Universitat Oberta de Catalunya

Master Ciencia de Datos

Web Scraping sobre Amazon.

Dataset: *Price tracking* sobre la oferta de ordenadores y tablets

---

Irene López Ruiz

Javier Samir Rey

Índice

1. Contexto

El comercio *on-line* ha supuesto un incremento exponencial en la oferta de productos. Con tantas opciones y con la constante actualización por parte de los vendedores en un intento de adaptarse a las necesidades de los clientes por encima de la competencia, el *price tracking* se ha convertido en una herramienta muy utilizada tanto por empresas como por particulares.

El *price tracking* es una técnica de rastreo, comparación y análisis de los precios de productos ofertados por diferentes vendedores en uno o más sitios web. *Price tracking* tiene dos posibles aplicaciones bien diferenciadas según quien emplee la técnica (de Wit, 2020): en primer lugar, el rastreo de precios por parte de una empresa se utiliza como un método de análisis de la competencia en aras de establecer una estrategia de precios óptima; en segundo lugar, el rastreo de un particular, más comúnmente denominado *price comparision* o comparación de precios en lugar de *price tracking*, va orientado a que un comprador individual encuentre la mejor opción en base a las características que éste demande y al precio del producto.

El sector de la tecnología, y en concreto de los ordenadores y tablets, presenta grandes fluctuaciones de precio debido a la gran variedad de prestaciones y de vendedores que estos productos pueden poseer. Por tanto, en este documento se presenta una descripción de un proyecto que pretende crear una herramienta que facilite el *web scraping* para recoger, procesar y almacenar en un *dataset* ordenado datos que permitan realizar un posterior análisis de *price tracking* sobre la oferta de ordenadores y tablets en Amazon.

Se ha seleccionado a Amazon como sitio web fuente de los datos por la gran oferta de productos que posee con diferentes características, vendedores, precios y valoraciones, características clave del *dataset* para el rastreo de precios.

2. Título Dataset

Teniendo en cuenta el contexto anterior, el título del *dataset* será “Price tracking sobre la oferta de ordenadores y tablets en Amazon”.

3. Descripción Dataset

Tal como se ha mencionado anteriormente, el *dataset* generado como resultado del proyecto de *web scraping* consta de diferentes registros de ofertas de ordenadores y tablets en Amazon. Para cada uno de estos registros se disponen de datos relevantes para realizar un análisis de *price tracking* sobre estos productos, como puedan ser el precio, la valoración media, o especificaciones técnicas como tamaño de pantalla, memoria o sistema operativo, entre otros.

Debido a la inmensa cantidad de ofertas que contiene Amazon sobre ordenadores y tablets, el *scraping* se ha limitado por defecto a los productos de las primeras 50 páginas del listado de ofertas que devuelve Amazon al acceder a los departamentos “Computadoras -> Computadoras y Tablets” desde las compras para Estados Unidos, aunque este límite de páginas es configurable por el usuario. Podemos acceder a este listado de resultados con el siguiente enlace genérico, que se encuentra almacenado en un archivo CSV en la carpeta `/input_data` para que sea accesible al código de *web scraping*:

https://www.amazon.com/s?i=specialty-aps&bbn=16225007011&rh=n%3A16225007011%2Cn%3A13896617011&language=es&ref=nav_em__nav_desktop_sa_intl_computers_tablets_0_2_6_4

Como cabría esperar de una empresa como Amazon, la tasa de actualización de las ofertas es muy elevada, lo cual supone un factor importante a la hora de realizar *Price tracking*. Por ello, el sistema creado para el *web scraping* de Amazon almacenará los resultados en carpetas diferentes del directorio de resultados según la fecha en la que se obtuvieron. De esta manera, los resultados de un análisis realizado, por ejemplo, el 26 de marzo de 2021 se almacenarán en el directorio `\output_data\26_03_2021\`.

4. Representación gráfica

La siguiente imagen representa la naturaleza el proyecto de *web scraping* a desarrollar:

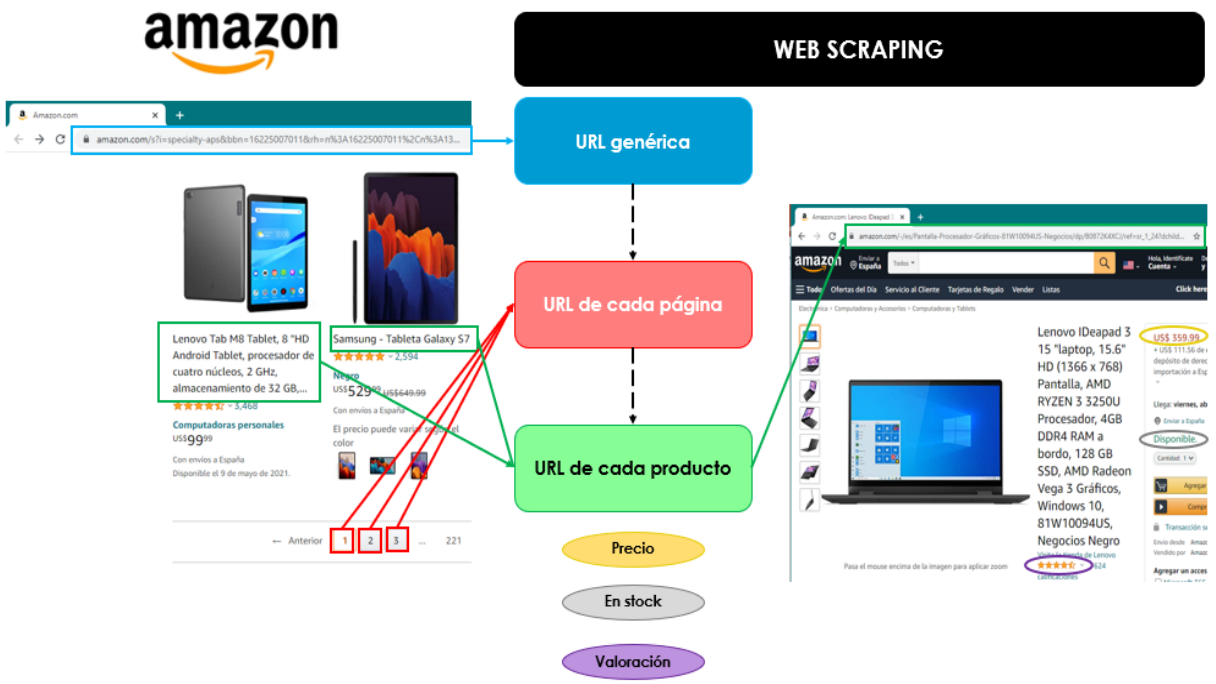

Figura 1. Esquema representativo del proyecto.

Como se presenta en el esquema, la rutina de *scraping* comienza desde una URL genérica sobre la oferta de ordenadores y tablets que debe ser proporcionada por el usuario. Partiendo de ésta, se obtienen las URL de las páginas del listado de productos, hasta un cierto límite de páginas (por defecto, 50). Desde la URL de cada página se obtiene la URL de cada producto que aparece en ella. Para todas las URL de producto encontradas se extraerán datos de interés para el *Price tracking* como puedan ser el precio, la disponibilidad o la valoración media, entre otros, y se almacenarán en un *dataset* en formato CSV.

5. Contenido

En el *dataset* adjuntado, realizado con las configuraciones por defecto del código, cuenta con un total de 1177 registros, resultado de los productos ofertados en las primeras 50 páginas que Amazon ofrece para productos dentro de la categoría “Computadoras y Tablets” de su página de mercado en EEUU. Para cada uno de ellos se presentan las siguientes variables:

- **product_id**: número ASIN (Amazon Standard Identification Number) del producto. Se trata de una combinación de números y letras que Amazon otorga a sus productos para identificarlos(Axinte, s. f.).

- **url**: URL de la oferta en Amazon. Se habla, por tanto, de una cadena de caracteres.
- **product_title**: cadena de caracteres que contiene el título completo del producto.
- **product_brand**: cadena de caracteres con la marca del ordenador o Tablet.
- **product_rating**: valoración numérica promedio otorgada por los usuarios compradores del producto, siendo 1 la peor puntuación y 5 el máximo. Se trata de números decimales que utilizan el punto como separador.
- **product_rating_count**: número entero de valoraciones realizadas por compradores del producto. Es importante saber que utiliza comas como separador de miles.
- **product_image**: cadena de caracteres con la URL de la imagen principal de la oferta.
- **stock_status**: variable dicotómica con dos posibles valores: "in stock" o "out of stock", indicando si el producto se encuentra en stock o no, respectivamente.
- **vendor_price**: cadena de caracteres con el precio del producto y sus unidades, lo más común son dólares.
- **product_os**: sistema operativo del portátil, indicado como cadena de caracteres.
- **screen_size**: cadena de caracteres con el tamaño de la pantalla y las unidades de medida de la misma, normalmente pulgadas.
- **memory_size**: capacidad de memoria del ordenador o Tablet. Queda indicada como una cadena de caracteres con el valor y las unidades de medida.
- **question_answered**: número de preguntas contestadas por el fabricante. Los productos con más de 1000 respuestas quedan indicados por el valor "1000+".

6. Agradecimientos

Los datos se han extraído de Amazon. Más específicamente, del listado de productos que presenta en la categoría "Computadoras y Tablets" de su mercado en EEUU.

Cabe destacar que Amazon no se deja rastrear fácilmente. Para *scraping* de muchos productos, como ocurre en este proyecto, el bot de Amazon detecta el rastreo y bloquea el acceso. La forma de evitar ser detectados pasa por usar una configuración de proxies que cambie dinámicamente la IP origen de la petición, de manera que Amazon crea que las peticiones realizadas proceden de usuarios distintos. ScrapeOwl (https://scrapeowl.com/) ofrece estos servicios a sus usuarios Premium por lo que, este proyecto se vale de ScrapeOwl para poder realizar el *scraping* usando las credenciales de un usuario con estos privilegios.

7. Inspiración

Tal como se mencionaba en el apartado Contexto de este documento, el *Price tracking* posee múltiples aplicaciones, tanto de análisis de la competencia para una empresa como de análisis de precio para un particular interesado en comprar uno de estos productos. Con este *dataset* se podrán realizar análisis para determinar los productos con mejor relación calidad-precio, las marcas que más satisfacen a los clientes o qué modelos son los más vendidos de cada marca y por qué.

Por otro lado, como ya se ha mencionado, el *scraping* a Amazon no resulta fácil. Es por esto que, aunque abundan las plataformas de pago para realizar un rastreo de precios automático, son poco frecuentes los *datasets* públicos con este tipo de información. Por tanto, se pretendía construir desde cero un conjunto de datos con información relativa a diferentes características, y no solo al precio, para así permitir un posterior análisis de *Price tracking* completamente configurable por el usuario.

8. Licencia

El *dataset* adjuntado a este documento posee una licencia CC-BY-SA (Creative Commons Attribution-ShareAlike), lo cual implica (*License Selector*, s. f.):

- Se atribuye la autoría de dicho conjunto de datos a los autores del proyecto.
- Se posibilita el uso de los datos para realizar otros trabajos derivados o basados en los mismos siempre que dichos trabajos se compartan con licencias compatibles a la presente.
- Se permite el uso de estos datos con fines comerciales.

9. Código

Este proyecto de *web scraping* se ha desarrollado en el lenguaje de programación Python. El código está disponible en el repositorio https://github.com/jasam/ciclo_vida_datos_scraping

Encontramos tres archivos `.py` para realizar el proyecto: `helper_class.py`, `interface_class.py` y `main.py`.

Además, existe un archivo muy relevante para el proyecto: `config.json`. Dicho archivo contiene aspectos de configuración para realizar el *scraping* que pueden ser modificados por el usuario:

- **output_data_folder**: ruta de la carpeta donde guardar los resultados del *scraping*. En ella se crearán las diferentes carpetas correspondientes a las fechas de cada extracción de datos. Por defecto, será la carpeta `/output_data`.
- **current_marketplace**: dominio de Amazon sobre el que realizar el *scraping*. Su valor es "com".
- **input_urls_file**: ruta del archivo donde se encuentra la URL que dirige al listado genérico de productos, desde la que comenzar a hacer el *scraping*. Como ya se comentó, esta URL se encuentra en el archivo `/input_data/urls_to_scrape.csv`.
- **page_limit**: número de páginas de resultados de Amazon sobre las que realizar *scraping*. El límite por defecto, como ya se mencionó, es de 50 páginas.

El archivo `helper_class.py` programa la clase `Helper()`, que incluye una serie de métodos auxiliares que serán usados por el programa principal. Entre estos métodos se contemplan: lectura y escritura de archivo CSV, escritura del *dataset* en formato CSV, obtener fecha actual, lectura y escritura de archivo JSON, comprobación de la existencia de carpetas y ficheros y listado de directorios.

El archivo `interface_class.py` programa la clase `INTERFACING()`, que define los métodos necesarios para realizar el *scraping* de un producto concreto a partir de su URL específica. En concreto, define métodos para comprobar la disponibilidad de la URL que se desea consultar, para obtener la respuesta en HTML de la URL realizando una petición desde ScrapeOwl para evitar ser detectados por Amazon, para espaciar en el tiempo estas peticiones de modo que no se sature el servidor (Subirats Maté & Calvo González, 2019), y para parsear el contenido HTML en un objeto `Soup`, propio de la librería `BeautifulSoup` (*Beautiful Soup Documentation — Beautiful Soup 4.9.0 documentation*, s. f.).

Por último, el archivo `main.py` programa la clase `AMAZONCLASS()` que se vale de las dos clases anteriores para realizar todo el *scraping* e inicia todo el proceso. Esta clase cuenta, entre otros, con los dos métodos principales que permiten construir el *dataset* buscado:

- `start_scraping_products()`. Partiendo de la URL genérica, del límite de páginas y de la carpeta destino de los resultados (todo indicado en el archivo `config.json`), guarda todas las URL de las páginas de resultados hasta alcanzar el límite de páginas en un fichero `processed.json`, un archivo creado dentro de la carpeta `\log` del directorio de resultados, también indicada en el archivo `config.json`. Este método, apoyándose en otro de la misma clase llamado `get_search_results()`, guarda, para cada una de las páginas almacenadas en el fichero `processed.json`, el número de dicha página y la URL de cada uno de los productos que contiene en el archivo `scraped_urls_data.csv`, ubicado en la carpeta de resultados.
- `start_scraping_each_product_details()`. Una vez se tienen en el archivo `scraped_urls_data.csv` las URL de todos los productos que quedarán registrados en el *dataset*, se lee dicho archivo. Para cada una de las URL de producto que contiene llama al método `get_product_information()`, que extrae el valor de los campos definidos para el *dataset* y los guarda en el fichero `product_complete_data.csv` de la carpeta de resultados. Por último, escribe la URL del producto procesado en el fichero `products_rocessed.json` de la carpeta `\log` dentro de la carpeta de resultados, como un registro de las páginas que hayan podido procesarse respecto al total, ya que será común que alguna página no esté disponible por algún motivo.

Finalmente, para simplificar la ejecución de los *scripts* se ha creado un archivo Batch (*How to Create a Batch File to Run Python Script*, 2020) que se encargue de esta tarea. Este archivo, `start.bat`, contiene una llamada al ejecutor de Python, que se encuentra dentro de la carpeta `\python` del proyecto, a la cual no se debe de modificar el nombre; y al archivo principal del proyecto `main.py`. Por tanto, con hacer doble click sobre `start.bat` se iniciará la ejecución y se abrirá una consola del sistema sobre la que se irá informando del progreso del proceso de *scraping* que se está realizando.

10. Dataset

El *dataset* creado durante el desarrollo del proyecto se encuentra en formato CSV en la carpeta `\output_data` del repositorio GitHub https://github.com/jasam/ciclo_vida_datos_scraping

DOI en Zenodo

11. Bibliografía

Axinte, M. (s. f.). *Número Amazon ASIN: ¿Qué es y cómo obtenerlo?* DataFeedWatch. Recuperado 26 de marzo de 2021, de https://www.datafeedwatch.es/blog/número-amazon-asin-qué-es-y-cómo-obtenerlo

Beautiful Soup Documentation — Beautiful Soup 4.9.0 documentation. (s. f.). Recuperado 26 de marzo de 2021, de https://www.crummy.com/software/BeautifulSoup/bs4/doc/

de Wit, J. (2020, julio 15). *What is a price tracker and how does it work?* . Pricesearch. https://www.pricesearch.io/what-is-a-price-tracker-and-how-does-it-work/

How to Create a Batch File to Run Python Script. (2020, febrero 24). Data to Fish. https://datatofish.com/batch-python-script/

License Selector. (s. f.). Recuperado 27 de marzo de 2021, de https://ufal.github.io/public-license-selector/

Subirats Maté, L., & Calvo González, M. (2019). *Web Scraping* (pp. 1-66). Universitat Oberta de Catalunya.

ANEXO: Contribuciones

Contribuciones	Firma
Investigación previa	ILR JSR
Redacción de las respuestas	ILR JSR
Desarrollo de código	ILR JSR