2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET) | 979-8-3503-2919-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSEIET58677.2023.10303328

Unlocking the market insight potential of data extraction using Python-Based web scraping on Flipkart

Kavya Singh Dogra
Computer Science Engineering
Graphic Era Hill University
Dehradun, Uttarakhand India
kavyasinghdogra19@gmail.com

Nidhi Nirwan
G.L Bajaj Institute of management
Greater Noida, Uttar pradesh, India
nidhi.chauhan20@gmail.com

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

Abstract- Data scraping is not only to scrounge or extract data from websites but extracting data in an organized structured format so that it can be used for data analysis, collecting business data, enabling researchers and businesses to gain better insights and make informed choices etc. Since internet is the biggest storehouse of data but the data available on the web cannot be directly used for data analysis purposes and other techniques because of its unstructured form and this can be one of the biggest challenge users need to overcome in order to gain valuable information within a limited time frame.

The process of web scraping requires the usage of machines just as computers to access web, extract data from its pages and then to store the extracted information. Also it is important to be careful and make sure these tools are used correctly because there are some websites that may prohibit website downloading and may even consider web scraping as illegal.

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

I. INTRODUCTION

Web scraping may be used to extract useful and valuable data from the WWW. Programming languages and tools may be used to perform this data extraction. Python is a popular programming language for this. Due to its simple syntax, portability, and availability of many libraries, it is regarded as one of the finest languages for web development. With the help of these features, Python enables academics and organizations to search the web, collect pertinent data, and convert it into designs without any difficulty. They receive the data from it that they need for analysis and decision-making. [1] The ideas and methods that go into web scraping with Python include using related libraries, exploring the web, and obtaining the right data [2].

Scraping is crucial for transforming unstructured data into files in a variety of forms, including CSV, spreadsheets, and PDFs. However, special emphasis must be placed on the value of accountability, the moral and legal ramifications of online scraping, and the need for rigorous data extraction procedures [3]. It will be explained in detail how to use Python-based tools like web scraping, which will expand the possibilities for data analysis and decision-making. Figure.1 shows the various domains of web scarping ranging from real state, marketing, E-commerce, travel and tourism, sports analytics and social media. E-commerce has a huge potential for web scraping.

Flipkart's website is a valuable resource for company research because it is well known for its items and customer feedback. With this tool, researchers and businesses can take full advantage of Python's web scraping capabilities to access product information, pricing information, customer reviews, and other information that will help them better understand customer preferences, market trends, business competencies, and rivalries. Understanding consumer and business behavior has become more crucial for firms to be competitive in the market as e-commerce has grown [4].

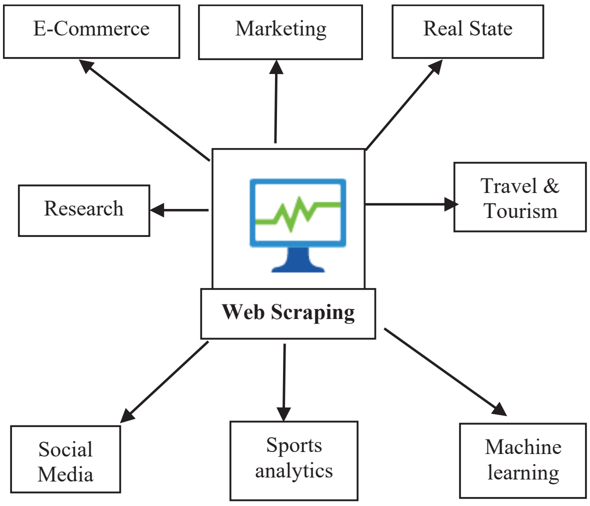

Fig.1 Domains of web scraping

Web scraping using Python, with particular focus on the famous website Flipkart, a famous e-commerce platform in India is mentioned here. Web scraping has several applications in many industries, making its utilization a diverse activity. Web scraping has a variety of purposes, such as:

- Data gathering and assembling: Web scraping mechanically pulls data from websites, enabling companies and researchers to gather big or redundant data for analysis [5].
- Market research and competition analysis: Web scraping offers information on the market, rivalry tactics, and consumer preferences.
- Lead Generation: Web scraping may be used to harvest contact details from websites, assisting organizations in lead generation and client database building.

979-8-3503-2919-3/23/$31.00 ©2023 IEEE

- Price comparison and monitoring: Online retailers may use the website to keep an eye on the prices of their rivals and modify their pricing strategy as necessary.
- Sentiment Analysis and Brand Monitoring: Web scraping may be used to harvest data from social media and review websites in order to analyze consumer behavior or user reviews of a certain brand or product.
- Material collection and news monitoring: [6] Web scraping is used to collect news, blog articles, and other online material from a number of sources. This data may be used by news organizations to arrange material, track breaking news, and examine new patterns.
- Academic Research and Data Analysis: Researchers may follow research data, gather data for analysis, analyze social media, and undertake social analytical thinking via web scraping.
- Training of machine learning and artificial intelligence models and algorithms: Web scraping is used to gather data for these models and algorithms. Researchers may build robust models that can forecast, categorize, or carry out tasks in acceptable language by merging different data [7].

II. LITERATURE SURVEY

The majority of people employ web scraping, which is a highly popular approach for obtaining data from various websites on the internet. [14] Web scraping using Python has been the subject of several research and resources throughout the years. While highlighting the advantages and possibilities of online scraping, [8] they also note its drawbacks and concerns, such as restrictions, legal and ethical issues, and the development of websites and technology.

TABLE I. MAJOR CONTRIBUTIONS IN WEB SCRAPING

S. No	Year	Author	Methods	Limitations
1	2017	John Smith	Python libraries such as BeautifulSoup and Scrapy. HTML parsing and data extraction techniques.	Lack of standardized protocols for web scraping, CAPTCHA handing leading to potential conflicts with website owners.
2	2018	Ryan Mitchell	Includes various scraping techniques such as HTML parsing, APIs, dynamic content.	Performance challenges when dealing with large-scale scraping tasks.
3	2019	Jane Doe et al.	Methods for handling user authentication, rate limiting, and data quality.	Difficulty in handling rate limiting and maintaining data quality when scraping social media sites. Ethical concerns regarding user privacy and data usage from social media platforms.
4	2020	Sarah Johnson	Ethical considerations in web scraping and provides guidelines for responsible scraping practices.	Challenges in interpreting and adhering to website-specific terms of service and scraping policies.
5	2021	Michael Anderson et al.	Comparison between different techniques for scraping websites that heavily rely on JavaScript to generate content.	Potential inconsistencies and compatibility issues between scraping libraries and JavaScript frameworks.

III. METHODOLOGY

Using technologies like beautiful soup, python scripts, and other sources like various websites, the goal is to collect all information and data. The Python-based web crawler scrapy may assist in obtaining the necessary outcomes and provides the correct URL to download data, with choices to obtain the data in an organized manner.

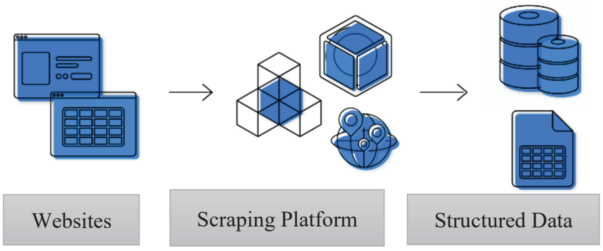

Fig.2 A standard web scraping process

Figure.2 demonstrate a standard three stage web scraping process. Websites data has been feed to the scarping platform and they will generate a structured data in return. The strategy utilized uses Python to qualitatively scrape data from the Flipkart website. The first step is to evaluate Flipkart's website's architecture and pinpoint the pertinent data that needs to be extracted. This offers details about the product, such as name, price, and ratings. The HTML text of a webpage is parsed to retrieve the essential data using Python's Beautiful Soup Library [9]. Additionally, Python's application library is used to send HTTP requests and store web pages for further processing.

The major libraries used are

a. Beautiful Soup: [10] Beautiful Soup is a well-liked Python web development library. It offers a fantastic method to filter, organize, and extract data from web pages and operates effortlessly and gracefully in HTML. Users may discover certain things, such as tags, classes, or qualities, and extract vital information thanks to Beautiful Soup's user-friendly interface. In the Python ecosystem, it is a popular option for web scraping projects since programmers can easily create scripts to use it to accomplish data extraction. By examining the source code and the data structure, it offers a simple method for extracting data from HTML and XML files.

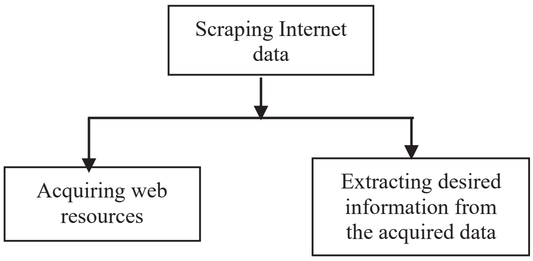

Fig.3 Two types of web scraping techniques

b. Requests: The library for requests enables sending HTTP requests to get web pages and offers crucial features for managing cookies, headers, and sessions.

c. Pandas: [11] The panda's package, which offers strong data structures like data frames, is used for data manipulation and analysis. Steps to be followed for web scraping

Step 1: Obtaining webpage content from the Flipkart website is step one. The HTTP GET request is sent by the Requests library to the requested URL, and then the response is retrieved. The answer includes the web page's HTML content, which is saved for later processing.
Step 2: The HTML material that was downloaded from the Flipkart website is parsed by the BeautifulSoup library. It offers tools for navigating and searching the HTML tree structure, making it simpler to retrieve particular information and parts from a website [2].
Step 3: The important data components, such as the product name, price, rating, and customer reviews, are located using the HTML structure and the inspection capabilities offered by contemporary web browsers. These elements often contain unique class or identification properties that enable targeted extraction, and they are encased in certain HTML tags.
STEP 4: The identified data components are extracted from the parsed HTML using the BeautifulSoup library. The library offers ways to locate particular tags and get the text or properties related to those tags. The proper variables or data structures are used to store the extracted data for further processing.
Step5: To successfully handle the retrieved data and construct data frames for data cleaning, utilize the Pandas package.
Step6: The data is saved in an appropriate format for future analysis. It has been cleaned and organized. CSV (Comma Separated Values) or Excel files are frequent choices. Functions to save data frames as CSV or Excel files are available in the Pandas library.

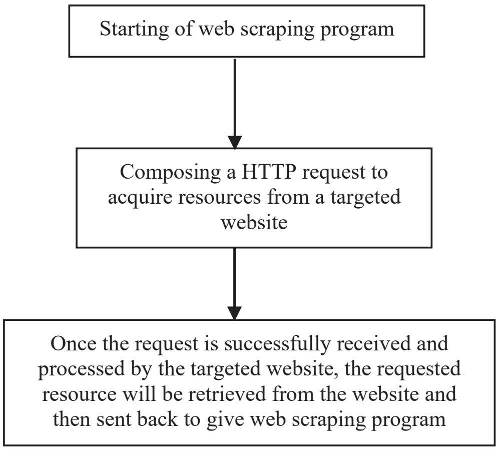
Starting of web scraping program
Composing a HTTP request to acquire resources from a targeted website
Once the request is successfully received and processed by the targeted website, the requested resource will be retrieved from the website and then sent back to give web scraping program

Fig.4 A systematic approach for web scraping algorithm implementation

Step 7: Care is taken to respect moral standards and legal obligations when scraping the web. To avoid excessive or disruptive scraping, the system respects the robots.txt file on the website, abides by the terms of service, and caps the number of requests done per second or minute.

LIBRARIES REQUIRED FOR WEB SCRAPING

The following libraries are required for web scraping using Python:

1. autopep8==1.6.0: A library for automatically formatting Python code to match the guidelines in PEP 8.
2. beautifulsoup4==4.11.1: [4] It offers a fantastic method to filter, organize, and extract data from web pages and operates effortlessly and gracefully in HTML.
3. certifi==2022.6.15: A library that provides a set of curated root certificates for verifying SSL/TLS connections.
4. charset-normalizer==2.1.0: A library for normalizing and converting character encodings in text data.
5. click==8.1.3: A library for creating command-line interfaces in Python.
6. Flask==2.1.2: A micro web framework for building web applications in Python.
7. idna==3.3: A library for handling Internationalized Domain Names (IDNs) in URLs.
8. importlib-metadata==4.12.0: A library for accessing and manipulating metadata about Python packages.
9. itsdangerous==2.1.2: A library for securely signing data and generating tokens in web applications.
10. Jinja2==3.1.2: A template engine for Python, used for generating dynamic HTML content.
11. MarkupSafe==2.1.1: A library for escaping and unescaping HTML, XML, and other markup languages.
12. numpy==1.23.0: A library for performing mathematical and logical operations on arrays and matrices.
13. pandas==1.4.3: A library for data manipulation and analysis, providing data structures and functions for efficient data handling.

14. pycodestyle==2.8.0: A library for enforcing the PEP 8 style guide for Python code.
15. python-dateutil==2.8.2: A library for parsing, manipulating, and formatting dates and times in Python.
16. pytz==2022.1: A library for working with time zones in Python.
17. requests==2.28.1: A library for making HTTP requests in Python.
18. six==1.16.0: A library that provides utilities for compatibility between Python 2 and Python 3.
19. soupsieve==2.3.2.post1: A library for parsing CSS selectors, used in conjunction with BeautifulSoup for web scraping.
20. toml==0.10.2: A library for parsing and manipulating TOML (Tom's Obvious, Minimal Language) configuration files.
21. urllib3==1.26.9: A library for handling HTTP requests and connections in Python.
22. Werkzeug==2.1.2: A library for building web applications in Python, providing routing, request handling, and more.
23. zipp==3.8.0: A library for handling ZIP archives in Python.

IV. DISCUSSION

Python was used to gather pertinent data from Flipkart's product listings for future analysis or use, and its practical use involved scraping information from the Flipkart website.

Python has a number of potent web scraping packages, including BeautifulSoup and Scrapy. In order to retrieve the needed data, the procedure involved sending HTTP queries to the Flipkart website and parsing the HTML of the web pages.

There may be moral and legal issues with web scraping. It is essential to abide with the terms of service for the website and refrain from sending the server an excessive number of requests.

It is possible to identify particular information from the Flipkart website, such as product names, pricing, and ratings. There are several uses for the data that was taken from the Flipkart website. It may be used, for instance, for product recommendation systems, competition analysis, pricing comparison, and market research. Insightful information about consumer patterns and preferences may be found in the scraped data, which can help ORGANIZATIONS make wise decisions.

V. CONCLUSION

The efficiency of web scraping with Python to retrieve data from the Flipkart website was proved in this study report. We can gather crucial data, such as items, pricing, and ratings, by using web scraping technologies.

The findings of this study highlight the significance of websites for gathering large amounts of data from e-commerce companies like Flipkart. The data may be utilized for a variety of things, including market research, pricing monitoring, rivalry analysis, and client analysis.

The challenges associated with web scraping, such as handling anti-scraping mechanisms, maintaining ethical practices, and ensuring data privacy also have been highlighted. Adhering to the terms of service and legal guidelines is crucial to avoid any legal or ethical issues during the web scraping process.

VI. FUTURE WORK

There are many opportunities for future research and development to enhance the capability, effectiveness, and technological ethics related to online scraping, among other prospective fields of study. Researchers may increase understanding in this area and allow beneficial applications across numerous fields by addressing these characteristics.

The following suggestions provide potential directions for further exploration and improvement:

1. Additional Data Extraction: Data components including customer reviews, seller information, and rating data may be extracted. In doing so, a larger dataset for analysis would be made available, allowing for a better comprehension of market patterns and consumer preferences.
2. Effectively Handling Dynamic Webpages: Future study can investigate effective handling methods for dynamic webpages.
3. Scalability and Performance Optimization: Scalability and performance optimization of web scraping processes is a crucial component for further research.
4. Data Cleaning and Preprocessing: Future research can concentrate on creating automated methods to deal with frequent data quality concerns including missing values, inconsistent formats, or outliers. This would speed up the data preparation procedure and boost the precision and dependability of subsequent analysis.
5. Ethical Considerations and Legal Compliance: Upcoming research may examine frameworks and rules for ethical web scraping, making sure that all applicable rules and laws are followed. Additionally, looking at other data sources or using open APIs may offer a more moral and compliant method of acquiring data.

The study paper's recommendations for future work include improving web scraping methods for obtaining data from Flipkart. Researchers can help to develop reliable and responsible web scraping practices by addressing issues with dynamic web pages, enhancing performance, overcoming anti-scraping mechanisms, improving data extraction capabilities, ensuring data quality, and taking ethical considerations into account. These developments will make it easier to get insightful information from Flipkart, allowing researchers to make wise choices and make significant contributions to a variety of fields including e-commerce analysis, market research, and consumer behavior studies.

REFERENCES

[1] Karthikeyan T., et al. "Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques." IJWP vol.11, no.2 2019: pp.41-52. http://doi.org/10.4018/IJWP.2019070103.

[2] Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, Florentino Fdez-Riverola “Web scraping technologies in

an API world *Briefings in Bioinformatics*, Volume 15, Issue 5, September 2014, Pages 788–797, https://doi.org/10.1093/bib/bbt026.

[3] Massimino, B. Accessing online data: Web-crawling and information-scraping techniques to automate the assembly of research data. Journal of Business Logistics, 37(1), 34–42.2016.

[4] Rose, M, "Web Scraping with Python and BeautifulSoup. Towards Data Science" 2020

[5] No Starch Press. Ryan Mitchell, "Web scraping with Python," O'Reilly Media, 2012.

[6] Bar-Ilan, J. Data collection methods on the web for informetric purposes – A review and analysis.2001

[7] Wu, L., Mattila, A. S., Wang, C. Y., & Hanks, L. The impact of power on service customers' willingness to post online reviews. Journal of Service Research, 19(2), 224–238.2015

[8] 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.

[9] O'Reilly, S. Nominative fair use and Internet aggregators: Copyright and trademark challenges posed by bots, web crawlers and screen-scraping technologies. Loyola Consumer Law Review, 19, 273.2006

[10] Landers, R. N., Brusso, R. C., Cavanaugh, K. J., & Collmus, A. B. A primer on theory-driven web scraping: Automatic extraction of big data from the internet for use in psychological research. Psychological Methods, 21(4), 475–492.2016

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

[12] Doran, D., & Gokhale, S. S. Web robot detection techniques: Overview and limitations. Data Mining and Knowledge Discovery, 22(1), 183–210.2011

[13] Lawson, R., & Sharp, J. Web Scraping with Python: Collecting More Data from the Modern Web. O'Reilly Media.2015

[14] Chinnathambi, A. Web Scraping with Python and Selenium. Packt Publishing.2018

[15] McKinney, W., & Wes, M. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.2012