Tipología y ciclo de vida de los datos

Práctica 1: ¿Cómo podemos capturar los datos de la web?

Raúl García Díaz | email: raulgd@uoc.edu
Juan Luis Andión Tápiz | email: jandion@uoc.edu

25 de abril de 2023

Índice de Contenidos

1. Contexto **2**

2. Título **2**

3. Descripción del dataset **2**

4. Representación gráfica **3**

5. Contenido **3**

6. Propietario **3**

7. Inspiración **4**

8. Licencia **4**

9. Código **5**
9.1. ¿Cuál es el contenido al que se le va a realizar el raspado de información? 5
9.2. ¿Como se evita ser bloqueado y pasar desapercibido? 5
9.3. Si dejan de funcionar los proxys, ¿cómo se renuevan? 6
9.4. Tantas peticiones ralentizará el programa, ¿cómo se solventa? 6
9.5. ¿Por qué solamente busca hasta 10000 películas o series? 6
9.6. Ubicación del repositorio 6

10. Dataset **7**

11. Vídeo **7**

12. Firma **7**

1. Contexto

La industria cinematográfica es una de las más importantes y rentables del mundo, generando una cantidad impresionante de más de 100 millones de dólares anuales. Esta cifra incluye tanto la venta de entradas en cines tradicionales como el uso de servicios digitales de visualización de contenido audiovisual, que cada vez tienen más presencia en nuestra sociedad.

En este contexto, la plataforma en línea IMDb ha logrado consolidarse como la página de cine más popular del mundo. En ella, los usuarios pueden encontrar una amplia selección de películas, series y documentales, así como leer críticas y opiniones de otros usuarios y acceder a información detallada sobre cada producción, como el reparto, la duración, el género, la dirección y muchos otros detalles.

El proceso de recopilación de información sería inviable si se hiciera manualmente. Es por eso que hemos decidido utilizar técnicas de web scraping para recopilar información valiosa de IMDb de manera automatizada. Con la ayuda de esta técnica, podemos extraer información detallada de una manera más rápida y precisa.

Esta práctica tiene como objetivo recopilar información valiosa de IMDb , esturarla y almacenarla en un fichero de datos que pueda ser utilizado parar llevar a cabo análisis sobre la industria cinematográfica. El conjunto de datos generados podrá ser empleado para el desarrollo de cuadros de mando que permitan tener una visión general del mercado del cine y el entretenimiento audiovisual [1].

2. Título

El título que recibirá el fichero generado, deberá tanto reflejar el contenido del dataset, como la propiedad de los datos recolectados, para que cualquier otra persona que quiera emplearlos para su análisis sepa la procedencia y fuente de los datos. El formato de archivo deberá ser globalmente reconocido y se priorizará la sencillez de su estructura y entendimiento.

Por lo tanto, el fichero generado es: IMDb_data.csv

3. Descripción del dataset

El código permite elegir el número de películas o series a descargar, hasta un máximo de 10000 películas y también es necesario especificar el género del contenido que se quieren almacenar hasta un máximo de diez mil. El dataset publicado contiene información sobre las 10000 películas mejor valoradas de comedia de IMDb, como el nombre, año de estreno, género(s), directores, etc. Hay que mencionar que este dataset no ha sido depurado y se publica tal y como lo ha generado el código.

4. Representación gráfica

Por explicar nuestro proyecto de forma más visual, nos apoyamos en el siguiente esquema:

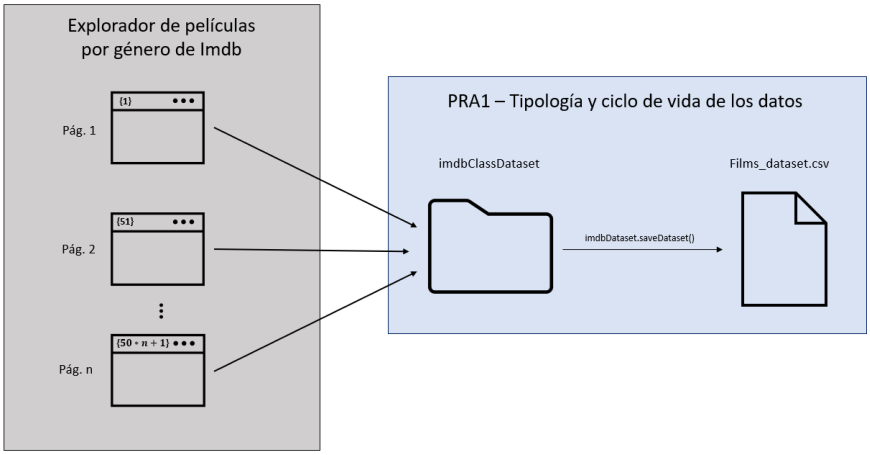

Figura 1: Esquema del proyecto

Nuestro paquete recorre los resultados de películas que se muestran en el explorador de películas por género de IMDb. Cada página tiene 50 resultados y esto queda indicado en la url de la web, así que aprovechamos este patrón para raspar toda la información de cada resultado y accediendo a cada uno de ellos para conseguir más datos. Finalmente el paquete genera el dataset con el método saveDataset()

5. Contenido

Los datos han sido recolectados el 19/04/2023 de la página de IMDb, por lo que podemos encontrar tanto películas recientes como clásicos del cine. A continuación, nombre y descripción de cada campo del dataset:

- **NameContent:** nombre de la película.
- **ReleaseYear:** Año de estreno.
- **Certificate:** Público para el que está dirigido.
- **TimeContent:** Duración.
- **Genres:** Género(s).
- **RatingImdb:** Puntuación en Imdb.
- **RatingMetacritic:** Puntuación en Metacritic.
- **Casting**: Reparto.
- **Directors**: Directores.
- **Writers**: Escritores.

6. Propietario

IMDb tiene a disposición de los usuarios un acceso a varios ficheros comprimidos con información sobre los actores, directores y títulos de parte de las películas registradas en la web, también en Kaggle podemos

encontrar varios datasets similares al que hemos creado con datos obtenidos de IMDb y también algunos centrados en otros aspectos con información sobre la sinopsis y críticas que se muestran en la web [2, 3, 4].

Además, existen varias plataformas que permiten acceder a cuadros de mando sobre la industria cinematográfica. Estos cuadros de mando pueden ser de gran utilidad, por ejemplo, a las productoras de cine para determinar el tipo de contenido que más está demandando el mercado o, en general, tener una vision global de la industria(cuadros de mando) [5, 6].

Para seguir unos principios éticos que se adecuaran al alcance del proyecto, hemos tenido presente durante el desarrollo del proyecto dos aspectos principales:

- Establecer tiempos de espera entre las distintas consultas y peticiones que se han ido haciendo para no saturar sus servidores
- Establecer un tamaño máximo del dataset cuya publicación no suponga un problema legal pero que también permita realizar análisis profundos

7. Inspiración

Nuestro dataset es lo suficientemente grande como para elaborar análisis de datos con fines académicos o de aprendizaje: podemos analizar la industria cinematográfica y sus tendencias de consumo a lo largo de la historia viendo, por ejemplo, la evolución del número de estrenos de películas para cada año, clasificándolas por género o intervalos de duración de la película, etc. Podemos encontrar numerosas implementaciones en python para raspar datos de IMDb empleando las librerías requests y beautifulspoup sobre las páginas del explorador de películas. Nuestra implementación presenta dos principales mejoras que repercuten directamente en la calidad del dataset:

- Nuestra implementación no solo obtiene datos de los resultados que se muestran de cada película en el buscador de IMDb, sino que además, accede a la enlace de cada película y obtiene información como los directores y reparto de cada película. Gracias a esta técnica, logramos añadir más variables de interés al dataset y ampliamos las posibilidades de análisis.
- Uso de múltiples direcciones Proxy para evitar ser rechazados por la gran cantidad de peticiones lanzadas a la web, pese a establecer tiempos de espera entre las peticiones. Para optimizar el proceso de descarga, se han implementado varios hilos para las peticiones. Esto nos ha permitido alcanzar un tamaño máximo para el dataset de 10000 observaciones con un bajo porcentaje de valores nulos.

Ejemplos de webs scrpaing:

- https://www.geeksforgeeks.org/scrape-imdb-movie-rating-and-details-using-python/
- https://www.freecodecamp.org/news/web-scraping-sci-fi-movies-from-imdb-with-python/

Recordamos que el fichero generado no ha sido depurado ni limpiado, por lo que una correcta limpieza de los datos será necesaria para extraer la mayor información del fichero, lo cual supone una oportunidad de aprendizaje en el la fase de limpieza de los datos y tratamiento de variables categóricas y temporales.

8. Licencia

La licencia escogida para este proyecto es CC BY-NC-SA 4.0 License.

Hemos elegido una de las licencias de Creative Commons por ser la más popular y reconocida para este tipo de proyectos, queremos que todo el mundo pueda acceder y compartir este proyecto, siempre y cuando se reconozca la autoría del mismo y que se haga sin fines comerciales y esto es precisamente lo que permite la licencia seleccionada

https://creativecommons.org/licenses/by-nc-sa/4.0/

9. Código

El eje principal de la parte del código del proyecto se basa en una clase llamada *ImdbDatSet*. Esta contiene numerosas funciones que se llaman unas a otras pero de uso interno. La forma de usar esta clase se basa en instanciarla, pasándole el tipo de contenido que se desea realizarle scraping, por ejemplo, si es una película y queremos que el género sea de comedia se tendrá que realizar lo siguiente:

```
our_class = ImdbDataSet(type_='movie', genre='comedy')
our_class()
```

Cabe destacar que en la segunda línea se llama al método *___call___()*. Este se encargará de realizar el proceso de scraping. Una vez iniciado, habrá que esperar una hora aproximadamente. Tras la ejecución, *ImdbDataSet* dispone de un método para guardar los datos en formato csv:

```
our_class.save_dataset()
```

Este método guardará los datos en el directorio donde se haya ejecutado el programa.

Tras esta breve explicación de cómo usar el programa, se van a responder a varias preguntas básicas que han ido surgiendo a lo largo del desarrollo del proyecto.

9.1. ¿Cuál es el contenido al que se le va a realizar el raspado de información?

Este contenido ha sido mencionado en el apartado 5. La mayor parte de este se encuentra en una url como la que se muestra a continuación:

- *https://www.imdb.com/search/title/?title_type=**title_type**&genres=**genre**&start=**num_page**&explore=title_type,genres&ref_=adv_nxt*

En esta habría que añadir el contenido al que se le quiere realizar el raspado, el género y el número con el que va a aparecer la primera película. Este último argumento tiene su particularidad, ya que tiene que ir en saltos de 50, empezando por el 1. Por ejemplo, si se decide realizar web scraping a las películas de comedia, empezando por la película 51, su url sería:

- https://www.imdb.com/search/title/?title_type=movie&genres=comedy&start=51&explore=title_type,genres&ref_=adv_nxt

Si se quisiera pasar a la siguiente página, habría que sumar 50 al 51 que se encuentra en la anterior url.

Por último, en esta url se encuentra la mayoría de la información, pero la lista de los actores es corta, además de que no aparece ni los directores ni los escritores. Para acceder a esta información, hay que dirigirse a la página web que hay exclusivamente de ese contenido.

9.2. ¿Como se evita ser bloqueado y pasar desapercibido?

Existen tres mecanismos que se han implementado en este proyecto:

- **cloudscraper[7]:** Esta librería proporciona ventajas frente a *request* para no ser detectado. Su funcionamiento es exactamente igual al de *request* pero con la particularidad de que hay que instanciar la clase.
- **proxies:** Se han usado proxies para que así estos realicen la petición por nosotros. Los proxies se han obtenido a través de la siguiente página web: free-proxy-list.net/
- **headers:** Se ha extraído las cabeceras que pide *IMDb* cuando se le realiza una petición y se han introducido en todas las peticiones que se van a realizar.

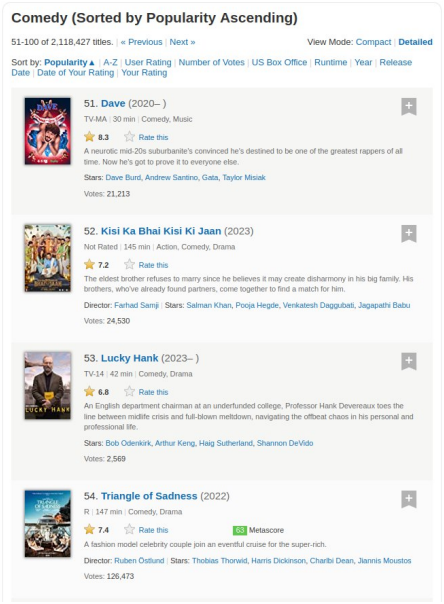

Figura 2: Ejemplo de como se muestra el contenido en la url mencionada.

9.3. Si dejan de funcionar los proxys, ¿cómo se renuevan?

Ha ocurrido que tras varias peticiones un proxy puede dejar de funcionar por numerosas razones. Para solventar este problema, se a propuesto una cola que almacena todos los servidores proxy que se van a utilizar. Estos se obtienen de la página web mencionada en el apartado 9.2, realizando a la página un raspado. Conforme se van utilizando, se desechan si no se ha conseguido una respuesta exitosa o se guardan si se ha finalizado la petición de forma exitosa. Cuando se queda vacía la cola se vuelve a realizar el raspado. Esta solución esta implementada en otro módulo a parte de la clase, llamado *load_proxy()*.

9.4. Tantas peticiones ralentizará el programa, ¿cómo se solventa?

Este es uno de los mayores problemas que se ha tenido debido a que la solución que nos proporciona los proxys, también implica otro problema: la lentitud. Muchos no funcionan o tardan mucho en proporcionar la respuesta. Para solventar este problema se ha propuesto la programación secuencial. Se ha hecho uso de hilos, para que así, mientras que una parte del código esta esperando la respuesta, otro este trabajando, reduciendo el tiempo de ejecución.

Es importante aclarar que estos hilos solamente se usan para las peticiones a las páginas web de las películas, no del buscador.

9.5. ¿Por qué solamente busca hasta 10000 películas o series?

Esto se debe a que, a partir de la página 100001, no te lleva a la url que debería. Si uno se dirige a la pagina con la película que empieza por 9951 y pasa de página de forma manual, se podrá observar que el patrón se pierde y genera una url totalmente diferente. Para solucionar este problema se podría rascar la *uri* que ofrece el botón de *next* pero no se ha implementado debido a que 10000 películas o series se considera ya suficiente como para generar un gran dataset para poder trabajar con él.

9.6. Ubicación del repositorio

El repositorio se encuentra en la siguiente url:
https://github.com/raulgdUOC/PRA1-Tipologia-y-ciclo-de-vida-de-los-datos

10. Dataset

El dataset está subido en la siguiente dirección: https://doi.org/10.5281/zenodo.7860478

11. Vídeo

Se adjunta un enlace a una carpeta de Google Drive con acceso al vídeo de presentación del proyecto: https://drive.google.com/file/d/1dhr3rs6oXTRZnyzyyDG9Uf66RgZ6Zd4B/view?usp=sharing

12. Firma

Contribuciones	Firma
Investigación previa	Raúl García Díaz, Juan Luis Andión Tápiz
Redacción de las respuestas	Raúl García Díaz, Juan Luis Andión Tápiz
Desarrollo del código	Raúl García Díaz, Juan Luis Andión Tápiz
Participación	Raúl García Díaz, Juan Luis Andión Tápiz

Referencias

[1] M. Pictures, "Theme report 2021: A comprehensive analysis and survey of the theatrical and home/mobile entertainment market environment for 2021."

[2] IMDb, "Imdb datasets."

[3] Lakshmipathi, "Imdb dataset of 50k movie reviews."

[4] H. Shankhdhar, "Imdb movies dataset."

[5] G. V. Research, "Movies and entertainment market size, share & trends analysis report by product (movies, music & videos), by region, and segment forecasts, 2022 - 2030."

[6] Maciej, "Dashboard for movie producer."

[7] VeNoMouS., "cloudscraper." https://github.com/VeNoMouS/cloudscraper, 2023.