Práctica 1 [P1]
Web Scraping

Tipología y Ciclo de Vida de los Datos

Muñoz, D., Deniz, L.

Máster en Ciencia de Datos

Universitat Oberta de Catalunya

20 de marzo de 2022

Universitat Oberta de Catalunya

Índice

Contextualización

La Comisión Sectorial de Investigación Científica (CSIC) es un servicio de la Universidad de la República, principal institución educativa de alto nivel en Uruguay concentrando el 90% de la población universitaria. Dentro de los cometidos de la CSIC está la administración, estudio y financiación de programas de investigación, cuyos proyectos abarcan todas las disciplinas en todas las áreas.

Uno de estos programas es el Programa de Apoyo a la Investigación Estudiantil (PAIE), que permite a los estudiantes universitarios de grado tener una primer experiencia participando en sus propios proyectos de investigación, orientados en todo momento por profesores responsables y expertos en las áreas en las que estos se desarrollan.

Este instrumento es el dispositivo de salida, en muchos casos, de motivación y de *hook*, para ampliar la capacidad de investigación y de fomentar el interés por la carrera como investigador. Si bien el monto total financiado es demasiado bajo para una investigación de alto impacto, no se busca que estos proyectos tengan una trascendencia internacional, sino alcanzar una rigurosidad en términos de metodología y razonamiento. Bianco y Sutz (2014), lo definen como un programa que se basa en aprendizaje basado en problemas y aprendizaje basado en proyectos.

El título para este DataSet será entonces “PAIES - *Proyectos de investigación de estudiantes de grado en UdelaR*”, y el nombre del fichero “paies.csv”.

La información se toma desde la web del programa, accediendo a cada página que contiene la información del listado de proyectos por año del llamado, de ahí se guardan las URLs de cada proyecto en cuestión y luego se toman los datos de esa segunda instancia.

El total de registros asciende a 825 filas y 6 columnas, fichero que se adjunta a este documento, abarcando los años desde el 2008 al 2019 inclusive.

La información que aquí se presenta fue recolectada utilizando técnicas de *web scraping* desde la web https://estudiantes.csic.edu.uy, estando esta de manera pública a toda la comunidad a través de un WordPress.

En la imagen adjunta, se pueden apreciar los primeros y los últimos registros que han sido incorporados en el dataset.

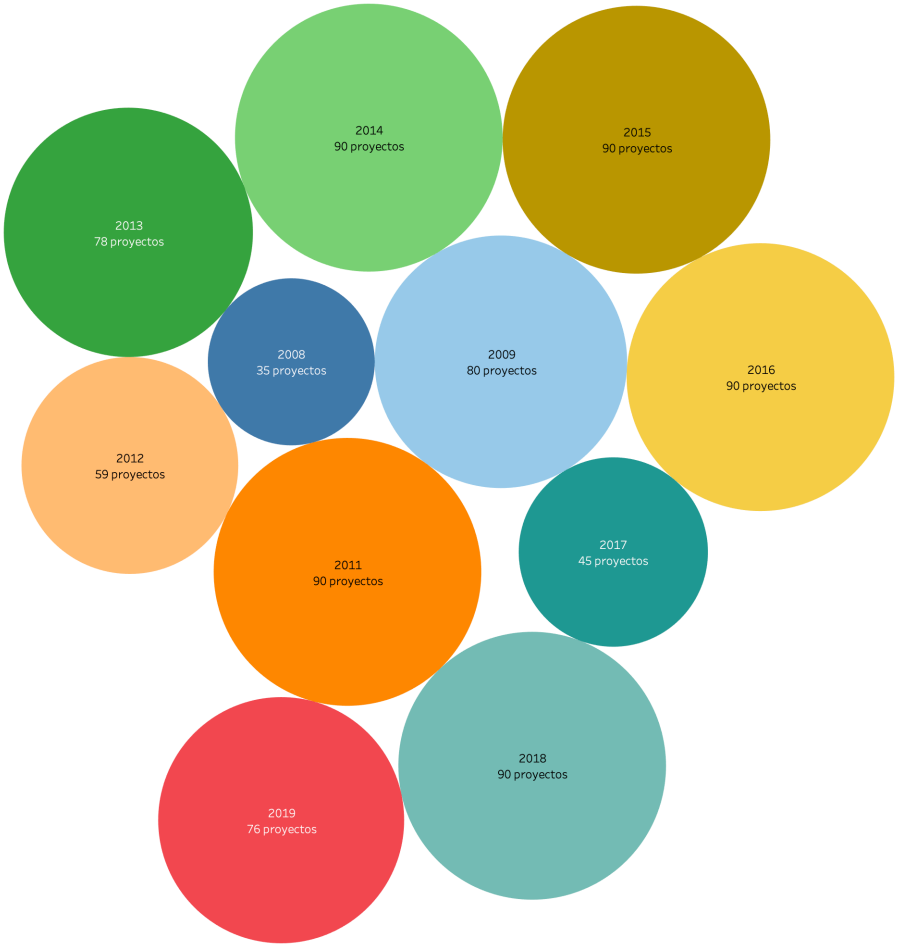
	Title	Link	Description	Image	Year	Published
0	[RN] Estudio de la prevalencia de consumo de a...	https://www.estudiantes.csic.edu.uy/2015/09/02...	Integrantes del equipo: Jose de Paula, Ines Fa...	img/0.jpg	2008	2016-12-06T14:38:21-03:00
1	[FQ] Caracterización de frutos nativos del Uru...	https://www.estudiantes.csic.edu.uy/2015/09/02...	Las frutas y vegetales tienen un rol important...	img/1.jpg	2008	2016-12-06T14:38:22-03:00
2	[FQ] Caracterización Botánica y de los volátil...	https://www.estudiantes.csic.edu.uy/2015/09/02...	La miel es un producto natural elaborado por l...	img/2.jpg	2008	2016-12-06T14:38:23-03:00
3	[FENF] Primer Control Puerperal	https://www.estudiantes.csic.edu.uy/2015/09/02...	Integrantes del equipo: Laura Ferreira, Ana Pa...	img/3.jpg	2008	2016-12-06T14:38:23-03:00
4	[FVET] Fasciola hepatica: dinámica integral de...	https://www.estudiantes.csic.edu.uy/2015/09/02...	Integrantes del equipo: Soledad Birriel, Nicol...	img/4.jpg	2008	2016-12-06T14:38:23-03:00
..	...	...	...	...	...	...
820	Representaciones sociales de personas adultas ...	https://www.estudiantes.csic.edu.uy/2021/08/03...	El presente informe expone los resultados y la...	None	2019	2021-08-03T15:19:36-03:00
821	Relevamiento e incidencia de mortalidad antrop...	https://www.estudiantes.csic.edu.uy/2021/08/03...	Los mamíferos marinos (MM) son depredadores to...	None	2019	2021-08-03T14:55:54-03:00
822	Economía muscular en corredores	https://www.estudiantes.csic.edu.uy/2021/08/03...	Como pregunta problema se expuso: ¿Cuáles son ...	None	2019	2021-08-03T14:51:37-03:00
823	Aproximación al estudio de la cardiomiopatía d...	https://www.estudiantes.csic.edu.uy/2021/08/03...	La diabetes mellitus (DM), es una enfermedad m...	None	2019	2021-08-03T14:48:37-03:00
824	Producción de corporalidad en dispositivos est...	https://www.estudiantes.csic.edu.uy/2021/08/03...	El proyecto indaga la relación entre producció...	None	2019	2021-08-03T14:45:06-03:00

[825 rows x 6 columns]

Los campos que se han almacenado son los siguientes:

- **Título**: identificador del proyecto y contextualizador general.
- **Link**: permite el acceso directo al proyecto publicado.
- **Description**: contenido relativo al proyecto y datos opcionales.
- **Image**: póster asociado al trabajo donde puede extraerse información adicional.
- **Year**: edición del proyecto donde fue financiado.
- **Published**: año en que se publica en la página web desde donde se extrae.

Para la visualización de los datos del diagrama siguiente, aunque de una manera muy básica, se ha utilizado Tableau Public.



Distribución de los 825 proyectos, por año

En el capítulo titulado "*Los jóvenes y la investigación*" de Bianco y Sutz (2014), se encuentra un análisis más pormenorizado del impacto que ha tenido el programa en la vida académica de los estudiantes, así como su evolución en el tiempo. Los datos son propiedad de CSIC y, en ausencia de licenciamiento explícito, serán tratados bajo licencia CC-BY-ND 4.0 como se explicita en el apartado correspondiente.

La estructura de los datos recolectados son poco numéricos, podría decirse que su contenido es más bien de carácter textual lo que pondría a disposición un conjunto de datos para posteriores análisis semánticos. Podría a través de este dataset encontrarse patrones sobre los principales intereses en términos de investigación durante el grado, a fin de potenciar aquellas áreas o disciplinas que pudieran ser menos explotadas, o bien incorporar aquellos proyectos en el marco de uno superior dentro de una de sus líneas de investigación.

Por otro lado, sirve para la definición de políticas de investigación y reestructuración de cara a la creación de investigadores en áreas desde temprano en los procesos formativos, alcanzando niveles de madurez más avanzados en el menor plazo si estos proyectos se extendieran en el tiempo y, en algunos casos, se rijan por definiciones de importancia.

Código fuente

La estructura de ficheros es la siguiente:

```
TCVD-P1/
  |_________ img/
  |          |_________ 5_archivo.jpg
  |          |_________ 6_archivo.jpg
  |          |_________ ...
  |          |_________ 384_archivo.jpg
  |          |_________ 385_archivo.jpg
  |
  |_________ output/
  |          |_________ paies.csv
  |
  |_________ app.py
  |_________ utils.py
```

La raíz de la aplicación se encuentra en *app.py*, desde donde se lanzan las instrucciones que dan lugar a la recolección de datos. Además, las funciones reiterativas pueden encontrarse en *utils.py*, responsable de tener aquellas funcionalidades que son consumidas por el proceso de *scraping*.

En cuanto a la estructura de directorios, *img* contiene las imágenes que se descargan desde el sitio web de destino relacionadas a cada proyecto obtenido, si es que lo tiene. En el directorio *output* en cambio, se guarda el dataset generado a través del *scraping* en formato CSV.

Si bien la mayor parte del código se encuentra comentado para permitir una lectura transversal y entender qué se hace paso a paso, sería conveniente aclarar las diferentes medidas que se han tomado para evitar prevenciones de *web scraping* en sitios web.

Primeramente se creado una función que permite elegir aleatoriamente entre una serie de cabeceras para las peticiones HTTP, haciendo que cada una se haga desde distintos navegadores.

Por otro lado, las llamadas a través de Requests tienen un tiempo de *delay* de 4 veces el tiempo de retardo entre que se ejecuta la petición y se devuelve el resultado de la anterior, tal y como se puede ver en la bibliografía recomendada de la asignatura. La diferencia de 4 veces el *delay* es para las llamadas a *scrapear* la primera vista donde se listan los proyectos. De cara a pedir proyecto a proyecto, el tiempo aumenta a 6 veces este *delay*.

La implementación de implementarlo sobre un proxy que permita variar las direcciones IP desde donde se hacen las Requests no se ha implementado por dos motivos principales: el primero de ellos es por el costo que representa para este equipo y el segundo aún más importante, es que el caso de uso elegido no presenta limitaciones a la hora de hacer las peticiones, por lo que no fue *conditio sine qua non* para la obtención del dataset.

Código fuente de la aplicación

```
import requests # para manejar las urls
import pandas as pd # para manejo de dataframes
from bs4 import BeautifulSoup
from utils import *
import time
from tqdm import tqdm
```

```
# Años donde buscar la información, 2010 se excluye por falta de datos
years = [2008, 2009, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# DataFrame en pandas a rellenar
df = pd.DataFrame(columns = ['Title', 'Link', 'Description', 'Image', 'Year', 'Published'])

# Enlaces a todos los proyectos donde recuperar la información
projects = []

# Procesa año a año
print(f" > Processing editions by year")

for year in tqdm(years):

  # Hará variar el tiempo de respuesta
  t0 = time.time()

  try:
    # A través de f-strings parametrizar la consulta para tomar cada una de las ediciones de los proyectos
    page = requests.get(f'https://www.estudiantes.csic.edu.uy/category/proyectos-aprobados/
                proyectos-{year}/', headers = {'User-Agent': get_header()})
  except requests.exceptions.Timeout:
    # Si cae el servidor o se demora, intenta con el siguiente
    pass

  # Mide el tiempo de respuesta
  response_delay = time.time() - t0

  # Demora la siguiente llamada 10 veces el tiempo de respuesta inicial
  time.sleep(4 * response_delay)

  # Obtener la información de cada url a visitar para tomar los datos
  soup = BeautifulSoup(page.content, "html.parser")

  try:
    # Obtiene todos los títulos h2 que contienen los enlaces
    for tmp in soup.find_all('h2'):

      # Toma los enlaces propiamente
      tmp = tmp.a['href']

      # Los añade a la lista de enlaces a proyectos que necesita
      projects.append([tmp,year])
  except Exception as e:
    print(e)
    pass

# Indica el número de proyecto que está evaluando
count = 0

# Rascar la información de cada uno de los proyectos partiendo del enlace
print(f" > Processing projects")
for project in tqdm(projects):

  # Hará variar el tiempo de respuesta
  t0 = time.time()
  page = requests.get(project[0], headers = {'User-Agent': get_header()})

  # Mide el tiempo de respuesta
  response_delay = time.time() - t0

  # Demora la siguiente llamada 10 veces el tiempo de respuesta inicial
  time.sleep(6 * response_delay)

  soup = BeautifulSoup(page.content, "html.parser")

  # Verifica que los campos tengan contenido, sino los pone a None
  try:
    p_title = soup.h1.string.strip()
  except Exception as e:
    p_title = None

  try:
    p_description = soup.div(class_='post-content')[0].p.getText().strip()
  except Exception as e:
    p_description = None

  try:
    image_path = soup.div(class_='post-content')[0].p.a['href'].strip()
    p_image = load_requests(image_path, count).strip()
```

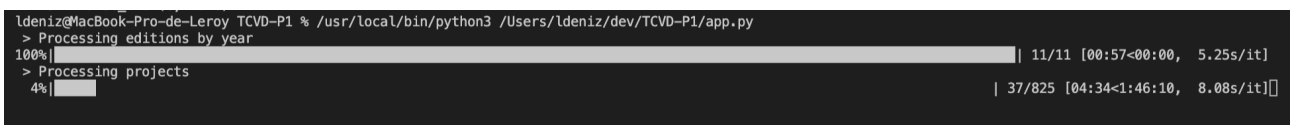
```
  except Exception as e:
    p_image = None

  try:
    p_pub_date = soup.find('span', {'class': 'updated'}).text
  except Exception as e:
    p_pub_date = None

  # Añadir la nueva fila al dataframe
  df = df.append({'Title': p_title, 'Link': project[0], 'Description': p_description, 'Image': p_image,
        'Year': project[1], 'Published': p_pub_date}, ignore_index=True)

  count += 1

# Crea el fichero CSV en el directorio output-
df.to_csv('output/paies.csv')
print(df)
```

Código fuente de utilidades

```
# Fichero con utilidades
import requests
import os.path
import random

# Función para obtener la imagen de una URL
def load_requests(source_url):
  r = requests.get(source_url, stream = True)
  if r.status_code == 200:

    # Separar la urle /: ['http:', '', 'www.estudiantes.csic.edu.uy', 'wp-content', 'uploads', '2015',
        '09', '32114scr_654aa682ea813a1195.jpg']
    aSplit = source_url.split('/')

    # Tomar el nombre del fichero
    filename = aSplit[len(aSplit)-1]

    _, ext = os.path.splitext(filename)
    if verify_extension(ext):

      # Separar el nombre del fichero de imagen: ['32114scr_654aa682ea813a1195', 'jpg']
      ruta = f"img/{id}{ext}"
      output = open(ruta,"wb")
      for chunk in r:
        output.write(chunk)
      output.close()
      return ruta
  return None


# Función para verificar la extensión de la imagen
def verify_extension(ext):
  if extension in ['.png', '.jpg', '.jpeg', '.webp', '.svg', '.gif']:
    return True
  return False


# Función para hacer variar las cabeceras de las peticiones HTTP
def get_header():
  # Definir cabeceras para envío de petición HTTP
  headers_options = ['Mozilla/5.0','Chrome/42.0.2311.135','Safari/537.36','Edge/12.246','AppleWebKit/537.36']

  # Elige un tipo de cabecera aleatoria por llamada
  return random.choice(headers_options)
```

Vista de procesamiento

Para facilitar al usuario un seguimiento del avance del procesamiento de los datos, se utilizó la librería tqdm que permite visualizar una *progress bar* en función de un iterable.



Accesos y licencia

Tal y como se solicita en el enunciado, se pone a disposición el siguiente link a GitHub donde pueden encontrarse el código, los ficheros auxiliares y la documentación relativa a esta práctica.

https://github.com/leroydeniz/TCVD-P1

Además, el mismo fue publicado en Zenodo bajo licencia Creative Commons, habiéndose asignado el **DOI 10.5281/zenodo.6371455** y puede accederse a través del siguiente enlace:

https://doi.org/10.5281/zenodo.6371455

La selección de esta licencia permite al usuario compartir y redistribuir, así como adaptar, transformando, remezclando y creando a partir de los datos iniciales. Esta licencia es lo suficientemente libre para permitir el manejo de los datos con suma maleabilidad, sin necesidad de restricciones de creación o manipulación a partir de ellos.

El sitio web desde donde se extrae la información no aclara nada acerca del licenciamiento de sus datos publicados, por lo que en el entendido que merece un reconocimiento el poner los datos a disposición, así como el trabajo de recabarlos, corresponde su cita a través de la licencia BY. No obstante, como son datos públicos pero que representan una realidad en el tiempo ya invariable, además de pertenecer a casos reales desde donde fueron creados, no se permite su modificación; de esta manera se asegura la integridad de los datos a la hora de trabajar con ellos.

Podría citarse al blog García Nieto (2018), que en una entrada titulada "*Licencias Creative Commons explicadas para dummies*" define a la licencia elegida como "*Puedes usar una obra con esta licencia, sea o no con ánimo de lucro, siempre y cuando cites al autor y no hagas una obra derivada de ella. A modo de ejemplo, podrías coger una foto de un atardecer para ilustrar un artículo, pero no editarla para añadir texto, otras imágenes, etc.*".

Logo de la licencia CC-BY-ND 4.0

Bibliografía

- Bianco, M., & Sutz, J. (2014). *Veinte años de políticas de investigación en la Universidad de la República. Aciertos, dudas y aprendizajes* (junio de 2014). TRILCE. Recuperado 13 de marzo de 2022, de https://www.csic.edu.uy/sites/csic/files/libro_veinte_anos_de_politicas_de_investigacion_en_la_universidad_de_la_republica.pdf
- Creative Commons. (s. f.). *Reconocimiento-CompartirIgual 4.0 Internacional (CC BY-SA 4.0)*. Recuperado 13 de marzo de 2022, de https://creativecommons.org/licenses/by-sa/4.0/deed.es_ES
- García Nieto, J. (2018, septiembre 15). Licencias Creative Commons explicadas para dummies [Blog]. *GENBETA*. https://www.genbeta.com/herramientas/licencias-creative-commons-explicadas-para-dummies
- Lawson, R. (2015). Web Scraping with Python. Packt Publishing Ltd. Chapter 2. Scraping the Data.
- Masip, D. El lenguaje Python. Editorial UOC.
- Simon Munzert, Christian Rubba, Peter Meißner, Dominic Nyhuis. (2015). Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining. John Wiley & Sons.
- Subirats, L., Calvo, M. (2018). Web Scraping. Editorial UOC.
- Tutorial de Github https://guides.github.com/activities/hello-world.

Contribución

Contribución	Firma
Investigación previa	DM, LD
Redacción de las respuestas	DM, LD
Desarrollo del código	DM, LD

Muñoz, D., Deniz, L.
Máster en Ciencia de Datos
Universitat Oberta de Catalunya
20 de marzo de 2022