

## **Phase 2:**

### **Literature Review and Data Description**

#### **Introduction:**

Web scraping refers to the process of extracting data from websites. The extracted data can be used for various purposes, including data analysis, data mining or even for building machine learning models. Python is a popular programming language for web scraping, and there are several libraries available that make web scraping in Python easier. Two of the most used libraries for web scraping in Python are BeautifulSoup and Selenium. In this literature review, we will discuss the basics of web scraping using BeautifulSoup and Selenium.

#### **Literature Review:**

For this project, I identified several reputable journals in the field of Web Scraping, including Web Scarping tools and Techniques in Python with After identifying the journals, I conducted a search for peer-reviewed articles related to the topic of interest. I filtered out at least 16 relevant articles based on the title and abstract.

#### **References:**

(Ashouri et al., 2022; Bricongne et al., 2023; Chaitanya et al., 2023; Chanda & A, 2020; De & Sirisuriya, 2015; Dewi et al., 2019; Han & Anderson, 2021; Khder, 2021a, 2021b; Eloisa Vargiu, Xiaofei Liao, Hai Jin, Yunhao Liu, Lionel M. Ni, and Dafu Deng, Mirko Urru1, n.d.; A Accetturo, A. R Lamorgese, S Mocetti, D Pellegrino, Housing supply elasticity and growth: evidence from Italian cities, Journal of Economic Geography, Volume 21, Issue 3, May 2021, Pages 367–396.;M et al., 2018; Muehlethaler & Albert, 2021; Nagoya Kōgyō Daigaku & Institute of Electrical and Electronics Engineers, n.d.; Niu et al., 2023; Polidoro et al., 2015; Zhao, 2017)

#### **Description:**

##### **Beautiful Soup:**

Beautiful Soup is a Python library used for web scraping. It is used to parse HTML and XML documents and provides a simple way to extract data from them. BeautifulSoup is a powerful library widely used by web scraping professionals. It is simple to use and has many features. The pip package installer for Python can be used to install the library.

To begin using BeautifulSoup, we must first import it into our Python script. After we import the library, we can use it to parse HTML and XML documents. The library provides several methods for parsing HTML and XML documents, including the `parse()`, `prettify()`, and `find_all()` methods.

## Selenium:

Another popular Python library for web scraping is Selenium. It is used to automate web browsers and can extract data from websites that require user interaction, such as those requiring login credentials. Selenium can be used to interact with a webpage as if it were a user. Pip can be used to install the library.

To begin using Selenium, we must first install a web driver for the browser we wish to automate. To automate the Chrome browser, for example, we must first download and install the Chrome Driver. After installing the web driver, we can use Selenium to automate the browser.

## Drawbacks of Selenium:

Selenium is a popular tool for automating web browsers, including web scraping tasks that require interaction with JavaScript-driven websites. However, as with any tool, it has some drawbacks that you should be aware of before using it:

- **Slower than other web scraping libraries:** Because it requires starting and interacting with a web browser instance, Selenium can be slower than other web scraping libraries such as BeautifulSoup and Requests-HTML. This can lengthen the time required for each scrape, especially if many pages need to be scrapped.
- **Excessive resource consumption:** Running a web browser instance can be time-consuming, especially when dealing with large-scale web scraping projects. This can result in slower performance and higher memory usage on the Selenium machine.
- **Knowledge of Web Browser Automation:** Using Selenium effectively necessitates some understanding of web browser automation and web page structure. This can be a barrier for those who are new to web development or have no prior experience.
- **Limited support for some websites:** Some websites use CAPTCHA challenges or IP blocking to prevent automated access. Selenium may be unable to circumvent these safeguards and may be barred from accessing the website.

Overall, Selenium is a powerful tool for web scraping tasks that require interaction with JavaScript-driven websites, but it has some drawbacks to consider. It's important to weigh the benefits and drawbacks of using Selenium.

## Drawbacks of BeautifulSoup:

Beautiful Soup is a popular Python library used for web scraping and parsing HTML and XML documents. However, like any tool, it has some drawbacks that you should be aware of before using it:

- **Limited web crawling support:** BeautifulSoup is primarily designed for parsing HTML and XML documents, and it does not have built-in support for crawling multiple pages or websites. It can be used in combination with other libraries like Requests to crawl websites, but it may require more configuration and setup.
- **Limited support for JavaScript rendering:** BeautifulSoup does not have built-in support for rendering JavaScript, which means that it may not be able to scrape data from some JavaScript-driven websites.

- **Parsing errors:** Sometimes, HTML and XML documents can have errors that can cause issues with parsing using BeautifulSoup. While the library tries to handle errors gracefully, it may still encounter issues in some cases.
- **Requires understanding of HTML structure:** BeautifulSoup requires some understanding of the structure of HTML and XML documents to be used effectively. This can be a barrier for beginners or those without experience in web development.
- **Performance:** For large-scale web scraping projects, BeautifulSoup may not be the most performant option due to its pure Python implementation. In comparison, some other libraries like lxml have faster parsing speeds.

Overall, BeautifulSoup is a powerful tool for parsing HTML and XML documents and can be used effectively for many web scraping projects. However, it has some limitations to consider, especially for larger-scale web crawling and scraping projects that require more advanced features like JavaScript rendering and faster parsing speeds.

### **Challenges and Limitations of Web Scraping:**

Web scraping is a powerful tool for extracting data from the internet. However, there are several obstacles and constraints. One difficulty is that web pages change all the time. This means you may need to update your web scrap code on a regular basis to ensure that it is still functional.

Another issue is that some websites may prohibit web scraping. This is since web scraping can be used to extract data from websites without the website owner's permission.

Finally, web scraping can be both time-consuming and costly. This is since you may need to use a variety of different tools and libraries to extract data from various websites.

### **Research Paper Descriptions:**

A recent paper by Shreesha M, Srikara S B , Manjesh R. (2021) Intelligent Smart Parser is a web-based application that displays news from different sources on a single platform. It uses web scraping and natural language processing techniques to retrieve and summarize news articles into three to four lines of text. This helps users quickly grasp the main points of a news story. The application also displays news based on the user's location and trending news based on views. Overall, Intelligent Smart Parser simplifies the process of accessing news and helps users save time. It is a useful tool for those who want to stay informed about the latest events and developments happening around the world.

Another recent paper by Bricongne, Jean Charles. (2023) utilized BeautifulSoup for web scraping of news articles related to COVID-19 from Chinese news websites. The study aimed to analyze the sentiment of the news articles and identify the topics discussed in the articles. The authors used BeautifulSoup to extract the title, content, date, and source of the news articles. The extracted data were then analyzed using sentiment analysis and topic modeling techniques.

Another paper by Chanda, Siddhant Vinayak. (2020) use of web scraping technologies to extract financial data from Yahoo Finance using a Python model. The proposed model provides an economical, reliable, and time-efficient tool for business analysts who require quantitative outputs for analysis and decision-making. The web application offers a user-friendly interface to facilitate the data extraction process and minimize human error. The paper also highlights the importance of respecting data privacy laws and ethical web scraping practices.

Another paper by Thivaharan. S (2020) compares three popular Python libraries used for web scraping: beautiful Soup, LXML, and RegEx. A statistical study found that RegEx is faster on average but has limited rule extraction capabilities. beautiful Soup and LXML are slower but better suited for extracting web content in complex environments and are based on the DOM model. The paper also notes that web scrapers influence modern content grading systems for regional languages in social media. Overall, RegEx is best suited for simple contexts while beautiful Soup and LXML are more scalable for complex environments.

### **Conclusion:**

According to the literature review, web scraping with Python using BeautifulSoup and Selenium has become an important tool for data extraction and analysis in a variety of fields, including finance, e-commerce, and social media. BeautifulSoup and Selenium are popular web scraping libraries, with BeautifulSoup being better suited for parsing HTML and XML files and Selenium being useful for automating web browsers. However, web scraping's legal and ethical implications should not be overlooked, and appropriate steps should be taken to ensure compliance with data privacy laws and ethical practices. Overall, Python's web scraping capabilities have opened new avenues for data-driven decision-making, and its potential for streamlining complex data extraction processes is significant.