Tipología y ciclo de vida de los datos

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

Aula 1

Miguel Martínez Ruíz y Miguel Lima Medín

Índice de Contenidos

Contexto

Recargos de metales

En la industria de la fundición de metales es común aplicar recargos de metal (metal surcharge). Su propósito es desvincular la fluctuación del precio de mercado de una materia prima del precio negociado. La parte compradora y vendedora acordarían un precio base y la aplicación de un recargo vinculado al precio de mercado de la materia prima subyacente.

Hay dos lados en cada recargo: el aplicador del recargo y el pagador del recargo. Una empresa puede ser pagadora de recargos en los componentes que compra a sus proveedores, y a su vez aplicadora de recargos a sus clientes al vender el producto terminado elaborado a partir de esos componentes.

El acuerdo de revisión de precios marcará una periodicidad de actualización y una fórmula de cálculo. Puede que se acuerde aplicar la actualización de precio solamente tras superar una determinada variación porcentual del precio de partida.

Cada materia prima tiene un precio marcado en un momento dado para un mercado geográfico y moneda determinada.

El valor de negocio de contar con la información actualizada

Al pagador del recargo le va a interesar estar constantemente pendiente de la evolución del precio de la materia prima para reclamar una reducción a su proveedor cuando este baja.

Como aplicador del recargo, la empresa también debe estar alerta para subir el precio sin demora.

Incluso en los casos en los que no se cuenta con un acuerdo de revisión de precios, interesa revisar periódicamente la evolución del mercado para poder renegociar el precio inicial si las materias primas subyacentes han sufrido una variación significativa.

Por tanto, cualquier retraso en el procesamiento de esta información puede resultar en un precio subóptimo y la perdida de dinero.

El coste de acceder periódicamente a la información

Estos serían los pasos típicos para acceder a la información de precios manualmente:

1. Abrir la página web
2. Navegar hasta el metal que se desea consultar
3. Escoger las fechas, y la moneda aplicable
4. Consultar el precio
5. Repetir pasos 2 a 4 para cada metal

Con el web scrapper diseñado se automatizarían todos estos pasos grabando la información en un fichero .csv

Página web seleccionada

Se selecciona Index Mundi por contener la información que se necesita.

Robots.txt

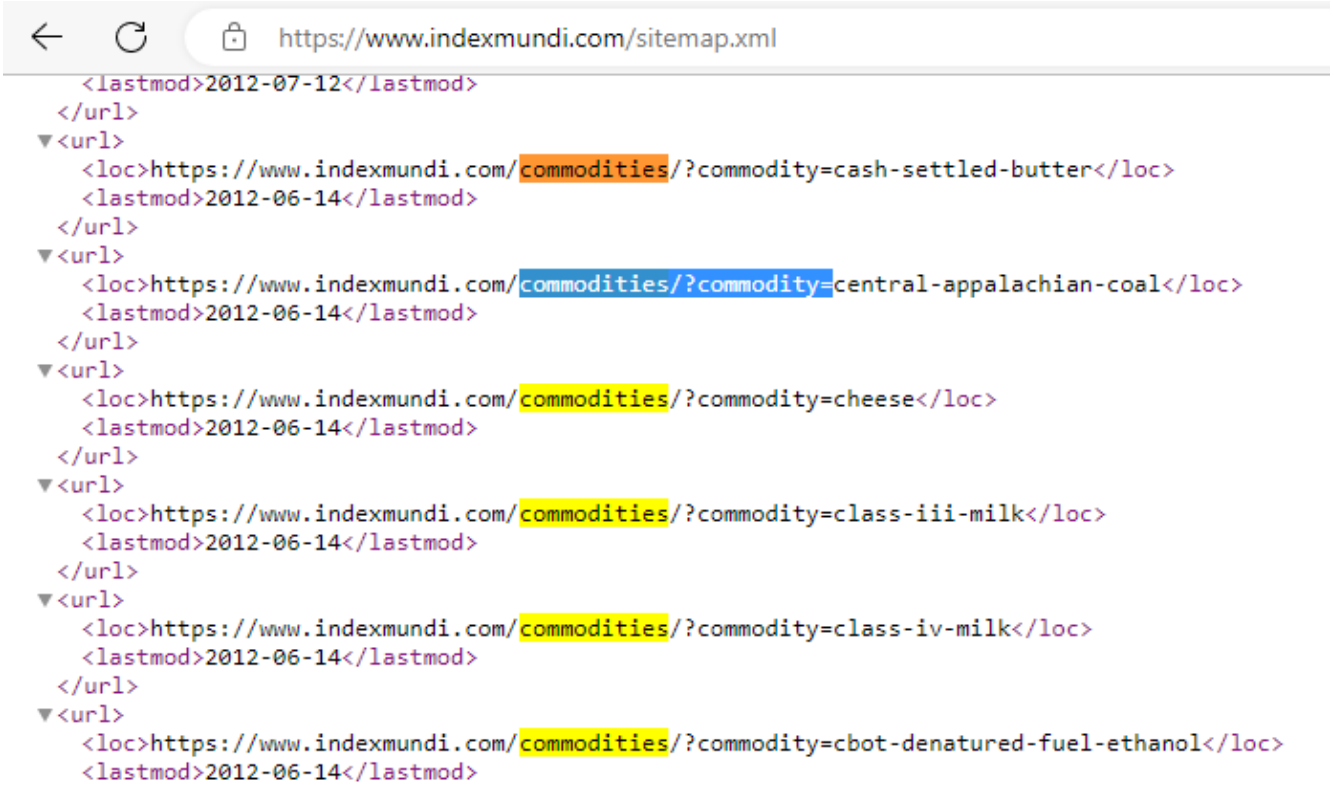
Se comprueba en su fichero robots.txt que en general se permiten los web scrappers, con la salvedad de una serie de robots específicados en su lista negra. Respetaremos la limitación indicada, no accediendo al área /api/v2.

```
Sitemap: http://www.indexmundi.com/sitemap.xml

User-agent: *
Disallow: /api/v2/

User-agent: compatible; attributor
Disallow: /

User-agent: attributor
Disallow: /

User-agent: kalooga
Disallow: /

User-agent: Mp3Bot
Disallow: /

User-agent: WebAlta Crawler
Disallow: /

User-agent: TurnitinBot
Disallow: /

User-agent: GbPlugin
Disallow: /

User-agent: Domain Re-Animator Bot
Disallow: /

User-agent: trendkite-akashic-crawler
Disallow: /
```

Sitemap.xml

Verificamos el fichero Sitemap.xml

Identificamos la nomenclatura de las páginas que nos interesa extraer para obtener la información de las commodities:



Título

Histórico de precios de materiales base

Descripción del Dataset

El Dataset consta de información de precios para varias commodities.

Commodities extraidas

Las commodities a extraer son seleccionadas por el usuario o usuaria:

A) Si se indica una lista de commodities deseadas se extraerán exactamente esas commodities

B) Por defecto el programa toma la lista "commodities_list.txt" en la carpeta donde se encuentra main.py

C) También se pueden pedir todas las commodities que el web scrapper encuentre en el menú a través de BeautifulSoup, con la opción --download_from_menu

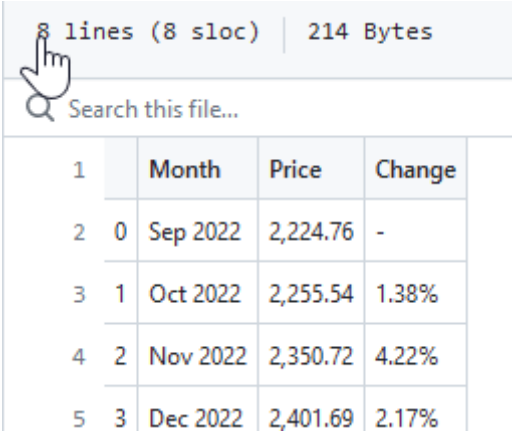
D) También se pueden pedir todas las commodities que el web scrapper encuentre en sitemap.xml, con la opción --download_from_sitemap

Estructura de carpetas y ficheros

Existe una carpeta para cada día en el que se ha ejecutado el script, en ../dataset/. Dentro de esa carpeta existe un fichero .csv para cada commodity.

Cada fichero tiene la siguiente estructura:

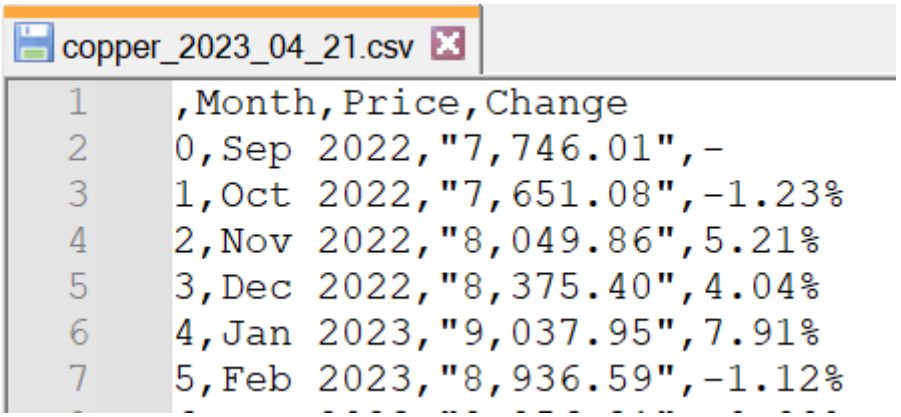
1. Una primera fila de cabecera con el nombre de las tres columnas Month, Price y Change
2. Una fila para cada registro mensual


8 lines (8 sloc) | 214 Bytes

Search this file...

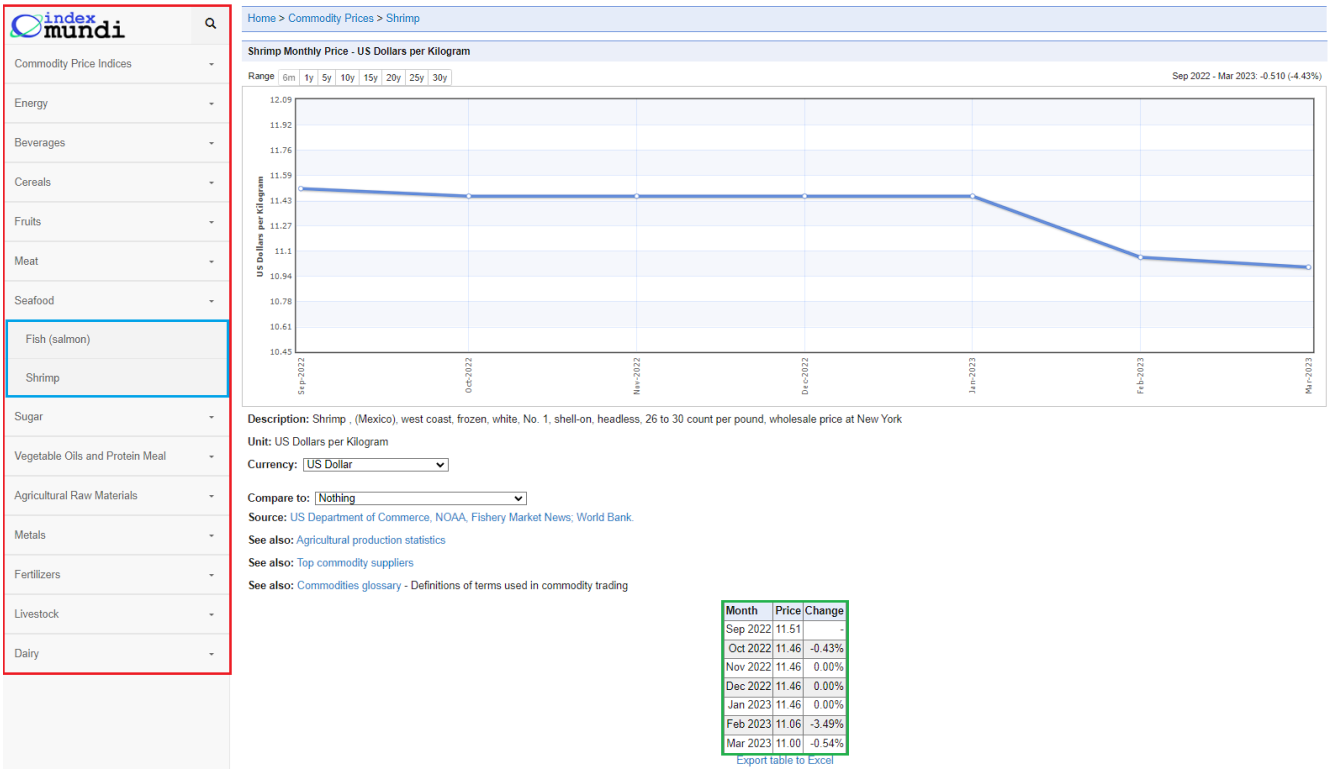
		Month	Price	Change
2	0	Sep 2022	2,224.76	-
3	1	Oct 2022	2,255.54	1.38%
4	2	Nov 2022	2,350.72	4.22%
5	3	Dec 2022	2,401.69	2.17%

El formato es el de un texto plano separado por comas:


copper_2023_04_21.csv

```
1  ,Month,Price,Change
2  0,Sep 2022,"7,746.01",-
3  1,Oct 2022,"7,651.08",-1.23%
4  2,Nov 2022,"8,049.86",5.21%
5  3,Dec 2022,"8,375.40",4.04%
6  4,Jan 2023,"9,037.95",7.91%
7  5,Feb 2023,"8,936.59",-1.12%
```

Representación Gráfica



En la imagen podemos ver la estructura de la página web elegida.
En el cuadro rojo se pueden observar los índices.
Dentro de estos (cuadro azul) se encuentran los subíndices con los que accedemos a las páginas que contienen los datos de las commodities.
Al acceder a uno de estos, en la página contamos con la tabla (cuadro verde) de la que extraemos la información que guardamos en el dataset. De esta forma, nuestro dataset contará con un archivo CSV con los datos de esta tabla para cada commodity. Si no hubiera tabla, registramos un log con este hecho. Estos datos se guardan en una carpeta correspondiente a la fecha de ejecución de nuestro programa.

Contenido

En nuestro dataset incluimos en cada archivo, los datos del precio de la commodity para cada mes en un periodo de seis meses, así como el cambio porcentual del precio respecto al mes anterior.

Propietario

IndexMundi es un portal de datos que recopila hechos y estadísticas de múltiples fuentes.

Buscamos en Zenodo contenido sobre precios de metales. Restringimos la busqueda a nuestro dominio:

Se revisan los títulos y descripción de los 51 resultados, pero ninguno encaja con lo que se pretende.

Buscando datasets abiertos de "*commodities prices*" resultan 275 entradas. Revisadas todas, solo se identifican dos análisis similares a nuestro ejercicio:

Dataset Candidato	Análisis
Evolució del preu de les matèries primeres	Toma tambien los datos de IndexMundi Contiene información sobre todo tipo de materias primas, pero no está actualizado En nuestro caso no necesitamos extraer algunos de los campos que se escogieron en este análisis previo.
Practica 1 Web Scraping Oil Price Data	Solo incluye el precio de petroleo

Inspiración

Inserta aquí la información de la inspiración.

Licencia

Inserta aquí información sobre la licencia del dataset.

Código

Módulos

En nuestra estructura de código, hemos separado las funcionalidades del software en tres archivos diferentes:

- **main.py**: archivo principal, que contiene la configuración de los argumentos con los que podemos ejecutar la función principal.
  Dependiendo de estos parámetros, realizaremos el proceso de web scraping, a partir del sitemap, del índice del sitio web, o de una lista de commodities incluidas en un archivo de texto. De cara a evitar posibles bloqueos de IP, realizaremos un retardo aleatorio entre cada petición al sitio web.
- **read_available_commodities**: en este archivo definimos las dos funciones que se pueden ejecutar a través de los argumentos que le pasamos a la función main.py (obtener la lista de commodities a procesar desde el sitemap o bien desde el menú del índice).
- **extract_data_material.py**: contiene la función a la que llamamos desde el archivo main, que realiza propiamente el proceso de web scraping. En concreto, busca la tabla que contiene los datos de la commodity, y los devuelve en formato dataframe.

Dificultades enfrentadas

El sitemap.xml no estaba actualizado, así que creamos una segunda función que descubre las páginas enlazadas en el site.

Algunas páginas no contenían la tabla de información, así que se habilitó la gestión de errores para capturar ese fallo en el fichero de log y que el código continúe su ejecución.

Estructura logs

En la carpeta ../logs/ se guarda un fichero log para cada día. Multiples ejecuciones en un mismo día añaden líneas al fichero ya existente.

Se registra en el log tanto las páginas procesadas correctamente

```
491  DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.indexmundi.com:443
492  DEBUG:urllib3.connectionpool:https://www.indexmundi.com:443 "GET /commodities/?commodity=gold HTTP/1.1" 200 57287
```

como aquellas en las que se ha producido algún error

```
481  DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): www.indexmundi.com:443
482  DEBUG:urllib3.connectionpool:https://www.indexmundi.com:443 "GET /commodities/?commodity=wood-pulp HTTP/1.1" 200 53257
483  ERROR:root:20:10:33 Error scraping data for wood-pulp: 'NoneType' object has no attribute 'find_all'
```

Consideraciones código

Indicación	Aplicación
Descubrimiento de enlaces y navegación autónoma	Creadas dos funciones de navegación autónoma para obtener el listado de URLs a procesar: 1) Explorando el contenido de sitemap.xml 2) Navegando el menú lateral con BeautifulSoup

---

Indicación	Aplicación

Mecanismos que permitan ejecutar un uso apropiado del web scraping	Parada de un tiempo aleatorio entre 1 y 3 segundos entre cada página a procesar
	`for commodity in commodities_to_process:`
	`  # Pausa por un tiempo aleatorio entre 1 y 3 segundos para no sobrecargar el servidor`
	`  sleep_time = random.uniform(1, 3)`
	`  time.sleep(sleep_time)`

User-Agent	`headers = {`
	`  'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) '`
	`  'Chrome/89.0.4389.82 Safari/537.36 OPR/75.0.3969.14'`
	`}`

APIs	No se utilizan
Modularidad	Fichero main. Dos ficheros .py adicionals con 2 y 1 funciones.

Consideraciones página web

Indicación	Aplicación
Idioma	Inglés
[x] Sitio real	

Dataset

Inserta aquí la descripción del dataset utilizado en la práctica.

Vídeo

Inserta aquí el enlace al vídeo de presentación de la práctica.

Bibliografía utilizada

● Subirats, L., Calvo, M. (2018). Web Scraping. Editorial UOC. ● Masip, D. (2019). El lenguaje Python. Editorial UOC. ● Lawson, R. (2015). Web Scraping with Python. Packt Publishing Ltd. Chapter 2. Scraping the Data. ● Tutorial de GitHub https://guides.github.com/activities/hello-world.

- Herramienta Learn de PyCharm

Tabla de firmas

Ambos hemos contribuido en cada uno de los apartados.

Contribuciones	Firma Martínez	Firma Lima
Investigación previa	mmr	mlm

Contribuciones	Firma Martínez	Firma Lima
Redacción de las respuestas	mmr	mlm
Desarrollo del código	mmr	mlm
Participación en el vídeo	mmr	mlm