Mini Project 1 : Flight to Palma

Harry Setiawan Hamjaya

***Abstract*— Web scraping is a fundamental skill that every data expert should master. This abstract discusses the importance of web scraping using Google Colab, focusing on the ease of evaluation due to its Python library dependencies. The primary tool employed for web scraping in this study is Selenium, chosen for its versatility compared to Beautiful Soup, which was used in previous experiments. Furthermore, the study underscores the necessity of presenting the analyzed data to end-users to prevent valuable insights from going to waste. The ultimate objective is to create a user-friendly interface that recommends preferred flights based on user input, thereby enhancing the overall user experience.**

Keywords: Web Scraping, Google Colab, Selenium, Exploratory Data Analysis (EDA), User Interface (UI)

I. Introduction

In the process of scraping flight websites, I encountered several challenges. Firstly, it was time-consuming to identify suitable websites for scraping due to the unique and complex security measures some websites implement, particularly those associated with actively operating companies. For instance, in my initial attempts, I successfully scraped data from Google Flights, but within a week, an unexpected "Stale Element Error" occurred, which had not happened before.

The second significant obstacle was the emergence of "Time Limit Errors" on certain websites, even though the same code had previously worked on those webpages. Agoda, another website I attempted to scrape, had a more robust security system that required users to sign in before accessing the HTML content, making data fetching more challenging.

This led me to consider alternative websites that could be scraped in a more ethical manner. It's crucial to address this issue, especially since scraping data from companies without permission is often seen as unethical, particularly for students who may not have sought authorization in the past.

Additionally, capturing layover information proved to be difficult, especially on websites other than Kayak and Momondo. Most web security systems have advanced over time, while web scraping using Selenium is not the most updated technology; people tend to favor APIs for their convenience and ethical and legal compliance. I spent a considerable amount of time testing various websites to capture either the "Next" button or the details about layovers.

Another challenge I faced was the need to clarify which data should be collected. The problem specification did not provide clear information about the specific data to be scraped; instead, it used the term "etc." However, in a later section, data features were mentioned without prior explanation.

Lastly, the mini project was intended to be completed by a single person, which made it feel more substantial than the typical lab project that could involve two or three team members collaborating. The issue was not so much the size of the task but rather the tight deadline. Starting with scraping three websites, followed by data preprocessing, data visualization, and creating a simple user interface and ends with IEEE-format report with simpler explanation, it seemed almost impossible to complete within the initial deadline without an extension. This situation may have been less stressful for someone who didn't start from the beginning, but it was quite challenging for students who had to manage multiple courses, recognizing that everyone has only 24 hours in a day and the need for sufficient rest.

II. Data Collection

A. Website

I successfully completed web scraping for three websites: Kayak, Momondo (similar to Kayak), and Booking.com. Each of these websites presented unique characteristics and challenges during the scraping process. However, I encountered difficulties with Google Flights and was unable to complete scraping on this website.

In the case of Kayak and Momondo, I aimed to gather a more extensive dataset by repeatedly clicking the "Show more results" button, doing so a total of ten times. The objective was to acquire a dataset exceeding 50 entries, as I believed that having a larger dataset would enhance the quality of the subsequent analysis.

However, when it came to Booking.com and Kiwi (Similar to Booking.com), the approach was different. Instead of utilizing a "Show more results" button like Kayak and Momondo, Booking.com employed a pagination system. To collect data from Booking.com, I utilized the Selenium web driver to scrape information from all available pages. For getting the necessary information, I checked the class names and CSS selectors for each required element.

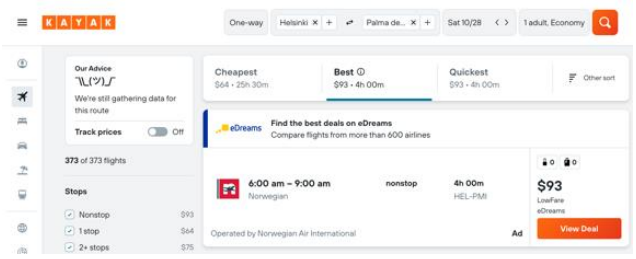

Figure 1: Kayak Website

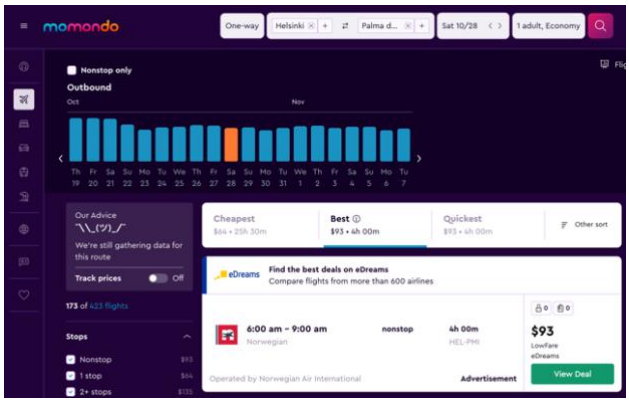

Figure 2: Momondo Website

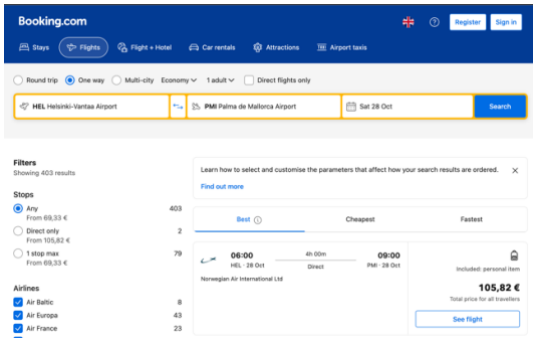

Figure 3: Booking Website

B. *Feature Data*

There are several lists of features that must be easily fetched as they are readily available on the initial webpage, specifically within the "Flights Card" section, without the need to click any buttons. Examples of these features include:

- Airline Name
- Departure Time
- Arrival Time
- Duration
- Price
- No. Transit (Layover)

On the other hand, there are several features that needs additional efforts as accessing them necessitates clicking on the "Flight Card" to access to gain the information, such as:

- Layover Time
- Layover Place
- Layover Place Code

The challenge arises from certain websites like Kayak and Momondo, which offer a dropdown card interaction when the flight card details are clicked, whereas Booking.com uses a pop-up card approach, occasionally with a distinct class name.

III. Data Analysis

After scrapping several column data from several websites, we are able to do some preprocess to clean the data and fix the format such that there is a consist data between each website, even they are scrapped from several resources.

The Preprocessing is done in several steps such as:

1. Changing the Date Time format for Departure and Arrival,
2. Changing the timestamp format for Duration Time and Layover Time,
3. Fixing the format by using Regex for string input such as Airline, Layover Place, and Layover Place Code,
4. Eliminating the Nan value,
5. Removing inconsistency data such as when Layover Time > Duration Time

Moreover, when the data has the similar format to each other, we can start the Data Visualization, to give us the sense of how the behavior of the available data is.

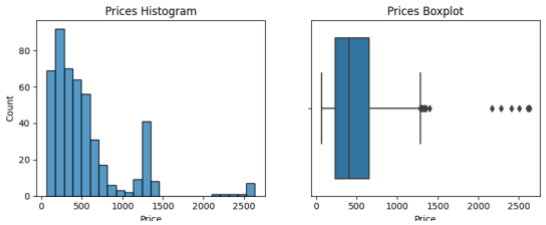

Figure 4: Price Distribution

Based on price distribution, we can note that:

- Most of Price falls around 200-700 dollars range, with median around 300 dollars
- There are 2 several notable outliers which are around 1200 dollars and approximately 2500 dollars.

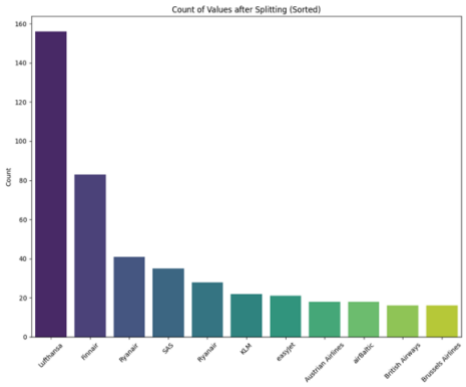

Figure 5: Top 11 Airline from the website

Based on airline count distribution, we can note that:

- Lufthansa has the most data (around 160 datapoints) that is nearly twice of the second airline which is Finnair (approximately 80 datapoints). It is making sense since based on the Wikipedia, it stands as the second-largest airline in Europe in terms of passengers carried, after the ultra-low-cost carrier Ryanair.
- On the other hand, Finnair would also be favorite because of this flight is from Helsinki, Finland and most of the people would prefer Finnair, compared to the other.
- The other top 9 data besides Lufthansa and Finnair aren't that significant, with maximum 41 and minimum 15 flights.

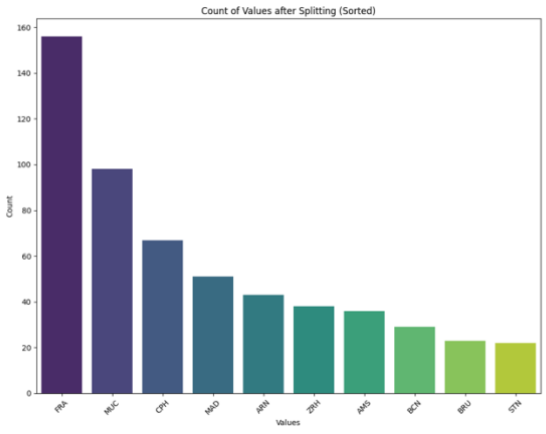

Figure 6: Top 10 Places for Flight Layover

Based on layover places count distribution, we can note that:

- France (FRA) is the most favorite place to layover (around 140 datapoints) that is nearly twice of the second place which is Munich (MUC datapoints) (approximately 80) and followed by Copenhagen (CPH) at the third place with (roughly 60 datapoints)
- The top 4 is very reasonable to be at the top. France, Munich, and Copenhagen airport are all in the Southwest part of Finland and one of the shortest ways to travel to Spain. The fourth one which is Madrid the capital city of Spain, it would be reasonable to have it in the top contender as layover places.
- The other top 9 data besides Lufthansa and Finnair aren't that significant, with maximum 41 and minimum 15 flights.

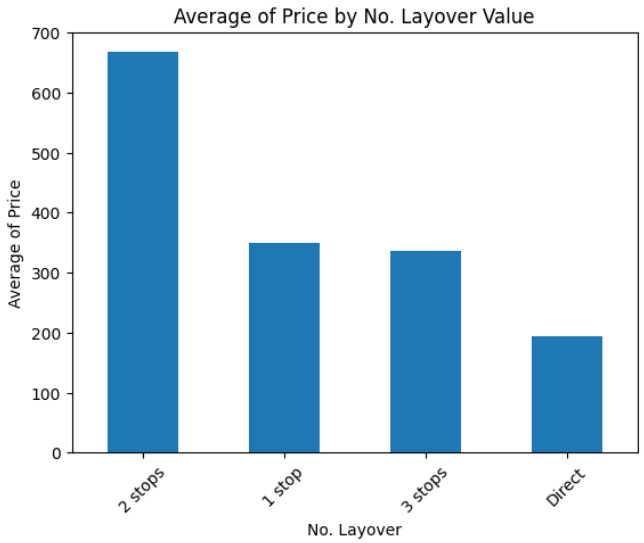

Figure 7: Price Average Based on Layovers

Based on No. Layover Average Aggregation, we can note that:

- The average price of "2 stops" cost the most expensive around 660 Dollars.
- Followed by the average price of "3 Stops" and "1 Stop" respectively with small difference, even though there are "2 stops" difference between them.
- The average price of "Direct" cost the cheapest which is logically True, since the least flight compared to others.
- Note that flights with "2 stops" have more data than each other, for imbalanced data, it is more than one and half times more expensive compared to flights with "1 stop". Moreover, after reducing the number is still one half of the flights with "1 stop".
- On other hand "Direct" flights with have less than 10 data, around 7 data, while flights "3 stops" have only 2 data.

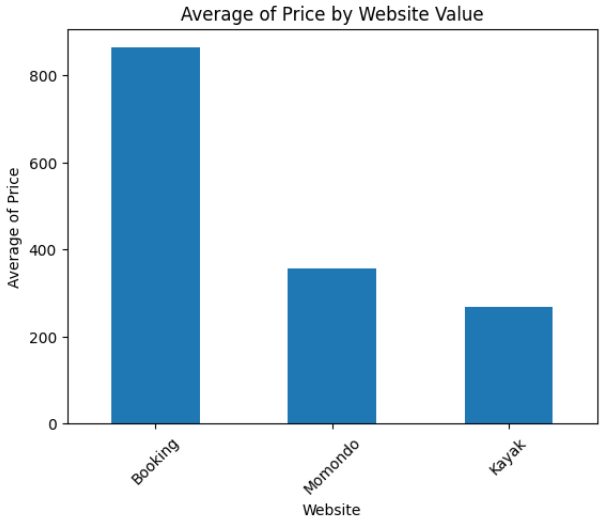

Figure 8: Price Average Based on Website

Based on Website Average Aggregation, we can note that:

- The average price of Booking cost the most expensive, which is around one and half times more expensive the average price of Kayak.
- The average price of Kayak is always the cheapest with the comparison around 14 dollars compared to Momondo, the second cheapest.

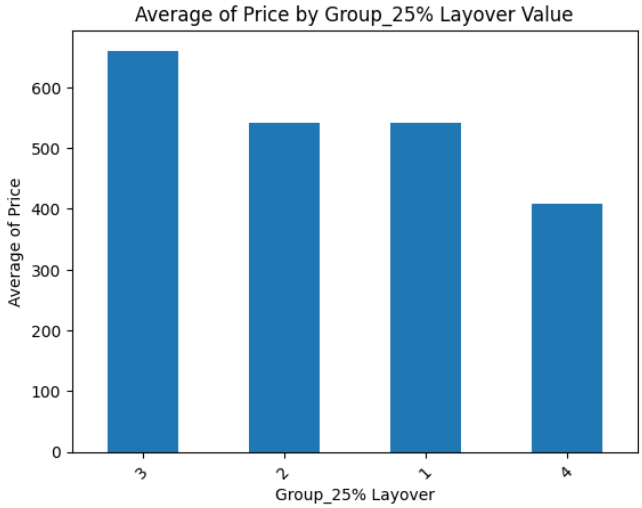

Figure 9: Price Average Based on Layover Time

The Layover Group is distributed uniformly such that all the group will have the same amount of data (25% of the data). For each group description:

- Group 1: 0 days 00:00:00 - 0 days 03:35:00
- Group 2: 0 days 03:40:00 - 0 days 06:30:00
- Group 3: 0 days 06:40:00 - 0 days 10:00:00
- Group 4: More than 0 days 10:00:00

Based on Layover Group Average Aggregation, we can note that:

- The average price of Group 3 is the most expensive, which could be explained since most of the flights are having "2 stops" before arriving at Palma Mallorca.
- Followed by group 1 and 2 at the same level, since the difference is 1 dollar which is less significant. It is good to note that since this calculation for layover time, and we have more data with "1 stop" then, the duration from group 1 is mostly affected by it compared to the "Direct" flight.
- Finally, Group 4 is the cheapest with average price around 400 dollars. The difference between Group 4 and Group 1,2 are in equal level compared to the difference between Group 3 (The Most Expensive) and Group 1,2. This one is also reasonable. Since the waiting time is extremely long (More than 10 hours, The maximum from my data for the uniform distributed data around 41 hours). In my opinion, the lower price is the compensation for the waiting time.

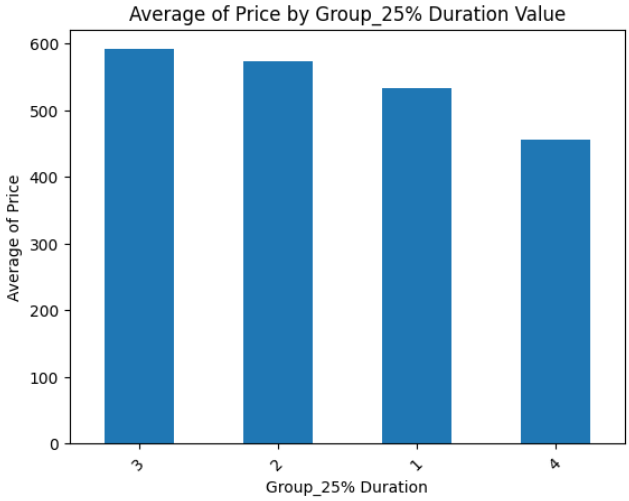

Figure 10: Price Average Based on Duration

The Group is distributed uniformly such that all the group will have the same amount of data (25% of the data). For each group description:

- Group 1: 0 days 00:00:00 - 0 days 09:20:00
- Group 2: 0 days 09:20:00 - 0 days 12:30:00
- Group 3: 0 days 12:30:00 - 0 days 15:55:00
- Group 4: Equal/More than 0 days 16:00:00

In general, the analysis is the same with layover time, because most of the duration is the summation from the layover time with the travel time.

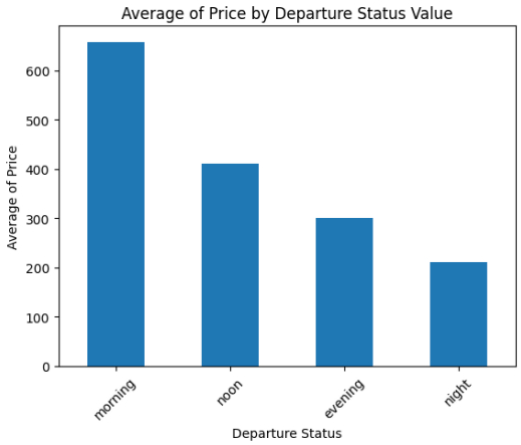

Figure 11: Price Average Based on Departure Time

Based on Departure Status Average Aggregation, we can note that:

- The average price of morning cost the most expensive around 650 Dollars which is greater than the sum of the average price of Evening (300) and Night (200) which is around 500 dollars. It is reasonable because most people are working on the morning, then they are used to do get up early rather than stay up late night to wait for the flight, especially for the Family and Friend that usually deliver the one.

- Followed by the Noon at the average price 400 dollars, with large difference approximately 250 dollars.
- The last two categories are quite cheap, flight at Evening at around 300 dollars and flight at Night at around 200 dollars, which both have difference 100 and 200 dollars respectively to the Noon flight (half of the difference between morning and noon). It is also reasonable, because most of the people and their family wouldn't be convenient to go to airport at the night and would preferably choose the evening instead. It is also because usually airport is far from way from the city, and it is inconvenient to go far away in the middle of night.

IV. Conclusion

In conclusion, web scraping is a valuable technique for retrieving valuable information from various websites. Nevertheless, due to the increasing awareness among companies about web scraping activities, many websites now employ measures to prevent such practices. The websites I mentioned earlier, which I attempted to scrape, are dynamic in nature and often rely on user input to generate specific results, such as search outcomes for a particular destination within a specified date range. Obtaining the desired data elements from these types of websites can be a more challenging task.

The initial solution pertains to situations where a website stops working due to a change in its elements. In such cases, it's not advisable to swiftly switch to another website, as this could result in a significant waste of time. The second solution for addressing Time Limit Errors involves taking a screenshot of the website to check for any cookie pop-ups that might be obstructing the page. If a website requires signing in, it's preferable to consider switching to an alternative website.

Regarding capturing layover information, more time is needed, as it involves capturing the entire pop-up data and then analysing the essential information using regular expressions (regex). Some websites employ a technique known as lazy loading to fetch data from their backend. This means that there is a delay, and when attempting to access the website's source without allowing all content to load, not all the necessary elements are immediately available. To address this, we can use explicit waiting techniques with Selenium to ensure that we wait until the required elements have fully loaded before proceeding.

Although there might be a lot of room for improvement, especially in areas such as task and grades specification, and establishing a reasonable deadline schedule. I believe the overall project has provided invaluable experience as a data engineer. This experience includes extracting raw data from several websites, preprocessing and visualizing the data, delivering it to users, and documenting the key points that have been learned during the first mini-project.