Exploratory Data Analysis of Flight Data

Motunrayo Ibiyo

1. Introduction

The selection of an airline for travel is typically influenced by a variety of factors, including but not limited to price, flight duration, and layover duration. The initial steps in the process of reserving a flight to a specific destination consist of the following steps:

1. Select a date or range of dates and determine if the trip is one-way or round-trip.
2. Using the above information as a guide, locate flights that are currently available.
3. Sort according to price
4. Filter itineraries based on the number of layovers.
5. Other individual criteria

Despite the simplicity of this process, it is necessary to iterate through the web pages of multiple airlines or booking agencies to obtain the optimal result or flight schedule. As a result, it is common to dedicate a significant amount of time and effort to deciding which flight to book.

One alternative approach to save time and effort is to use flight comparison websites or apps that aggregate the information from multiple airlines and booking agencies in one place. These platforms allow users to easily find prices, departure and arrival times, layovers, and other criteria all at once.

These flight aggregation websites are however limited in by the available airline options, and filter options. Therefore, it is still necessary to visit multiple flight aggregator sites before reaching a decision.

To reduce the time spent searching for the best plan and optimize user experience, this document reports on an approach to collect similar flight features from multiple flight comparison websites through web scrapping. Specifically, Selenium and beautiful soup python libraries will be used to collect flight information from the webpages. Popular python libraries like pandas, NumPy, and general string and list operations are used to preprocess the collected data. To analyze and Visualize the data, the matplotlib.pyplot and seaborn library was utilized. Finally, a simple user interface was designed to show the data and allow for two stage filtering options.

2. Data Collection

The objective of this step is to collect useful flight information from flight aggregator webpages in a format that can be preprocessed for analysis. Flight features such as, flight duration, number of layovers, layover durations, flight departure and arrival time, airplane type, bag types allowed, and others are extracted from the booking sites.
Three websites were selected to flights information for a one-way trip from Helsinki-Vantaa Airport to Manchester Airport on 18th December 2023, across all sites. A major criterion considered in choosing the website is the availability of over 50 flights for the selected date and trip type. The data collected were saved in csv files and uploaded to my GitHub account for easy accessibility. The websites selected were:

- https://www.kayak.com/
- https://www.momondo.com/
- https://flights.booking.com/

146,143 and 70 records were obtained from the three websites respectively.

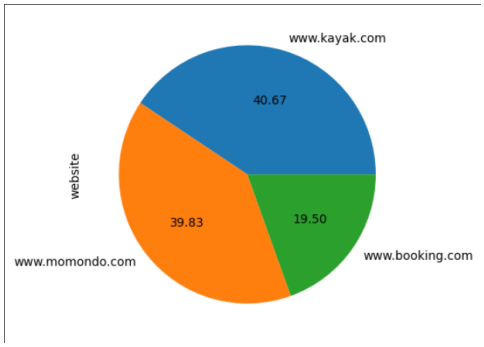

Figure 1: Data collected from the websites.

3. Data Preprocessing

The input for this step is a group of csv files containing the collected data in its raw form. Each file has a distinct structure hence preprocessing was done separately. The aim of this step is to ensure the data quality and generate a dataset with atomic features (single valued attributes) to enable easy analysis.

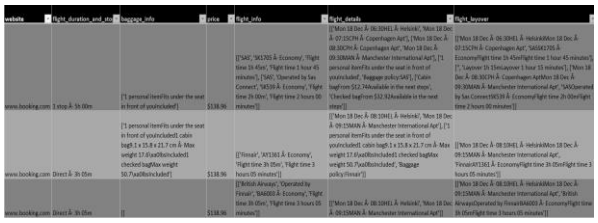

Figure 2: Sample data from Booking.com

Data processing was carried in two stages:

- Extracting Features from raw file.
- Converting each feature to the appropriate datatype and ensuring atomicity.

3.1. Extracting Features from raw file

The raw file consisted of field with features combined in a list. For each field, relevant data were extracted by manipulating the strings and list. Data atomicity was ensured in this step. Also, missing data were replaced with an appropriate value based on the data type of the field. Numerical data like number of layovers were replaced with zero values, while categorical data types were replaced with 'NA' implying not applicable or 'NAV' not available.

3.2. Converting feature to the appropriate datatype and ensuring atomicity

The data type of each feature was determined by the type of data in the column. The features were then cleaned to represent the data in the right data type.
The features cleaned include:

- Flight price: price data were collected as $xxx.xx, the aim of this step is to remove the dollar symbol.
- Durations Data: the total flight duration, and each layover duration were collected as (xx)hrs (xx)min. To enable the use of this features for analysis, they were converted to float data type. Therefore, a value of 02hr 30mins is converted to 2.30.

The result of preprocessing the records from the three websites were combined to a single dataset. The resulting dataset had 24 features and 359 records. One of the features was the data source i.e., the name of the website. Three features representing the type of airplane used for the flight was not available for records from booking.com.

4. Exploratory Data Analysis

To understand the data, exploratory data analysis was conducted. A logical way to do this is to analyze the data based on their different data type. The data analysis phase was divided into three main parts conceptually.

- Numerical data analysis
- categorical data analysis.

4.1. Numerical data analysis

The dataset consists of 11 features with numerical data type. The statistical distributions of this features were first analysed.

index	num_of_stops	price_dollars	total_flight_duration	first_layover_duration(hr)	second_layover_duration(hr)
count	359.00	359.00	359.00	359.00	359
mean	1.60	311.93	10.19	3.62	1.83383844
std	0.55	184.65	4.28	3.83	2.740818278
min	0.00	89.00	3.08	0.00	0
25%	1.00	172.00	7.67	1.25	0
50%	2.00	258.70	9.58	2.33	1.167
75%	2.00	390.65	11.75	4.71	2.917
max	3.00	1153.00	32.50	23.33	22.167

Figure 3: Statistical distribution of some numerical data

Then other graphs like the histogram, distribution plot, pair plot and scatter plot were used to visualize the values. The visualization was used to identify information that could not be seen easily. For example, it was observed that the flights with the price above the median flights had two layovers.

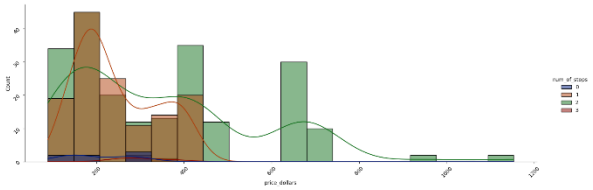

Figure 4a: distribution of price for various number of layovers.

Another interesting fact noted from the visualization was that a large percentage of flight from Ryanair were above $600.00.

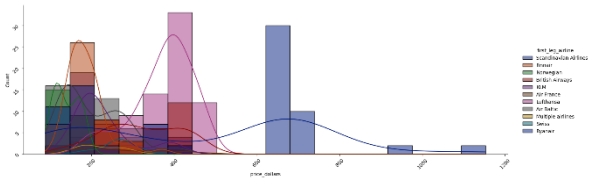

Figure 4b: distribution of price for each airline

4.2. Categorical data analysis

Categorical attributes like number of layovers, airline, departure time, cabin bags and checked bags were visualized using pie chart, bar charts.
The visualization of the departure time showed that a high percentage of flights to Manchester United Kingdom on the 18 left before 14:00. Drilling down further, fig 5b indicated that direct flights departed at 08:00 in the morning.

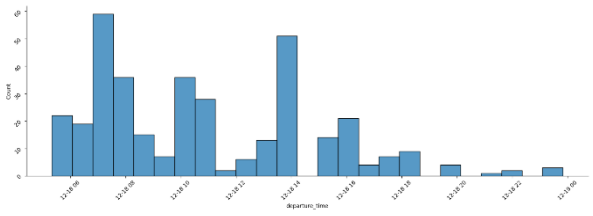

Figure 5a: distribution of departure time

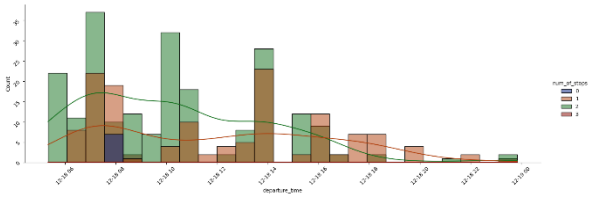

Figure 5b: distribution of departure time against number of layovers

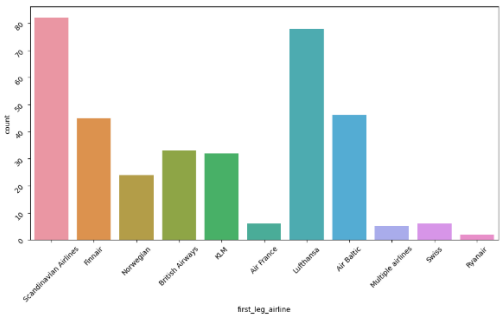

Figure 6:distribution of airlines

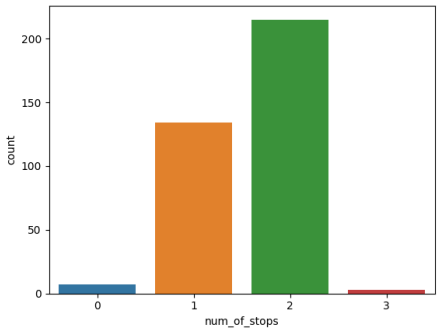

Figure 7: distribution of layovers

It was also observed that the Airbus plane is the most common plane type used.

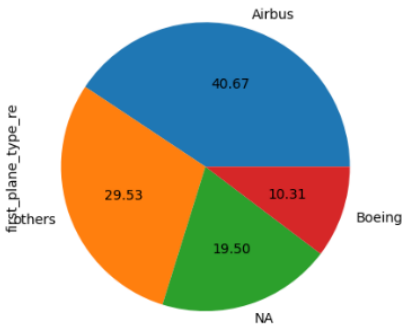

Figure 8: Types of planes

4.3. Correlation between Numerical and Categorical data

The association between key numerical data and categorical data were visualized with box plots, violin plot and grouped bar charts.
From the analysis, the distribution of prices across the number of layovers can be seen clearly. Figure 7 shows that the price of most direct flights is lower than $200, 00. Also, the flights with 3 layovers had a cost between $200, 00 and $500, 00.

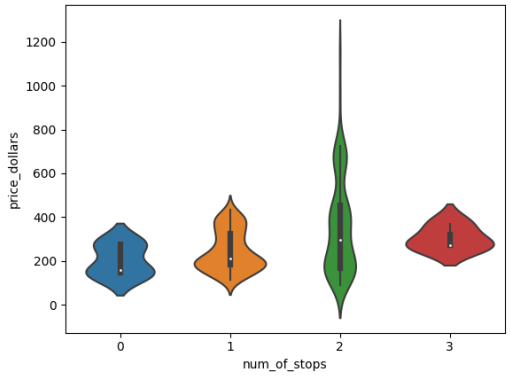

Figure 9: Distribution of price on the number of layovers.

Another interesting fact that was observed was that the higher the baggage allowed the higher the price of the flight. Flights with 2 checked bags had the highest price of over $1000, 00. Also, for each similar trend in seen in the direct flight as well as the flights with other number of layovers.

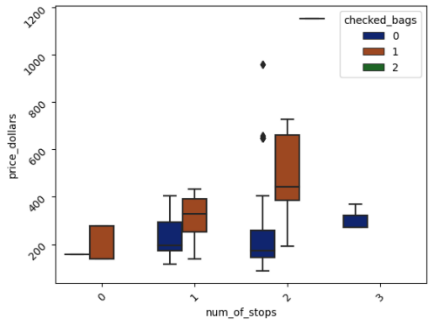

Figure 10: Distribution of price, number of stops and Quantity of checked bags allowed.

5. Interactive User Interface

A simple user interface was designed to allow users query the data from the three websites. The user engagement is in two stages.

The first stage allows users to specify price range, duration range, airlines, if direct flight only, if layover is allowed and others filter criteria. This returns records of data that fits into the criteria provided sorted by a user defined feature.

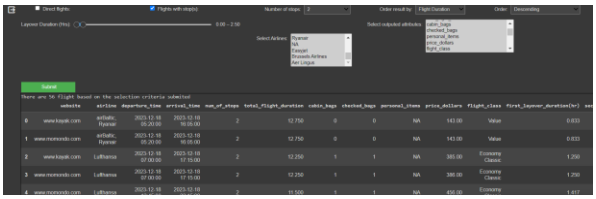

Figure 11: Overview of first part of the user interface

The user is then allowed to streamline this search again by providing a range of departure time, airline, and total flight duration. This stage returns information about the cheapest and the fastest (shortest duration) flight that is within all the filter criteria.

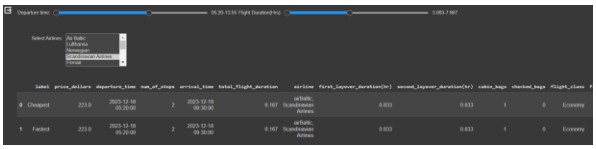

Figure 12: User Interface for streamlining search.

6. Conclusion

Through this project, a method that can be used to simplify the process of choosing a flight for a trip was illustrated. Three aggregating websites were scraped and preprocessed for analysis. The data was analyzed using several visualizations and the following questions can be answered with the visuals.

1. What is the price range of flights on 18th December 2023?
2. What is the total duration of flights that are available?
3. What is the minimum no of layovers for flights available?
4. What is the price range of flights for a specific number of layovers?
5. What are the airlines that is available?
6. What is the ratio of direct to flights with layovers?
7. Are flights with layovers cheaper than direct flight?

The final stage was to provide a user interface that will enable easy querying of the data.

Web scrapping comes with some challenges. Some of the challenges experienced on the site includes:

- Dynamic Document Object Model (DOM) of the websites. The logical structure of the websites changes frequently. This change could be as simple as a different class name or a change in the entire layout of the HTML document. Each scenario disrupts the process of scraping the site causing the code to break. To resolve the error usually requires a reevaluation of the webpage to identify the new structure. Also, the scraped data were saved to csv files to reduce the impact of this challenge.
- Element not Found Exceptions in some records: while scraping booking.com, it was observed that the selenium code breaks with "NoSuchElementException" after obtaining some record. However, this break can occur randomly. This experience was because the booking.com site requires a refresh after some time, and different class is used to represent same information in each HTML container (Div). To overcome this challenge, page refresh was added to the script. Also, common exceptions were caught so and dealt with so that the process will not break when they occur. A page refresh after an exception helps to reload the DOM preventing the next phase from being another exception. In addition, several implicit and explicit wait periods was introduced especially after opening an overlay. This is to ensure that the DOM is fully loaded before reading the data.

Despite the challenges highlighted, the project can be improved in several ways.

1. Addition of other booking sites.
2. The analysis of the information from the various website to remove flight details that were found across them.
3. Furthermore, more filter criteria like baggage allowance (cabin and checked bags), layover city can be added to the filter options available in the user interface.

The script for this project is available in my github repository:
https://github.com/Motunrayo244/EDISS-MP/blob/3fdf62c2a50fba3f05b9f14b7e73a65194b4a7cb/EDISS-MP/Data%20Science/Project_work/Project1/First_Mini_Project.ipynb