginas HTML, por este motivo se han logrado recopilar mediante la técnica de *Web Scraping* a través del lenguaje de programación Python.

Agradecimientos

El propietario del conjunto de datos es el *National UFO Reporting Center*, cuyo director se llama Peter Davenport, y el encargado de la web es Christian Stepien.

Inspiración

Lo desconocido siempre llama la atención y aunque la existencia de OVNIS no ha sido demostrada, poder analizar los datos recogidos sobre las personas que afirman haber sufrido alguna experiencia parece muy interesante.

Gracias a este conjunto de datos se podrán realizar análisis para responder a preguntas como:

- ¿Es cierto que solo se dan avistamientos en Estados Unidos?
- ¿Hay un periodo de tiempo al año que se dan más avistamientos?
- ¿Existe algún patrón de avistamientos durante los años?
- ¿Existe alguna correlación entre avistamientos y días de fiesta nacionales?

Licencia

La licencia escogida es **CC BY-NC-SA 4.0**, ya que al no ser la propietaria de los datos, no se cree justo que se puedan comercializar estos.

Esta licencia implica que los usuarios que usen este dataset: [2]

- *Deben dar crédito, proveer de un link a la licencia escogida, e indicar si se ha realizado algún cambio.* Con esta cláusula se reconoce el trabajo ajeno.
- *No pueden usar el material con propósitos comerciales.*
- *Las obras derivadas deberán tener la misma licencia.* Se evita así que obras derivadas puedan comercializarse.

Dificultades

La principal dificultad que se ha encontrado ha sido la codificación de las webs scrapeadas. Estás webs tenían dos codificaciones (Figura 1), así que se han tenido que realizar varias pruebas hasta que símbolos como mismamente unas comillas “`” no causaran fallos y símbolos “raros”.

```
1 <HTML>
2 <HEAD>
3 <META HTTP-EQUIV="Content-Type" CONTENT="text/html;charset=windows-1252">
4 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"><HTML><HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
  1">   <META NAME="GENERATOR" CONTENT="Mozilla/4.5b2 [en] (WinNT; I)
```

Figura 1 - Código fuente página web scrapeada

El problema ha sido que la librería BeautifulSoup automáticamente utiliza la codificación utf-8, y para poder obtener el texto en la codificación correcta (cp1252) se ha tenido codificar este ignorando los caracteres que den error, y decodificarlo

seguidamente. Un ejemplo de esto se encuentra en la Figura 2. El código de esta figura muestra cómo cada texto se codifica (encode) y decodifica (decode) utilizando una codificación dada.

```
row_elements = [x.text.replace(
    ";", "").encode("cp1252",
                    errors='ignore').decode(
    "cp1252") for x in row.find_all("font")]
```

Figura 2 - Texto

Como los textos se han ido guardando en un *dataframe* de la librería Pandas, para convertir estos a *.csv* también se ha tenido que tener en cuenta la codificación. Aunque los textos estén en el formato correcto, al guardarlos por defecto no se utiliza la codificación “cp1252”, así que esto hace que algunos caracteres se conviertan en símbolos “raros”. Por este motivo, en la instrucción para convertir el *dataframe* a un fichero *.csv* se añade el parámetro “encoding” (Figura 3).

```
try:
    dataframe.to_csv(output_file, index=False, encoding="cp1252")
except Exception as e:
    raise Exception("Error saving the file: {}".format(str(e)))
```

Figura 3 - Guardar fichero csv

Se necesitó bastante tiempo para resolver el anterior inconveniente descrito, ya que la codificación “cp1252” es idéntica a “iso-8859-1”, excepto por un rango de posiciones. Por este motivo y por falta de conocimientos, se estuvo intentando resolver el problema usando la segunda codificación nombrada. Además de que tener que codificar un texto para volver a decodificarlo según una codificación determinada no es muy intuitivo.

Anexo I. Necesidad de autenticación

Si una web requiere de autenticación para realizar un *scraping* el primero paso será obtener la url utilizada para iniciar sesión.

Por ejemplo, en la Figura 4:

1. Se define el usuario y contraseña como parámetros de una petición tipo POST.
2. Se realiza la petición POST con los parámetros anteriores a la url de inicio de sesión (*auth_url*).
3. Se obtiene la *cookie* de la respuesta. Este parámetro es el que habrá que mandar en cada petición que se quiera realizar e indicará que estamos identificados.
4. La cookie se guarda en un diccionario Python para futuras llamadas.

```
data = {'username': FANFIC_USERNAME,
        'password': FANFIC_PASSWORD}
r = requests.post(auth_url, data=data)
session_cookie = r.cookies.get(cookie_name)

cookies = {cookie_name: session_cookie}
```

Figura 4 - Inicio de sesión

Ahora si se quiere obtener el código *html* de una web que requiere de inicio de sesión, simplemente se hace la llamada correspondiente enviando la *cookie* en los *headers* de la petición (Figura 5).

```
page = requests.get(url, cookies=cookies,
                    timeout=(connect_timeout, read_timeout)).text
page_html = BeautifulSoup(page, 'html.parser')
```

Figura 5 - Petición de una página protegida

La página se obtendrá correctamente.

Anexo II. Descarga de imágenes

Para descargar imágenes estáticas suponiendo que estas están incrustadas en un sitio web:

1. Se obtendrá el *html* de dicha web de forma normal.
2. Se preparará una lista de enlaces de aquellas imágenes que se desea descargar.

   Ayudará el método *find_all* de BeautifulSoup, la etiqueta “img”, y los identificadores o las clases que creamos oportunas.
3. Se hace una petición a la url de cada imagen, y la respuesta (datos binarios) se guarda en un fichero:

```
if r.status_code == 200:
        r.raw.decode_content = True
        f = open(nombre.jpg, "wb" )
        shutil.copyfileobj(r.raw, f)
        f.close()
```

Bibliografía

[1] The National UFO Reporting Center

http://www.nuforc.org/General.html (accedido el 26-10-2019)

Information and Policies

[2] Creative Commons

https://creativecommons.org/licenses/by-nc-sa/4.0/ (accedido el 26-10-2019)

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)