

Práctica 1: Web Scraping

Autor: Alejandro Ortega de los Ríos

Introducción

El objetivo de esta práctica será la extracción de un dataset de criptomonedas de una página web.

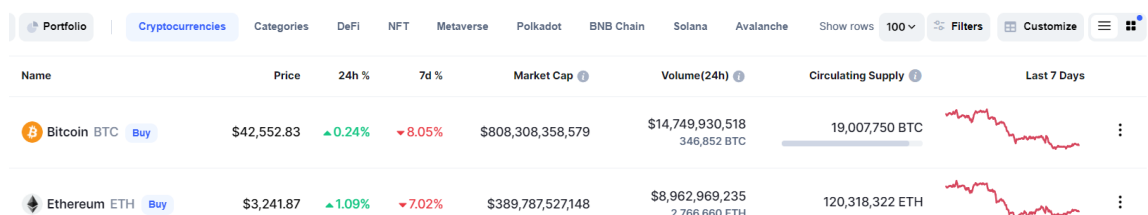
Se trata de un tema de especial interés al ser un mercado bastante moderno, y por ello altamente volátil. La creación de un dataset con información de las distintas criptomonedas en un determinado intervalo de tiempo puede ser de gran utilidad para aquellos proyectos orientados al trading, que precisen de modelos predictivos para optimizar ganancias y reducir riesgos.

El problema ahora es el siguiente: el acceso a los datos. La descarga de un dataset normalmente es posible a través de suscripciones premium de distintas páginas web. El histórico de datos está disponible únicamente para visualización en los sitios web. Es aquí donde entra en juego el web scraping.

Contexto

Para la recopilación de datos se han analizado los siguientes sitios web:

- <https://coinmarketcap.com/>





Name	Price	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last 7 Days
Bitcoin BTC Buy	\$42,552.83	▲0.24%	▼8.05%	\$808,308,358,579	\$14,749,930,518 346,852 BTC	19,007,750 BTC	
Ethereum ETH Buy	\$3,241.87	▲1.09%	▼7.02%	\$389,787,527,148	\$8,962,969,235 2,766,660 ETH	120,318,322 ETH	

Figura 1: Coinmarketcap, contenido de <table>

Al hacer click en Bitcoin p.ej., nos lleva a la siguiente página:

Historical Data for Bitcoin

Date	Open*	High	Low	Close**
Apr 09, 2022	\$42,282.08	\$42,786.82	\$42,183.25	\$42,782.14
Apr 08, 2022	\$43,505.14	\$43,903.02	\$42,183.29	\$42,287.66

Figura 2: Coinmarketcap, histórico de datos

Donde aparecen el histórico de valores de mercado de la criptomoneda en cuestión. En vista a la estructura de la página (bastante simple al tener el contenido estructurado en una tabla), y al potencial de datos que ofrece, el esfuerzo de este proyecto se enfocó inicialmente en esta página.

Inicialmente se decide realizar extracción de datos usando las librerías requests y bs4, al ser estas las empleadas en la teoría de la asignatura. Además, dada la simplicidad de la página, no requeriría una interacción avanzada (buttons, input values, etc.) del script con esta.

Durante este periodo se encontraron los siguientes inconvenientes:

- Tras extraer los elementos de la tabla (Fig. 1) se observa lo siguiente: tras el décimo registro empiezan a aparecer valores faltos cuando no existen cambios en la estructura html de la tabla.
- Si bien al inspeccionar el contenido html de la página manualmente aparece la tag "table" (Fig. 2), tras imprimir el código obtenido con requests y bs4 esta no aparece.

En vista a lo anterior se decide emplear técnicas anti-webscraping: generar aleatoriamente distintos user-agent dentro del http header para cada búsqueda.

Tras su implementación, el problema seguía persistiendo. Se decide descartar el sitio web y buscar uno más adecuado.

- <https://www.tradingview.com/markets/cryptocurrencies/prices-all/>



Overview Performance Oscillators Trend-Following USD BTC									
NAME 835 MATCHES		MKT CAP	FD MKT CAP	LAST	AVAIL COINS	TOTAL COINS	TRADED VOL	CHG %	
 Bitcoin		808.451B	893.187B	42532.71	19.008M	21M	14.646B	-0.52%	
 Ethereum		389.593B	389.593B	3238.02	120.318M	120.318M	8.972B	-0.63%	

Figura 3: tradingview, contenido de <table>

La estructura es bastante similar al anterior. Existe un inconveniente a primera vista: al clicar en una criptomoneda en vez de aparecer el histórico de datos salen noticias en su lugar, por lo que el registro temporal del dato requerirá otro enfoque distinto.

Analizando el contenido html en profundidad se encuentra lo siguiente:

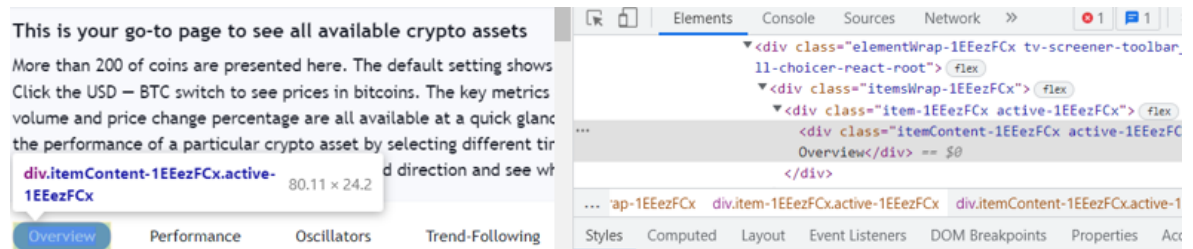


Figura 4: tradingview, explorando botones

La etiqueta div asociada a los botones no tienen un enlace asociado, sino que la interacción es a través de JavaScript, por lo que no se podrá buscar ese elemento y acceder los links correspondientes a través de BeautifulSoup.

Ante la posibilidad de extraer un volumen de datos mayor, y en vista a las limitaciones que presenta BeautifulSoup (extrae únicamente contenido html, no puede gestionar JS ni AJAX requests), se plantea el empleo de Selenium en su lugar^[1]:

- Se crea un Chrome webdriver.
- Con el método `find_element(By.PATH, ...).click()` se interactuará con el contenido web (botones, etc.) para acceder a nuevas tablas y así extraer un volumen mayor de datos.

Tras unas cuantas pruebas, se comprueba que poco después de haber hecho un par de interacciones del código con el sitio web (inicialmente, al gestionar las cookies), el servidor web cierra la sesión y finaliza la ejecución del código.

De hecho, tras la ejecución del código, al abrir el navegador Google Chrome avisa de que estamos usando un script:

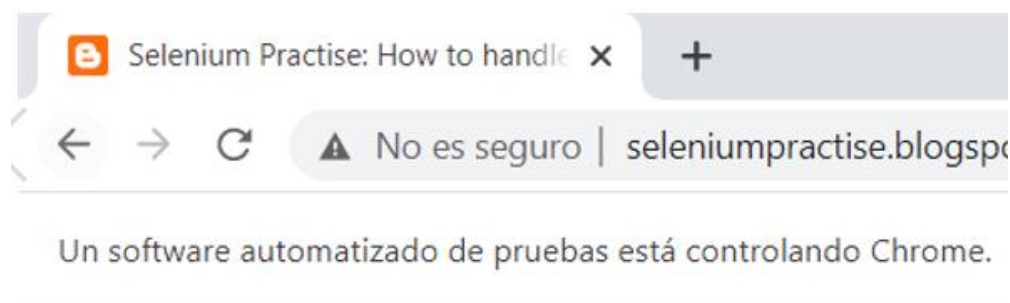


Figura 5: Primeros pasos con Selenium

Por este motivo, se decide emplear técnicas que eviten anti-scraping. Más adelante se detallan las técnicas empleadas para resolver esta problemática.

En vista a todo lo anterior, finalmente se decide hacer lo siguiente:

- Emplear técnicas que eviten anti-scraping
- Interacción entre script y sitio web a través de Selenium.
- Emplear BeautifulSoup para extraer código HTML.
- Generar un dataset y exportarlo a un fichero CSV.

TradingView

El objetivo de este apartado es realizar un análisis en profundidad del sitio web, en términos de HTML y JS.

En primer lugar, se muestra una captura del archivo robots.txt:

```
User-agent: *
Disallow: /chat/m/
Disallow: /search/
Disallow: /idea-popup/
Disallow: /*mobileapp=true
Disallow: /*popup=true
Disallow: /*dark=false&popup=true
Disallow: /widgetembed/
Disallow: /embed-widget/
Disallow: /*?support
Disallow: /badbrowser/
Disallow: /jobs/
Sitemap: https://www.tradingview.com/sitemap.xml
```

Figura 6: robots.txt

Como se puede apreciar, todos los bots estarían excluidos de los directorios listados en el fichero.

En segundo lugar, no existe una API como tal^[2]. En su lugar, existe una REST API específica para aquellos Brokers que estén admitidos en su plataforma, lo que hace el web scraping el único método viable en principio.

A continuación, se procederá con un estudio de la estructura web de TradingView, haciendo uso de la herramienta de inspección de código del navegador.

Durante el crawling, se partirá inicialmente desde la página principal, gestionando las Cookies y navegando por el menú hasta llegar a la página que alberga las tablas objetivo.

A continuación, se muestra la estructura de la página principal:

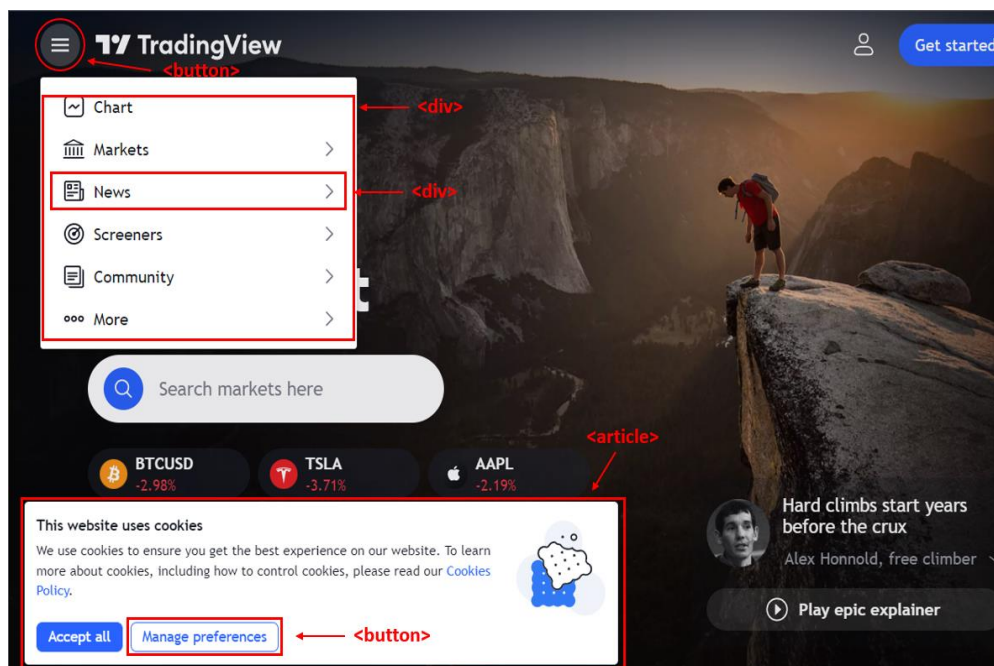
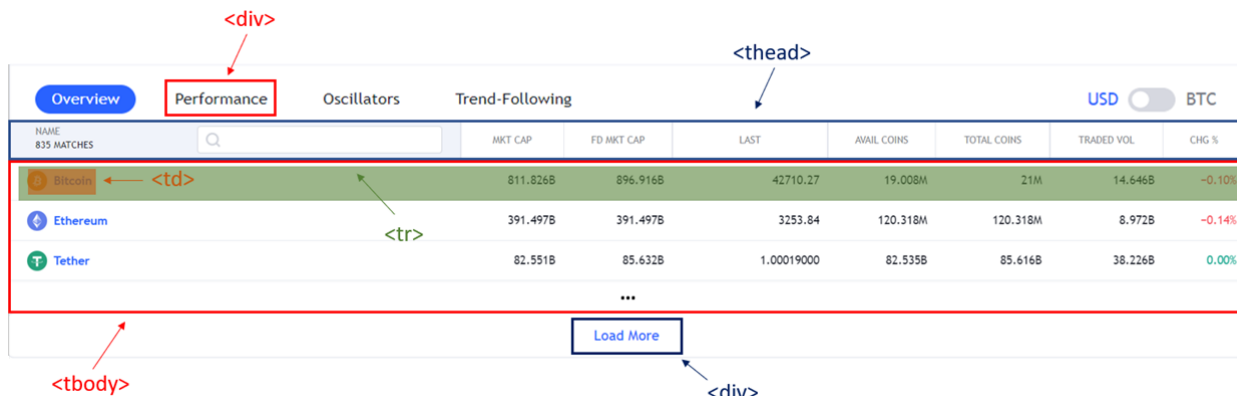


Figura 7: TradingView.com, esquema simplificado

Tras navegar por el menú, se llega a la siguiente página:



NAME	MKT CAP	FD MKT CAP	LAST	AVAIL COINS	TOTAL COINS	TRADED VOL	CHG %
Bitcoin	811.826B	896.916B	42710.27	19.008M	21M	14.646B	-0.10%
Ethereum	391.497B	391.497B	3253.84	120.318M	120.318M	8.972B	-0.14%
Tether	82.551B	85.632B	1.00019000	82.535B	85.616B	38.226B	0.00%

Figura 8: Análisis del contenido objetivo

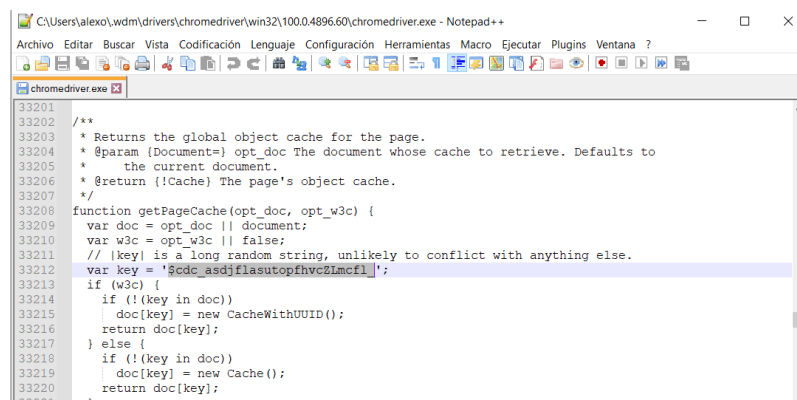
En esta página se extraerá el contenido de la tabla. Un aspecto importante es la presencia de botones en la parte superior e inferior, que modifican el contenido y el número de registros respectivamente. Estos botones no tienen un enlace asociado que permita su parseo con BeautifulSoup, este es otro caso que requiere de Selenium.

Anti-Scraping

Si bien el objetivo de esta práctica es extraer datos web, el objetivo de un sitio web puede ser la contraria, evitar la extracción automatizada de sus datos.

En la presente práctica se emplearon las siguientes técnicas ^[3]:

- WebDriver Flag: indica al navegador el empleo de herramientas de automatización, por lo que se debe eliminar.
- User-Agent: se genera uno aleatoriamente para cada sesión.
- ChromeDriver.exe: es aconsejable modificar algunas de las variables del código de JS, dado que muchos softwares de detección de Bots las buscarán. Por medio de Notepad se editó el código del siguiente modo ^[2]:



```

33201
33202 /**
33203  * Returns the global object cache for the page.
33204  * @param {Document=} opt_doc The document whose cache to retrieve. Defaults to
33205  *   the current document.
33206  * @return {!Cache} The page's object cache.
33207  */
33208 function getPageCache(opt_doc, opt_w3c) {
33209   var doc = opt_doc || document;
33210   var w3c = opt_w3c || false;
33211   // |key| is a long random string, unlikely to conflict with anything else.
33212   var key = '$cdc_asdfflasutopthvc2Lmcfl';
33213   if (w3c) {
33214     if (!(key in doc))
33215       doc[key] = new CacheWithUUID();
33216     return doc[key];
33217   } else {
33218     if (!(key in doc))
33219       doc[key] = new Cache();
33220     return doc[key];
33221   }

```

Figura 9: Editando ChromeDriver.exe

La variable sombreada fue modificada (se mantuvo el número de caracteres únicamente).

- Búsqueda de posibles Honeypots: no se encontró nada significativo, con una salvedad: los campos <thead> y <tbody> aparecen duplicados, y en su primera aparición no tienen contenido. Esto puede dificultar la extracción de datos si no se detecta durante el análisis previo.
- Retardos aleatorios: a cada interacción con el sitio web (click a nuevos enlaces, botones, etc.) se le introdujo un tiempo de espera de 1 a 3 segundos.
- Mouse Action: simular la interacción de un puntero con el sitio web. De este modo se evita hacer click en elementos sin haber tenido un “hover action” previo (el cambio de color al posicionar el puntero sobre un botón).

Dataset

Como se adelanta en la Fig. 3, el dataset es lo forman el conjunto de tablas que aparecen tras el click de cada botón. Para la práctica se hará click hasta cuatro veces en el botón Load More (a partir de este momento se considerarán poco relevantes los registros, dado que están ordenados por su valor de mercado).

Para cada tabla se generará un dataset, siendo el dataset final el resultante de un JOIN entre los anteriores. Se obtendrán los siguientes datasets, junto con los campos enumerados a continuación:

- **Overview:**

- NAME: nombre de la criptomoneda.
- MKT CAP: capitalización de mercado, es el valor total de todas las criptomonedas en circulación.
- FD MKT CAP: sus siglas, fully diluted market capitalisation, hacen referencia al valor total de todas las criptomonedas (en circulación o pendientes de ser minadas)^[4].
- LAST: último valor de mercado (USD).
- AVAIL COINS: número de monedas disponibles en circulación.
- TOTAL COINS: número total de monedas.
- TRADED VOL: volumen de monedas comerciadas.
- CHG %: cambio en su valor desde el último periodo.

- **Performance:**

- NAME: nombre de la criptomoneda.
- CHG %: cambio en su valor desde el último periodo.
- 1 W CHG %: cambio en su valor semanal.
- 1 M CHG %: cambio en su valor mensual.
- 3-MONTH PERF: cambio en su valor trimestral.
- 6-MONTH PERF: 1 W CHG %: cambio en su valor semestral.
- YTD PERF: cambio en su valor desde el 1 de enero hasta la fecha actual.
- YEARLY PERF: cambio en su valor en un año natural
- VOLATIBILITY: a partir de los valores anteriores indica en qué proporción ha variado su valor de mercado ^[5].

- **Oscillators:**

- NAME: nombre de la criptomoneda.
- OSCILLATORS RATING: puntuación en base a los distintos osciladores.
- ADX: índice direccional medio. Indica cómo de fuerte es una tendencia. Se considera fuerte por encima de 25 y bajo por debajo de 20 ^[6].
- AO: oscilador asombroso. Compara el “market-momentum” actual con el genérico y predice cambios de tendencia, si la tendencia es alcista o bajista, etc. ^[7].
- ATR: promedio del rango verdadero. Se emplea para predecir la evolución futura de un activo a través de la medición de la volatilidad en los precios ^[8].
- CCI20: Commodity Chanel Index. Mide el comportamiento estacional o cíclico de los últimos 14 periodos ^[9].

- **Trend-Following:**

- MOVING AVERAGES RATING: puntuación sobre la media móvil.
- LAST: último valor de mercado (USD).
- SMA20: media móvil 20.
- SMA50: media móvil 50.

El dataset final tendría las siguientes dimensiones: 614 registros con 35 campos.

Los campos NAME, LAST y CHG % se encuentran duplicados. Esto se tendrá en cuenta al unir los datasets.

Lanzamiento del Proyecto

De acuerdo con el enunciado de la práctica, el proyecto será publicado en las siguientes plataformas:

PLATAFORMA	CONTENIDO	ENLACE
Github	Memoria (.pdf) Código (.py)	https://github.com/aortegade/PAC1-WEB-SCRAPING.git
Zenodo	Dataset (.csv)	DOI: 10.5281/zenodo.6449890
GoogleDrive	Vídeo (.mp4)	https://drive.google.com/drive/folders/1DSFcViZM9rfsdgT1XBHRnOY4Uy4_OU-?usp=sharing

Para el dataset se ha escogido la licencia Creative Commons Zero v1.0 Universal. Este tipo de licencia es una renuncia de derechos de autor en favor del dominio público. Dado que se trata de un proyecto de ámbito académico y no empresarial, se considera la licencia más idónea.

Agradecimientos

En primer lugar agradecer el trabajo del personal de TradingView por el desarrollo y mantenimiento de su sitio web, además de mantener sus datos actualizados.

En segundo lugar, gran parte de este trabajo ha sido posible gracias a los artículos de Medium, que mediante ejemplos explicativos han hecho el proceso de aprendizaje mucho más rápido.

Bibliografía

- [1] <https://medium.com/analytics-vidhya/scrapy-vs-selenium-vs-beautiful-soup-for-web-scraping-24008b6c87b8>.
- [2] <https://es.tradingview.com/support/solutions/43000474413/>
- [3] <https://piprogramming.org/articles/How-to-make-Selenium-undetectable-and-stealth--7-Ways-to-hide-your-Bot-Automation-from-Detection-0000000017.html>
- [4] <https://capital.com/fully-diluted-market-capitalisation-in-cryptocurrency-definition>
- [5] <https://www.coinbase.com/es/learn/crypto-basics/what-is-volatility#:~:text=Definition,up%20or%20down%20over%20time>.
- [6] <https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/adx#:~:text=ADX%20stands%20for%20Average%20Directional,System%20developed%20by%20Welles%20Wilder>.
- [7] <https://www.moneycontrol.com/news/business/markets/technical-classroom-how-to-use-awesome-oscillator-in-trading-strategy-4201371.html>
- [8] <https://economipedia.com/definiciones/atr-average-true-range.html#:~:text=El%20ATR%20o%20promedio%20del,la%20volatilidad%20en%20los%20precios>.
- [9] <https://economipedia.com/definiciones/commodity-channel-index-cci.html>
- [10] <https://creativecommons.org/licenses/?lang=es>
- [11] <https://www.fastcompany.com/3014553/what-coders-should-know-about-copyright-licensing#:~:text=On%20GitHub%20the%20three%20main%20types%20of%20software%20licenses%20are%3A&text=It%20permits%20users%20to%20do,grants%20patent%20rights%20to%20users>.
- [12] Subirats, L., Calvo, M. (2018). Web Scraping. Editorial UOC.
- [13] Lawson, R. (2015). Web Scraping with Python. Packt Publishing Ltd. Chapter 2. Scraping the Data.