WEB SCRAPING CON PYTHON

LEYNA ROXANA SALINAS VEYZAGA, M.Sc.

leynasud@gmail.com

Ingeniería Informática

Universidad Nacional "Siglo XX"

Llallagua, Bolivia

RESUMEN

El Web Scraping con Python es una técnica poderosa y accesible que permite extraer, organizar y analizar grandes volúmenes de datos desde páginas web, facilitando tareas como la investigación, el monitoreo de redes sociales y la automatización de procesos. Herramientas como Beautiful Soup, Selenium y Scrapy hacen posible esta práctica incluso para usuarios con conocimientos básicos de programación. Más allá de extraer información, el Web Scraping convierte la web en una fuente estratégica de conocimiento útil para la toma de decisiones, el desarrollo de proyectos y el análisis de tendencias.

1. INTRODUCCIÓN

Vivimos en una era donde los datos se han convertido en uno de los recursos más valiosos. Sin embargo, gran parte de esta información se encuentra dispersa en páginas web, no siempre accesible en formatos descargables o estructurados. En este contexto, surge el Web Scraping, una técnica que permite extraer información de sitios web de forma automatizada, con el fin de analizarla, almacenarla o utilizarla en distintos contextos.

Con Web Scraping, es posible recolectar grandes volúmenes de datos de forma rápida y precisa, algo que sería casi imposible hacer manualmente. Esta técnica resulta especialmente útil para tareas como la investigación de mercado, el seguimiento de precios, la recopilación de contenido, o incluso como parte de flujos de trabajo en proyectos de ciencia de datos y machine learning.

Una de las herramientas más potentes y accesibles para realizar Web Scraping es Python, debido a la riqueza de sus bibliotecas y a su sintaxis clara. Este artículo explora los fundamentos del Web Scraping, las bibliotecas más utilizadas, y herramientas específicas para la extracción de datos desde redes sociales.

2. DESARROLLO

¿Qué es Web Scraping?

El Web Scraping, también conocido como "scrapeo de contenidos" o "scrapeo de datos", consiste en utilizar software para obtener automáticamente información publicada en páginas web. Es similar al proceso que utilizan los motores de búsqueda como Google para indexar páginas web: un robot recorre los sitios, extrae los datos en formato HTML y los transforma en un formato estructurado como hojas de cálculo, bases de datos o archivos CSV.

Más allá de indexar, el objetivo del Web Scraping es transformar información no estructurada de la web en datos organizados y listos para ser analizados. Esto permite realizar estudios estadísticos, generar modelos predictivos, y realizar propuestas de análisis exploratorio o de inteligencia artificial.

Aplicaciones del Web Scraping

Entre los usos más comunes del Web Scraping destacan:

- Investigación de mercado
- Monitoreo de precios
- Seguimiento de noticias o publicaciones
- Análisis de redes sociales
- Generación de datasets para modelos de aprendizaje automático
- Extracción de datos educativos, científicos o estadísticos

Gracias a esta técnica, navegar por la web se convierte en una acción inteligente y estratégica, útil tanto para usuarios individuales como para instituciones.

Herramientas útiles para Web Scraping

Beautiful Soup

Es una biblioteca de Python ampliamente utilizada para extraer datos de archivos HTML y XML. Funciona generando un "árbol" de elementos del documento web, lo que permite localizar y extraer la información deseada de forma sencilla.

- Tiene una documentación completa
- Es fácil de usar
- Tiene una comunidad activa que ofrece soluciones variadas

¿Por qué el nombre “Beautiful Soup”? En desarrollo web, el término “tag soup” se usa de forma crítica para referirse a HTML mal estructurado. Beautiful Soup, como contrapartida, es una herramienta para ordenar y procesar esa sopa de etiquetas, extrayendo datos valiosos de ella.

Ejemplo práctico con Beautiful Soup

Uno de los ejemplos típicos es extraer datos sobre población mundial desde el sitio Worldometer. Este sitio ofrece estadísticas globales abiertas, ideales para prácticas de Web Scraping.

Selenium

Selenium es una herramienta de automatización de navegadores que permite simular la interacción humana con páginas web. Es muy útil cuando la información se genera dinámicamente con JavaScript o cuando es necesario interactuar con formularios, botones, etc.

- Compatible con varios lenguajes: Python, Java, C#, PHP, entre otros
- Permite grabar, depurar y ejecutar pruebas automáticas en la web

ChromeDriver

Es el ejecutable que permite a Selenium controlar el navegador Google Chrome. Gracias a él, se puede navegar, ingresar datos, hacer clics, ejecutar scripts y extraer resultados.

Scrapy

Scrapy es un framework de Web Scraping desarrollado en Python que permite construir bots capaces de rastrear páginas web y extraer información de forma eficiente y estructurada. Está diseñado para manejar grandes volúmenes de datos y facilitar su almacenamiento.

Herramientas para extracción de datos en redes sociales

Además del scraping en sitios web comunes, también se pueden extraer datos desde redes sociales, ya sea mediante Web Scraping directo o a través de las APIs oficiales.

Entre las herramientas más utilizadas tenemos:

- **Twint**: herramienta avanzada que permite obtener tweets sin necesidad de usar la API de Twitter.
- **Witterscraper**: extrae datos usando la API de Twitter.
- **Ultimate-Facebook-Scraper**: permite scrapear fotos, videos, posts y listas de amigos de un perfil de Facebook.
- **Facebook-miner**: accede a posts y comentarios mediante la API Graph de Facebook
- **Facebook-scraper**: permite scrapear publicaciones públicas sin necesidad de la API oficial.

Ejemplo con facebook-scraper en Python:

```
from facebook_scraper import get_posts
for post in get_posts('<id_cuenta>', pages=1):
    print(post['text'][:50])
```

Flujo de control típico del Web Scraping en redes sociales

1. **Indicar perfil y red social**: el usuario accede con su navegador a un perfil (Twitter, LinkedIn, Facebook).
2. **Solicitar scraping**: la herramienta identifica al usuario y solicita datos al servidor de la red social.
3. **Datos extraídos**: el servidor responde con los datos.
4. **Descarga**: el resultado se descarga y queda disponible para el usuario en el navegador.

3. CONCLUSIÓN

El Web Scraping con Python representa una de las formas más poderosas, flexibles y accesibles de recopilar y transformar información disponible en la web. Su utilidad se extiende a múltiples áreas como la investigación académica, el análisis de mercados, la vigilancia competitiva, el seguimiento de noticias, la creación de datasets para inteligencia artificial, y el análisis de redes sociales, entre muchos otros. Gracias a estas técnicas, es posible acceder a grandes volúmenes de datos en tiempo real, estructurarlos y convertirlos en conocimiento útil y accesible.

Bibliotecas como Beautiful Soup, Selenium y Scrapy han democratizado el acceso a esta tecnología, permitiendo que incluso personas con conocimientos básicos de programación puedan automatizar la recolección de datos y

aplicarlos en proyectos reales. Su capacidad de adaptación a diferentes tipos de páginas web, ya sean estáticas o dinámicas, las convierte en herramientas indispensables en el arsenal de cualquier analista de datos, programador o investigador digital.

Además, la posibilidad de aplicar Web Scraping sobre plataformas de redes sociales abre una nueva dimensión para el estudio del comportamiento humano en entornos digitales, la detección de tendencias, el monitoreo de opinión pública y la recopilación de contenido para análisis semántico y visualización de datos. Todo esto permite no solo acceder a la información, sino entenderla en su contexto y tiempo adecuado.

Figura 1: Fotografía de presentación de la ponencia