web_scraping

May 13, 2025

1 Web Scraping Exercise

Web Scraping allows you to gather large volumes of data from diverse and real-time online sources. This data can be crucial for enriching your datasets, filling in gaps, and providing current information that enhances the quality and relevance of your analysis. Web scraping enables you to collect data that might not be readily available through traditional APIs or databases, offering a competitive edge by incorporating unique and comprehensive insights. Moreover, it automates the data collection process, saving time and resources while ensuring a scalable approach to continuously updating and maintaining your datasets.

Ethical web scraping involves respecting website terms of service, avoiding overloading servers, and ensuring that the collected data is used responsibly and in compliance with privacy laws and regulations.

Use Python, `requests`, `BeautifulSoup` and/or `pandas` to scrape web data:

1.1 Import Libraries

[34]:
```
import json
from bs4 import BeautifulSoup
from urllib.request import Request, urlopen
import pandas as pd
```

1.2 Define the Target URL

[35]:
```
url = "https://polymarket.com/"
```

1.3 Send a Request to the Website

Do not forget to check the response status code

[36]:
```
req = Request(url, headers={"User-Agent": "Mozilla/5.0"})
response = urlopen(req)

print("Response received. Content-Type:", response.info().get_content_type())
```

```
Response received. Content-Type: text/html
```

1.4 Parse the HTML Content

Use a library to access the HTMl content

[37]:
```
bs = BeautifulSoup(response.read(), "html.parser")
print(bs.title.text)
```

```
Polymarket | The World's Largest Prediction Market
```

1.5 Identify the Data to be Scraped

Write a couple of sentence on the data you want to scrape

TODO: We want to scrape the topics, subtitles, and percentages of YES and NO shares on Polymarket, the largest prediction market platform. The data is loaded dynamically via JavaScript and is easily accessible through a structured JSON format. We store the date in a csv file and additionally save the whole html into a .html file.

1.6 Extract Data

Find specific elements and extract text or attributes from elements (handle pagination if necessary)

[38]:
```
script = bs.find("script", {"id": "__NEXT_DATA__"})
data = json.loads(script.string)
events = data["props"]["pageProps"]["dehydratedState"]\
            ["queries"][0]["state"]["data"]["pages"][0]["events"]

data_rows = []
for e in events:
    event_title = e["title"]
    for m in e["markets"]:
        question = m["question"]
        probs = m["outcomePrices"]
        yes = float(probs[0]) * 100
        no = float(probs[1]) * 100
        data_rows.append({
            "event": event_title,
            "question": question,
            "yes_percent": yes,
            "no_percent": no
        })

print(f"{len(data_rows)} rows collected.")
```

```
116 rows collected.
```

1.7 Store Data in a Structured Format

Give a brief overview of the data collected (e.g. count, fields, …)

[39]:
```
df = pd.DataFrame(data_rows)
print(df.head(3))
```

```
                                              event  \
0  US-China tariff agreement before 90 day deadline?
1                             May Inflation - Annual
2                             May Inflation - Annual

                                          question  yes_percent  no_percent
0  US-China tariff agreement before 90 day deadline?         74.5        25.5
1  Will annual inflation increase by 2.2% or less…           9.5        90.5
2     Will annual inflation increase by 2.3% in May?         25.0        75.0
```

1.8 Save the Data

[40]:
```
with open("polymarket.html", "w", encoding="utf-8") as f:
    f.write(str(bs))

df.to_csv("polymarket_events.csv", index=False, encoding="utf-8")
print(str(bs)[:1000])
```

```
<!DOCTYPE html>
<html id="__pm_html" lang="en"><head><meta charset="utf-8"/><meta
content="width=device-width" name="viewport"/><meta content="website"
property="og:type"/><meta content="summary_large_image"
name="twitter:card"/><meta content="@polymarket" name="twitter:site"/><meta
content="Polymarket is the world’s largest prediction market, allowing you to
say informed and profit from your knowledge by betting on future events across
various topics." name="description"/><title>Polymarket | The World's Largest
Prediction Market</title><meta content="Polymarket | The World's Largest
Prediction Market" name="title"/><meta content="Polymarket | The World's Largest
Prediction Market" property="og:title"/><meta content="Polymarket is the world’s
largest prediction market, allowing you to say informed and profit from your
knowledge by betting on future events across various topics."
property="og:description"/><meta
content="https://polymarket.com/images/homepage-twitter-card.png" property=
```