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

HECTOR MARTÍNEZ HIDALGO
JORDI MARTÍNEZ JOYEUX

Tabla de contenido

Contexto

En este proyecto, se ha decidido investigar el mercado de tokens no fungibles (NFT) en la plataforma AtomicHub. Esta plataforma, que opera en la red WAX (Worldwide Asset eXchange), permite a los usuarios comprar, vender, intercambiar y crear NFT. Estos NFT representan activos digitales únicos, como arte digital, coleccionables, objetos virtuales y más. Los NFT han ganado popularidad en los últimos años debido a su capacidad para verificar la autenticidad y la propiedad de activos digitales, lo que les confiere un valor único.

Dado el creciente interés en el mercado de NFT y su potencial para generar ingresos, resulta relevante analizar las tendencias y preferencias de los usuarios en AtomicHub, los precios a los que los usuarios ponen en venta sus activos digitales, la evolución de los precios que ha tenido un activo a lo largo del tiempo, la cantidad de unidades en venta de una determinada colección y su evolución en el tiempo, etc.

Esto permitirá a los creadores y vendedores de NFT adaptar su enfoque y desarrollar estrategias para atraer a más compradores e inversores.

En este trabajo, se ha elegido el sitio web https://wax.atomichub.io, el cual es una plataforma popular en la red WAX que ofrece un mercado de NFT. Este sitio fue seleccionado porque proporciona información detallada sobre las colecciones y NFT disponibles en la plataforma, incluyendo datos sobre el histórico de ventas realizadas y NFTs en venta en la actualidad. La información recopilada de AtomicHub es esencial para realizar un análisis exhaustivo del mercado de NFT y comprender mejor las preferencias de los usuarios.

Mediante el uso de web scraping, se recolectó información sobre una colección y los NFTs en venta que tiene actualmente en AtomicHub. El objetivo de esta práctica es obtener datos relevantes sobre el número de NFTs que tiene en venta una colección para analizar las tendencias y preferencias del mercado de NFT. Además, Atomichub facilita acceso público a su API de manera que en esta práctica se han contrastado los resultados del web scraping con la información extraída de la API para comprobar que el modelo académico de web scraping desarrollado por los autores funciona correctamente.

Título

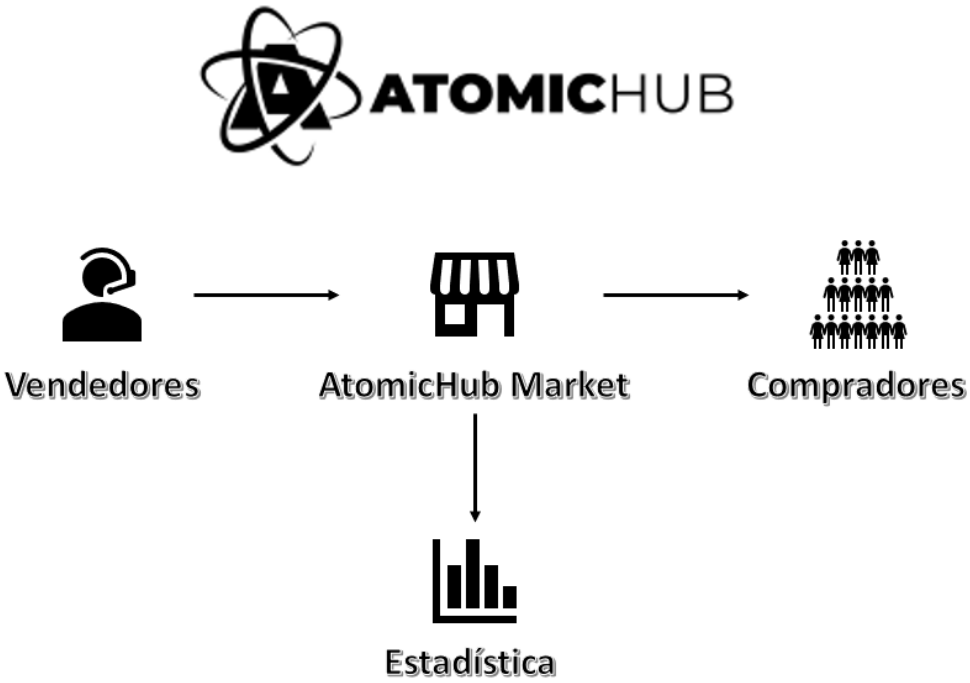
Datos relevantes de NFTs en venta en la plataforma Atomichub.

Descripción del Dataset

El dataset se compone de los datos extraídos en el momento de su ejecución de los NFTs en venta en la plataforma Atomichub. Durante la ejecución del código se realiza un listado con las Ids de los NFTs en venta que se encuentran en la URL facilitada. Una vez se han identificado las ventas se accede a cada URL de cada venta y se extrae la información relevante con la que se alimenta el dataset. Los datos del dataset son fiables ya que antes de finalizar la ejecución se comprueba a través de la API pública que facilita Atomichub que los datos extraídos son correctos.

El dataset contiene el nombre del NFT, su ID único, el nombre único de la colección, la cuenta del vendedor, el nombre del schema, el precio en WAX y su equivalente en USD. Los datos se presentan en formato CSV que facilitan un posterior análisis.

Representación gráfica



Contenido

El dataset contiene información sobre los NFTs que hay en venta en el momento de realizar el web scraping. Debido a que Atomichub tiene cientos de miles de NFTs en venta lo ideal es proporcionar una url sobre la que se quiera hacer la extracción de datos, para el proposito de esta práctica se ha utilizado la colección Dolpchainwax que es propiedad de Jordi Martínez, uno de los autores de este trabajo.

- A. **Name:** Nombre del NFT. Este nombre no es único ya que un NFT puede tener varias copias numeradas.
- B. **Asset ID**: Número de identificación del NFT, este identificador es único.
- C. **Collection Name:** Nombre de la colección, es un identificador único.
- D. **Seller**: Es la dirección de la wallet correspondiente al vendedor, es un identificador único.
- E. **Schema Name**: Las colecciones se dividen por schemas, lo que permite una subdivisión de los NFTs dentro de una colección. Un ejemplo podría ser, una colección con 2 schemas packs y cards, los packs serían los sobres de cartas y cards contendría solamente cartas.
- F. **Price:** Precio del NFT en WAX.
- G. **Price USD:** Precio del NFT en USD

Propietario

El conjunto de datos recolectado en este proyecto proviene de la plataforma AtomicHub, que es propiedad de la empresa Pink Network (pink.io). Hasta el momento de la realización de este estudio, no se han encontrado análisis similares que involucren la recolección y el análisis de datos de NFT en venta en AtomicHub.
Dado que los datos analizados en este proyecto están disponibles de manera pública en el sitio web de AtomicHub, se considera que su recolección y análisis son legales y éticos. Sin embargo, es importante destacar que se han seguido ciertos principios éticos y legales para asegurar el respeto y la protección de la información y los derechos de los involucrados en la plataforma.

Entre los pasos tomados para garantizar la adecuada consideración ética y legal en el contexto del proyecto, se incluyen:

- Asegurar que la información recolectada no incluye datos personales o información confidencial de los usuarios de AtomicHub.
- Utilizar los datos recolectados exclusivamente con fines de análisis y estudio, evitando su uso para actividades ilegales o malintencionadas.
- Respetar los términos de uso y las políticas de privacidad de AtomicHub y Pink Network al realizar la recolección y el análisis de datos.
- Dar crédito a AtomicHub y Pink Network como las fuentes de la información utilizada en el proyecto, mencionándolas adecuadamente en la documentación y publicaciones relacionadas.

Siguiendo estos principios y consideraciones éticas y legales, este proyecto busca contribuir al conocimiento y comprensión del mercado de NFT en AtomicHub de una manera responsable y respetuosa.

Inspiración

Este conjunto de datos resulta interesante debido al crecimiento y la popularidad de los tokens no fungibles (NFT) en el mundo digital y su impacto en la economía de los activos digitales. La información recolectada ofrece una visión detallada de las transacciones en el mercado de NFT en AtomicHub, específicamente en una colección particular. Con datos como el nombre del NFT, el nombre de la colección, el ID del NFT, el vendedor y el precio de venta en WAX y su equivalente en USD, es posible obtener una comprensión más profunda de las tendencias y preferencias de los usuarios en la plataforma.
Las preguntas que se pretenden responder con este conjunto de datos incluyen:

- ¿Cuáles son los NFT más populares y valiosos dentro de la colección?
- ¿Cuál es el rango de precios en el mercado de NFT para esta colección en particular, y cómo se distribuyen los precios entre los diferentes NFT en venta?
- ¿Existen patrones o tendencias en los precios y la demanda de NFT en función de sus características, como el nombre, la rareza o el diseño?
- ¿Cuál es la relación entre el precio en WAX y el precio en USD en las transacciones de NFT en AtomicHub, y cómo ha evolucionado esta relación con el tiempo?
- ¿Cómo se compara la popularidad y el valor de esta colección con otras colecciones disponibles en AtomicHub?

Al responder a estas preguntas, el conjunto de datos puede proporcionar información valiosa tanto para los creadores y vendedores de NFT como para los compradores e inversores interesados en el mercado de NFT en AtomicHub. Además, el análisis de estos datos puede contribuir al conocimiento general sobre el mercado de NFT y ayudar a identificar oportunidades y tendencias emergentes en el espacio de los activos digitales.

Licencia

Una licencia adecuada para este conjunto de datos es la CC BY-SA 4.0 License. La elección de esta licencia se basa en su adecuación a las características y objetivos del proyecto de recolección y análisis de datos de NFT en AtomicHub. Las cláusulas de la licencia CC BY-SA 4.0 respaldan los siguientes aspectos:

- Atribución del creador del conjunto de datos y reconocimiento de los cambios realizados: Al utilizar esta licencia, se garantiza que se atribuya adecuadamente la autoría del conjunto de datos y se reconozcan las modificaciones realizadas en relación con el trabajo original.
- Uso comercial permitido: La CC BY-SA 4.0 License permite el uso comercial del conjunto de datos, lo que aumenta las oportunidades de que empresas y organizaciones se interesen en la información recolectada. Esto facilita el desarrollo de nuevos proyectos y colaboraciones, al mismo tiempo que asegura el reconocimiento del autor original del conjunto de datos.
- Compartir bajo la misma licencia: La licencia CC BY-SA 4.0 exige que cualquier contribución o adaptación del conjunto de datos se publique bajo los mismos términos y condiciones. Esto garantiza que el autor original sea reconocido en todo momento y que sus derechos y deseos sobre el uso del conjunto de datos se respeten y mantengan a lo largo de futuras iteraciones y adaptaciones.

En resumen, en nuestra opinión la licencia CC BY-SA 4.0 se ajusta a las necesidades y objetivos del proyecto, ya que permite el uso y adaptación del conjunto de datos, fomenta la colaboración y el desarrollo de nuevos proyectos y asegura el reconocimiento y respeto de la autoría original en todo momento.

Se puede consultar el texto legal de la licencia en el repositorio GitHub:
https://github.com/hectormh88/TCVD-PRA1/blob/main/LICENSE

Código

Librerías utilizadas:

- requests 2.28.2
- selenium 4.8.3
- pandas 2.0.0
- bs4 4.12.1
- os (incluido en Python 3.11.0)
- time (incluido en Python 3.11.0)
- concurrent.futures (incluido en Python 3.11.0)
- random (incluido en Python 3.11.0)

Se puede consultar el archivo de requerimientos en el repositorio GitHub:
https://github.com/hectormh88/TCVD-PRA1/blob/main/source/requirements.txt

El proceso de recolección de datos:

Función para inicializar el navegador

Se declara la función *get_driver()* que permite iniciar una instancia de un navegador Chrome gracias a la librería Selenium. El archivo chromedriver.exe debe estar en el directorio actual. Se configura el driver para que se ejecute en segundo plano y establece el User-Agent:

> *Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/112.0.5615.138 Safari/537.36*

que igualmente es el que utiliza por defecto.

Función para extraer los page_ids únicos de los assets

Se declara la función *get_unique_page_ids(url)*, que inicia una instancia del navegador Chrome con la función anterior, para navegar de forma autónoma en una página con contenido dinámico de una búsqueda del market de NFTs AtomicHub para lograr cargar todos los NFTs arrojados por la búsqueda y luego extraer el código HTML con la librería BeautifulSoup para obtener los page_ids de cada NFT de la búsqueda, que luego son almacenados en una lista, la cual es retornada por la función. Se implementó el scroll down de selenium configurado a 600px para ir haciendo web scraping de manera secuencial al descenso de la web ya que los elementos que contienen los NFTs se generan dinámicamente con el scroll down.

Extracción de los page_ids de una búsqueda

Se declara una variable que alberga un string con la URL a la búsqueda en AtomicHub. Luego se ejecuta la función *get_unique_page_ids(url)*, para almacenar una lista de todas las page_ids presentes en el link de búsqueda. Muestra por pantalla el número de page_ids recolectados.

```
# Definir la URL de búsqueda
url = "https://wax.atomichub.io/market?collection_name=dolpchainwax&data:text.rarity=legendary&order=desc&sort=created&state=1&symbol=WAX"

# Obtener los page_ids únicos
unique_page_ids = get_unique_page_ids(url)
```

✓ 38.1s

```
Número de page_ids únicos: 35
```

Función para extraer los datos de los assets

Se declara la función *get_asset(page_id)*, para obtener los datos de cada *asset* a partir de su página web, construida con el page_id. Utiliza Selenium para obtener el código HTML y BeautifulSoup para extraer los datos del *asset* y almacenarlos en un diccionario.

Función para ejecutar la función anterior de forma concurrente

Se declara la función *process_assets(asset_ids)* para poder ejecutar la función *get_asset()* de forma concurrente en 4 threads a la vez y disminuir el tiempo de construcción del dataframe.

Extracción de datos de assets

Se ejecuta la función *process_assets(asset_ids)* y se descargan los datos:

```
[9]  1  # Procesar los assets y obtener los resultados en un DataFrame
     2  df = process_assets(unique_page_ids)
```

```
Output exceeds the size limit. Open the full output data in a text editor
Datos del asset 1099598474219 descargados correctamente
Datos del asset 1099596712324 descargados correctamente
Datos del asset 1099600692578 descargados correctamente
Datos del asset 1099597863191 descargados correctamente
Datos del asset 1099596983341 descargados correctamente
Datos del asset 1099557129293 descargados correctamente
Datos del asset 1099533983622 descargados correctamente
Datos del asset 1099611647804 descargados correctamente
Datos del asset 1099596675169 descargados correctamente
Datos del asset 1099597304476 descargados correctamente
Datos del asset 1099596892497 descargados correctamente
Datos del asset 1099676203709 descargados correctamente
Datos del asset 1099529438187 descargados correctamente
Datos del asset 1099521183790 descargados correctamente
Datos del asset 1099596712323 descargados correctamente
Datos del asset 1099597366251 descargados correctamente
Datos del asset 1099809618296 descargados correctamente
Datos del asset 1099598064253 descargados correctamente
Datos del asset 1099597285655 descargados correctamente
```

Luego se almacenan los datos en un archivo CSV.

Extracción de datos mediante API y comparación de resultados

También se han extraído los mismos datos mediante API (excepto los precios, que no están disponibles directamente). Se ha construido un dataframe con los datos extraídos y se ha comparado con el dataframe construido con webscraping, resultando ser iguales:

```
1  # Comparar los dataframes
2  df[['Name', 'Asset ID', 'Collection Name', 'Seller', 'Schema Name']].equals(df_api)
```

```
True
```

Dificultades que presenta el sitio web elegido:

- *Bloqueo de solicitudes por requests, no descargando la totalidad del contenido HTML:*
  Al intentar implementar la función *get_asset()* con la librería requests para obtener el contenido HTML de la página web de cada *asset*, el código HTML descargado estaba incompleto.
  Es probable que el servidor bloquease las solicitudes realizadas por scripts para proteger sus datos. Aunque se haya utilizado un encabezado de User-Agent para simular un navegador, es posible que el servidor aún esté bloqueando las solicitudes.
  Se ha utilizado la librería Selenium, que utiliza un driver que permite interactuar con páginas web y que sí logró obtener el código HTML completo.
- *Tiempo excesivo para ejecutar cada solicitud a la página de un asset:*
  Si bien Selenium permite simular el comportamiento humano en la navegación, también es más lento en la obtención del código HTML. Por lo que se ha creado una función para realizar las solicitudes de forma concurrente, con 4 threads paralelos que permiten obtener los datos de 4 *assets* a la vez.
- *Errores en la obtención de datos por no alcanzar a descargarse el codigo HTML:*
  Al no haber tiempo entre que se ejecutaba la descarga del código HTML con Selenium y la búsqueda de texto con BeautifulSoup, se generaban errores.
  Se han establecido tiempos de espera entre ambos procesos para evitar este error.
- *Posibilidad de bloqueo de IP por tiempos de espera idénticos entre solicitudes:*
  Para evitar evidenciar patrones idénticos en los tiempos de espera entre solicitudes por multiproceso, se ha establecido un tiempo de espera aleatorio entre 1,0 y 5,0 segundos, para simular el comportamiento humano.
- *Rendimiento y velocidad:*
  Se ha priorizado utilizar requests por sobre Selenium donde fuese factible, para aumentar la velocidad de verificación de conexión y de descarga de datos, al ser una librería que requiere menos recursos y más rápida.

Se puede acceder al código fuente en el repositorio GitHub:
https://github.com/hectormh88/TCVD-PRA1/blob/main/source/Webscraping-AtomicHub.ipynb

Dataset

El dataset ha sido publicado en Zenodo y puede accederse a él en el siguiente enlace:
https://zenodo.org/record/7856463

Contiene el dataset generado con técnicas de *webscraping* y el dataset generado mediante el uso de la API de AtomicHub, para su comparación.

También es accesible dentro de la carpeta *source* del repositorio público de GitHub:
https://github.com/hectormh88/TCVD-PRA1/tree/main/source

Vídeo

https://drive.google.com/file/d/1gwAb9Kxnzic7OTJYWEMBTaaTSKGP620z/view?usp=sharing