AIRBNB Price Prediction

by Tigran Zohrabyan

Abstract

- Airbnb is the world's most prominent online hospitality marketplace. It has a publicly available dataset, divided region wise which has been used in my predictions and analytics to determine how various features of property correlate to the price of the listing in the Airbnb website. In order to make my predictions more robust, I also used the text reviews by the customers and performed sentiment analysis on it, subsequently adding it as a feature for training the model. For the training part, I used data for Toronto but the model can be used for any region since the data structure is uniform throughout. For future work, I hope to collect more extensive and more diverse dataset to enhance its performance and generalizability.

Introduction

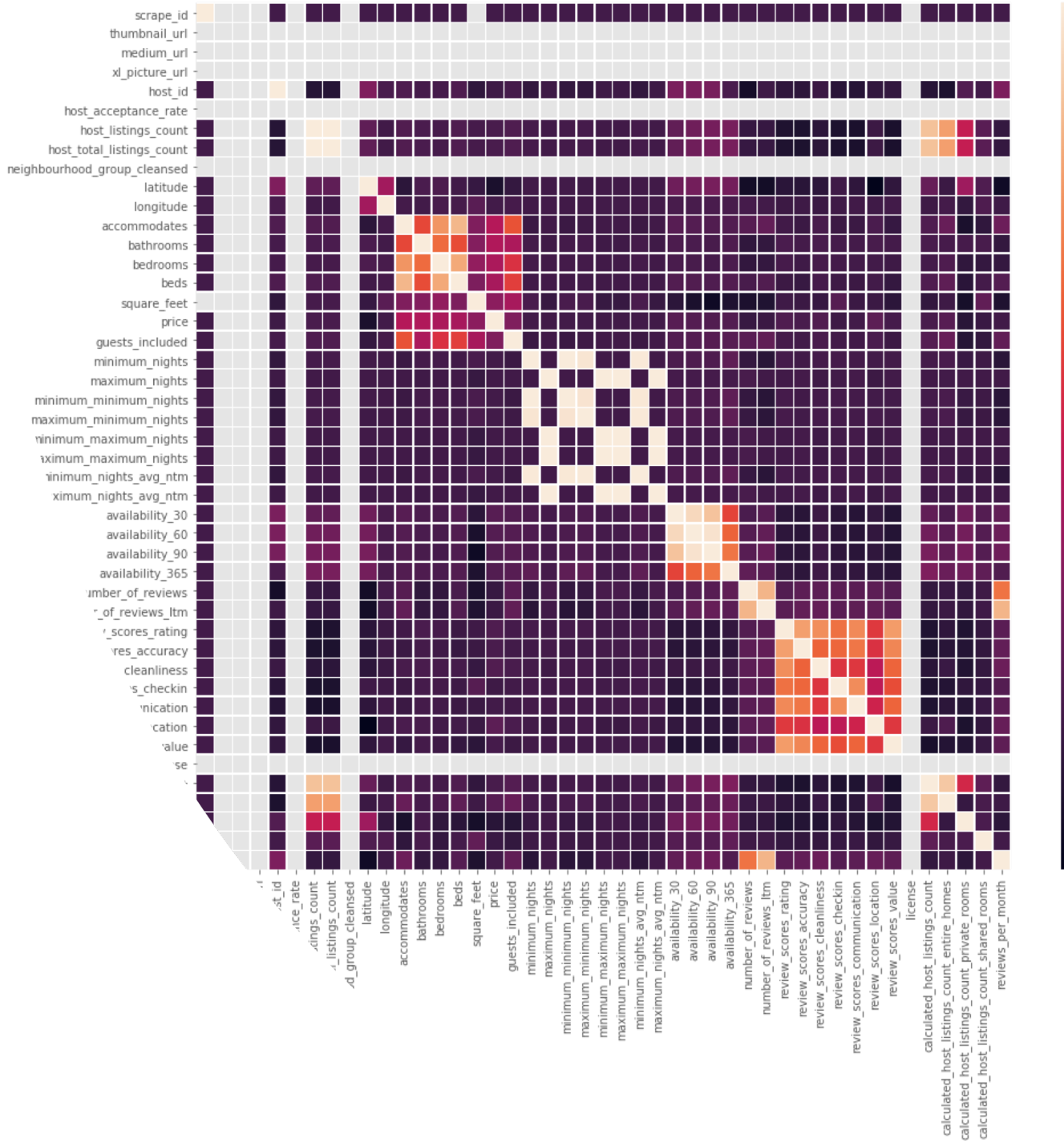
- ‘Inside Airbnb’ provides the data for various listings of the property in the Airbnb website categorized region wise. Its prudent to have a good understanding of the features which are the driving factor in determining the price of a listing for the Airbnb as the business and homeowners as well. In order to determine how various housing attributed influence the pricing of a listing in Toronto the models described in this paper makes a systematic analysis of several typical aspects using python and its libraries.

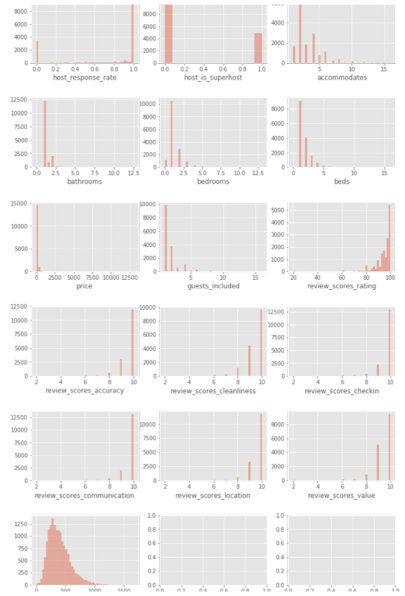
Data understanding

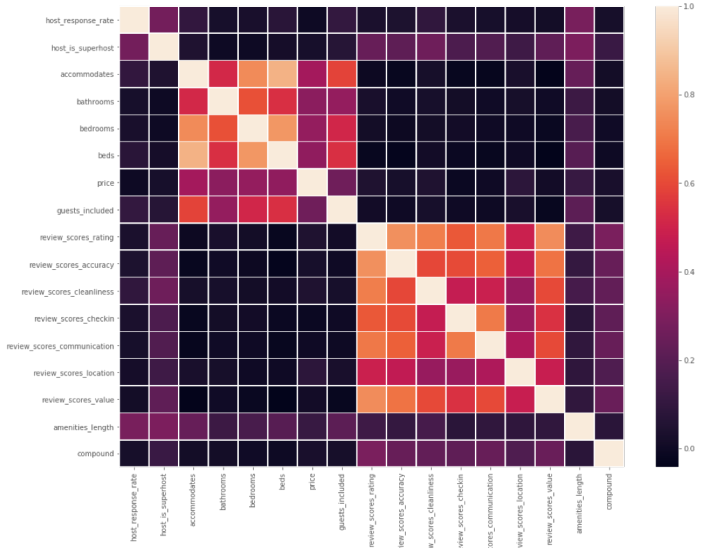
- For my model, the data was downloaded from Inside Airbnb, which contains the database of various cities worldwide. The data utilizes public information compiled from the Airbnb website including the availability calendar for 365 days in the future, and the reviews for each listing, and hence seems a reliable source for my analysis. I performed the analysis using data for Toronto, which can potentially be later replaced with data for any other city using the same source since the data for all the cities is structured in the same way. The dataset for each city provides a listing file which contains all the listings on the Airbnb website for that particular city which contains about 96 features, ranging from listing id, listing URL, to information about the property, property type, host type, amenities.

Data preparation

- For my model, the data was downloaded from the Inside Airbnb. The two datasets downloaded were listing and reviews. The data preparation started with cleaning the data. Data was cleaned by removing the row with lots of missing values and also removing the NaN values. Features like host_is_superhost, property_type, bed_type are recoded to get uniform values. After transformations and selecting the relevant features, I get the following correlation between the features. Other variables like zipcode, property_type, room_type, bed_type are then converted to dummy variables in order to convert them into categorical variables.







Data understanding

- Other variables like zipcode, property_type, room_type, bed_type are then converted to dummy variables in order to convert them into categorical variables. After the above transformations, I get the following correlation between the features.

Feature Engineering

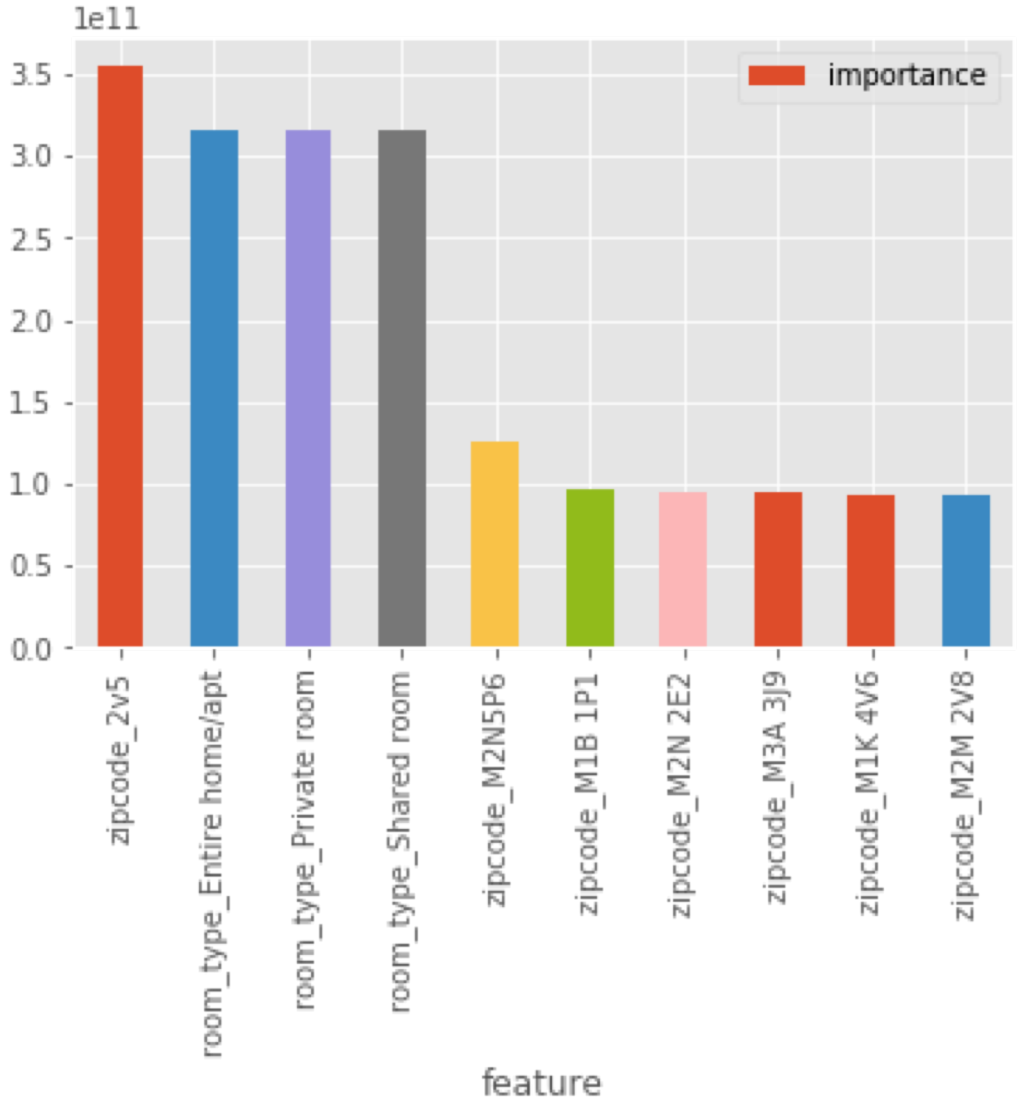
- After cleaning the data, Pandas was used to describe various features and were evaluated on various parameters like count, mean, median etc. to determine its aptness to be used for the modeling. I have used Vader module of NLTK to perform sentiment analysis and provide each review with a sentiment score. The sentiment score was classified as positive, negative or neutral.

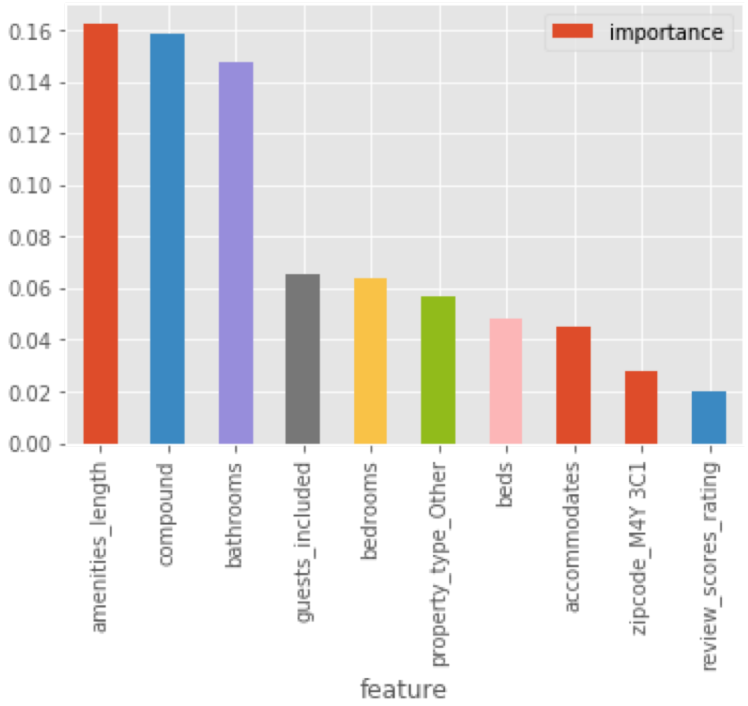
Modeling and Evaluation

- The target variable is the price of an airbnb property. Given the nature of features after modifications and the problem at hand, a simple logistic regression algorithm is one very efficient way to predict the target variable. Due to the dataset-size and it's computationally inexpensive nature, this algorithm is a great choice for a baseline model. Logistic regression models are easily comprehensible and transparent in nature. I also used Decision trees, as they are known to be compatible with both types of tasks i.e regression and classification. Along with Logistic Regression and Decision trees, I applied a Random Forest model. The Random Forest model is a complex model and is suitable for both regression and classification tasks. Even though Random Forest is a complex algorithm, its level of understanding of decision trees is quite simple.

Regression

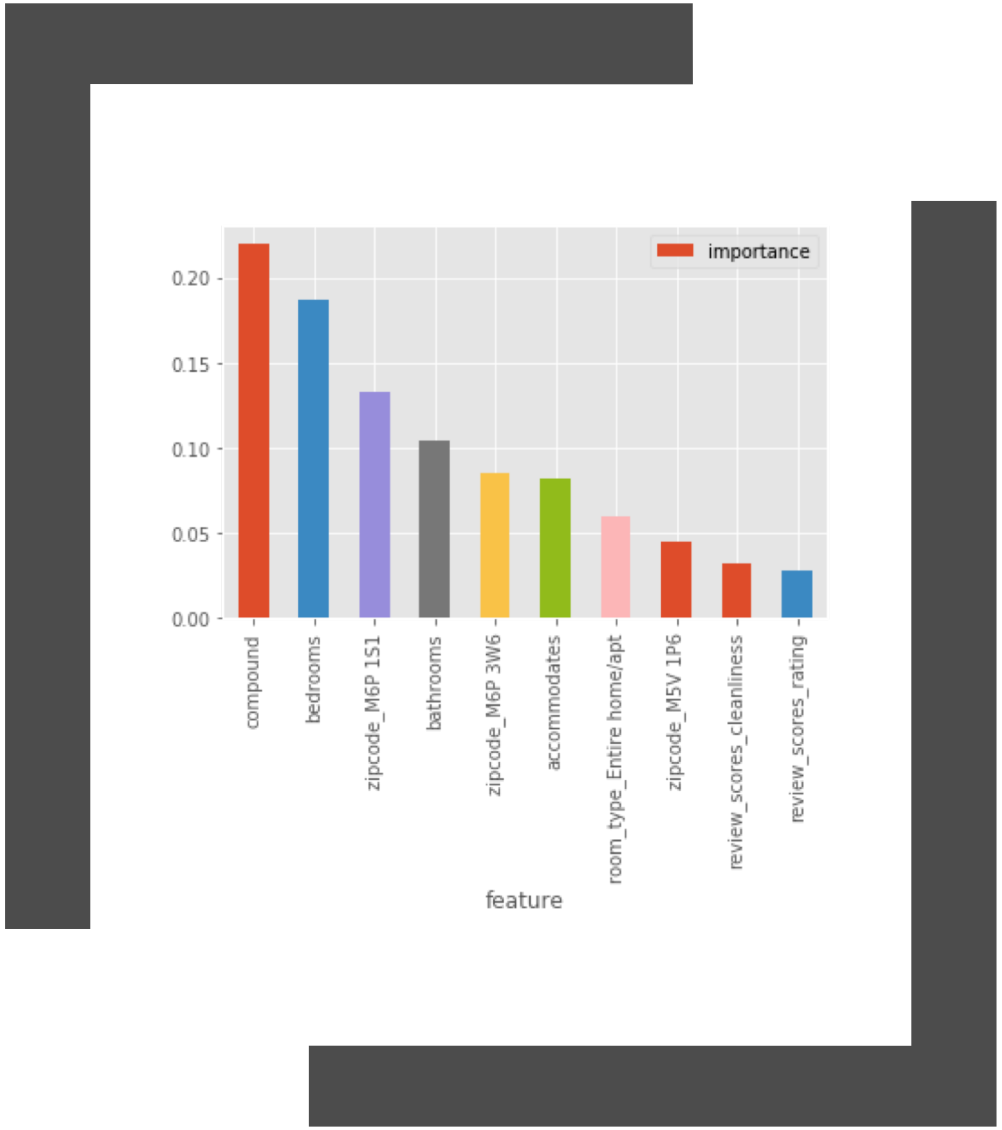
- Since Logistic Regression has been proven to be very predictive when it comes to medium size dataset I applied my clean and feature engineered dataset to get the prediction on the prices. I used cross-validation to deal with overfitting. It gave me a Mean Squared Error value of 13836547592.2. To better understand what is driving the predictive decisions, below is the weighting of the features in the regression model.





Decision Trees

- Since decision trees have in-memory classification, are not computationally expensive and are easy to use, this model is my second choice to predict my target variable. One other important reason I used a decision tree is their capability to handle a dataset that contains a higher degree of errors and missing values. I applied a decision tree classifier from the Sklearn library with varying maximum depths and a random state set to zero. It gave me a Mean Squared Error value of 336.6. To better understand what is driving the predictive decisions, below is the weighting of the features in the decision tree.



Random Forest

- Random Forest is the ensemble of the decision tree. It generally gives a better result than a decision tree as it helps averages out the result of all the decision tree leading to canceling of the noise. I used the same clean and featured engineered data to run this algorithm. It gave me a Mean Squared Error value of 110.3. To better understand what is driving the predictive decisions, below is the weighting of the features in the random forest model.

```
print("OLS MSE",ols_mse)
print("Decision Tree MSE:", dtree_mse)
print("Random Forest MSE:", rf_mse)
```

```
OLS MSE 13836547592.224438
Decision Tree MSE: 336.60805998447455
Random Forest MSE: 110.35120208010724
```

```
print("OLS R^2",ols_r2)
print("Decision Tree R^2:", dtree_r2)
print("Random Forest R^2:", rf_r2)
```

```
OLS R^2 -1.075105769495634e+16
Decision Tree R^2: -5.362748125398965
Random Forest R^2: 0.3161673276387116
```

Model selection

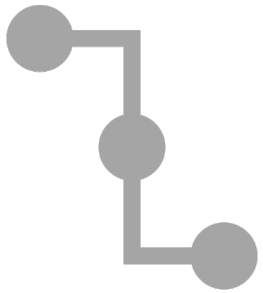
Based on the models we get the below results.

Conclusion

- From a business and implementation standpoint, I have succeeded in creating a model that does a fair job in predicting the prices. A varied number of features were successfully cleaned, pre-processed and modeled on to get the final regression models with optimized parameters.

Appendix

For my model, the data was downloaded from: http://insideairbnb.com/



Project link: https://github.com/tikoz86/Final-Project