**Airbnb Listing Price Strategies**

**Data Science Problem and Background**

Our project aims to explore the Airbnb listings in the city of New York to better understand that how different factors like bedrooms, location, house type amongst others can be used to predict the price of a new listing that is optimal in terms of the host lucrativeness and still be affordable to the guests.

Airbnb is a well-known online marketplace and hospitality service, enabling people to list or rent short term lodging. It is a platform that connects people looking to rent out their homes to earn income with people who are looking for convenient and affordable accommodations. People use Airbnb for the purposes for traveling, business, and homestay. However, many hosts and guests have less knowledge about how to fix a fair rental price. Thus, we think it is a good idea to make a price predictor to generate a fair price of Airbnb listings for reference.

We intend to make a model using machine learning with suitable attributes that can be considered while making a booking and will keep on refining it after applying algorithms to reach the optimality to give out the best price.

We aim to deliver a visualization of the Airbnb data through a GIS or a heatmap for prices, which will show the various listings with the accommodation types and their prices. We also aim to deliver city based predictive models for Airbnb price that gives us the idea of -- the best time to book with Airbnb, is it competitive to join Airbnb market in a city and what will be the best price to start Airbnb business in a growing city?

**Potential Analyzes**

In order to answer our data science question, we plan to obtain four datasets, which represents the four most important questions that a traveler will ask before traveling: “where to eat around”, “how to travel in a city”, “Airbnb or hotel”, “what can Airbnb provide me”.

* Airbnb listings for New York City which contains attributes like the area and type of the accommodation, the price of the house, the neighborhood around, the most convenient transportation to reach the stay, latitude – longitude, number of bathrooms, information of the host, etc. This is our main dataset through which we will examine the features of and around the house and will correlate it with the price that is mentioned.
* Yelp data for analyzing how popular is the location of the Airbnb stays. We will take into account the number of restaurants around the stay, how great are their reviews. This will give us an idea of how convenient it is to find good quality food around the stay and hence we can judge the footfall in this Airbnb location.
* Hotels’ price dataset provides the information about the price range of hotels in New York City. This dataset has variables like prices of hotels other than Airbnb homestays including hotel names and their zip codes. Since hotels are the main competitors of Airbnb, this data provides us the price range of accommodations in a particular area.
* Transportation/Utilities data will provide information about the closest subways to the Airbnb stay, and train/bus stations nearby. This will help us to deduce the convenience of reaching a Airbnb stay and to travel to other locations around the city and hence we can comment on the suitability of a particular accommodation.

**Collecting New Data**

We have used python to collect our data by using APIs of websites as well as scraping directly.

* Airbnb Listing - To collect Airbnb data, we use Scrapy along with Scrapy-Splash. Scrapy is a Python application framework for extracting structured data from websites. It provides crawlers called Spider, to scrape the website by user-convenience defined functions. Scrapy-Splash is a Python library provides JavaScript integration for Scrapy. Therefore, we use docker to run a Splash HTTP API, to provide the working environment for Scrapy. To avoid banning by Airbnb’s anti-scraping detection, with the limit of one page per 6 seconds, we scrape the website at a rate around 500 per hour. We scraped everything listed on Airbnb listing webpage including: host’s general information (host response rate, host superhost status, listing counts, review scores, etc), location information (neighborhood, city, state, zip code, longitude, latitude, etc), room information (amenities, price, accommodates, rooms, cancellation policy status, etc).
* Yelp data - Restaurant data is directly scraped from yelp.com. Restaurants are selected based on distance to Airbnb houses, all within 2 miles of walking distance. Restaurants information are including restaurant name, rating, number of reviews, price range, address, etc. In order to scrape data, we use python package LXML, therefore we are able to parse the HTML Tree Structure using [Xpaths](https://www.scrapehero.com/xpaths-and-their-relevance-in-web-scraping/). With the help of the scraper, all information can be captured even in multiple different pages.
* Transportation data - Transportation data was first planned to retrieve through Walkscore API, which provides a convenience evaluation of a given location, by returning a “Transit Score”. We’ve finished the Python script for requesting transit score data, but unfortunately the API key that we requested is not working and we’re still waiting for Walkscore’s response. As a substitution, we’ve prepared a script to request Google Maps API data, which provides organized json data like nearby bus/metro stations with user defined location and radius. Data will be selected by using longitude and latitude of the Airbnb houses to measure. As the project moves on, we will either get the new API key and request data or switch to Google Maps API
* Hotels’ price dataset - As competitors, hotel is also affecting the pricing strategy of Airbnb houses. Price of nearby hotel is collected from Hotels.com, with information including hotel names, zip codes, prices, addresses, country names, region and rating. We use Chromedrive with Selenium, which provides methods to auto control Chrome with user defined functions to click, scroll, type, submit on the target webpage, and lxml to parse the target HTML page using Xpath.

**Data Issues**

One of the main issue of the Airbnb dataset is noise. There are many irrelevant information such as a description that a host puts on the page, review contents. There are also duplicated information such as market, neighborhood, host neighborhood, smart location, street, which contains very similar - sometimes the same contents. There are also wrong listings in the scraped data such as the listing city is New York but he zip code is in California. We call this directly scraped data metadata, and in order to perform better evaluation of the data, we will do a clean for the metadata. The clean for metadata consists of dropping irrelevant text contents, and exclude the error contents such as zip codes outside New York City. The hotel dataset is relatively simple compare to the Airbnb dataset. It only contains hotel names, zip codes, prices, addresses, country names, region and rating. For the planned analysis we need only the zip codes and hotel prices.

For Yelp data, the raw data is collecting all restaurant information base on a zip code. So, each zip code will have hundreds associated restaurants. Next step of cleaning will be grouping related data together. After counting the amount of high quality restaurants, this attribute will be assigned to corresponding Airbnb house base on the zipcode. Also, measuring restaurant quality based only on rating also has its limitation. Scale of rating is from 1 to 5, it’s obvious not the most accurate measurement. Therefore, other influence factors, such as number of reviews or price range, might need to be considered to determine the quality of restaurants.

**Data Cleaning**

During the data cleaning process, as we mainly have numerical, boolean, and categorical data, we build a well-organized function to detect the incorrect value and ‘null’ value based on above three types of data to analyze three different datasets.

Our function, named Score\_Cleanliness, will detect the ‘null’ value for each variable and check incorrect value based on different types of data, finally, it will calculate a final score. Our function will check below missing and incorrect values:

* Counting and Reporting all NULL for our data.
* Detecting Non-integer values for a Numerical Variable: Some variables will only have integer values, the function will detect those non-integer values. e.g., the variable, “accommodates”, could not have a value like 2.3.
* Finding Non-positive values for a Numerical Variable: Some variables will only have negative values, the function will detect the negative values. e.g., the variable, “beds”, could not have a value like -3.
* Searching Incorrect values for a Boolean Variable: After changing all uppercase of a string-type boolean variable into lowercase, the function will catch those values is not: “t”, “f”, “true”, “false”, “1”, or “0”.

Then we calculated how much percentage of data is cleaned.

Data Cleanliness Metric:

Final Score = (the number of missing values + the number incorrect values)/(total number of data)

For example, in our Airbnb\_listings data set, there are 227 missing values out of 48884 records and 16 incorrect values, then the score is 99.964%. After cleaning all incorrect data and missing data, the score after cleaning should be 100%. Therefore, compared with two scores, we can know if the dataset is cleaned or not.

**Feature Generation**

We implemented the web crawler to collect data from Airbnb. Our dataset has the really comprehensive data type. We got the attributes such as property type, room\_type, accommodation, and number of the bathrooms. They are categorical variables, boolean variables, and numerical variables. During the process of feature generation, we converted categorical variables to dummy variables. There are 30 properties types under the property\_type column, such as Apartment, house and cabin, so we construct each property types as individual column and convert them to dummy variables. For example, if the property type of this listing is a house, then the value only under the house column is 1 and is 0 in other property type column. Similarly, bed\_type and room\_type have the similar method to generate features.

**Reference**

1. ScrapeHero. 2018. How to scrape Yelp.com for Business Listings. (February 2018). Retrieved October 8, 2018 from https://www.scrapehero.com/how-to-scrape-yelp-com-for-business-listings/