



Máster Ciencia de Datos – *Data Science*

Tipología y ciclo de vida de los datos / Profesor colaborador: Diego Perez

Fernando Antonio Barbeiro Campos – fbarbeiro@uoc.edu

PEC1: Preliminares

Entrega prevista: 08 de Octubre de 2018.

Índice General

Índice General	1
Introducción	2
Ejercicio 1	3
Ejercicio 2	7
Bibliografía	9

Introducción

Presentación

En esta Prueba de Evaluación Continuada se trabajan los conceptos generales de cuál es el ciclo de vida de los datos, y se identifican y conocen sus características. También se trabajan los conceptos esenciales de *Web Scraping*.

Competencias

En esta PEC se desarrollan las siguientes competencias del Máster de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento, almacenamiento y administración de datos.

Objetivos

Los objetivos concretos de esta Prueba de Evaluación Continuada son:

- Conocer el ciclo de vida de los datos y los principales tipos de datos.
- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinarios.
- Desarrollar las habilidades de aprendizaje que permitan continuar estudiando de una manera que tendrá que ser en gran medida autodirigida o autónoma.
- Desarrollar la capacidad de busca, gestión y uso de información y recursos en el ámbito de la ciencia de datos.
- Entender la utilidad, la legalidad y algunas características de web scraping.

Ejercicio 1

Enunciado

Después de leer el recurso “Fundamentos de Data Science” contesta las siguientes preguntas:

1. *Explique los tres tipos de datos y proponga un ejemplo para cada uno sin utilizar los ejemplos ya planteados en el recurso (máximo 200 palabras).*

Respuesta:

- a. **Simples:** Esencialmente un dato atómico con significado propio. Por ejemplo: Supongamos un sensor de temperatura posicionado en la cumbre de una montaña define una temperatura para una fecha específica.



Figura 1: Sensor recolectando dato simples.

- b. **Compuestos / Estructurados:** Cuando reunidos datos simples (o mismo otros datos compuestos) en una estructura fija, entonces tenemos los datos estructurados. Por ejemplo: siguiendo con la línea del ejemplo anterior del sensor de temperatura, si reunimos las temperaturas de muchos días en sitios distintos y después agregamos todo a un fichero CSV.

	dt	# AverageTemperatu	# AverageTemperatu	City	Country	Latitu
1	1849-01-01	26.704	1.435	Abidjan	Côte D'Ivoire	5.63N
2	1849-02-01	27.434	1.362	Abidjan	Côte D'Ivoire	5.63N
3	1849-03-01	28.101	1.612	Abidjan	Côte D'Ivoire	5.63N
4	1849-04-01	26.14	1.386999999999998	Abidjan	Côte D'Ivoire	5.63N
5	1849-05-01	25.427	1.2	Abidjan	Côte D'Ivoire	5.63N
6	1849-06-01	24.844	1.402	Abidjan	Côte D'Ivoire	5.63N
7	1849-07-01	24.058000000000003	1.254	Abidjan	Côte D'Ivoire	5.63N
8	1849-08-01	23.576	1.265	Abidjan	Côte D'Ivoire	5.63N
9	1849-09-01	23.662	1.226	Abidjan	Côte D'Ivoire	5.63N
10	1849-10-01	25.263	1.175	Abidjan	Côte D'Ivoire	5.63N
11	1849-11-01	26.331999999999997	1.507	Abidjan	Côte D'Ivoire	5.63N
12	1849-12-01	25.45	1.838	Abidjan	Côte D'Ivoire	5.63N
13	1850-01-01	25.803	1.943	Abidjan	Côte D'Ivoire	5.63N
14	1850-02-01	27.89	1.43	Abidjan	Côte D'Ivoire	5.63N

Figura 2: Fichero de datos estructurados CSV. Fuente: ¹ Kaggle, 2018.

- c. **Semiestructurados / No estructurados:** Pueden seguir una estructura parcial (que tiene la flexibilidad de cambiar según el contexto) o ni eso, es decir, pueden no tener ninguna estructura.

Ejemplo:

Supongamos que estamos construyendo un servicio de stream de videos como Netflix. Las peticiones de búsqueda de títulos pueden cambiar según la cuenta de la persona conectada, es decir, los eventuales **JSONs** (que son datos **semiestructurados**) con la información de que tenemos disponibles para la persona, cambia por usuario. Por su vez, cuando la persona elige una película, el servicio de stream enviará bytes datos no estructurados.

2. *Con la ayuda de un ejemplo real, explique cada una de las fases del ciclo de vida de los datos necesarias para la resolución de un cierto problema (máximo 250 palabras).*

Respuesta:

Captura: Manera con que los datos son creados, extraídos o encontrados. Ejemplo: supongamos un dato simples como la figura 1, un sensor recoleta datos de temperatura en horas y fechas diferentes de una determinada localidad. A

¹ "Climate Change: Earth Surface Temperature Data | Kaggle."

<https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data>. Accessed 1 Oct. 2018.

posteriori, los datos del ejemplo seguirán el ciclo de vida.

Almacenamiento: Estamos trabajando con datos más rudimentarios de sensores, la parte de almacenamiento sería la agregación de distintos sensores en un formato como un CSV o tabla en una BBDD.

En escenarios más avanzados, suelen haber *datawarehouses*, *datamarts* (o *datalakes*) que ahora empiezan a moverse también a cloud con soluciones como *BigQuery* de Google.

Preprocesado: Ni todos los datos están totalmente completos y perfectos. Quizás tendremos que hacer merge de ficheros CSV; quizás hay demasiados datos (hay que filtrar); podríamos tener que limpiar los datos (eliminando NAs, erróneos) y también convertir formatos adecuados (como de fechas).

Análisis: Objetiva la creación de modelo(s) que explique(n) estos datos – aquí empieza la parte de aplicación de estadística, saber si se trata de un modelo supervisado o no supervisado y así por delante. Si quisiéramos, por ejemplo, descubrir más información de una muestra y crear un cluster de regiones (modelo de agregación) segundo las temperaturas.

Visualización: Presentar la información para humanos de manera que se pueda leer bien y comprenderlas. Ej.: Gráficos o mapas que enseñan los cluster diagnosticados en la análisis.

Publicación: Fase de visando garantizar la preservación y diseminación del conocimiento adquirido. Sería por ejemplo la publicación del conocimiento que la investigación de grupos de regiones por temperatura ha sacado.

3. *En la fase de captura del ciclo de vida de los datos, ¿estos siempre pueden ser creados?. Explique los dos principales mecanismos para la captura de datos y desarrolle un ejemplo para cada uno de los mecanismos (máximo 200 palabras).*

Respuesta:

No, ni siempre pueden ser creados. Por ejemplo, hay momentos que hay momentos que es necesario capturar datos conforme los encontramos. Dicho este argumento soporta un de los dos mecanismos básicos complementarios de captura, la extracción (el otro es precisamente la creación).

Ejemplos:

Creación: Empresa de venta de vuelos hace una venta de un itinerario de Barcelona a Madrid (toda la información del vuelo / pasajero está disponible para ser capturada y almacenada en su base de datos).

Extracción: La misma empresa quiere hacer seguimiento para garantizar la satisfacción del cliente y se suscribe en una API de las aerolíneas para extraer información de los vuelos del pasajero en cuestión y mirar si no hubo retrasos y cosas del tipo. Otra aproximación aun en la parte de extracción sería capturar el contenido que el mismo cliente publica en sus redes sociales y sacar información si hubo algún tipo de quejas sobre el servicio prestado.

Ejercicio 2

Enunciado

Después de leer el Capítulo 1 “Introduction to Web Scraping” del recurso “Web Scraping with Python” contesta las siguientes preguntas:

1. *¿Cuándo es legal utilizar web scraping y cuando no?. Explica con un ejemplo para cada caso (máximo 100 palabras).*

Respuesta:

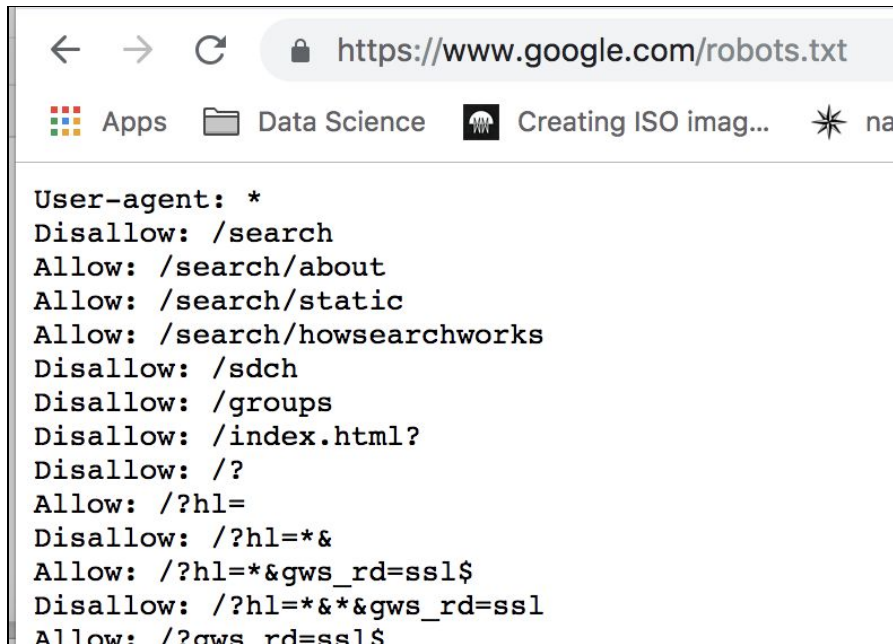
Como el tema de *web scraping* es relativamente nuevo, aún están estableciendo criterios de que es permitido con tal técnica. Lawson (2015) [2] menciona que casos de juzgado alrededor del mundo donde en la medida que se hizo scraping de hechos (como una lista de teléfonos) no había problemas. Mientras tanto otro caso en Australia demostró que datos con un autor identificable puede tener derechos del propio autor.

Desde mi experiencia personal, mi empresa actual sufrió una denuncia por hacerlo simplemente porque la *webpage* no quería ver sus productos anunciados en otro sitio que no fuera el suyo.

2. *¿Qué son los robots.txt? (máximo 50 palabras).*

Respuesta:

Robots.txt es un fichero de texto creado con intuito de instruir a webcrawler sobre cómo rastrear páginas en su website. Además de instruir, hace explícitas las restricciones (meramente sugerencias, pero recomendable seguirlas) sobre ejecutar crawlers en dicha página. Verificando el fichero antes de hacer scraping disminuirá el riesgo de ser bloqueados.



```
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
Allow: /?gws_rd=ssl$
```

Figura 3: Robots.txt de [www.google.com](https://www.google.com/robots.txt)

3. *¿Cual es la ventaja de establecer un user agent ? (máximo 50 palabras).*

Respuesta:

Hay dos ventajas principales:

- a. Si dejamos el *user-agent* por defecto, algunas páginas pueden tenerlo bloqueado (obviamente, después de una posible mala experiencia con scraping).
- b. En caso de un problema con el crawler, permitimos que los dueños del site sepan que está ocurriendo y que crawler está causando el problema.

4. *¿En qué consiste el Throttling download? (máximo 50 palabras).*

Respuesta:

Throttling download está específicamente relacionado con la velocidad que nuestro crawler “ataca” peticiones a un website en concreto pudiendo causar daños como la sobrecarga de un servidor. Escusado será dizer que hay que implementar mecanismos para causar un espera/*delay* entre las peticiones y evitar el riesgo de ser bloqueado por la página a la cual estamos haciendo scraping.

Bibliografía

[1] **Kaggle (2018)**. "*Climate Change: Earth Surface Temperature Data*" [artículo en línea].
[Fecha de consulta: 01 de octubre del 2018].
<<https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data>>

[2] **Lawson, R. (2015)**. "*Web Scraping with Python*" - Packt Publishing. ISBN 978-1-78216-436-4