Tipología y ciclo de vida de los datos

Comparativa de portátiles 'El Corte Inglés'

Alicia Amores y Carlos Núñez

1. Contexto

Explicar en qué contexto se ha recolectado la información. Explicar por qué el sitio web elegido proporciona dicha información. Indicar la dirección del sitio web.

Cuando vas a comprar un portátil, y llega la época navideña, puedes quedar abrumado por las características técnicas del equipo, dado que si no entiendes un poco de hardware probablemente no signifiquen nada para ti y termines cayendo en las habituales «trampas» de marketing, con cifras astronómicas que no lo son tanto en realidad.

Puedes pasarte días y días comparando portátiles uno a uno con el objetivo de escoger aquel que sea más idóneo para ti. Y por ello hemos creado una herramienta para realizar una comparativa entre los portátiles ofertados en El Corte Ingles cuyo sitio web es https://www.elcorteingles.es/electronica/ordenadores/portatiles/

Creemos que este sitio web nos proporciona la información que necesitamos, ya que en su catálogo tiene una gran variedad de portátiles, con distintas características, marcas y precios. Además, otras páginas web de interés como MediaMarkt o Amazon, está restringido el webscraping o deben realizarse algoritmos más avanzados.

Otra posible aplicación de nuestro código sería, no solo la comparativa para un cliente final, sinó la comparativa entre empresas. Imagínate ser el propietario de una tienda de electrónica y quieres comparar los precios que ofrece tu competencia, en este caso el Corte Inglés, para así tomar decisiones a partir de los datos y así aumentar tus ganancias. Las aplicaciones son infinitas.

2. Título

Definir un título que sea descriptivo para el dataset.

Comparativa de portátiles 'El Corte Inglés'

3. Descripción del dataset

Desarrollar una descripción breve del conjunto de datos que se ha extraído. Es necesario que esta descripción tenga sentido con el título elegido.

El dataset cuenta con varios portátiles y sus características:

- Nombre
- Procesador
- RAM
- Disco Duro
- Tamaño de la Pantalla
- SO
- Tarjeta Gráfica
- Precio

4. Representación gráfica

Dibujar un esquema o diagrama que identifique el dataset visualmente y el proyecto elegido.

5. Contenido

Explicar los campos que incluye el dataset y el periodo de tiempo de los datos.

Los campos que se encuentran en el dataset son los siguientes:

- Nombre: Nombre del portátil
- Procesador: Modelo del procesador
- RAM: Capacidad de memoria RAM
- Disco Duro: Capacidad de memoria interna

- Tamaño de la Pantalla: Número de pulgadas de la pantalla
- SO: Sistema Operativo
- Tarjeta Gráfica: Tarjeta gráfica del portátil
- Precio: Precio de venta en El Corte Inglés.

Los datos son actuales.

6. Propietario

Presentar al propietario del conjunto de datos. Es necesario incluir citas de análisis anteriores o, en caso de no haberlas, justificar esta búsqueda con análisis similares. Justificar qué pasos se han seguido para actuar de acuerdo con los principios éticos y legales en el contexto del proyecto.

Los datos son propios del producto, aunque el precio es propio de 'El Corte Inglés'. En este caso es información pública, ya que en el propio sitio web no es necesario aceptar una serie de términos y condiciones, así que asumimos que el uso moderado de webscraping es adecuado.

"Este tema, cobró gran importancia en nuestro país allá en el año 2012, a raíz de la Sentencia del Tribunal Supremo de 9 de octubre, nº 572/2012, de RyanAir VS Atrápalo, en la cual se determinó que, en ese caso, el web scraping llevado a cabo por Atrápalo era legal. Si bien esta decisión señaló que no todo web scraping lo era, la importancia de esta sentencia radica en el hecho de que el Tribunal Supremo considerase legales las técnicas de web scraping, siempre y cuando, eso sí, se cumpliesen una serie de condiciones."

Podría considerarse ilegal en el caso de:

"Una eventual **violación de los términos legales y condiciones de uso establecidos por los titulares del website objeto de scraping**, desde el momento en el que los mismos sean aceptados por los usuarios que naveguen por la página web y tengan acceso a la información contenida en la misma."

Pero como no hay términos ni condiciones a aceptar, como hemos podido investigar, el webscraping puede realizarse. Puede leerse más en este enlace https://ecija.com/web-scraping-legal-ilegal/

7. Inspiración

Explicar por qué es interesante este conjunto de datos y qué preguntas se pretenden responder. Es necesario comparar con los análisis anteriores presentados en el apartado 6.

El siguiente conjunto de datos puede ser utilizado para comparar precios entre diferentes empresas, hacer un seguimiento de los productos de la competencia y tomar medidas para ganar mercado o decidir que portátil es más adecuado para un comprador particular. Como hemos visto, la captación de estos datos entra en el ámbito legal y todo el mundo puede acceder a ellos.

También podría ser utilizado para realizar modelos predictivos, data mining, como árboles de decisión o regresión, así poder predecir el precio de un portátil o ver que característica hace

aumentar su precio, (puede ser que por solo ser de Apple triplique el precio, aunque las otras características sean iguales).

8. Licencia

Seleccionar una licencia adecuada para el dataset resultante y justificar el motivo de su elección.

Se ha elegido la licencia de 'Released Under CC0: Public Domain License' ya que los datos están disponibles para todo el mundo, aunque se han recopilado para poder realizar una comparativa de forma sencilla. Por ello se ha decidido que el dataset resultante sea de dominio público.

9. Código

Aquí presentamos nuestro código:

```
# Importar módulos
import requests
import pandas as pd
from bs4 import BeautifulSoup
from time import sleep
from numpy import random

# Creamos listados vacíos para guardar los portátiles dependiendo de las características que tengan
elementos=[]
elementos_8=[]

contador=1

while(contador<18):
  # Para cada iteración creamos un enlace nuevo
  url = "https://www.elcorteingles.es/electronica/ordenadores/portatiles/"+ str(contador)+"/"
  # Ejecutar GET-Request
  response = requests.get(url)

  # Analizar sintácticamente el archivo HTML de BeautifulSoup del texto fuente
  html = BeautifulSoup(response.text, 'html.parser')

  # Creamos un listado de elementos
  lista= html.find_all('p', class_='product_preview-desc')
  # Creamos un listado de precios
  lista_precios=html.find_all('span', class_="price _big _sale")

  # Para cada par portátil, precio lo añadimos alos listados vacíos
  for element, precio in zip(lista, lista_precios):
    element=element.get_text().split(", ")
    element.append(precio.get_text())
    if len(element)>7 and len(element)<9:
      elementos_8.append(element)
    elif len(element)<8:
      elementos.append(element)

  contador+=1

  # Detenemos el código de forma aleatoria para no saturar el servidor
  sleep(random.randint(2,5))

# Creación de los datasets finales
df=pd.DataFrame(elementos, columns=['Nombre', 'Procesador', 'RAM', 'Disco Duro', 'Pantalla', 'SO', 'Precio'])
df_8=pd.DataFrame(elementos_8, columns=['Nombre', 'Procesador', 'RAM', 'Disco Duro', 'Tarjeta Grafica', 'Pantalla', 'SO', 'Precio'])
df_final = pd.concat([df, df_8])
df final.to csv('dataframe.csv',index=False, encoding='utf-8')
```

Nuestro código va iterando y navegando por cada una de las webs que se generan en cada iteración. Usamos la función requests y la respuesta la transformamos a un archivo html gracias a BeatifulSoup. Encontramos las clases que nos interesan, en este caso tuvimos que inspeccionar la web para encontrar a qué clases pertenecía la información que buscábamos.

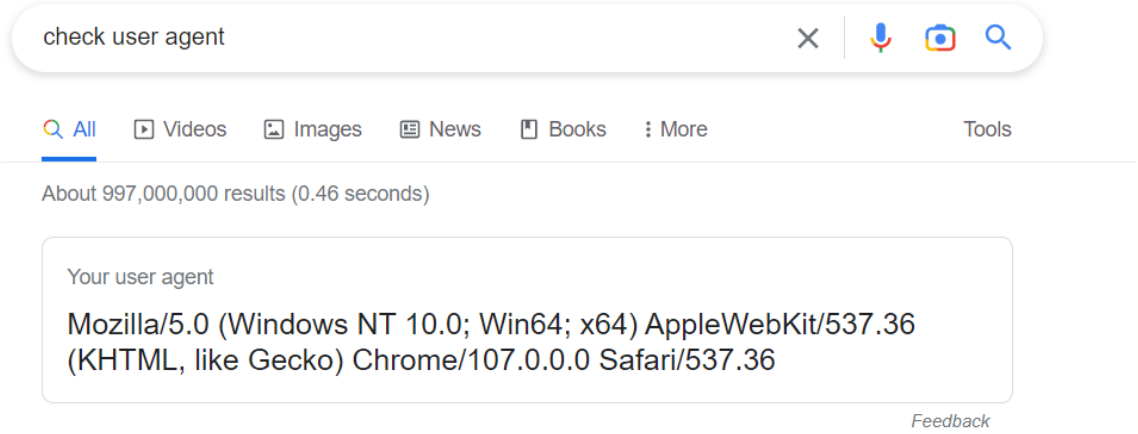
Después, para cada una de estas clases las vamos guardando en un listado, en concreto, las clases donde aparece la información del portátil la separamos por "," para obtener distintas columnas. Seguidamente aumentamos una unidad el contador para crear una nueva url y nos esperamos unos segundos, entre 2 y 5 de forma aleatoria, para no saturar el servidor y reproducir el comportamiento humano. El dataset resultante no ha sido limpiado, este se guardará para la Práctica 2.

El User-Agent utilizado por el código se puede obtener con la función `requests.utils.default_headers()`

Lo que nos ha devuelto lo siguiente:

```
{'User-Agent': 'python-requests/2.23.0', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-alive'}
```

En nuestro caso la propia librería de forma predeterminada ha establecido su propio *user agent*, en nuestro caso no ha sido necesario hacer modificaciones en nuestra cabecera HTTP pero de ser el caso podríamos usar nuestro *user agent* como se muestra a continuación



10.Dataset

El dataset resultante puede encontrarse en el siguiente enlace de Zenodo:
https://doi.org/10.5281/zenodo.7331774

11.Vídeo

Link de acceso al vídeo:
https://drive.google.com/file/d/1XFPWUcYnaQmQo1raUwPP1dLcauEh859D/view?usp=sharing

Contribuciones	Firma
Investigación previa	**AAS, CNA**
Redacción de las respuestas	**AAS, CNA**
Desarrollo del código	**AAS, CNA**
Participación en el vídeo	**AAS, CNA**