AAE 722 Machine Learning 2019 Summer Final Project

Airbnb Price Prediction Via Machine Learning

Yuxuan Li

Zhirui Shi

Department of Agricultura and Applied Economics

University of Wisconsin, Madison

September 6, 2019

1.Introduction

The pricing of Airbnb listings, which is neither standardized comparing to hotels, nor authoritative compared to apartment, is complex to predict. Every listing has its unique features that affecting its price. Those features may interact in complex ways[1], which makes the prediction even harder with traditional method. In this report, we conduct a comparative analysis of machine learning methods for predicting prices of Airbnb listings.

1.1 Primary contributions

Our primary contributions are two-fold. First, we provide the accuracy of different machine learning methods in predicting Airbnb price. This accuracy is summarized by RMSE (Root Mean Squared Error). Second, we try to synthesize the Airbnb pricing with the field of machine learning. Airbnb needed to offer people a better way - an automated source of pricing information to help hosts come to a decision. Some teams already started building pricing tools in 2012 and have been working to make them better ever since.[2] We try to do the same thing applying the machine learning methods we learnt this summer. And provide ways to predict Airbnb price.

1.2 Hypothesis

We make 2 hypotheses about the price generation mechanism.

(1) The Airbnb-predicted price is the market equilibrium price. Airbnb uses its own algorithm to predict for hosts the optimized price and the probability of being booked under a range of prices. The prediction is based on the existing booking record, which reflects the market equilibrium to some extent.

---

[1] Aerosolve: Machine learning for humans
https://medium.com/airbnb-engineering/aerosolve-machine-learning-for-humans-55efcf602665#9040
[2] The Secret of Airbnb's Pricing Algorithm
https://www.infoq.cn/article/decryption-airbnb-pricing-algorithm

(2) The expectation of the host-posted price is the market equilibrium price. Airbnb hosts can adjust the price based on probabilities provided by the platform. One may lower the price to ensure the chance of being booked; or may increase the price to screening for better quality guests. Thus, the host-posted price can be decomposed into two parts: the Airbnb-predicted equilibrium price and the host will.

$$P_{host-posted} = P_{Airbnb-predicted} + \Delta P_{host\ will} \approx P_{equilibrium} + \epsilon$$

We assume the host will is normally distributed with large sample size. Therefore, the expectation of the host-posted price is the market equilibrium price, which is the target variable in our study. In the following, the price without specification refers to the host-posted price.

1.3 Why apply machine learning methods to Airbnb pricing?

Gu, S., Kelly, B., and Xiu, D. (2019)[3] indicated the situation that machine learning methods are attractive: (1) when the goal is to predict, (2) when the set of candidate predictors is large, and (3) when how predictors enter the model is unclear.

In our predicting question, although the number of candidate predictors is not that much, half of those are categorical variables. For example, the property type has 35 categories, and the listed amenities has 197 types, which are not easy to handle in traditional methods. In the meanwhile, the way of variables entering the model is not obvious. We will discuss this further in section 2.1.

2. Data

The dataset we use in this report contains price and relative features of 36,000 Airbnb listings in 6 major U.S. cities. This dataset contains all the information that a guest can find on

[3] Gu, S., Kelly, B. and Xiu, D., Empirical asset pricing via machine learning. NBER working paper series. Working Paper 25398. http://www.nber.org/papers/w25398

the Airbnb platform, including the geographical information, properties and amenities, booking policy, reviews and host profile. Details are listed in the Table 1.

Table 1 Variables in the Airbnb dataset

Geographical information	City Neighborhood Longitude and Latitude Zip code
Properties	Accommodates Number of beds Number of bedrooms Number of bathrooms Property type Bed type
Amenities	Amenities listed by host
Booking policy	Cancellation policy Cleaning fee (logical) Instant bookable
Reviews	First review date Last review date Number of reviews Review score rating
Host profile	Profile picture (logical) Identity verified (logical) Host since date Response rate

Data Source: AAE 722

2.1 Data exploration

The six cites in the Airbnb data are Boston, Chicago, Washington DC, Los Angeles, New York City and San Francisco. Figure 1 shows that the price distribution are similar in those cities. There is no significant difference or pattern in prices between the West Coast, the East Coast and the Central Region. Figure 2 shows the geographical distribution on city level. Note that only one twentieth of observations in each city are shown on the map, avoiding overlapping of data points. Within each city, the listings are typically located along the main traffic, yet the price distribution has no obvious pattern. High prices appear in cities or suburbs, inland or coastal.

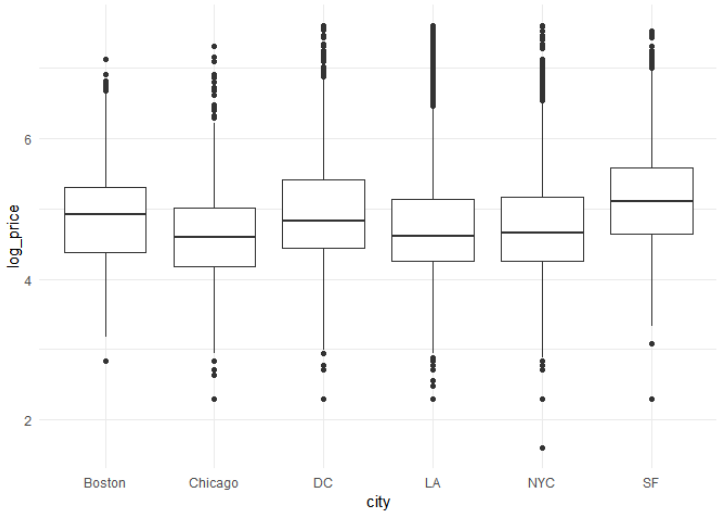

Figure 1 Price distributions across cities

Note: No significant difference or distribution pattern across the West Coast, the East Coast and the Central Region. Take the logarithm of the price to curve down the scale.

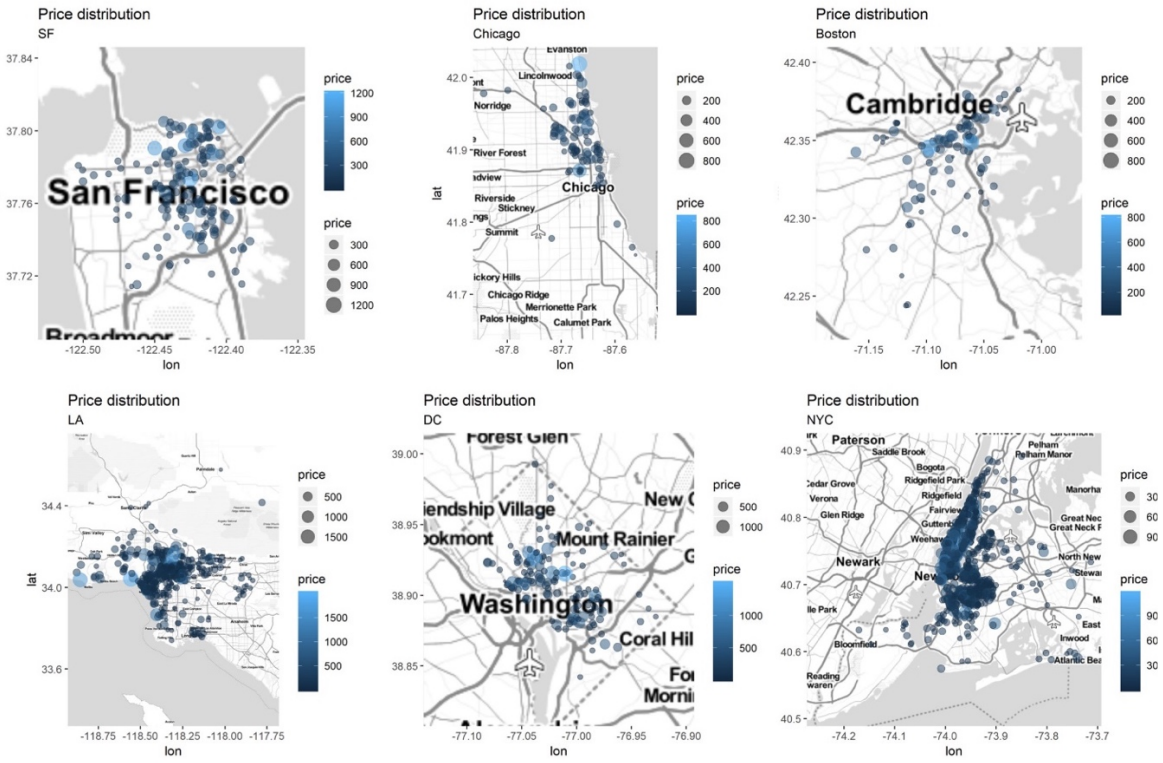

Figure 2 Price distribution within each city

Note: In order to avoid overlapping of data points, we randomly sampled one-twentieth of observations within each city to draw bubble map. The distribution of price is geographically random. Larger, lighter-colored bubbles present listings with higher price. Note that the scales of legend are different in each city.

Indeed, location is not the only factor of price. Downtown dorms may be cheaper than suburban villas. Other properties matter. Figure 3 shows the positive correlation of the price and the number of properties, while the variety is large especially where the property numbers are small, which suggests that after controlling for the size of listings, other affecting factors exits.

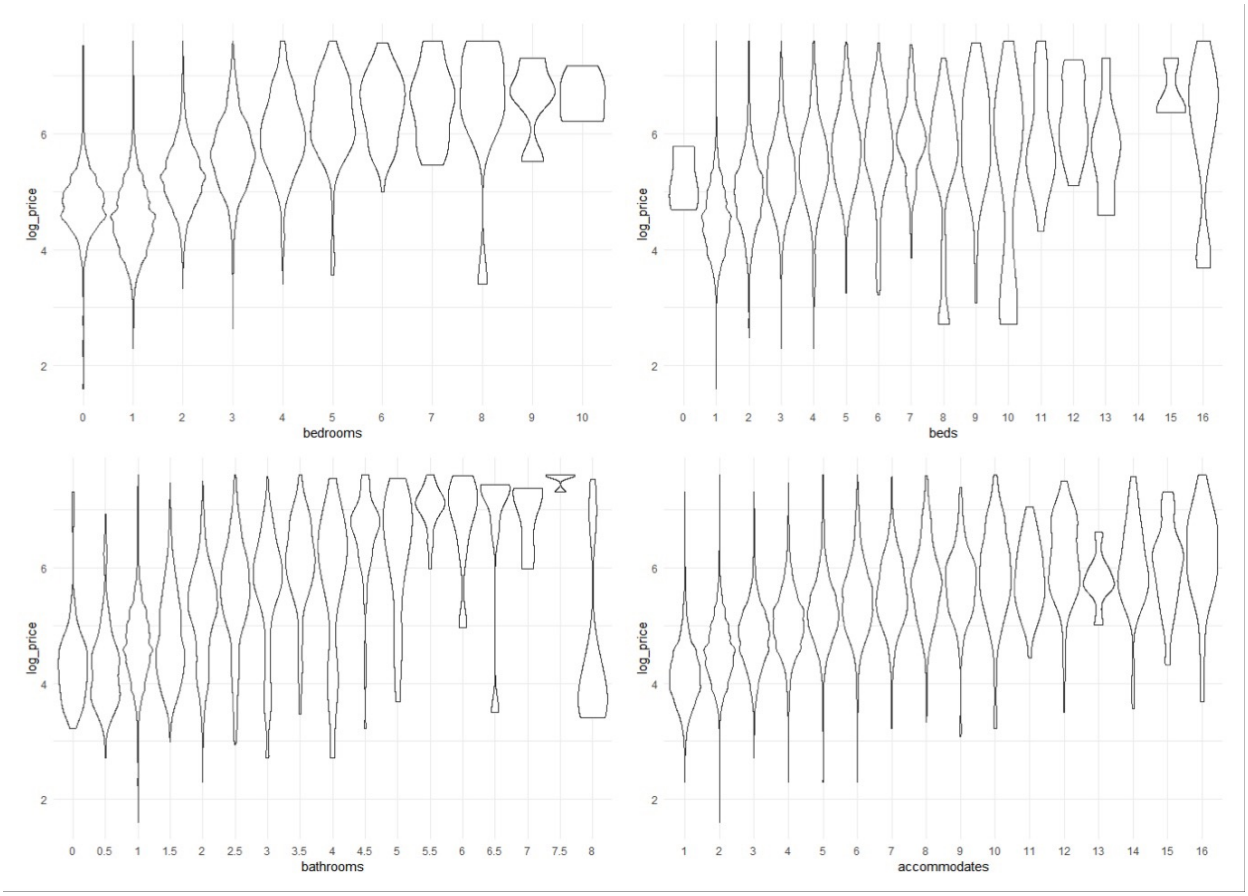

Figure 3 Price distributions against properties

Note: Price distributions against the number of bedrooms, beds, bathrooms and accommodates. Take the logarithm of the price to curve down the scale. Observations contains missing value are removed (detailed NA process discussed in section 2.2).

Considering the effect of amenities, we use word cloud to show the frequency at which the amenities appear in the host description, as shown in Figure 4. From highest to lowest, the top 10 listed are wireless Internet (34428), kitchen (32553), heating (32191), smoke detector (29841),

essentials (29632), air conditioning (26623), hangers (23282), carbon monoxide detector (22874), shampoo (22170), and laptop friendly workspace (21184). Some of them are rigid needs of users, such as access to the Internet, air conditioning and shampoo. Some are basic facilities, which may not affect user choice but are necessary to list, such as smoke detector, carbon monoxide detector. Others like essentials and laptop friendly workspace, do not have clear definition, yet do affect impression of the listing.

Figure 4 Word Cloud of top listed amenities

For further understanding the impact of the amenities, we used box plots to show the price distribution between groups with and without certain amenity. The “certain” amenities are selected based on the frequency of listed plus some intuition. Except for the first aid kit, the price of the listings with certain amenity in all groups is higher than those without that amenity.

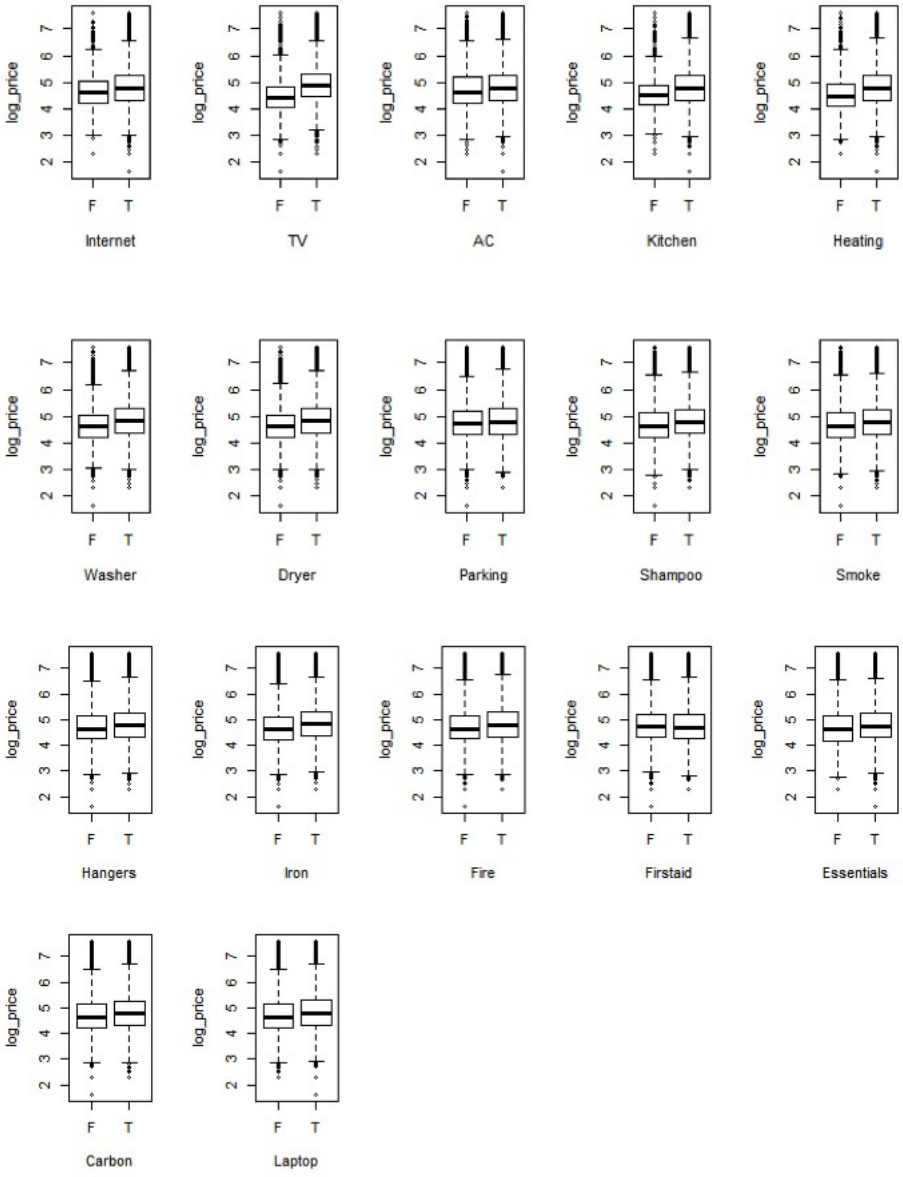

Figure 5 Price distributions against amenities

Note: In general, the price of listings with certain amenity is higher than those without that amenity.

Airbnb users usually refer to reviews and the host profile when selecting a listing. Whereas in Figure 6, we do not see the clear correlations of price with reviews and host profile. Whereas Figure 7 indicates that price differences exist between those without review and those with at least one review, which suggests that hosts tend to lower the price after the first booking out.

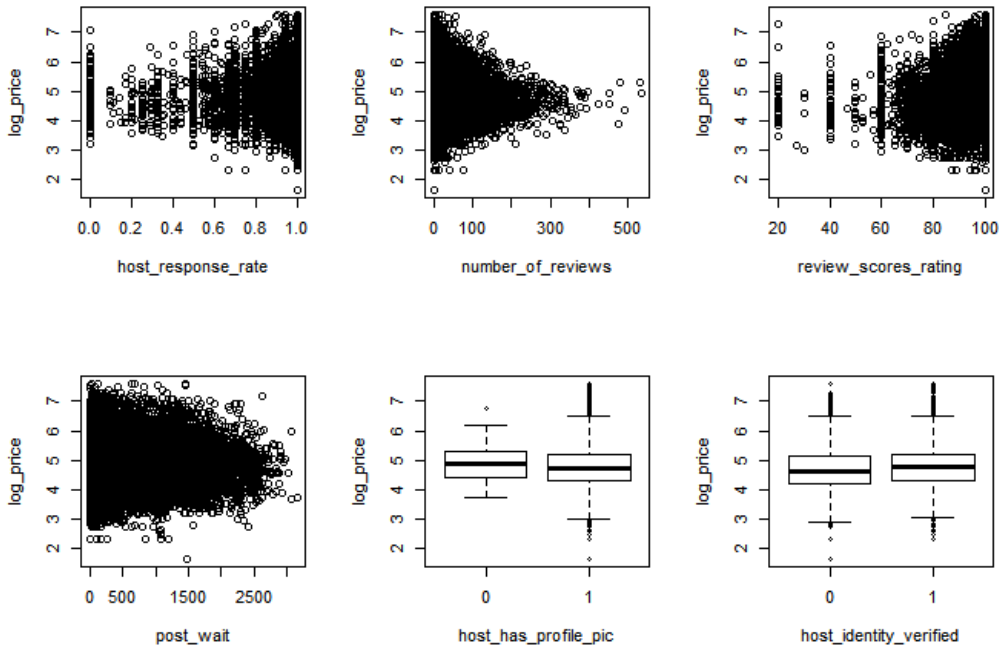

Figure 6 Price distributions against review and host profile

Note: As discussed in section 2.2, post_wait is the day length from host register date to the first review date, measuring the probability of a listing being booked out. No clear correlations of price with reviews and host profile can be observed from graphs.

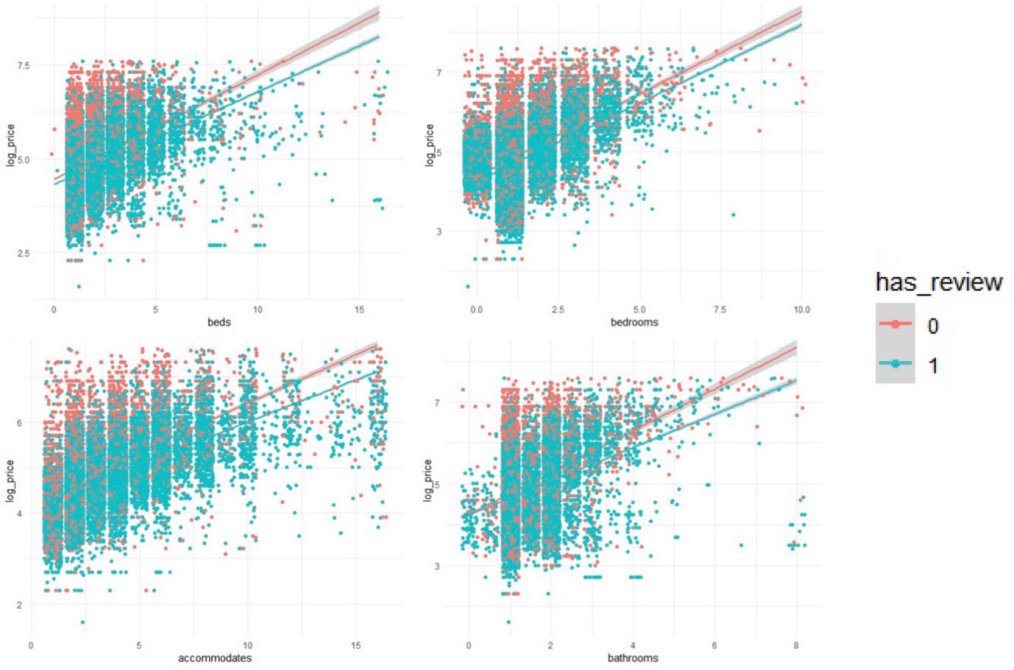

Figure 7 Price comparison of listings with/without reviews

Note: Red for listings with no review; Green for listings with at least one review. The solid lines are simple linear regressions of log price and numbers of beds, bedrooms, accommodates and bathrooms, respectively. Price differences exist between those without review and those with at least one review.

Throughout the statistical description, it is hard to say that prices of listings has a definite relationship with any certain feature, nor say which group of features is the determinant of price. In the meanwhile, it is nearly impossible to describe some subjective factors in the data, such as room pictures, host profile pictures, etc. The price of Airbnb listings is complex. It cannot be determined by a simple, sparse model. That is partly the reason why we want apply machine learning method to predict Airbnb listings price.

2.2 Data Processing

2.2.1 Variable transferring

In the dataset, we have 3 date-type variables: host since (the register date of the host), first review date and last review date. Those dates contain information about how popular the property is, how much experience the host has, how trustworthy the reviews are. Nevertheless, directly apply regression method on date-type data can be meaningless. Thus, we construct a new variable “post wait” as the date length between host since and first review, meaning to measure the probability of a listing booking out. Until the first guest post their review, one list is chosen by user without any review information. During this period, the better the quality of the listing property itself, the sooner the house can be booked out.

For the amenities of listing, we transfer it into a series of dummy variable, as shown in Figure 5. We also create a variable “number of amenities”, counting the total number of listed amenities. The distribution of new variables “post wait” and “number of amenities” are shown in Figure 8.

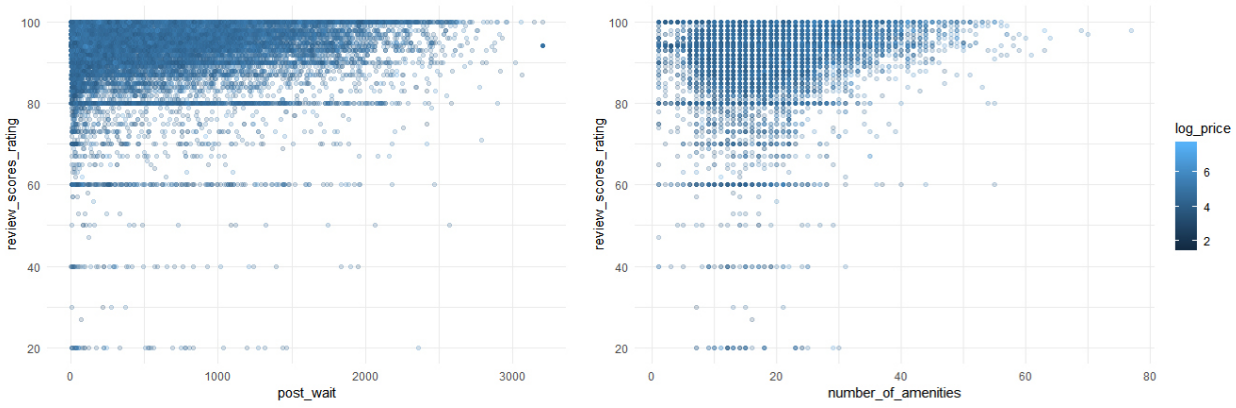

Figure 8 Distribution of new variables

2.2.2 Dealing with missing values

We have 8,333 observations containing missing values out of 36,000. We define missing values into 2 types: random and systematical missing values. Random missing values are rare, randomly happened, including bathrooms (95), bedrooms (41), beds (59), host has profile pic (91), host identity verified (91), and host since (91). Systematical missing values are more common, typically because of no review, which includes: first review (7717), last review (7696), and review scores rating (8143).

We simply dropped random NA's. For systematical one's, we assign sample mean to review scores rating, and sample max to post wait, which is the day-difference between host since and first review. Then remove first review and last review, since we won't use those variables in further study.

2.2.3 Data splitting

We randomly divide three-quarters of the data as training data, and the rest one quarter as testing data. The splitting is the same throughout 5 prediction methods for comparison.

3. Model and Methods

3.1 Linear regression

We choose 19 variables from data set to follow the forward stepwise selection procedure. As there are 4 qualitative variables, 34 dummy variables are added in the procedure. Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.[4]

We use cross-validation to estimate the errors among 54 different models. To be more specific, we randomly divide the set of observations into 10 folds. The first fold is treated as validation set, and the method is fit on the remaining 9 folds. The mean squared error, MSE_1, is then computed on the observations in the held-out fold[5]. This procedure is repeated 10 times; each time, a different group of observations is treated as a validation set. This process results in k estimates of the test error. The 10-fold CV estimate is computed by averaging these values,

$$CV = \frac{1}{10}\sum_{i=1}^{10} RMSE_{10}$$

We use the same cross-validation RMSE to evaluate among the 5 methods.

The MSE computing by cross-validation for each model are shown in the Figure 9.

[4] P207 G. James et al., An Introduction to Statistical Learning: with Applications in R,. Springer Texts in Statistics, DOI 10.1007/978-1-4614-7138-7 2,.

[5] P181 G. James et al., An Introduction to Statistical Learning: with Applications in R,. Springer Texts in Statistics, DOI 10.1007/978-1-4614-7138-7 2,.

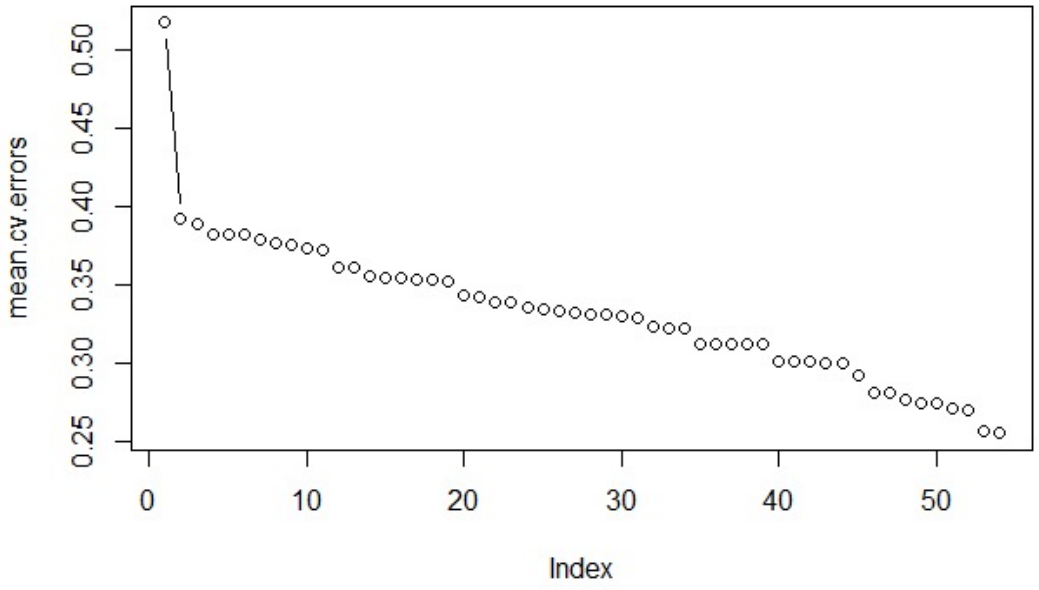

Figure 9 Cross-validation MSE of models containing different variables

As we can see, the MSEs are decreasing when there are more predictors in the model, the mean cross-validation error is smallest when all 54.

Then we use testing data to compute the MSE, which equals to 0.48579.

3.2 LASSO Regression

In order to trade bias off against variance, we use LASSO to select variables:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{54}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{54}|\beta_j|$$

λ is the regularization term for LASSO. Smaller λ stands for the lots of regularization, and larger λ allows more variables enter the model. As we can see the figure below.

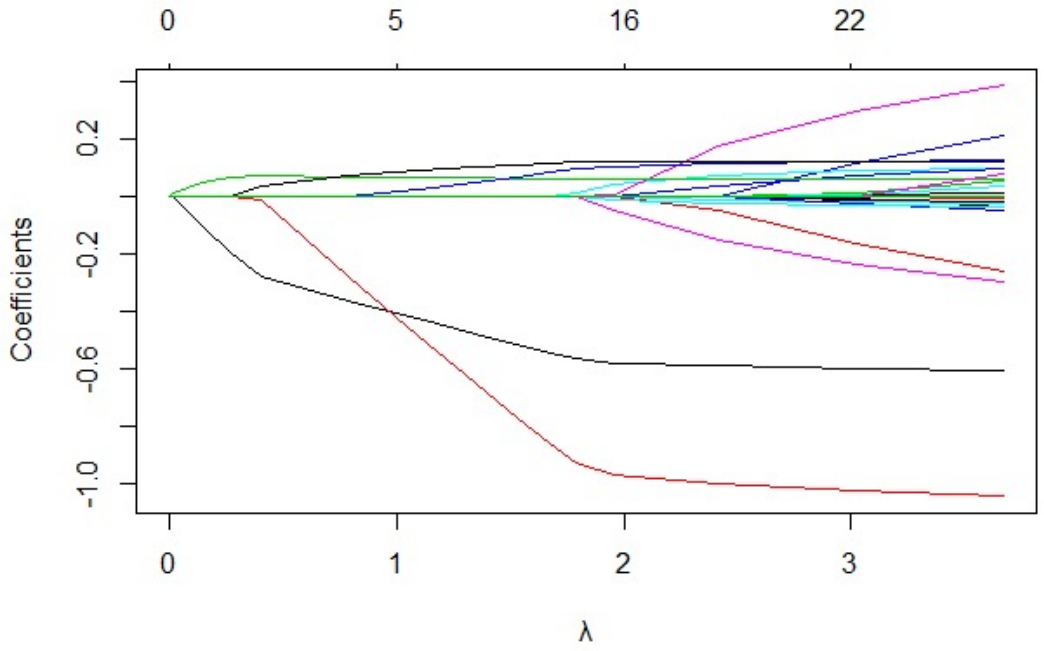

Figure 10 The relationship between λ and coefficients

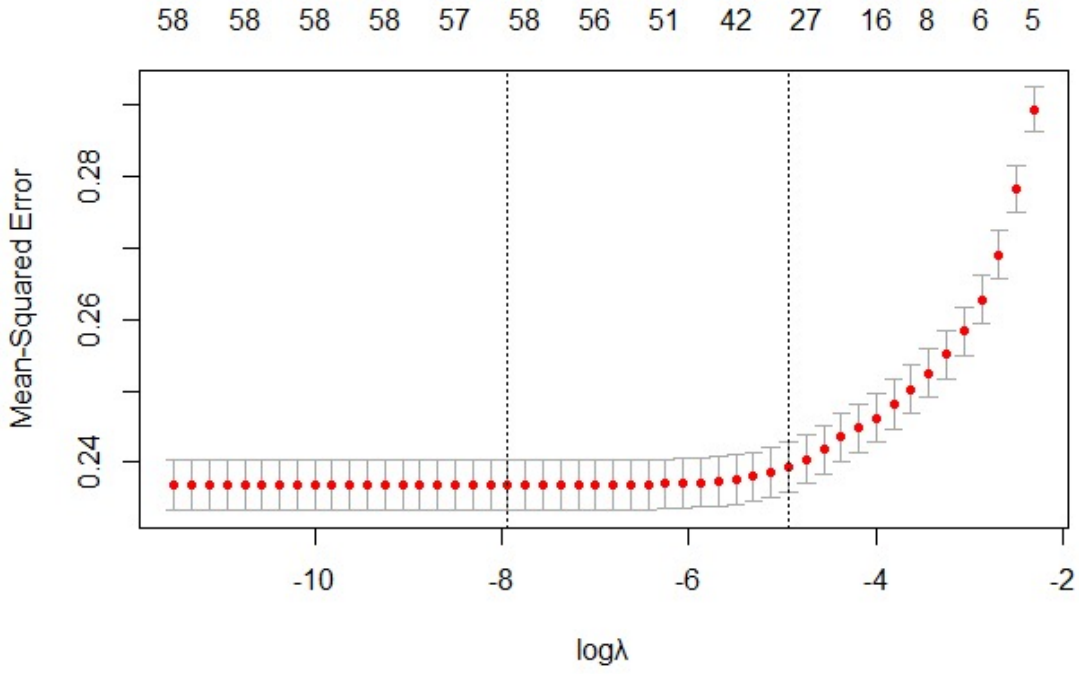

Figure 11 The relationship between log λ and MSE

As we can see in Figure 11, when $\lambda = 1.676833 \times 10^{-4}$, cross-validation RMSE is the smallest equaling to 0.48528, which is similar to the RMSE computing before.

3.3 KNN

We apply KNN method to predict price. The most primitive price prediction algorithm of Airbnb is to use the idea of KNN, which predicts the most appropriate price of a new listing from a set of geographically close listings. Since KNN-regression users Euclidean distance to define the neighbor, only quantitative predictors will be used. We add in all feasible quantitative predictors by 3 steps, as shown in Table 2. The first group starts with geographical information, as the primitive Airbnb algorithm, and then add in other predictors. Within each group, we conduct cross-validation to determine the best number of neighbors—k.

Table 2 Training and testing RMSE of KNN regression

	Group 1 Longitude, latitude		**Group 2** Group 1 + beds, bedrooms, accommodates, bathrooms, amenities		**Group 3** Group 2 + review scores rating, number of reviews, host response rate	
k	Train RMSE	Test RMSE	Train RMSE	Test RMSE	Train RMSE	Test RMSE
1	0.00	1.02	0.00	1.02	0.00	1.02
5	0.54	0.86	0.42	0.92	0.44	0.91
10	0.58	0.83	0.46	0.90	0.47	0.89
25	0.60	0.82	0.49	0.88	0.50	0.87
50	0.61	0.81	0.50	0.87	0.52	0.86
250	0.63	0.79	0.53	0.84	0.54	0.84
500	0.64	0.78	0.55	0.83	0.56	0.82
840	0.65	0.77	0.56	0.81	0.57	0.80
1,000	0.65	0.77	0.57	0.80	0.58	0.80
3,000	0.67	0.75	0.61	0.77	0.62	0.77
5,000	0.69	0.74	0.64	0.76	0.65	0.75
10,000	0.72	0.72	0.70	0.73	0.70	0.73
20,000	0.72	0.72	0.71	0.72	0.71	0.72

Note: All predictors except for longitude and latitude are scaled before training model. Data are randomly divided into training and testing datasets. The best k is searched from a series from 1 to 20,000. Note that in training dataset, the numbers of listings in each city are in the range of 1,264 (Boston) to 11634 (New York City).

3.4 Regression Tree

We apply regression tree method to predict prices of Airbnb listings. We grow a tree using all the feasible predictors in training data and conduct a 10-fold cross-validation to prune the

tree. The initial complexity parameter is set to 0.0002. As shown in Figure XX, the unpruned tree has 235 splits. The best prune tree with minimized CV-RMSE has 166 splits.

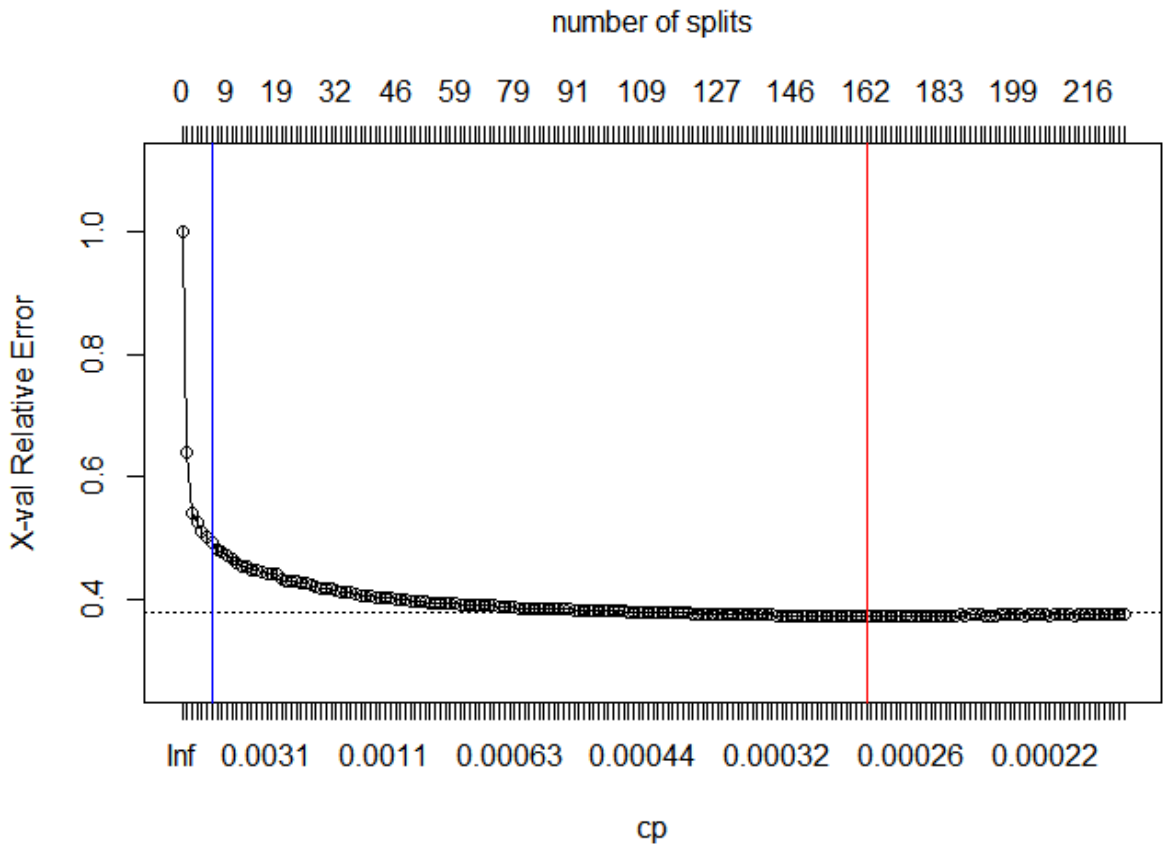

Figure 12 The cross-validation RMSE of regression tree

Note: The lower x-axis is the complexity parameter associated with each pruned tree. The upper x-axis is the number of splits. The red line indicates the best pruned tree with minimized cross-validation RMSE, which has 166 splits. The blue line indicates a smaller tree with 7 splits. The horizontal line is the 1SE above the minimum of the curve.

In Figure 12 we report the graph of the best tree. Since the best tree is too large to interpret, we also report a smaller tree with 7 splits, which is the trunk of the best tree. The most important factor of price is room type. Entire houses/apartments usually have higher prices than private rooms, and a private rooms have higher price than shared rooms. Location also affect price largely. From the graph, listings in Chicago, LA and San Francisco are more likely claiming a higher price. The training and testing RMSE of two models are reported in Table 3.

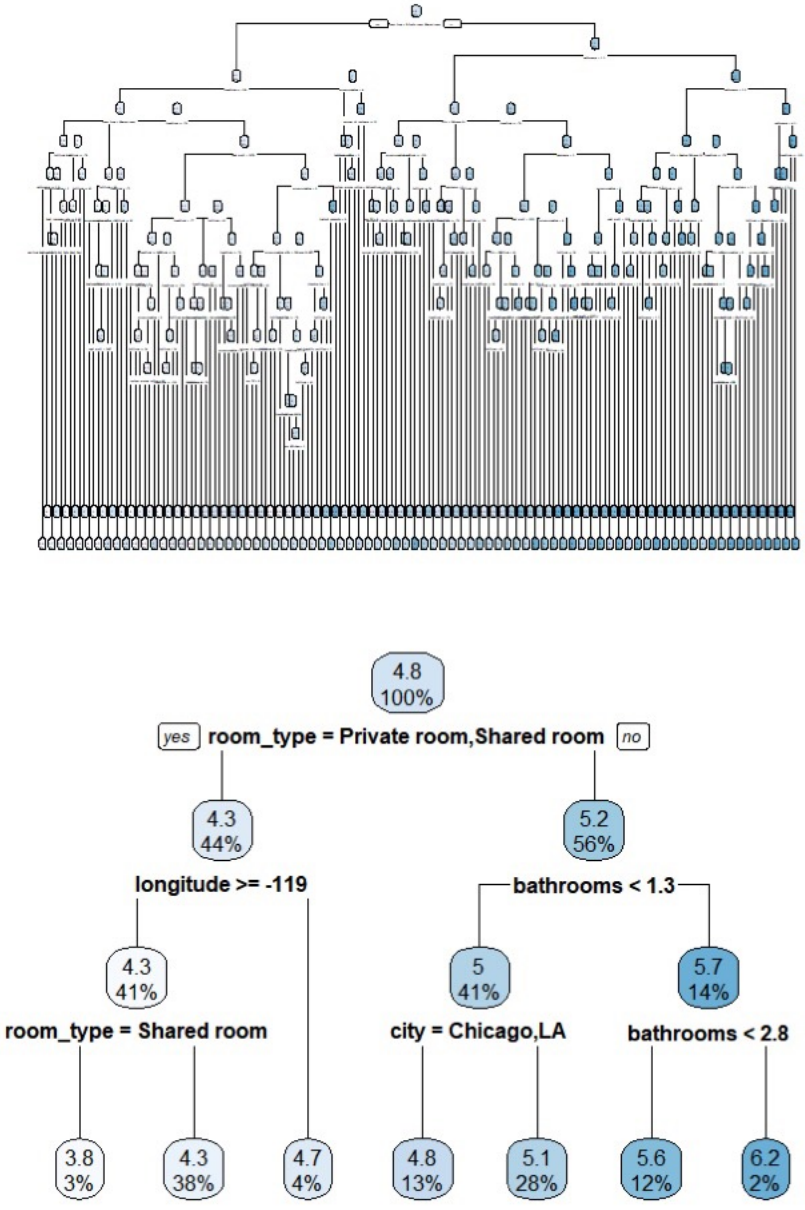

Figure 13 Pruned trees

Note: The upper graph is the best pruned tree with lowest CV-RMSE. The lower graph shows the trunk of the best tree. Longitude -119 is roughly the eastern edge of the California state, bounding out SF and LA in our data.

Table 3 Training and testing RMSE of pruned trees

	Best Tree (166 splits)	7-split Tree
Training RMSE	0.4117	0.5026
Testing RMSE	0.4456	0.5045

Note: Data are randomly divided into training and testing datasets. The best tree is pruned using 10-fold cross-validation with training data.

3.5 Neural Networks

Arguably the most powerful modeling device in machine learning, neural networks have theoretical underpinnings as "universal approximators" for any smooth predictive association[6]. We use 14 variables as predictors to contribute neural nets. And we begin with 1 hidden layer, 3 hidden units.

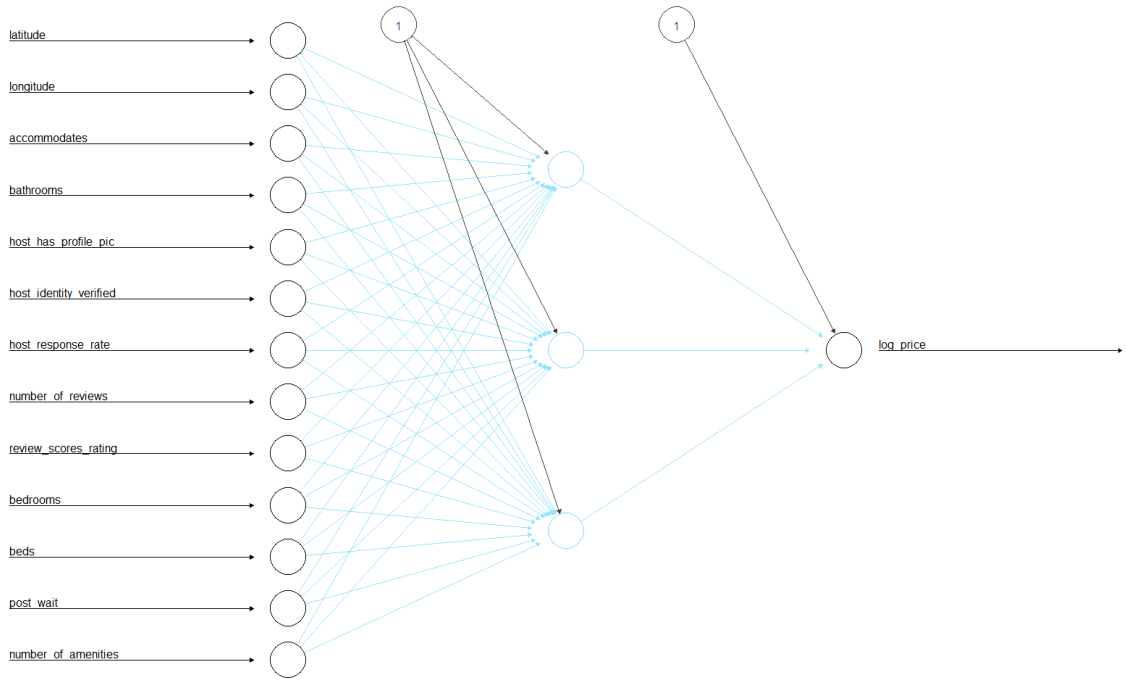

Figure 14 Neural Net with single hidden layer and 3 hidden units

[6] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White, 1989, Multilayer feedforward networks are universal approximators, Neural networks 2, 359–366.

The relationship between real price and predicted price estimated by test data is shown in Figure 15.

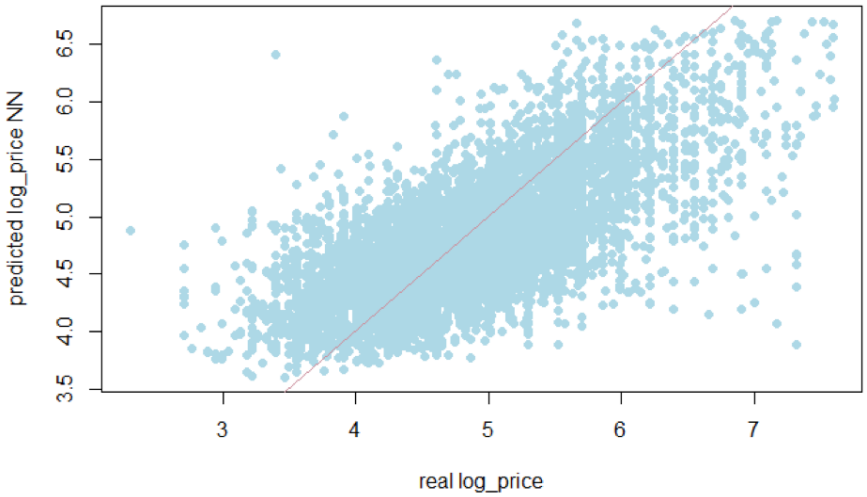

Figure 15 The predicting accuracy of Neural Nets with 1 hidden layer and 3 hidden units of each layer

The testing RMSE of this model is 0.52999, which is little bit greater than the RMSE of the linear regression model. Then we try to complicate the neural net to reduce the RMSE.

We add 2 more hidden units into the model, as shown in Figure 16.

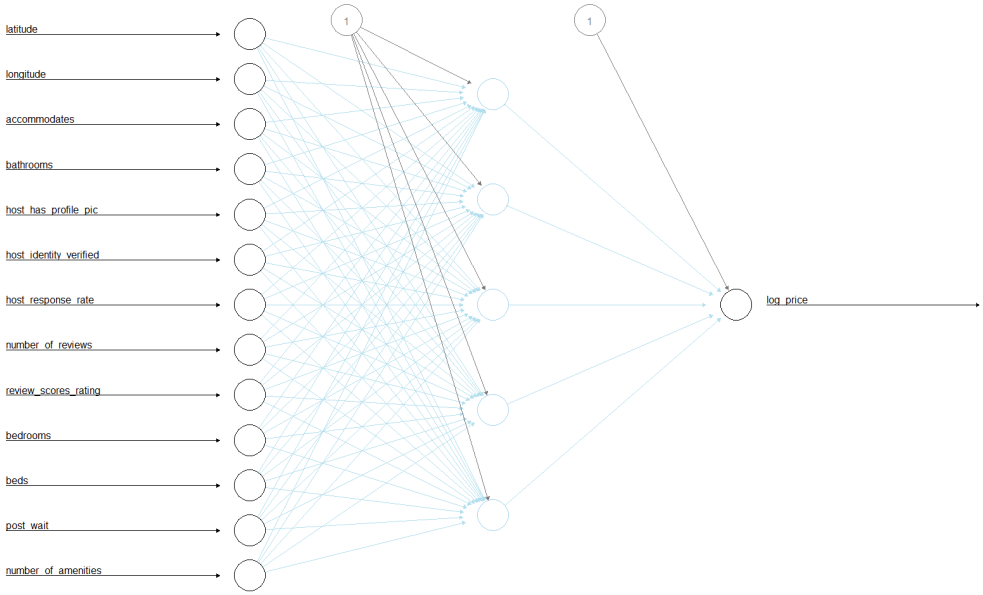

Figure 16 Neural Net with single hidden layer and 5 hidden units

Now we have a neural net of 1 hidden layer with 5 hidden units. Similarly, we draw a figure indicating the relationship between real price and predicted price estimated by test data.

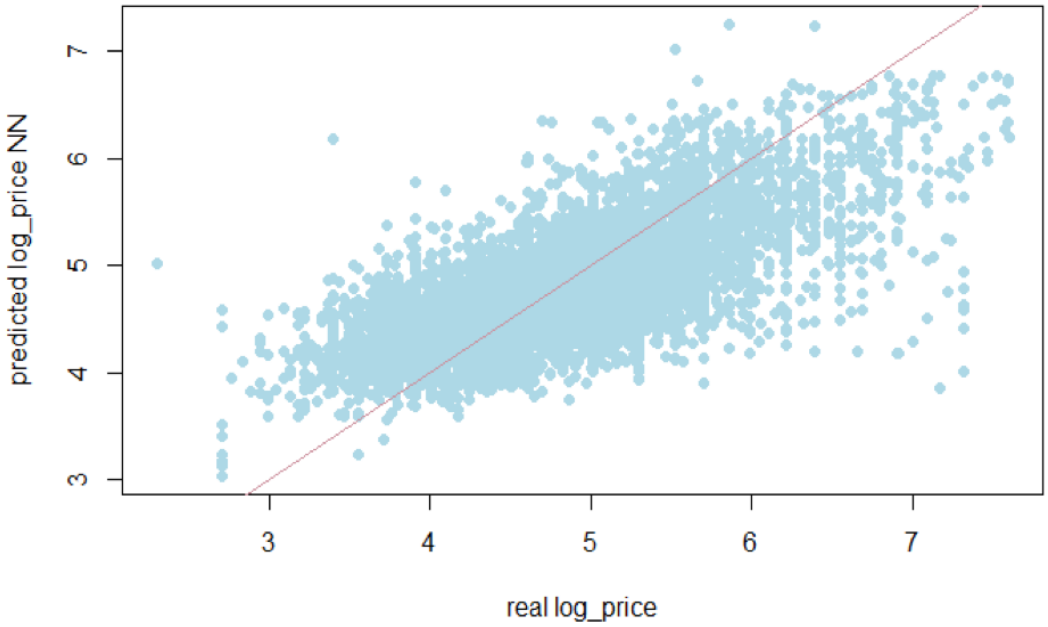

Figure 17 The predicting accuracy of Neural Net with single hidden layer and 5 hidden units

The testing RMSE of this model is 0.51239, which is smaller than the first model.

Then we add 1 more hidden layer with 5 hidden units.

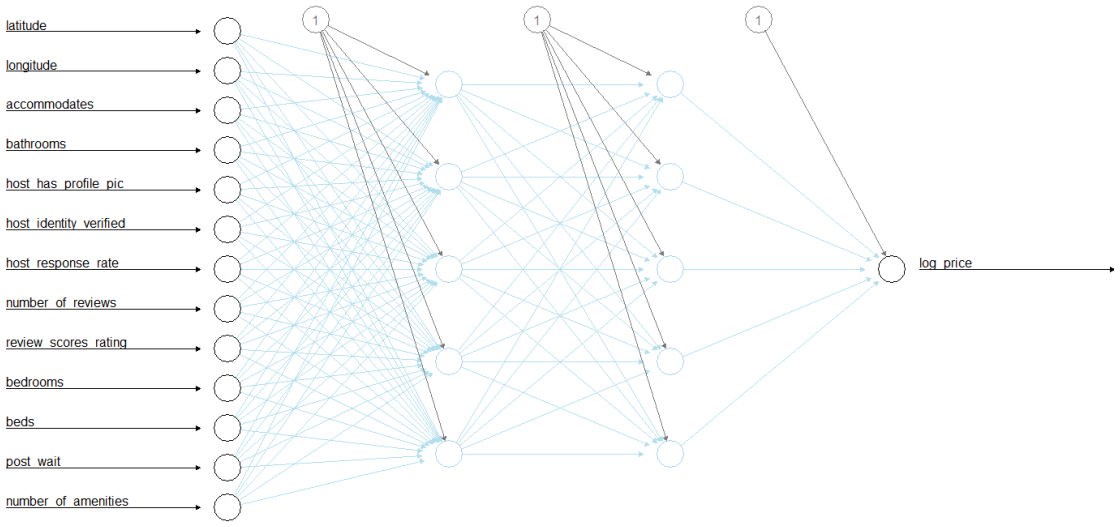

Figure 18 The Neural Net with 2 hidden layers and 5 hidden units in each layer

The relationship between real price and predicted price estimated by test data is shown in Figure 19.

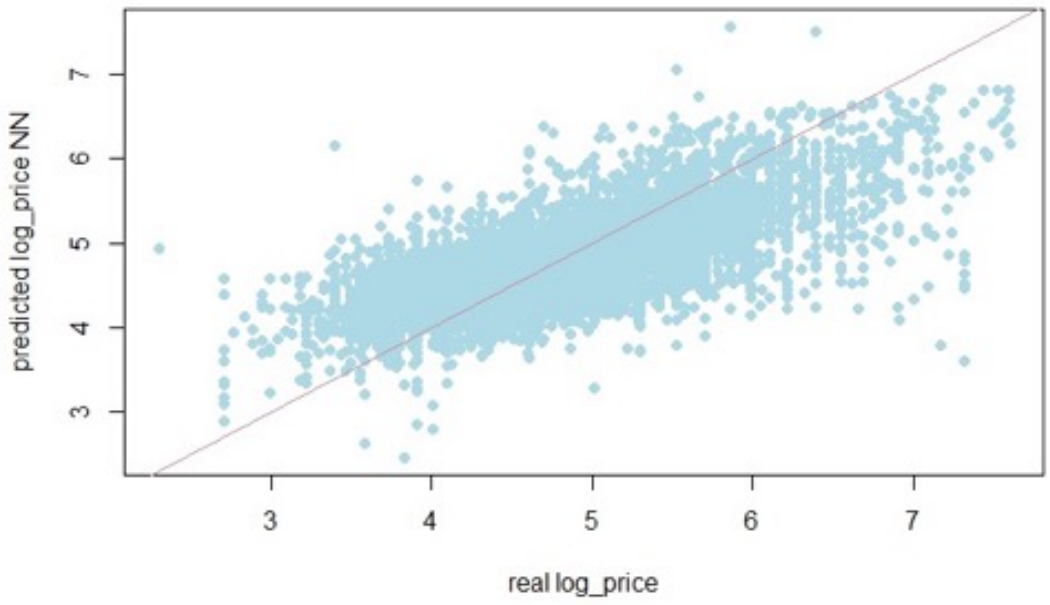

Figure 19 The predicting accuracy of Neural Net with 2 hidden layers and 5 hidden units in each layer

The RMSE of this model – 2 hidden layers with 5 hidden units of each is 0.51414, which is

a little bit greater than the single layer model with 5 hidden units, but still smaller than the first model.

We can also continue complicating the neural nets, but it takes too long to fit the model. For example, the model with 3 hidden layers and 5 units of each layer takes 3 hours. Therefore, we stop there.

3.5 Comparison

We use the same testing data set to compute the RMSE of the best model for each method. The mean value of *log_price* is 4.77 in testing data set. We use the formula below to roughly calculate the error rate of each method.

$$error\ rate = \frac{RMSE}{\text{log\_price}}$$

where, $RMSE = \sqrt{\left(\widehat{log\_price} - log\_price\right)^2}$

Table 3 Methods Evaluation

Method	RMSE	Error Rate/%
Linear Regression	0.4858	10.18
LASSO	0.4853	10.17
KNN	0.72	15.09
Regression Tree	0.4456	9.34
Neural Networks	0.5124	10.72

As we can see, Regression preforms best. The error rate of linear regression, LASSO and neural networks is approximate 10%. The error rate of KNN is 15.09% which is relatively greater, but still not huge.

4. Discussion

From the results of RMSE above, we can find that machine learning methods perform well in prediction. Although linear regression is even better than neural networks and KNN, we still need machine learning technology to select predictor when the set of candidate predictors is large. In this case, we use forward stepwise select subset in linear regression, and find that the more variables in the model, the less RMSE estimating by test data set, which means all the variables in the data set are related with the price.

Therefore, the RMSE results in this case are quite related with the number of variables we use. For linear regression, LASSO and regression tree, with the aim of picking up all significant variables, we add all feasible variables into the model. Technically, we can use the same subset of variables to run the neural networks model. But due to the high complexity of neural net, the compute process becomes considerably slow. We believe that the accuracy of neural networks will become better if we keep adding input variables, hidden layers and hidden units.

5. Conclusion

Using the empirical context of Airbnb price prediction, we perform a comparative analysis of methods in the machine learning. We find that machine learning methods can help improve our empirical understanding of Airbnb prices. Regression trees is the best performing methods. We can use the existing booking record to estimate models, and then use these models to predict the Airbnb price, as the prediction errors are relatively small which are no more than 16%. When the set of candidate predictors is large, we can use machine learning to select feasible subset of predictors.