**Analysis of TripAdvisor Data with the help of Selenium and Python**

A Practical Tutorial on Scraping and Analyzing TripAdvisor Data using Selenium and Python

A picture containing diagram, text, screenshot, design

Description automatically generated

*Photo from* [*https://scrapfly.io/blog/how-to-scrape-tripadvisor*](https://scrapfly.io/blog/how-to-scrape-tripadvisor)

Web scraping is a powerful method for extracting data from websites. TripAdvisor, with its extensive traveler reviews and ratings, offers valuable insights into customer preferences. However, manual extraction and analysis from TripAdvisor can be time-consuming.

This tutorial focuses on using automation and programming with Python and Selenium, a web scraping tool, to efficiently scrape and analyze TripAdvisor data. It also covers the use of Python libraries like Pandas and Matplotlib to clean, structure, and visualize the scraped data. Additionally, the tutorial explains how to perform sentiment analysis on the collected reviews, providing further insights into customer opinions and preferences.

**. . .**

**Installing and setting up Selenium**

To get started, you need to install the Selenium library. Open your command prompt or terminal and run the following command:



Selenium requires a WebDriver to interact with web browsers. The WebDriver acts as a bridge between your Python code and the browser. The most commonly used WebDriver is for Chrome, called ChromeDriver. If you don’t have the driver installed on your computer, you can download it from [here](https://sites.google.com/a/chromium.org/chromedriver/downloads).

Then, you can use the following command in your python script so that Selenium can locate it:



**. . .**

**TripAdvisor dataset – What do we need to extract and how??**

In this tutorial, you will learn how to scrape data for all the reviews of 'Coffee & Tea' and 'Bars & Pubs' establishments in Thessaloniki from the TripAdvisor website. For each review, you will extract the following data:

1. Business Reviewed: The name of the establishment that was reviewed.
2. Reviewer Username: The username of the reviewer who left the review.
3. Review Date: The date when the review was posted.
4. Visit Date: The date of the reviewer's visit to the establishment.
5. Review Title: The title or heading of the review.
6. Review Text: The main content of the review.
7. Review Rating: The rating given by the reviewer to the establishment.

To extract the data, we'll be using Python and the Selenium library. Make sure you have Python and Selenium installed and the appropriate WebDriver for your browser.

You will use this [url](https://www.tripadvisor.com/Restaurants-g189473-Thessaloniki_Thessaloniki_Region_Central_Macedonia.html) to extract the information needed. To verify that the setup is working correctly, you can execute the following script:

A close-up of a computer screen

Description automatically generated with low confidence

Then, the actions that should take place are the following:

1. Click the "show more" element in the page.
2. Find and click checkboxes for "Coffee & Tea," "Bars & Pubs," and unclick "Restaurants"
3. Find and iterate over all business elements on the current page.
4. Open the business page in a new window or tab.
5. Iterate over all pages of reviews for the current business and extract review information mentioned above. Click the "show more" elements in the page to retrieve the extended review text.
6. Close the current window and switch back to the previous one.
7. Repeat steps 4-6 for the next business element.
8. Find and click the "Next" page link if available.

**. . .**

**TripAdvisor dataset – Web scraping process**

The scrapping process involves two primary while loops. Initially, we iterate through each page of the search results by clicking on the pagination buttons. For every page, we open each review in a new tab. Considering that a business may have numerous reviews separated on multiple pages, we ensure to retrieve reviews from all pages by clicking the pagination button. The information from the reviews is stored in a dictionary. After iterating through all businesses, we store the dictionary into a csv file.

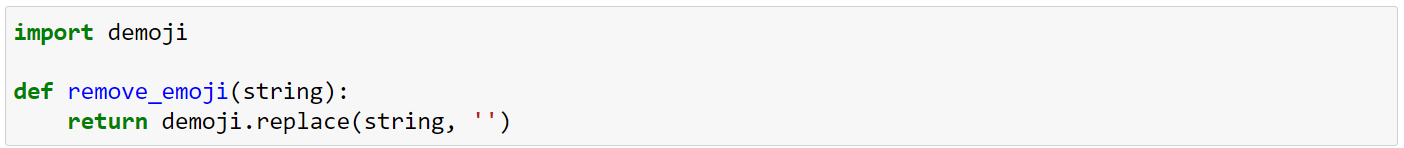
Be aware that network delays can occur, leading to exceptions like ElementNotInteractableException. To address this issue, we incorporate delays between interactions using the time.sleep command. Additionally, we implement a retry technique to handle any unexpected errors that may arise during the process.

**. . .**

**Data Preprocessing**

Before diving into the analysis of the scraped data, it is essential to preprocess the data effectively. In this tutorial, we will walk through the necessary steps to clean and prepare the data for further analysis. By following these steps, you will learn how to preprocess text data in a systematic and efficient manner. Let's get started!

1. Removing emojis from review text: Emojis are non-textual characters that do not contribute significantly to the analysis. To eliminate emojis from the review text, we will utilize the demoji library, which provides a convenient way to remove emojis from strings.



1. Removing non-alphanumeric characters from review text: Text data often contains non-alphanumeric characters such as punctuation and special symbols. These characters can interfere with the analysis and are typically removed. We can achieve this using regular expressions (regex) to match and remove any character that is not a letter, number, or whitespace.

A picture containing text, line, font, white

Description automatically generated

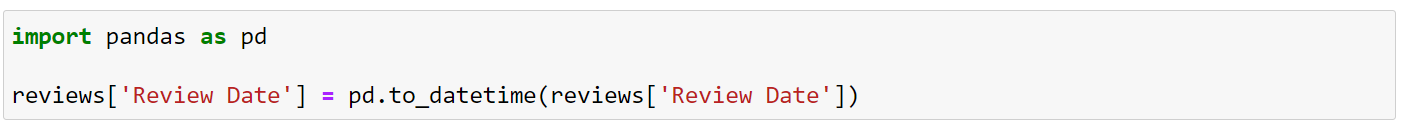
1. Applying stemming on words in review text: Stemming is the process of reducing words to their base or root form. It helps to unify variations of words that have the same meaning, thereby reducing the vocabulary size. We will use the Porter stemmer from the Natural Language Toolkit (NLTK) library to perform stemming on individual words.

A picture containing text, font, screenshot, line

Description automatically generated

1. Removing stop words in review text: Stop words are common words that do not carry significant meaning and are often excluded from text analysis. NLTK provides a list of commonly used stop words that we can utilize.A picture containing text, font, screenshot, line

   Description automatically generated
2. Converting review text to lowercase: Converting the review text to lowercase is a common preprocessing step in natural language processing tasks. It ensures that the same word is treated consistently regardless of its capitalization.
3. Converting review date to dateTime: In addition to preprocessing the review text, we may also need to preprocess other columns in the dataset. For example, if we for the 'Review Date' column containing dates, we might want to convert it to the appropriate data type for further analysis. We can use the pd.to\_datetime() function from the pandas library to convert the 'Review Date' column to datetime format.



By following these steps, you have successfully preprocessed the data and prepared it for further analysis. The overall code, combining all the preprocessing steps for review text, can be found below:

A picture containing text, font, line, screenshot

Description automatically generated

**. . .**

**Data – Final Format**

After completing the data preprocessing steps outlined earlier, the dataframe will have the following columns: Business\_name, Username, Review Date, Visit Date, Review Title, Review Text and Rating.

**. . .**

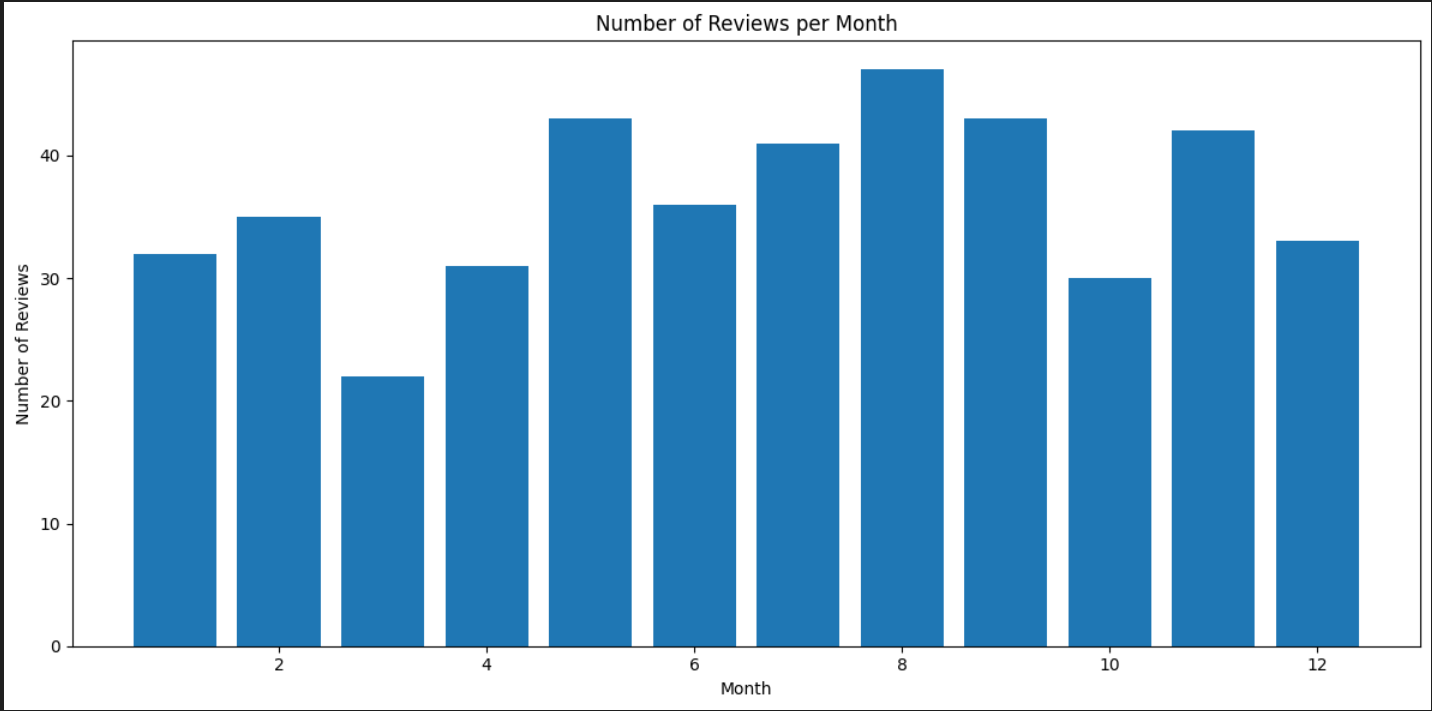
**Data Analysis**

Below, we have produced some interesting visualizations to get a better understanding of the data:

***Question 1****:*

*In order to retrieve the number of monthly reviews over the years, we group reviews by year and month and count the number of reviews in each month.*

*Overall, the plot suggests that the volume of reviews is relatively stable throughout the year, with August standing out as the month with the highest number of reviews, potentially indicating a seasonal trend.*

**

***Question 2****:*

***Question 3****:*

***Question 4****:*

*To visualize the most common n-grams, we use a vectorization technique to represent the words based on their frequency. After identifying the top 20 words, we create word cloud for most common words, bi-gram and tri-gram.*

*A close up of words

Description automatically generated with low confidence*

*In order to visualize the most common words, bi-grams, and tri-grams in 5-star versus 1-star reviews, we applied the aforementioned steps with a slight modification. We filtered the dataframe and create 2 new ones that include only reviews with 1-star and 5-star ratings.*

*A picture containing text, screenshot, plot, line

Description automatically generated*

***Question 5****:*

***Question 6****:*

**. . .**

**Insightful conclusions**

*I hope you find this tutorial useful. Please let us know if you have any thoughts or concerns.*

*Thanks for reading!*