THE UNIVERSITY OF SYDNEY

Group Project: Airbnb Price Predictions

Due date: Friday 31 May 2019

Group Members:

Lingxiao Li
Xiaoni Wu
Sivial Wang

Introduction

AirBnb is a shortened version of its original name, AirBedandBreakfast.com, it is an American online marketplace and hospitality service brokerage company. The company does not own any of the real estate listings, it receive commissions from each booking, and the members can use its service to offer lodging, tourism experiences, and primarily homestays (wikipedia, 2019). The advantages of Airbnb include wide selections of property types, guests can enjoy additional services such as classes and sightseeing offered by local Airbnb hosts.

The purpose of this project is to explore the Airbnb dataset downloaded from kaggle, and build five different models to predict the nightly price of Airbnb listings base on the existing data. The dataset include training set and testing set, and each row in these two dataset corresponds to a separate Airbnb listing in London. The task is to use the prediction variables such as the security_deposit, cleaning_fee, and extra_people, to predict the response variable, price, which is the British pound sterling (GBP) price per night for each listing.

The main procedures contain data cleaning and preprocessing, implementing Exploratory Data Analysis (EDA), and build five different models to predict the price in the testing set. The five models that implemented in this assignment include Linear Regression (OLS), Lasso, Ridge, Random Forest and Model Stacking. The predicting results are uploaded to Kaggle, and then Kaggle would randomly splits the observations in the test set into validation and test set, and compute the RMSE of the predictions. The corresponding RMSE are 61.87, 72.35, 72.81, 43.14 and 40.74, so the optimal model selected for this project is stacked model with smallest RMSE of 40.74.

Data Preprocessing and Exploratory Data Analysis

- Data Cleaning and Preprocessing

After loading two dataset separately, we check the variable type of each column and the first five rows of both datasets. For training data processing, we start by computing the descriptive statistics, and we found the number available observations for each variable are not the same, which means the dataset contains some missing

values. Also, the result shows that there contains some outliers in the dataset. For removing the outlier more correctly, we draw the boxplot for each variable. Based on boxplots and the descriptive statistics results, we dropped some outliers, such as the security deposit which is large than 2500, the number of beds is larger than 9, etc. As total, we removed 18 rows from the training data that contain outliers.

Moreover, at the beginning, we checked both dataset and found both of them contain categorical variables, so we need to create dummy variables for them. For creating the dummy variable, we decided to concat two datasets which will ensure that the dummy variables are consistent in both datasets. Before we concat two datasets, we added a column ‘label’ in both dataset and set it as 0 in all training set and set it as 1 in all test set, which can help us to split them later. Then we check the number of missing values and the correlation in the dataset. By calling the isna() method we get to know that there are lots of missing values we need to deal with. Before doing so, we want to drop some unuseful columns first. We draw a Pearson Correlation Matrix(Figure 1) and observe some multicollinearity among the features. The ‘beds’, ‘bedrooms’ and ‘accommodates’ variables are highly correlated. The ‘review_scores_accuracy’, ‘review_scores_cleaniliness’, ‘review_scores_communication’, ’review_scores_value’ and ‘review_scores_rating’ are also multicollinear. Therefore, we decide to remove some of these features in order to avoid redundancy and make the prediction more precise.

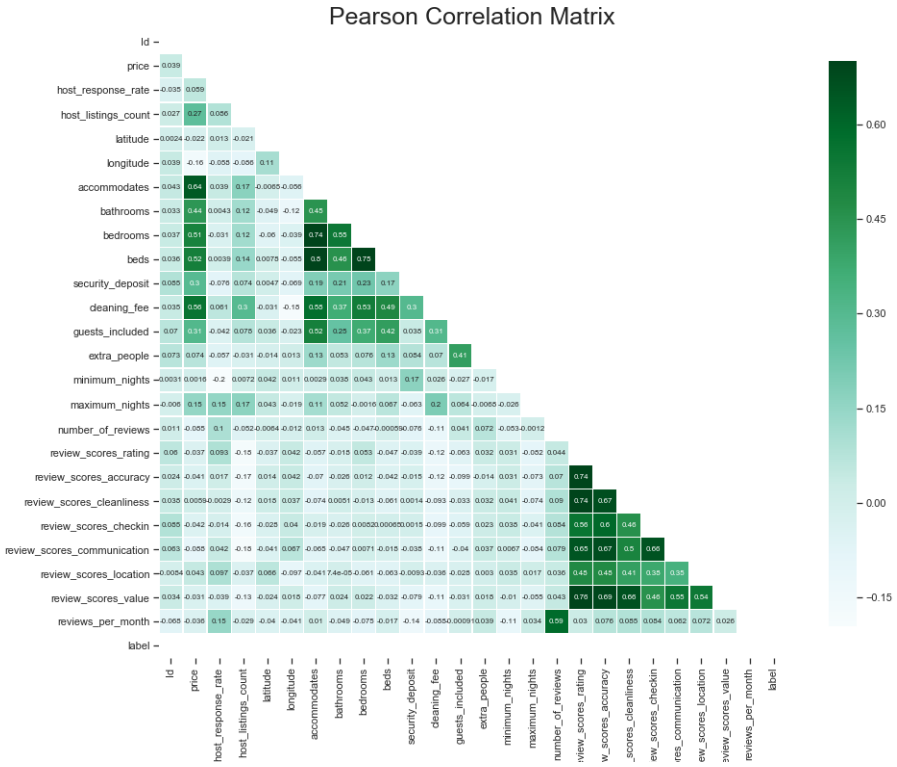

Figure 1: Pearson Correlation Matrix

Once we remove the collinear features, we start working on dealing with missing values. We cannot just simply drop the rows containing missing values because we have to make sure that there are 1,000 rows in the test set after data cleaning. Therefore, we use different methods to replace the missing values. For some of the numerical features, we replace the missing data with column mean, and for others we replace with a 0 value. For categorical features, we replace the missing data with ‘f’, ‘no response’ or ‘none’. The reason we doing so is because the data is missing due to the lack of responses from the users or hosts. After we filling out the missing data, we create a new feature ‘distance’ by calculating the coordinate difference between each house and central of London using the features ‘longitude’ and ‘latitude’, and we drop these two columns once we get the distance variable. We also encode smaller number of categorical variables for property type. Then we need to transform categorical columns to binary columns by creating dummy variables. After that, we are done with data cleaning and we split the combined data back to test set and training set by using the ‘label’ variable we have created at the very beginning.

- **Exploratory Data Analysis**

We first plot the histogram of the response variable ‘price’ of the training data to gain some insight of the distribution. The plot (Figure 2) shows that the variable has a pronouncedly right skewed distribution. A substantial number of listings have a low nightly prices. Then we use the describe() method with skewness and kurtosis added to get the basic statistical details of this variable. The maximum price is 400.000 and the minimum price is 25.000. The skewness is 1.697 which indicates the price variable is positively skewed and the kurtosis is 3.460 which means that the variable has heavier tails than a normally distributed variable. To make the distribution normal, we can do some transformation on the price variable.

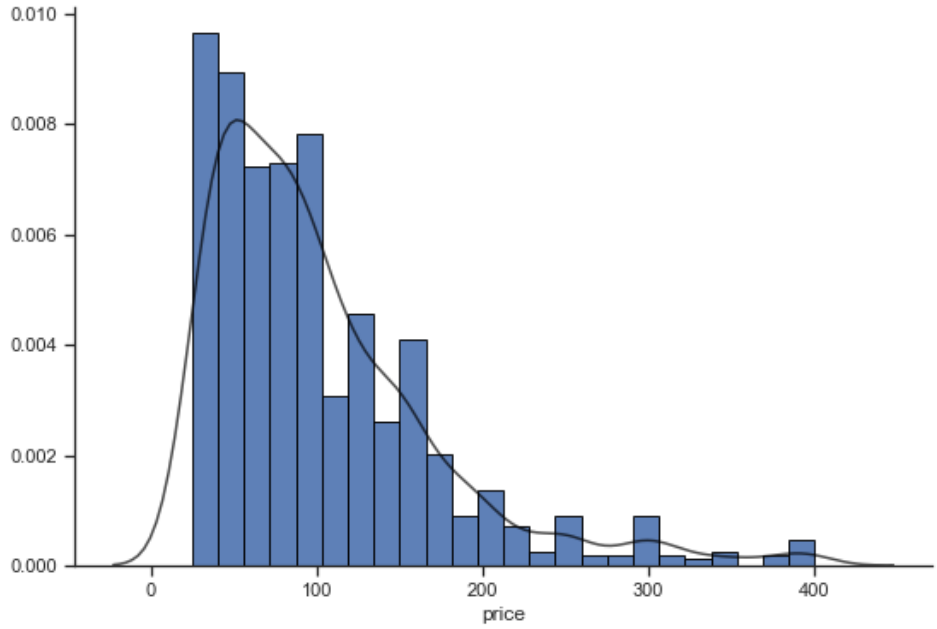

Figure 2: Histogram of Price

Also we plot the histograms of other selected variables to see the distribution of each feature. We observe that most of the variables do not have normal distribution. Only the distance variable is normally distributed.

Moreover, Figure 3 shows the relationship between price and each feature. The following figure just shows the scatter plots of the price verses some of the features, with the added linear regression line. Let’s pay special attention on the scatterplot of the sale price versus the distance. The downward regression line shows a

negative correlation between the price and distance, which indicates that when the distance between the house and the central of London increases, the price decreases. From the plot we can see that when the distance to central of London is within five kilometers, the highest nightly price is 400 GBP. When the distance is more than five kilometers, the highest price suddenly drops to about 300 GBP. Most of the prices are gathered between 25 to 150 GBP.

The scatter plots also shows that some features have the non-constant noise variance, and there is the nonlinear relationship between price and some features.

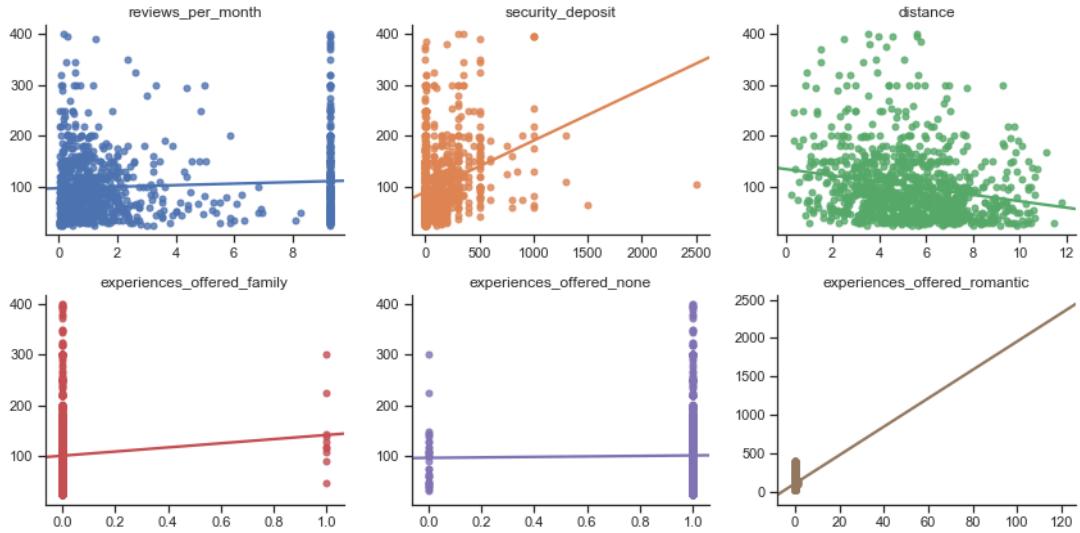

Figure 3: Scatterplot and Regression Line of some Features

Based on this situation, we decide to do the transformation on the response to improve performance, which is an easy way to reduce skewness, make a almost constant noise variance and explain the nonlinearities. Thus, we choose to use log transformation on response variable. Once we do the log transformation on price, we plot the histogram (Figure 4) again and we can see that the distribution are now very close to normal and there is positive skewness anymore.

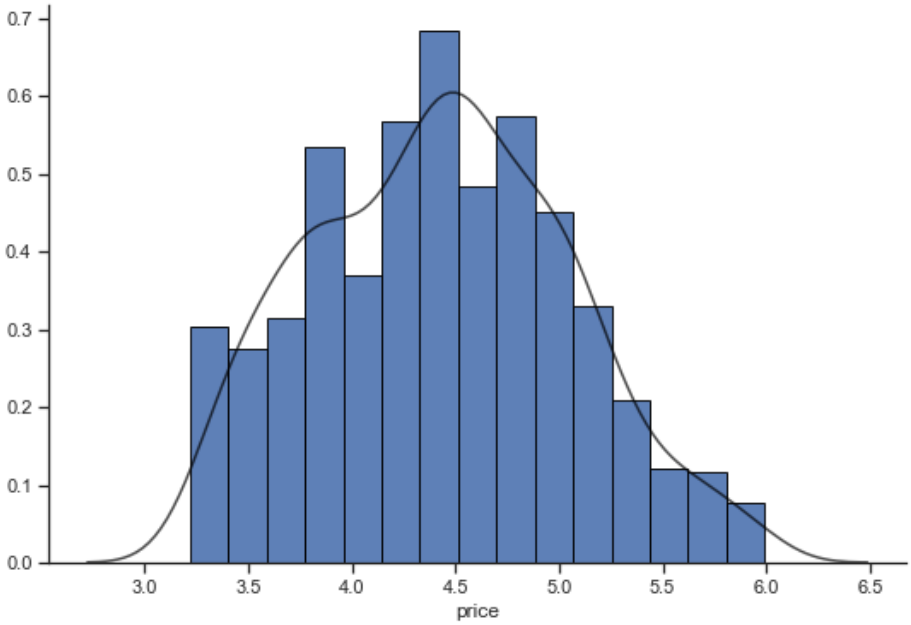

Figure 4: Histogram of Price after Taking Log

After done with the data transformation, we split the test set and training set into response and predictors for building models. However, we standardize the data first before we build the model since the regularised linear methods, such as Ridge and Lasso, do not perform well if predictors have different scales.

- **Feature Engineering**

To observe the relationship between price and the distance from the London central, we create a new variable ‘distance’, and calculate the related distance between each property and the central of London. The latitude and longitude of London is 51.51 and -0.118 respectively, after applying longitude and latitude of each property into the Pythagorean Theorem, the related distance between London central and each property can be determined. To observe the relationship between distance from London central and price, a scatterplot had been applied, and it can be observed that as the distance from the central of London increases, the price decreases. The scatterplots and detail explanation of their relationship is illustrated in the previous Exploratory Data Analysis section.

Methodology

1. **Ridge Regression**

- **Model Description and Motivation**
  One of the models implemented in this project is ridge regression. Ridge regression is a technique used to create a model when the number of predictor variables in a set exceeds the number of observations, or when the data has multicollinearity. Ridge regression model explain data with a minimum number of parameters or predictor variables. When multicollinearity occurs, least squares estimates are unbiased, but the variance could be large. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors (NCSS Statistical Software). However, ridge regression does not get rid of the irrelevant features but rather minimize their impacts on the train model, therefor ridge regression is not an optimal solution for this project.

- **Model Implementation**
  During the process of data cleaning and exploratory data analysis, the training set and testing set had been combined together to implement data cleaning, so when building ridge regression model, we split the combined data set to the training and testing set again according to the 'label' column we have added in the beginning, which makes them the same as the original dataset except that all the invalid and missing values had been cleaned up.

  The next step is to split the training set into y_train and x_train, y_train corresponds to the response values 'price' in training set, and x_train corresponds to the prediction variables which are all the variables in training set except 'price'. Then set all variables except the 'price' in testing set as x_test, then apply the y_train, x_train and x_test into the Ridge Regression model and implement prediction. What the model do is use the response variables and prediction variables in training set to build a model, then put the prediction variables of test set into the model and predict the response variable 'price'.

  In addition, the regularization strength alpha is one of the most important factors in Ridge Regression model, which have significant impact on the prediction accuracy.

Therefore, we apply different alpha into the model and select the optimal values of it according to the prediction accuracy. The optimal value of alpha is approximately 63.47. Figure 5 below is the plot of Ridge Coefficients versus Regularization Parameters. It can be observed that as alpha is getting larger, the predictor coefficient is approaching to 0. At the end of the path, as alpha tends toward zero and the solution tends towards the ordinary least squares, coefficients exhibit big oscillations.

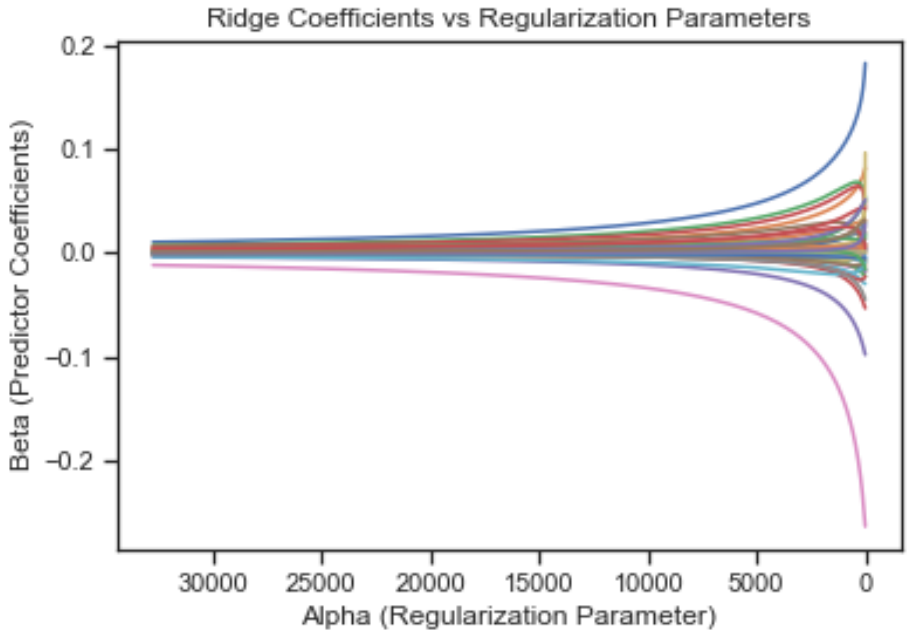

Figure 5: Ridge Coefficient versus Regularization Parameters

- **Model Interpretation**

  After implementing prediction and upload it to the Kaggle, the calculated RMSE of this Ridge Regression is 72.81. The graph (Figure 6) below illustrates the values of the twenty largest (in magnitude) estimated coefficients. It can be observed from the graph that accommodates have the largest positive coefficient, followed by bathrooms and bedrooms, while private room type have the strongest negative impact on the dataset.

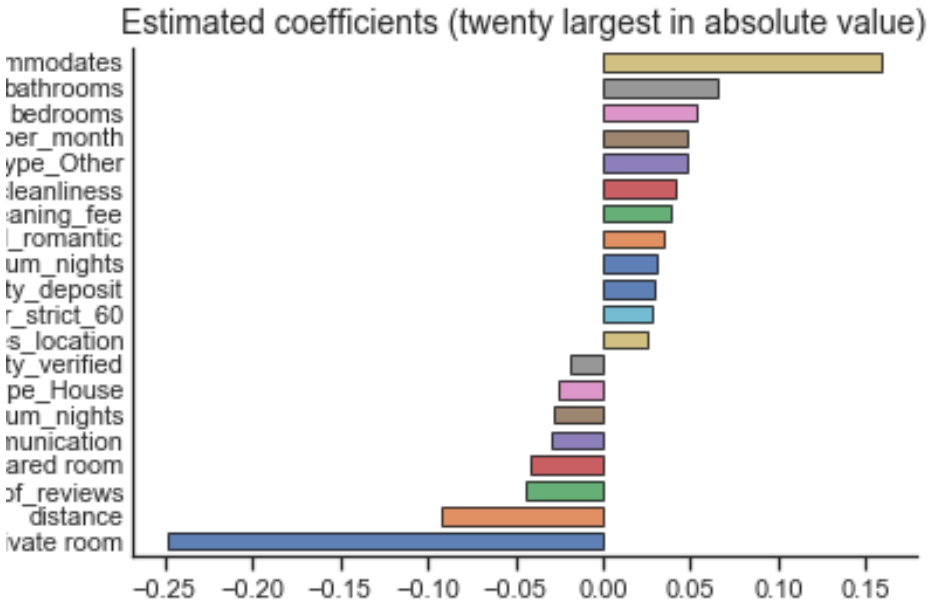

Figure 6: Twenty largest Estimated Coefficients of Ridge Regression

2. Random Forest

- **Model Description and Motivation**

  One of the advanced models implemented in this project is the random forest, which is an ensemble method and use aggregated results from ensemble of trees. The single tree regressor does not have same level of predictive power compared to other models as it relies on predicting results from a single tree and is prone to overfitting. Thus, random forest is expected to give an improvement on prediction accuracy. In addition, random forest decorrelates the trees by introducing splitting on a random subset of features so that variance can be averaged away.

- **Model Implementation**

  Similarly to the ridge regression implementation, the random forest model is implemented and trained on the training data that is free of outliers, invalid and missing values. The predictors including ‘property_type’, ‘distance’, ‘accommodates’ and so on as determined in previous discussions, and the response variable is ‘price’ of the

training data. To get the optimal random forest model, we use RandomizedSearchCV() to tune hyperparameters. The RandomizedSearchCV() function applies all possible combination of parameters specified into model and return the best parameter combinations, best estimators and corresponding best score (mean cross-validated score of the best_estimator). In addition, 5-fold cross-validation is applied to the parameter tuning for preventing model overfitting. The tuned hyperparameters and their explanations are as following:

'n_estimators': number of trees to grow.
'max_features': the size of the random subsets of features to consider when splitting a node
'min_samples_leaf': minimum number of samples in a leaf

We try different values of 'n-estimators' in the range [1500, 2000], different values of 'max_features' ranging from 1 to the number of predictors in training data, and different number of 'min_samples_leaf' in the range [1,5, 10, 20, 50]. And the best parameters found by randomized search is {'n_estimators': 2000 , 'max_features':16 , 'min_samples_leaf': 1}.

- **Model Interpretation**

After implementing prediction on test data and uploading results to the Kaggle, the calculated RMSE of this random forest model is 43.25, which has an improvement on the benchmark of 50 RMSE, so the results is acceptable. The graph (Figure 7) below illustrates the variable importance in affecting the house price. And It can be observed from the graph that room type has the significant effect on Airbnb price, followed by accommodates, cleaning fee and

bedrooms.

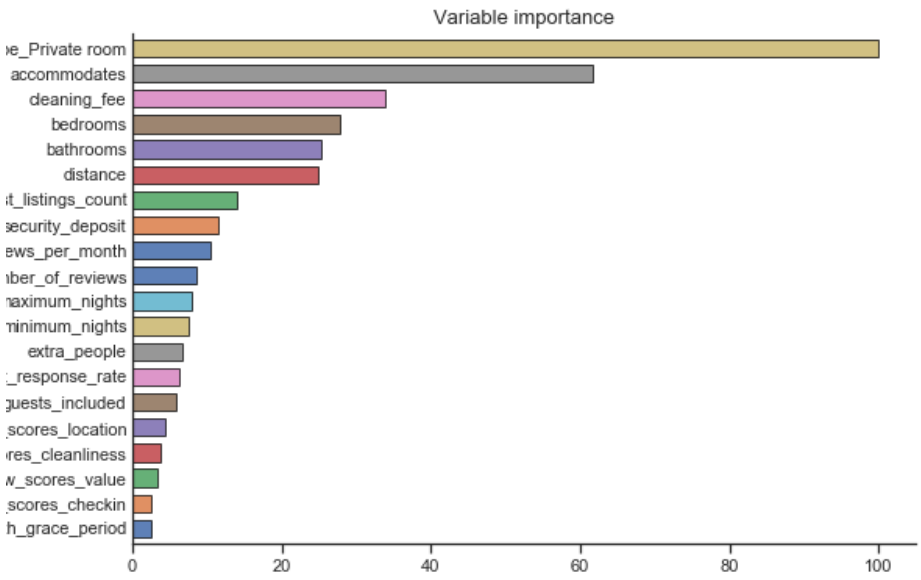

Figure 7: Variable importance of random forest

Validation Set Results from Kaggle and Model Comparison

Table 1:

Model	OLS	Lasso	Ridge	Random Forest	Stacked Model
RMSE from Kaggle	61.87	72.35	72.81	43.14	40.74

As illustrated in the table 1 above, the RMSE of these five models are 61.87, 72.35, 72.81, 43.14, and 40.74 respectively. Base on this result, the optimal model for this assignment is the stacked model, with the smallest RMSE of 40.74. The Ordinary Least Squares Regression models is one of the most widely used modelling method, because it is easy to explain and understand, however, it is sensitivity to outliers and tend to overfit the data, so it is normally considered as a benchmark model.

Ridge and Lasso model is an improvement of OLS, and the only difference between Ridge and Lasso is that Ridge minimize the impacts of irrelevant variables on the training model, while Lasso completely remove those irrelevant values, therefore Lasso could be an improvement of Ridge. However, Lasso selects at most n variables before it saturates, and it cannot do group selection. If there is a group of variables among which the pairwise correlations are very high, then the LASSO tends to arbitrarily select only one variable from the group, and this can affect the accuracy of predictions.

In terms of Random Forest, it helps to overcome the problem of over-fitting since it average the results of different decision trees, and it also have less variance than a single decision tree. However, it requires careful tuning of different hyper-parameters, which means Random Forest is time and computational expensive compared to other models discussed above. In addition, it does not have good interpretation power.

Final Remarks

In conclusion, the purpose of this project is to make predictions on the Airbnb Listing price base on the existing data by using five regression models. The training and testing set was combined together to implement data preprocessing and Exploratory Data Analysis. After that, the combined dataset was split back to the training and test set, these two sets are the same as the original training and testing set except that all the missing values and invalid values were cleaned up. Then the two cleaned datasets were applied into five different models including OLS, Ridge, Lasso, Random Forest and Model Stacking and make a prediction. Finally the predictions were uploaded to Kaggle and the RMSE were calculated. The RMSE of OLS, Lasso, Ridge, Random Forest and Model Stacking is 61.87, 72.35, 72.81, 43.14, and 40.74 respectively, and base on these results, the optimal model for this project is the stacked model with the smallest RMSE of 40.74.

References

NCSS Statistical Software, Ridge Regression, https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf