Zhenwei Wei
Kaggle Competition Report

This Kaggle competition aims to accurately predict the price for an Airbnb rental using variables on the property, host, and past reviews. Each participant will utilize two data files: the analysis data for building a model and the scoring data for applying predictions. Specifically, the former contains 90 variables for predictions plus the price variable, and the latter has only 90 predictive variables. I consider the analysis data as a train dataset and the scoring data as a test dataset. In the end of competition, price prediction submissions are evaluated based on the Root Mean Square Error (RMSE) of model prediction; The lower the RMSE is, the better the model is.

Since we have about 90 variables, and many of them are in different formats, we must first understand the meaning of each variable and its effect on predicting price. I created a table of variable descriptions that explains most variables and rate each variable with levels based on their effects. For example, the *home name* variable is not meaningful when people make rental decisions, and the host picture could influence people' decision-making. After this first selection, I excluded a few variables that do not have much predictive power under common sense. Next, I deleted variables that are difficult for me to analyze, such as *neighborhood overview*, *house rules*; they are string variables that are too long to be utilized in the sample model. In the second round, I conducted a multivariate feature selection and refined about 50 most important variables based on both the filter result and my intuition.

Furthermore, I filled the missing value of some variables with the mean value or the most likely level as I cannot predict missing values. This step enhances the quality of dataset and provides a correct data format for using the vtreat function. In addition, for variables that has a large portion of missing values, such as *license* and *jurisdiction names*, they are dropped since they are prone to increase noise and cause bias in the model. Another way of dealing with missing values is to remove blank rows. However, I will need to reduce about one-third of the total data under this method, so I doubt that it is an effective way of processing data. Moreover, I found that some variables are formatted differently in the analysis data and the scoring data, such as *host since*, *first* and *last review*, so I prefer not to use them in my prediction. As I converted the dataset to a format that only has numeric columns and has no null values, I used vtreat to design a treatment plan that stores selected variables in a statistically sound manner, meaning the data will have fewer exceptional cases in future model training and prediction.

To build the model, I firstly experimented with the linear regression model and Random Forest model, but considering the complexity of my data, I switched to a more advanced method. In my final submission, Extreme Gradient Boosting (xgboost) is the supervised learning model I built to predict price. As we know, xgboost is very popular in Kaggle competition, and it is very effective in raising the performance of machine learning. In the class lecture, we also covered how to build it, so I am confident to use it. In my first few attempts, I reached a low RMSE of around 75, which is not quite a competitive score. In order to improve the score, I spent more time on tuning parameters. Xgboost feature importance score was used to identify the distinct variables further, it reflects the gain and coverage aspect of a model. Gain implies the improvement in accuracy brought by a feature on its branches, and coverage measures the relative quantity of observations concerned by a feature. I removed a few variables in my model based on these two indicators and finally obtained about 40 variables. Moreover, *xgb.cv*, a k-folds cross-validation function, was also adopted to seek the most appropriate parameter, fold numbers as well as learning rate (shrinkage at every step). By conducting cross-validation and testing different combinations

of parameters, I finally improved the model outcome from 75 to near 66. About the limitation, overfitting is one of my main concerns since this could easily happen in the extreme gradient boosting model, so I was cautious in adjusting parameters like nfolds and learning rate.

Although I have improved my model prediction by a difference of 10, it is still possible to reach a lower RMSE. For instance, I could do more data processing works since I have a few very important but complicated variables such as *zipcode* and *amenities*, and I could also build the Light Gradient Boosting Machine model (LGBM) instead of xgboost model. LGBM model runs faster than xgboost since it has faster training speed, higher efficiency, lower memory usage, better accuracy than any other boosting algorithm, and better compatibility with large datasets.

In conclusion, the competition time is pressing, I encountered many obstacles regarding optimizing the model, and some remain unsolved. However, I appreciate this experience since it reinforces my ability to process data and build the machine learning model, and a real-world data project also allowed me to learn data analysis knowledge in the rental industry.