Data Cleaning and Wrangling for AirBnB Price Prediction
Kseniya Kruchok

Data Source

For AirBnB price prediction I have downloaded three data sets from open data source on http://insideairbnb.com/get-the-data.html These data sets are:

- Listings - over 60K rows - information about each individual listing at the download time
- Calendar - over 22MM rows - information about listing availability throughout the year and corresponding prices at each date
- Reviews - over 1.1MM rows - information with transcript of all reviews to date

The data was downloaded on 2019-06-05 and has listing information on that date, availability calendar for the year after that, and information on reviews posted before that.

All files were in .gz compressed format because of the volume.

Cleaning Steps

There were several steps performed for each data set.

- Steps for Listings data set
    1. *Loading data* - read data from the file. There were originally 106 columns in data set.
    2. *Check data content* - I performed pandas_profiling on the data to identify columns that cannot be used in further analysis. Half of the columns were dropped after this step. Some of the columns did not contain relevant for price prediction information, like ‘host_id’, ‘scrapped_date’ etc. Others contained the same information for each listing, like 'host_acceptance_rate', 'has_availability' etc, or were highly correlated with remaining in data set columns, like 'availability_60', 'minimum_minimum_nights' etc.
    3. *Data transformation* -
        a. Explore ‘country’ column entry for Switzerland - confirming that it is in fact Paris listing, dropping ‘country column’
        b. Transform ‘neighbourhood’ and ‘host_neighbourhood’ columns into a new one ‘is_host_near’ indicating ‘t’ if entries in both columns match. Both original columns were dropped afterwards (there is also ‘neighbourhood_cleansed’ column that contains generalized entries for neighbourhoods)

    c. Transform ‘host_verifications’ and ‘amenities’ columns from list like strings to actual lists. ‘host_verifications’ was further transformed to the numbers of verification counts. I kept ‘amenities’ column as list of amenities for each listing and created a new column ‘amenities_count’ with the same transformation as for host verifications
    d. Transform price containing columns (‘price’, ‘security_deposit’, ‘cleaning_fee’) from strings to floats

4. *Filling Missing Values* - divided columns with missing values into several categories(based on the data to be filled in each of them) and with the help of defined function filled NaNs as below
    a. Empty string ‘’ for columns 'name', 'summary', 'space', 'description', 'transit', 'house_rules'
    b. ‘f’ indicating False for columns 'host_is_superhost', 'host_identity_verified'
    c. ‘unknown’ for 'host_response_time'
    d. Mean or median for 'host_response_rate', 'host_since', 'review_scores_rating', 'review_scores_accuracy', 'review_scores_cleanliness', 'review_scores_checkin', 'review_scores_communication', 'review_scores_location', 'review_scores_value'
    e. Maximum of ('accommodates'/2) and 1 for ‘bedrooms’
    f. Maximum of ('bedrooms' - 1) and 1 for ‘bathrooms’
    g. Maximum of 'bedrooms' and 1 for ‘beds’
    h. Mode for 'cancellation_policy' and 'zipcode'
5. *Exploring outliers* - the most important exploration for outliers was done for ‘price’ column. The original data was extremely skewed and therefore I transformed it to log prices and distribution curve looked closer to normal distribution one. There are three types of listings offered: ‘entire home/apt’, ‘private room’ and ‘shared room’. It makes sense to explore corresponding prices separately. I defined a function that identifies outliers based on their distance from the mean (3*standard deviation(std) for left outliers and 4*std for right ones as data is still skewed) and removes outliers from the dataset if they make less than 1% of the data. After this function was applied on prices for each room type category, outliers in ‘entire home/apt’ and ‘private room’ (about 0.4% of the total data combined) were removed. ‘Shared room’ outliers were slightly over 1%(just 5 actual listings) and further exploration allowed to drop them as well.

A more general function (with lower outliers acceptance of 0.5%) was defined to explore outliers in 'accommodates', 'bedrooms', 'bathrooms', 'beds', 'minimum_nights', 'maximum_nights' columns. As output it would provide information about outliers quantities and range and made suggestions on keeping or removing outliers. 'bedrooms', 'bathrooms', 'minimum_nights', 'maximum_nights' showed to have less than 1% of outliers combined (each not more than 0.5%) and they were dropped. 'accommodates' and ‘beds’ columns had more outliers and further exploration showed them to be valid entries, thus they were kept

- Steps for Calendar and Reviews data
    1. Loading data
    2. Dropping missing values as their amounts were insignificant given datasets volumes
    3. Data transformation
        a. Price columns: from strings to floats
        b. Date columns: from strings to datetime objects
    4. Merging with listing_ids from Listings data set to ensure that dropped values from main data set are not used in these ones.