Introduction:

The Superhost program at Airbnb gives the host more visibility, earning potential, and exclusive rewards. The requirements include 4.8+ overall rating, 10+ stays, <1% cancellation rate in the past year, and respond to 90% of new messages within 24 hours. However, the original intention of Airbnb is to promote more diverse traveling experience.

From Airbnb's 10K report: instead of traveling like tourists and feeling like outsiders, guests on Airbnb can stay in neighborhoods where people live, have authentic experiences, live like locals, and spend time with locals in approximately 100,000 cities around the world. From our point of view, the hard criteria rather promotes professional management teams who have more labor to respond to messages, decorate and clean the houses and even look for suitable places for their properties. However, the uniqueness of listings on the platform will be negatively impacted if the number of individual hosts does not grow at the same rate. Therefore, in this project, we are promoting a new way of identifying "Honest Host" whose reviews given by residents are matching with the host's description, using text data mining and clustering.

Potential Audience:

By identifying "Honest Host," Airbnb can in turn support individual hosts who are willing to dedicate leisure time to decorate their houses, manage their properties, and interact with interested residents. Living with this kind of hosts, guests will have the chance to feel like locals and interact with local residents. This undoubtedly gives a unique experience to people who choose Airbnb over traditional hotels. Therefore, our project can benefit Airbnb by encouraging more individual, diverse hosts to join. The results might be valuable for them to decide for new marketing campaign, or have more personalized marketing campaign for different host group.

Datasets

My data comes from Insideairbnb.com, which is an investigatory website launched by Murray Cox in 2016. The site was originally established by Murray Cox to identify possible illegal behaviors in NYC, where the same apartment appeared in different listings. The site uses the following Open Source technologies: D3, Bootstrap, Python, PostgreSQL, and Google Fonts, and is served by an Amazon S3 “bucket". Nowadays, Insideairbnb manages the data and updates the newly scraped data of dozens of cities and countries around the world quarterly for public uses.

For this project, I used two datasets: review.csv and listings-details.csv. “Listing-details” has dimension 38277*74, where each row represent a listing on airbnb and features include information of the listing, including bedroom and bathroom number, property type, and price, host information, and neighborhood characteristics. Wording information such as amenities, general description about renting place, neighborhood overview, and host's personal information was concatenated together to gather information given by the host.

“Review” has dimension 891964*6, where each row represents a comment to one listing. The columns includes listing_id, review_id, reviewer_id, date of review, and the comments. By joining these two datasets together, we connect the comments with the individual listings and make comparison between the description given by the host and the reviews given by past residents.

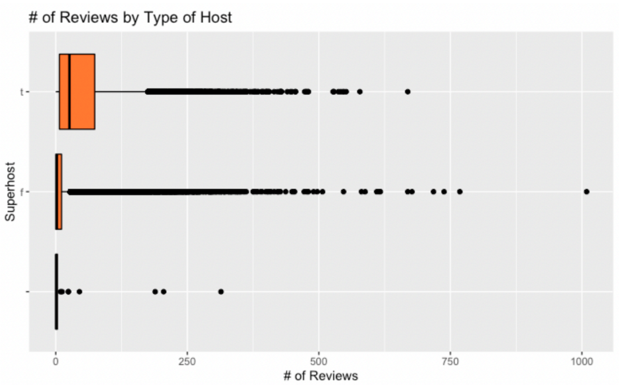
- Data quality issues

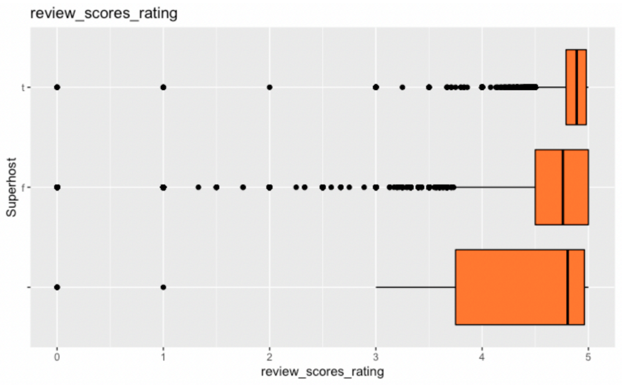
Features like “listing_url", “picture_url", “host_url", “host_thumbnail_url", “host_picture_url", and “calendar_last_scraped" were filtered out based on domain knowledge. By finding column-wise count of the NA values, I filtered out “bathrooms" and “calendar_updated" which consists of only NAs.

Furthermore, by checking the number of comments associated with each listing, we filtered out listings with at least 5 comments.

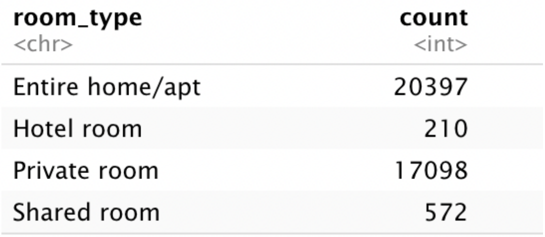
- Data summary and Visualization

Superhost characteristics:



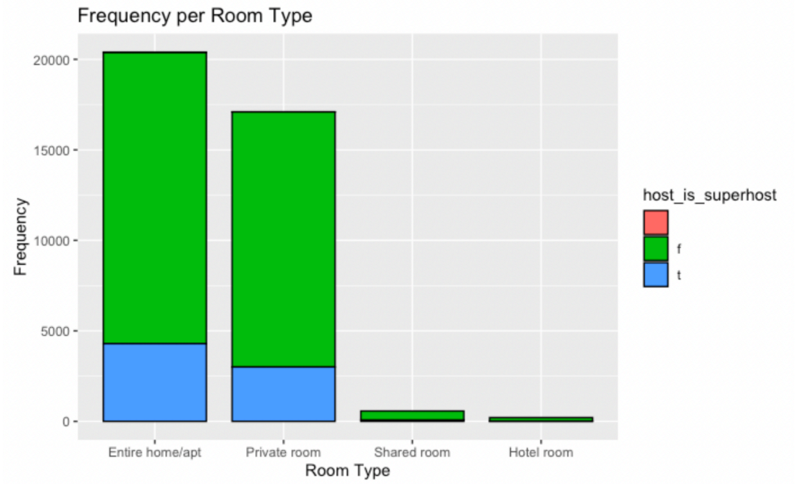


By the two boxplots above, it’s easy to see that if the host has a “SuperHost” badge, he or she will receive a significantly larger amount of reviews as well as higher review scores, even with a much lower variance. This coincides with what’s written on the official website: The Superhost program celebrates and rewards Airbnb’s top-rated and most experienced hosts.



room_type <chr>	count <int>
Entire home/apt	20397
Hotel room	210
Private room	17098
Shared room	572

In addition, most of the Airbnb listings are entire home/apt or private room. This is true for both Superhost and regular hosts.



Algorithm:

Information of each listings from both the hosts and residents was concatenated by each listing. We collected all descriptions written by the hosts, including description, amenities, neighborhood overview, and host about. For the information from residents of each listings, we concatenate all comments of the listing as the “documents” for future text data mining.

We did the following steps to clean up the text data:

- lower case all words
- remove html tags, such as <br>: the line break element
- correct the misspelled words using hunspell_check in hunspell package in R
- remove stopwords to avoid the most common but less meaningful words such as ‘of,’ ‘the,’ ‘and,’ and so on.
- perform stemming to avoid repeated words with different forms.

Furthermore, udpipe package was used to tokenize words in both the reviews and descriptions and then count the occurrence of each word in each documents. Two data frame were obtained with each row corresponding to each listing on the Airbnb website and each columns corresponding to each word appears in the respective text documents.

insert output data frames as example

Then, we used term frequency–inverse document frequency to calculate how important a word is to a document in a collection. Term frequency tf(t,d) is the relative frequency of term t within a document d and is calculated by the raw frequency divided by the raw frequency of the most frequently occurring term in the document. Inverse document frequency is a measure of how much information the word provides: whether it is common or rare across all documents. It is calculated by $\log(N/n_t)$, where N is the total number of documents in the collection, and n_t is the number of documents where the term appears. The final TF-IDF matrix was calculated by matrix multiplication tf * idf. High weights in TF-IDF are obtained by a high term frequency in a given document and a low document frequency of the term in a collection of documents. This tends to eliminate common terms.

Using the resulting TF-IDF, we cluster the comments and reviews using hierarchical clustering. Clustering results were compared with the dummy variable SuperHost in the listings details dataset: True if the host is SuperHost, else False. The resulting cluster was consistent with the label.

insert cluster dendrogram and comparison between the “True” label

I will complete this part later

- feature engineering: for each listing, we calculate “matching rate” as the amount of tokens in common divided by the total amount of tokens in both review and description

- clustering the matching rate. For this clustering, the resulting cluster is not consistent with the label. Therefore, SuperHost is not necessarily honest as defined earlier.
- Also tried bigram to gather information such as “not clean” but did not get much significant results

Conclusion:

- what we found
- future steps: due to computational constraint, we only run our algorithm on a small subset of the Airbnb listings. To further explore the similarity between description and review, we should perform a similar analysis on Airbnb listings in other major cities in the U.S.