

Práctica 1- Tipología y ciclo de vida de los datos

Autores

- David Fernández Álvarez
- Sara Robisco Cavite

1. Contexto

El contexto de esta práctica es la obtención de las detecciones de ondas gravitacionales por parte de LIGO, VIRGO y KAGRA, tanto aquellas confirmadas y publicadas como las descartadas.

Para ello obtendremos los datos de la web del consorcio LIGO, VIRGO y KAGRA. En esta web no sólo exponen información sobre los diferentes observatorios y sus actividades, sino que además muestran y comparten sus datos de manera pública. El enlace es <https://www.gw-openscience.org/> (<https://www.gw-openscience.org/>).

Hemos escogido esta web porque nos ha parecido un conjunto de datos original dentro de una web lo suficientemente compleja para la elaboración de la práctica.

2. Título del dataset

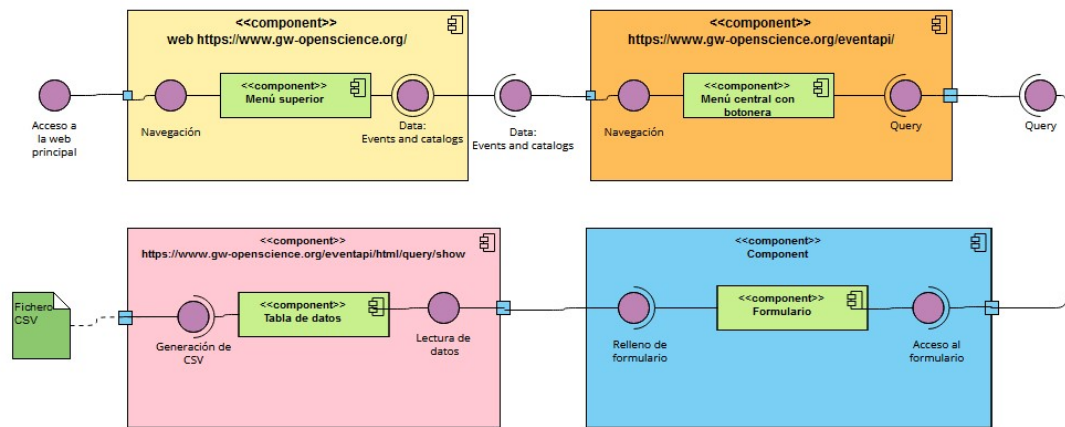
Hemos decidido denominar al dataset `detecciones_ondas_gravitacionales.csv`

3. Descripción del dataset

El conjunto de datos extraído son los datos de las detecciones de ondas gravitacionales, tanto aquellas confirmadas como aquellas que se han asociado a ruido y se han descartado. Nuestra aplicación recoge aquellas detecciones dentro del intervalo de fechas que seleccionemos con la intención de ir las añadiendo a una base de datos sin necesidad de traerse todos los datos. Esto también lo hacemos porque la carga de los datos es lenta para la tabla completa, por lo que es preferible ir descargando los últimos datos periódicamente e insertarlos en una base de datos mediante herramientas como Pentaho.

4. Representación gráfica

La representación visual de la generación del dataset es la siguiente:



5. Contenido

Los campos de nuestro dataset son:

- **Name:** identificador de la detección.
- **Version:** versión de la detección. Se revisan periódicamente.
- **Release.**
- **GPS:** fecha y hora de la detección en formato GPS.
- **Mass_1:** masa del primer objeto en masas solares.
- **Mass_1_upper:** valor máximo del rango de error de la masa del primer objeto.
- **Mass_1_lower:** valor mínimo del rango de error de la masa del primer objeto.
- **Mass_2:** masa del segundo objeto en masas solares.
- **Mass_2_upper:** valor máximo del rango de error de la masa del segundo objeto.
- **Mass_2_lower:** valor mínimo del rango de error de la masa del segundo objeto.
- **Network_snr:** ratio señal/ruido en la red.
- **Network_snr_upper:** valor máximo del rango de error del ratio señal/ruido en la red.
- **Network_snr_lower:** valor mínimo del rango de error del ratio señal/ruido en la red.
- **Distance:** distancia a la que se ha producido la colisión en Megapársecs.
- **Distance_upper:** valor máximo del rango de error de la distancia.
- **Distance_lower:** valor mínimo del rango de error de la distancia.
- **chi_eff:** correlación de campo z de las fusiones de agujeros negros binarios.
- **chi_eff_upper:** valor máximo del rango de error de la correlación de campo.
- **chi_eff_lower:** valor mínimo del rango de error de la correlación de campo.
- **Total_mass:** masa total de ambos cuerpos. Medida en masas solares.
- **Total_mass_upper:** valor máximo del rango de error de la masa total.
- **Total_mass_lower:** valor mínimo del rango de error de la masa total.
- **Chirp_mass:** masa efectiva de un sistema binario. Medida en masas solares.
- **Chirp_mass_upper:** valor máximo del rango de error de la masa efectiva.
- **Chirp_mass_lower:** valor mínimo del rango de error de la masa efectiva.
- **Detector_Frame_Chirp_Mass:** marco del detector de la masa efectiva. Medida en masas solares.
- **Detector_Frame_Chirp_mass_upper:** valor máximo del rango de error del marco del detector de la masa efectiva.
- **Detector_Frame_Chirp_mass_lower:** valor mínimo del rango de error del marco del detector de la masa efectiva.

- **Redshift**: corrimiento al rojo, marca la velocidad a la que se alejan de nosotros.
- **Redshift_upper**: valor máximo del rango de error del corrimiento al rojo.
- **Redshift_lower**: valor mínimo del rango de error del corrimiento al rojo.
- **False_Alarm_Rate**: tasa de falsa alarma. La medida es años elevado a -1.
- **P_astro**: probabilidad de que el evento tenga un origen astrofísico.
- **Final_mass**: masa final del objeto resultante tras la colisión. Medida en masas solares.
- **Final_mass_upper**: valor máximo del rango de error de la masa final.
- **Final_mass_lower**: valor mínimo del rango de error de la masa final.

6. Propietario

Para obtener el propietario de la web hemos hecho un script de python, para poder ejecutarlo hemos instalado la librería python-whois. A continuación se muestra el script y los datos del propietario de la web:

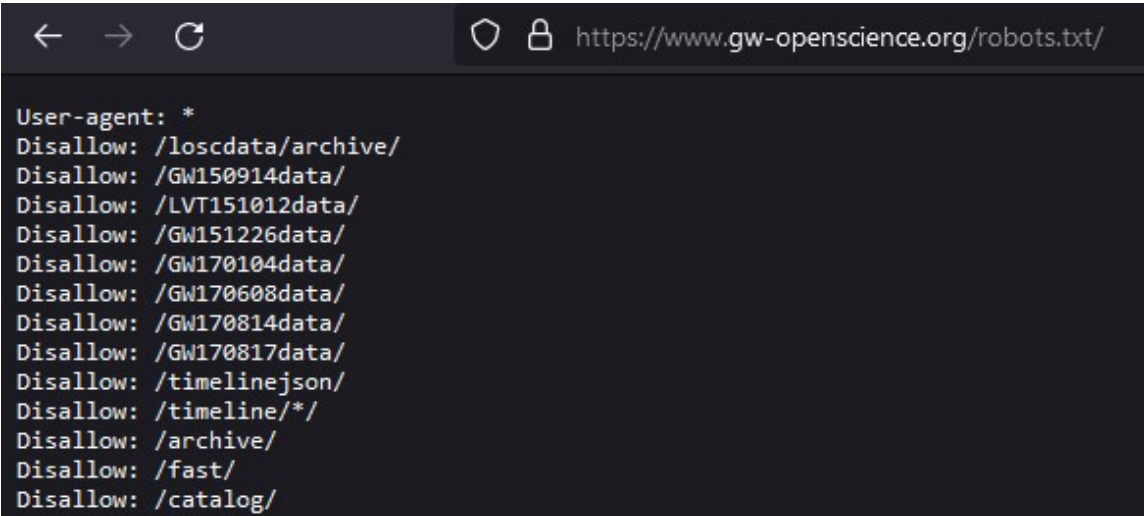
```
In [1]: ▶ import whois
w = whois.whois('https://www.gw-openscience.org')
print(w)

{
  "domain_name": "gw-openscience.org",
  "registrar": "Domain.com, LLC",
  "whois_server": "http://whois.domain.com",
  "referral_url": null,
  "updated_date": "2022-06-12 19:25:09",
  "creation_date": "2017-09-04 21:11:08",
  "expiration_date": "2027-09-04 21:11:08",
  "name_servers": [
    "ligo.ligo.caltech.edu",
    "mercutio.ni.caltech.edu",
    "tepid.ni.caltech.edu"
  ],
  "status": [
    "clientTransferProhibited https://icann.org/epp#clientTransferProhibited",
    "clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited"
  ],
  "emails": "compliance@domain-inc.net",
  "dnssec": "unsigned",
  "name": "REDACTED FOR PRIVACY",
  "org": "Domain Privacy Service FBO Registrant.",
  "address": "REDACTED FOR PRIVACY",
  "city": "REDACTED FOR PRIVACY",
  "state": "FL",
  "registrant_postal_code": "REDACTED FOR PRIVACY",
  "country": "US"
}
```

Observamos que el propietario de la web es "Domain Privacy Service FBO Registrant.", esto suena a que no quieren mostrar abiertamente los datos del propietario. Sabemos que están en Estados Unidos y por los nombres de los servidores podemos deducir que realmente el dueño es el Caltech.

Al observar lo celosos que son de su privacidad nos dio miedo que no nos dejaran hacer

webscraping, así que lo primero que hicimos fue consultar su archivo Robots.txt, situado en <https://www.gw-openscience.org/robots.txt/> (<https://www.gw-openscience.org/robots.txt/>). Veamos su contenido:



```
User-agent: *
Disallow: /loscdata/archive/
Disallow: /GW150914data/
Disallow: /LVT151012data/
Disallow: /GW151226data/
Disallow: /GW170104data/
Disallow: /GW170608data/
Disallow: /GW170814data/
Disallow: /GW170817data/
Disallow: /timelinejson/
Disallow: /timeline/*/
Disallow: /archive/
Disallow: /fast/
Disallow: /catalog/
```

Por lo que vemos sólo hay restricciones a ciertas páginas, evitando acceder a ellas no incumpliremos sus restricciones y actuaremos de manera ética. Además hemos insertado en el código retardos de tiempo para evitar saturar el servidor con nuestra actividad. Para evitar saturar su web con consultas, nuestra idea es hacer una extracción de datos semanal, con los datos de la última semana cada vez. De este modo no saturaremos su servidor, además no se detectan ondas gravitacionales a diario, por lo que el intervalo semanal es lo mejor para todos.

7. Inspiración

Lo que nos ha hecho tomar la decisión de obtener este conjunto de datos es conocer más sobre las detecciones de ondas gravitacionales. Cada vez que el consorcio formado por LIGO, VIRGO y KAGRA publican una detección, ésta se convierte en un acontecimiento en medios y nos surgen muchas preguntas:

- ¿Qué intervalos de masas de objetos son los más detectados?
- ¿Hay periodos del año donde haya más probabilidad de detecciones? Si es así ¿De qué región del espacio provienen?
- ¿Qué hace que una señal se considere buena o se descarte?
- ¿Cuáles son las detecciones más cercanas? ¿Y las más lejanas?

Estos son ejemplos de preguntas a las que nos gustaría dar respuesta. Lo positivo es que los datos que necesitamos para ello están en partes de la web de acceso permitido, por lo tanto es viable obtenerlos.

8. Licencia

Hemos decidido compartir el dataset bajo la licencia CC BY 4.0. Esta licencia consiste en:

- Permisos:
 - Permite compartir y distribuir los datos en cualquier medio.
 - Permite remezclar y transformar los datos.
- Restricciones:
 - Debes dar crédito a los autores, proveer un enlace a la licencia e indicar si has

hecho cambios sobre los datos originales.

Se pueden conocer los detalles de esta licencia aquí: <https://creativecommons.org/licenses/by/4.0/> (<https://creativecommons.org/licenses/by/4.0/>).

El motivo es que la licencia de los datos originales de la web es CC BY 4.0 y creemos que es ético que los compartamos con el mismo tipo de licencia. Además queremos que todo el mundo pueda usar nuestro dataset, pudiendo compartirlo y adaptar los datos a sus necesidades (coger las columnas que quieran y descartar las que no, por ejemplo). La idea es que siempre que lo usen den atribución al consorcio LIGO, VIRGO y KAGRA, pues son el autentico origen de datos.

9. Código

El código de nuestro proyecto se encuentra en el siguiente enlace: <https://github.com/SRobiscoUOC/Practica1> (<https://github.com/SRobiscoUOC/Practica1>). Se ha dividido la aplicación en un programa principal, llamado scraper.py, y otro archivo con las funciones requeridas para el correcto funcionamiento de la aplicación, llamado scraper_functions.py. Veamos su contenido:

- **Scraper.py**: contiene el programa principal que recibe como parámetros Fecha_inicio y Fecha_fin. Con estos parámetros invoca a la función execute_form, situada en scraper_functions.py, y rellena el fichero csv con el dataset resultante.
- **scraper_functions.py**: contiene la función execute_form que accede a la web <https://www.gw-openscience.org/> (<https://www.gw-openscience.org/>) empleando Selenium, esta función va navegando por los menús de la web hasta encontrar el formulario para generar la tabla de datos. Una vez ahí, rellena el formulario y pulsa el botón commit generando la tabla. Podría parecer que al generar la tabla sólo habría que recorrerla pero no es así: al trabajar con la pantalla de Chrome minimizada (hemos usado Chrome debido a que firefox no permite hacer webscrapping en sus últimas versiones) no mostraba la totalidad de las columnas. Esto ha hecho que, tras generar la tabla esperemos unos segundos y pulsemos un botón que abre el desplegable del selector de columnas, una vez abierto hemos tomado los elementos del selector y, mediante una función que hemos hecho para comprobar si el elemento estaba sin seleccionar y seleccionarlo en ese caso, hemos dejado todos marcados, quedando la tabla completa y lista para recorrerla. Otro problema encontrado han sido los datos: había datos con espacios que hemos tenido que tratar, pero lo peor ha sido la presencia de subíndices y superíndices que ha habido que tratar, dando lugar a nuevas columnas con los márgenes de error. Ha sido todo un reto al que ha habido que añadir que la mayor parte de los elementos de la web carecían de id.

Además de las dificultades descritas, debemos añadir que al usar Chromedriver con Selenium nos encontramos unos extraños errores de lectura de USB que no tenían nada que ver con nuestro código. Tras buscar documentación lo solucionamos añadiendo el siguiente código en la declaración de nuestro navegador:

```
In [ ]: ▶ options = webdriver.ChromeOptions()
        options.add_experimental_option('excludeSwitches', ['enable-logging'])
        driver = webdriver.Chrome(options=options)
```

Con esto el programa se ejecuta de manera fluida sin lanzar errores. Es muy importante

que lo lancemos desde la carpeta source, pues tiene todo direccionado a ese directorio raíz. Para ejecutar la aplicación lo haremos del siguiente modo: *python scraper.py --Fecha_inicio YYYYMMdd --Fecha_fin YYYYMMdd*. Un ejemplo de ejecución sería la siguiente:

```
python scraper.py --Fecha_inicio 20150101 --Fecha_fin 20221107
```

Con estas fechas aseguramos tomar desde la primera detección de ondas gravitacionales, que tuvo lugar el día 14/09/2015. Damos un margen porque queremos obtener también las detecciones descartadas.

En la carpeta source hemos añadido el archivo requirements.txt con las librerías requeridas tanto para la elaboración de la práctica como para la creación de este documento. Su contenido es el siguiente:

appdirs==1.4.4

async-generator==1.10

brotilpy==0.7.0

builtwith==1.3.4

certifi @ file:///C:/b/abs_ac29jvt43w/croot/certifi_1665076682579/work/certifi

cfffi @ file:///C:/Windows/Temp/abs_6808y9x40v/croots/recipe/cfffi_1659598653989/work

click==8.1.3

colorama @ file:///C:/Windows/TEMP/abs_9439aeb1-0254-449a-96f7-33ab5eb17fc8apleb4yn/croots/recipe/colorama_1657009099097/work

cryptography @ file:///C:/b/abs_36x9ifdcl4/croot/cryptography_1665612655344/work

exceptiongroup==1.0.1

fastprogress @ file:///tmp/build/80754af9/fastprogress_1634699150215/work

future==0.18.2

h11==0.14.0

idna @ file:///C:/b/abs_bdhbebrioa/croot/idna_1666125572046/work

importlib-metadata==5.0.0

itsdangerous==2.1.2

Jinja2==3.0.3

mkl-fft==1.3.1

mkl-random @ file:///C:/ci/mkl_random_1626186184308/work

mkl-service==2.4.0

numpy @ file:///C:/b/abs_53f_dbvhzc/croot/numpy_and_numpy_base_1665773185489/work

outcome==1.2.0

pycparser @ file:///tmp/build/80754af9/pycparser_1636541352034/work

pyee==8.2.2

pyOpenSSL @ file:///opt/conda/conda-bld/pyopenssl_1643788558760/work

pyppeteer==1.0.2

PySocks @ file:///C:/ci/pysocks_1605307512533/work

python-whois==0.8.0

selenium==4.6.0

six @ file:///tmp/build/80754af9/six_1644875935023/work

sniffio==1.3.0

sortedcontainers==2.4.0

tqdm @ file:///C:/b/abs_0axbz66qik/croots/recipe/tqdm_1664392691071/work

trio==0.22.0

trio-websocket==0.9.2

urllib3 @ file:///C:/b/abs_a8_3vfznn_/croot/urllib3_1666298943664/work

websockets==10.4

Werkzeug==2.2.2

win-inet-pton @ file:///C:/ci/win_inet_pton_1605306162074/work

wincertstore==0.2

wsproto==1.2.0

Dataset

Nuestro dataset se ha publicado en Zenodo con el siguiente enlace:

Vídeo

El vídeo explicativo se encuentra en el siguiente enlace