Predicting Airbnb Prices in Six U.S. Major Cities

AAE 722 Final Project

Weihang Wang

Shuai Tan

December 2022

1 Introduction

Airbnb is an internet marketplace for short-term home and apartment rentals. According to Airbnb's latest data, it has over six million listings, covering more than 100,000 cities and towns and in more than 220 countries worldwide(Airbnb, 2022). At this home-sharing platform, homeowners and renters (hosts) can rent out their properties (listings), and more importantly, they need to set their own prices for their listings. This could be challenging, especially for listings in big cities due to competition, and even small differences in prices can make a big difference. Luckily, machine learning can help us solve this problem by making highly accurate predictions based on historical data. In this project, we conduct a comparative analysis with the Airbnb dataset to predict the price of listings in six major U.S. cities (Boston, Chicago, DC, LA, NYC, and SF). Specifically, we applied backward stepwise regression, random forest, and neural network, respectively, to find the most important predictors and the best model to predict Airbnb prices.

Machine learning has several advantages over traditional methods of forecasting. First, machine learning can identify complex patterns and make highly accurate predictions. In our

case, the correlation between listing prices and potential predictors can be rather complex and hard to identify by traditional methods. In the following sections, we will show that by applying random forest, we could get pretty good predictions. Secondly, machine learning can deal with large-scale and high-dimensional data more efficiently and effectively. Thirdly, by training machine learning models, we could find the best model with an optimal balance of bias and variance to avoid overfit and underfit. Moreover, machine learning can also adapt to changes in the data set, whereas traditional methods can become less accurate over time.

2 Data

The initial data consists of 36,000 observations with 26 variables. The output variable is the log of price (*log_price*), and the rest variables are potential predictors, including listing characteristics, geographic information, and review information. In the data cleaning step, variables including *first_review, host_response_rate, last_review*, and *neighborhood* are deleted since more than 20 percent of observations have missing values in these variables. Although the variable *review_scores_rating* contains more than 20 percent missing values, it is not excluded since it might be an important predictor[1]. The variable *zip_code* is deleted because the data set already contains specific latitude and longitude information for each listing. To simplify the prediction process, the variable *amenities* and *host_since* are also dropped, but including the most common amenities and when did the host register might improve the accuracy of predictions. After removing rows with missing values, the final sample consists of 27,667 observations.

There are 32 property types in the data set, 8 types (*Apartment, House, Condominium,*

[1]In the presentation, we exclude this variable to maintain a comparable size with other groups. This time we keep this variable in our sample, and as the following sections show, all predictions using the same three methods improved, and still, the random forest gave us the best performance.

Townhouse, Loft, Bed Breakfast, Guesthouse, Bungalow) and the type "other" have more than 100 observations. Therefore, we classify types with less than 100 observations altogether and create a new variable named "others". Moreover, we convert factor variables including *room_type, bed_type, cancellation_policy*, and *city* as well as logical variables (*cleaning_fee, host_has_profile_pic, host_identity_verified, instant_bookable*) into dummy variables accordingly. After cleaning the data, the final sample consists of one output variable and 34 potential predictors.

Table 1 presents the summary statistics of selected variables. The mean of the log of price is 4.783, and the distribution of the log of price is shown in Figure 1. The most common property type is *apartment*, which accounts for about two third of listings. Table 1 also shows that the number of reviews and the review score rating both have large standard deviations. Figure 2 shows the listing by room types across cities. NYC and LA have the most listings and the most common room type are *entire home/Apt* and *private room*. Table 2 presents the mean of the log of price by room type and property type across the six major cities. *Entire home/Apt* and *condo* are the most expensive room type and property types, respectively, and the average prices in SF and DC are higher than in other cities.

3 Methods

We divide our sample into training and test sets by the ratio of 3:1 to estimate the model subject to a specific set of tuning parameter values and evaluate a method's predictive performance, specifically. We apply backward stepwise regression, random forest, and neural network using the same training and test set.

3.1 OLS with Backward Stepwise Selection

We start our prediction with the multiple linear predictive regression model estimated via ordinary least squares (OLS) with backward stepwise selection. This model does not allow for nonlinear effects or interactions between predictors. Stepwise regression is a way of selecting important variables to get a simple and easily interpretable model. Backward stepwise regression begins with a full model and at each step gradually eliminates variables from the regression model to find a reduced model that best explains the data. Therefore, our initial model includes all 34 potential predictors. We choose stepwise selection over best subset selection because the latter one may suffer from computational limitations when p is large. We did not use the forward stepwise selection because it is not guaranteed to find the best possible model, and the number of our predictors is not larger than the sample size. Therefore, the backward stepwise selection is more suitable in our case.

We fit the model using 5×5-fold cross-validation. The minimal root mean square error (RMSE) for the cross-validation is 0.4116133, given by an optimal model which includes 28 predictors. Figure 3 shows the RMSE changes versus the number of predictors in the model. Applying the optimal model to our test set, the RMSE we get is 0.4104787.

3.2 Random Forest

Unlike linear models, regression tree is a nonparametric method that creates a binary tree by recursively splitting the data on the predictor values. Moreover, regression tree ensembles to model and visualize interaction effect(Schiltz, Masci, Agasisti, & Horn, 2018). Trees for individual bootstrap samples tend to be deep and overfit, making their individual predictions inefficiently variable. Random forest combines forecasts from many different trees, and its most significant advantage is it decorrelates trees by considering only a randomly drawn subset of

predictors for splitting at each potential branch. This lowers the average correlation among predictions to further improve the variance reduction relative to standard bagging.

We use the grid search to find the optimal model, in which all possible combinations of given discrete parameter spaces are evaluated(Probst, Wright, & Boulesteix, 2019). Since we have 34 potential predictors, we set the range of the number of variables randomly sampled to 8 to 20 and the range of minimal node size to 5 to 12. These tuning parameters are optimized via 5-fold cross-validation. Graph 4 shows the cross-validation RMSE with different numbers of randomly selected predictors, and the minimal RMSE is given by the model of 16 variables randomly sampled and 10 minimal node sizes. Fitting the optimal model, the test set RMSE is 0.34367.

Graph 5 shows the top 20 important variables. Basically, the higher a variable is on the graph, the more important it is determined to be. For example, the mean squared error would increase by more than 100 percent if we were to exclude longitude from the model. According to this graph, the most important predictors include the listing's location, the number of bathrooms, the number of people can be accommodated, room type, review scores rating, number of reviews, property type, and the number of beds.

3.3 Neural Network

The second nonlinear method that we analyze is neural network. Neural networks can flex themselves to capture complex underlying data structures, but on the other hand, they are also the least transparent, least interpretable, and most highly parameterized machine learning tools due to their complexity. In addition to an input layer of raw predictors and an output layer that aggregates hidden layers into an ultimate outcome prediction, the model incorporates more flexible predictive associations by adding hidden layers between the inputs and output.

When we construct the architecture of neural networks, there are many combinations

of the number of hidden layers, the number of neurons in each layer, and which units are connected. Recent studies show that deeper networks can often achieve the same accuracy with substantially fewer parameters, while some argue that in small data sets simple networks with only a few layers and nodes often perform best(Eldan & Shamir, 2016). We consider architectures with up to four hidden layers. Our shallowest neural network has a single hidden layer of 32 neurons. The second one has two hidden layers with 32 and 16 neurons, respectively; and the third one has two hidden layers with 32 and 8 neurons, respectively. The fourth one has three hidden layers with 32, 16, and 8 neurons, respectively. The last one has four hidden layers with 32, 16, 8, and 4 neurons, respectively. When choosing the number of neurons in each layer, we generally followed the geometric pyramid rule(Masters, 1993). All architectures are fully connected so each unit receives an input from all units in the layer below. We use the same activation function at all nodes and choose a popular functional form in recent literature known as the rectified linear unit ($ReLU$). Moreover, we simultaneously employ early stopping and batch normalization regularization techniques in the estimations. We tried with and without a 0.001 learning rate combined with epochs 100, 200, and 300, respectively, and models with 300 epochs at a 0.001 learning rate give us better performances.

The third architecture gives us the minimal cross-validation RMSE of 0.3907554, which is slightly smaller than the RMSE using the backward stepwise selection. Figure 6 shows the loss and mean absolute error in the training set and cross-validation. Then we tried other architectures combined with different dropout rates to avoid overfitting. However, the results were even worse than the previous estimations. We fitted the test set using the optimal model, and the RMSE is 0.4090677, and Figure 7 and Figure 8 illustrate the prediction performance.

4 Discussion

4.1 Comparison of Learning Algorithms

Table 3 presents the number of parameters, the minimal cross-validation RMSE, and the test set RMSE using the above three methods. We assess the predictive performance by comparing the test set RMSE. Backward stepwise regression has 34 parameters and gives us the largest RMSE. Random forest has 2 parameters and gives us the best fit among these three models. Neural network has much more parameters, but it only performs slightly better than the backward stepwise regression.

All three methods have their advantages and limitations. Backward stepwise regression is easy to implement and computationally efficient. However, it is applicable only if the solution is linear, and the algorithm assumes the input residuals to be normally distributed and input features to be mutually independent. Moreover, the performance of stepwise regression is generally worse than alternative methods because it is too greedy. Therefore, it makes choices that are locally optimal at each step but suboptimal in general. Stepwise selection also cannot go back to revise its past choices, so it does not consider all possible combination of potential predictors and tend to produce an unstable selection of variables. Random forest is flexible to both classification and regression problems. Moreover, as our results indicated, it provides much more accurate predictions and handles overfitting efficiently. Despite these advantages, random forest requires much computational power and much time for training. Moreover, it is also hard to interpret. Neural networks can learn and model non-linear and complex relationships without imposing restrictions on the distribution of input variables. In addition, after learning from the initial inputs and their relationships, neural networks can infer unseen relationships on unseen data to generalize the model and make predictions. However, neural networks are "black boxes", we do not know how or why our neural network came up with a

certain output. Moreover, neural networks usually require much more data than traditional machine learning algorithms. They are also more computationally expensive than traditional algorithms.(Tu, 1996).

4.2 Machine Learning vs Causal Inference

Finally, this project also shows the differences between prediction and causal inference. In prediction, we compare outcomes across different combinations of predictors, and machine learning can help us find correlations. At the same time, machine learning can be limited to generalizing the patterns they find in a training set. In causal inference, we want to understand what would happen to the outcome as a result of a treatment or intervention, and causal inference would help us to figure out what would happen if we changed some of the underlying assumptions in the model. That is what machine learning cannot do.

Specifically, our models focus on predicting prices rather than understanding what causes the change in prices. In the random forest model, which gives us the best prediction, longitude and latitude are the two most important variables. However, that does not mean these variables have causal relationships with price because there are likely to be some omitted variables. For example, listings with better amenities tend to have higher prices, but relevant variables were not included in the model.

To estimate the causal relationship between listing prices and potential determinants, we need to address the omitted variable bias and endogeneity. One possible way is to exploit an exogenous shock, and the most popular methods include difference in differences, regression discontinuity design, and instrumental variables. Our sample is cross-sectional data with 36,000 variables initially, so we can consider using the regression discontinuity design if we can find a proper exogenous shock.

In recent years, people have started combining machine learning with traditional esti-

mation methods to understand causal relationships using large-scale and high-dimensional heterogeneous data in some areas, such as program evaluation and biological networks.(Lecca, 2021; Linden & Yarnold, 2016).

5 Conclusion

In this project, we perform backward stepwise regression, random forest, and neural networks to predict Airbnb prices in six major U.S. cities. Random forest is the best-performing method since it gives us the smallest RMSE. The performance of neural networks is unsatisfactory as it is only slightly better than backward stepwise regression. Machine learning techniques can help us identify covariates that have an important influence on the prices of listing, but it does not identify causality. To take advantage of transitional methods and machine learning, we can combine machine learning and causal inference to improve experiment design and analysis in some cases.

References

Airbnb. (2022). *About us.* Retrieved 12-01-2022, from `https://news.airbnb.com/about-us/`

Eldan, R., & Shamir, O. (2016, 23–26 Jun). The power of depth for feedforward neural networks. In V. Feldman, A. Rakhlin, & O. Shamir (Eds.), *29th annual conference on learning theory* (Vol. 49, pp. 907–940). Columbia University, New York, New York, USA: PMLR. Retrieved from `https://proceedings.mlr.press/v49/eldan16.html`

Lecca, P. (2021). Machine learning for causal inference in biological networks: Perspectives of this challenge. *Front Bioinform*, *22*.

Linden, A., & Yarnold, P. R. (2016). Combining machine learning and matching techniques to improve causal inference in program evaluation. *Journal of evaluation in clinical practice*, *22*.

Masters, T. (1993). *Practical neural network recipes in c++.* Academic Press.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs*, *9*.

Schiltz, F., Masci, C., Agasisti, T., & Horn, D. (2018). Using regression tree ensembles to model interaction effects: a graphical approach. *Applied Economics*, *50*, 6341-6354.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, *49*, 1225-31.

Appendix: Tables and Graphs

Table 1: Summary Statistics of Selected Variables

	Mean	SD	Min	Max	N
log of price	4.783	0.721	0	7.6	36,000
accommodates	3.155	2.168	1	16	36,000
bathrooms	1.240	0.591	0	8	35,905
bedrooms	1.270	0.859	0	10	35,957
beds	1.714	1.274	0	16	35,939
property type					
apartment	0.659	0.474	0	1	36,000
house	0.224	0.417	0	1	36,000
condominium	0.036	0.186	0	1	36,000
number of reviews	20.733	37.577	0	532	36,000
review score rating	94.100	7.771	20	100	27,857
free cleaning	0.269	0.443	0	1	36,000
has profile pictures	0.997	0.055	0	1	35,909
identity verified	0.671	0.470	0	1	35,909
instant bookable	0.261	0.439	0	1	36,000

Table 2: Price by Room Type and Property Type Across Cities

	Room Type			Property Type		
	Entire home/Apt	Private room	Shared room	Apartment	House	Condo
Boston	5.23	4.30	4.43	4.93	4.57	5.01
Chicago	4.97	4.13	3.74	4.59	4.60	4.72
DC	5.26	4.47	3.85	5.00	4.94	5.16
LA	5.11	4.28	3.66	4.65	4.81	4.80
NYC	5.15	4.30	4.06	4.72	4.52	5.09
SF	5.50	4.71	4.22	5.16	5.13	5.43

Table 3: Comparison of Learning Algorithms

	Number of Parameters	Optimal CV RMSE	Test Set RMSE
Backward Selection	34	0.4116133	0.4104787
Random Forest	2	0.3516430	0.3436700
Neural Network	1,393	0.3907554	0.4090677

Figure 1: Distribution of Log of Price

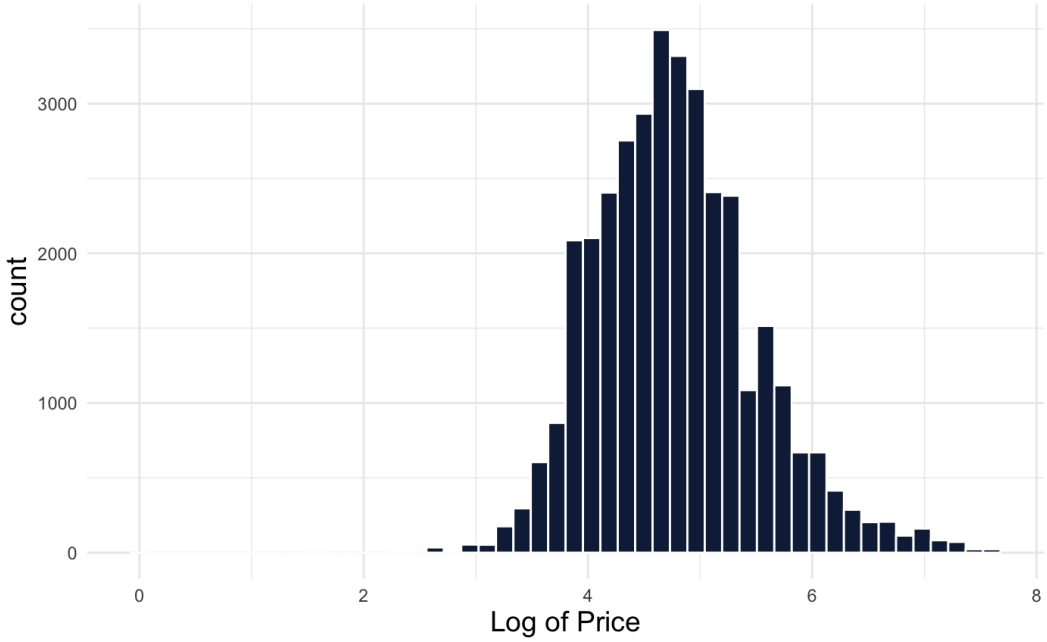

Figure 2: Listing by City and Room Type

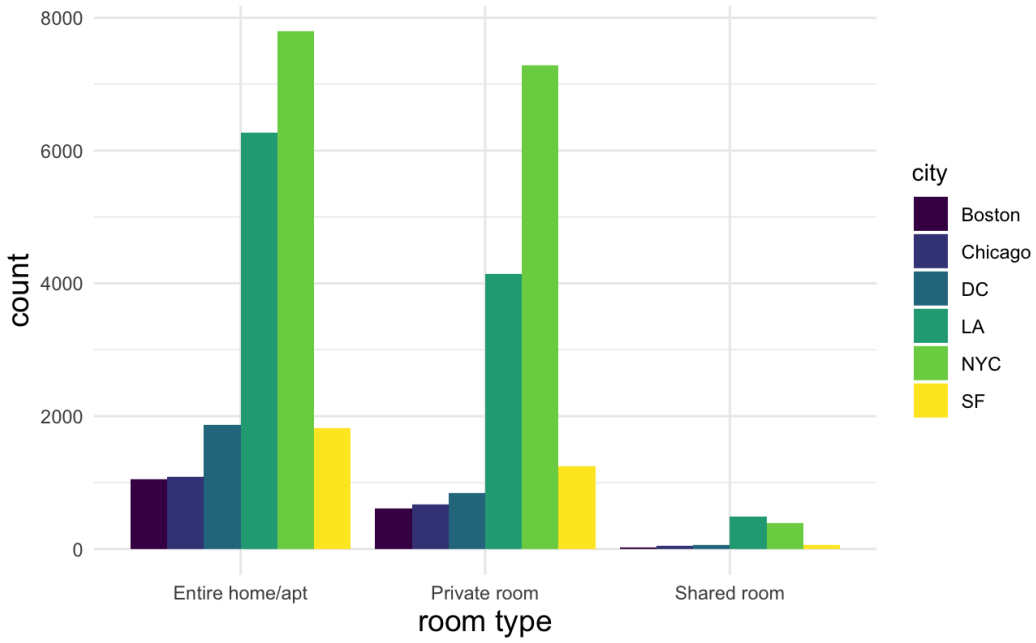

Figure 3: Backward Stepwise Regression: RMSE vs Number of Predictors

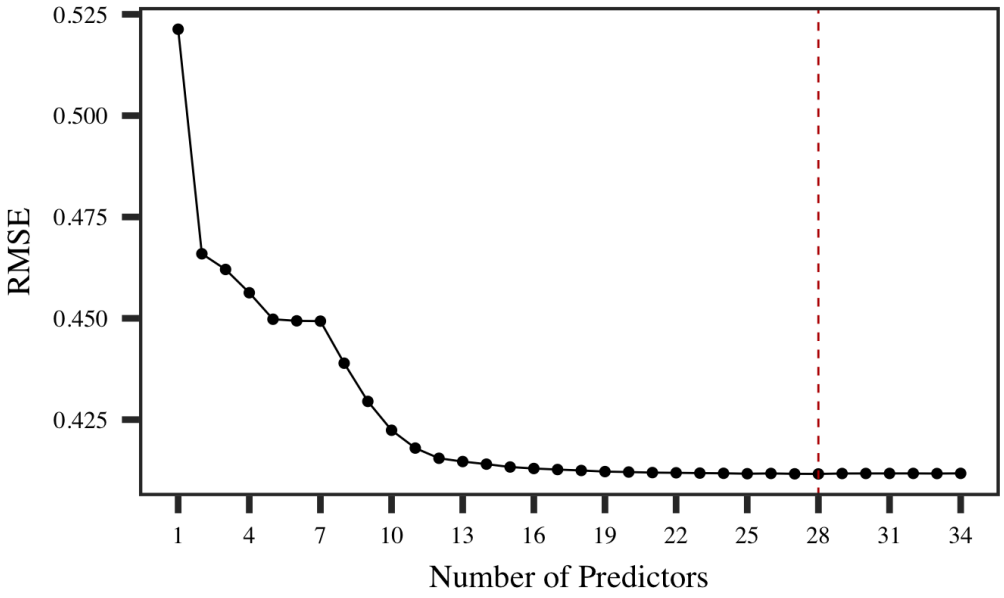

Figure 4: Random Forest: RMSE vs Different Parameters

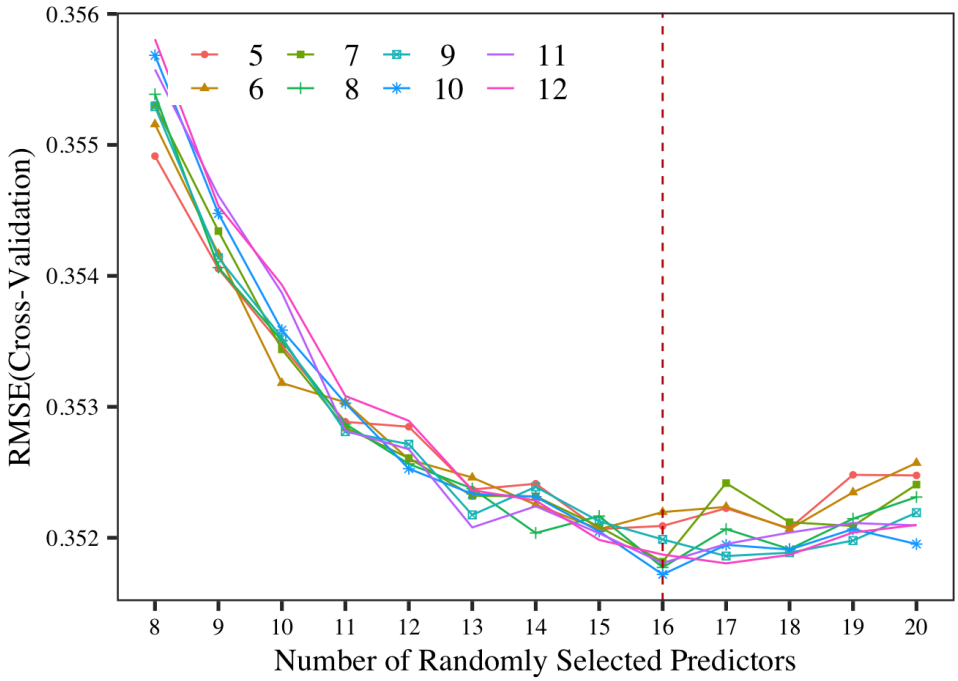

Figure 5: Random Forest: Top 20 Important Variables

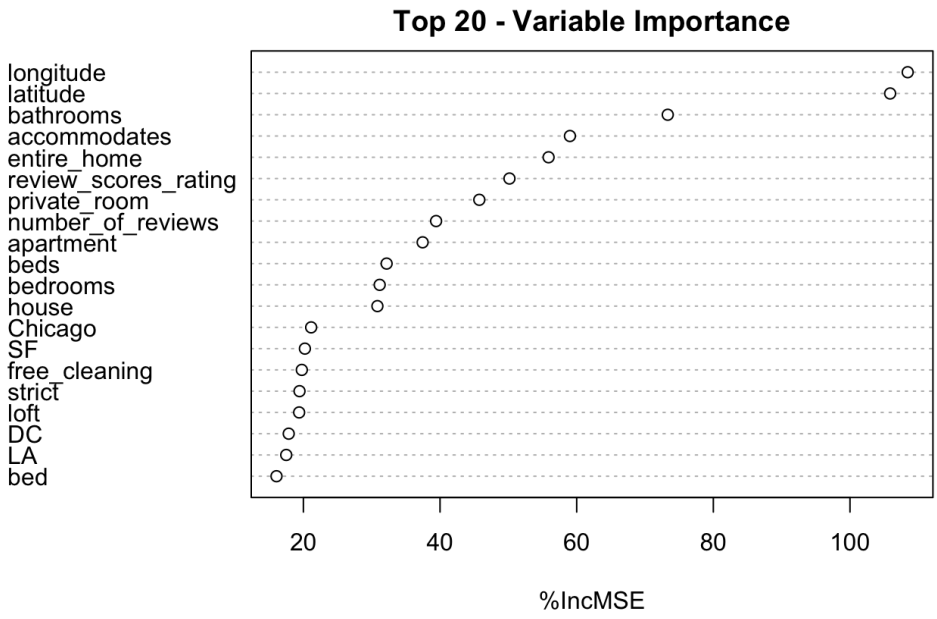

Figure 6: Neural Network: Loss and MAE in Training Set and Validation

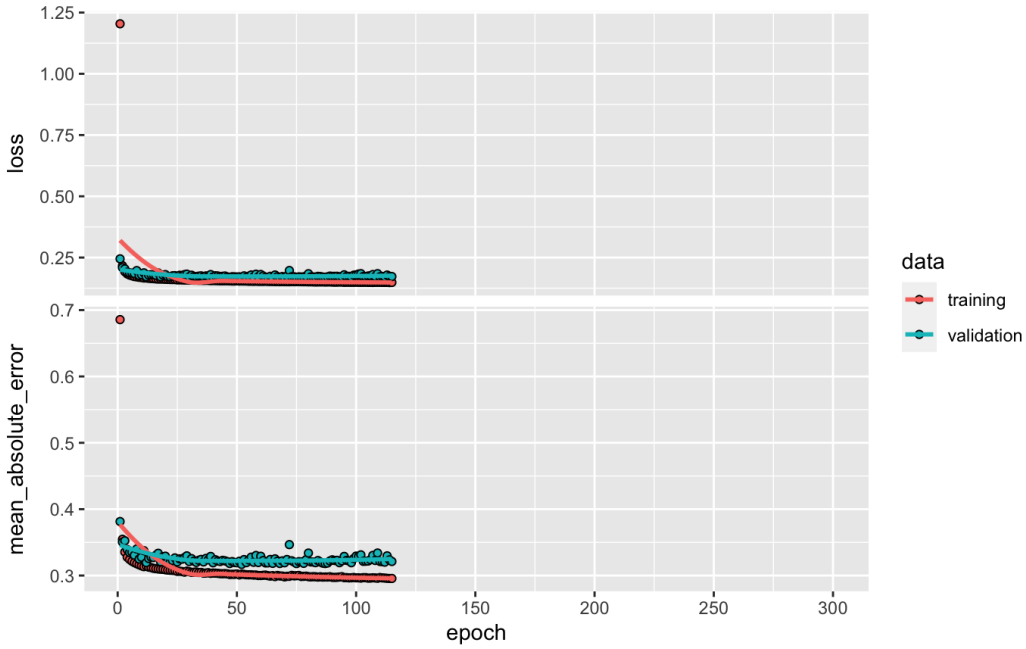

Figure 7: Neural Network: Predicted vs Actual

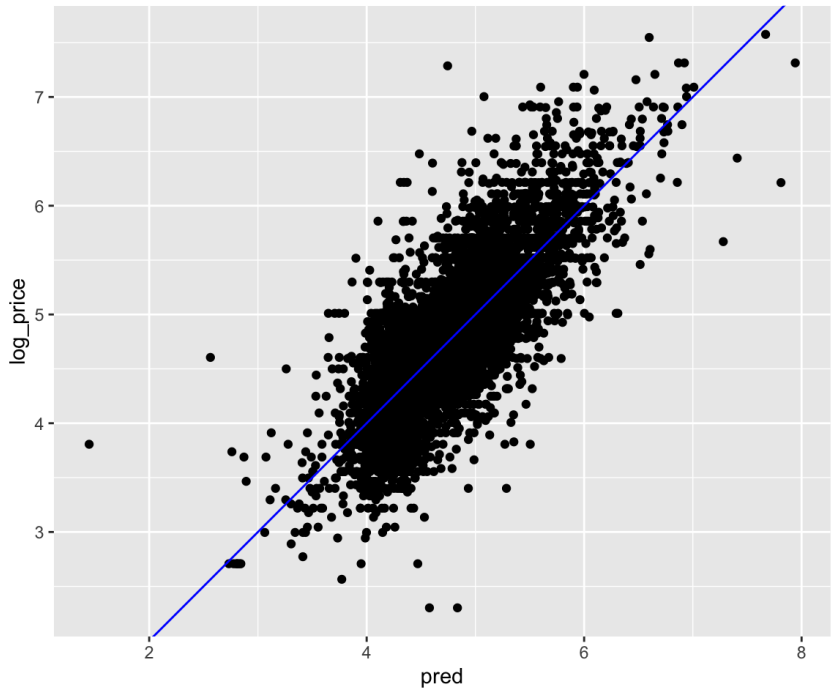

Figure 8: Neural Network: The Difference Between Predicted and Actual