



Flight Fare Exploration Using Web Scraping

¹Aditya Katre, ²Aniket Vaidya, ³Nvvs Karthikeya, ⁴Sakshi Pathare, ⁵Prof. V.R. Joshi

^{1 2 3 4}Students, Department of Computer Engineering, ISBM College of Engineering, Nande, Pune, India.

⁵Professor, Department of Computer Engineering, ISBM College of Engineering, Nande, Pune, India.

Abstract: A Python-based web scraping solution that automatically discovers airlines and optimizes bookings for budget travelers. The web scraper will collect various airline information from major airline websites, travel agencies, and metasearch engines using BeautifulSoup, Selenium, and request libraries. The extracted data includes departure/arrival city, travel date, stop, flight and price and is processed with HTML and JSON parsing technology. The system analyzes the details of this information, determines the parking lot, connecting airports and announcement dates and determines the lowest price for the reservation. Algorithm-driven recommendations will suggest the best bookings, increase efficiency in searching for budget airline options, reduce manual searches and improve the booking process for users.

Index Terms - Web scraping, Python, Flight fares, BeautifulSoup, Selenium, Budget-conscious travelers.

I. INTRODUCTION

Airline prices may vary between different airlines and destinations depending on many factors such as demand, seat availability, promotions and dynamic pricing strategies. Companies often adjust prices based on these factors, resulting in variable prices that can change between platforms and over time. Web scraping provides a great solution for collecting flight data via website, especially when using Python-based tools and libraries such as BeautifulSoup, request, and Selenium. Through web scraping, people can look at behavior by creating Python scripts to navigate to different planes and around the web. It can access web pages, find relevant information such as departure/arrival locations, travel dates, flight options and prices, and extract this information for later use. BeautifulSoup helps parse HTML and extract specific content from web pages. Additionally, Selenium can modify the web browser interface, allowing scripts to interact with passive elements such as clicking a button or filling out a form to enter flight prices. Web scraping can collect various flight price data from different countries by leveraging Python libraries. Web site. The data collected can be aggregated, cleaned and analyzed to identify patterns, trends and the best deals for special offers, travel dates or other interests. using Python for web scraping it simplifies the process of collecting data from various sources, allowing users to collect flight data, compare options on the platform, and make decisions. It improves the efficiency of collecting flight price information that would otherwise limit the use of manual crawling across multiple websites.

II. LITERATURE REVIEW

Web scraping is the automatic birth of information from websites. It deals with data using tools and ways similar as regular expressions, HTML parsing, computer vision, and orthogonal aggregation. crucial operations include-commerce value shadowing, social media analytics, sphere enrollment, data collection exploration, news/ content aggregation, and business intelligence. Python libraries similar as Scrapy, BeautifulSoup and Selenium give the capability to download web runners. Important considerations when choosing a web scraping tool are data, point structure, and integration conditions. As websites evolve grounded on data, web scraping becomes essential for business-wide data discovery and analysis. Overall, web scraping enables large- scale, automatic data birth from the internet for a variety of downstream analysis and business cases.

III. METHODOLOGY

To scrap the data from the website we use request to send request and retrieve data from the website, selenium is used to automate the process, using beautiful soup library we can extracts the targeted data all the process describe below.

3.1 Requests:

Requests, a Python library, facilitates HTTP requests to airline websites or travel platforms, retrieving HTML content for flight data access. This tool enables navigation across pages or URLs, crucial for gathering comprehensive flight information. BeautifulSoup, another Python library, parses HTML structures fetched by Requests. It identifies and extracts specific HTML elements—tags, classes, IDs—housing flight details like prices or schedules. With its intuitive interface, BeautifulSoup efficiently traverses the HTML document tree, aiding in organized data extraction. This tandem of libraries orchestrates the initial access and subsequent extraction of flight-related information, forming the foundation for further processing and analysis in web scraping endeavors.

3.2 BeautifulSoup:

Beautiful Soup, a Python library, serves as a crucial component in web scraping by parsing HTML content retrieved through Requests. This tool adeptly extracts targeted data elements—like flight details and prices—from the HTML structure. Its navigational capabilities within the HTML tree structure facilitate the pinpointing and extraction of desired data using defined tags, attributes, and classes. Through these specified criteria, BeautifulSoup systematically traverses the HTML hierarchy, locating and isolating pertinent information. This process enables the focused extraction of flight-related specifics, providing the foundational data required for subsequent analysis and informed decision-making in the realm of flight fare exploration and comparison.

3.3 Selenium:

Selenium, a robust web scraping tool, automates web page interactions, mirroring human behavior in browsing scenarios. Its primary function involves emulating user actions, enabling interactions with dynamic website content, including clicks, form submissions, and navigation through JavaScript-rendered elements. This functionality proves invaluable when scraping websites with intricate interactivity or those reliant on JavaScript rendering. By simulating user behavior, Selenium seamlessly accesses and retrieves updated or complex web content, facilitating the extraction of essential flight price data and other pertinent information critical for analysis and comparison in the web scraping process.

3.4 Process:

In the flight fare data scraping process, utilizing Python libraries like Requests and BeautifulSoup initiates by sending GET requests to targeted airline or travel websites containing flight details. Upon obtaining responses, Requests facilitates the retrieval of HTML content, subsequently parsed by BeautifulSoup. This parsing enables the identification and extraction of specific HTML elements encapsulating flight specifics such as prices, dates, airlines, and stopovers. For sites with dynamic content or JavaScript dependencies, Selenium mimics user actions, ensuring access to updated information. BeautifulSoup then filters and extracts desired flight data from HTML elements, while subsequent processing involves cleaning and organizing extracted information for storage in structured formats like CSV, JSON, or databases, enabling comprehensive analysis or presentation of flight fare insights.

IV. ETHICAL CONSIDERATIONS

Ethical considerations are paramount in web scraping endeavors, especially when extracting data from websites. Respecting website terms of service and adherence to ethical standards is crucial. Firstly, it's imperative to review and abide by a site's "robots.txt" file, respecting directives that outline allowed or disallowed scraping activities. Additionally, web scrapers must avoid causing server overload by implementing reasonable request frequencies and ensuring minimal disruption to the website's performance. Respecting copyright laws by not scraping proprietary or copyrighted content is essential. Moreover, explicit consent or adherence to a website's terms and conditions regarding data usage, scraping permissions, and user privacy is crucial. Lastly, transparency in data collection practices and attribution to data sources uphold ethical standards, fostering trust and responsible utilization of scraped information.

V. CONCLUSION

Flight ticket fares depends upon the time of departure and the availability of ticket from different online travel agencies (OTA). It shows the different prices for the same ticket. If one is looking for a flight ticket X which has a different price on OTA A and a different price on OTA B, comparing and analyzing the data from different websites can be difficult. Web scraping helps to scrap the data from the multiple websites, by analyzing and comparing the data one can get the best available fare for any flight ticket from the OTA.

REFERENCES

- [1] Chaimaa Lotfi, Swetha Srinivasan, Myriam Ertz and Imen Latrous, "Web Scraping Techniques and Applications: A Literature Review", In: Raju Pal and Praveen Kumar Shukla (eds), SCRS Conference Proceedings on Intelligent Systems, SCRS, India, 2022, pp. 381-394. <https://doi.org/10.52458/978-93-91842-08-6-38>
- [2]. R. N, A. M. S and C. N. Babu, "An Optimal Data Entry Method, Using Web Scraping and Text Recognition," 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 2021, pp. 92-97, doi: 10.1109/ICIT52682.2021.9491643.
- [3]. D M. Thomas and S. Mathur, "Data Analysis by Web Scraping using Python," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 450-454, doi: 10.1109/ICECA.2019.8822022.

