Analysis of AirBnB data with the help of Selenium/Beautiful Soup and Python

# Authors:

Marios Christos Malamatinos

Mpouzianas Nikolaos 162

### Keywords: Data science, Python, Data Analysis, Web Scraping, Beautiful Soup, Selenium, MongoDB, Streamlit, Machine Learning

# Introduction

The main objective of the conducted work is to unveil new knowledge by extracting data from AIRBNB [1] with Selenium [2] and Beautiful Soup [3]. For the storage of data, Mongo DB [4] was utilized. The depiction of data is conducted with a Streamlite Python package [5]. The AirBnB data were collected from specific district regions of Thessaloniki, which is the second most populated city in Greece. The examined regions were Evosmos, Stavropoli and Menemeni-Ampelokipoi. Figure 1 present the examined region from Thessaloniki, which is line-shaded with black color.



Figure 1:Examined region of Thessaloniki

A completed dataset was formed based on specific filters that were applied to the AirBnB platform. The filter consists of a number of visitors, check-in and check-out dates and the name of the region. The name of the region was Evosmos, Stavropoli, and Menemeni -Ampelokipoi for three different iterations of the scraping process. The number of guests was selected 3 and check-in and check-out dates from September 3rd to September 10th for the year 2024. Let’s get started!

# Presentation of the AirBnB dataset

In this part. we discuss the room feature that the authors focus on collecting from every room utilizing the Beautiful Soup package. The web scraping process is analyzed in the AirBnB dataset web scraping process section. The desirable features are :

* Title: The title of the room.
* Review Index: A hosted visitor rating.
* Number of reviews: The total number of reviews from hosted visitors.
* Name of host: The name of the room’s owner.
* Price per night: Expressed in euros per night.
* Superhost (if exists): A characterization of the room given from the AirBnB policy.
* Guest Favorite (if exists): A characterization of the room, from the AirBnB policy.
* Properties: Refers to the number of beds/bedrooms/bathrooms and visitors that can stay in the room.
* Characteristics: Check in with an electronic locker, free cancellation until specific data or animal hospitality. More characteristics are presented in the Visualizations section.
* Geographical coordinates: The latitude and the longitude of the room.

# Setting up Selenium/Beautiful Soup

The first step for our scraping journey is to import the necessary libraries in our Python code. The overall web scraping process was hosted in a Chrome Browser. The next step is to initialize Selenium’s web driver which is going to interact automatically with the AirBnB site given a URL. Figure 2depicts the driver’s initialization and package setup.

Figure 2: Imports for web-scraping script and initialization of Selenium web driver

# AirBnB dataset web scraping process

After the initialization of the web driver, a URL is given to open a Chome window with an AirBnB page for a given region. Also, it is observed that for every region 15 pages were produced as search results, where each page contained 18 rooms. The total number of rooms equals 270 per region. Figure 3 present the starting page for a given region, which consists of 18 rooms.

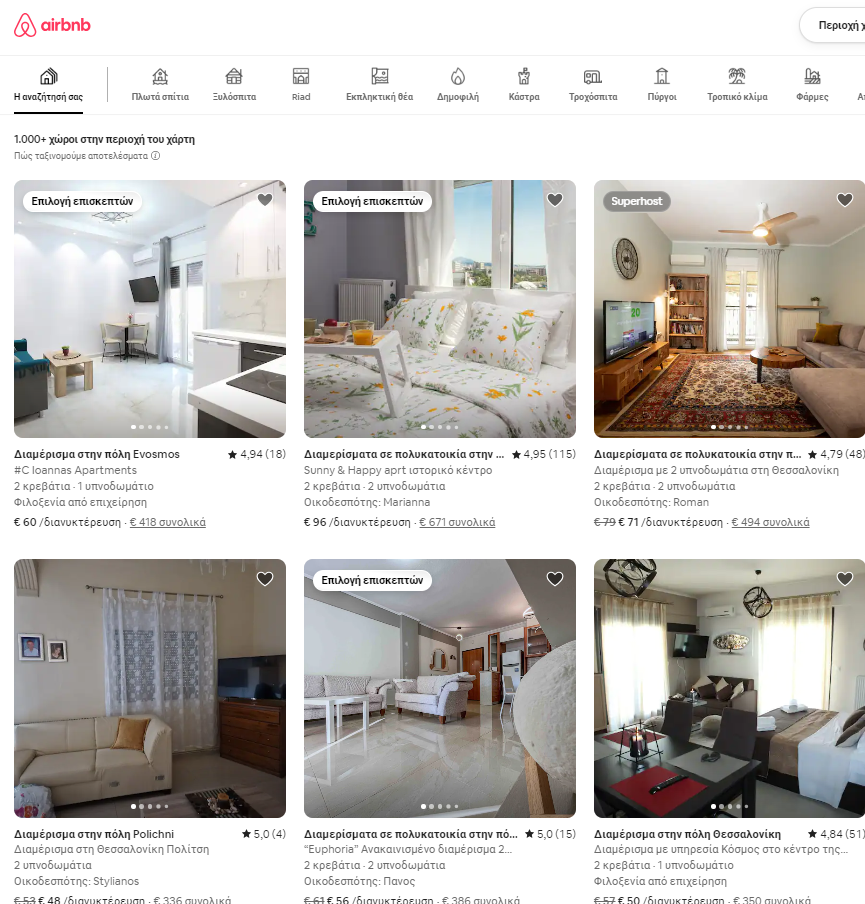


Figure 3: Scraping process starting page

It is worth mentioning that the web driver clicks on sequential room photos that generate a new window. The driver must be directed from the window from all rooms to the individual room window using the method switch\_to with the aim of extracting data from the individual room window using Beautiful Soup by calling the function extract\_room\_attibutes. The function will be analyzed at a later point in this section. After the data extraction, the driver will return to the main window to continue the process for the remaining room pages. An important issue is that the driver cannot identify all rooms within a certain page. To resolve this issue, the method execute\_scipt implements an automated scrolling to increase the driver’s view. The scrolling is being conducted after the examination of 4 rooms which is identified with i%4 ==0 statement. Also, there are time.sleep methods that are necessary so the driver can render the corresponding element correctly (e.g.room\_url). Figure 4 presents the code to assess the URL for the Stavroupoli region.



Figure 4:Driver iteration for all pages and rooms for URL in page Stavropoli

All data from every room are stored in a local directory in a JSON format. When the driver has searched all 270 rooms for a given region then it stores the data in MongoDB. The code is presented in MongoDB. Figure 5 presents the corresponding code that creates the JSON file for the individual room.



Figure 5:Code for storing JSON

When all rooms of a specific page are examined then the driver needs to press the next button to render the new page with another 18 rooms (Figure 6).



Figure 6: Button to acess the next page

Figure 7 depicts the code to switch pages.



Figure 7: Code to switch page

Apart from a page that contains all rooms (Figure 3) the selenium driver interacts with the individual room page. Figure 8 depicts the room page.

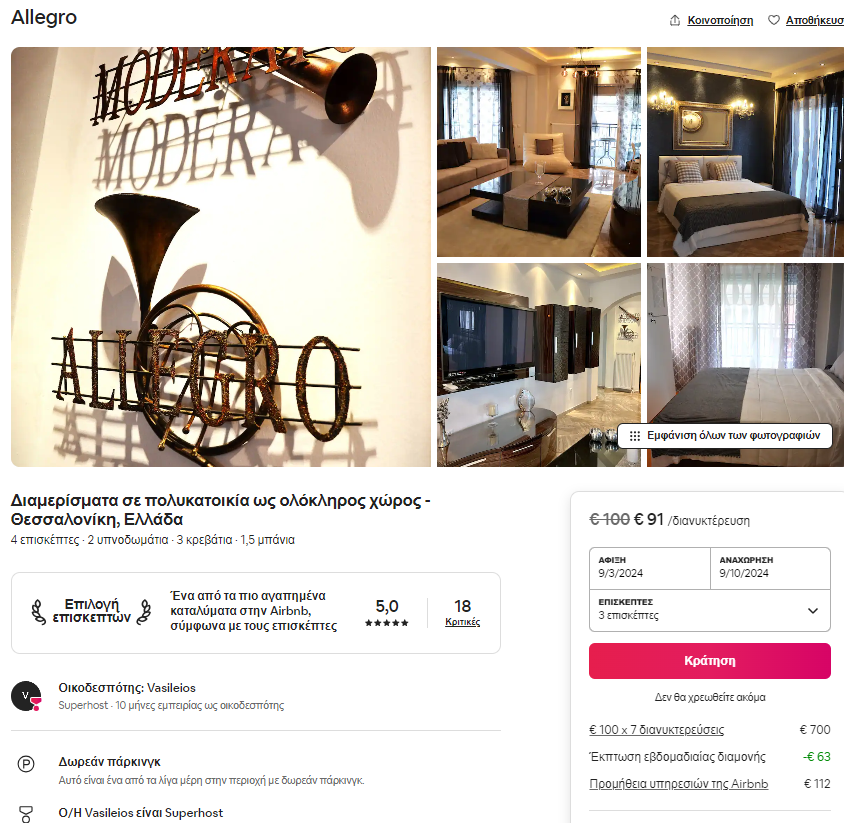


Figure 8: AirBnB room's page

When the web driver enters the browser window of each room the function extract\_room\_attributes is called. The function receives as input the page which is given by the driver as shown in Figure 9.



Figure 9: Page rendered from driver and input to extract\_room\_attributes function

The extracted features are collected in two different ways. All features except coordinates are collected using the Beautiful Soup. The coordinates that exploit the driver functionality are collected with Selenium functionality.

Figure 10 presents the code for the initialization of the Beautiful Soup and the extraction of all features except geographical coordinates. All features are collected with the assistance of CSS components. CSS components consist of class name attributes (.tilhkqq), ordered list (ol), headings (h3) and div.



Figure 10: Data extraction code with Beautiful soup

Regarding the coordinates, the driver has to scroll to the bottom side of the room page and acquire the coordinates from the Google Maps link. Figure 11 shows the code for the extraction of the geographical coordinates.



Figure 11: Code to extract the geographical coordinates.

Figure 12 present the final JSON format from a specific room before preprocessing.



Figure 12:Final JSON format before preprocessing

# Data preprocessing

After the completion of the scraping process, all data are stored in a local repository and the data preprocessing starts. It is worth noting the initial total number of data was 810 (270 for each region), whereas after preprocessing the total number was limited to 81.

The data preprocessing consisted of multiple steps to ensure the quality of data. The first step was to identify the room’s properties and characteristics and split the corresponding lists into individual features as presented in the section Final Data format. The next step was to remove the duplicated values, fill different empty fields and filter the rooms into the corresponding municipality because the results of the scraping process return room across all Thessaloniki regions. Room filtering was conducted with the Geopy Python package [6]. Figure 13 showcases the function that identifies the municipality of a room given coordinates.



Figure 13: Python function to identify regions

The final preprocessing step was to create some new features like distance from Aristortelous Square, subways and bus stations through the exploitation of the room’s coordinates. Figure 14 presents the implemented function for calculating the distance.



Figure 14: Function for distance calculation

# Final Data format

The final data format consists of 81 JSON files that are characterized from:

* Index
* \_id: The unique Mongo id.
* Title: The room’s name from AirBnb.
* New: Boolean if the room is characterized as New.
* Superhost: Boolean if the room is characterized as Superhost.
* Guest Favorite: Boolean if the room is characterized as guest favorite.
* Hostname: The name of the room’s owner.
* Coordinates: Room’s latitude and longitude.
* Price: Price per night.
* Distance: Distance features that describe the distance from Aristotelous Square, subway station and bus station.
* Properties: Refer to individual fields that describe the number of beds, baths, visitors and bedrooms.
* Review Index: A string value that expresses the overall rating from the visitor (e.g 4,83)
* off\_region: Based on the room’s coordinates the corresponding municipality is identified.
* Room’s characteristics: Individual fields that refer to boolean values and refer to telework place, fast wifi, arrival experience, experienced host, great location, check-in, furry friends, excellent communication from the host, cancellation, parking, and studio.
* Number of characteristics: The total number of characteristics per room.
* Number of properties: The total number of properties per room.

# Store data to MognoDB

After the formation of the final data format, the data are stored in a local repository to MongoDB. Three main functionalities were developed to connect to the database and retrieve specific collections (mongo\_connect), insert data (store\_data\_to\_mongo) and delete data (delete\_mongo\_data). It is worth mentioning that the connection to the database is conducted through a private VPN of the Aristotle University of Thessaloniki. Figure 15 depicts the final data in the MongoDB compass.

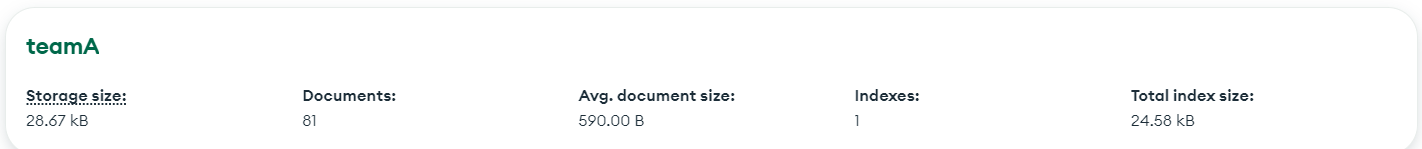


Figure 15: Data depiction in MongoDB Compass

Figure 16 depicts the 3 main functions that were developed for interaction with MongoDB.



Figure 16: Function for interaction with MongoDB

# Basic visulizaiton with Streamlit

After the data preprocessing the next step is to create meaningful visualizations that will unveil knowledge for our data. Towards the creation of a holistic solution, the Streamlit package was utilized. To enhance the user interaction of the proposed all plots were created with the Plotly [7] package. It is mentioned that this section presents only Streamlit platform visualizations because most of the author's code is based on tutorials from the official Streamlit page.

The Streamlight app consists of a main page that redirects the user to three sub-pages that describe the data statistics, data analytics and price prediction. Figure 17 show the Streamlit main page.

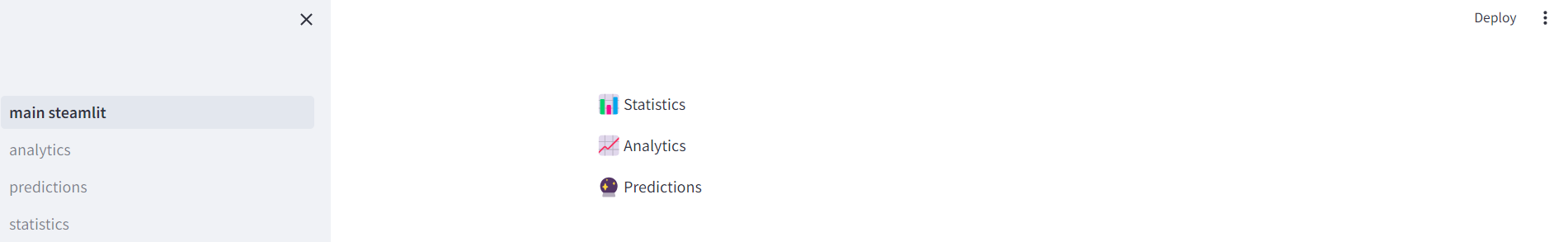


Figure 17:Streamlit main page

## Statistics

On the statistics page, the user should provide which region is interested in calculating statistics. Furthermore, users can select the price range, the review index (total rating) range, and the review numbers (total reviews) range. Apart users can filter rooms that are super hosts and guest favorites with a check box. Figure 18 depicts the mandatory input from the user.



Figure 18:Mandatory input from the user

An important aspect of the statistics page refers to the counterplots, which help the platform user identify useful insights about the selected data. Countplots refer to the review index (Figure 19), price (Figure 20), visitors (Figure 21) and available listing per municipality (Figure 22).

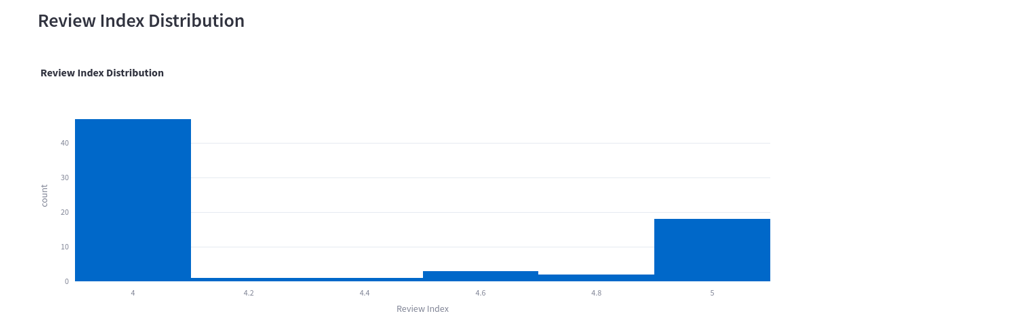


Figure 19: Review Index Distribution

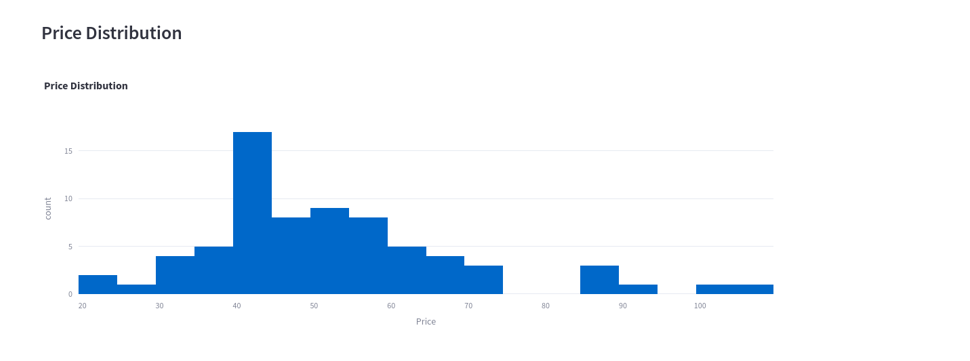


Figure 20:Price distribution

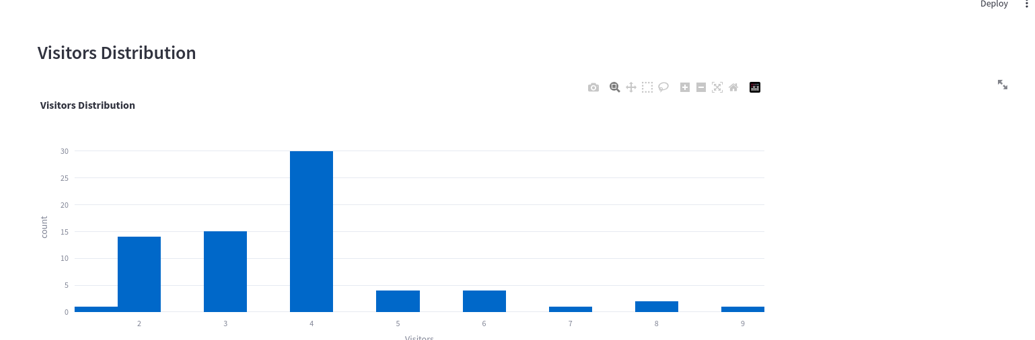


Figure 21: Visitors Distribution

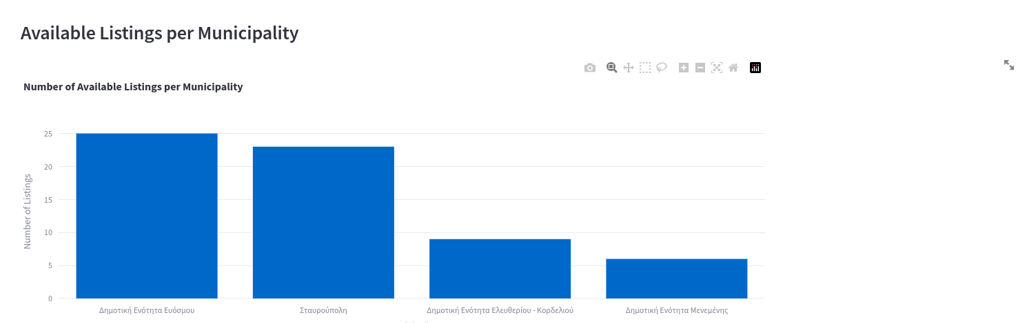


Figure 22: Available room distribution

Afterward, the platform presents a barplot to depict the distribution of a selected feature for all rooms in a given region. In this example, the total price per night is examined but the user can select any feature presented in the section Final Data format. Consequently, a table presents all the filtered data and a second table provides basic statistics for every feature of the filtered data. Figure 23 presents the visualization of the previous components.



Figure 23: Statistic page visualization

Apart from the aforementioned components the user can extract useful statistic insights for from a visitor vs price boxplot. Figure 24 depict the visitors vs price boxplot.

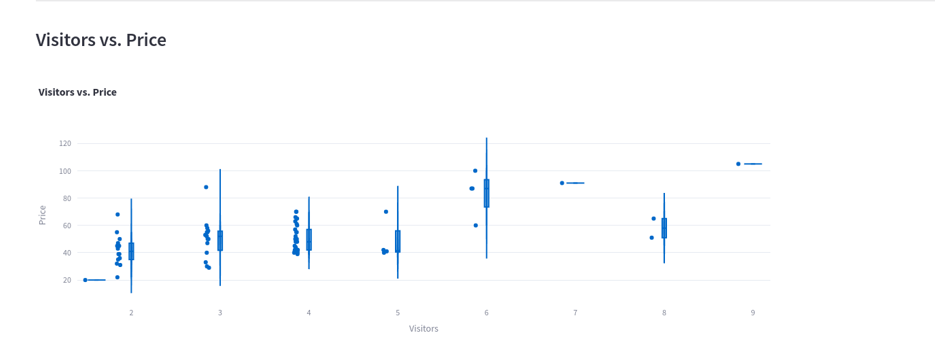


Figure 24: Visitors vs Price boxplot

## Analytics

Initially, the user can create a clustering of rooms based on specific characteristics that define the type of property. The user can define the number of deployed room clusters, which express the number of room types. The characteristics include price, beds, baths, rooms and guests. A 3d graph is created to enhance user awareness about the types of properties based on the first three selected characteristics. Figure 25 depicts the 3-d visualization chart based on price, visitors and beds.

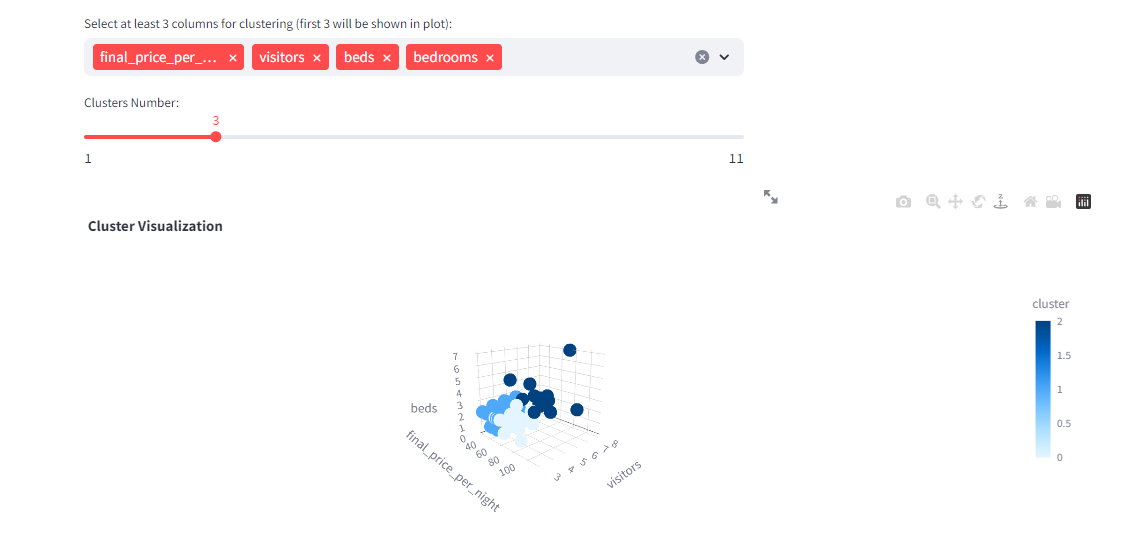


Figure 25: 3-d visualization chart for specific parameters

Following the clustering rationale the user can select the cluster type and the columns (Figure 26) to plot a correlation matrix (Figure 27).

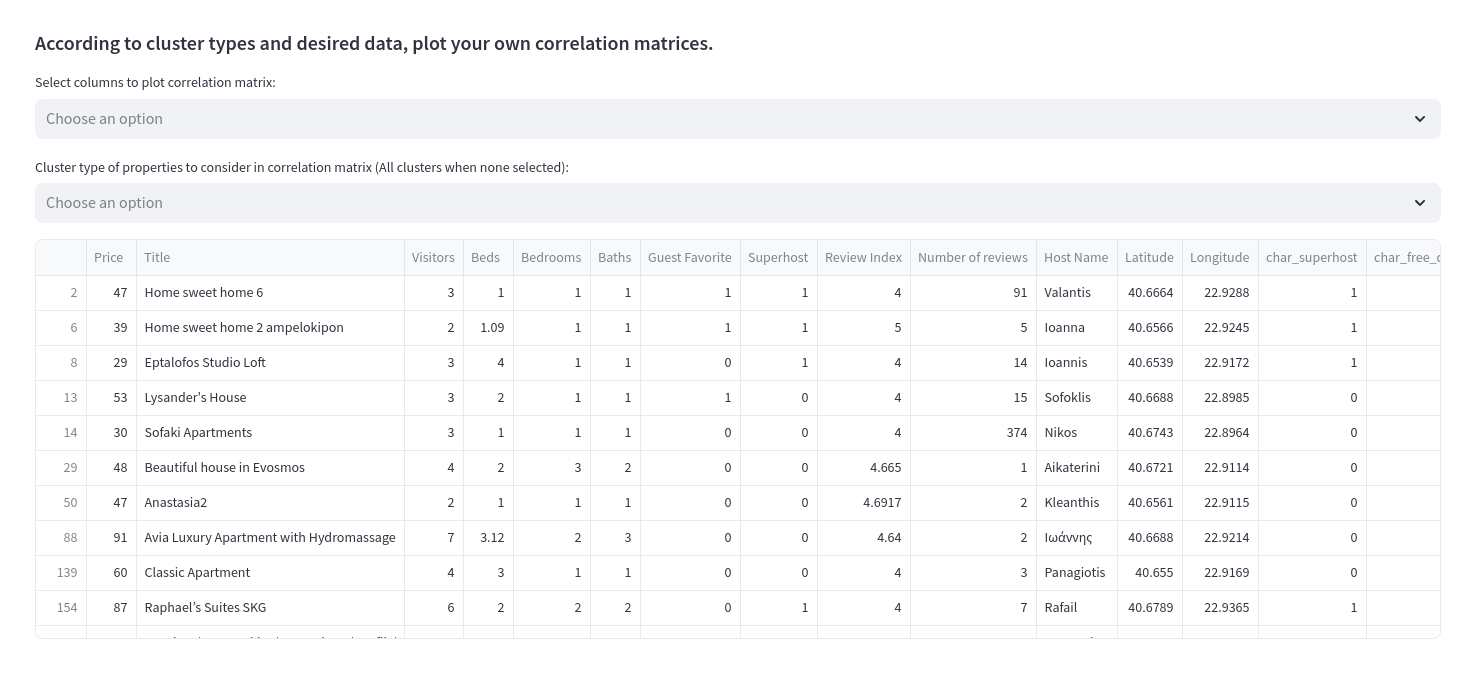


Figure 26: Selection of cluster type and characteristics

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, τετράγωνο, ορθογώνιο παραλληλόγραμμο

Περιγραφή που δημιουργήθηκε αυτόματα

Figure 27: Correlation matrix for specific characteristics and cluster type

Following we need to identify the top 10 and bottom 10 rooms based on the following formula:

Where c is a very small value to penalize the small number of reviews. For example, let’s assume that we have a rating of 4 and a number of reviews of 2 and 102 and c equals 1.5. Then the weighted will have a much greater impact small values than the value of 102 for a given room.

The top-rated and bottom-rated visualization count plots are depicted in Figure 28.



Figure 28: Top and bottom rated based on the weighted score.

The analytics section provides the page where the user can access the most common characteristics. The characteristics (check-in, cancellation) are depicted with values 1 (if existed) and 0 (if not existed). The authors calculated the total summation of every characteristic for all rooms to form a count plot that depicts the most popular (Figure 29).

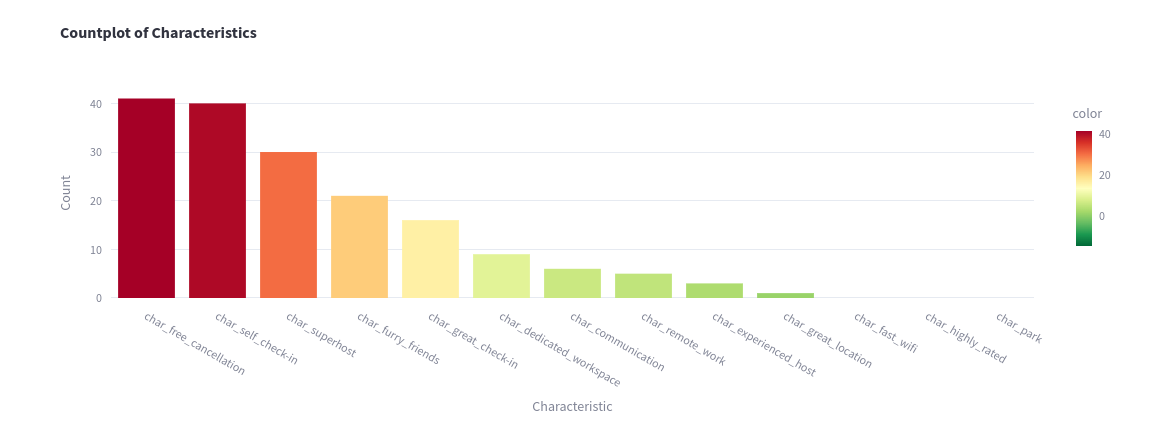


Figure 29: Popular characteristics

Figure 30 provides insightful statistics by forming a correlation matrix between characteristics.

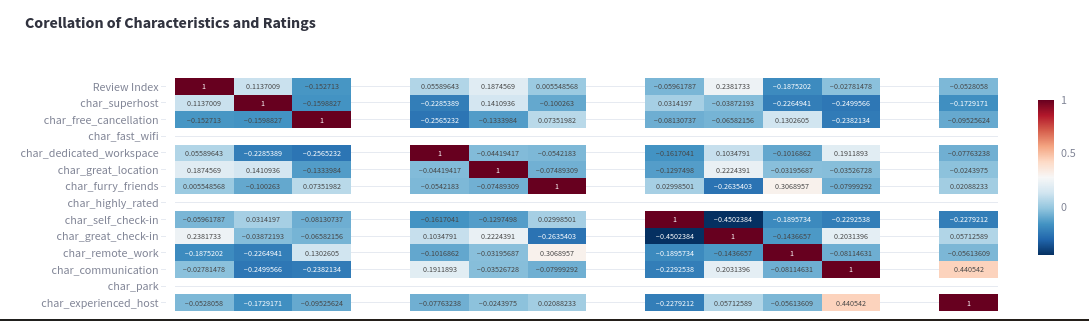


Figure 30: Characteristics correlation matrix

Also, a map of the examined region was created with the combination of Streamlit and Folium package [8]. Figure 31 depicts the region maps with room attributes.

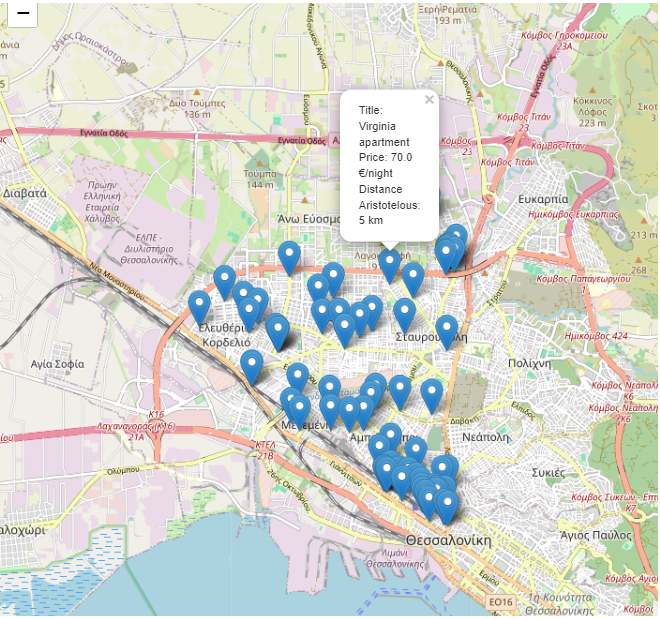


Figure 31: Folium map of the examined region

The conclusion of the analytics page includes a radar graph (Figure 32), where the user can insert the title of different rooms and conduct a benchmark based on different features with a radar plot.

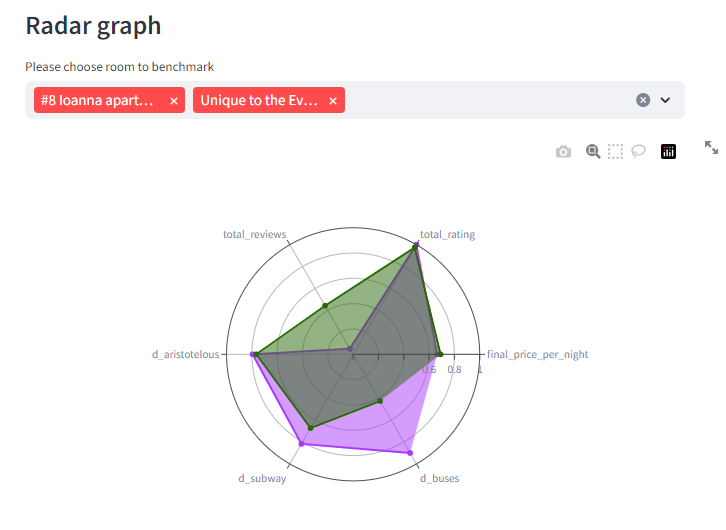


Figure 32: Radar plot for different rooms

## Price Prediction

The price prediction page consists of two main parts. The first part refers to user input created that will be given as input to the selected machine learning model. Figure 33 presents the required input for the ML model.

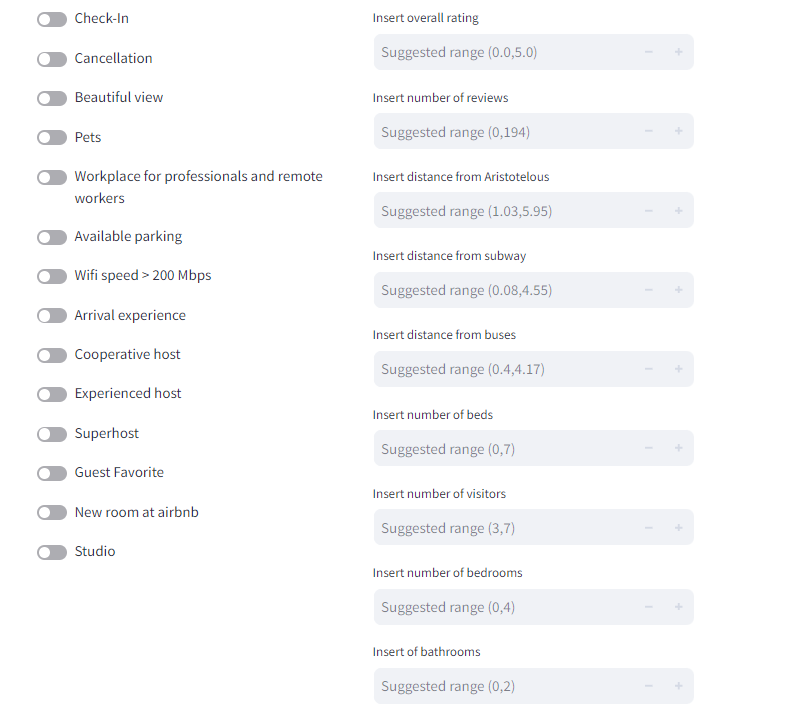


Figure 33: Required input for the ML model.

The second part consists of the model prediction values and visualizations regarding model interpretation. A table that contains all the training and testing data is depicted in the table. Also, users can select among three different regressors elastic net, random forest and Logitboost. Figure 34 presents the data table and the selection box of the ML model.

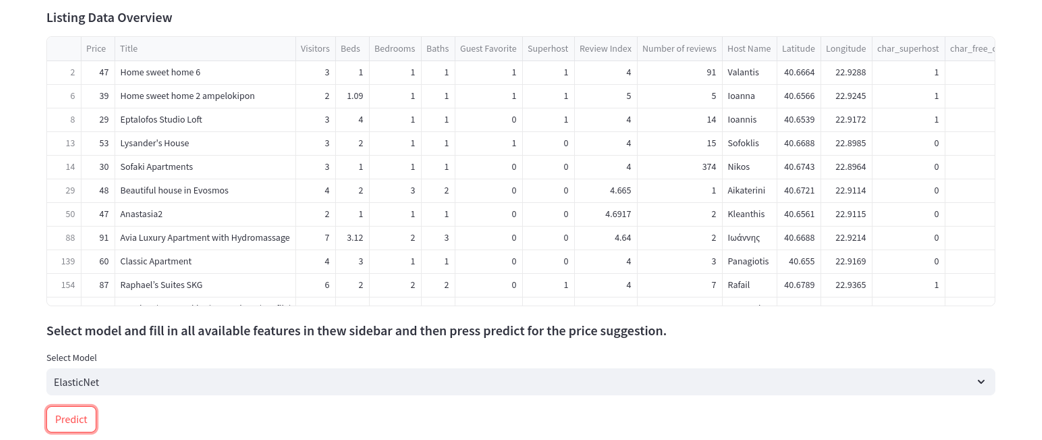


Figure 34: Data table and model selection before predictions

Finally, the model estimates a price and two graphs are formed based on the Shapley Python package. Figure 35 depicts the estimated price for a given input instance and the first Shapley plot.

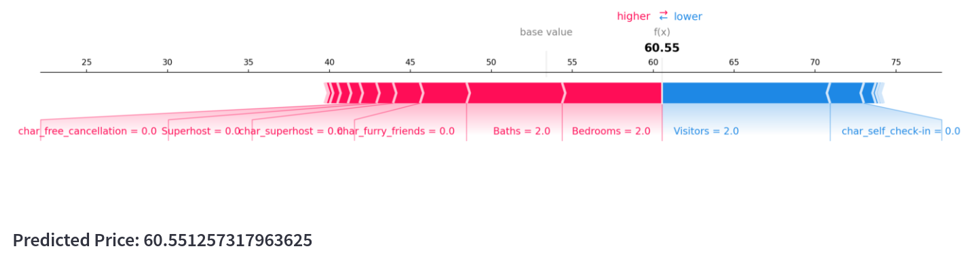


Figure 35: Predicted price and first Shapley plot

In general, the Shapley plot depicts values that increase (red color) or decrease blue color in the estimation of the price. The Shapley plot expresses that for a given input of a user the visitors with value 2 and check-in with value 0 decrease the price. In contrast baths and beds with a value of 0 increase the price estimation.

Figure 36depicts the second shapley plot for data interpretation.



Figure 36: Second Shapley graph

The second graph explains the impact of the predictive price which derives from the feature range. For example, low values of feature visitors (blue color) result in price decreases and high values (red color) of feature visitors result in price increases. Another example is with feature beds when the number of features is decreasing the the price increases and the opposite. Additionally, there are features that do not affect the price like parking, fast wifi, and great location of the room.

# Main observations from visualizations

From the creation of meaningful visualization with the Streamlit app, some useful insights were made. Figure 37 depicts that most of the rooms found in the AirBnB platform are located in the Evosmos and Stavroypoli municipalities.

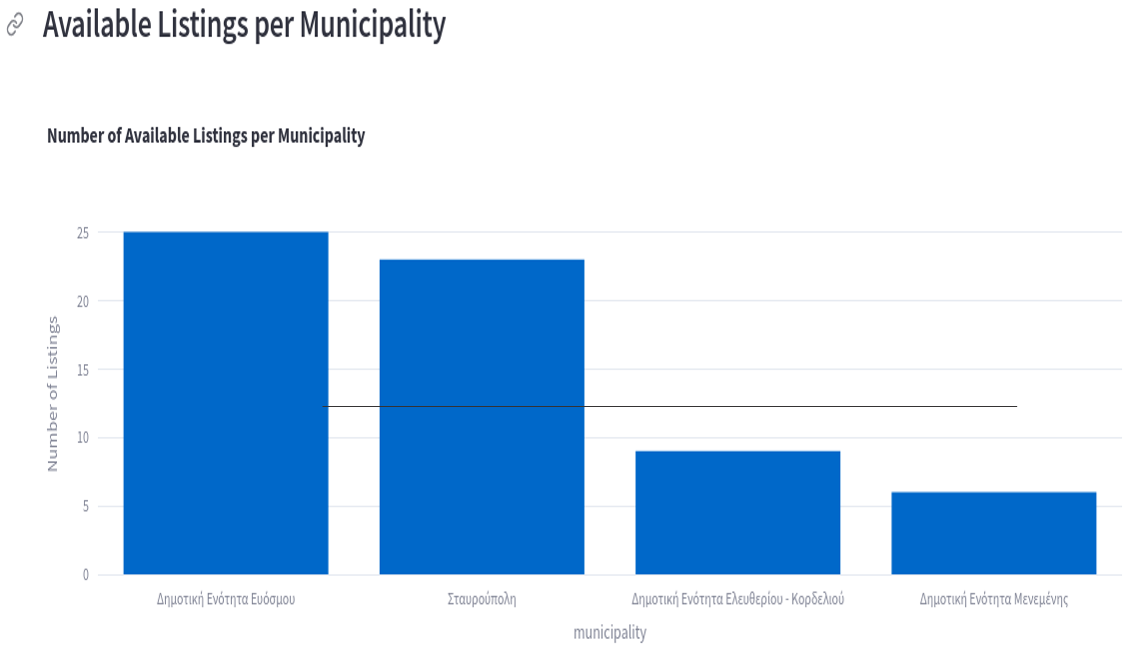


Figure 37: Municipalities countplot

Figure 38 illustrates that price distribution ranges from 30 to 70 with an outlier with a value above 190 euros.

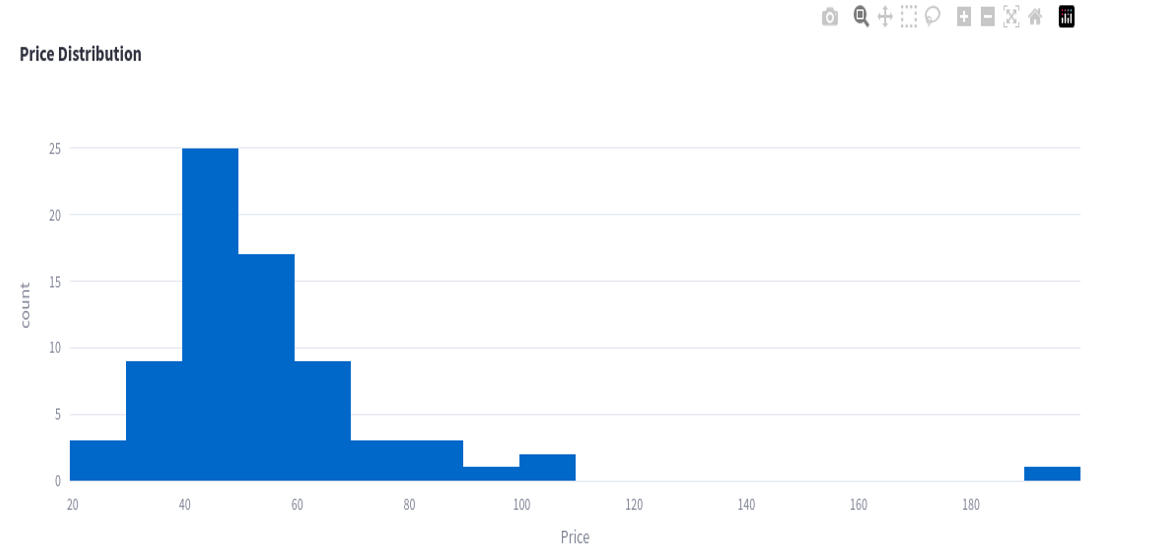


Figure 38: Price distribution

Additionally the review index distribution (Figure 39) shows that the review index varies between 3.9 and 5, while the majority of the list has value of 4 stars.

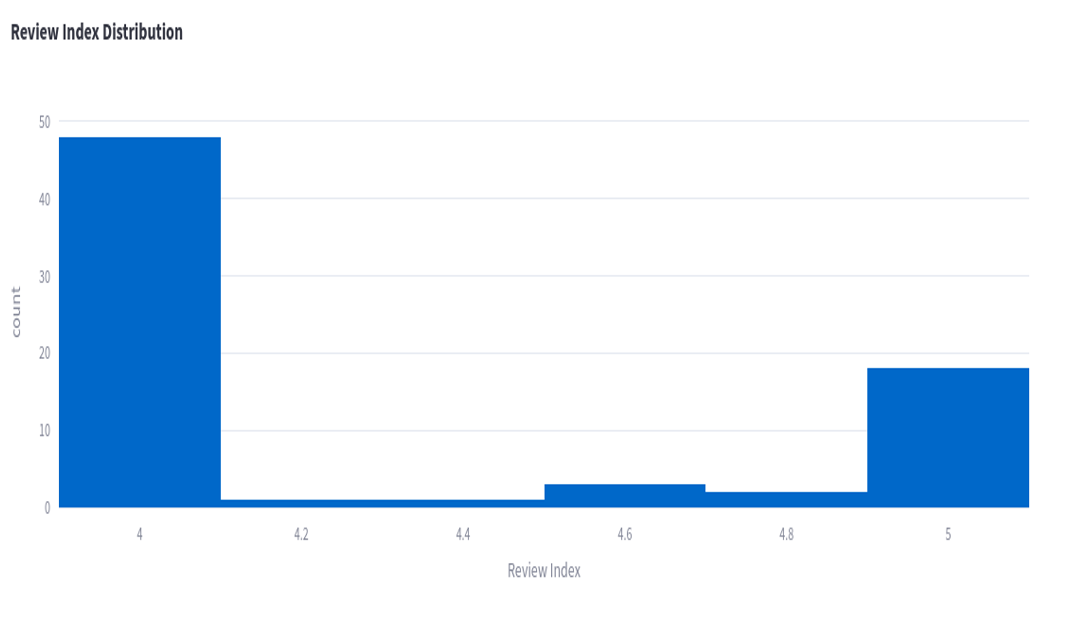


Figure 39: Review Index distribution

Another important observation refers to the most common characteristics from all available lists (Figure 40). The majority of the rooms have free cancellation and self-check-in. In contrast, the most uncommon characteristics are great location, experienced host and remote workplace.

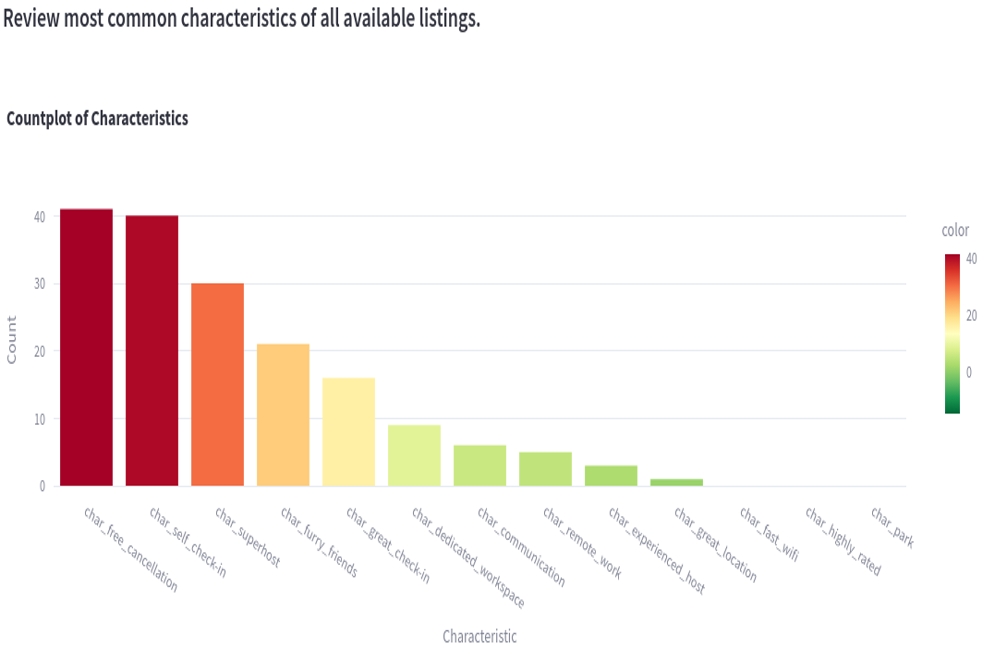


Figure 40: Countplot of most common characteristics

Regarding the correlations, Figure 41 depicts that there is a strong correlation of Price and Visitors. Moreover over review index and review number present a slight negative correlation.

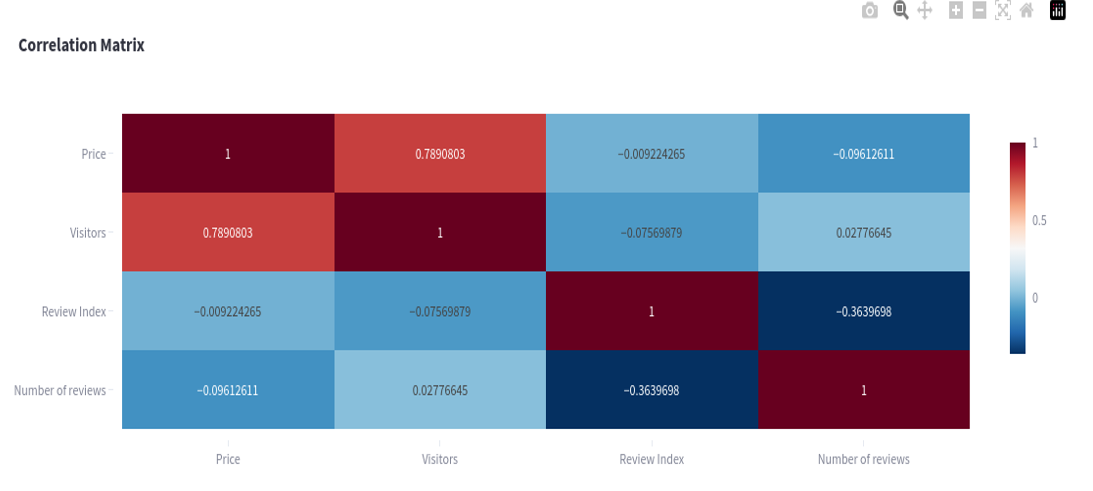


Figure 41: Correlation matrix for price, visitors, review index and number of reviews

Finally, Figure 42 illustrates that remote work and free cancellation seem to have a slight negative correlation with the review index.

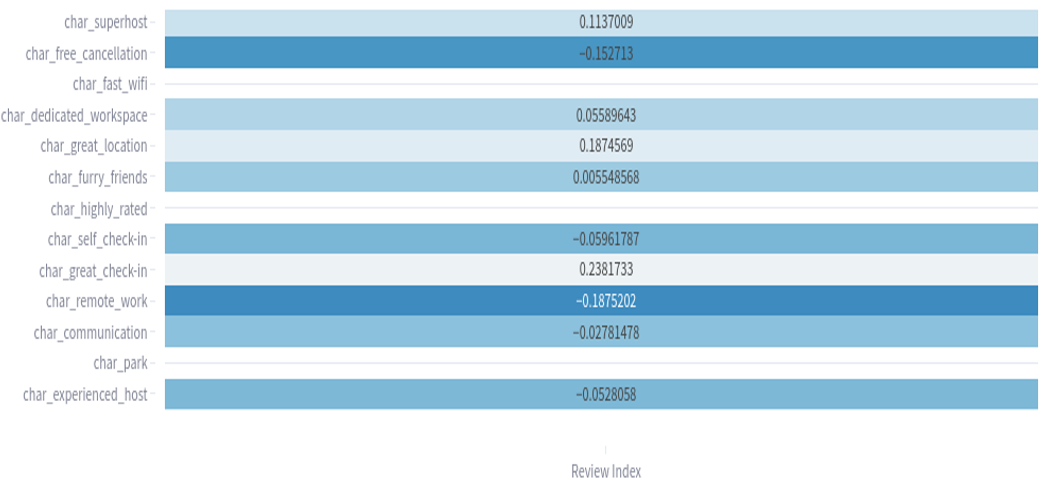


Figure 42: Coorelation matrix of characteristics and review index

# Insightful conclusions

To sum up, a scraping process to extract data from the AirBnB platform is created, data are cleaned and stored inside a Mongo DB and useful insights are unveiled with Streamlit visualization. As a challenge, it is to underline the limited data after the preprocessing and the high execution time of the proposed solution. As the next step an evaluation of the ML model must be conducted through corresponding metrics and fine-tuning of hyperparameters. Hope you enjoyed this article.

# References

[1] Airbnb, “https://www.airbnb.gr/,” 2024.

[2] Selenium Docs, “https://www.selenium.dev/.”

[3] Beautiful Soup Docs, “ https://www.crummy.com/software/BeautifulSoup/bs4/doc/.”

[4] MongoDB Docs, “https://www.mongodb.com/,” 2024.

[5] Streamlit Docs, “https://docs.streamlit.io/get-started/tutorials/create-an-app.”

[6] GeoPy Docs, “https://geopy.readthedocs.io/en/stable/.”

[7] Plotly docs, “https://plotly.com/python/.”

[8] Folium, “https://python-visualization.github.io/folium/latest/,” 2024.