Airbnb price prediction in NYC

Project Final Report

Kewei WANG
b00739642@essec.edu

Jingyi LI
b00740918@essec.edu

Tandez Sarkaria
b00747092@essec.edu

Carlo Dalla Quercia
b00747370@essec.edu

ABSTRACT

Airbnb is a company that operates an online marketplace and hospitality service which is accessible via its websites and mobile apps. Members can use the service to arrange or offer lodging, primarily homestays, or tourism experiences. The company does not own any of the real estate listings, nor does it host events; as a broker, it receives commissions from every booking. With the sharing economy booming, airbnb is becoming more and more popular in the hospitality industry. People now are used to sharing their private houses or rooms and booking other people's houses or rooms when travelling. However, how to price the house and how to distinguish overpricing has become a big problem.Our prediction work is to help these people reasonably determine their houses'price when renting and help people distinguish the acceptable house price when ordering. Since the guests can search for lodging using filters such as lodging type, dates, location, and price, and hosts provide prices and other details for their rental or event listings. Therefore, firstly we are searching for the suitable dataset, the final dataset we chose to use inluded the housing price and related variables like the type of the house, how many rooms in the house, the location of the house, the applications in the house, and even the reviews about the house.

1 INTRODUCTION

This project is using the dataset from airbnb houses'related information such as types, locations to decide the accurate price for the house renting in the airbnb platform. Actually, with the development of Airbnb, more and more householders tend to rent their houses in the airbnb platform and more and more travellers tend to accommodate in the houses with the airbnb platform. So there will be a problem about the price setting, correct price setting will help householders find renters easily and save unnecessary costs for the renters. How to set the price reasonably and how to select a reasonably priced house are important for all the airbnb users, which is also the questions our project wants to answer. And our work is to build a suitable model to predict the reasonable housing price in order to help householders to set price properly and for renters to select suitably priced house. What's more, the potential application is that once such predicting model is built, and proves to be feasible, the Airbnb platform can introduce such predicting model to give houseowners a recommended price based on input he/she shares about the listing. It will be an improvement for airbnb platform. It can also be used by some real estate agencies when doing short-term renting business.

2 PROBLEM DEFINITION

We downloaded the original data from Kaggle platform, and the original data has following different variables: log_price: The log form of the house renting price; property_type : The different types of houses; room_type : The different types of rooms; amenities: The stuff included in the renting house; accommodates: The number of rooms in the house; bathrooms: The number of bathrooms in the house; bed_type: The type of beds; cancellation_policy: The cancellation policy of the renting; cleaning_fee; city; description: The householders's descriptions of their houses; first_review: The time of first review; host_has_profile_pic; host_identity_verified; host_response_rate; host_since: The time this house was presented in the Airbnb; instant_bookable; last_review: The time of last review; latitude; longitude; name; neighbourhood; number_of_reviews; review_scores_rating; thumbnail_url; zipcode; bedrooms; beds. - Among these variables, log_price is our target variable, and we should make use of other left variables to build the feasible model to predict the price of the renting house. And our goal is also to make the accuracy of our prediction as high as possible, so before our model's building, we set the 70% accuracy threshold for the final test progress.

3 RELATED WORK

In the previous papers like Georgios Zervas (2017)[2] mentioned the airbnb price setting is an important part if airbnb wants to be the actual leader in the hotel industry, and this paper listed many variables that will have influnce on the final price decision, which also have been included in our final input variables. When it comes to how to make the prediction model, we also refer to the previous work by Dan Wang (2016)[1] . This paper summaries the sharing economy based accommodation rentals and hotel price determinants's model that previous researchers used, and we get a lot of inspirations from it. The aim of this study is to identify the price determinants of sharing economy based accommodation offers in the digital marketplace. Specifically, it uses a sample of 180,533 accommodation rental offers in 33 cities listed on Airbnb.com using ordinary least squares and quantile regression analysis. Twenty-five explanatory variables in five categories (host attributes, site and property attributes, amenities and services, rental rules, and online review ratings) are explored for the intricacies of the relationships between pricing and its determinants. And from this, we also use some of its ways to process our existing dataset.

4 METHODOLOGY

To address the regression problem of predicting airbnb price, we followed the machine learning pipeline of inputing data, feature

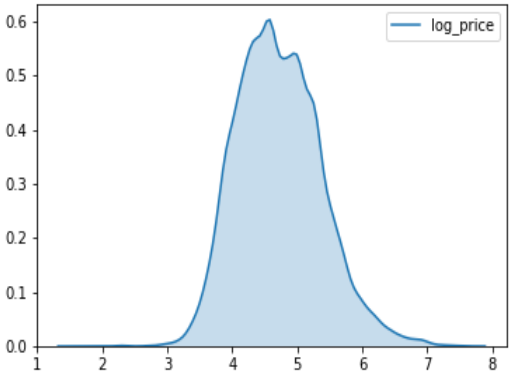

Figure 1: Distribution of log_price.

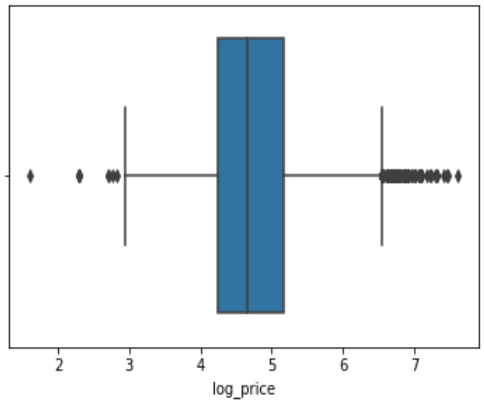

Figure 2: Boxplot of log_price.

and model selection, fit the model on training set and then test the accuracy on the test set and kept the process going until the model reach an ideal accuracy rate.

4.1 Feature engineering

The original dataset from Kaggle contains missing values and many categorical variables that can't be input directly to the model. It contains 28 variables in total with 14 categorical values and some of them can be redundant and useless. Therefore, feature engineering is a critical step to take before actually putting data into models.

4.1.1 Data cleaning.

First of all, we checked with the number of missing values and the dataset is very clean with few missing values, so we deleted the lines with missing values and we still retain 17855 rows of records. As is shown in Figure 1 and Figure 2, the target value(log_price) is almost normally distributed, the variance is low and there are few outliers.

4.1.2 Categorical values.

After checking the type of variables, we transformed the categorical values which actually are numerical values or boolean values. The host response rate is stored as object in a form like "98%", so we transform the variable into a float with 2 decimal points like 0.98. For variables such as "host_has_profile_pic", "host_identity_verified" and "instant_bookable", the original data uses text "t" and "f" for boolean values True and False, so we transformed then into 1 and 0 respectively.

Table 1 shows the unique number of categorical variables with an example in the original dataset. For categorical variables with small number of unique value such as bed type, cancellation policy and room type, we simply create dummy variables for them. For variables with large number of unique values, one way is to extract useful information from the variable and create a new feature, the other is to classify it into smaller amount of categories with current information.

Table 1: Frequency of Special Characters

Variable	Unique	Example
thumbnail_url	17853	https://a0.muscache.com/im/...
Name	17753	Spacious Private Room in Brooklyn
Description	17723	Welcome to RMH, a co-ed hostel vibe home...
Amenities	16950	{"Wireless Internet","Air conditioning",...}
host_since	2787	2014-11-02
first_review	2112	2017-09-04
last_review	727	2017-09-24
zipcode	402	11221
neighbourhood	193	Williamsburg
property_type	24	Apartment
bed_type	5	Real bed
cancellation_policy	5	Strict
room_type	3	Private room

Amenities

According to a consumer survey commissioned by Airbnb, an overwhelming 97 percent of US travelers surveyed say amenities impact their travel experience[4]. In the Airbnb report, amenities are classified into 3 main categories, which are functionality, thoughtfulness and special. The functionality represents for basic functional equipment like air conditioner, coffee maker, washing machine and anything else supports the basic needs. Amenities which are prepared out of traveler's expectation, such as a bottle of wine, a bicycle to get around town, or a beach bag already packed with everything needed for a day in the sun, are considered as thoughtfulness. The last category special represents for amenities that are quite unique and provide extra added-value to the airbnb, for example, a swimming pool, a free parking, pet-friendly conditions.

The dataset contains 127 unique amenities in total, and each row (property) has several amenities stored in a cell. To extract the information from amenities, we separate the amenities by comma for each row and it becomes a matrix with null values because each row does not contain the same amount of amenities. Then according

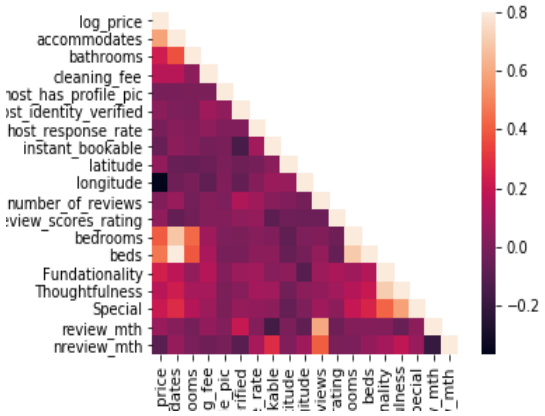

Figure 3: Heatmap of numerical values.

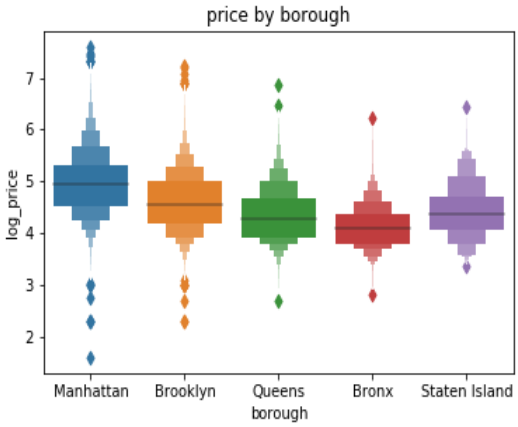

Figure 4: Boxplot for 5 boroughs.

to the classification described above, we count the frequency of functionality, thoughtfulness and special for each row and thus created 3 new numerical features in columns, which are functionality, thoughtfulness and special. For example, if one row has 5 for functionality, 4 for thoughtfulness and 1 for special, it means that the property provides 5 equipments in functionality category, 4 equipments in thoughtfulness category and 1 equipments in special category.

Review date

The dataset contains information of first review date, last review date and number of reviews for per property. We created a new feature called number of reviews per month. The gap between the last review date and first review date is transferred into month.

$$nreview\_mth = \frac{\text{number of reviews}}{\text{last review date} - \text{first review date}}$$

After the above transformation, the number of review between each property can be compared in a similar scale, because the number of reviews for an airbnb which exists for several years are likely to be larger than those for an airbnb which exists for only one month.

Location

The dataset contains some variables related to the location of the property, including longitude, latitude, zipcode and neighbourhood with a decreasing order of precision. To avoid duplication of information in terms of location, we only use neighbourhood in our model. The dataset contains 193 different neighbourhoods, then we classified the neighbourhoods into 5 main boroughs according to the dataset containing neighbourhood information from the US government[3] and finally made dummy variables from boroughs.

4.1.3 *Numerical values.*

After transforming some of the categorical values into numerical values, we use the heatmap to have a preview of which kinds of numerical values can be essential to the model to predict the log price.

Figure 3 shows the correlation between each variable. We can observe that accommodates, bathrooms, cleaning fee, bedrooms, beds and 3 types of amenities show relatively high correlation with log price. Longitude and latitude may not have direct influence on price, so use the location information of borough can be more useful. As is shown in figure 4, the price in 5 boroughs has different levels and the price in Manhattan tends to be high, however, the price in Bronx seems to be lower. In this process, we also deleted the redundant numerical information such as ID because it varies from each row.

4.1.4 *Difficulties.*

The main difficulties of feature engineering come from categorical values, especially those containing text and descriptive information. We haven't made full use of the dataset because categorical variables such as thumbnail_url, description and name are hard to use. However, such information can be useful because the description from the property owner, whether it is detailed or not and whether it is attractive or not, can also influence the decision of customer. Unfortunately, in the current stage, we need more advanced technique to process this kind of information to improve the accuracy of our model.

4.1.5 *Conclusion.*

In conclusion, the final variables put into the initial model are shown in Table 2 and there are 19 variables in total. Categorical values are split into dummy variables.

4.2 Modeling

As the goal is the prediction of a continuous values we had a limited choice of algorithms to use. We had retained most of the variables using either feature engineering or transformations to dummy variables for categorical. At this point we tried prediction using different methods. In each of the models we used k-fold cross validation and analysed the variance to estimate the generalisation error on the test prediction.

Table 2: Variables put into the initial model

Variable	Type	Example
property_type	object	Apartment
borough	object	Manhattan
bed_type	object	Real bed
cancellation_policy	object	Strict
room_type	object	Private room
cleaning fee	bool	1
accommodates	int	7
bathrooms	float	1.0
host_has_profile_pic	bool	1
host_identity_verified	bool	0
host_response_rate	float	1.0
instant_bookable	bool	0
review_scores_rating	float	93.0
bedrooms	float	3.0
beds	float	3.0
Functionality	int	9
Thoughtfulness	int	3
Special	int	2
nreview_mth	float	3.0

4.2.1 *Train and test split.*

In order to reduce overfitting at the moment of the prediction and using information about the data to predict to fit the model, we decided to split the data into training and test set.

4.2.2 *Linear Regression.*

First of all we tried with linear regression, fitting the model on the variable we disposed of and log price as independent. We have therefore been able to reach a 0.6744 accuracy on training set and 0.6933 on test set. We also used feature selection to prevent the risk of overfitting by selecting the three most decisive predictors. The RFE method gave us room_type_Private room, room_type_Shared room, bed_type_Airbed as the most highly correlated variables with log_price. Still, the accuracy of the model did not improve, but it did not worsen either.

4.2.3 *Ridge Regression.*

After that we tried a Ridge regression, which, differently from the Linear Regression estimator, also adds l2 regularisation. The Ridge regression is similar to the linear regression, but it does not just minimize the sum of the squared residuals, but also a product (lambda x slope squared), also called the Ridge regression penalty. Lamda can be any number from 0, when th Ridge becomes a least squares problem, to infinity. The Ridge regression has many uses, and in this case in particular we want to minimize the risk of overfitting, which tipically occurs when the amount of dimensions is great and the amount of data points is not. If overfitting occurs, then the testing error may turn out larger than the training error. Thanks to regularization the influence of each variable is reduced and also the risk of overfitting. In order to pick the best value for the alpha parameter we computed the accuracy (R^2) and error (MSE) for alpha assuming as value all the integers from 1 to 50. We then

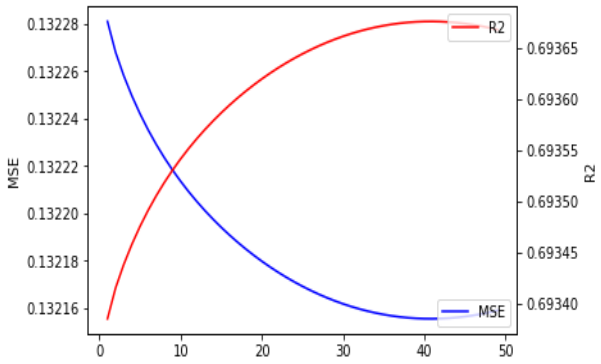

Figure 5: Ridge Accuracy by parameter value.

plotted the results into a graph. Looking at the plot we could tell that the best result was achieved for a level of alpha around 41, with an accuracy of 0.6731 on training and 0.6937 on test and MSE of 0.1322.

4.2.4 *Lasso Regression.*

Lasso is also a linear model that uses l1 prior as a regulariser. Lasso Regression is very similar to Ridge Regression. Again the main objective is to reduce the impact that certain variables have on the prediction model. This time the additional part that the model tries to minimize is (lambda x |slope|). Another difference is that Lasso not only can reduce the weight, but also disregard predictors not informative about the independent variable altogether. In fact, the weights for the variables can decrease as low as zero. In our example it may be useful to figure out which of the many variables should be considered and which are not important, to prevent overfitting. We also used different values of the alpha parameter to compute the best model. In particular we used the values from 0.01 to 0.1 at interval of 0.01. We can say that the model does not have a good prediction accuracy for this situation. The best accuracy picked by the optimal alpha is 0.6759 on test set with a mse of 0.1398

4.2.5 *Support vector regression.*

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). In the same way as with classification approach there is motivation to seek and optimize the generalization bounds given for regression. They relied on defining the loss function that ignores errors, which are situated within the certain distance of the true value. This type of function is often called epsilon intensive loss function. We tried to apply the support vector regression in our model and got a accuracy of 0.7326 and 0.7038 respectively on training set and test set.

4.2.6 *Random Forest Regression.*

Ensemble learning is a learning technique that combines together different models and mixes them to compute the final value of the prediction. We used Random forest Regression, which is an ensemble learning techniques which works by randomly creating a

Table 3: Accuracy and MSE by model

Model	Accuracy	MSE
Linear Regression	0.69	0.13
Ridge Regression	0.69	0.13
Lasso Regression	0.67	0.14
Random Forest Regression	0.72	0.12

different numbers of decision trees and combining them together. In particular, most of them will yield a wrong prediction but the final accuracy will be given by the few working ones.
This time in order to select the best hyper parameter for the model we used the method RandomizedSearchCV from sklearn. We provided it with a grid of hyperparameter ranges (for example for the number of estimators, maximum depth and maximum number of features). The function would then performs different fittings for sampling taken from the hyperparameters grid and determine the best ones. At the same time it also performs a k-fold cross validation to prevent overfitting and giving a better generalisation estimate, using a value we provide for the number of folds.
In this way we read a R^2 accuracy on test set of 0.7157, accuracy on training set of 0.8526 and a mean squared error of 0.1226, which proves to be the best accuracy with the models tested so far, although the gap between training accuracy and test accuracy indicates that the model is bit overfitting.

4.2.7 *Preprocessing and Scaling.*

After taking a first trial on the regression models discussed above, we focused on improving the random forest regression model which achieves the highest accuracy on test sets. We applied the standard scaler method and it ensures that for each feature the mean is 0 and the variance is 1, bringing all features to the same magnitude. The accuracy and MSE after scaling is almost the same as the previous model, but the model computing time is much faster than unscaled data as gradient descent converges much faster with feature scaling. The advantage of computational efficiency makes it easier to tune the random forest model in the following steps.

4.2.8 *Feature selection.*

Then we focus on optimizing the performance of the best model - random forest regression. As is shown in figure 6, the initial model contains some features that actually made few contributions to the model, so feature selection is necessary to help the model neglect the noise, reduce the level of overfitting and perform better on test set.
We use the SelectFromModel function in scikit learn to select the features. The function takes in the initial model of random forest and the threshold value to filter the feature. The random forest model has an attribute of feature_importances_ after fitting the model and the threshold will eliminate the variables which have less feature importance value than the threshold. In order to select the best threshold, we first check the distribution of the feature importance value and select a range from $1e^{-6}$ to $3e^{-5}$ with a step of $1e^{-6}$.
Figure 7 illustrates the change of MSE and test accuracy with the

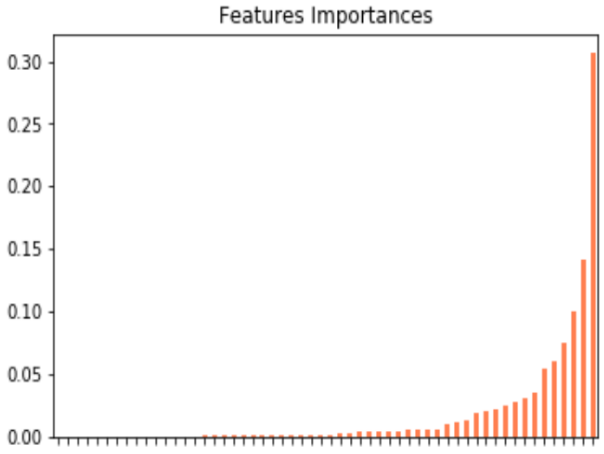

Figure 6: Feature importance of random forest regression.

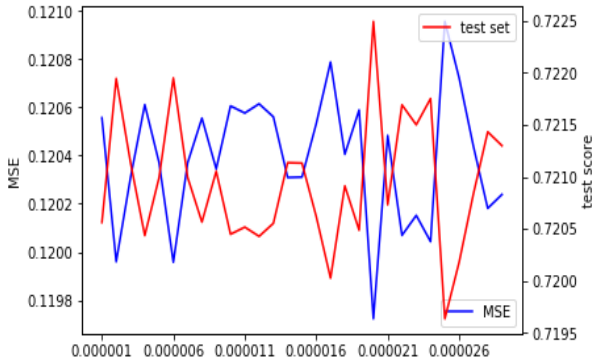

Figure 7: Threshold for picking the best features.

change of threshold value. The threshold of $2e^{-5}$ maximizes the accuracy in test set and minimizes the mean square error on prediction. After the feature selection, the accuracy of the model on test set improves to 0.7219 with a lower MSE of 0.1199.
Here we further check the deleted features after selection. The deleted features are cancellation_policy_super_strict_30 and 8 property types such as Serviced apartment, In-law, Chalet, Cabin, Yurt, Castle. The number of total records having such feature is less than 4 rows among the total 17855 rows, so these feature somehow created some noise to the model.

4.2.9 *Limitations and difficulties.*

While working on the project our computational capacity was limited. To that end we had to use a subset of the original dataset which had about 75000 listings from the six biggest metropolitians of the US. The dataset was pretty granular; one of the variables listed the neighborhood of the listing. We used a subset of the dataset that had listings from the NYC region. This made it feasible for us to classify the neighborhoods into the 5 boroughs of the city and make dummy variables for each borough. Due to time and computational constraint it wasn't feasible to do something

similar for all the cities. For the NYC subset though we observe that the boroughs variables caputre a lot of the information present, not only in the neighborhood variable but also in the latitudes and longitudes variable.

A variable that we would have liked to use but were unable to was Description. This variable gives us a description of the property in the lister's own words. A well written description is generally indicative of the price and any home owner serious about earning money on the platform takes time out to write an appropriate one. If a property is high end and luxurious, generally a homeowner will mention that. On the other end if a property is for budget travellers and a good bargain, homeowners will like to specify that in the description. To the end, our idea was to do Natural Language Processing on the 'description' variable and cluster listings based on similarities in description. To be more specific, we wanted to use the TfidfVectorizer available in scikit learn to get a weighted term frequency matrix of the description. This would build a matrix with all the listings as rows and all the words that appear in all the description as different columns. The values would be the weighted frequency of the words for a particular listing. A general word like 'bed' or 'private' that appears in the description of most listing would be downeighted while a word like 'jacuzzi' that appears in few listings will have more weight. After we get this matrix, we calcualte the cosine similarity between all the listings based on the matrix. Further we planned to cluster the similar listings and use the clusters as dummy variables in the model. However due to lack of computational power, we weren't able to run the cosine similarity function on a matrix that had 17,000 rows and about 27,000 columns.

5 EVALUATION

The first model we tried was Linear Regression, for which we got an MSE of 0.132 and train and test accuracy of 0.67 and 0.69 respectively. This was below our goal so we tried other models to achieve higher accuracy. To effectively use Ridge Regression we needed to find the optimal alpha. To that end we ran a for loop (1 to 50) to find the alpha that maximizes R2 and minimizes the MSE of the model. We found the optimal alpha to be 41. However running ridge regression with the optimal didn't yield much better results than Linear Regression. There was no significant improvement in the accuracy or MSE. Moving on we tried Lasso Regression in the hope it would take out undesirble variables and give better results. But the accuracy score goes lower for Lasso. We get some improvement with Support Vector Regression as the trainig accuracy goes to 0.73 while test is at 0.70. The MSE is also the lowest yet at 0.127. This was the best we had so far and the first model we reached our goal of 70 percent accuracy with. However we thought we could further improve it with Ensemble Learning and feature selection. After finding the optimal parameters for Random Forest Regression we run the model. MSE is the lowest so far at 0.122. The training accuracy goes up significantly to 0.85 however the test accuracy sees a much more modest imrovement and is at 0.7157. This suggests some overfitting. To that end we try to do some feature selection to remove any redundant variables that might be causing the model to overfit. The optimal value for importance threshold removes some very low frequency variables like a super strict cancellation policy and rare property types eg. Castle. We also transform all the variables to the same scale. We fit the final model having removed redundant variables and scaling the ones to be used. The MSE for the model is 0.119, training and test accuracy are at 0.84 and 0.72 respectively. We run a 5 fold Cross Validation on the various variations of the Random Forest Model we have. To be specific we have Random Forest Regression without scaled data, one with scaled data, and one with Feature Selection. The models perform similarly with test accuuracies ranging from 0.67 to 0.71 for different random samples from the data.

Table 4: Test Accuracy and MSE by model

Model	Test Accuracy	MSE
Linear Regression	0.69	0.13
Ridge Regression	0.69	0.13
Lasso Regression	0.67	0.14
Support Vector Regression	0.70	0.127
Random Forest Regression	0.71	0.12
Random Forest Regression (Feature Selection)	0.72	0.119

Table 5: Cross Validation Scores for Random Forest models

Model	Mean Cross Validation Score
Unscaled	0.6931
Scaled	0.6928
Random Forest Regression (SelectFromModel)	0.6912

6 CONCLUSION

After comparing each model and do the evaluation for them, we choose the Random Forest Regression model as our final prediction model as it has the best ability to predict the airbnb prices in New York City according to the analysis we made before. -

When it comes to which variable that has more important effects on the airbnb housing price, we can see from the feature importance ranking in figure 8. The types of the houses and the number of rooms in the houses and even the review's conditions have the most important influences. These two parts are the direct conclusion we can make from our previous analysis. -

However, we can go further as applying our results into the reality is practical: the airbnb house owners can use our prediction model to set a reasonable price for their houses, and if they want to improve their prices they can also refer to the most important variables to adjust these parts in order to influence the prices. The airbnb customers can also use our prediction to distinguish whether the airbnb housing price is set higher or lower so that they can make orders rationally. The airbnb company can also make use of our prediction model when doing recommendations for housing owners or platform users. Therefore, our project will have a lot of real-life applications.

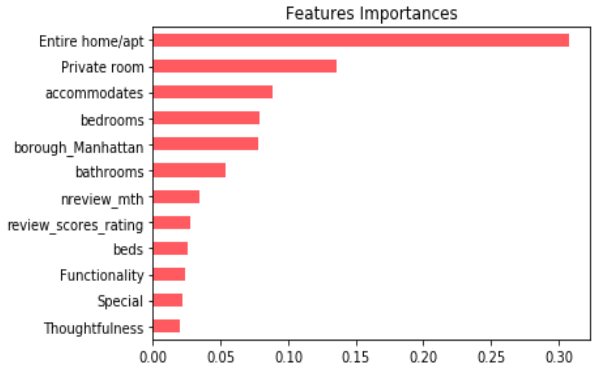

Figure 8: Top features ranking.

REFERENCES

[1] Juan L.Nicolaub Dan Wanga. 2016. Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on Airbnb.com. Retrieved May, 24, 2016 from https://www.sciencedirect.com/science/article/pii/S0278431916305618

[2] Davide Proserpio Georgios Zervas and John W. Byers. 2017. The Rise of the Sharing Economy: Estimating the Impact of Airbnb on the Hotel Industryt. Retrieved October, 2017 from http://journals.ama.org/doi/abs/10.1509/jmr.15.0204

[3] US government data. 2018. Neighborhood Names GIS. Retrieved February 3, 2018 from https://catalog.data.gov/dataset/neighborhood-names-gis

[4] The Airbnb Group. 2018. Amenities Do Matter: Airbnb Reveals Which Amenities Travelers Value Most. Retrieved August 28, 2018 from https://press.airbnb.com/amenities-do-matter-airbnb-reveals-which-amenities-guests-search-for-most/