Hotel Pricing Estimation Database

Weishan He, Meichang(Scarlett) Ma, Pranjal Srivastava
March 13, 2021

Table of Contents

Executive Summary

FunTrip planned to launch a feature that estimated hotel pricing. The first version of the feature aims to provide an estimate of the cost of a 5-night stay in three popular tourist destinations: San Diego, Los Angeles, and Yosemite National Park. To obtain a more fair market price, the in-house analytics team leveraged web-scraping techniques to scrape the major competitor booking.com to obtain hotel information for these three prominent visiting destinations in California, thereby enriching the database. The data was stored in the non-relational database Mongodb and will be used by the analytics team to build a dynamic estimation model using popular machine learning algorithms such as linear regression, random forest, and XGBoost.

Business Requirements

FunTrip is an online travel agency that provides hotel reservations as well as other travel services. According to recent marketing research, many customers are hesitant to plan a trip because they lack a good estimate of their travel expenses. To encourage travelers, FunTrip intends to launch a new hotel pricing estimation feature, in which customers can obtain an estimated cost by simply choosing the desired hotel labels. Subsequently, the application can continue to make recommendations according to the preferences of the customers.

The company has managed to accumulate information about the hotels that they have partnership with. More data from rivals is required for a more appropriate pricing estimate. We started this project as the company's analytics team to build a database that provides hotel listing information for modeling.

Database Design

Booking.com is the major rival of FunTrip. We scraped the listings on booking.com from March 20 to March 25, for three different popular vacation locations in California – San Diego, Los Angeles, and Yosemite National Park. To accomplish this task, we went through five major processes.

Firstly, we used Selenium to open a browser and manually navigate to booking.com. By doing so, we found that booking.com is not actively detecting Selenium and hence it is a good target for web scrapping.

Secondly, we leveraged Selenium to automate the search procedures, including typing in the trip destination, selecting the check-in and check-out dates, and ultimately reaching the result pages. We then saved the first 5 result pages for each of the three locations to disk.

Thirdly, we utilized BeautifulSoup to pulled out the saved pages and parsed out critical information including location, hotel name, link, distance from center, price, review score, number of reviews, star ratings, free breakfast, and free cancellation options.

The fourth step we took was to query each hotel address's geolocation via the API provided by positionstack.com. Steps 1 - 4 have been implemented in the DDR_Final_Project.ipynb file.

Lastly, we create a MongoDB collection called "spring_vacation_hotels" that stores all the information we collected for each hotel, and we exported the collection as a JSON file. Figures 1 - 3 in the appendix show a demonstration of sample documents for each location. The final database contains the following attributes:

- *_id* (Int32): it is a unique identifier of all the documents in the database

- *Location* (String): It indicates the location of the hotels. In this database, we have three locations, San Diego, Los Angeles, and Yosemite National Park.
- *hotel_name* (String): name of the hotel
- *hotel_link* (String): an URL that leads to the result page of each hotel
- *hotel_address* (String): the address of the hotel
- *hotel_geolocation* (String): the longitudinal and latitudial of each hotel
- *hotel_distance_from_center* (String): it indicates the distance between the hotel and the city center. This attribute is null for all the hotels in Yosemite because no information is shown in the web page due to its geographical peculiarities.
- *five_nights_prices* (String): total cost of five-night stay of each hotel in US dollars
- *hotel_score* (String): the score given by the booking.com users for each hotel
- *hotel_total_reviews* (String): the number of reviews for each hotel
- *hotel_star_rating* (Int32): the star rating of the hotel
- *free_breakfast_option* (Int32): the hotel offers breakfast (1) or not (0)
- *free_cancellation_option* (Int32): the order can be cancelled for free (1) or not (0)

Database Application

The analytics team plans to develop a dynamic hotel pricing model by using linear regression model, random forest, and XGBoost machine learning algorithms, in order to predict the 5-night cost of various types of hotels that meets the needs of the customers. Features in the database, including *hotel_score, hotel_total_reviews, hotel_star_rating, free_breakfast _option, and free_cancellation_option* will be the candidates for explanatory variables. The model is dynamic because the choice of input variables depends on customers' preferences. Specifically,

the new hotel pricing estimation feature will provide some options for the customers to choose from. When one option is selected, the corresponding variable will be included into the estimation model. That is to say, those options left unselected will be excluded from the model. The estimated 5-night cost of the hotel will immediately pop out one the screen for the users' reference. The options we will include in the first pilot are:

- Destination: San Diego, Los Angeles, and Yosemite National Park. In this option, customers can choose any places they want to visit among the three locations. The *location* attribute in the dataset will be included in the model.
- Number of people: for the pilot, we only offer estimation for 2 people. As the project eveloves, more options and data will be available.
- Duration: for the pilot, there will only be estimation for 5-night stay. As the project eveloves, more options and data will be available.
- Hotel star rating: customers will be able to choose the star rating of the hotel from the drag down menu. This option is related with the *hotel_star_rating* variable in the dataset.
- Distance form the center: customers will be able to select different range of distance from the center. The attribute *hotel_distance_from_center* will be included in the model if customers pay attention to the accessibility to the city center.
- Popularity: we measure popularity of the hotel by the number of reviews. We classify the hotel into three categories - Hot, Normal, and Niche ranking from large to small number of reviews. We can leverage the *hotel_total_reviews* feature to determine the thresholds of classification. And the same feature will go into the model if popularity option is selected.

- Free Breakfast: customers can select yes or no to indicate whether they want to include breakfast service when booking hotels. The *free_breakfast _option* attribute will be used here.
- Free Cancellation: customers can select yes or no to indicate whether they want to include free cancellation right when booking hotels. The *free_cancellation_option* attribute is related to this option.

When finishing model construction, we will apply the model on the testing data to obtain out-of-sample predication and compare the result with the prices recorded in the *five_nights_prices* attribute to evaluate the accuracy of the model.

Following the presentation of the estimated price, the app will recommend some potential hotels that meet the customers' needs, including *hotel_name*, *hotel_address*, and *hotel_geolocation* information.

We chose non-relational databases over relational databases due to their superior scalability performance. Aside from Booking.com, we have plenty of other competitors, including Expedia and Airbnb. Furthermore, all the platforms tend to add more information that helps customers to make a decision. In terms of both features and observations, the database will continue to grow. Non-relational databases can handle large amounts of data quickly and without regard for data structure. It will save us a significant amount of time in data management, allowing us to focus more on model accuracy.

Summary & Conclusion

In this project, we leveraged web scraping techniques to obtain hotel information from the key competitor, booking.com, and stored the data in a non-relational database Mongodb. The

data will be utilized to train a hotel pricing estimation model, which will serve as the core of the hotel pricing estimation feature that FunTrip aims to launch. In order to facilitate a more accurate estimation, we will use the same technique to obtain data from other major competitors' websites, such as Airbnb and Expedia. Web scraping significantly improves the efficiency with which public information is acquired, and the non-relational database simplifies the process of augmenting the dataset.

Appendix

Figure 1: Hotel Information Document for Los Angeles

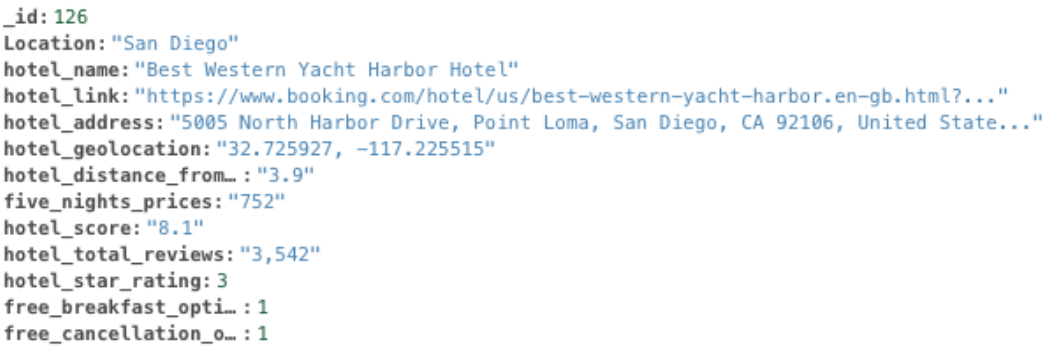

Figure 2: Hotel Information Document for San Diego

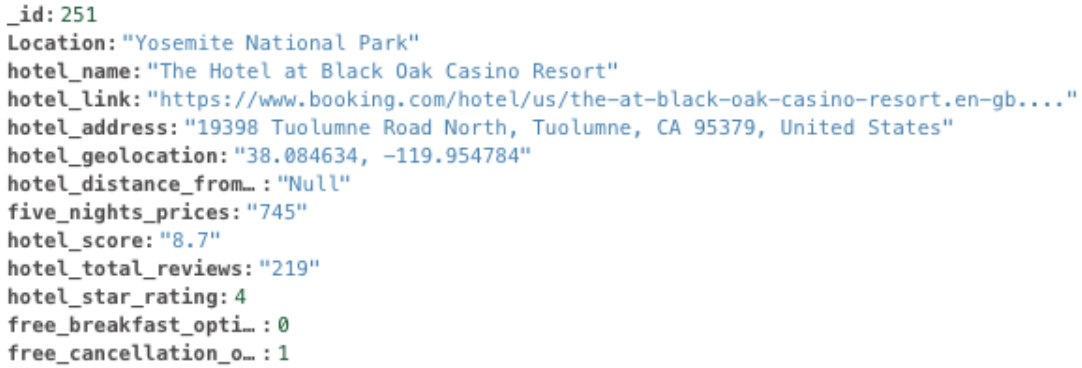

Figure 3: Hotel Information Document for Yosemite National Park