Kimia Teymouri Nik

300384851

Assignment #2

CSIS 4260-002

Instructor:

[Bhardwaj, Nikhil](mailto:bhardwajn@douglascollege.ca)

Contents

[1. Web Scraping 2](#_Toc193811688)

[Overview of Libraries 3](#_Toc193811689)

[Benchmark Result (Beautifulsoup VS. Playwright) 4](#_Toc193811690)

[Additional Research “Curiosity” 5](#_Toc193811691)

[Part 2. Text Analysis 5](#_Toc193811692)

# Web Scraping

What is web scraping?

Web scraping is the term that used for extraction data from a website. Collected information is converted to a format that is useful for users like spreadsheets or an API. Web scraping can be done manually but it’s not time-efficient and fast that’s the reason this task can be automated by python libraries for large data.

What topics are popular for web scraping?

* Real Estate Listing Scraping: real estate agency scrapes listings in different website to build an API that directly populates this information onto their website. By this way they become an agent for the property when someone finds listing on their site.
* Industry Statistics and Insights: some companies use web scraping to build comprehensive databases and draw industry-specific insights from these. They are technically selling their insights and information about a specific industry like oil prices, exports, and imports for selling to oil companies across the world.
* Shopping site comparison: some websites help users to compare prices of the same product from different retailers. This is also done by web scraping techniques.
* Lead Generation: to collect contact information about potential customers web scraping is used. This is very common in the business-to-business space.

## Overview of Libraries

1.Scrapy:

It is an open-source web scraping framework built in Python, designed to handle **large-scale web** scraping projects. It provides useful tools for scraping, parsing, and storing data.

Main Features:

* Fast and efficient for large-scale projects.
* For handling requests, responses, and data storage it has built-in support.
* For optimizing speed, it uses asynchronous requests.
* Crawling entire websites and following links is supported.

1. BeautifulSoup (with MechanicalSoup for browser automation):

BeautifulSoup is a Python library used for parsing HTML and XML documents. It’s often combined with “requests” for static web scraping (No dynamic module like event handlers). MechanicalSoup extends BeautifulSoup to automate interactions with web pages, such as filling out forms or clicking buttons.

Main Features:

* Simpler and more lightweight compared to Scrapy.
* Great for small to medium scale scraping projects.
* Works well with static websites but lacks native support for dynamic content (like JavaScript rendering).
* Easy to use.

1. Playwright (Browser Automation)

Playwright is a Node.js library used for browser automation. It can also be used for web scraping, particularly on JavaScript-heavy websites. It enables us to control headless (without GUI) browsers like Chrome, Firefox, and WebKit.

Main Features:

* Automate browser interactions for dynamic contents
* Useful for scraping modern website because it supports JavaScript rendering.
* It can handle popups, mouse movement, and advanced interactions.
* Supports multiple languages like Python, JavaScript, and C#.
* Best for content loading site via JavaScript.

**Use Case Recommendations:**

* Scrapy 🡺 best for large-scale scraping projects that require handling multiple pages (Large-Scale projects)
* BeautifulSoup 🡺 best for small to medium-scale tasks, especially for static websites (small and basic tasks).
* Playwright 🡺 Best for websites that rely on JavaScript for rendering content (dynamic content with JavaScript).

## Benchmark Result (Beautifulsoup VS. Playwright)

**I built a dashboard through streamlit, and the following screenshots are from my dashboard:**

**A blue rectangles and a rectangle

AI-generated content may be incorrect.**

A close-up of a computer screen

AI-generated content may be incorrect.

In this project, I compared two popular Python web scraping libraries, Beautiful Soup and Playwright. Using the same website, BooksToScrape. Both libraries were used to scrape 10 book entries including title, price, and availability.

Website Address:

<http://books.toscrape.com/>

The results showed a clear difference in performance:

* BeautifulSoup completed the task in 0.45 seconds.
* Playwright completed the same task in approximately 5.59 seconds.

While BeautifulSoup was significantly faster due to its simplicity and ability to handle static HTML efficiently, Playwright offers greater flexibility for scraping modern, JavaScript-heavy websites.

Based on these findings, I decided to proceed with Playwright for the next phase of my assignment scraping 100 articles related to real estate in Canada from a subreddit. Since Reddit uses **dynamic content loaded via JavaScript**, Playwright is better suited for handling such pages and collecting accurate data.

## Additional Research “Curiosity”

A close-up of a menu

AI-generated content may be incorrect.

# Part 2. Text Analysis

For this assignment, I collected 100 posts from Reddit using Playwright to scrape the Old Reddit interface. I specifically chose Old Reddit because it has a more consistent and scrape-friendly HTML structure compared to the newer version. The focus of my analysis was the Real Estate industry, covering a wide range of topics related to housing, renting, buying, and investment discussions. This allowed me to gather organic user opinions and sentiments in an unfiltered environment, giving a rich base for natural language analysis.

To analyze the collected data, I used TextRank for extractive summarization (via the summa library) and VADER Sentiment Analyzer (from NLTK) to evaluate the emotional tone of each post. I chose these algorithms because they are free, well-documented, and work efficiently without the need for training on additional labeled data. Additionally, I implemented CountVectorizer and TF-IDF from sklearn to identify the most frequently mentioned and most informative terms across the corpus. Each post was scored for importance based on a combination of summary length and sentiment strength. In the following section, you will find visualizations generated from this analysis, along with brief explanations for each figure.

Screenshots:

A graph of purple rectangular objects

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A graph with a line and a graph

AI-generated content may be incorrect.

Many posts lean toward a strongly positive sentiment (scores near +1), indicating that discussions in the real estate subreddit are generally optimistic or hopeful.

A graph with different colored squares

AI-generated content may be incorrect.

The results show that most real estate discussions on Reddit are **positive**, while negative and neutral sentiments appear in significantly smaller portions.

**“analysis\_output.csv” dataset:**

The importance score was derived using a custom formula that combines the length of the summary (as a proxy for content richness) and the absolute value of the VADER sentiment score (as a measure of emotional intensity). To ensure scores are comparable across posts, a normalized importance score was calculated by dividing each importance score by the highest score in the dataset, scaling values between 0 and 1. This allows us to easily identify which posts were both emotionally charged and content-rich — highlighting them as more “important” in the context of user sentiment and information density.

A screenshot of a spreadsheet

AI-generated content may be incorrect.