Step 0: Practical Web Scraping for Data Science

University: Tek-Up
Group: ING-4-SDIA

Tutor: Haythem Ghazouani

1 Introduction

Web Scraping is a fundamental skill for data scientists to acquire unique datasets that are not readily available in CSV format. In this tutorial, we will learn how to extract financial news to enrich a machine learning project.

2 Concrete Objectives & Use Cases

Scraping is never the end goal; it is the means to an end. Here are concrete Data Science objectives for this task:

- **Prediction (Sentiment Analysis):** Scraping headlines from `https://www.cnbc.com/finance/` to predict stock market trends or customer churn based on market sentiment.
- **Clustering:** Scraping diverse financial reports to group companies into clusters based on the topics discussed in their annual letters.
- **Anomaly Detection:** Scraping real-time exchange rates to detect unusual spikes in currency volatility.

3 Methodology

To scrape a website effectively, we follow these steps:

1. **Inspect:** Right-click on the webpage element and select "Inspect" to find the HTML tags and CSS classes.
2. **Request:** Use the `requests` library to fetch the HTML content.
3. **Parse:** Use `BeautifulSoup` to navigate the HTML tree.
4. **Store:** Save the structured data in a Pandas DataFrame and export to CSV.

4 Implementation Code

Below is the script `code/scraping.py` used to extract headlines from CNBC.

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd

def scrape_cnbc():
    url = "https://www.cnbc.com/finance/"
    headers = {"User-Agent": "Mozilla/5.0"}

    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.content, "html.parser")

    news_items = []
    cards = soup.find_all("div", class_="Card-textContent")

    for card in cards:
        title_tag = card.find("a", class_="Card-title")
        if title_tag:
            news_items.append({
                "headline": title_tag.text.strip(),
                "link": title_tag["href"]
            })

    return pd.DataFrame(news_items)

# Save results
df = scrape_cnbc()
df.to_csv("data/financial_news.csv", index=False)
```

5 Exercise

Try to modify the script to extract the **timestamps** of the articles. How would you use this time data to correlate news with the "Bank Customer Churn" dataset's timeline?