Capstone Two Project – Final Report

Background and Problem Identification

Airbnb is one of the innovative phenomena in the sharing economy. It's an online marketplace for vacation rentals, homestays and tourism activities. Although Airbnb enjoys around 5% of the tourism accommodation revenue, setting a price for their listings has been a challenging task for Airbnb hosts. Inefficient pricing has led to a loss of 46% of additional revenue among some hosts (LearnAirbnb.com, 2015). This is because, unlike in the hotel industry that has professionals and benchmarking reports and other technical means to set prices, pricing for Airbnb listings is generally handled by the hosts themselves (Gibbs et al., 2018). In addition to this, producing a pricing model for an Airbnb listing is generally complex, because in addition to the inherent listing features, one needs to consider the demand for their listing too.

This project attempts to produce a pricing model to help new hosts in the New York area to determine a price for their listing based on several features. That is, this considers only the features that are inherent to the listing. For instance, the location of listing, type of housing, rating, available amenities, reviews about the listing, reviews etc. Also, this will help the hosts to identify which aspects or features should be given more priority over others to offer value for the price they are going to charge.

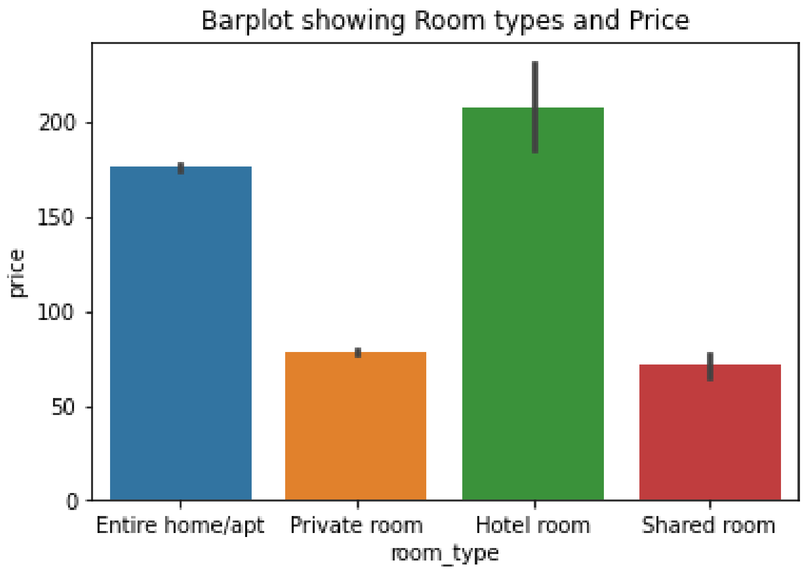
Data and Method

Data was obtained from an online data site (http://insideairbnb.com/get-the-data.html) which sources publicly available Airbnb listings' data using APIs. Data consisted of 36,000 listings (36,000 rows) from 5 different Boroughs of New York such as Manhattan, Brooklyn, Queens, Bronx and Staten Island. This data covers early periods of April 2021. There were around 72 columns describing features ranging from details of listing to information about host.

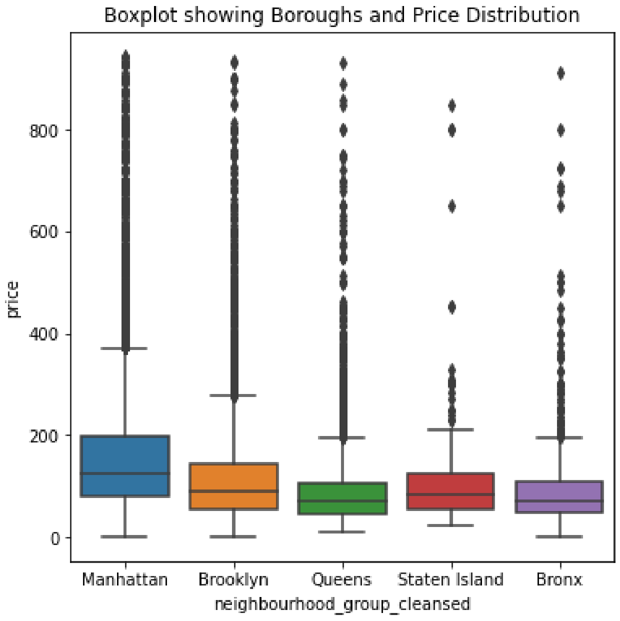
Data Wrangling and Exploratory Analysis:

After determining that the missing values in some of the features were missing at random and their missingness was below 10%, they were imputed with suitable values. For instance, number of baths and number of beds had missingness of 0.3% and 1.6% respectively, these were imputed with median values. Feature namely, if a host is super-host has missingness of 3.58%, this was imputed with 0s. However, rating score had 38% of missingness, since it's much above the threshold of 10%, it was imputed with values -10,000, so that decision tree model will treat them as missing values only.

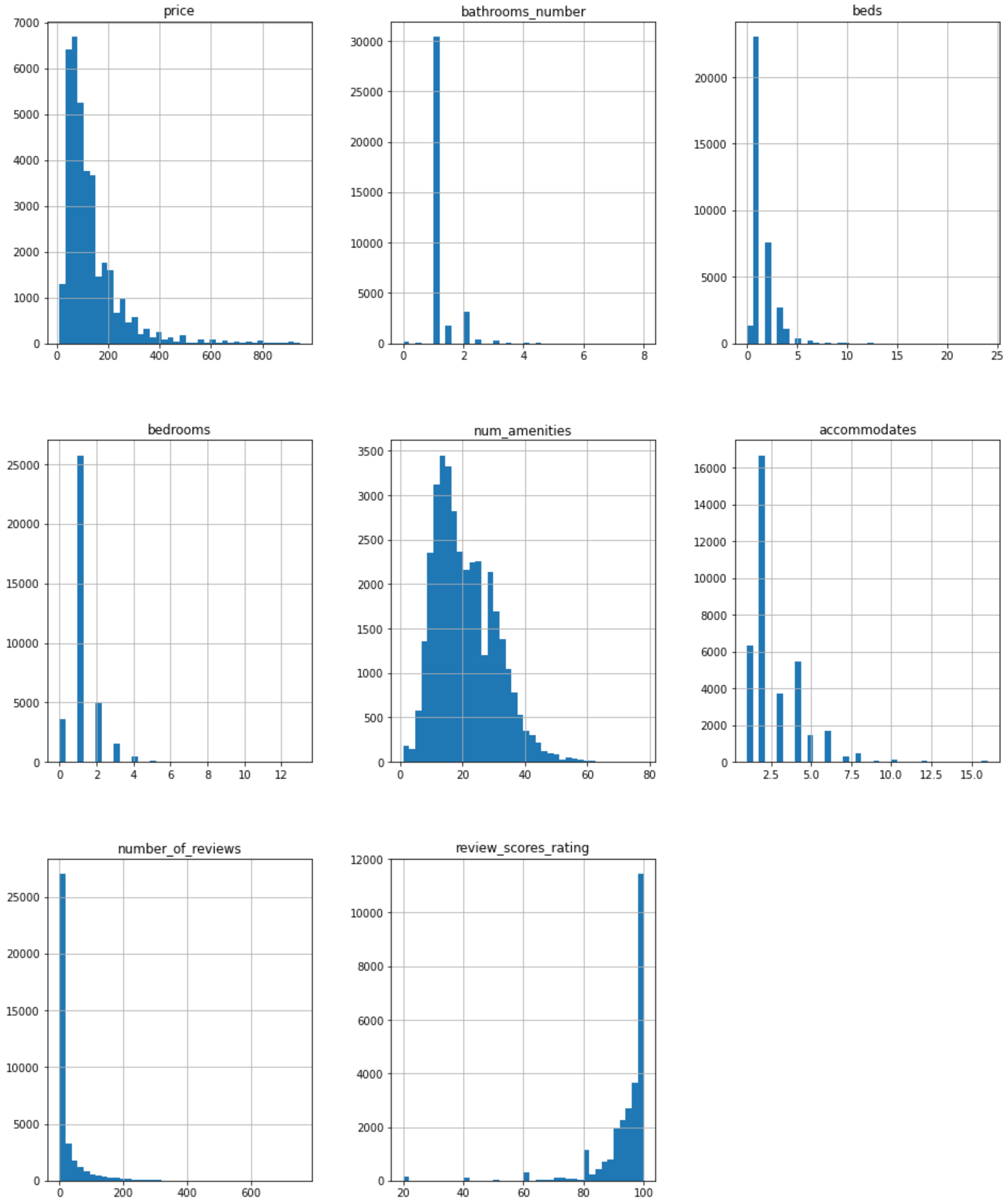
After careful data wrangling, I was able to retain a total of 16 features. Of them, 7 of them were numeric variables namely number of baths, number of bedrooms, number of beds, number of amenities, number of accommodates, number of reviews and rating scores. Rest of them were dummy variables reflecting location of listing (Manhattan, Brooklyn, Queens, Bronx, or Staten Island), type of listing (private room, entire home, shared room, and hotel room), host is super-host or not, and instant-bookable or not. While encoding for dummies, last category in each feature was dropped to avoid multicollinearity issues. Below is a bar plot, that shows how the mean prices differ between room-types. Hotel room prices tops the list followed by entire home prices.



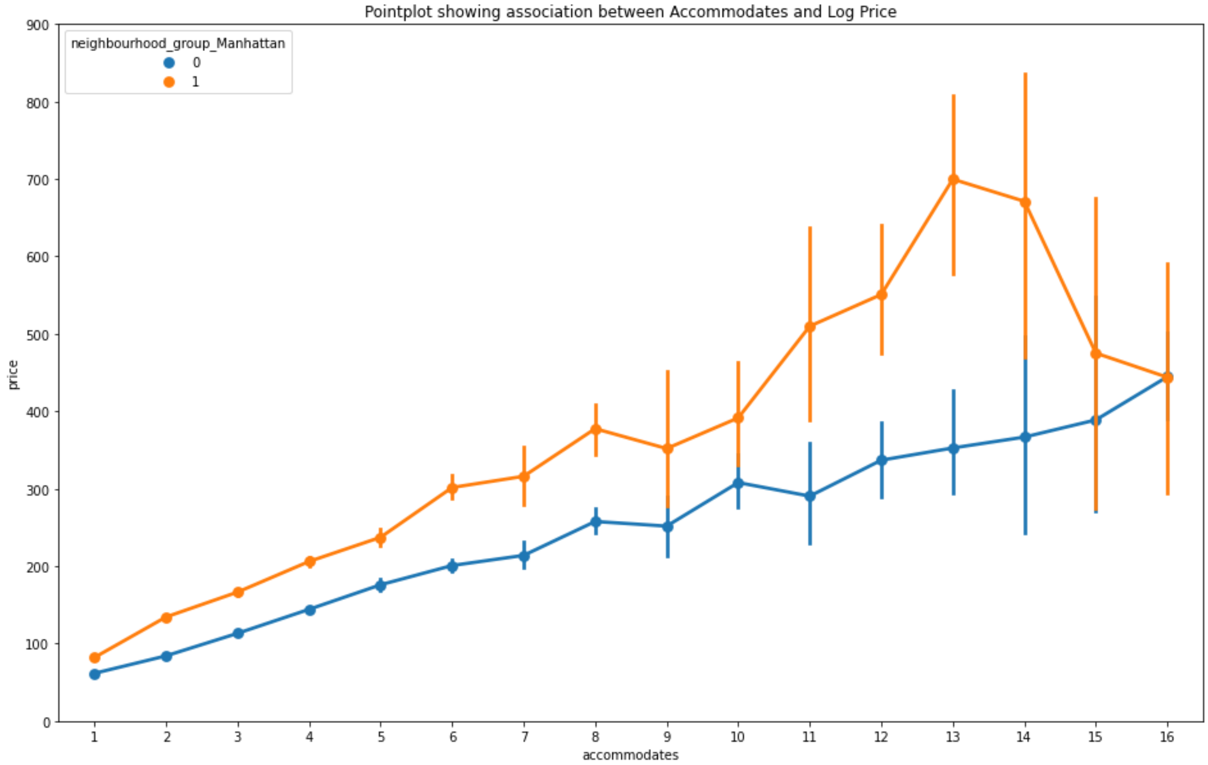
The above box plot shows the price distribution between different locations. Manhattan prices are the highest, followed by Brooklyn prices. In Queens and Bronx, the maximum prices are lower than the median price of Manhattan. Black thick lines in each plot shows the outliers in prices. This implies that the price distribution is skewed.



Following are some histograms showing the distribution of numeric features. Except for the number of amenities feature, all other features show either a left or right skewed distribution with outliers.

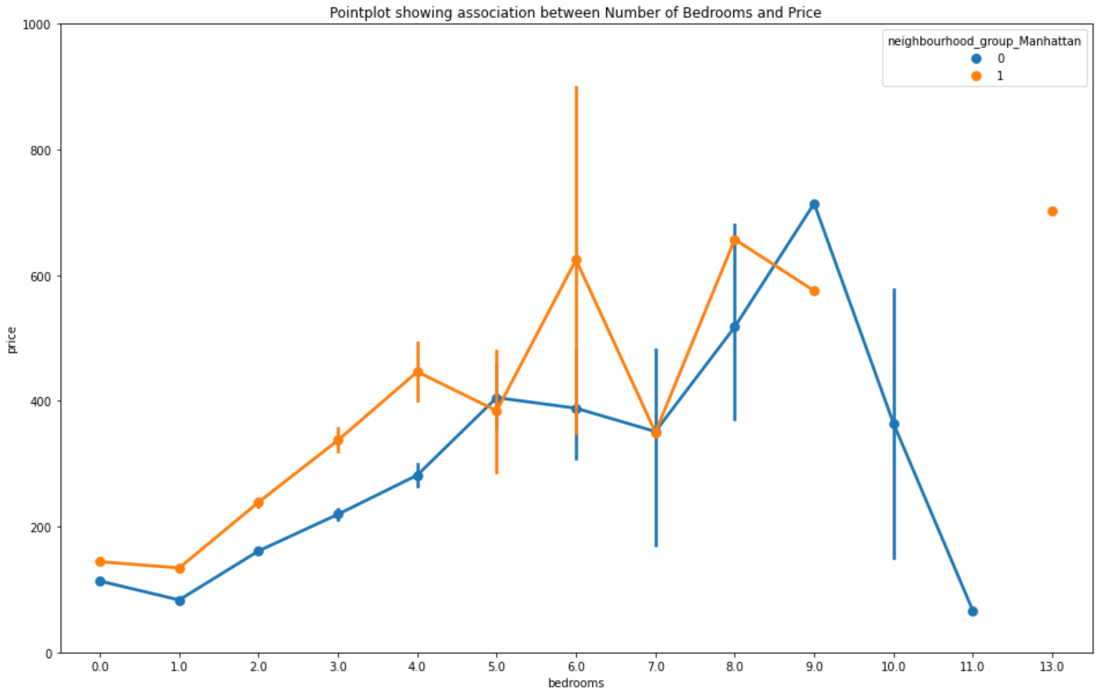


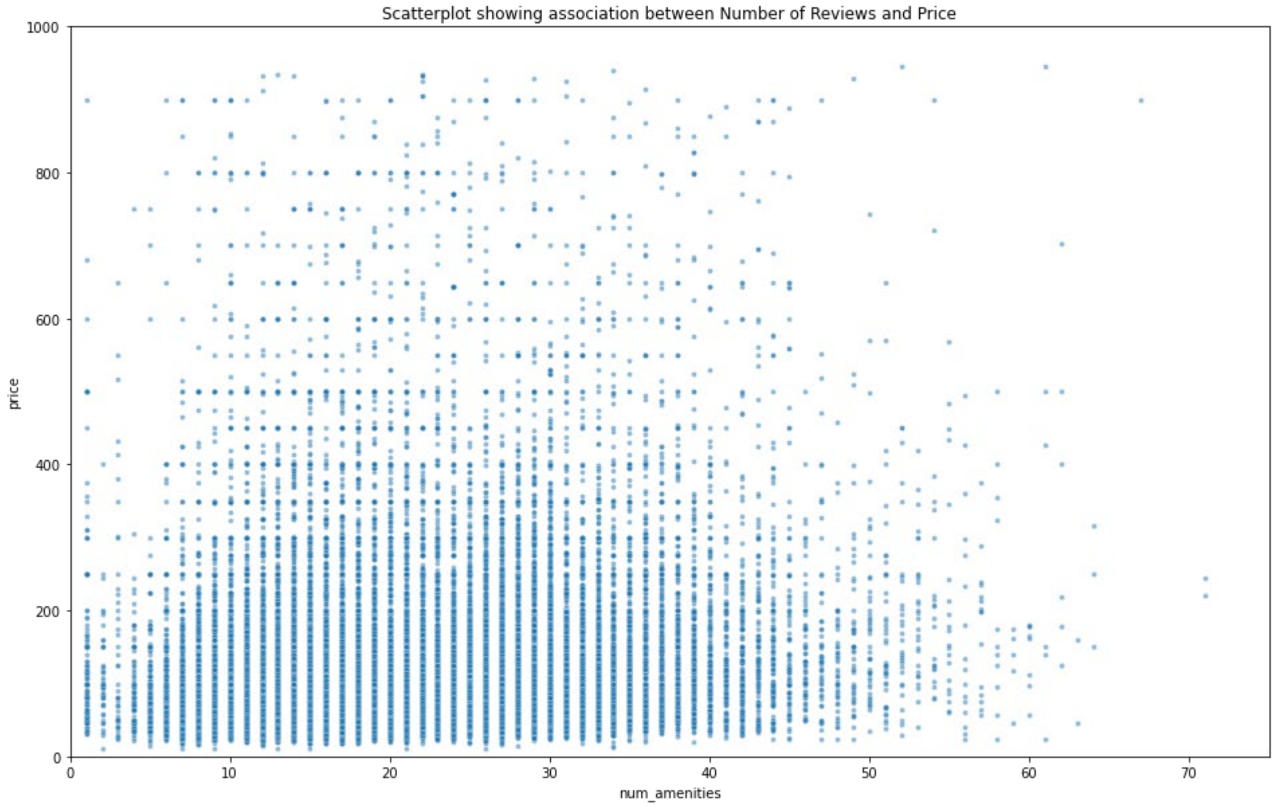
These are some plots to illustrate how some of the features are related to price.



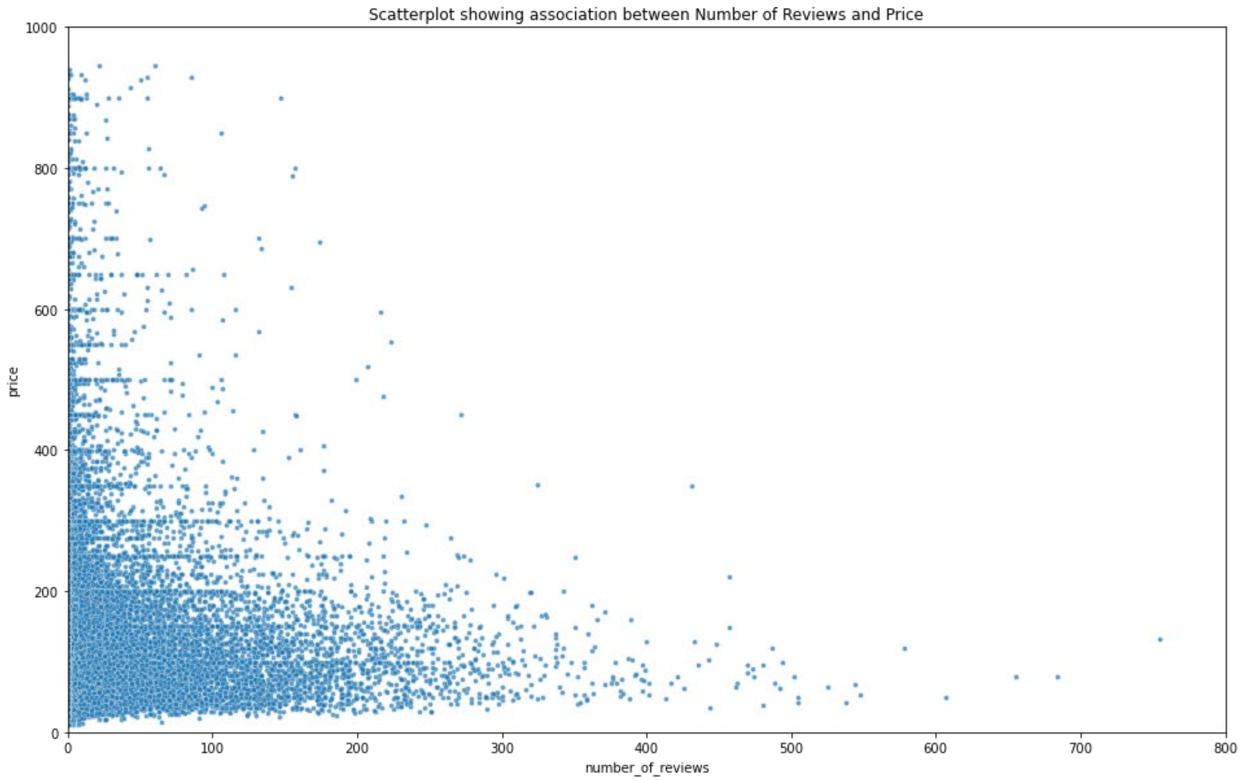
This point plot shows that average prices generally increase with number of accommodates and we can also notice that for Manhattan, this is higher than other places. When the accommodates increase above 10 or so, the price distribution varies highly and the gap between Manhattan and other places increases.

Point plot below shows that with number of bedrooms, average price increases, but when it goes above 5 rooms, the relationship shows an erratic pattern.

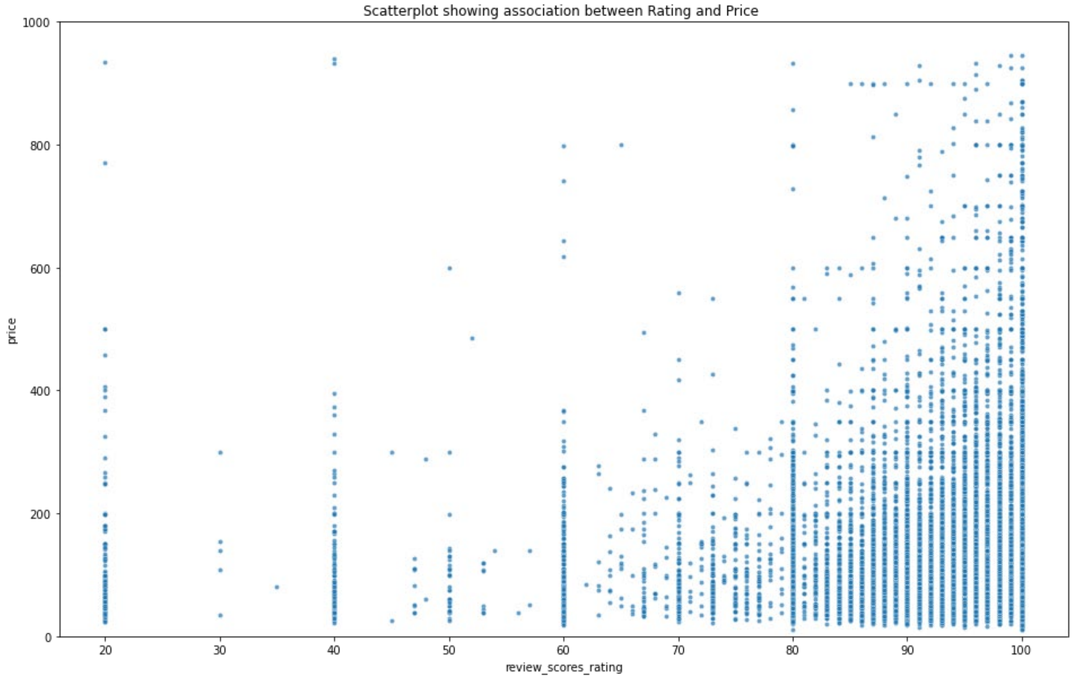




From the above scatterplot we can infer that there's no clear relation between number of amenities and price.

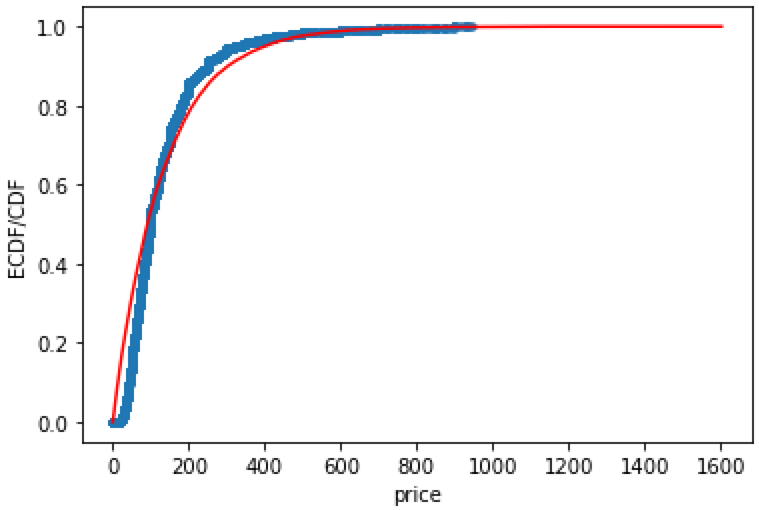


Here again, we cannot deduce any clear association between number of reviews and price, however it seems as though, highly priced listings have a smaller number of reviews. Low priced listings in general have higher number of reviews as the occupancy or demand for them will be higher.

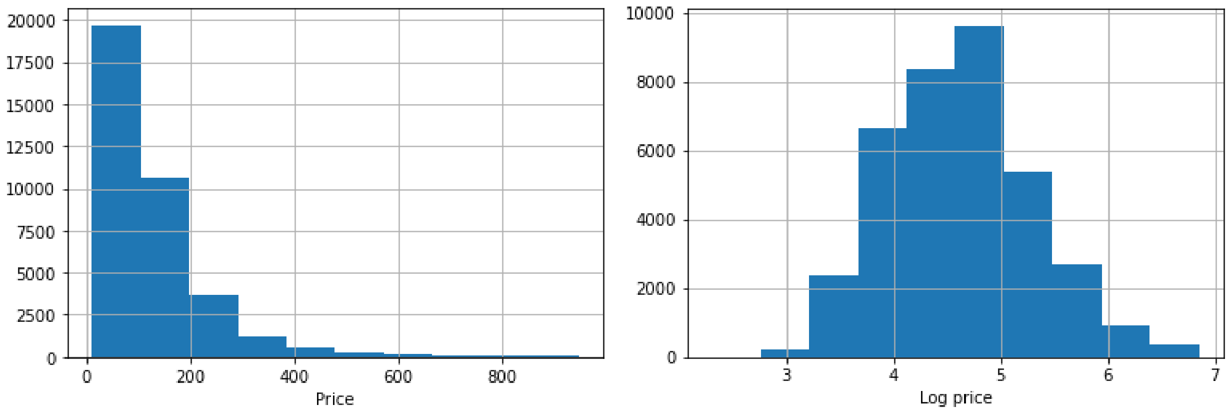


The above scatterplot gives a slight hint that in general, when ratings are higher, prices are somewhat higher too. But there are listings where the prices are lower with heir ratings in the upper range.

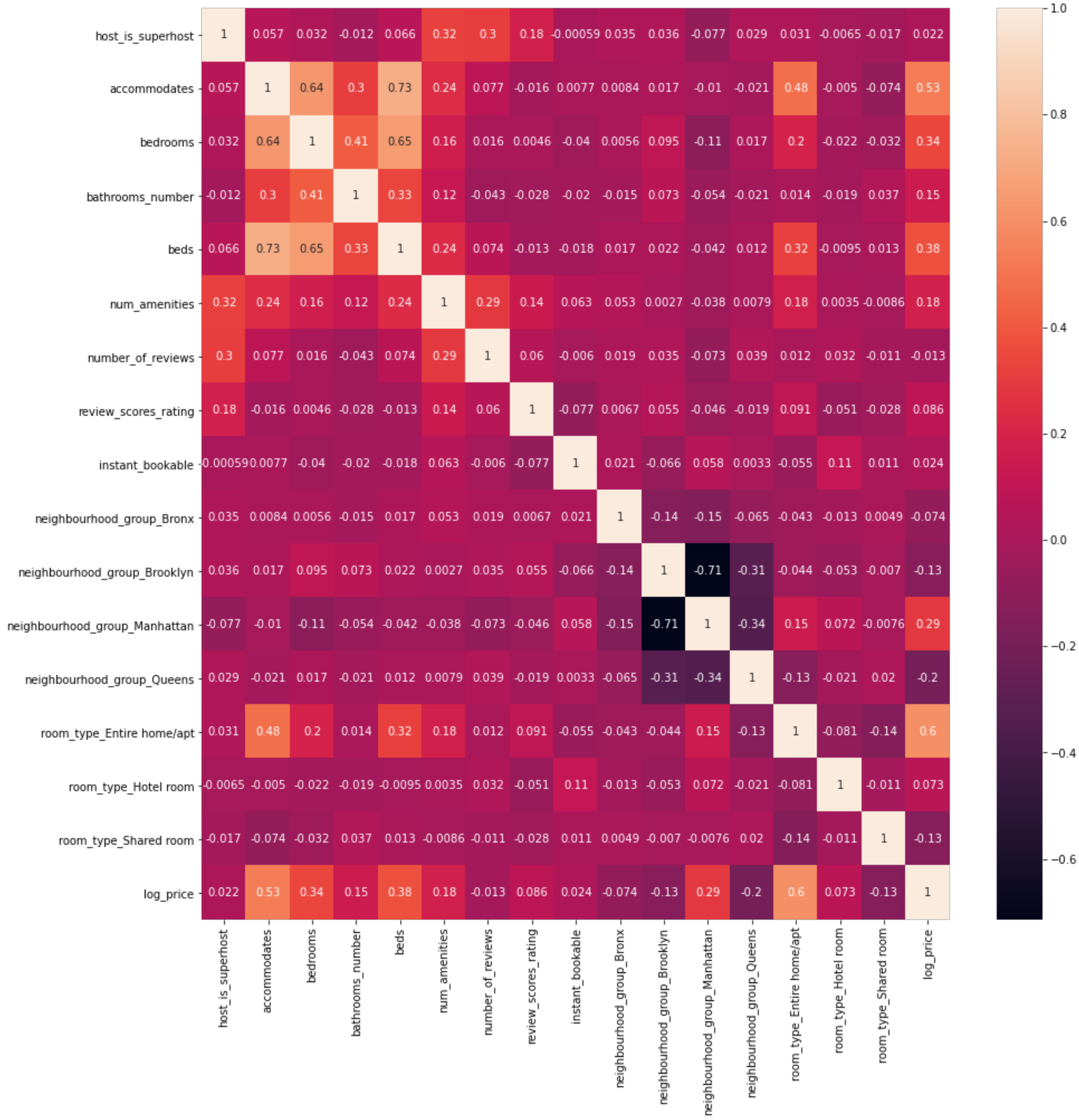
With target feature price, 35 listings had 0 prices, so it had to be removed. Similarly, prices after 99th percentile were also removed as it consisted outlier values like $2,000. After this step, the price ranged between $20 and $946. However, still the price showed an extremely skewed distribution with 75% of samples ranged between $20 - $156 and rest of the percentile ranged between $156 and $946. The ECDF of actual price data (in blue in below graph) showed an exponential curve and it lays in line with the theoretical ECDF (in red) of exponential curve. So, to rectify this to some extent, price was log-transformed for further analysis.



Price distribution before and after log transformation



Following heatmap which was produced using correlation between features shows the associations at a glance. Correlation between price and accommodates and price and room_type_Entire home/apt are above 0.5 and positive. Correlation between Private room type and price is above 0.5, but negative. Similarly, correlations between price and bedrooms, price and beds and price and neighborhood_Manhatton are positive and in the range of 0.3. Surprisingly, number of reviews or review_scores_rating did not show any significant correlation with price. Price and number of amenities is also correlated with a magnitude of 0.18.

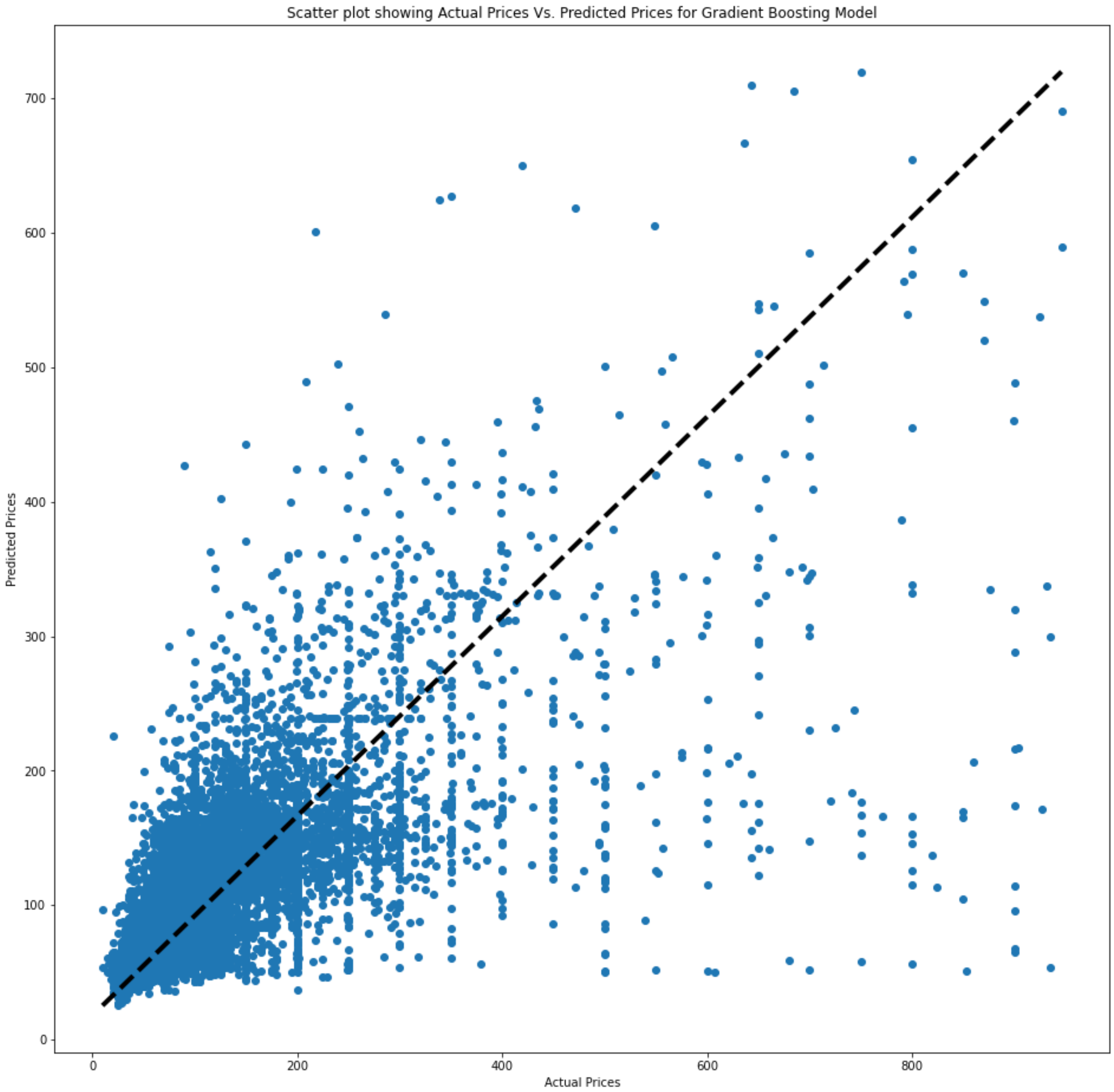


Modeling

Before modeling, preprocessing steps like splitting train and test data (test-size was 0.25) and standardization of numeric features were done. Different models with incremental complexity were used in predicting log prices. Mean Absolute Error (MAE) was used as the performance metric for each model. That is, model that produces the least MAE was chosen.

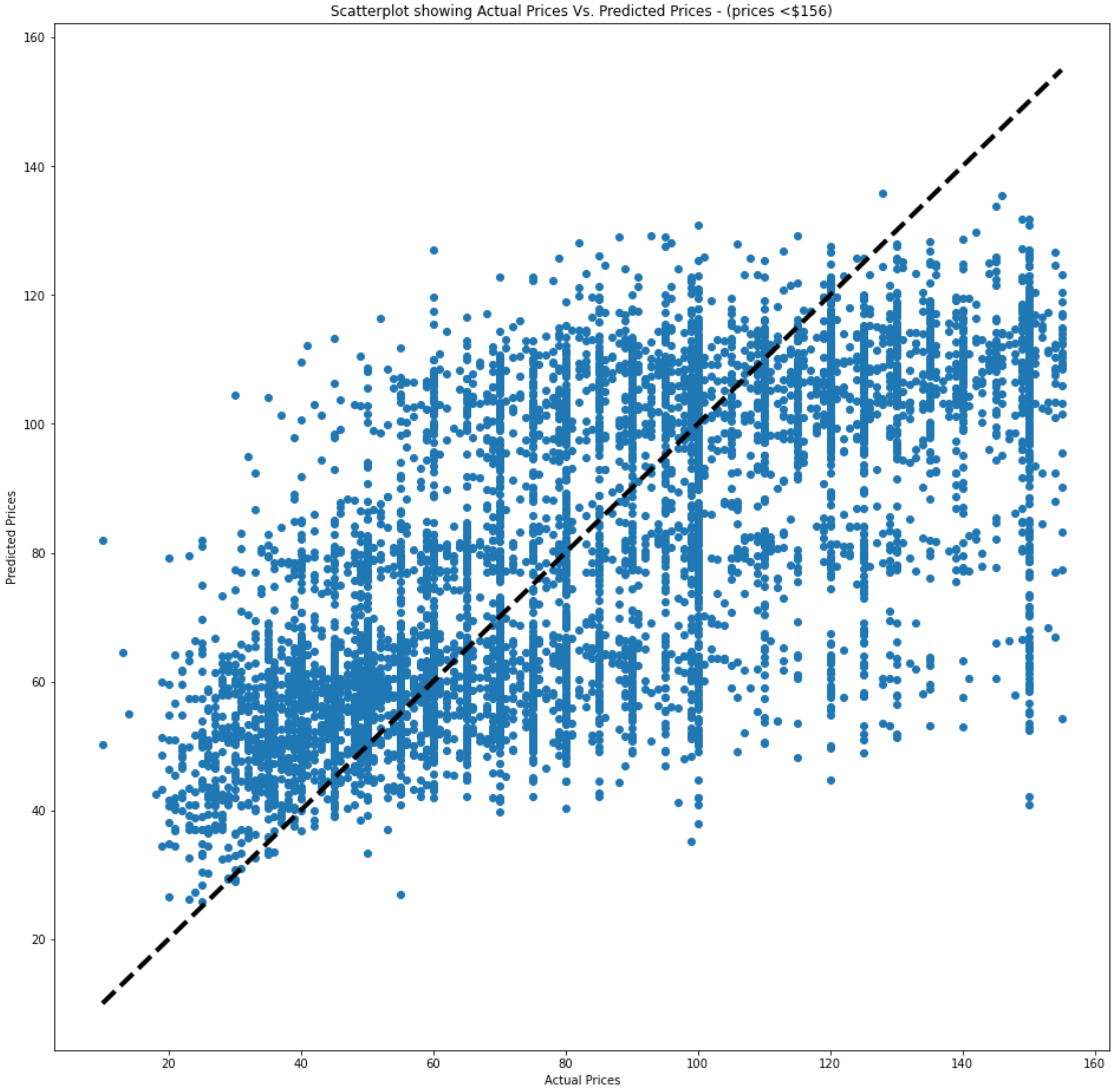
First a dummy regressor with median was used. I chose median instead of mean, as the distribution of log price is not even. This generated a MAE of $67.61. This means the prediction error on average is $67.61. Then a linear regression was used, which reduced the MAE to $48.27. R squared for the model was 0.53. In order to avoid overfitting and to reduce the size of MAE further, Lasso regression with Grid Search CV was used. MAE remained the same even after Lasso regression, so it didn't do any betterment for the model. Also, the optimal alpha obtained was small that it didn't make any significant change from the linear regression model. So next, I tried ensemble methods such as Random Forest (RF) and Gradient Boosting (GB). For decision tree models like RF and GB, feature standardization is not needed as the magnitude do not affect the outcome. RF was used with Randomized search CV. This helped in choosing optimum values for hyperparameters such as number of estimators, maximum features to be used, and maximum depth for the RF model. RF model further reduced the MAE to $43.26 now. This solution was obtained with number of estimators = 286 and a maximum depth = 10. And the best score was 0.57. Finally, GB model was used to check if it improves the prediction any further. With GB model, the MAE has reduced even more to $41.17. The GB model reached its solution with a smaller number of estimators (50) and maximum depth (7). Hence, it was decided that this model should be used to predict prices for test data/hold-out data set. For test data the MAE was 45.20.

In order to check how well this model predicts prices for test-data, I created a scatterplot of actual prices vs. predicted prices (see next page). The closer the blue dots are to the black diagonal-dotted line, the better will be the predictions. This diagram shows that for prices less than $400, the predictions are somewhat better than for prices higher than $400. That is for price range greater than 400, most of the predictions are lower than the actual values. This must have been caused by the unbalanced data set where there's 75% of prices are less than $156. One way to minimize this issue is to predict prices separately for different price ranges, for instance, different price model for less than $156 (price at 75th percentile), next one for prices in the range of $156 and $600 (price at 99th percentile) and another one for prices above $600.

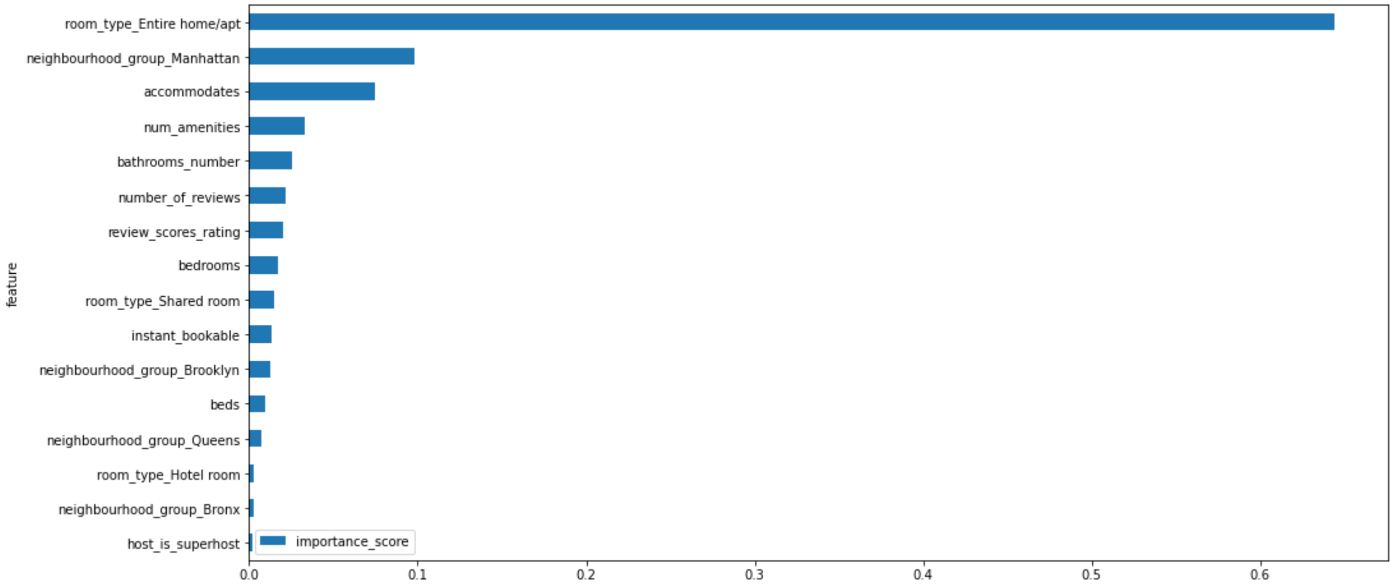


Model for Prices less than $156

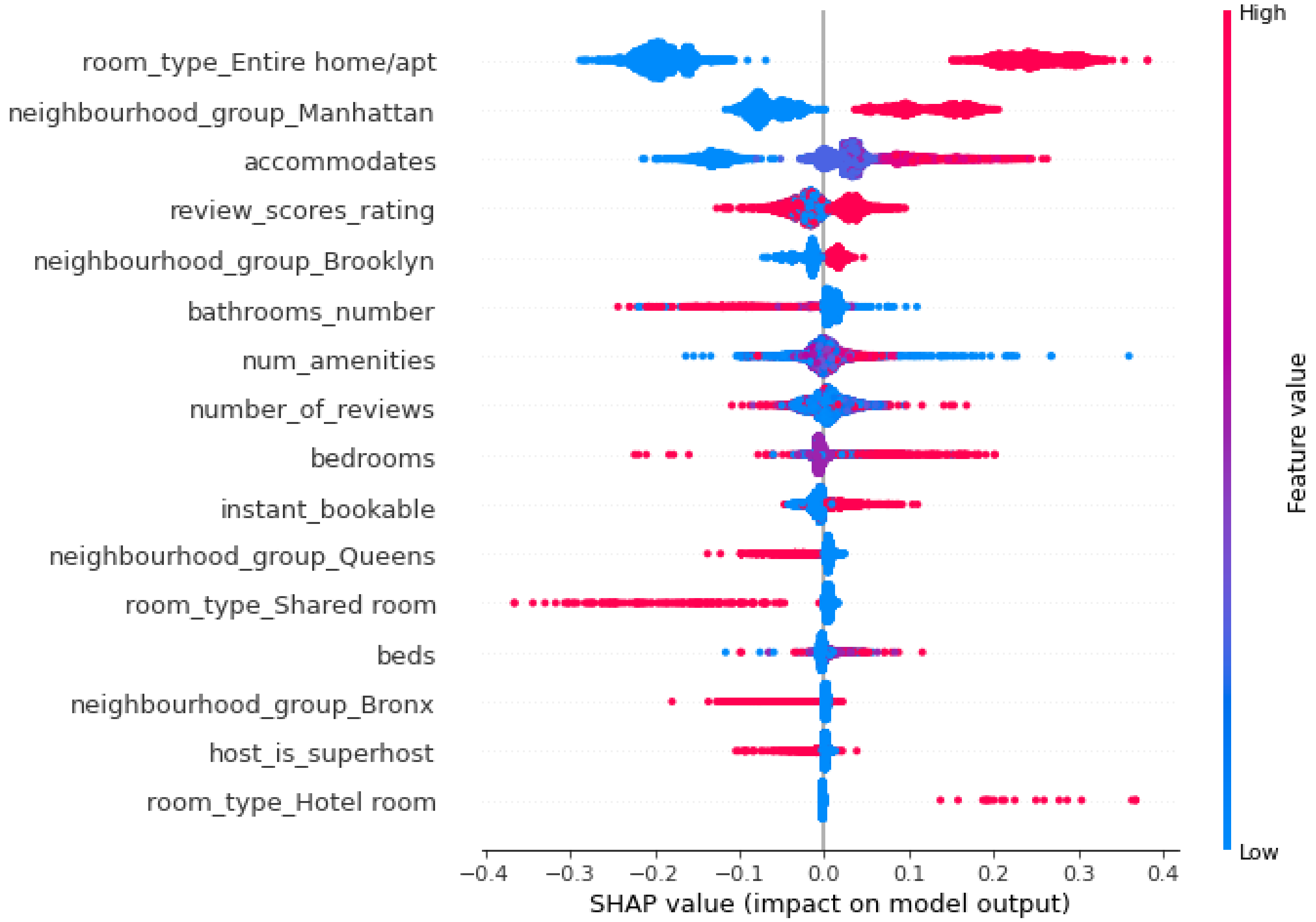
All the preprocessing steps were the same as earlier except for sub setting the data set for target prices less than $156. Train data MAE was 19.64 and test data MAE was 20.49. Solution was reached with estimators = 100 and max-depth = 5. Below is the scatterplot for comparing the actual vs. predicted prices.



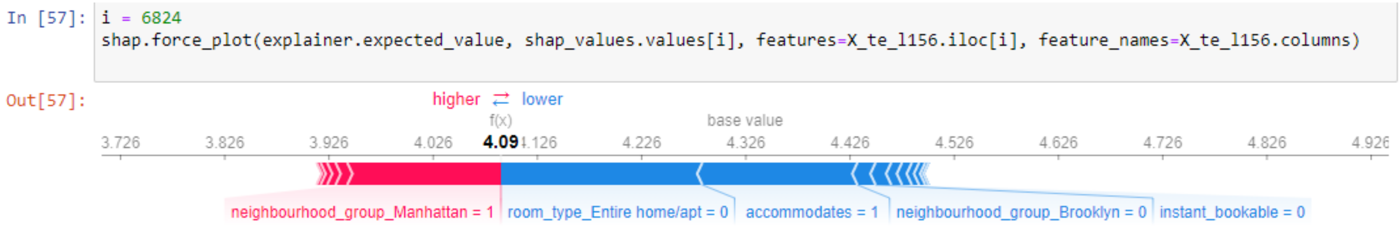
This time, the plot looks better than earlier. However, it seems like the prices in the lower ranges are better predicted than those on the upper ranges. A bar plot exhibits the feature importance for this model's performance on test-data.



This illustrates that the feature on the top are the key features in determining which leaf node a sample belongs to while modeling. Among all the features, room_type_Entire home/apt has been instrumental in segmenting the data points. The next important feature has been neighborhood group - Manhattan and number of accommodates. In order to understand further about how this model predicts different samples, I also conducted an error analysis where the prediction error, that is, actual value minus the predicted value was compared against all the features of the sample instances. Out of 6,838 samples, around 58% of them had absolute error less than $20 and 95% of them had absolute error less than $50. For better comprehension about how each feature instances contribute to its prediction, Shapley Additive Explanations (SHAP) were used. SHAP is a method for interpreting machine learning models' predictions through estimates called Shapley values, which is the mean marginal contribution of an instance of a feature among all possible combinations. The main goal is to estimate the Shapley values for each feature of the sample that needs to be interpreted. The calculated Shapley value denotes the impact that this feature, will have on the outcome or the prediction of the model (Lopez, 2021). Accordingly, method called summary_plot() from shap library offers details about feature importance and the impact of Shapley values on the prediction. Red means high feature values and blue means low feature values. So, for instance, if room_type_Entire home/apt has 1 instead of 0, it will increase the predicted price (because red values are on the right side), whereas for room_type_shared room if the value is 1 instead of 0, it will decrease the predicted price (because the red values are on the left side).



Force-plot from Shap additives provides a snapshot of which feature values have influenced the predicted values. For example, the force-plot below shows that the features in red color pushes the values towards higher log prices, whereas the blue feature values push for lower values and 4.09 will be the predicted log price for the sample.

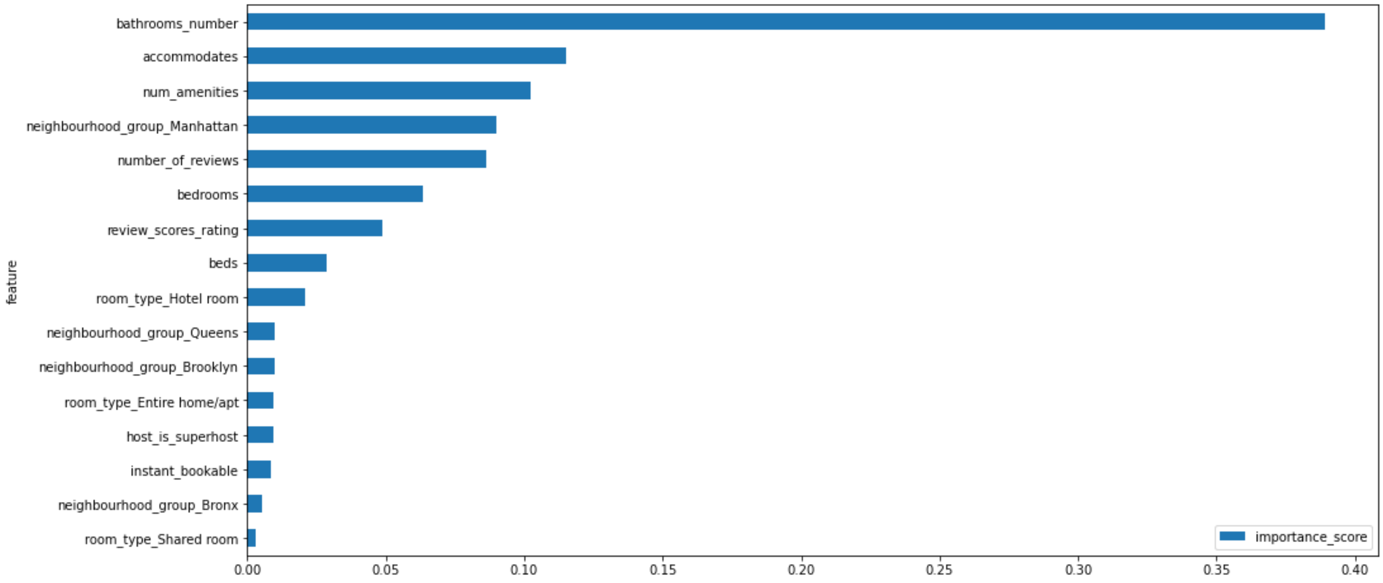


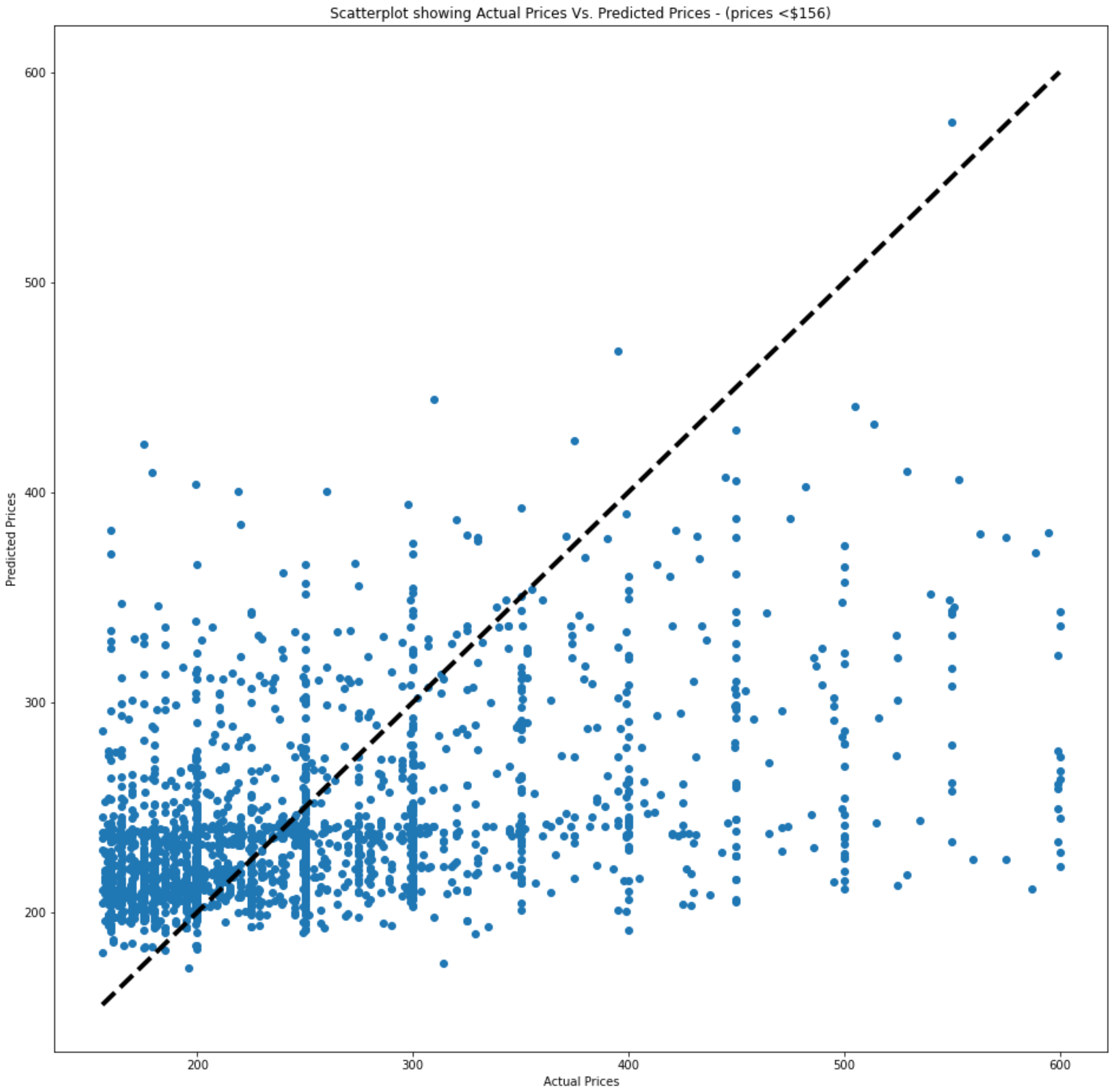
```
In [57]: i = 6824
         shap.force_plot(explainer.expected_value, shap_values.values[i], features=X_te_l156.iloc[i], feature_names=X_te_l156.columns)

Out[57]:
```

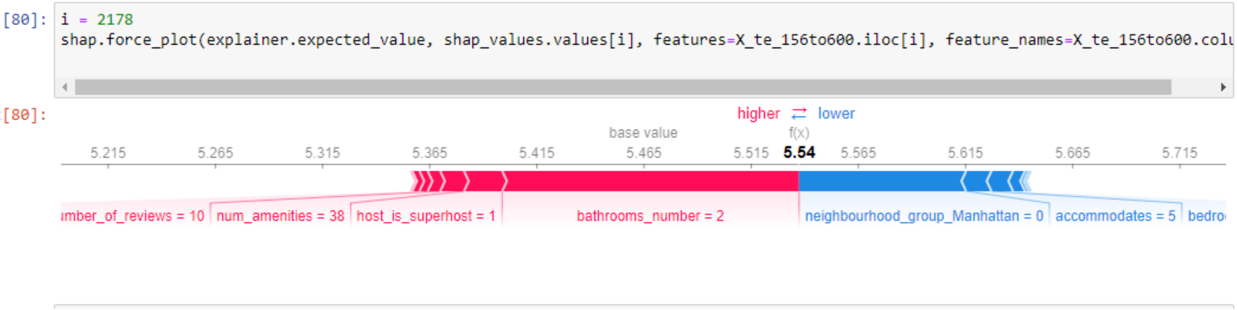
Modeling for Prices between $156 and $600

Data was sub-set for prices between $156 and $600. Model converged with an optimum number of hyperparameters; estimators = 50 and max-depth = 5. MAE for train data was $54.18 and MAE for test data was $57.45. The scatterplot showing the actual values vs. predicted values depicts that for prices in the lower range like $156 - $350 the predictions are better. However, for prices above, $350 or so, the predictions are very much lower than the actual values.



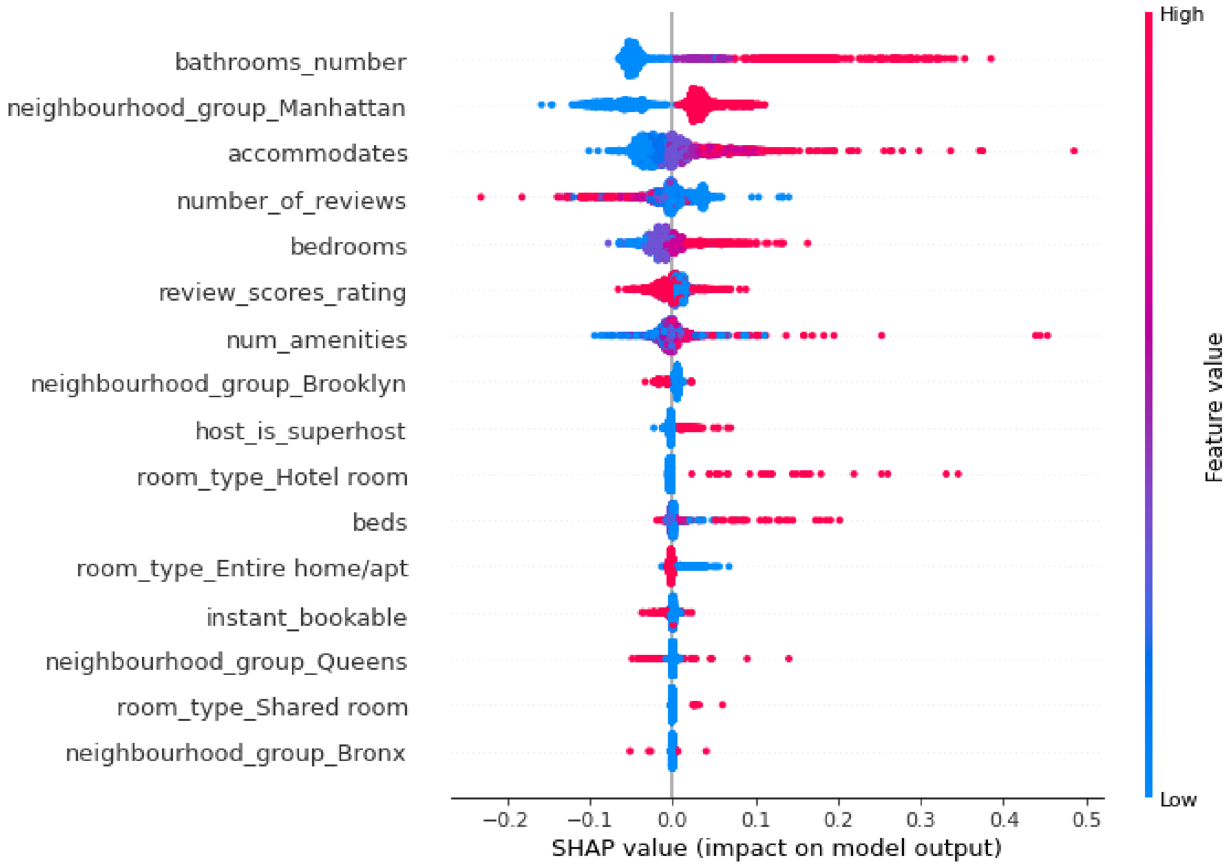


This could be because the samples are unbalanced, meaning proportionately there's way too many lower prices than higher prices. The feature importance bar chart revealed number of bathrooms to be the most important feature in segmenting the samples. This was followed by number of accommodates and number of amenities. Out of 2185 predictions, 11.4% of them have absolute error less than $10 and 61% of them had absolute error less than $50. However, 17 of them, most of their actual prices are close to $600, had absolute prediction error of $300. A shap summary plot offers information about the feature importances. Force plot depicts which features contributed towards arriving at the predicted value.



```
[80]: i = 2178
      shap.force_plot(explainer.expected_value, shap_values.values[i], features=X_te_156to600.iloc[i], feature_names=X_te_156to600.colu
```

```
[80]:
```



Limitations and Future Directions

Solution for the pricing model may be applicable only for listings in the New York area. The resulting model could be used as a rough estimate to set a price for their listings. This model may also be applicable only for the first quarter of the year as the data covers the early period of

April 2021. One important limitation of this study was the imbalance in data set, with large number of lower prices and small number of higher prices. Although this problem was minimized to some extent with log transformation, it still led the model to predict lower prices even for higher prices. This led to higher prediction error for higher prices. We can use algorithms like, SMOTER; which is a minority class oversampling technique modified for continuous target feature/regression to handle unbalanced data (Torgo et al., 2013). Another important issue with it is the lack of data on the demand side of the listings. Because, in addition to the features of listings, Airbnb pricing models also consider traditional demand aspects such as seasonal changes, local events, and actual demand for the place (Hill, 2015). One way to get data on this is to analyze the text-data on reviews of listings and find if the customers have mentioned anything specific about the location, for instance its proximity to subways or highways. Another method is to calculate how far the listings are from tourist attractions and historic places, as distance between landmarks and listings has been found to influence prices (Perez-Sanchez et al., 2018). Usually when people look for listings, the listings with high response-rate (measures how consistently the host responds within 24 hours to guest inquiries and booking requests) and higher acceptance rate (rate at which a host accepts booking requests) show up at the top in their web-search, so it's clear that these two features will have an impact on the price. However, these features couldn't be used in this model, as it had more than 50% missing values. Lot of new listings do not have any reviews or ratings, so this research couldn't find a better pattern between these features and listings' price. A feature that takes account of both, that is having higher rating with higher number of reviews may help the model. This project only counted the number of amenities that the listings offered, however it didn't differentiate between the type of amenities it offered. For instance, 'host greets you' and 'coffee maker' cannot be equally compared with each other, as the former will make a significant difference in the experience.

References

Gibbs, C., Guttentag, D., Gretzel, U., Yao, L. and Morton, J. (2018), "Use of dynamic pricing strategies by Airbnb hosts", International Journal of Contemporary Hospitality Management, Vol. 30 No. 1, pp. 2-20. https://doi.org/10.1108/IJCHM-09-2016-0540

Hill, D. (2015),"How much is your spare room worth?", IEEE Spectrum, Vol. 52 No. 9, pp. 32-58.

LearnAirbnb.com (2015), "Airbnb pricing strategy and tools", available at: http://learnairbnb.com/ airbnb-pricing-strategy-tool/ (accessed 3 September 2016).

Lopez, Fernando (2021). "SHAP: Shapley Additive Explanations". towards data science, 11 July, 2021. https://towardsdatascience.com/shap-shapley-additive-explanations-5a2a271ed9c3

Perez-Sanchez VR, Serrano-Estrada L, Marti P, Mora-Garcia R-T. The What, Where, and Why of Airbnb Price Determinants. Sustainability. 2018; 10(12):4596. https://doi.org/10.3390/su10124596

Torgo L., Ribeiro R.P., Pfahringer B., Branco P. (2013) SMOTE for Regression. In: Correia L., Reis L.P., Cascalho J. (eds) Progress in Artificial Intelligence. EPIA 2013. Lecture Notes in Computer Science, vol 8154. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40669-0_33.