2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) | 979-8-3503-2579-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAISS58487.2023.10250745

An Extensive Review on Web Scraping Technique using Python

Rahul Chauhan
Computer Science Engineering
Graphic era Hill University
Dehradun, Uttarakhand, India
chauhan14853@gmail.com

Ayush Negi
Computer Science Engineering
Graphic era Hill University
Dehradun, Uttarakhand, India
2002negiayush@gmail.com

Mahesh Manchanda
Computer Science Engineering
Graphic era Hill University
Dehradun, Uttarakhand, India
manchandamahesh@gmail.com

***Abstract*— The goal of web scraping or data scraping is not only to search for or extract data from websites, but also to extract data in a systematic manner and make better decisions. Choose wisely and more. This can be one of the most difficult problems for users to overcome, as the internet is the largest source of knowledge, yet information found on the web cannot be used directly for data analysis and other processes because the language is not necessary. In order to get. It is vital information in a short period of time. Web scraping employs the usage of a machine, such as a computer, to browse a web page, extract data from the web page, and then store the extracted data. It is critical to exercise caution and ensure your safety.**

Keywords—Web scrapping, Python, Data Analysis, Data mining, websites.

I. Introduction

Data analysis is the process of generating solutions by querying and interpreting data. The analytical process includes identifying the problem, addressing the availability of appropriate information, identifying which methods can help find a comprehensive solution, and communicating the results. For analysis, data must be divided into several levels, for example, specification, organization, maintenance, replication, use of models and algorithms, and finally results. Web scraping and crowdfunding are great strategies for creating web content. Many people use these ideas in research and business to create products or create criticism, to spread the truth in the advertising industry, and to get people to raise money to improve the business [1]. Web scraping is called "screen scraping", "network data extraction".

Web Scrubber programming aims to get all the important information from different online stores and mines and bring them together in a new website. The web crawling tool is used to extract information from the web and is part of web usage, web mining and data mining, online price change analysis and price matching quality, search terms (see contest), record land records, weather observation data, page traffic, comments, social media and reputation, network links and network links information. Pages are created using extended content (HTML and XHTML) an often have information integrated into content.

This is because most web pages are at least designed for human end users, not robots. This creates a box for accessing network data. A web scraper is an API for scraping data from a site. Consortia such as Amazon AWS and Google offer web scraping tools, organizations, and open-source data to end users for free. Since this article will focus on the advantages of using Python as a programming language for data analysis, it is a good choice as a language for data centric applications, and the Python version used for this will be Python 3.

The abundance of information on the World Wide Web is both a blessing and a curse. A lot of useful information is provided to be the best; however, this information is difficult to obtain. Much of the information on the web is presented in the form of HTML documents, a graphic format that communicates with people rather than computers, and such information is unnecessary.

The increase in information on the Internet is increasing rapidly every year and most of the information is unnecessary. Data clutter is not easy as redundant data does not fit into data structures. It's our job to get more information from the Internet and save it for future work. This extraction can be done using a software solution called Web Scraping Software. We offer a cloud-based scraping architecture to retrieve redundant data from the web.

Data Collection is a scientific research and development exercise in which we obtain data primarily from private sources (such as company sales records, financial data) or public sources (such as magazines, sites such as open data) or through purchasing data [2].

Again it is Web Scrapper technology or simply copying information from a web browser. The concept of scraping can be divided and studied into web builder automation, parser, filtering, and formatting and storage of results. Likewise, in distributed computing, processing of large numbers of fetch requests can be learned. It can be built using many libraries such as Scrappers, BeautifulSoup, Scrapy, and Request, but they only work on static pages [3].

We choose Selenium and Web Driver API as tools for automating web pages, especially for web application testing. Similarly, Python's HTML Parser library is the basis for parsing HTML formatted documents, and XPath, a language that uses XML syntax to find the content of web pages, is used to extract data from HTML content downloads using the Request library. This call is from a traveler. We extract our data with Crawler and save it in CSV format. Web scraping is the process of extracting information from websites through scraping software. Many people and organizations use scanners, and many companies have now developed their own scanner tools to extract data.

There are many reasons to use Scarper for your competitor's marketing and pricing research, tracking any changes, and analyzing key data. Extracting data with the help of this tool is the easiest process [4]. It reduces labor and saves time. It is an effective and powerful technique for collecting large amounts of information. Now online scraping techniques have been adapted to suit many fields, from human services to full-scale systems that can turn entire web pages into good information.

APPLICATIONS

The use of web scraping is a multifarious affair as it serves many purposes in different industries. Some uses of web scraping include:

- Data collection and compilation: Web scraping automatically extracts data from websites, allowing businesses and researchers to collect large or redundant data for review.
- Market Research and Competitive Analysis: Web scraping provides insight into the market, competitive strategies, and customer preferences.
- Lead Generation: Web scraping can extract contact information from websites, helping businesses generate leads and create customer databases.
- Price monitoring and comparison: E-commerce companies can monitor their competitors' prices using the website and adjust their pricing strategies accordingly.
- Sentiment Analysis and Brand Monitoring: Web scraping can be used to gather data from social media platforms, forums, and review sites to analyze customer sentiment and opinions about a particular brand or product.
- Content Gathering and News Tracking: Web scraping helps gather news, blog posts, and other online content from a variety of sources. News organizations can use this information to organize content, monitor news updates, and analyze emerging trends [5].
- Academic Research and Data Analysis: Researchers can use web scraping to collect data for research studies, analyze social media, conduct social analysis thinking, track research data, and collect data for analysis.
- Machine Learning and Artificial Intelligence Training: Web scraping is used to collect data to train machine learning models and artificial intelligence algorithms. By combining disparate data, researchers can create powerful models that can predict, classify, or perform tasks in good language [6].

II. LITERATURE SURVEY

Web scraping is an effective way to extract harmless information from a website and convert that information into information that can be stored and analyzed in a database. Mesh scraping is also called mesh scraping, mesh scraping, mesh picking or screen scraping. Web scraping is a form of data mining [7]. The purpose of web scraping is to extract data from the web and convert it into a readable format such as a spreadsheet, database, or single file (CSV), as shown in Figure 1. Information such as selling price and stock price. Web scraping allows you to gather a variety of information about a product, such as market price and details. Extracting targeted information from your website helps your business make better decisions.

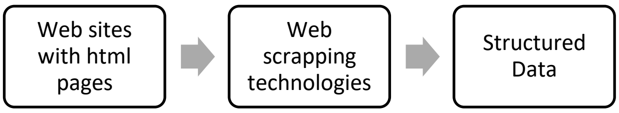


Fig. 1 Process of web scrapping

To understand how the data extraction process has changed, it is important to understand the process involved in web scraping, which is almost the same as it used to be on the web. The impact behind web scraping business is the confidence to get business benefits easily, along with things like making a competitor special price, getting ahead, managing transactions, modifying APIs, and stealing data in and out. Major meetings and check engines seem to be dealing with the impact of the web marketing boom and it didn't have much of an impact until legal issues in the mid-2000s. Early browsers were pretty simple physically reordering all the crap on the web. Grab is an upgrade of the Unix grep command or standard interface manager when software engineers arrive, troubleshoot HTTP requests with communication, define sites with programming files, and specify sites with question papers. But today the situation is very different: web scraping is big business that requires powerful tools and management to manage.

Data extraction and analysis is often used by digital distributors and catalogues, travel, real estate and e-commerce. But with the advancements in Real Databases, storage and innovation in analysis and recomputing: data is seen as data and processed as data for data analysis. An important change was the emergence of the RDB (Relational Database) in the 1980s, which allowed users to create Sequels (SQL) to retrieve information from the database [8].

The advantage of RDB and SQL for customers is the ability to separate their data. It simplifies the way we acquire knowledge and continually use knowledge. Data Warehouse: Unlike traditional relational data, data warehouses are often easy to improve response time to queries. Advances in data mining have led to an understanding of data and data storage systems that allow organizations to collect more data and classify it as needed. Based on the evaluation of the generated data, a business model was created in which managers began to "see" the needs of customers [9].

Data analysis shows that it is important to know the data if you focus on relevant questions. Public research through provision of plans, creation of statistics and reconstruction, etc.

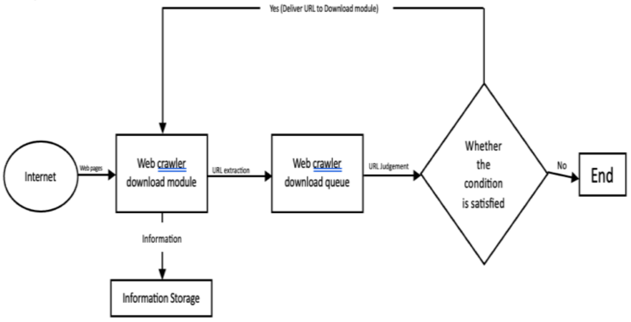

Yes (Deliver URL to Download module)
Internet
Web pages
Web crawler download module
URL extraction
Web crawler download queue
URL Judgement
Whether the condition is satisfied
No
End
Information
Information Storage

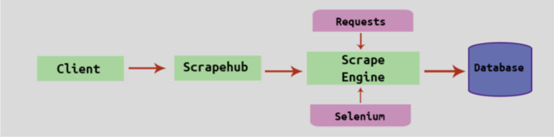
Fig.2 Framework of scraping process

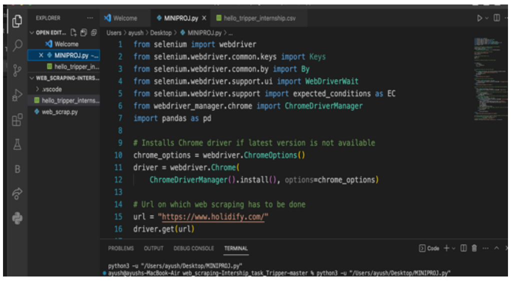
III. METHODOLOGY

The project's methodology entails gathering all the data that has been extracted from various sources using the web crawler Scrapy's vivid features and Python scripts, then further analysing it to meet the client's needs before the data is stored in the business's database [10]. As we analyse the process using specific code and provide the desired url for the iteration to execute for scraping the data from the source url, the web crawler scrapy, which is based on Python, may also help us obtain the intended result.

Selenium

Selenium is a popular open-source tool used for automating web browsers. It provides a programming interface for interacting with web pages and performing various actions like clicking buttons, filling out forms, and extracting data. Selenium supports multiple programming languages, including Python [11].
To use Selenium with Python, you'll need to install the Selenium library and a web driver specific to the browser you want to automate (such as Chrome, Firefox, or Safari).



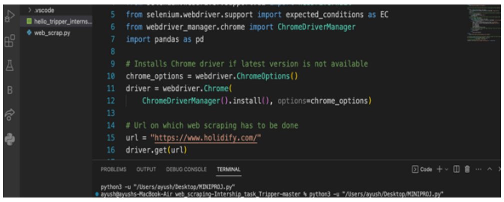


```python
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd

# Installs Chrome driver if latest version is not available
chrome_options = webdriver.ChromeOptions()
driver = webdriver.Chrome(
    ChromeDriverManager().install(), options=chrome_options)

# Url on which web scraping has to be done
url = "https://www.holidify.com/"
driver.get(url)
```

```
python3 -u "/Users/ayush/Desktop/MINIPROJ.py"
ayush@ayushs-MacBook-Air web_scraping-Intership_task_Tripper-master % python3 -u "/Users/ayush/Desktop/MINIPROJ.py"
```

Pandas

Pandas is a popular open-source Python library used for data manipulation and analysis. It provides easy-to-use data structures and data analysis tools, making it efficient and convenient to work with structured data.

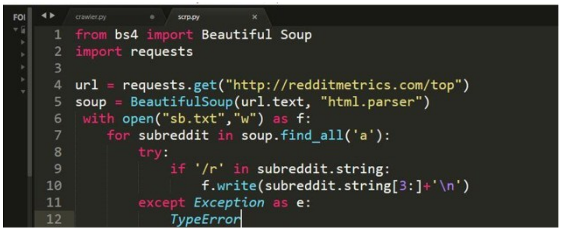


```python
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd

# Installs Chrome driver if latest version is not available
chrome_options = webdriver.ChromeOptions()
driver = webdriver.Chrome(
    ChromeDriverManager().install(), options=chrome_options)

# Url on which web scraping has to be done
url = "https://www.holidify.com/"
driver.get(url)
```

```
python3 -u "/Users/ayush/Desktop/MINIPROJ.py"
ayush@ayushs-MacBook-Air web_scraping-Intership_task_Tripper-master % python3 -u "/Users/ayush/Desktop/MINIPROJ.py"
```

BeautifulSoup

Beautiful Soup is a popular Python library used for web scraping. It helps extract data from HTML and XML documents by providing convenient



```python
from bs4 import Beautiful Soup
import requests

url = requests.get("http://redditmetrics.com/top")
soup = BeautifulSoup(url.text, "html.parser")
 with open("sb.txt","w") as f:
    for subreddit in soup.find_all('a'):
        try:
            if '/r' in subreddit.string:
                f.write(subreddit.string[3:]+'\n')
        except Exception as e:
            TypeError
```

Choosing Web Scraping Framework Specify and select the appropriate Python web scraping framework/library. Popular options include BeautifulSoup, Scrapy, and Selenium. The choice is favorable based on features such as ease of use, flexibility, and compatibility with research requirements.

1. Target Website ID

Identifies the website from which the information should be discarded. Consider factors such as the impact, accessibility, and usability of information on these websites.

2. Ethical Considerations

discusses ethical considerations related to web scraping, including compliance with the website's terms of service, respect for the robots.txt file, and being aware of proprietary information and intelligence. Emphasize the importance of obtaining the appropriate license, if any.

3. Data Extraction Policy

Defines the strategy for extracting data from the target website. Check for specific web pages (eg HTML markup, CSS selectors) that contain the required information. Consider different methods such as scraping static HTML pages, interacting with JavaScript rendered pages, or using APIs.

4. Implementation of Web Scraping

Code Implementation of web scraping code using selected frameworks/libraries. Explains rules procedures and standards. Add code snippets or pseudocode to explain the important steps involved in scraping.

5. Interfering with mechanisms to prevent scratching

Addresses the ability to protect tools such as captchas, IP blocking or user agent detection used by websites. Discuss strategies or tools that can be used to resolve these issues, such as using names, changing user agents, or using CAPTCHA resolvers.

6. Data Cleaning and Preprocessing

Provides details of steps for cleaning and preprocessing scraped data. This may include removing HTML markup, processing missing or incorrect data, normalizing the text, and converting the data into a format suitable for analysis.

7. Data Storage and Management

Explains the storage and management strategy for scraped data. Discuss whether data is stored in a database, file system, or memory. Consider data security, backup procedures, and data recovery strategies for effective data management.

8. Evaluation and Evaluation

Defines criteria for evaluating the performance and reliability of web scraping solutions. Discuss test cases, test data, and metrics such as data removal speed, accuracy, and ability to manage different versions of the website.

9. Scalability and Performance Optimization

discusses strategies for improving the scalability and performance of web scraping solutions. Explore methods such as clustering, distributed fetch, or caching to optimize the retrieval process. Documentation and User's Guide creates detailed documentation and instructions for developing web scraping solutions. It includes setup, configuration, and usage instructions, as well as math formulas and troubleshooting tips.

10. Ethical Considerations and Limitations

Review the ethical considerations of web scraping and discuss any limitations or biases in the information obtained. It addresses the potential impact of logging on website performance and server load and recommends mitigation strategies that can be implemented.

11. Research and Analysis

Displays target website access test results. Add statistics, visualizations or other methods to show the effectiveness of solutions.

12. Discussion and Conclusion

Analyze the findings, discuss the strengths and weaknesses of mesh scraping solutions, and evaluate their success in meeting these research goals.

Testing

I tested the project using the various previously mentioned components and got it work in a browser. The performed extraction proves to be entirely pertinent, and the estimated analysis is done.

Python-based web scraping has many benefits, the ethical and legal considerations associated with this practice must be acknowledged.
Adhering to the website's terms of service, monitoring the robots.txt file, and avoiding excessive or destructive access are the keys to maintaining ethical access. In addition, understanding legal processes such as copyright law and data protection laws is very important to avoid legal problems [12].

Despite its advantages and potential, network scraping with Python has some limitations. Websites update their standards frequently, causing them to break or conflict with the script.
Maintaining and updating your login credentials regularly to accommodate these changes can be time- and resource-intensive. In addition, websites may use access protection systems such as CAPTCHA or IP blocking, which may block the login process or require additional strategies to overcome.

In conclusion, our research demonstrates the effectiveness and versatility of Python-based web scraping techniques. Python's user-friendly syntax combined with powerful libraries allows researchers to extract valuable information from many websites. The potential uses of web scraping with Python are vast and include market research, sentiment analysis and data analysis.
However, researchers need to think carefully about ethical and legal aspects and be prepared for issues such as field change and preventive measures. Future research in this area will focus on developing new web application techniques, exploring the integration of machine learning algorithms to improve data extraction, etc. It can focus on the ethical and legal evolution of web scraping.

IV. CONCLUSION

In conclusion, this research article explores the topic of web scraping with Python and demonstrates its importance and potential applications. Throughout the research, we discuss the various techniques and tools available in web scraping, highlighting Python's potential and strengths in this area.
Additionally, we address the challenges and limitations of web scraping such as dynamic pages, captchas, and IP blocking. These limitations require careful consideration and appropriate strategies to overcome, including the use of nameservers, user rotation, and CAPTCHA resolution services [13].
However, while the book ensures the integrity and reliability of the recorded material, it is important to properly manage the website and respect the laws and privacy limits.
As technology and the web continue to evolve, it is important for practitioners to keep up with new tools, techniques and legal procedures in order to do well in the field of beautiful web scraping. With continued research and responsible practice, web scraping with Python will undoubtedly continue to be an essential tool for knowledge discovery and advancement in the digital age [14].

V. FUTURE WORK

In summary, this research article explores the topic of web scraping with Python and demonstrates its importance and potential applications. Throughout the research, we discussed the various techniques and tools available in web scraping, highlighting Python's capabilities and advantages in this area.
Additionally, we address the challenges and limitations of web scraping such as dynamic pages, captchas, and IP blocking. These limitations require careful attention and resolution of appropriate strategies, including the use of nameservers, user rotation, and CAPTCHA resolution services [15].

However, while this guide ensures the integrity and reliability of the data recorded, it is important that the site is properly managed and that legal and technical constraints and privacy are respected.

As technology and the web continue to evolve, healthcare providers need to keep up with new tools, techniques and legal frameworks to be successful in the ever-changing world of web scraping. With continued research and responsible practice, web scraping with Python will undoubtedly continue to be an essential tool for the pursuit of knowledge and progress in the digital age.

The future work outlined in this research paper aims to advance web scraping techniques for extracting data from Flipkart. By addressing challenges related to dynamic web pages, improving performance, overcoming anti-scraping mechanisms, enhancing data extraction capabilities, ensuring data quality, and considering ethical considerations, researchers can contribute to the development of robust and responsible web scraping practices. These advancements will facilitate the extraction of valuable insights from Flipkart, enabling researchers to make informed decisions and provide valuable contributions to various domains such as e-commerce analysis, market research, and consumer behavior studies.

REFERENCES

[1] David Mathew Thomas, Sandeep Mathur. "Data Analysis by Web Scraping using Python", 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019.

[2] Ghazvinian, Holbert, Viswanathan. “Simple WebScraping.”Internet:https://seanholbert.wordpress.com/2011/07/15 /scrappy-simple-web- scraping/, Jun. 2015.

[3] Bellarosey.“Crowdsourcing-Definition. Internet:http:// crowdsourcing. typepad.com/cs/ 2006/06/crowdsourcing_a.html, Jun. 02, 2006.

[4] Naveen Ashish and Craig Knoblock. ”Wrapper Generation for semi-structured Internet Sources. In Proc” ACM SIGMOD Workshop on Management of Semi Structured Data, Tucson, Arizona, May 1997.

[5] Datahen.3 Advantages of web scraping for yourenterprise"Internet:https://www.datahen.c om/3-advantages-web-scraping-enterprise/,May.17,2017.

[6] Pythonversion3.6,http://www.python.org.7.”Kengtel,W:Wagner,M.Proteins1999,37,334- 345.

[7] BrightPLanet.com Deep web White Paper. http://www.completeplanet.com/Tutorials/Dee pWeb/index.asp.

[8] Wu, L., Mattila, A. S., Wang, C. Y., & Hanks, L. (2015). The impact of power on service customers’ willingness to post online reviews. Journal of Service Research, 19(2), 224–238.

[9] Anand V. Saurkar, Kedar G. Pathare, Shweta A. Gode, “An overview of web scraping techniques and tools” International Journal on Future Revolution in Computer Science and Communication Engineering, April 2018.

[10] O’Reilly, S. (2006). Nominative fair use and Internet aggregators: Copyright and trademark challenges posed by bots, web crawlers and screen-scraping technologies. Loyola Consumer Law Review, 19, 273.

[11] Landers, R. N., Brusso, R. C., Cavanaugh, K. J., & Collmus, A. B. (2016). A primer on theory-driven web scraping: Automatic extraction of big data from the internet for use in psychological research. Psychological Methods, 21(4), 475–492.

[12] Vanden Broucke, S., & Baesens, B. (2018). Practical web scraping for data science. Apress.

[13] Doran, D., & Gokhale, S. S. (2011). Web robot detection techniques: Overview and limitations. Data Mining and Knowledge Discovery, 22(1), 183–210.

[14] Ram Sharan Chaulagain, Santosh Pandey, Sadhu Ram Basnet, Subarna Shakya. "Cloud Based Web Scraping for Big Data Applications", 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017