Los Angeles Airbnb Price Prediction

Problem statement:

Airbnb is an online marketplace for short-term lodging and tourism related activities. It provides a platform for hosts to list their real estate while it's empty, offering tourists a home-like experience. In order to generate more profit for the host and for the home-sharing company, Airbnb introduced a Price Tips feature that provides hosts with recommended time-varying prices to increase the likelihood of getting a booking.

The aim of the project is to understand the factors that affect the price of Airbnb listings in Los Angeles- one of the top tourist cities, predict the price of listings and investigate the effect of overpriced and underpriced property on its future availability. The model not only provides insight to customers so that they have the resources to plan their travel lodging in advance and make better informed decisions, but also guides hosts to set the price that increases the booking rate.

Even though the Airbnb listing price prediction has been a popular topic for data scientists for a while, there aren't any online documentations of a project for Los Angeles specific Airbnb listings price prediction. Therefore, I decided to dive into the Los Angeles Airbnb listings, and build prediction models on the data.

Dataset Description:

The data is publicly accessible from the InsideAirbnb website: (http://insideairbnb.com/get-the-data.html)

Monthly updated datasets are available from May 2015 to Dec 2019.

For the scope of this project. The Listings.csv files from January 2016 to December 2019 are used. The csv file contains 106 variables for each listing ID. The variables include: name, description, host_id, bedrooms, bathrooms, number_of_reviews, neighbourhood_cleansed, longitude, latitude, host_id, cleaning_fee and price etc.

Data Collection and Wrangling:

Collection:

I wrote a simple web scraper and utilized the BeautifulSoup package, to retrieve all the csv files that are Listings files for Los Angeles.

Wrangling:

The wrangling of the dataset involves the following steps:

1. Importing only the necessary variables.
2. Removing outliers.
3. Filling/removing NaNs and zeros.
4. Transforming variables.

In practice, some data wrangling steps were performed after some data exploration and visualization that helped me better understand the distribution of variables and the real-world implication of the data.

Specifically, for the removing outliers' step, the following outliers are removed:

1. One listing indicates that it has 50 beds, 1 bedroom and the price is only $50, which is very likely to be an incorrect data point.
2. One listing indicates that it has 21 bedrooms, but only 1 bedroom instead.
3. One listing has reviews per month value of 48.38, which is too large to be reasonable, given there are only 30 days in a month.
4. By plotting the cumulative distribution of the listing prices, I found that the vast majority of the listing prices are below $2000. However, the highest listing price is $25000. Since the characteristics of the most luxurious listings may not be captured in our dataset, the listing prices above $2000 are considered outliers in order to ensure the quality of the prediction model we build in the next step.
5. By looking into the listing description and logging into the Airbnb App, I found that some high-quality properties have very low listing prices as low as $10. It is because some properties are only rented monthly. The host set a very low price for the initial date and increased the price gradually across the month. In addition, for these listings, the cleaning fee is multiple times higher than the daily fee. To capture these listings, I created a fee_price ratio variable (cleaning fee/listing price). The listings with the ratio higher than 2 are considered outliers. The fee_price ratio variable is later removed for the prediction model.

For the filling/removing NaNs and zeros step:

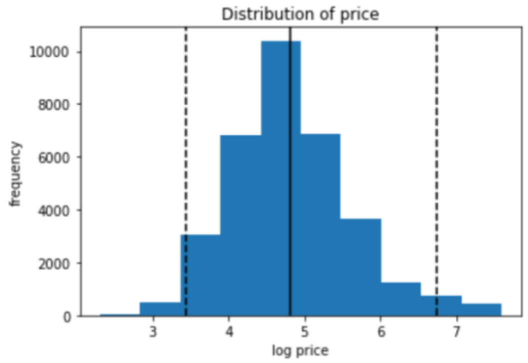
1. Remove listings where the prices are zero.
2. Fill empty security deposit, cleaning fee and reviews per month with zero.
3. Impute empty review scores by means of the review category.
4. Drop observations where the bathrooms, bedrooms and beds are NaN, because these variables are potentially important predictors of the dependent variable.
5. Confirm that rows that miss first_review/last review values are those listings without reviews. Create a review_timespan variable (last_review-first_review) and fill the empty values with zero.
6. Host related columns: create a new column host_info with value 't' if host related columns are not empty, 'f' if the host related columns are empty; then fill in host_is_superhost with 'f', host_listings_count with 'f' and host_has profile_pic with 'f' for NaN values

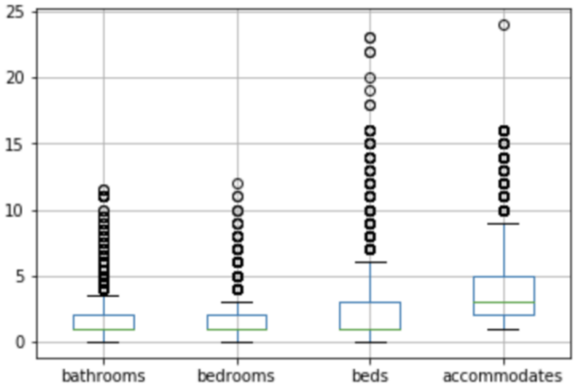
For the transforming variables part, the following steps are conducted:

1. Convert the binary variables of 't' and 'f' to 0 and 1.
2. Remove the '$' sign in the price variables and convert the type to numerical.
3. Convert the date variables to datetime object.
4. Create dummy variables for categorical variables with more than 2 categories. Perform special treatment on neighbourhood_cleansed and amenities variables:
    a. Neighbourhood_cleansed: count the number of listings in each neighbourhood, select the top 100 neighbourhood as separate categories, and group the rest together as 'other area'
    b. Amenities: transform the string of amenities to list, count the number of individual amenities across listings, select the top 20 amenities and remove the rest.

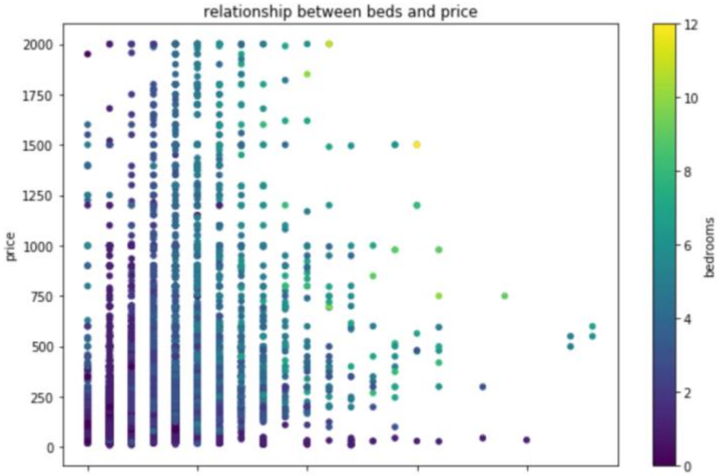
Exploratory Analysis:

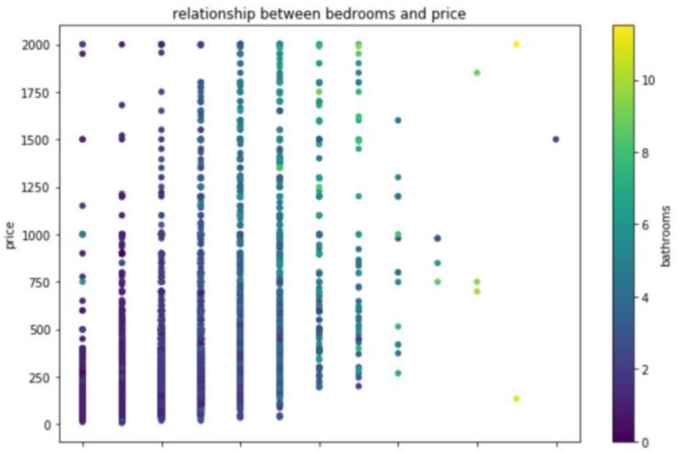
1. What are the distributions of price, bedroom, bathroom, beds, accommodate variables?



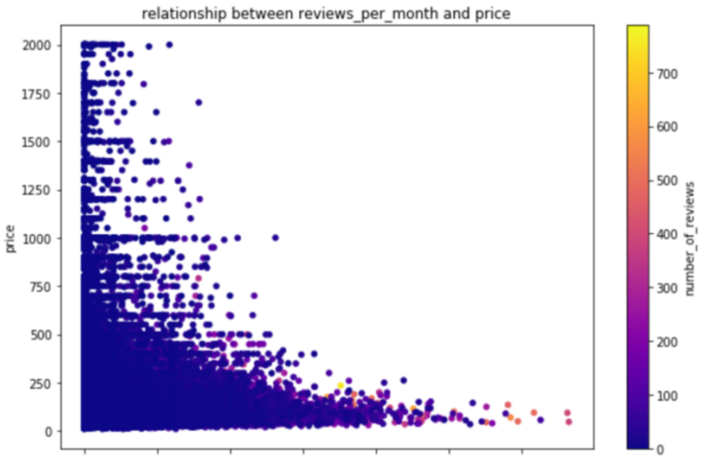


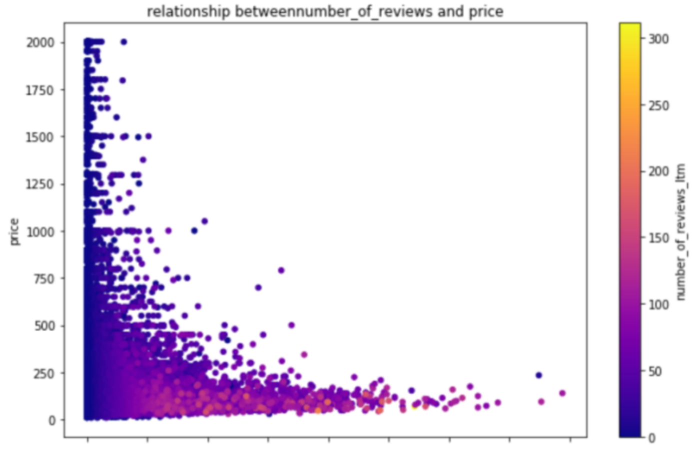
2. What is the relationship between bedroom, bathroom, beds with price?



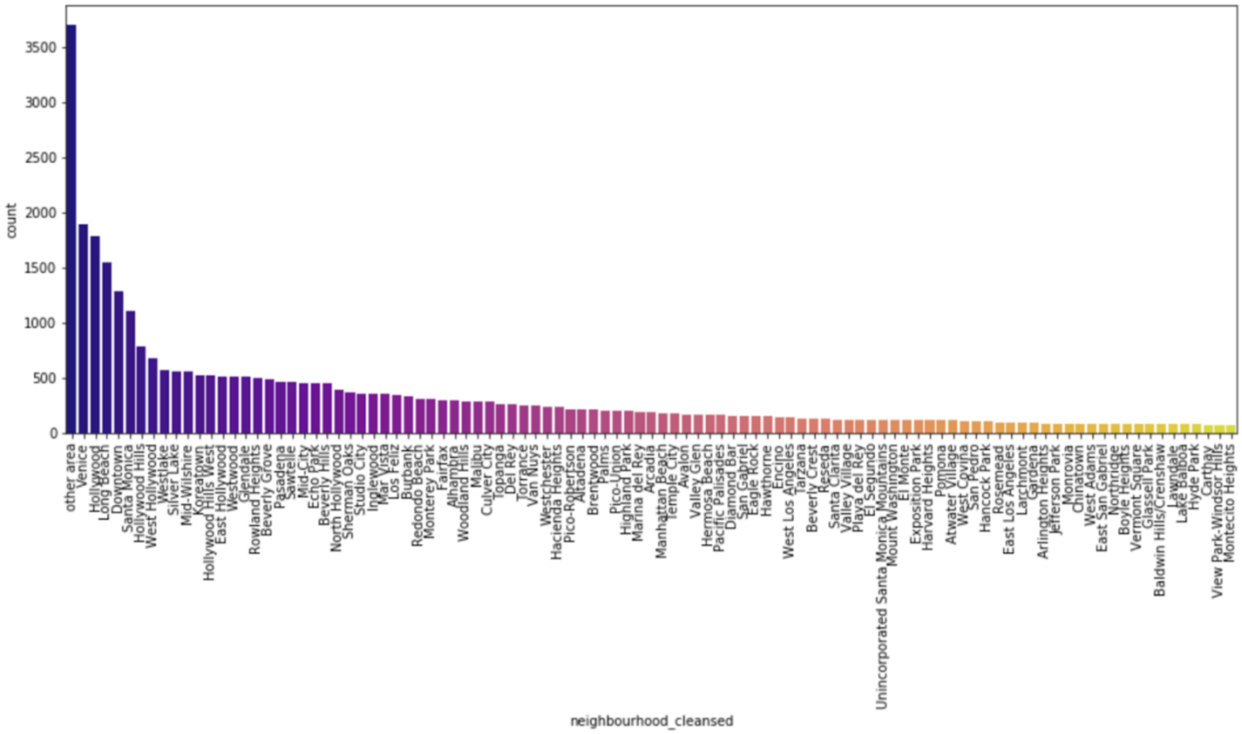


3. What is the relationship between number of reviews, monthly reviews, number of reviews ltm with price?

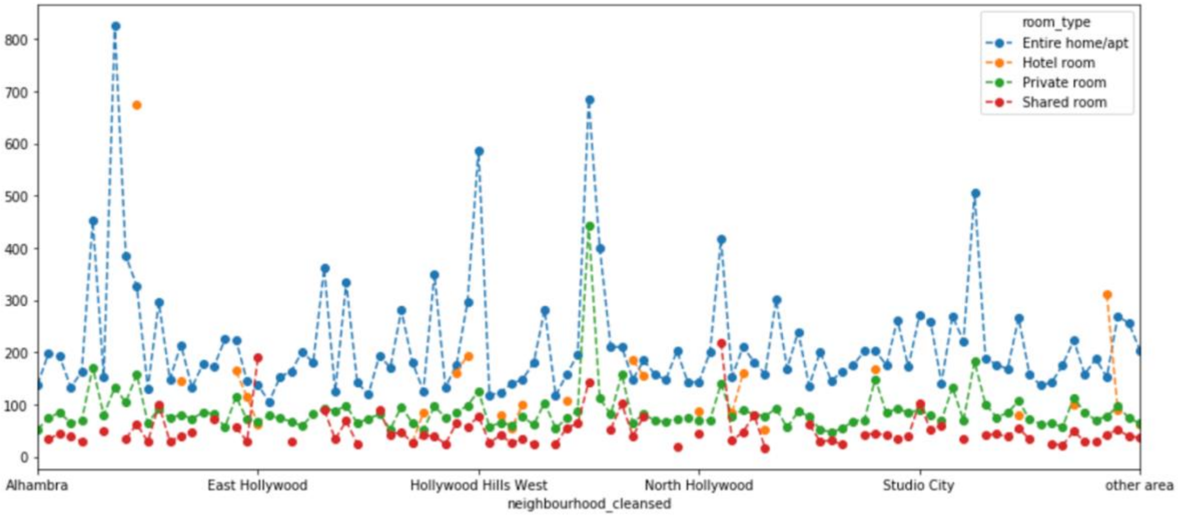




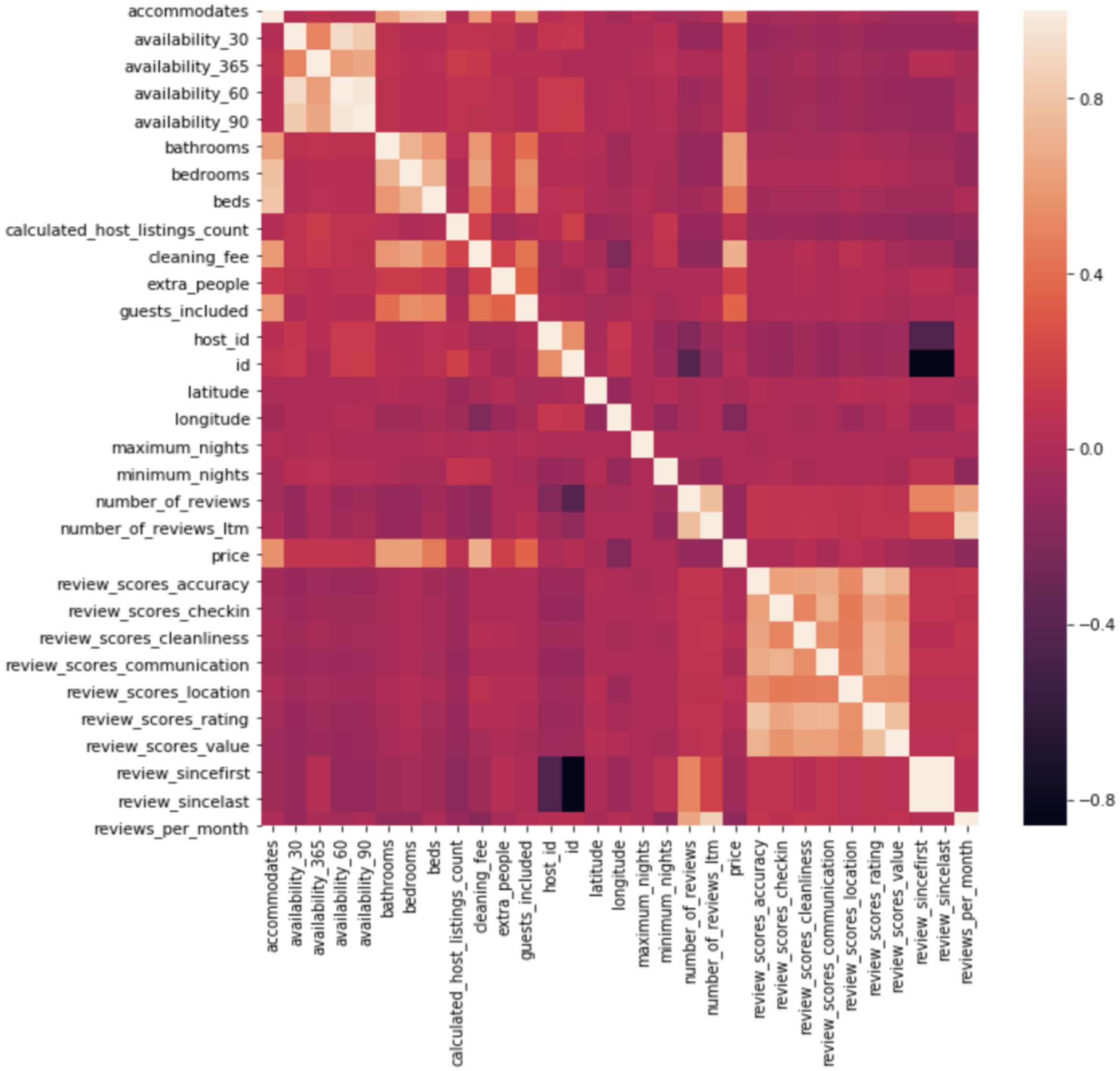
4. What is the count of listings by neighborhoods?



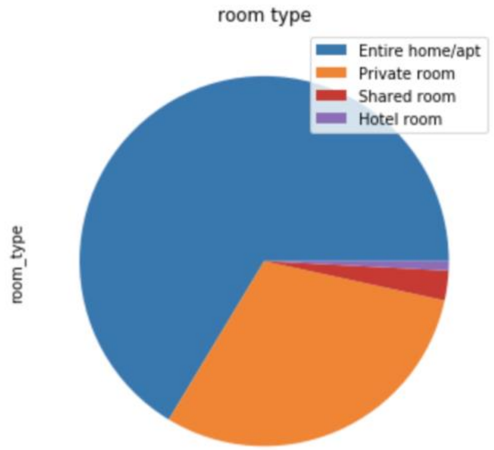
5. What is the mean price for different neighbourhoods grouped by room type?

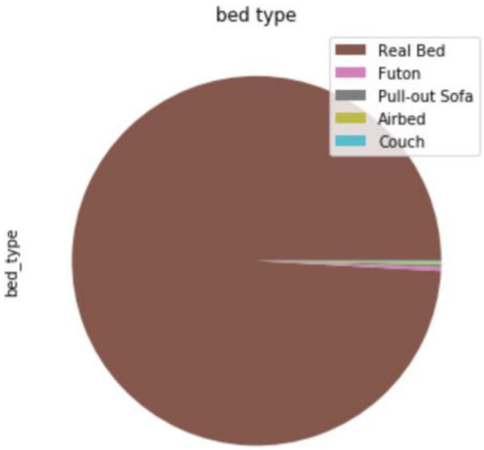


6. How are the variables correlated? (not plotting the dummies)

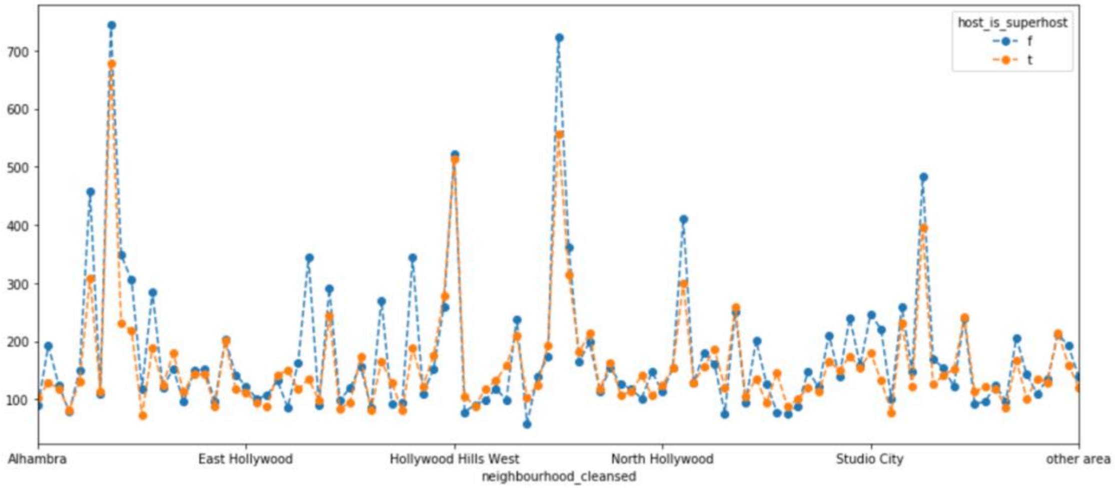


7. What is the proportion of listings for room types and bed types?

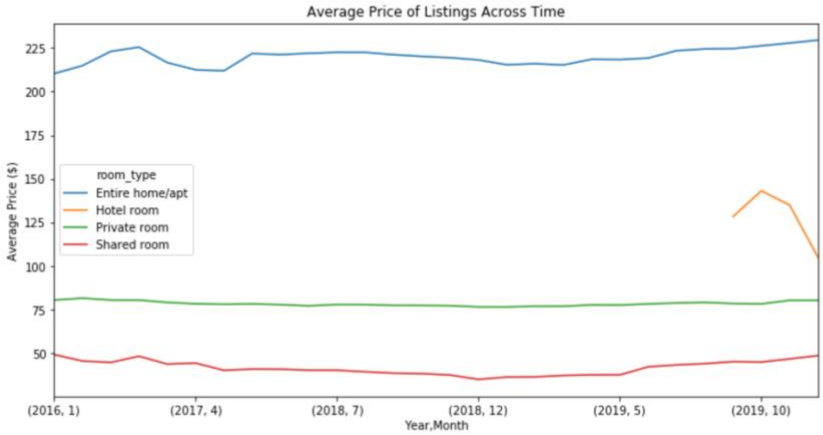


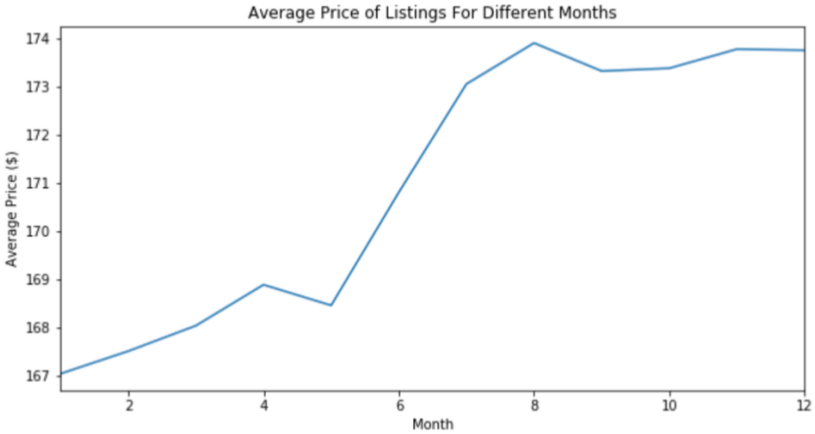


8. How is the factor super host affect price, grouped by neighborhood?

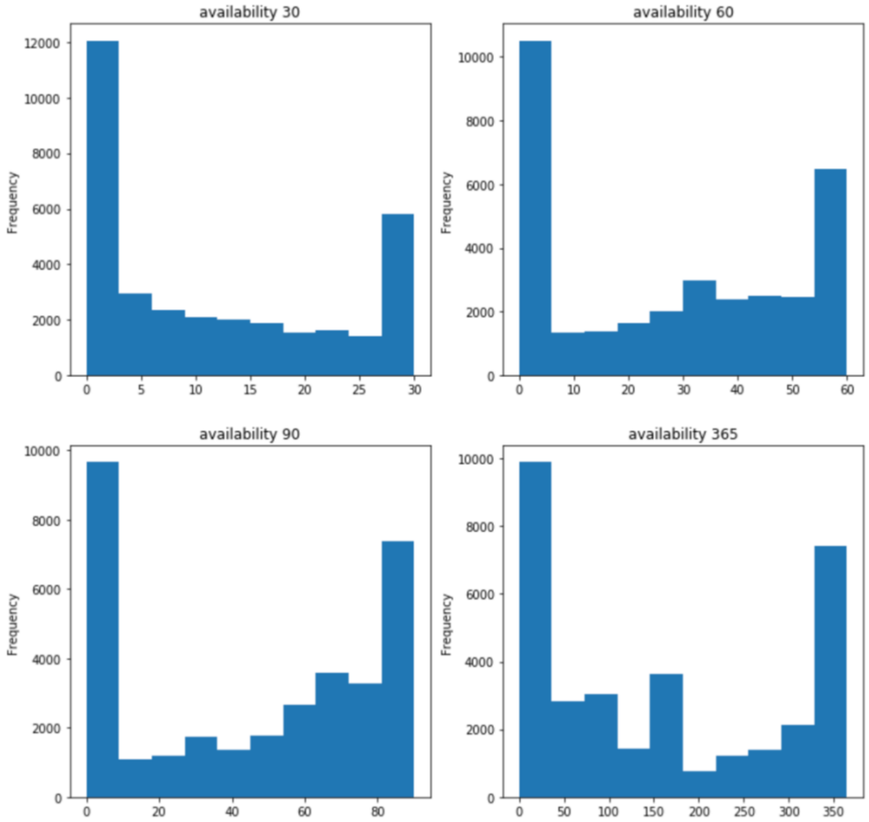


9. What is the time series trend of price from 2016 to 2019? How about the average trend of price by months?





10. What is the distribution of future availability for listings?



From the visualization above, there are several noteworthy findings:

1. Price is positively correlated with bedroom, bathroom, bed
2. Price is negatively correlated with the number of reviews.
3. The top 5 neighborhoods besides the ‘other areas’ category are: Venice, Hollywood, Long Beach, Downtown, Santa Monica.
4. Entire-room is the most expensive room type, while the shared-room is the cheapest, across neighborhoods. Entire-room and Private-room accounts for the majority of listings.
5. Listings by the super host are cheaper than listings by non-super host across neighborhoods.
6. The average price is stable across years, with a slight increase towards the end of 2019. The price is highest in December and cheapest in January. When building the model, I take the seasonality effect into account.

Machine Learning and In-depth Analysis Methodology:

The problem is a supervised, regression type machine learning problem. The goal is to use the features in the cleaned dataset to predict the price of the listings. We start by building and training the model on the December 2019 data, and then use the model overlaid by seasonal adjustments to predict the price for other months.
Since the distribution of the dependent variable-listing price is positively skewed, and the single day price might not be an accurate reflection of the cost of a total trip, multiple transformations of the price are computed and used as the dependent variable:

1. Price
2. Log price: monotonic transformation of prices.
3. 3-day price plus cleaning fee: cleaning fee can be a big part of the total cost.
4. Log 3-day price plus cleaning fee

The models constructed include linear and non-linear models: Linear models include Elastic Net, LASSO and Ridge model. Non-linear models include Support Vector Machine, Random Forest and Gradient Boosting model.

70% of the data is randomly fed into the training set, while 30% is kept for the testing set. To choose the optimal hyperparameters for the model and prevent overfitting, GridSearchCV is used to perform the tasks simultaneously. For the linear models, highly correlated predictor variables are removed to prevent the problem of multicollinearity.

The evaluation metrics to compare the models is the R-squared on the testing set, because it measures the accuracy of the model and it is comparable for different selection of models and versions of the dependent variable.

Feature importance tables/graphs are obtained from each of the models, which shows the top contributing factors for the listing price.

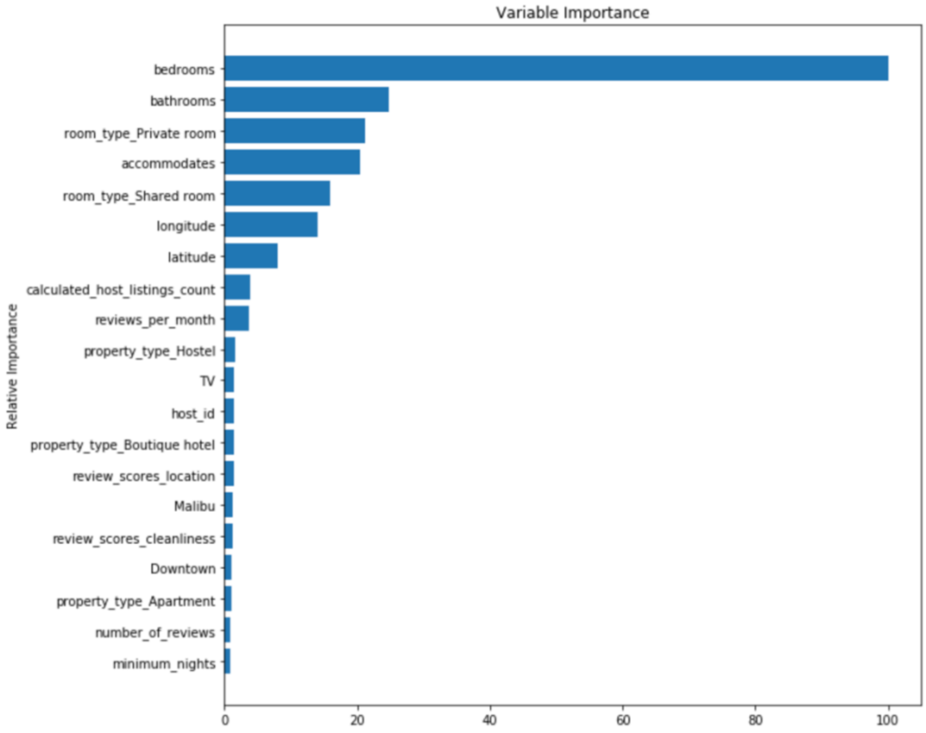
In addition, to validate the hypothesis that underpriced properties attract more customer bookings and overpriced properties cause less bookings, a linear regression is run on 30-day availability (days) against the difference between actual price and predicted price.

The last step is to apply the model on other months available in the dataset. The price data is scaled by the average price of the month/average price of December to adjust for seasonality, and then feed into the model.

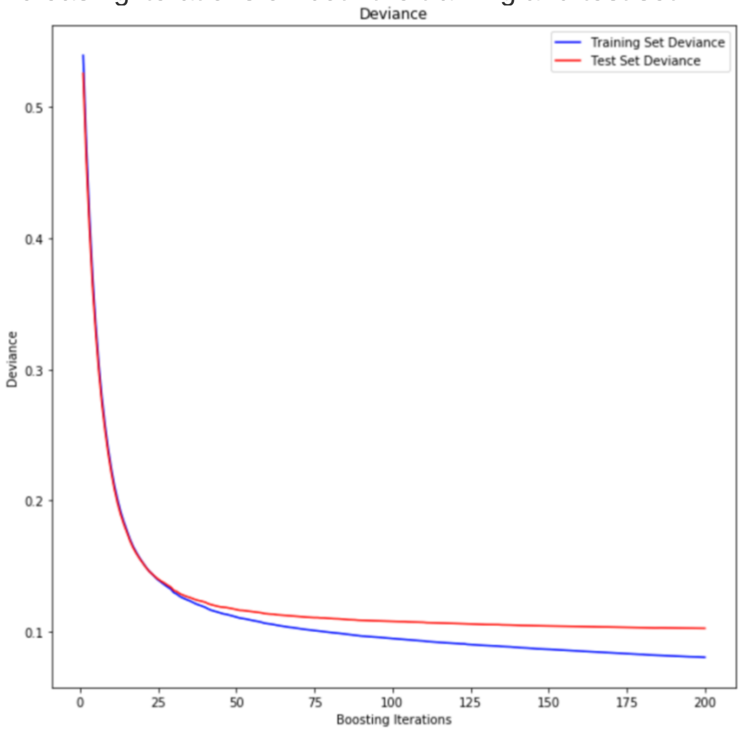
Model Result:

Using the log of 3-day total price (3*daily rate+cleaning fee) as the dependent variable results in the highest test set R-squared for all the models I tried. The best model is built using the gradient boosting algorithm, with an R-squared value of 0.836. The tuned best hyperparameters are: n_estimators: 200, max_depth: 5, min_samples_split: 2, learning_rate: 0.1 and loss: 'ls'.

The following graph shows the top variables ordered by their importance:



Deviance is the loss function for gradient boosting. The deviances follow very similar trends with increasing iterations on both the training and test set.



In addition, the prediction model using random forest algorithm also performs pretty well, which achieved an R-squared of 0.71 using log 3-day total price as the dependent variable; the prediction model using LASSO and Ridge achieved R-squares of 0.75.

The algorithm with the worst result is the Support Vector Machine. It is computationally inefficient to do cross-validation and grid search in the model, and the accuracy score is only about 0.5 on the baseline model.

The most important features vary across different models, but bedroom, bathroom, longitude, latitude, host-id and reviews per month are consistently ranked high in feature importance.

After the model is built, run a linear regression of the 30-day availability bucket on the difference between actual and predicted price. The p-value of the coefficient estimate is 0, which means that the coefficient is statistically significant. The result confirms the hypothesis that setting the listing price lower than the predicted price leads to less availability, which means more bookings. The host could gain an advantage by setting the price slightly lower than comparable listings to get more booking. The total revenue generated from the property might even be higher than setting a higher than predicted listing price.

Applying the model to the data from November 2019, an R-squared value of 0.79 is achieved. The relationship between availability and underpricing/overpricing still holds and is statistically significant.

Future Work:

1. The name and description of the listings could be processed into features to predict the price of the listings. A well written description would show the professionalism of the host, and it might impact the pricing behavior.
2. Los Angeles is one of the most sprawling cities in the world, which adds difficulty to the identification of the neighborhood. The neighborhood data could be manually processed and regrouped to less categories to increase the predictive power of the models.
3. Quantify how much lower the host set the price below the predicted price to generate a higher revenue (price* no. of booking days) over a month/year.
4. The model could be used to predict future data gathered in 2020 and in earlier years 2016-2018 to test if it would be generalized.