IS5152 Data-Driven Decision Making
FINAL PROJECT

Analysis and Prediction of Airbnb Listings' Prices

26th April 2020

Done by Project Group 4

Joerck Andreas Brink	A0124458A
Kublikova Daria	A0209623H
Nandi Shuvam	A0212248M
Patel Toshal	A0198874A
Satija Parth	A0105203A

Table of Contents

1. Introduction

Airbnb is an online platform that enables people to rent out their homes for a short period and connects them with customers who are looking for an accommodation in that particular location [1]. The idea behind this business model is to provide travelers with more affordable accommodation while also giving an opportunity to homeowners to generate some revenue. Airbnb leaves the decision of pricing the properties on the hosts with a suggestion to have the price comparable to other listings in that neighborhood. But there are a lot of factors, such as number of bedrooms and amenities provided, that a host might need to consider before setting the value. At the same time, it is essential for hosts to price their properties competitively for rental especially in a city like New York where there are a large number of listings. As of April 2020, Airbnb has over 7 million listings worldwide spanning across more than 220 countries ascertaining that it is indeed a huge player in the tourism industry. Therefore, studying and analyzing Airbnb data is not only of great interest to researchers but also can provide immense economic benefits to the owners and the customers. Useful insights can be gained on how a property should be priced optimally in a free market that maximizes the satisfaction for both the parties.

Previously, there has been a body of work done trying to predict the listing prices. Li et. al [4] in 2016 used Multi-Scale Affinity Propagation clustering combined with Linear Regression model in each cluster to study how the listing distance from city landmarks and popularity of those landmarks can impact the price. The study was limited to geographical dataset and did not take into consideration the features unique to a listing. They suggested using additional data and more sophisticated models in addition to a simple multi-variable linear regression used in that experiment. Another research by Rezazadeh et.al [5] in 2019 studied limited features from the New York City Airbnb data using a range of models including support-vector regressions, decision trees and neural networks but did not include geospatial information in the research which proved to be an important dataset in the study done by Li et. al [4].

This study tries to build upon the existing literature by focusing not only on limited traditional features from Airbnb dataset but also enriching the data from other sources. In this novel approach the geospatial data is combined with data from Inside Airbnb. Thus, also correlating features like accessibility from the nearest subway station, eateries close to the accommodation and count of attractions within 200m with the price of the listing. In addition, we take into account seasonality by discretizing weather data while engineering data models which helps understand the shift in price across months. To the best of the knowledge, similar study with an extensive feature set has not been done until now in the current body of the literature pertaining to Airbnb price prediction.

This project aims to research on the New York City Airbnb data by exploring various decision making and machine learning techniques to best predict the price of a property given some input features. The dataset includes scrapped data from Airbnb [2] combined with geospatial data from OpenStreetMap [3]. Multiple linear regression is used as a baseline model followed by experiments done with more sophisticated models, namely Random Forests, XGBoost and Neural Network.

2. Dataset Preparation

2.1 Data Collection

2.1.1 Airbnb Dataset

The Airbnb dataset was downloaded from Inside Airbnb for New York City, USA. This dataset consisted of listings scraped within the time period of January 2019 to February 2020. For the purpose of this report, a “listing” is defined as an apartment or lodge which is put up by a person known as “host” on Airbnb for rentals and bookings to the public.

This dataset comprises 106 columns as feature attributes and 694,977 rows across the timeframe stipulated above. The dataset consists of prices for listings scraped across multiple dates in this time period, which may or may not vary across this time period. The attributes present in the original dataset are outlined below.

Attributes in the original dataset can be categorized as follows:

- Data Scraping: Date and IDs scrape;
- Property specifications: Written descriptions about the listing, no. of bedrooms, bathrooms, size, house rules, neighbourhood and borough information, list of amenities, property and room type, etc.;
- Host information: host description, if they are superhosts or not, number of listings across Airbnb, response and acceptance rates, time on Airbnb, etc.;
- Price information: price, cleaning fees, security deposits, charges for extra people;
- Stay Information: requirement to stay a minimum / maximum no. of nights, as well as availability for the coming period of time, cancellation policy;
- Summary of reviews: number of reviews and scores across categories.

2.1.2 Geospatial Data

New York City has the largest rapid transport system in the world based on the number of stations, which stands at 472. Subway stations are located in boroughs of Manhattan, Brooklyn, Queens and the Bronx. It also has a great bus stop connectivity with close to 16,000 bus stops across the city. Needless to say, being one of the major cities in the world, and apart from being one of the biggest financial hubs in the world, it is a city bustling full of energy, life and diversity. With a myriad number of attractions and restaurants across the city, a typical tourist is assumed to prefer to stay close to a place easily accessible and well connected to other parts of the city.

With this in mind, to further extract more insights into what makes a listing’s demand high and impact its price, its accessibility to public transport and proximity to convenience stores, malls and tourist attractions nearby was considered. The data set contained the latitude and longitude defining the geographical location for each of the listings. For each of the unique listings, the OpenStreetMap public API was used to extract counts of 5 location types present in 200m and 1000m radius of the location, giving us 10 new features to account for each listing. The below location types were searched for:

- Subway Stations;
- Bus Stops;
- Tourist Attractions;

- Restaurants;
- Convenience Stores/General Stores/Malls.

While extracting these features, it is important to note that it was assumed that the number of locations around the listing remained constant across the time period of this data, as it is not possible to search locations near a coordinate as of a given past date.

2.1.3 Weather Dataset

According to the British Retail Consortium, the state of the economy is the first influence consumer behavior, weather conditions have the second biggest influence[6]. Weather affects spending patterns and individuals tone as a consequence it influences decision making about travels. Initial assumption is that since average daily/monthly temperature affects demand it might also affect prices for listings.

National Center for Environmental info (NCEI) [7] - US organization that preserves, monitors, assesses historical weather data and provides public access to such data. Daily weather data for the period 2019-2020 in the central New York City Central Park area was requested from NCEI.

From the initial dataset obtained from NCEI the following features for each listings were calculated to further analysis:

- Average daily and monthly temperature;
- Maximum difference between temperature during day/month;
- The total amount of snow that fell in a month.

2.2 Data Validation

2.2.1 Handling Incomplete Data

From amongst the 106 features present in the original dataset, 72 relevant columns were identified for further analysis. All of the other columns were dropped due to being redundant or only meta-data specific columns, as follows:

- URLs: Host URL, Listing URL, Picture URL, etc.;
- Geographic Location attributes: Country, State, Street, Location Overview, Jurisdiction Names;
- Host Information: Verifications, Name, About, Interaction;
- Listing information: Experiences Offered, License Required, Space, Maximum Nights Average, House Rules, Area in sq. ft.

Data quality checks were conducted on the dataset to validate the input dataset and to filter out values which are of not much use, as a result of being null values. From amongst the 72 columns, 32 columns as below were found to contain null values across the dataset. This is detailed in the *Appendix Table 6.2.1.*

The **host_acceptance_rate**, **weekly_price** and **monthly_price columns** with 85% or more of the rows in the dataset containing null values. These columns were eventually eliminated as part of the data cleaning. Afterwards, to handle the incomplete data for other features, rows containing missing values were removed. After dropping these rows and columns, the number of rows was reduced from 694,977 to 244,905 and columns from 72 to 69.

2.2.2 Handling Outliers in Target Variable

In the exploratory data analysis, distribution plots for each variable were provided. The price variable (target variable) was found to be highly skewed, with a skewness of 17.6 and have extremely high outliers, as per the distribution plot and boxplot in *Figure 1.*

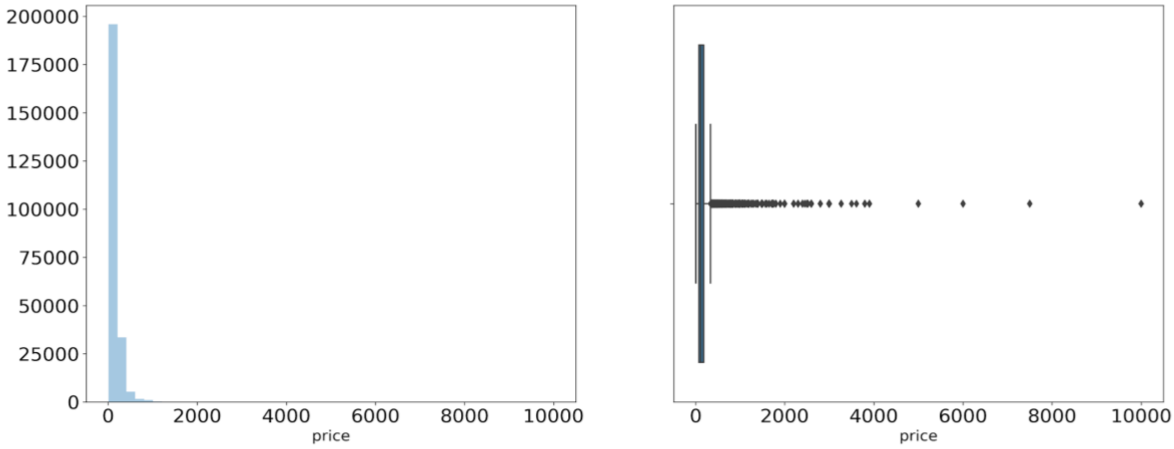


FIGURE 1.DISTRIBUTION OF PRICE

Two approaches were considered for transforming the price variable towards normality - Performing a logarithmic transformation and simply removing outliers. With regards to log-transforming; this approach makes it difficult to compare the predicted variables to the testset, due to issues arising when anti-logging the predicted values. The predicted values are anti-logged by exponentiating them, which causes the variance to be extremely high. Thus, this approach is not preferred, and we proceed with simply removing outliers.

Outliers are identified by calculating the interquartile range (IQR - range between 1st quartile, Q1, and third quartile, Q3), and any data points above Q3 + 3 * IQR or below Q1 - 3 * IQR is considered an outlier. By simply removing these, the skewness is reduced greatly as seen in *Figure 2*.

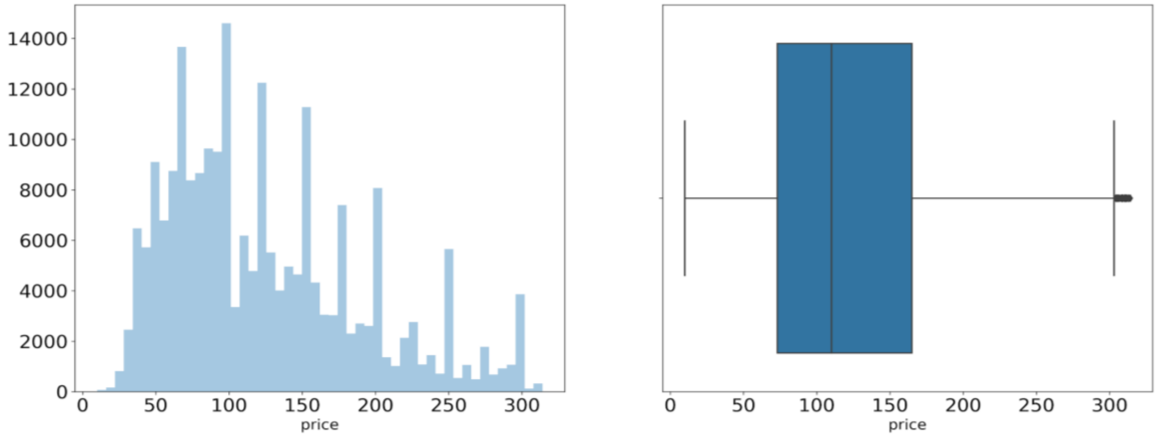


FIGURE 2. DISTRIBUTION OF PRICE AFTER REMOVING THE OUTLIERS

The factor of 3 was decided over the common factor of 1.5 [8], since this variable has many extreme outliers, and the vast majority of the observations are centered around the mean.

The changes of the price variable can be seen in *Table 1*.

Price variable outlier adjustment	Number of observations	Mean	Skewness
Before	244,785	152.09	20.09
After	229,526	123.45	0.87

Table 1. Before and after adjustment of the price variable

This will be used as the target variable for the prediction models.

2.3 Data Exploration and Transformation

2.3.1 Price Distribution

On studying the price distribution, it was found that around 94% of the Airbnb listings are priced below $350 per night. As discussed in the previous section, a considerable number of outliers with very high prices were removed that were skewing the distribution. The processed data was then plotted geographically to get an insight on the relationship between location and price.

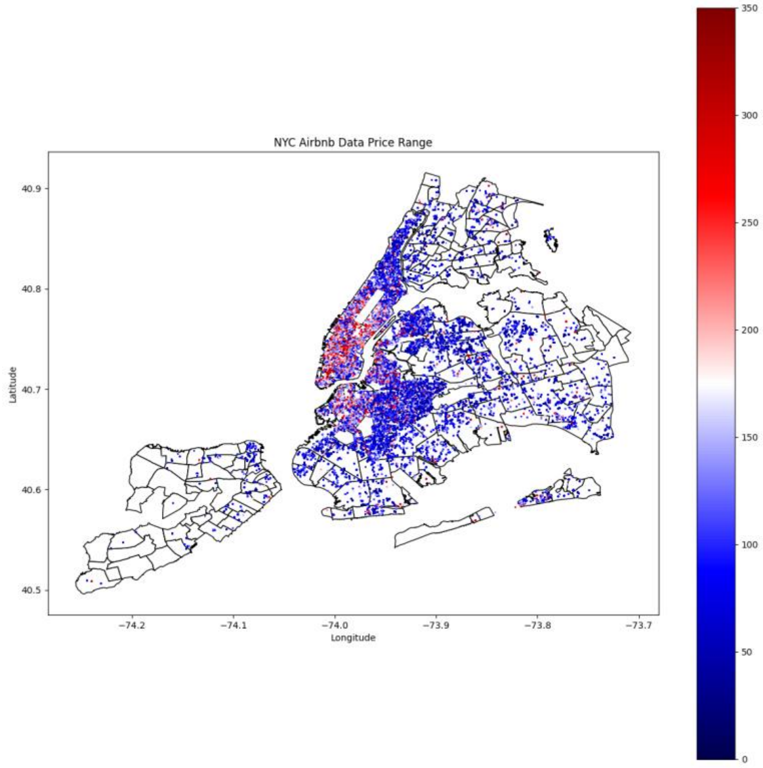


Figure 3. Evaluation Metrics of Feed Forward Network

It was observed *(Figure 3)* that some areas in the city (colored in shades of red) are generally more expensive than the others. The listings in Manhattan and Brooklyn are more expensive which are considered more popular and have many tourist attractions.

This further proves that including geospatial data in the analysis is imperative for a more accurate prediction.

2.3.2 Neighbourhood Data

To further improve the understanding of the relationship between price and location, the neighbourhood data was analysed. There were 3 columns defining a listing's location, 'neighbourhood', 'neighbourhood_cleansed' and 'neighbourhood_group_cleansed'. The 'neighbourhood_group_cleansed' column consists of 5 major boroughs of New York City, which are Brooklyn, Manhattan, Queens, Bronx, and Staten Island.

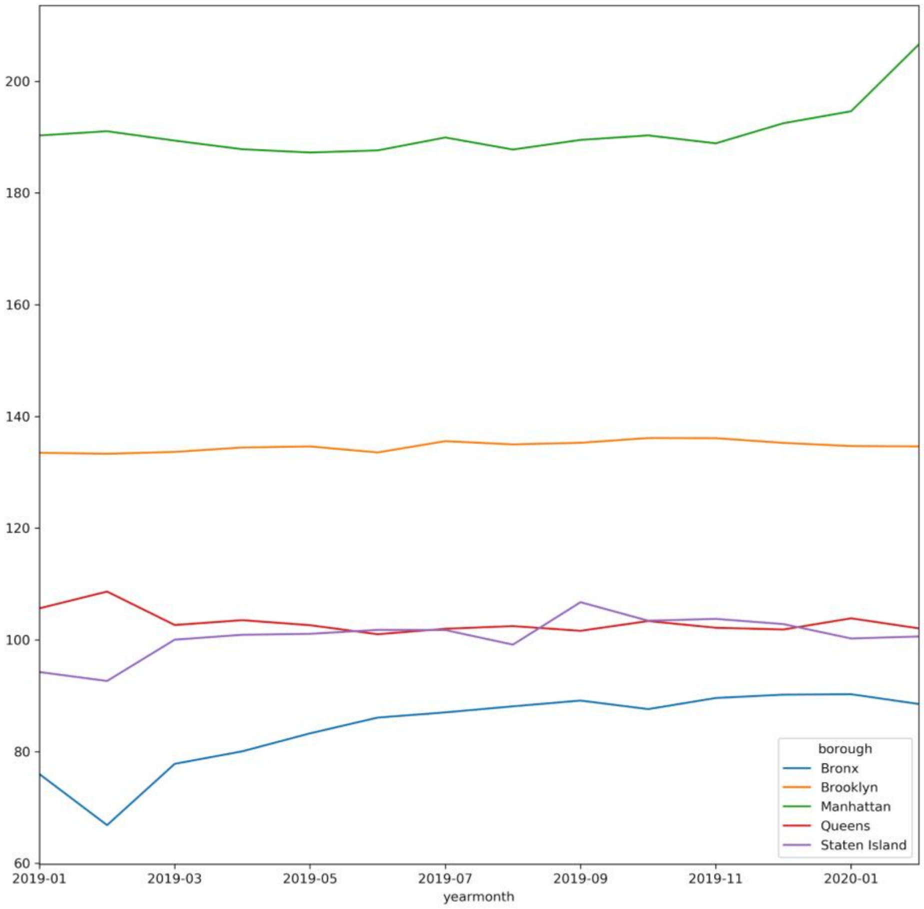


FIGURE 4. AVERAGE PRICE TRENDS ACROSS DIFFERENT BOROUGH

Figure 4 shows the average price trend across these 5 neighbourhoods over the period of 1 year. It is clearly observed that Manhattan is the most expensive area and there are slight shifts in the average prices across different months.

The 'neighbourhood_cleansed' column is a cleaned variant of the attribute 'neighbourhood', having 217 smaller city neighbourhoods' as its possible values. Only the 'neighbourhood_cleansed' and 'neighbourhood_group_cleansed' were retained in the dataset.

2.3.3 Host Specific Feature Extraction

Meaningful features were extracted from existing ones in a few cases. The difference of days between the date a host joined Airbnb ('host_since' attribute) and the last time scraping was done for their listing (‘last_scraped’) were used to define a ‘host experience’ feature describing their Airbnb membership. So, if a host joined Airbnb on 1st January 2019, and a listing was scraped on 30th December 2019, they would have had 364 days of experience.

A few custom defined functions were created to transform values in the dataset to appropriate data types for further analysis. This included converting ‘t’ and ‘f’ values (0/1 binary) to True and False (or None, if neither). Also, numerical values in string type were mapped to their integer equivalent values. Such data cleaning was performed on certain descriptors for a host, like ‘host_response_time’, ‘host_response_rate’, ‘host_acceptance_rate’, ‘host_is_superhost’, ‘host_total_listings_count’, ‘is_host_verified’.

2.3.4 Property Specific Feature Extraction

There were a number of physical attributes defining a property. These primarily include:

- Property Details
- Amenities
- Prices
- Availability / Stay
- Reviews

Each property has a fixed property type, room type, bathrooms, bedrooms, number of beds, and bed type. For all listings, the amenities available in it were represented in the dataset by a single string having multiple amenities available concatenated together. Example of one listing’s amenities are as below:

'{TV,"Cable TV", Internet, Wifi, "Wheelchair accessible", Kitchen, "Free parking on premises", Elevator, "Free street parking", "Buzzer/wireless intercom", Heating, "Suitable for events", Washer, Dryer, "Smoke detector", "Carbon monoxide detector", "First aid kit"}'

From this representation, a list representation from this string was extracted, and for all listings in the dataset, the 6 most commonly found amenities were found to be WiFi, heating, smoke detector, kitchen, air conditioning, and essentials (which is assumed to include the essential kitchenware, bed linen and pillows, toiletries, etc.). The percentage of listings which have these features are listed in *Table 2*.

Amenity	**wifi**	**heating**	**essentials**	**smoke detector**	**kitchen**	**air conditioning**
Count	99.01%	97.03%	96.46%	91.43%	91.18%	89.60%

TABLE 2. MOST COMMON AMENITIES ACROSS LISTINGS

These amenities were treated as rather important for any listing, hence 6 new boolean features were introduced in the dataset specifying whether each of these amenity is available in a given listing or not, e.g. 'is_wifi', is_kitchen, 'is_wifi', ‘is_heating’, ‘is_smoke_detector’, 'is_aircon'.

Each Airbnb listing's final price can be broken down into a few smaller components. These include Security Deposit, Cleaning Fee, and Charge for Extra People, all present as part of the dataset. On a total of 4 types of prices (including Total Price), cleaning was done to convert the string values with the "$" symbol into numerical float-type values.

Under the availability categories of data attributes, there were numerical and boolean type values representing stay criteria like minimum / maximum nights stay required, availability in the next 30, 60, 90 and 365 days, instantly bookable, etc. These were converted to Boolean values from 't'/'f' type of values. There was also a categorical column for cancellation policy for a listing, with values falling in six unique categories.

Data was available for each listing in the overall set showcasing the total number of reviews, reviews per month, and review scores across multiple categories such as overall rating, listing's information accuracy, cleanliness, check-in procedure, communication, location and venue. Majority of the reviews given to listings/hosts by consumers had a high score, therefore not adding much value to the data. Hence, **review scores are not an important feature as they do not provide great information**. A summary of distribution of review scores among different categories is shown in *Appendix Table 6.2.2.*

2.3.5 Seasonality

Preliminary analysis of price changes during the year shows that prices for the same apartments vary during the year. Hosts do not change prices daily - more than 85% of listings in the dataset change price no more than once a month. To understand what affects prices most - average monthly temperatures were compared with prices for the period 2019-2020:

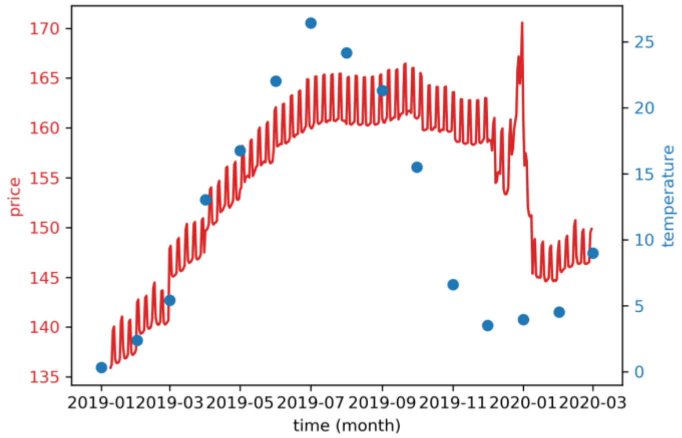


FIGURE 5. AVERAGE MONTHLY TEMPERATURE AND PRICE TRENDS

Observations from *Figure 5:*

- Curve for average monthly temperature has a similar shape as the price curve, but has lag from September to January. Beside temperature consideration it is worth considering seasons;
- The Price curve has a pick at the end of December/beginning of January, most likely it's caused by Christmas & New Year Holidays.

Therefore, few more features were added to initial dataset:

- Seasons based on weather in New York city, and the month of the 'last_scrapped' value (date of the scraping of the data) - December to February as Winter, March to May as Spring, June to August as Summer, and September to November as Autumn;
- Holidays variable - shows how many holidays are in the coming 4 weeks for 'last_scrape' date. In terms to calculate this amount of coming holidays, all 2019-2020 United States holidays were withdrawn from the holidays python library.

2.3.6 Data Encoding and Discretization

The weather data that is taken into account in the analysis, requires discretization as the data scraped is for a single month, and temperature and weather being a volatile variable, changes throughout the month. Hence, based on the month in which the data was scraped, and the temperature, the weather data was discretized into different seasons - Winter, Spring, Summer, Autumn.

The categorical data variables like borough, cancellation_policy, host_response_rate, room_type, and seasons were converted into dummy variables, which follow one-hot encoding. This proved to be a necessary step in handling the categorical data and understanding the importance of these features in predicting the target variable, i.e., price.

With all the above data cleaning, reduction and transformation, the eventual dataset resulted in **244,905 rows** and **78 columns** (77 features + 1 target variable, i.e., price) in our final dataset. Out of these, there were 35,021 unique listings.

3. Analysis and Modeling

The following section will describe all the modelling steps as well as reasoning for choosing specific models and evaluation metrics. With the goal of predicting the continuous variable 'price', it was decided to use a multiple linear regression model as a baseline for the prediction in order to evaluate the performance of other prediction models.

3.1 Models

This project considers four different types of models to predict the continuous target variable, which are:

- Multiple Linear Regression;
- Random forest;
- XGBoost;
- Neural Network.

Since, the input data is of higher dimension, taking into consideration various features that might be important for prediction of the prices of the Airbnb listings, the selected models seem to work well with higher dimensions of data, despite long training times.

3.2 Evaluation metrics for regression

In order to evaluate the performance of each model, the following metrics will be provided for each model:

- Mean Squared Error (MSE): Squared differences between predicted and actual values, referred to as error;
- Root Mean Squared Error (RMSE): Difference between predicted and actual values in the target variable's units, in this case dollars;
- Relative Root Mean Squared Error (RRSME): Shows the relative difference between the prediction and the mean of the true values, resulting in a percentage deviation from the actual values;
- Coefficient of Determination, R-squared (R2): Metric to show how well a model predicts the actual values, interpreted as the percentage of the total variance of the data explained by the independent variables.

Note, that the Mean Average Error (MAE) has been left out, and the study has maintained a focus on RMSE instead, since it was previously identified that there are potential high outliers. Failure to predict the remaining outliers will be very evident in the value RMSE, which penalizes high deviations by squaring the errors.

Furthermore, all models, including the baseline, will be fitted on the same training set, and tested on the same test set. It was decided to use 80% of the data for training and validation, while 20% is used for the final testing. The training-set is further split in 80% actual training data and 20% validation-set, thus the final training-set includes approximately 64% of the original data, while the validation-set constitutes approximately 16%.

3.3 Baseline for models

Two approaches for estimating a baseline for the prediction models were considered: always predicting a simple average and a multiple linear regression. These were both implemented and tested using the evaluation metrics above, which will later be used for comparing the other models to these as a baseline.

First of all, the mean of the target variable in the training set was calculated and used as the predicted value for all data points in the test set, with the results mentioned in *Table 3*.

Mean baseline	MSE	RMSE	RRMSE	R2
Test set	4,314	65.68	52.62	0.00

TABLE 3. EVALUATION OF MEAN BASELINE

Note, that there are no parameters to tune, thus the validation set has not been used in this case. This is far off, and will thus not be used.

Secondly, a multiple linear regression was used to form a better baseline for the prediction models. A simple linear regression including all possible independent variables was made, with the results shown in *Table 4*.

LinReg baseline	MSE	RMSE	RRMSE	R2
Train set	1,682	41.02	32.91	0.61
Validation set	1,686	41.07	32.81	0.61

TABLE 4. EVALUATION OF MULTIPLE LINEAR REGRESSION AS BASELINE

The multiple linear regression generalizes very consistently on the validation set, which is to be expected since the data is both uniform and balanced, and shuffled during the split into training, validation and test sets.

In order to lower the errors, a wrapper method with backward elimination using p-values to check for significance was applied. All features with a p-value > 0.05 are automatically removed, simply to see whether we can reduce the errors. This approach left 62 statistically significant features at the 5% confidence level, with the evaluation shown in *Table 5*. The feature importance is shown in *Appendix Table 6.2.3.*

LinReg baseline	MSE	RMSE	RRMSE	R2
Train set	1.683	41.02	32.91	0.61
Validation set	1.686	41.06	32.81	0.61
Test set	1.675	40.93	32.79	0.61

TABLE 5. EVALUATION OF LINEAR REGRESSION PERFORMED WITH ONLY SIGNIFICANT FEATURES FOR BASELINE

Interestingly, neither the errors dropped or the R2 changed by removing insignificant features, however these will be included in the other models, since they all use different approaches to best predict the price.

The results above will be used as the baseline for the prediction models, with an R-squared of 0.61 and consistent errors across the three datasets.

3.4 Support Vector Regression

Support Vector Regression (SVR) gives the flexibility to define how much error is acceptable in the model and finds the appropriate hyperplane to fit the data. SVR minimizes the coefficients, specially the L2 norm of the coefficient vector and not the squared error. It is tolerant of errors with an acceptable error margin, called *epsilon.*

The major drawback of the SVR was found to be the training time which was more than 12 hours, making the hyperparameter tuning extremely difficult. The results of the standard model on training and test data are shown in *Table 6*.

SVR	MSE	RMSE	RRMSE	R2
Train set	147.7593	12.1556	9.7388	0.966
Test set	2774.874	52.6771	42.2778	0.3606

TABLE 6. EVALUATION OF SUPPORT VECTOR REGRESSION

The difference in error in the training and test data is found to be very high and the R-squared value on the standard model was worse than the baseline. Hence, it was concluded that this model is not an effective model in terms of time invested, and was not explored further.

3.5 Random forest

A random forest takes a number of parameters, which can be tuned to optimize performance and reduce over-/underfitting the model on the training data. These parameters can be tuned specifically for this dataset using Python packages from the Sklearn library, namely RandomizedSearchCV and GridSearchCV. Both conduct cross-validation by leaving a subset out of the training data for cross validation, and the main difference between the two functions is the means of selecting the value of the parameters to be tested. Both methods were tested on a subsample of the full dataset, with very small deviations in the result, thus due to the size of the data, a randomized-search was used, since it reduces the runtime significantly and yields very similar results, as long as the iterations are reasonably high.

Specifically, the following random forest regressor parameters were tested:

- max_features: Number of features considered for splitting a node. A high number of features will fit the training data very accurately, but generalize poorly resulting in overfitting;
- max_depth: Number of splits / layers in each tree. Deeper trees will fit training data more accurately, however also increases chance of overfitting, thus a compromise is sought;
- n_estimators: Number of trees to be generated. More trees lowers the variance of the generalization error, however the bias of the model will remain and it takes significantly longer to train the model with a high number of trees;
- min_samples_leaf: Determines minimum number of samples per leaf. With a high number of samples per leaf (or percentage of total sample), the leaves will likely not be pure, but on the other hand this parameter can be used to reduce overfitting by ensuring a certain size of each leaf node.

Note, the folds for cross-validation was set to 3 in all cases [9,10].

During the tuning process, the training set achieved almost 100% prediction accuracy, however the model was overfitting heavily on the validation set, thus the min_samples_leaf parameter was added to the list of parameters to tune to account for this.

Finally, the tuned parameters were used to train a new random forest regression model in order to assess the performance on the testset and finally be compared to the other models. The results are shown in *Table 7.* A slight overfit is noticed on the training set, since the errors increase on the validation and test set, and the R2 decreased. Despite several attempts on the validation set, it was not possible to achieve a better fit.

Random Forest	MSE	RMSE	RRMSE	R2
Train set	83.97	9.16	7.35	0.98
Validation set	192.20	13.86	11.08	0.96
Test set	204.03	14.28	11.44	0.95

TABLE 7. EVALUATION OF RANDOM FORESTS

3.6 XGBoost

XGBoost is an optimized distributed gradient boosting algorithm that is designed to be highly efficient, flexible and portable. It provides a parallel tree boosting (also known as GBDT, GBM) that predicts the target variable in a fast and accurate way. Unlike Random Forest, XGBoost creates weak learners where the tree depth is lesser than Random Forest and the model aims to reduce both bias and variance with every iteration. It was found to be a lot faster than the Random Forest Algorithm.

Randomized Search was used to decide upon the best parameters from the given set of parameters. It was seen that with a learning rate of 0.075, tree depth of 15, and 400 estimators for the model proved to reduce the error metrics and increase the R-squared value. It was also seen that the increase in number of estimators paired with decrease in depth gave a better performance. Using the best set of parameters, the errors of XGBoost reduce by a considerable amount as compared to Random Forests Algorithm, also the R-squared value increased. The evaluation of the XGBoost model is shown in *Table 8*. Feature importance was one of the other important deliverables from XGBoost, as shown in *Figure 6*.

XGBoost	MSE	RMSE	RRMSE	R2
Train set	1.9803	1.4072	1.1269	0.9995
Validation set	124.4541	11.1559	8.9317	0.9714
Test set	126.4095	11.2432	9.0452	0.9707

TABLE 8. EVALUATION OF XGBOOST

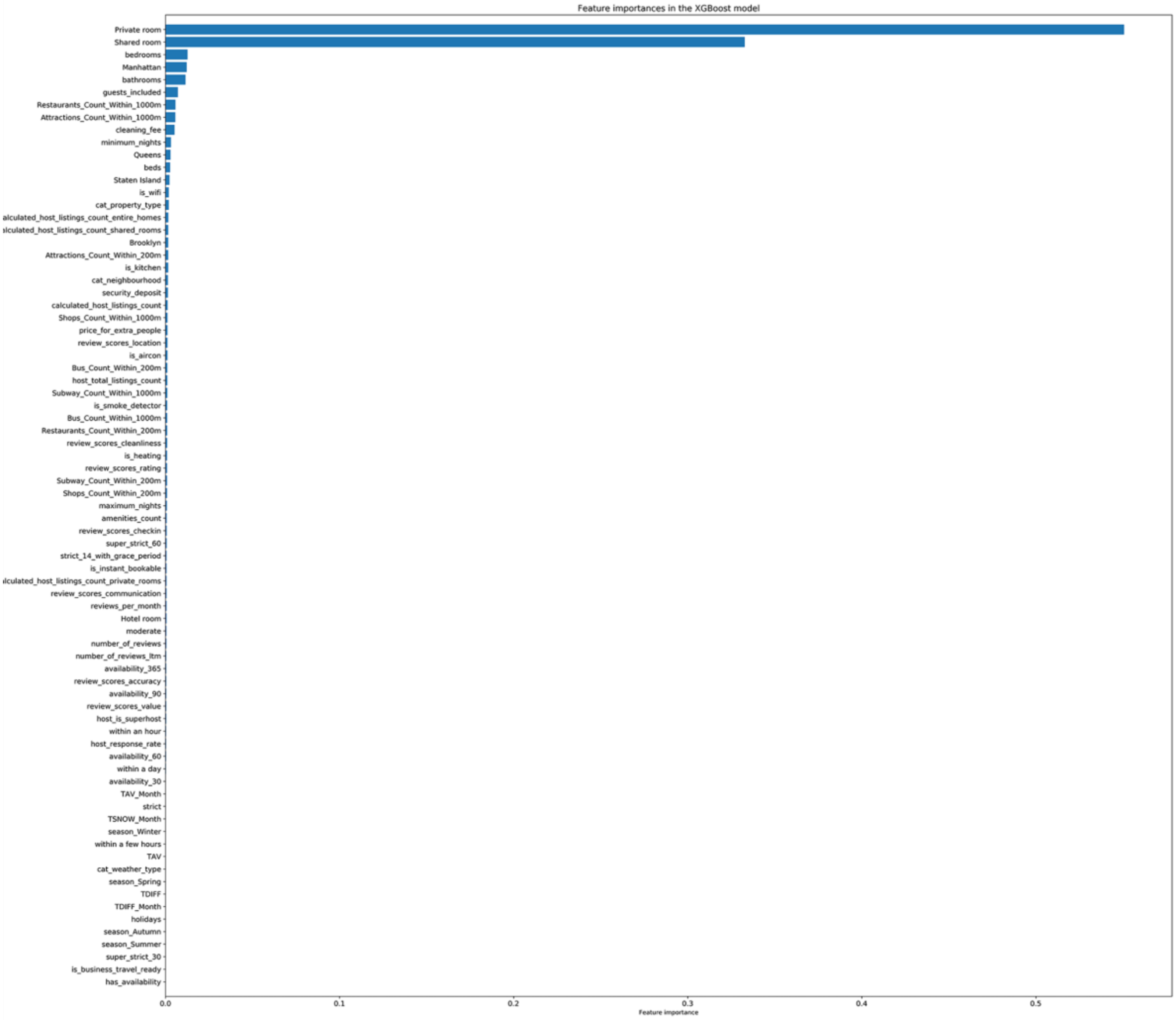


FIGURE 6. FEATURE IMPORTANCE FROM XGBOOST

Furthermore, embedded feature selection and importance from XGBoost was studied. From *Figure 6*, it can be seen that the top two features, 'Private room' and 'Shared room', highly dominate the feature importance graph. The other important features include the number of bedrooms, Manhattan location, if the guests are allowed and geospatial features. As expected, all these features are essential in the prediction of the prices.

3.7 Neural Networks

Neural networks is an excellent model to tap into the higher dimensional features for regression, classification, etc. It considers the relationship between all the features in determining the value of the target variable. A feed forward network is layers of fully connected neurons, that take input features as a matrix of numbers, with dimensions as - (number of examples) x (number of features). The data included 62 significant features derived from the Baseline Linear Regression model mentioned in the above sections. These features are used for predicting the target variable, i.e., price of the listing.

A simple feed forward neural network was used for addressing the regression task of predicting the prices of the Airbnb listings. As shown in *Figure 7* fully connected layers were used with ReLU activation function after every layer. Mean Squared Error was used as the loss function and Adam optimizer for optimization.

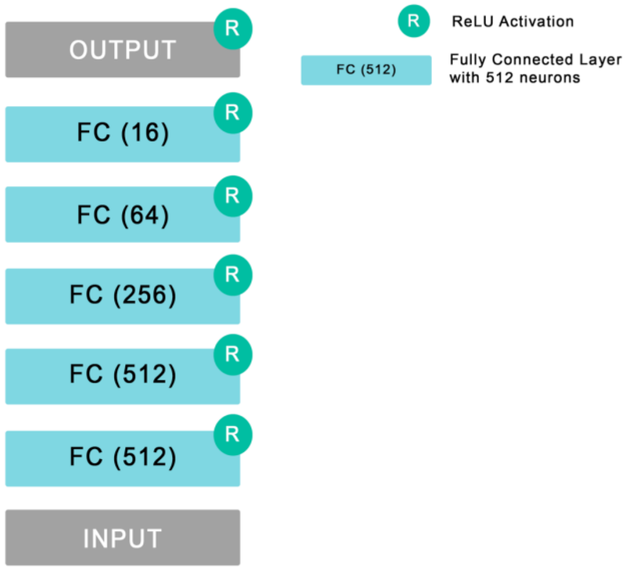
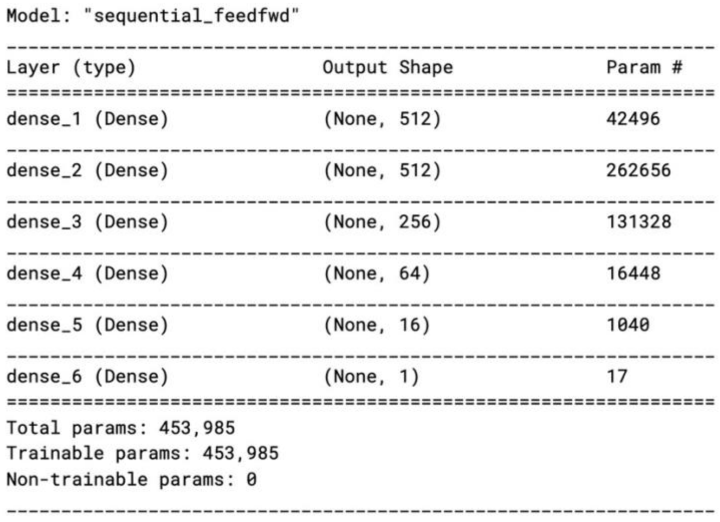


FIGURE 7. NEURAL NETWORK ARCHITECTURE

The hyperparameters like the number of hidden layers, the number of units in the hidden layers, learning rate, batch size and number of epochs were tuned to get the best performance. It was inferred that with more number of hidden layers and units in each hidden layer, the network was able to give a higher R-squared value with lower errors. The initial architecture was with 3 hidden layers and was increased to 5 hidden layers. This reflected a major improvement in the error reduction. The final sequential network that was implemented in *Figure 7* can be summarized as follows in *Figure 8*:



```
Model: "sequential_feedfwd"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 512)               42496
_________________________________________________________________
dense_2 (Dense)              (None, 512)               262656
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_4 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_5 (Dense)              (None, 16)                1040
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 17
=================================================================
Total params: 453,985
Trainable params: 453,985
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 8. THE FINAL SEQUENTIAL NETWORK THAT WAS IMPLEMENTED IN FIGURE 1. CAN BE SUMMARIZED AS FOLLOWS

Finally, the network was trained with 500 epochs and a batch of 256. The validation loss after each epoch, was used as the metric to save and select the best model and predict the price values in the test data. The model evaluation on the test data is given in *Table 9*.

Feed Forward Network	MSE	RMSE	RRMSE	R2
Train set	68.9908	8.3061	6.6473	0.9841
Validation set[1]	159.7477	12.6391	-	-
Test set	72.9212	8.5394	6.884	0.9831

TABLE 9. EVALUATION METRICS OF FEED FORWARD NETWORK

The difference between the error metrics of training data and test data, the values of the error metrics appear to be lower as compared to the models implemented above. It can be concluded that the performance of this feed forward network is the best among the other models implemented above.

The training loss vs validation loss is reported in *Figure 9*. It can be seen that the training and validation loss have some difference, which is an indication that the model is neither overfitted nor underfitted, and network architecture is appropriated for good performance. The correlation between the predicted and actual values for training and test data is shown in *Figure 10*.

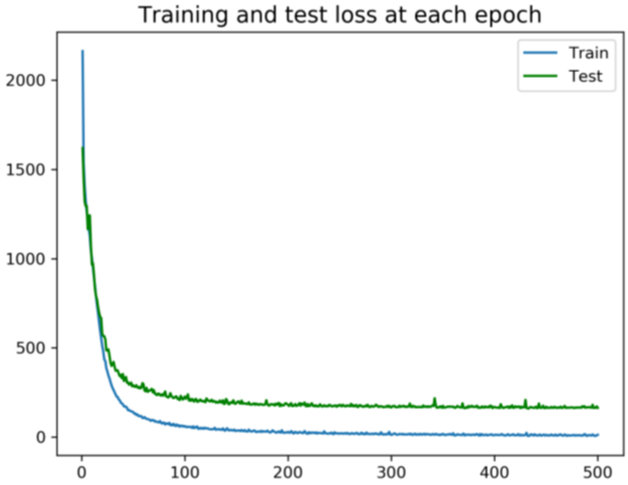


FIGURE 9. COMPARING TRAINING AND VALIDATION LOSS AT EACH EPOCH

[1] The validation loss was calculated after each training epoch, and the evaluation of the best epoch is reported.

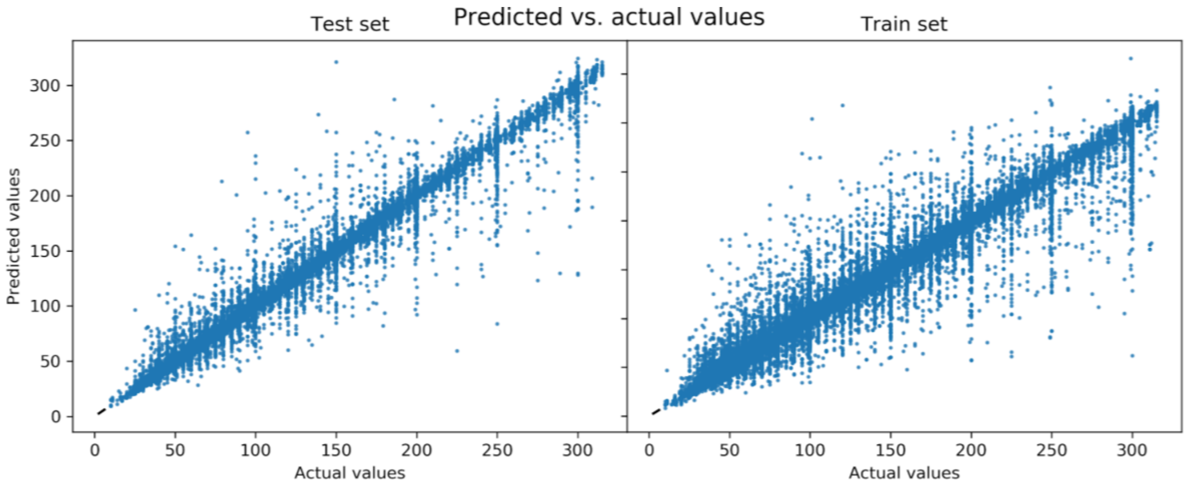


FIGURE 10. COMPARING PREDICTED AND ACTUAL VALUES IN TRAIN AND TEST DATASET

4. Discussion and Summary

Performing the preliminary analysis provided insight on how the various features affect the pricing of the listings in Airbnb. It is an interesting analysis that justifies the logical explanation of how the various features affect the pricing of listings. Starting with the variation of prices in different neighborhoods or boroughs, gave an idea that Manhattan and Brooklyn are the most expensive areas, with the former being significantly higher in price than the remainder.

Moving on to the effect of season on price of the listings throughout the year, it was seen that during the holiday seasons, the prices seem to be higher, but during the Autumn, from July to November, the prices seem to be higher for sometime and then slope down as the temperature drops. Higher prices are found in the seasons in the order Spring, Winter, Summer and Autumn. Hence, this reflects that Autumn is the time when the prices are higher, which may indicate that it is the peak tourism that began in Spring, ends with the onset of Winter, but rose as the Christmas and New Year holidays approached.

Seasonality and neighborhood logically explained the reasons of variation in prices, and were justified by the feature importance by the baseline Multiple Linear Regression model. The other significant features are explained as follows:

- is_superhost indicates that the hosts which are company verified "Super" hosts (i.e., are reviewed highly by the users) have higher impact on the prices of their listings;
- property features - number of bedrooms and bathrooms, property_type like Private room or Shared room, and various amenities like aircon, wifi, etc. have a higher weight in determining the variation in pricing;
- accessibility - the features subway_within_200m, attractions_within_200m and attractions_within_1000m stand out, and contribute to the listing being of higher price if they have a subway within 200m radius and attraction point within 200m and 1000m radius.

To regress the price variable, Multiple Linear Regression model was used as the baseline model, using the significant features, giving a R-squared value of 0.61 on test data with an RRMSE of 32.79. Next, Support Vector Regression was found to take a very large time to train the model and performed worse than baseline on default parameter values. Hence, a decision was made to not explore the model further to save time.

Moving on, Random Forest gave a higher R-squared value of 0.95 on the test data and reduced the RRMSE by 65.11% from the baseline metrics. To implement a faster decision tree model, XGBoost was the next choice. The R-squared value improved to 0.97 on the test data, *further* reducing the error by 21.73%. Neural networks were the next choice to leverage the higher dimensionality of the data and extract the most of the significant features. The feed forward network *further reduced* the error by 37.68%, giving an R-squared value of 0.98 on the test data. The network also reduced the difference between the training and test errors, and gave the best performance for the regression of the target variable. Finally the model has R-squared as 0.98, with errors of US$6.884. This is a close enough estimate for any new New York City Airbnb host to competitively price their property.

For a new host joining the Airbnb platform, given the features of the property like number of bedrooms, baths, whether it is an apartment or house, shared room or private room, the location and accessibility of the property, etc., using the best model proposed in the study, an apt price for the listing can be derived. Having selected more sophisticated models like Random Forest, XGBoost and Neural Network to predict the prices of the Airbnb listings and combining the insights derived from the seasonality and location, a good prediction accuracy was achieved. Thus, helping homeowners to find an appropriate price for their properties. It should be noted that the market variation and the changes in real estate prices, interest rate, etc., have not been considered in this study, and it is entirely based on the data provided by Airbnb and considering the type and features of the property.

This study considers the panel data of all listings over the period of one year which limits deeper understanding of variables and features affecting the prices over a larger time frame. Another limitation is that the frequency of the data collected is once every month. More granular data captured every day would help in pricing a property dynamically. Hence, an improved understanding of variation in prices over time accounting for the holidays, change in season, and a few hidden factors can be obtained. For example, with the current data, the holidays and seasonality clearly affect the variation of price throughout the year, but the monthly data collected does not allow us to analyze this deeper.

This experiment explores the prices of New York City for the year 2019 and partially for the year 2020. For future research, it can be expanded to understand the price change over time by conducting a time-series analysis based on various features in the data. Also, it can be exciting to research into the performance of these models on the listings of various cities in the same state in America or across states in the country. Having a comparative analysis of variation of prices across different cities and countries can be of great value.

5. References

[1] Airbnb, Inc., "Airbnb," *Airbnb, Inc.*, [Online]. Available: https://www.Airbnb.com/. [Accessed 10 February 2020].

[2] "Inside Airbnb", [Online]. Available: http://insideAirbnb.com/. [Accessed 20 February 2020].

[3] "OpenStreetMap", [Online]. Available: https://www.openstreetmap.org/. [Accessed 26 April 2020].

[4] Y. Li, Q. Pan, T. Yang, and L. Guo, "Reasonable price recommendation on Airbnb using Multi-Scale clustering," *2016. 35th Chinese Control Conference (CCC)*, 2016. [Accessed 26 April 2020].

[5] Kalehbasti, P. Rezazadeh, Nikolenko, Liubov, Rezaei, and Hoormazd, "Airbnb Price Prediction Using Machine Learning and Sentiment Analysis," *arXiv.org*, 29-Jul-2019. [Online]. Available: https://arxiv.org/abs/1907.12665. [Accessed 26 April 2020].

[6] "BRC," *BRC*. [Online]. Available: https://www.brc.org.uk/. [Accessed 26 April 2020].

[7] "National Centers for Environmental Information," *National Climatic Data Center*. [Online]. Available: https://www.ncdc.noaa.gov/. [Accessed 26 April 2020].

[8] J. Brownlee, "How to Use Statistics to Identify Outliers in Data," *Machine Learning Mastery*, 08-Aug-2019, [Online]. Available: https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/. [Accessed 26 April 2020].

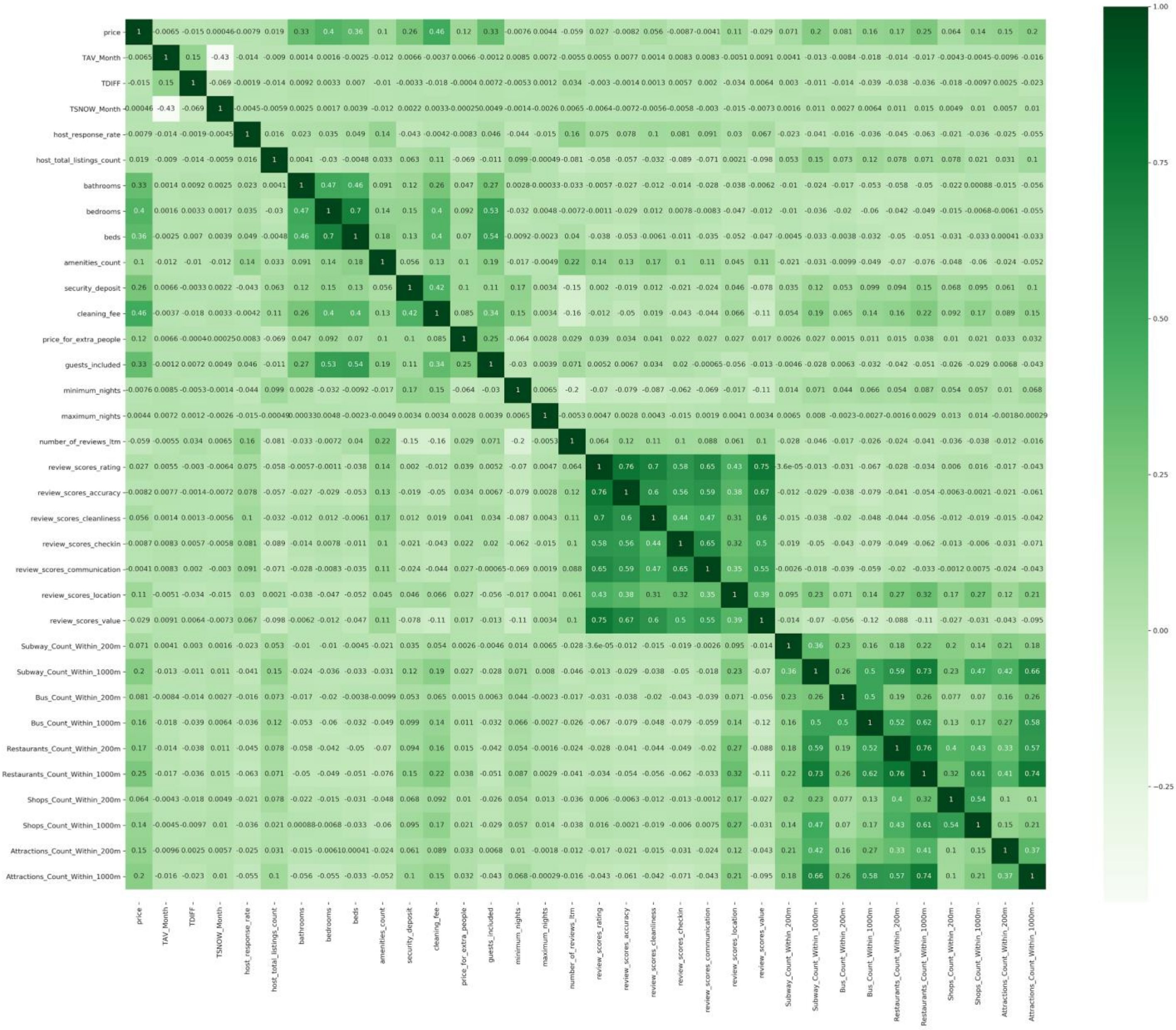
[9] M. B. Fraj, "In Depth: Parameter tuning for Random Forest," *Medium*, 21-Dec-2017. [Online]. Available: https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d. [Accessed 26 April 2020].

[10] W. Koehrsen, "Hyperparameter Tuning the Random Forest in Python," *Medium*, 10-Jan-2018. [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74. [Accessed 26 April 2020].

6. Appendix

6.1 Appendix Figures

6.1.1 Correlation Matrix of features



	price	TAV_Month	TDIFF	TSNOW_Month	host_response_rate	host_total_listings_count	bathrooms	bedrooms	beds	amenities_count	security_deposit	cleaning_fee	price_for_extra_people	guests_included	minimum_nights	maximum_nights	number_of_reviews_ltm
price	1	-0.0065	-0.015	0.00046	-0.0079	0.019	0.33	0.4	0.36	0.1	0.26	0.46	0.12	0.33	-0.0076	0.0044	-0.059
TAV_Month	-0.0065	1	0.15	-0.43	-0.014	-0.009	0.0014	0.0016	-0.0025	-0.012	0.0066	-0.0037	0.0066	-0.0012	0.0085	0.0072	-0.0055
TDIFF	-0.015	0.15	1	-0.069	-0.0019	-0.014	0.0092	0.0033	0.007	-0.01	-0.0033	-0.018	-0.0004	0.0072	-0.0053	0.0012	0.034
TSNOW_Month	0.00046	-0.43	-0.069	1	-0.0045	-0.0059	0.0025	0.0017	0.0039	-0.012	0.0022	0.0033	-0.00025	0.0049	-0.0014	-0.0026	0.0065
host_response_rate	-0.0079	-0.014	-0.0019	-0.0045	1	0.016	0.023	0.035	0.049	0.14	-0.043	-0.0042	-0.0083	0.046	-0.044	-0.015	0.16
host_total_listings_count	0.019	-0.009	-0.014	-0.0059	0.016	1	0.0041	-0.03	-0.0048	0.033	0.063	0.11	-0.069	-0.011	0.099	-0.00049	-0.081
bathrooms	0.33	0.0014	0.0092	0.0025	0.023	0.0041	1	0.47	0.46	0.091	0.12	0.26	0.047	0.27	0.0028	-0.00033	-0.033
bedrooms	0.4	0.0016	0.0033	0.0017	0.035	-0.03	0.47	1	0.7	0.14	0.15	0.4	0.092	0.53	-0.032	0.0048	-0.0072
beds	0.36	-0.0025	0.007	0.0039	0.049	-0.0048	0.46	0.7	1	0.18	0.13	0.4	0.07	0.54	-0.0092	-0.0023	0.04
amenities_count	0.1	-0.012	-0.01	-0.012	0.14	0.033	0.091	0.14	0.18	1	0.056	0.13	0.1	0.19	-0.017	-0.0049	0.22
security_deposit	0.26	0.0066	-0.0033	0.0022	-0.043	0.063	0.12	0.15	0.13	0.056	1	0.42	0.1	0.11	0.17	0.0034	-0.15
cleaning_fee	0.46	-0.0037	-0.018	0.0033	-0.0042	0.11	0.26	0.4	0.4	0.13	0.42	1	0.085	0.34	0.15	0.0034	-0.16
price_for_extra_people	0.12	0.0066	-0.0004	0.00025	0.0083	-0.069	0.047	0.092	0.07	0.1	0.1	0.085	1	0.25	-0.064	0.0028	0.029
guests_included	0.33	-0.0012	0.0072	0.0049	0.046	-0.011	0.27	0.53	0.54	0.19	0.11	0.34	0.25	1	-0.03	0.0039	0.071
minimum_nights	-0.0076	0.0085	-0.0053	-0.0014	-0.044	0.099	0.0028	-0.032	-0.0092	-0.017	0.17	0.15	-0.064	-0.03	1	0.0065	-0.2
maximum_nights	0.0044	0.0072	0.0012	-0.0026	-0.015	-0.00049	0.00033	0.0048	-0.0023	-0.0049	0.0034	0.0034	0.0028	0.0039	0.0065	1	-0.0053
number_of_reviews_ltm	-0.059	-0.0055	0.034	0.0065	0.16	-0.081	-0.033	-0.0072	0.04	0.22	-0.15	-0.16	0.029	0.071	-0.2	-0.0053	1
review_scores_rating	0.027	0.0055	-0.003	-0.0064	0.075	-0.058	-0.0057	-0.0011	-0.038	0.14	0.002	-0.012	0.039	0.0052	-0.07	0.0047	0.064
review_scores_accuracy	-0.0082	0.0077	-0.0014	-0.0072	0.078	-0.057	-0.027	-0.029	-0.053	0.13	-0.019	-0.05	0.034	0.0067	-0.079	0.0028	0.12
review_scores_cleanliness	0.056	0.0014	0.0013	-0.0056	0.1	-0.032	-0.012	0.012	-0.0061	0.17	0.012	0.019	0.041	0.034	-0.087	0.0043	0.11
review_scores_checkin	-0.0087	0.0083	0.0057	-0.0058	0.081	-0.089	-0.014	0.0078	-0.011	0.1	-0.021	-0.043	0.022	0.02	-0.062	-0.015	0.1
review_scores_communication	-0.0041	0.0083	0.002	-0.003	0.091	-0.071	-0.028	-0.0083	-0.035	0.11	-0.024	-0.044	0.027	-0.00065	-0.069	0.0019	0.088
review_scores_location	0.11	-0.0051	-0.034	-0.015	0.03	0.0021	-0.038	-0.047	-0.052	0.045	0.046	0.066	0.027	-0.056	-0.017	0.0041	0.061
review_scores_value	-0.029	0.0091	0.0064	-0.0073	0.067	-0.098	-0.0062	-0.012	-0.047	0.11	-0.078	-0.11	0.017	-0.013	-0.11	0.0034	0.1
Subway_Count_Within_200m	0.071	0.0041	0.003	0.0016	-0.023	0.053	-0.01	-0.01	-0.0045	-0.021	0.035	0.054	0.0026	-0.0046	0.014	0.0065	-0.028
Subway_Count_Within_1000m	0.2	-0.013	-0.011	0.011	-0.041	0.15	-0.024	-0.036	-0.033	-0.031	0.12	0.19	0.027	-0.028	0.071	0.008	-0.046
Bus_Count_Within_200m	0.081	-0.0084	-0.014	0.0027	-0.016	0.073	-0.017	-0.02	-0.0038	-0.0099	0.053	0.065	0.0015	0.0063	0.044	-0.0023	-0.017
Bus_Count_Within_1000m	0.16	-0.018	-0.039	0.0064	-0.036	0.12	-0.053	-0.06	-0.032	-0.049	0.099	0.14	0.011	-0.032	0.066	-0.0027	-0.026
Restaurants_Count_Within_200m	0.17	-0.014	-0.038	0.011	-0.045	0.078	-0.058	-0.042	-0.05	-0.07	0.094	0.16	0.015	-0.042	0.054	-0.0016	-0.024
Restaurants_Count_Within_1000m	0.25	-0.017	-0.036	0.015	-0.063	0.071	-0.05	-0.049	-0.051	-0.076	0.15	0.22	0.038	-0.051	0.087	0.0029	-0.041
Shops_Count_Within_200m	0.064	-0.0043	-0.018	0.0049	-0.021	0.078	-0.022	-0.015	-0.031	-0.048	0.068	0.092	0.01	-0.026	0.054	0.013	-0.036
Shops_Count_Within_1000m	0.14	-0.0045	-0.0097	0.01	-0.036	0.021	0.00088	-0.0068	-0.033	-0.06	0.095	0.17	0.021	-0.029	0.057	0.014	-0.038
Attractions_Count_Within_200m	0.15	-0.0096	0.0025	0.0057	-0.025	0.031	-0.015	-0.0061	0.00041	-0.024	0.061	0.089	0.033	0.0068	0.01	-0.0018	-0.012
Attractions_Count_Within_1000m	0.2	-0.016	-0.023	0.01	-0.055	0.1	-0.056	-0.055	-0.033	-0.052	0.1	0.15	0.032	-0.043	0.068	-0.00029	-0.016

	review_scores_rating	review_scores_accuracy	review_scores_cleanliness	review_scores_checkin	review_scores_communication	review_scores_location	review_scores_value	Subway_Count_Within_200m	Subway_Count_Within_1000m	Bus_Count_Within_200m	Bus_Count_Within_1000m	Restaurants_Count_Within_200m	Restaurants_Count_Within_1000m	Shops_Count_Within_200m	Shops_Count_Within_1000m	Attractions_Count_Within_200m	Attractions_Count_Within_1000m
price	0.027	-0.0082	0.056	-0.0087	-0.0041	0.11	-0.029	0.071	0.2	0.081	0.16	0.17	0.25	0.064	0.14	0.15	0.2
TAV_Month	0.0055	0.0077	0.0014	0.0083	0.0083	-0.0051	0.0091	0.0041	-0.013	-0.0084	-0.018	-0.014	-0.017	-0.0043	-0.0045	-0.0096	-0.016
TDIFF	-0.003	-0.0014	0.0013	0.0057	0.002	-0.034	0.0064	0.003	-0.011	-0.014	-0.039	-0.038	-0.036	-0.018	-0.0097	0.0025	-0.023
TSNOW_Month	-0.0064	-0.0072	-0.0056	-0.0058	-0.003	-0.015	-0.0073	0.0016	0.011	0.0027	0.0064	0.011	0.015	0.0049	0.01	0.0057	0.01
host_response_rate	0.075	0.078	0.1	0.081	0.091	0.03	0.067	-0.023	-0.041	-0.016	-0.036	-0.045	-0.063	-0.021	-0.036	-0.025	-0.055
host_total_listings_count	-0.058	-0.057	-0.032	-0.089	-0.071	0.0021	-0.098	0.053	0.15	0.073	0.12	0.078	0.071	0.078	0.021	0.031	0.1
bathrooms	-0.0057	-0.027	-0.012	-0.014	-0.028	-0.038	-0.0062	-0.01	-0.024	-0.017	-0.053	-0.058	-0.05	-0.022	0.00088	-0.015	-0.056
bedrooms	-0.0011	-0.029	0.012	0.0078	-0.0083	-0.047	-0.012	-0.01	-0.036	-0.02	-0.06	-0.042	-0.049	-0.015	-0.0068	-0.0061	-0.055
beds	-0.038	-0.053	-0.0061	-0.011	-0.035	-0.052	-0.047	-0.0045	-0.033	-0.0038	-0.032	-0.05	-0.051	-0.031	-0.033	0.00041	-0.033
amenities_count	0.14	0.13	0.17	0.1	0.11	0.045	0.11	-0.021	-0.031	-0.0099	-0.049	-0.07	-0.076	-0.048	-0.06	-0.024	-0.052
security_deposit	0.002	-0.019	0.012	-0.021	-0.024	0.046	-0.078	0.035	0.12	0.053	0.099	0.094	0.15	0.068	0.095	0.061	0.1
cleaning_fee	-0.012	-0.05	0.019	-0.043	-0.044	0.066	-0.11	0.054	0.19	0.065	0.14	0.16	0.22	0.092	0.17	0.089	0.15
price_for_extra_people	0.039	0.034	0.041	0.022	0.027	0.027	0.017	0.0026	0.027	0.0015	0.011	0.015	0.038	0.01	0.021	0.033	0.032
guests_included	0.0052	0.0067	0.034	0.02	-0.00065	-0.056	-0.013	-0.0046	-0.028	0.0063	-0.032	-0.042	-0.051	-0.026	-0.029	0.0068	-0.043
minimum_nights	-0.07	-0.079	-0.087	-0.062	-0.069	-0.017	-0.11	0.014	0.071	0.044	0.066	0.054	0.087	0.054	0.057	0.01	0.068
maximum_nights	0.0047	0.0028	0.0043	-0.015	0.0019	0.0041	0.0034	0.0065	0.008	-0.0023	-0.0027	-0.0016	0.0029	0.013	0.014	-0.0018	0.00029
number_of_reviews_ltm	0.064	0.12	0.11	0.1	0.088	0.061	0.1	-0.028	-0.046	-0.017	-0.026	-0.024	-0.041	-0.036	-0.038	-0.012	-0.016
review_scores_rating	1	0.76	0.7	0.58	0.65	0.43	0.75	-3.6e-05	-0.013	-0.031	-0.067	-0.028	-0.034	0.006	0.016	-0.017	-0.043
review_scores_accuracy	0.76	1	0.6	0.56	0.59	0.38	0.67	-0.012	-0.029	-0.038	-0.079	-0.041	-0.054	-0.0063	-0.0021	-0.021	-0.061
review_scores_cleanliness	0.7	0.6	1	0.44	0.47	0.31	0.6	-0.015	-0.038	-0.02	-0.048	-0.044	-0.056	-0.012	-0.019	-0.015	-0.042
review_scores_checkin	0.58	0.56	0.44	1	0.65	0.32	0.5	-0.019	-0.05	-0.043	-0.079	-0.049	-0.062	-0.013	-0.006	-0.031	-0.071
review_scores_communication	0.65	0.59	0.47	0.65	1	0.35	0.55	-0.0026	-0.018	-0.039	-0.059	-0.02	-0.033	-0.0012	0.0075	-0.024	-0.043
review_scores_location	0.43	0.38	0.31	0.32	0.35	1	0.39	0.095	0.23	0.071	0.14	0.27	0.32	0.17	0.27	0.12	0.21
review_scores_value	0.75	0.67	0.6	0.5	0.55	0.39	1	-0.014	-0.07	-0.056	-0.12	-0.088	-0.11	-0.027	-0.031	-0.043	-0.095
Subway_Count_Within_200m	-3.6e-05	-0.012	-0.015	-0.019	-0.0026	0.095	-0.014	1	0.36	0.23	0.16	0.18	0.22	0.2	0.14	0.21	0.18
Subway_Count_Within_1000m	-0.013	-0.029	-0.038	-0.05	-0.018	0.23	-0.07	0.36	1	0.26	0.5	0.59	0.73	0.23	0.47	0.42	0.66
Bus_Count_Within_200m	-0.031	-0.038	-0.02	-0.043	-0.039	0.071	-0.056	0.23	0.26	1	0.5	0.19	0.26	0.077	0.07	0.16	0.26
Bus_Count_Within_1000m	-0.067	-0.079	-0.048	-0.079	-0.059	0.14	-0.12	0.16	0.5	0.5	1	0.52	0.62	0.13	0.17	0.27	0.58
Restaurants_Count_Within_200m	-0.028	-0.041	-0.044	-0.049	-0.02	0.27	-0.088	0.18	0.59	0.19	0.52	1	0.76	0.4	0.43	0.33	0.57
Restaurants_Count_Within_1000m	-0.034	-0.054	-0.056	-0.062	-0.033	0.32	-0.11	0.22	0.73	0.26	0.62	0.76	1	0.32	0.61	0.41	0.74
Shops_Count_Within_200m	0.006	-0.0063	-0.012	-0.013	-0.0012	0.17	-0.027	0.2	0.23	0.077	0.13	0.4	0.32	1	0.54	0.1	0.1
Shops_Count_Within_1000m	0.016	-0.0021	-0.019	-0.006	0.0075	0.27	-0.031	0.14	0.47	0.07	0.17	0.43	0.61	0.54	1	0.15	0.21
Attractions_Count_Within_200m	-0.017	-0.021	-0.015	-0.031	-0.024	0.12	-0.043	0.21	0.42	0.16	0.27	0.33	0.41	0.1	0.15	1	0.37
Attractions_Count_Within_1000m	-0.043	-0.061	-0.042	-0.071	-0.043	0.21	-0.095	0.18	0.66	0.26	0.58	0.57	0.74	0.1	0.21	0.37	1

6.2 Appendix Tables

6.2.1 Counts of Null values rows

Column	Null Values	Column	Null Values	Column	Null Values	Column	Null Values
host_since	925	host_total_listings_count	925	beds	1157	review_scores_accuracy	155684
host_location	2978	host_identity_verified	925	**weekly_price**	610992	review_scores_cleanliness	155467
host_response_time	250202	neighbourhood	7072	**monthly_price**	621513	review_scores_checkin	155928
host_response_rate	250203	city	1149	security_deposit	246334	review_scores_communication	155626
host_acceptance_rate	657874	zipcode	7521	cleaning_fee	150379	review_scores_location	155988
host_is_superhost	925	market	1497	first_review	141611	review_scores_value	155977
host_neighbourhood	93688	bathrooms	799	last_review	141611	cancellation_policy	1
host_listings_count	925	bedrooms	622	review_scores_rating	155161	reviews_per_month	141611

6.2.2 A summary of review scores among different categories

Review Category	Review Score	Distribution (%)
Overall Rating	95-100	59.90
	90-94	21.96
	85-89	8.37
	80-84	5.96
	< 80	3.81
Location	10	65.44
	9	28.78
	< 8	5.78
Communication	10	81.04
	9	14.52
	< 8	4.44
Cleanliness	10	52.18
	9	33.42
	< 8	14.40
Value	10	49.85
	9	40.13
	< 8	10.02

6.2.3 OLS Regression Result

Dep. Variable:	price	**R-squared:**	0.613
Model:	OLS	**Adj. R-squared:**	0.613
Method:	Least Squares	**F-statistic:**	3759
Date:	Thu, 23 Apr 2020	**Prob (F-statistic):**	0
Time:	21:15:19	**Log-Likelihood:**	-7.30E+05
No. Observations:	142259	**AIC:**	1.46E+06
Df Residuals:	142198	**BIC:**	1.46E+06

Df Model:	60
Covariance Type:	nonrobust

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
host_response_rate	0.0382	0.013	2.952	0.003	0.013	0.063
host_is_superhost	4.6277	0.257	17.991	0	4.124	5.132
host_total_listings_count	0.0204	0.004	5.244	0	0.013	0.028
calculated_host_listings_count	0.6551	0.247	2.654	0.008	0.171	1.139
calculated_host_listings_count_entire_homes	-0.747	0.247	-3.021	0.003	-1.232	-0.262
calculated_host_listings_count_private_rooms	-0.7782	0.249	-3.127	0.002	-1.266	-0.29
calculated_host_listings_count_shared_rooms	-2.3162	0.286	-8.112	0	-2.876	-1.757
bathrooms	8.0469	0.308	26.096	0	7.443	8.651
bedrooms	16.0593	0.207	77.602	0	15.654	16.465
beds	3.2331	0.137	23.549	0	2.964	3.502
amenities_count	0.2098	0.012	17.128	0	0.186	0.234
is_wifi	-3.3118	1.19	-2.784	0.005	-5.644	-0.98

is_heating	-2.1293	0.727	-2.929	0.003	-3.554	-0.704
is_aircon	6.7679	0.389	17.401	0	6.006	7.53
security_deposit	0.0018	0	6.607	0	0.001	0.002
cleaning_fee	0.1341	0.003	47.237	0	0.129	0.14
price_for_extra_people	0.037	0.004	8.243	0	0.028	0.046
guests_included	8.0206	0.12	66.673	0	7.785	8.256
minimum_nights	-0.4131	0.007	-57.455	0	-0.427	-0.399
has_availability	-29.9606	2.405	-12.457	0	-34.675	-25.246
availability_30	0.3793	0.019	19.876	0	0.342	0.417
availability_90	0.0262	0.007	3.768	0	0.013	0.04
availability_365	0.0072	0.001	6.625	0	0.005	0.009
is_instant_bookable	1.5178	0.252	6.031	0	1.025	2.011
is_business_travel_ready	1.34E-13	1.80E-14	7.468	0	9.91E-14	1.70E-13
number_of_reviews	-0.0121	0.003	-4.421	0	-0.017	-0.007
reviews_per_month	-0.431	0.117	-3.69	0	-0.66	-0.202
number_of_reviews_ltm	-0.1563	0.012	-13.007	0	-0.18	-0.133
review_scores_rating	0.7835	0.033	23.62	0	0.719	0.849
review_scores_accuracy	-0.7722	0.254	-3.034	0.002	-1.271	-0.273
review_scores_cleanliness	3.1806	0.186	17.136	0	2.817	3.544
review_scores_checkin	-2.3128	0.265	-8.727	0	-2.832	-1.793
review_scores_communication	-2.6207	0.279	-9.402	0	-3.167	-2.074
review_scores_location	7.4328	0.205	36.272	0	7.031	7.834
review_scores_value	-3.7063	0.227	-16.294	0	-4.152	-3.26
Subway_Count_Within_200m	2.9626	0.223	13.298	0	2.526	3.399

Bus_Count_Within_1000m	-0.0977	0.013	-7.689	0	-0.123	-0.073
Restaurants_Count_Within_1000m	0.1196	0.002	48.577	0	0.115	0.124
Shops_Count_Within_200m	-0.6101	0.07	-8.692	0	-0.748	-0.473
Shops_Count_Within_1000m	0.1607	0.009	16.97	0	0.142	0.179
Attractions_Count_Within_200m	2.9165	0.24	12.14	0	2.446	3.387
Attractions_Count_Within_1000m	0.2166	0.021	10.403	0	0.176	0.257
TAV_Month	0.1077	0.028	3.793	0	0.052	0.163
TDIFF_Month	0.2724	0.072	3.789	0	0.132	0.413
holidays	0.4644	0.151	3.076	0.002	0.169	0.76
season_Autumn	-6.1423	0.63	-9.747	0	-7.377	-4.907
season_Spring	-8.054	0.696	-11.574	0	-9.418	-6.69
season_Summer	-6.5335	0.711	-9.19	0	-7.927	-5.14
season_Winter	-9.2309	0.719	-12.845	0	-10.639	-7.822
Brooklyn	12.5552	0.761	16.507	0	11.064	14.046
Manhattan	28.2823	0.785	36.034	0	26.744	29.821
Queens	3.9478	0.804	4.908	0	2.371	5.524
Staten Island	-10.3166	1.387	-7.44	0	-13.034	-7.599
moderate	-2.5173	0.404	-6.236	0	-3.309	-1.726
strict	-34.4056	7.841	-4.388	0	-49.775	-19.036
strict_14_with_grace_period	-2.8624	0.374	-7.659	0	-3.595	-2.13
super_strict_60	39.5792	3.784	10.459	0	32.162	46.996
within a day	-2.947	1.155	-2.552	0.011	-5.21	-0.684
within a few hours	-3.8388	1.25	-3.07	0.002	-6.289	-1.388
within an hour	-3.0227	1.271	-2.378	0.017	-5.514	-0.532

Hotel room	-13.0244	3.678	-3.541	0	-20.234	-5.815
Private room	-53.578	0.293	-183.005	0	-54.152	-53.004
Shared room	-75.2367	1.001	-75.157	0	-77.199	-73.275

Omnibus:	16426.185	**Durbin-Watson:**	2.011
Prob(Omnibus):	0	**Jarque-Bera (JB):**	41682.396
Skew:	0.667	**Prob(JB):**	0
Kurtosis:	5.292	**Cond. No.**	1.53E+16