Creación paso a paso del web scraper

Ricardo García Espinosa

November 2024

1 Importar Librerías Necesarias

El primer paso consiste en importar las librerías requeridas para crear un web-scraper. Esto podría incluir:

- Selenium: Para interactuar con elementos dinámicos de una página web.
- Pandas: Para manejar y estructurar datos en tablas.
- Matplotlib o Seaborn: Para la visualización gráfica de los datos.
- Otras librerías según las necesidades específicas del proyecto, como BeautifulSoup para análisis HTML o requests para descargar contenido.

```python
import pandas as pd
from bs4 import BeautifulSoup
from urllib.request import urlopen
import urllib.request
import requests
import time
from multiprocessing import Process, Queue, Pool
import threading
import sys
import numpy as np
import re
#from random_user_agent.user_agent import UserAgent
from random_user_agent.params import SoftwareName, OperatingSystem
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from fake_useragent import UserAgent
from selenium.webdriver.chrome.options import Options
import pandasql as ps
from IPython.display import display,HTML
from datetime import date
from datetime import datetime
import matplotlib
import matplotlib.pyplot as plt
```

Figure 1: Librerías

2 Creación de Funciones para la Búsqueda

Aquí se desarrollan funciones para automatizar las búsquedas:

- Configuración del navegador: Descargar un chromedriver y configurar la ruta en tu equipo para que Selenium lo use.
- Extracción de información: Navegar por los sitios web, identificar y extraer los datos relevantes utilizando selectores CSS o XPath.

```
def func_seminuevos(marca):
                                        # decargar el web driver para que corra el código
    path ="/Users/richa/Downloads/chromedriver" # carga del web driver (asignar ruta donde se encuentra el driver)
    driver=webdriver.Chrome(path)
```

Figure 2: Función para búsqueda

3 Almacenamiento de Datos en Tablas

Una vez extraída la información (nombre, precio, etc.), esta se guarda en estructuras como listas o directamente en un DataFrame de Pandas. Para cada entrada:

- Se identifican las clases o etiquetas HTML que contienen la información deseada.
- Los datos se estructuran en columnas, como "Producto", "Precio", "Fecha", etc.

```
time.sleep(5)
url="https://www.cars.com/shopping/results/?dealer_id=&keyword="+marca
driver.get(url) # instruccion de obtener url parametrizada
time.sleep(10)

productos=driver.find_elements_by_class_name("vehicle-card-main.js-gallery-click-card")

# asignacion de nombres
lista_nombres=[]
for i in range(0,len(productos)):
    try:
        lista_nombres.append(productos[i].find_elements_by_class_name("vehicle-card-link.js-gallery-click-link")[0].text)
    except:
        lista_nombres.append(np.nan)


# asignacion de precios
lista_precios=[]
for i in range(0,len(productos)):
    try:
        lista_precios.append(productos[i].find_elements_by_class_name("primary-price")[0].text)
    except:
        lista_precios.append(np.nan)

# asignacion de mensualidad
lista_mens=[]
for i in range(0,len(productos)):
    try:
        lista_mens.append(productos[i].find_elements_by_class_name("estimated-monthly-payments-tooltip.js-tooltip-container")[0].text)
    except:
        lista_mens.append(np.nan)

today= date.today()

df_seminuevos =pd.DataFrame(columns=["MODELO","PRECIO","MENSUALIDAD"])
df_seminuevos["MODELO"] = lista_nombres
df_seminuevos["PRECIO"] = lista_precios
df_seminuevos["MENSUALIDAD"] = lista_mens
df_seminuevos["SITIO"] = "cars.com"
df_seminuevos["FECHA"] = str(today)

driver.quit()

return df_seminuevos
```

Figure 3: Almacenamiento de datos en tablas

4 Transformación de Datos

Antes de guardar los datos:

- Se realiza limpieza de valores vacíos (eliminación de filas o columnas sin datos).
- Se transforman datos a formatos adecuados, como convertir precios a números flotantes y ajustar monedas.

```
df_seminuevos_final.PRECIO = df_seminuevos_final.PRECIO.astype(float) # cast de datos
df_seminuevos_final.PRECIO = df_seminuevos_final.PRECIO*(19.53) #convertir de usd a mxn



df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace("/","")
df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace("mo","")
df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace("est","")
df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace("*","")
df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace(",","")
df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace("$","")
df_seminuevos_final.MENSUALIDAD = df_seminuevos_final.MENSUALIDAD.str.replace(".","")
```

Figure 4: Transformación de los datos

5 Concatenación y Creación de la Tabla Final

Si se han extraído datos de múltiples fuentes o búsquedas, estos se consolidan en una única tabla. Esto implica:

- Concatenar DataFrames.
- Asegurarse de que todas las columnas tengan valores consistentes.

```
df_tabla_final = pd.concat([df_seminuevos_final,df_usados_final,df_olx_final])
```

```
df_tabla_final
```

	MODELO	MARCA	PRECIO	MENSUALIDAD	SITIO	FECHA
0	2020 Honda HR-V Touring	HONDA	585841.41	8222.13	cars.com	2022-12-15
1	2021 Honda Insight Touring	HONDA	582579.90	8163.54	cars.com	2022-12-15
2	2019 Honda Pilot Touring 8-Passenger	HONDA	606289.32	8495.55	cars.com	2022-12-15
3	2017 Honda Accord EX w/Honda Sensing	HONDA	448994.70	6288.66	cars.com	2022-12-15
4	2013 Honda Pilot Touring	HONDA	348278.49	4882.50	cars.com	2022-12-15
...	...	...	...	...	...	...
35	2019, KIA Forte	FORD	323000.00	6284.00	olxautos.com	2022-12-15
36	2017, Ford EcoSport	FORD	275000.00	5350.00	olxautos.com	2022-12-15
37	2020, Ford EcoSport	FORD	380000.00	7393.00	olxautos.com	2022-12-15
38	2021, Ford ford-bronco	FORD	740000.00	14396.00	olxautos.com	2022-12-15
39	2020, KIA Forte	FORD	323000.00	6284.00	olxautos.com	2022-12-15

212 rows × 6 columns

Figure 5: Transformación de los datos

6 Generación de Gráficas

Para analizar los datos:

- Se calculan estadísticas descriptivas como medias y medianas.
- Usando *matplotlib* o *seaborn*, se crean gráficas que visualicen las tendencias. Por ejemplo, comparar el promedio de precios por marca.

Figure 6: Transformación de los datos

7 Consultas SQL

La limpieza de datos incluye prepararlos para consultas SQL:

- Se eliminan caracteres especiales (puntos, comas, etc.).
- Se guardan los datos en un formato adecuado para su uso posterior, como un archivo Excel o CSV.

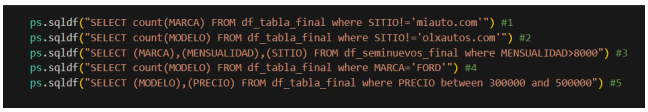

Figure 7: Transformación de los datos

8 Exportación de Datos

Finalmente, los datos se pueden guardar para compartirlos o utilizarlos en otros análisis:

Figure 8: Transformación de los datos