**Airbnb Price Prediction**

**1 Introduction**

Airbnb is one of the most popular platforms for short-term leasing and renting in the crossover world. Now, this platform has grown to 6+ million active listings and 4+ million hosts worldwide [(Matthew Woodward, 2022)](https://www.zotero.org/google-docs/?dBg8tx). The core service provided by the website for users will calculate a suggested price based on the selected listing features.

Considering the significant growth of demand, Airbnb would like to develop a model for predicting nightly prices of listings based on state-of-art techniques from statistical learning. Thus they approached us. This report will focus on how we develop a well-performed model for estimating the price of listing in Sydney, Australia. Typically, based on train data, we will do data processing and exploratory data analysis first, and then we will move to feature engineering and different models developed for estimation. Last but not least, we would pick “the best-performed model” based on the evaluation and comparison of test data. The ultimately selected model should be able to provide Airbnb with a predictive insight into the price level for their future listing.

After using the data to predict the price per night by comparing the test RMSE, cross validation RMSE and R2 in our models, we choose to using the model stack for prediction which has the lowest test RMSE, cross validation RMSE and highest R2 which means this model will gives Airbnb the most accurate prediction to the true value of the price per night for Airbnb compared to other models based on the dataset.

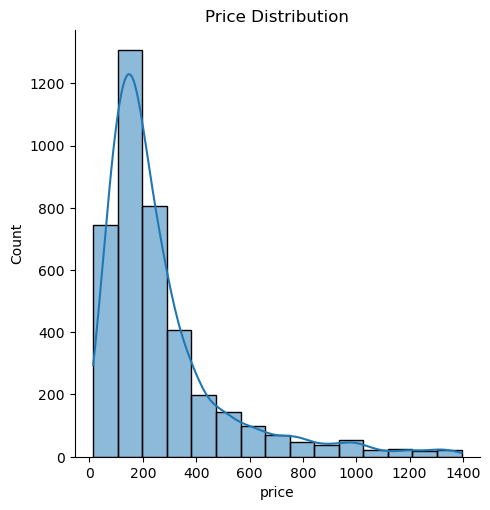
**2 Data Processing**

**2.1 Data description**

To make a more accurate price prediction for Airbnb, a training dataset which contains 4000 existing listings in the area of Sydney will be used to build different forecasting models. Other than price and id for each listing, there are 34 columns of variables indicating the condition of the listings. Information types could be categorized as x types: host and neighbourhood location, property type and capacity, and review scores. In order to explain different types of listing conditions, the dataset contains numerical, categorical and boolean data. Regarding the integrity of the training dataset, there are only eight columns out of 36, including missing values, most of which are from host and neighbourhood variables.

Besides, a test dataset is also provided, including the same number of columns as the training set, but without price. Other than 4000, only 2000 entities are contained in the test set. The idea is after appropriate processing and standardizing on both train and test datasets under the same pattern, five models will be built based on the training set's information. Then predict the price of the test set with the same predictors using the optimal model.

**2.2 Data processing**

Before constructing prediction models, the exploratory data analysis will be processed on the original training data and copying another training and testing set as a clean set to process feature engineering. Since this project focuses on predicting price, we start with a general inspection of the price distribution. From Figure X, we know there is a right skew on the price, and most listings have a lower price range of around 150 to 300. Also, a log transformation on price might be considered later during feature engineering for better performance on the linear regression models.

Then, since the integrity of the training data set is overall well with only a few columns of missing value, depending on different types and content of predictors, we use three strategies to handle the missing value:

1. After general EDA to each predictor, considering the price correlation and number of ‘NA’ in each column, eight columns with less correlation with price and many NA will be deleted from both the training and testing set.
2. For some of the vital predictors with a high correlation with price, we are going to fill the missing values under a rational logic, such as the filling method of “Accommodations,” “beds,” and “bedrooms,” which will explain later in EDA.
3. The missing value is filled with the string value “NA” as a new category for some categorical columns.

Since the dataset aims to provide detailed information on the listings of Airbnb, there are some columns that include a great deal of texting description but might be vital for price prediction. Therefore, categorical columns with text information have transformed into numerical for model application. Techniques include moving the string character out and counting the description word by splitting them with commas.

Finally, the last processing stage for the training and testing set is processing dummy variables for all processed categorical predictors in the feature engineering before training models.

**3 Exploratory data analysis**

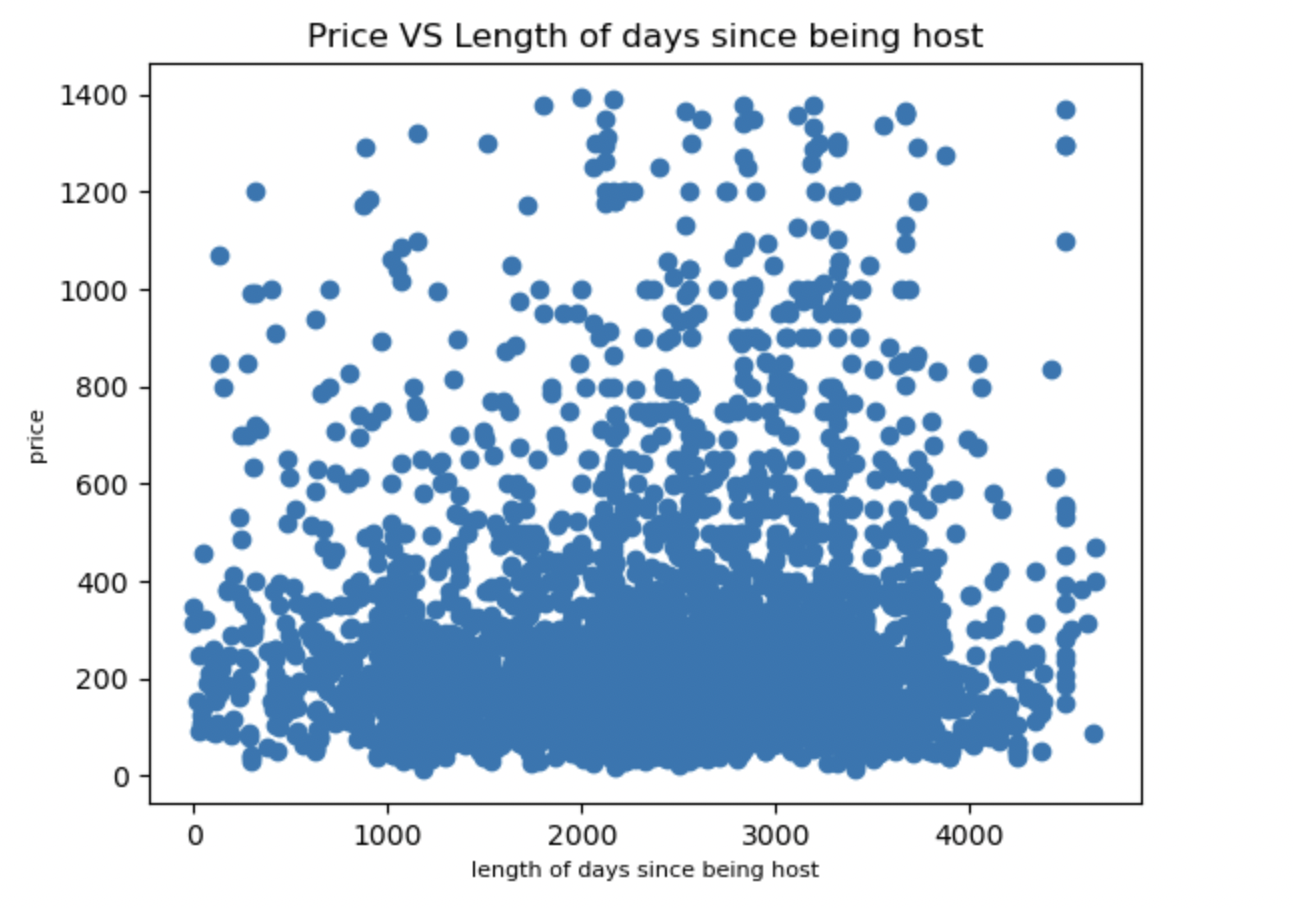
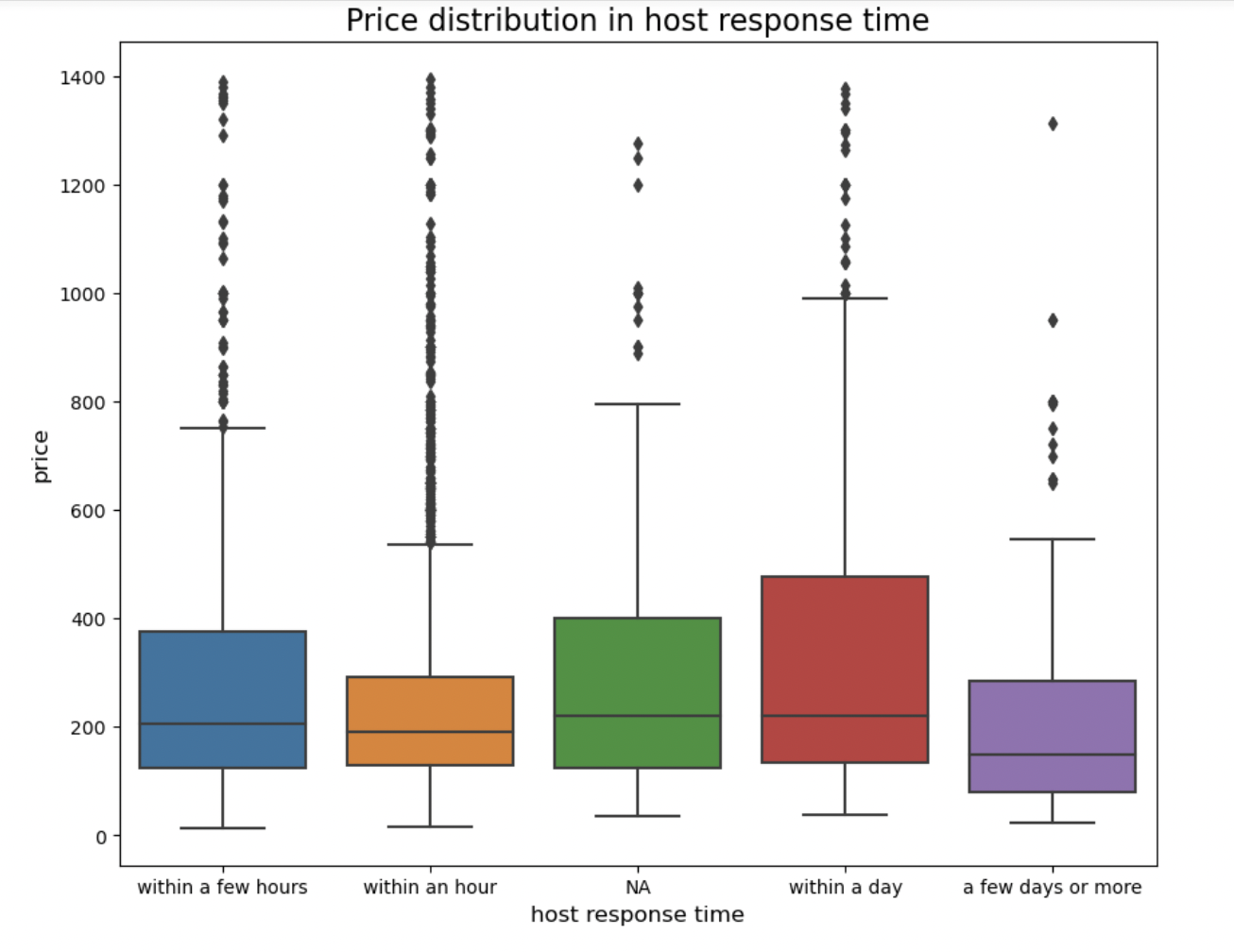
Before EDA on each predictor, we start from the general inspection of the whole train dataset with data information and the number of missing values in each column. Furthermore, based on different categories of predictors, we divided the 34 columns (except ‘id’ and ‘price’) into five parts for detailed EDA, which are

1. Description and neighbourhood overview
2. Host related
3. Neighbourhood / Longitude Latitude / Max Min Nights
4. Property Condition
5. Review

**3.1 Description and neighborhood overview**

The description and neighbourhood overview part contains mainly the host's description of the facilities and neighbourhood. These sentences contain much irrelevant information about the price with many missing values in the Neighborhood overview. After looking into the description and the neighbourhood overview, both columns will be dropped in the feature engineering.

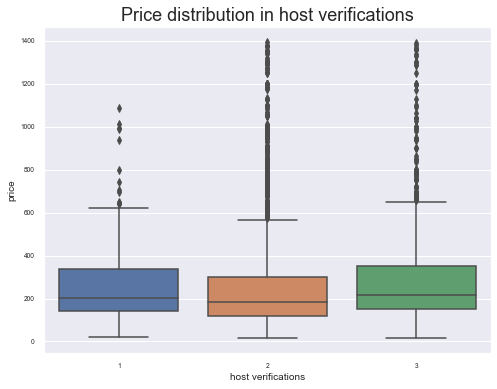
**3.2 Host related**

For the host section, the original form of the ‘host\_since’ is objects, which is unsuitable for applying time series-related analysis; therefore, we have first transformed “host\_since” into date time and generated the scatter plot. Below is the relationship between the length of days since one being a host with the price of their listings. From FigureX, it is hard to find a clear price correlation with hosting time. Also, since the location of the host is considered not to have a direct relationship with the listing price, both predictors will be dropped later.

Since the host response time includes missing values, we filled the null with ‘NA’ before further groupby and visualizations. From the barplot above and the groupby mean value on price, different response time does have a price gap. For example, the mean price for a response within one day is the highest at $350.28, while the mean price for a response within a few days or more has the lowest price at $237.51. Therefore we believe that the host response time will affect the price.

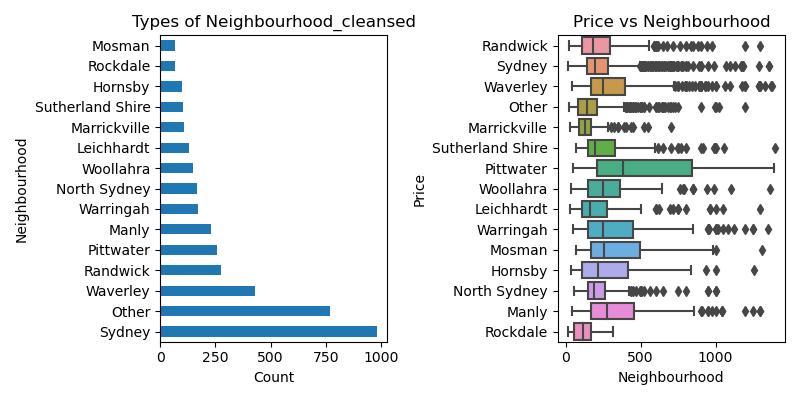
In terms of the response rate, the original data type was the object. To better analyze it, we first replace “%” with whitespace and then transform it into the float type. Missing values are filled with the median of the total host response rate. The scatter distribution from FigureX indicates that most hosts have a full response rate, but the price relationship with the predictor is unclear.

For the host\_is\_super\_host, which is a boolean data, there is a price difference of $20, and a super host ($255.56) generally has a higher mean listing price. From the above figurex we could find that for the host listings count, the prices are very scattered, but still a slight positive relationship with price. Similar to the other boolean data, the host\_identity\_verified, if the host’s identity is verified, then the price($271.10) will be higher than the non-verified ($237.59).

Another categorical predictor related to hosting is host verification, which includes email, phone, work email and a mix of these types. To better know how this impacts hosts’ listing prices, we transfer the categorical data into numerical, which indicates the number of verification types a host has by splitting with commas. Figurex and the groupby mean value show that hosts with three kinds of verification have a higher average price of $308 compared to two types ($259) and only one type ($258).

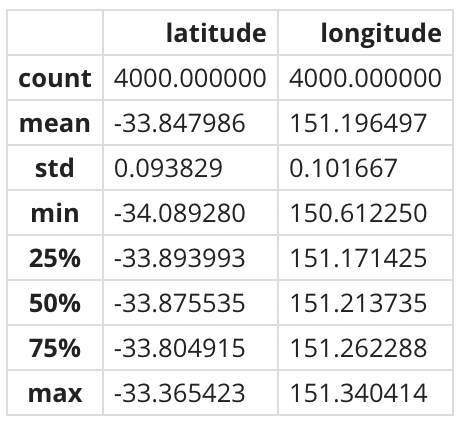
**3.3 Neighborhood / Longitude Latitude / Max Min Nights**

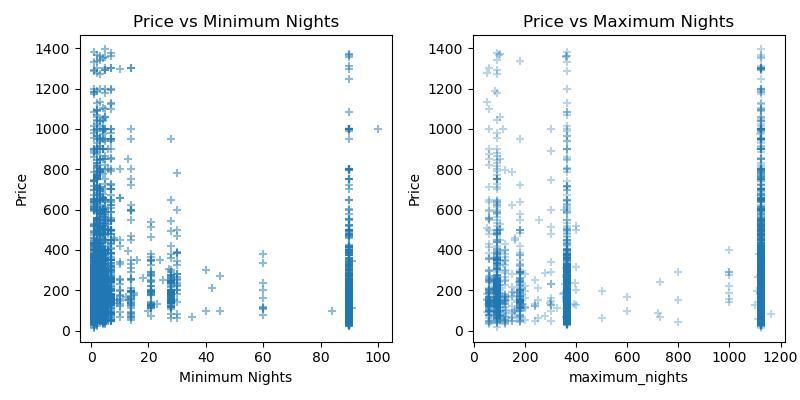
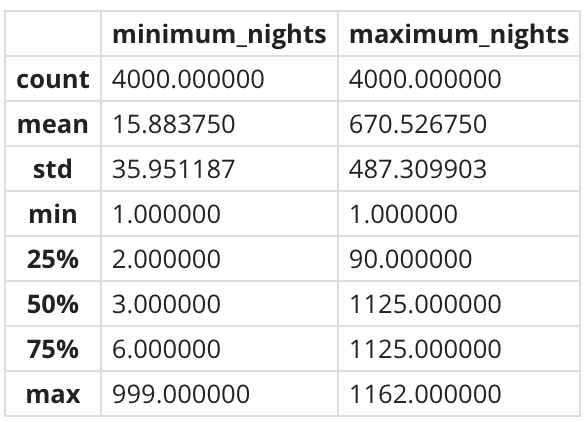
For the neighbourhood section, we decide to analyze the four columns together: neighbourhood, neighbourhood\_cleansed, latitude, and longitude. From the data processing section, we found out that there are over 400 missing values and so many different types for neighbourhood, we decided to use neigbourhood\_clean instead.

Since there are too many categories, we only selected the top fourteen most frequently appearing, and the rest are all combined into one category named “other. “ So, now totalling fifteen categories, as you can see from the figurex on the left side. In this approach, it is also convenient for us to transform this data into dummy variables since the number of categories is much reduced.

Furthermore, we analyze the relationship between price and these categories; the distribution is fascinating, which is shown in figurex above. From the boxplot chart, we can see that although there are outliers in many categories, it still presents a relationship between neighbour and price.

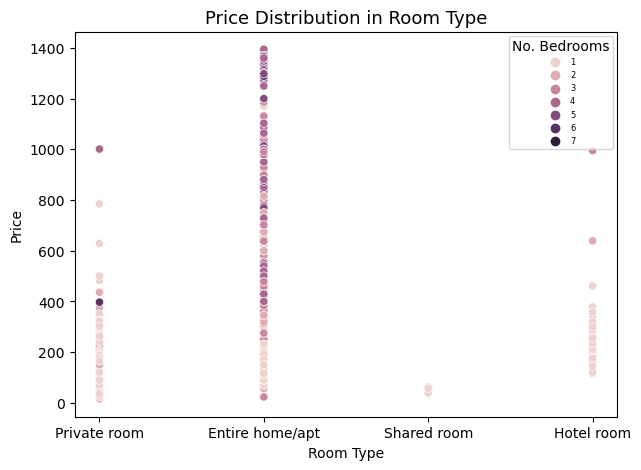
Next, we would move to latitude and longitude, which both have no missing values. Since the types for these two columns are all numbers, we could see their correlations between latitude and longitude with the price. We found that both of them have a close relationship with price somehow. To better analyze them with price together, we have created a new variable named “Distance,” measuring the distance between each location and the Sydney Tower as the datum based on a latitude and longitude of -33.870453 and 151.208755, respectively[(Latlong, n.d.)](https://www.zotero.org/google-docs/?I7erb0). Through this method, we are able to analyze them together.

For both the minimum and the maximum number of nights of a stay, since both of them are numerical values in the dataset. As usual, we first performed the statistical analysis and found some outliers. 

To minimize the impact of these outliers on our analysis, we decided to remove them first. After that, we have run the scatter plots for each versus price. Here, we have excluded data with minimum nights greater than 100 or maximum nights less than 50. As you can see in the graphs above, there is no strong evidence of a clear linear or non-linear relationship between minimum or maximum stay days and price. Therefore, we decide not to use them as predictors in our models. 

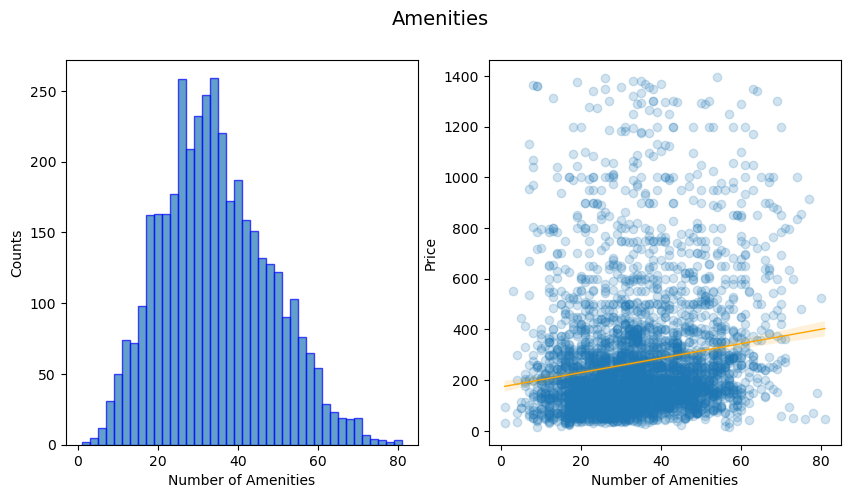
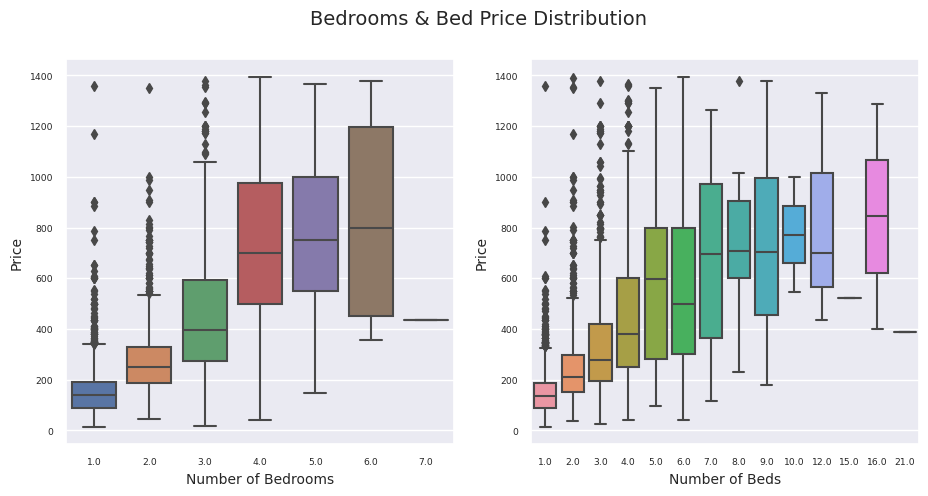
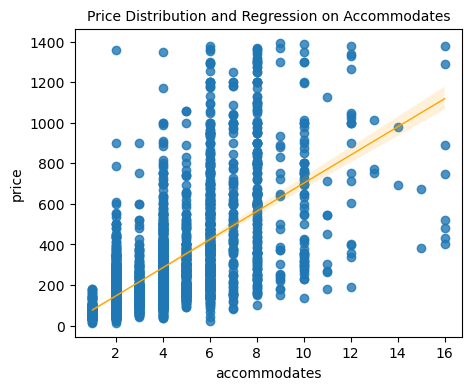
**3.4 Property Condition**

Other than the location of the property, another primary consideration for people when booking a place should be the condition and capacity. To understand and categorize a new property for the price, we divided their property condition into three aspects: type, capacity and the property amenities.

The variable related to type includes 'property\_type' and 'room\_type.' Through the general data processing, there are no missing values in those columns, and both are categorical. After inspecting the values, we found that the 53 unique types of properties are more detailed classifications under the room type. Based on the mean price of different room types, we consider this predictor has the ability to indicate the price level. The scatter plot of price distribution in room type shows that most room types have relatively clear price ranges except 'Entire home/apt.' Further analyzing the subcategory of 'Entire home/apt', it is not relevant to predict price level through variables such as 'Entire home' and 'Entire rental unit,' which might have a substantial price impact because of different capacities and amenities. For example, the legend in the scatter plot indicate the number of bedrooms, where under ‘Entire home/apt’, properties with more bedrooms tend to have a higher price. Therefore, we consider only maintaining the 'room\_type' as a predictor as the type indicator.

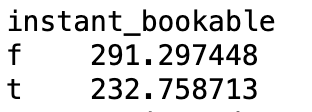
The variable that indicates the capacity ability of the property includes 'accommodates', 'bedrooms' and 'beds', which are all numerical columns. Data processing shows missing values in the bedrooms and beds column but not in the accommodates. To fill in the missing information, since the value of accommodates is complete; therefore, we can assume that as the missing value of beds, which means one bed at least could fit one person. Next, based on the supplemented column of beds, we can fill up the bedrooms column under the same logic, which means one bedroom at least contains one bed. Analyzing the mean price value of different numbers of capacities and some general boxplots indicates a positive correlation between capacity and price overall.

The hypothesis of the impact of capacity on price would be a great positive correlation. From value counts, we could know that majority of the property could accommodate two to five people with one to three bedrooms. The scatter plot between accommodate and price below indicates a positive price trend when the capacity level increases. Furthermore, the box plots also show that as the number of beds and bedrooms increases, the overall price will also be higher. However, one consideration between these predictors is that, with the high level of similarity and filling method of missing value, there might be multicollinearity with accommodates, bedrooms and beds.

The last property predictor is the amenities which contain categorical information. Considering the potential relationship between price and a property's amenities, we assume that rooms or apartments that provide more types of facilities indicate more types of services we could get. Therefore, the corresponding price should also be higher for those properties. Thus, by splitting each amenity with a comma among the categorical data, the new column is switched into numerical data, indicating how many different amenities each property provides. Figurex presents the distribution of amenities among training sets and the relationship with price through the scatter plot and regression line. It shows that most properties provide around 30 to 40 types of amenities and a slight positive correlation with price.

**3.5 Review**

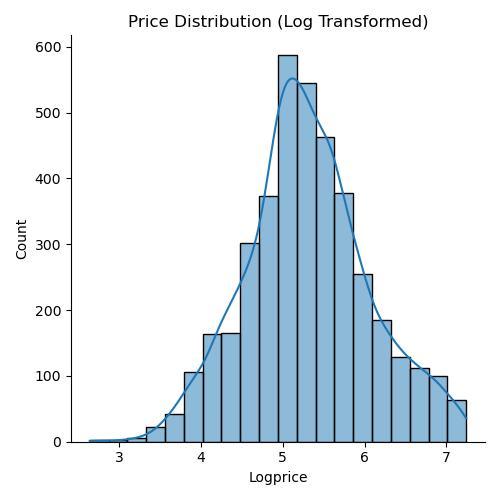
The last group that may impact customers' booking on Airbnb is the review section, which gives feedback to new customers on previous services that have been offered before. For all of the review sections, Airbnb divided the review into seven sections about the rating, accuracy, cleanliness, check-in, communication, location and value. Besides, Airbnb also offers data about the number of reviews and reviews per month to analyze customers' concerns. What's more, the instant bookable is also being considered as an impact to the house price per night.

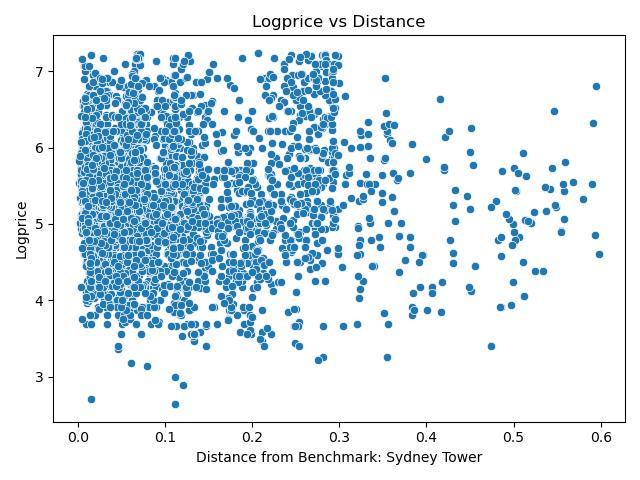
In order to help Airbnb have a more specific analysis of the relation between the price and the review section, there is a table to illustrate the correlation between the price and review variables.

The correlation between the price per night and the review sections is not significantly different from 0, which means there is a weak relationship between these two factors. Still, we found the instant bookable does have an impact on the house price per night; if the house could be instantly bookable it would be about $60 less than the price of that house which cannot be instantly booked.

**4 Feature engineering**

**Log transform for price**

From data processing, we know that price distribution is skewed. Therefore, to better analyze the relationship between variables and response (y or price here) and help linear regression models have a better performance, we consider having the log-transformed for the price column. Below is the log-transformed price distribution, which is tend to under a normal distribution now.

**Create a new location-related variable**

After we went through the correlations between latitude and longitude with logprice, we found that both of them have a close relationship with logprice somehow. In order to better analyze them with logprice together, we have created a new variable named “Distance” measuring the distance between each location and the Sydney Tower as the datum based on a latitude and longitude of -33.870453 and 151.208755 respectively. From the above left figure, we can conclude that the relationship between distance and logpric is not strong as we expected. The same process should also be conducted for test.csv.

**Treatment for missing and irrelevant values**

* Data processing shows missing values existed in the bedrooms and beds column but not in the accommodates. To fill in the missing information, since the value of accommodates is complete; therefore, we can assume that as the missing value of beds, which means one bed at least could fit one person. Next, based on the supplemented column of beds, we can fill up the bedrooms column under the same logic, which means one bedroom at least contains one bed.
* For host\_response\_rate and host\_acceptance\_rate, we used some fixed integer to fill all missing values based on the distribution of each respectively.
* Considering some columns include many missing values, and some variables may also have a weak relationship with price (or logprice), we simply create a dropcolumns list which covers description, host\_since, neighborhood\_overview, host\_location, host\_neighbourhood, 'neighbourhood, neighbourhood\_cleansed, and property\_type. The related explanation is given in the EDA sections.
* The same process should also be conducted for test.csv

**Other important processes**

* For Host\_response\_time and room\_type, through the EDA process, we knew that each had several categories, so we decide to create a dictionary that assigns each category a number starting from 1. This treatment makes it easier for us to build the model later
* For Host\_verifications and amenities, we conduct a for loop for each cell under these two columns and use the number of words contained to overwrite the original content via the split and len functions
* The last but not least, for columns of instant\_bookable, host\_identity\_verified, host\_is\_superhost, we adopted dummy function for them since they are all composed of two types
* The same process should also be conducted for test.csv

**Scaling - Standardisation**

Before Starting building and training prediction models, we also take the scale step for both training and testing set using standardisation method for some of our model, such as Lasso and Ridge regression. By doing that, variables with different scales will be transformed to a more comparable scales, which could help the regularised linear method perform better.

**Preparation for Models**

After we completed all steps above, we would double check several points as follow:

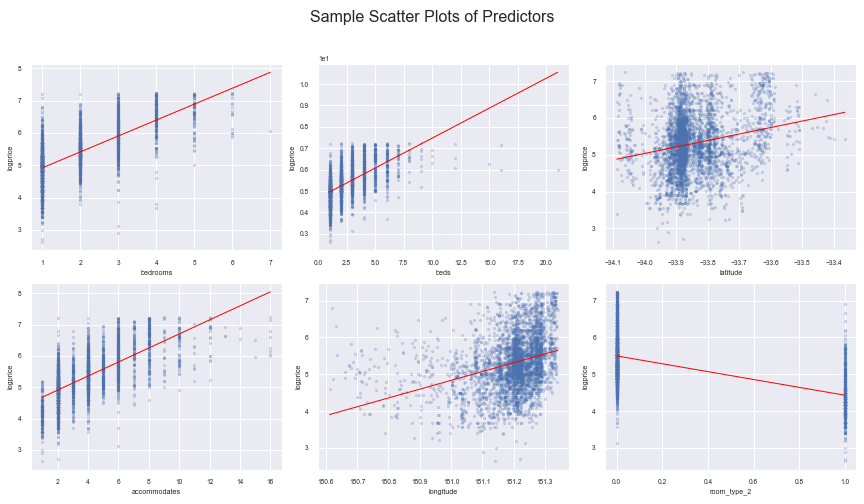
* All columns are same treated by both train and test dataset
* Both datasets will not contain any missing or NA values
* Run a correlation table to check the relationship between all variables and logprice in order, the related heatmap and top 7 of the correlated predictor with log price table will be provided in the appendix

**5 Methodology**

**5.1 Model descriptions**

To finalize an optimized model for Airbnb price prediction, five types of models have been built and trained with the training set, including linear regression, single regression tree, random forest, gradient boosting and model stack, respectively. Here we will emphasize three models; linear, simple regression tree and our final selected model, model stack.

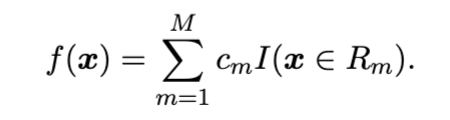
The **linear regression model** is a highly interpretable model that predicts based on the linear trend and correlation between each predictor with the dependent variable. Under the scope of linear regression, there are four different prediction models: OLS, Ridge (L2), Lasso (L1) and Elastic Net, respectively. OLS was used as the baseline model between these models, with the other three models playing as the regularisation method applied above it. To prevent OLS from overfitting with the training set, Ridge and Lasso could penalize on different parameters in two ways. Otherwise, considering with different nature of the dataset, Elastic Net could also be applied, which weighted L1 and L2 with the most appropriate ratio.

Since all of these regressions will provide predictions on different predictors with the parameter that is based on the linear correlation with price, we will overview the top six correlated predictors' linear relationship with price first to have a general idea and hypothesis of the linear regression. Below are the scatter plots with the linear regression of the top six predictors from the price correlation table. It indicates that bedrooms, beds and accommodations have a clear positive relationship with the price. Therefore, we might hypothesize that the greater the listing capacity, the higher the price. Furthermore, it also shows that if the listing is a type 2 room, which is a private room, the price will be lower than the Entire home/apt.

The **regression tree** continuously splits all predictors into partitions and fits in a small model in each region. As the two subsets are non-overlapping, this decision tree could then be interpretable and visualized. We model the response y(in this report is ‘logprice’) as constant in each region and make the same prediction for each observation of this region. We also used the mean squared error as a criterion to fit in the model.

We begin from the top of the tree by separating all the X into a single region. Further down the tree, we add two new branches, the two branches are separately the smaller value of a predictor and the larger value. In each of the regions, we predict the logprice using the average logprice of the training points falling in this region.

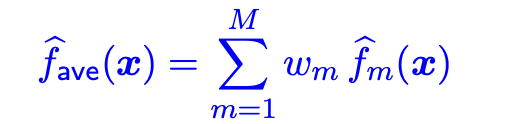
We predict the regression tree using the following model:



Rm here refers to the distinct and non-overlapping regions of predictors and the logprice value is the corresponding constant cm in this model. By controlling the maximum depth and minimum sample leaves of the tree, we can grow the tree to the complexity as we want.As we are using the RSS(Residual sum of square), we can also minimize the RSS through choice of regions.

A small tree might not capture all the important features of the model while a large tree can overfit the data. Therefore, we can also grow a big tree at first and then prune it using tree size as a tuning parameter. Compared with linear regression models, the regression tree is easy to explain and highly interpretable. The parameters can be easily handled, especially in that the categorical data doesn’t need to be transformed into dummy variables. Regression trees can even approximate nonlinearities even for the interactions. There are also some disadvantages of the regression tree. Due to its hierarchical structure, the tree is highly unstable and the variance is relatively high. Along with its lack of smoothness, the predictive accuracy of the regression tree could be low.

**Model Stacking** provides a great improvement in prediction by combining the results of multiple models. In this way, Model stacking often leads to better predictions, but at the expense of interpretability. Interestingly, the metamodel for stacking is linear regression model derived from the concept of model stacking as follows:

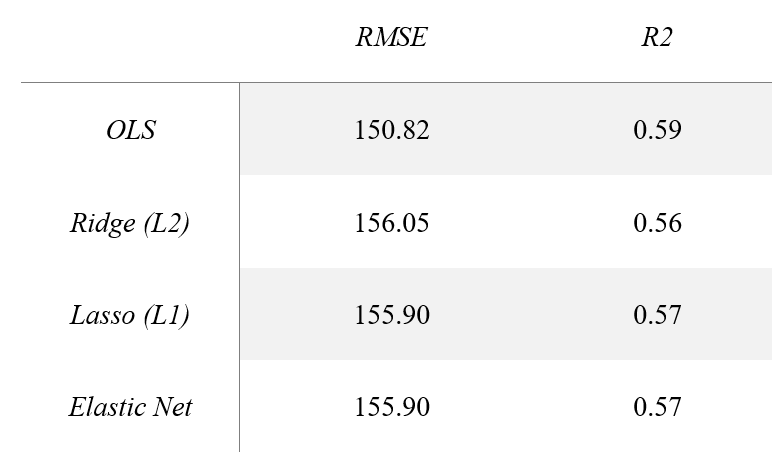
Here, fm(x) refers to the different models and Wm refers to the weights of each model. To avoid putting too much weight on the most complex model, it uses the natural k-fold cross-validation of the leave-one-out method.

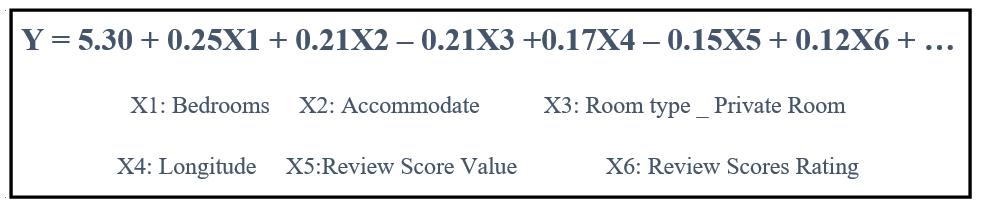
Once the tree set is constructed on the bootstrapped dataset, we generate each bootstrapped dataset by taking the same number of samples from the training data and performing replacement. This means that each dataset will be slightly different from the others, and therefore each tree will be slightly different from the others.

Since we include both random forest and gradient boosting, the overall advantages of Model Stack may be similar to that of the tree-based approach. In this case, we have many different types of data, continuous data for review scores, discrete data for number of bedrooms, some dummy variables, etc. Model Stack can easily handle all these data. It is able to handle outliers in the predictor space as they exist in some of the predictors based on our EDA. Moreover, even though we use normalized data from Ols, the rest or the main part of this model is not sensitive to the monotonicity transformation of the input. Moreover, this model is computationally scalable and has the ability to handle irrelevant inputs.

Furthermore, besides these three models, we also built random forest and gradient boosting as prediction models 4 and 5. With random forest, the selecting parameter and predictor of each layer of trees could be more comprehensive, training the prediction model with all possible related predictors instead of only focusing on the highly weighted ones. Furthermore, since parameters such as maximum depth and learning rate are adjustable, it provides us more flexibility to optimize each model according to the different sizes of the dataset and the nature of predictors.

**5.2 Model 1: Linear regression (OLS)**

As indicated in the feature engineering, four regressions have been fitted and evaluated by cross-validation after standardization on the training and testing set. Table X shows the RMSE and R square figure based on the training cross-validation. It indicates that OLS has the best performance predicting the price of listing among those linear regressions with the lowest RMSE and highest R2 figure, followed by Lasso and Elastic Net, which have similar outcomes. Since Lasso wipes out the less correlated predictors when penalizing OLS, we found that five predictors have been deleted from the mode, including four host-related predictors. Because the training set has only 33 columns, which is not a massive number of predictors, other than the issue of over-fitting, OLS might have a higher ability to predict Airbnb among linear regression models.

Here is the OLS regression equation that predicts the listing price for Airbnb in the Sydney area. Y, as the dependent variable, is the price of the new listing input that we want to predict. The equation here only indicates the top 6 independent variables from the 33 predictors.

From the equation, we could know that based on our hypothesis before; the listing capacity positively impacts the price. For example, with one more bedroom on the listing, the price will increase by 25% (since we have operated the log transformation for price in the feature engineering) and increase by 21% with one more person to accommodate. Similarly, for x3, the parameter indicates that a private room will have a 21% lower price than an entire home or apartment, which is rational. Furthermore, the positive relation of x4 could be interpreted that listing in the East area of Sydney tends to have a higher price. What surprised us is that for x5 and x6, both review score-related predictors, the score value negatively affects the price, which is against common sense.

**5.3 Model 2: A single regression tree**

Firstly we grow a small tree from all the cleaned train data. We decided that the maximum depth of this tree to be 2 and the minimum sample leaf to be 5. Therefore from all the cleaned predictors we retained, the outcome of the small tree shows in Figure 1. From the trained data, we can say that if the number of bedrooms is smaller than or equal to 1.5, then the logprice will be near 5.305. Under the situation that the number of bedrooms is smaller than or equal to 1.5, if the number of room types is smaller than or equal to 1.5, the predicted logprice will be 4.878. The tree then go down with different logprice according to different predictors. In this small tree, we always make the best split for a particular step without considering the consequences lower in this tree. Because we only used a few features in this small tree, it is highly possible that this small tree might fail to use all the important features. However, for a large tree, overfitting will happen with small numbers in each region leading to high variance estimators of the constants. Therefore we also use tree pruning to govern the complexity of the model.

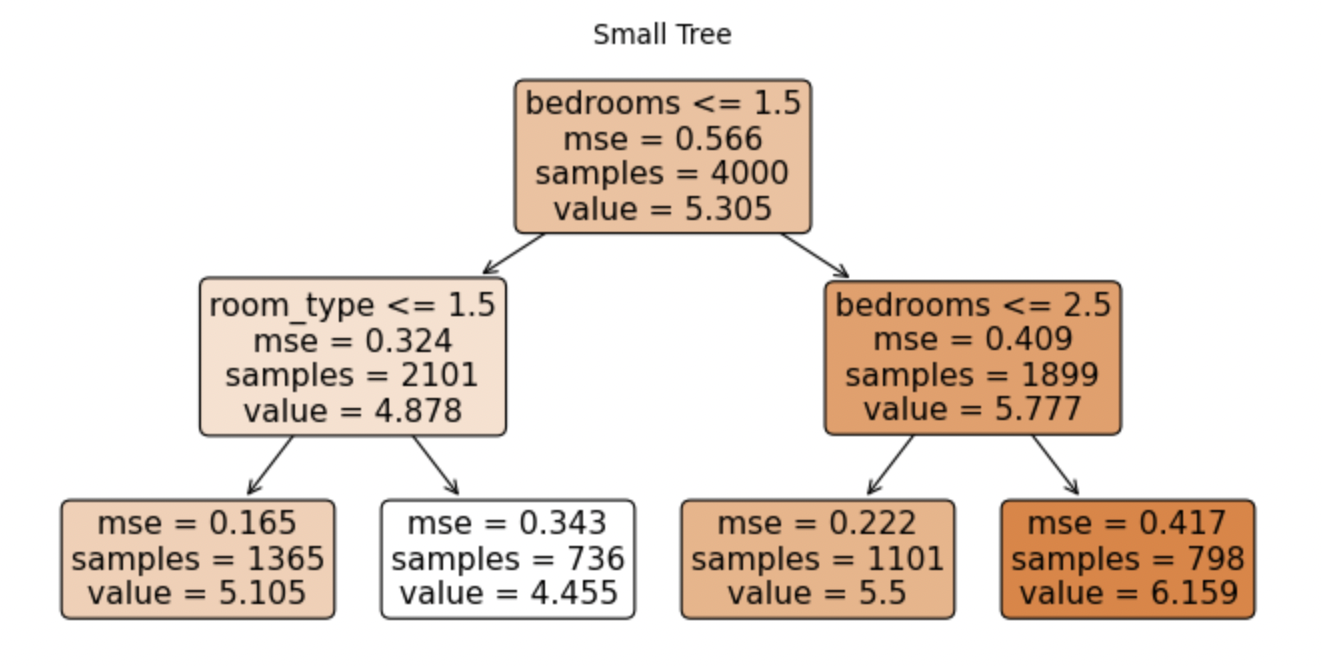


Figure 1 Small Tree

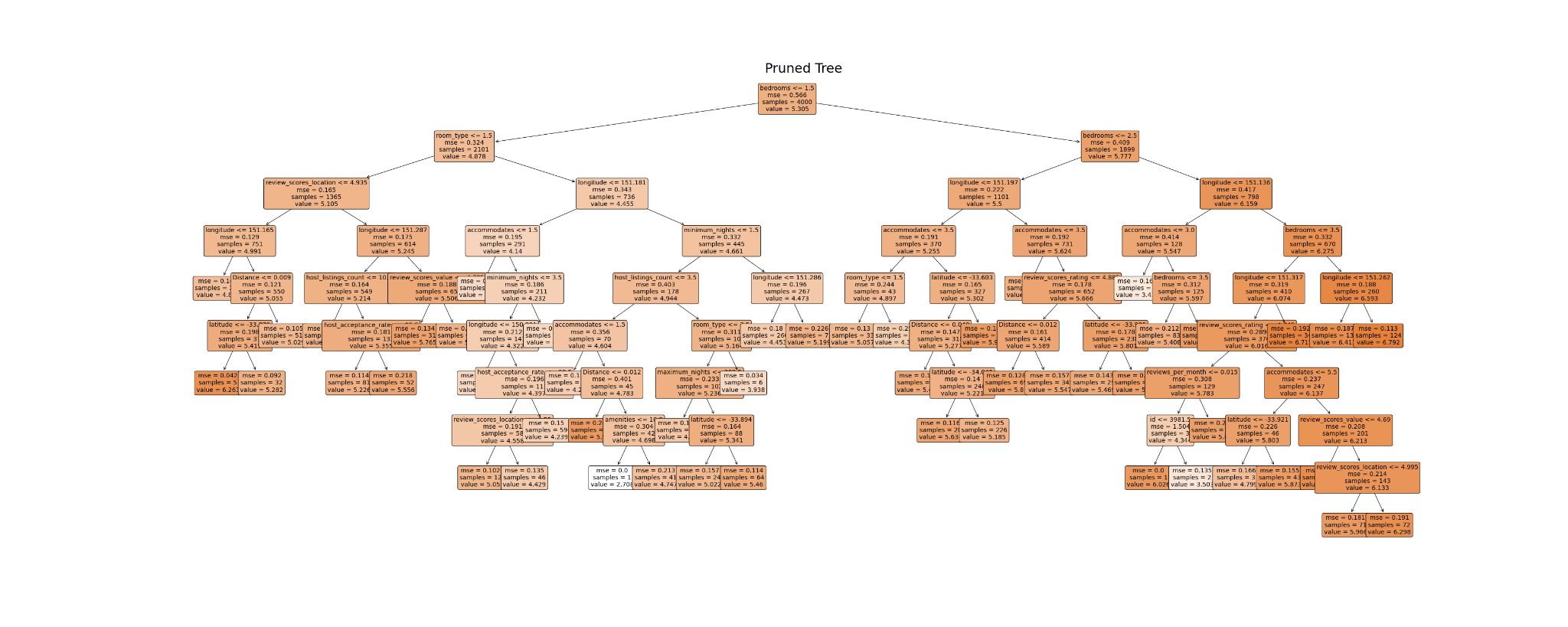


Figure 2 Pruned Tree

We use tree size as a tuning parameter and select it by the data. In this model, we use ccp\_alphas to indicate the tree size and make the minimum sample leaf to be 1. We also choose the best estimator in each step. The outcome of the Pruned tree is shown in Figure 2.

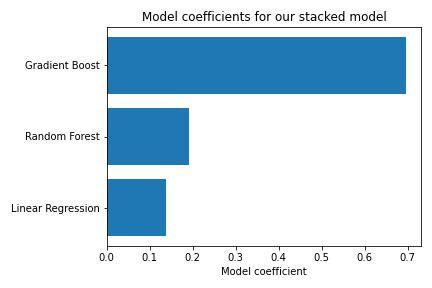
**5.4 Model 3: Model Stack (Model Average)**

To build Ols, first, we have our train and test datasets standardized so that our estimation would be more accurate as we have many predictors which are in different scales. The detail is also mentioned in the above linear regression section.

To build our random forest we use sklearn's RandomForestRegressor since the value of some predictors is continuous. Specifically, we create a random forest with 500 trees (n\_estimators=500). Also, in order to improve the results of our random forest, we will tune hyperparameters: min\_samples\_leaf and max\_features. We defined them as two lists respectively: min\_samples\_leaf: [10, 20, 30], and max\_features: [5,6,7,8,9].

To build Gradient boosting, we import sklearn's GradientBoostingRegressor. Similarly to the random forest, we also need to set tuning parameters for Gradient boosting: learning\_rate=0.05, max\_depth=5, n\_estimators=1000, and subsample=0.5. Moreover, the tuning parameters here are all decided by GridSearch Cross Validation for the best performer.

In this case, we have combined three different models as mentioned above: Ols, Random Forest, and Gradient boosting. To visualize the weight for each model or related model coefficients, we used Booststack.final\_estimator\_.coef\_ function for below bar chart:



As we tried different tuning parameters for both random forest and gradient boosting, we found out that the corresponsed coefficient for each model would be changed. The chart above shows the coefficients for each model in our Model Stack which has great points in Kaggle.

**5.5 Model Assumptions**

**Model**

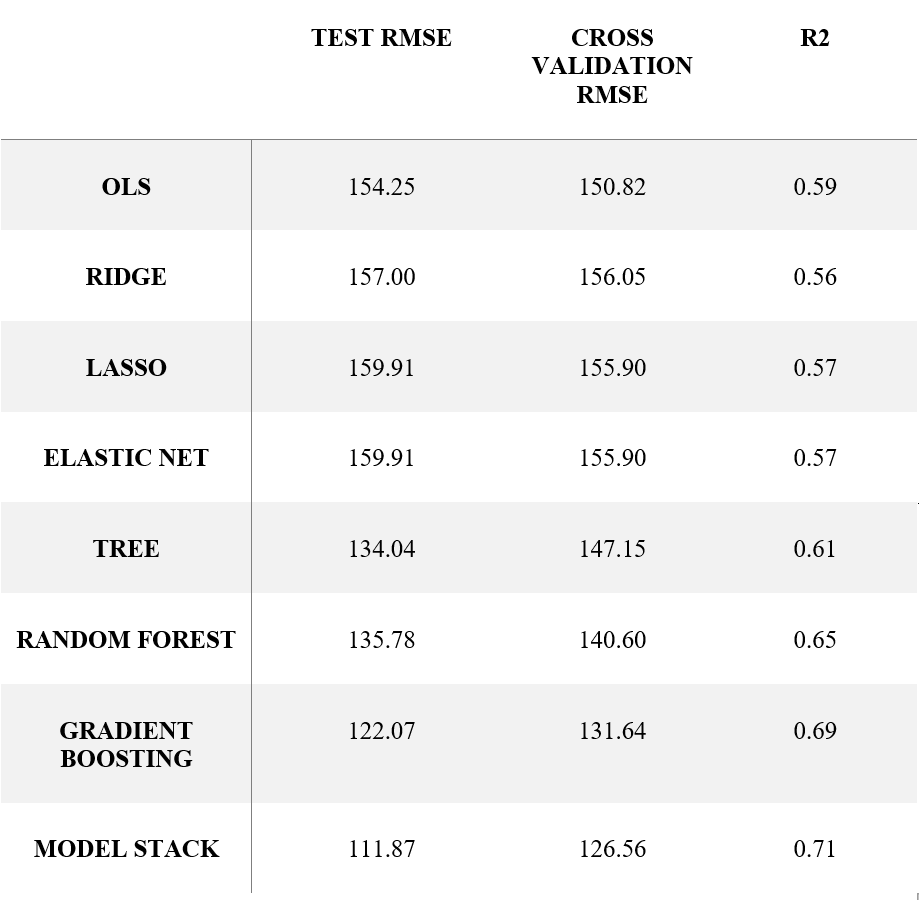
According to JMP(n.d) ,when we use linear regression to establish the relationship between response (price) and predictors, we make a number of assumptions. These assumptions are basically preconditions that should be satisfied before we make extrapolation of model estimates or use the model to make predictions shown as follows:

* Before we adopt the model, we assume that the true relationship between listing price and all predictors available in Airbnb dataset is linear
* The error, the difference between our estimation and true value, is normally distributed
* Within the range of estimation, the variance of error is distributed equally on the two side of line, also known as homoscedasticity
* Each observation is independence from others

In the regression tree, random forest and gradient boosting model, we barely use any assumption because the natural process does not follow the piecewise flat segmentation that is assumed by the tree model [(BigBendRegion, 2021)](https://www.zotero.org/google-docs/?f6TdKe). When implementing the regression tree model, we don’t expect that the true relationship between the price and the predictors follows a linear relationship, but we assume that the price will vary when the value of a predictors is larger or smaller than a specific number.

**Data and industry environment**

Since the five models aim to predict the price of new listings for Airbnb with different types of predictors, our models must assume that all predictors are true and objective. For example, the amenities and review contents are based on reality. Furthermore, the models are trained based on past data, which is the data we collected under nowadays' circumstances. Therefore, we are assuming that the predicting model only has the ability to make considerable and truthful price forecasts under a similar industry environment. For example, during Covid-19, because of the limitation of travel and other stringency policies, the whole operation of Airbnb might also be different due to the pandemic's impact. According to research, the rental price of listings in Sydney has decreased during the pandemic because of the decline of Airbnb activity [(William Thomas Thackway, 2021)](https://www.zotero.org/google-docs/?E735eR). Therefore, such events and factors are not considered within our prediction models.

**6 Validation and comparisons**

6.1 Training and validation scores

Training and validation scores as above.

This set of data is obtained from Cross Validation RMSE, the raw Test RMSE would be lower nine times out of ten. For instance, our best fit model is Model Stack, when we run it in Kaggle, the Test RMSE is 114.77084, still much lower than Cross Validation RMSE.

6.2 Model comparisons and limitations

Obviously, from the chart you can see that Model Stack got the lowest points, in other words, it is the best fit model from the 8 models we’ve tested. The least fit model is LASSO and ELASTIC NET.

The limitations of trees are unstable, non-smooth and not accurate.

At the same time, Gradient Boosting simplifies the minimization problem, getting an approximate solution.

**7 Conclusion**

In order to help Airbnb predict the price, we built five models using different prediction methods: the linear regression, single regression tree, random forest, gradient boosting and model stack. In this report, we detailed the process information from the method we filled in the missing data to grouping the factors that might have an impact on the price per night on Airbnb, and explained three well performed models in detail, which would help Airbnb to have a better understanding of price prediction. What’s more, we compared the assumption with our final model and found the limitations of our models.

We choose our final prediction model by comparing their RMSE, cross validation RMSE and R2, these three measurements are commonly used in model evaluation. The RMSE tells the differences between the true value and the predicted value which means the smaller the better, and we use this method not only in our training data but also in the test data. The R2 is measuring the explanation for our model and this is the higher the better.

In our five models, the model stack gives us the most accurate prediction with the lowest test RMSE, lowest cross validation RMSE and highest R2 by combining the gradient boost, random forest and linear regression.

**8 References**

[BigBendRegion. (2021, April 4). *Which assumptions should be checked for regression tree to validated model?* Stack Exchange. https://stats.stackexchange.com/questions/518167/which-assumptions-should-be-checked-for-regression-tree-to-validated-model](https://www.zotero.org/google-docs/?Ht077D)

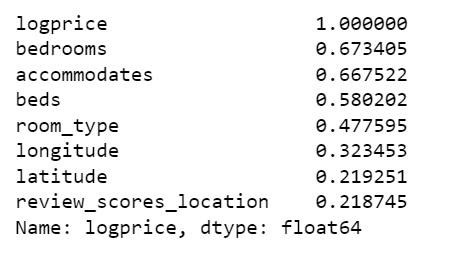
[JMP. (n.d.). *Regression Model Assumptions*. JMP Statistical Discovery. Retrieved November 5, 2022, from https://www.jmp.com/en\_us/statistics-knowledge-portal/what-is-regression/simple-linear-regression-assumptions.html](https://www.zotero.org/google-docs/?Ht077D)

[Latlong. (n.d.). *Sydney Tower, NSW, Australia*. LatLong.Net. Retrieved November 5, 2022, from https://www.latlong.net/place/sydney-tower-nsw-australia-22573.html](https://www.zotero.org/google-docs/?Ht077D)

[Matthew Woodward. (2022, August 31). *Airbnb Statistics [2022]: User & Market Growth Data*. Airbnb Statistics [2022]: User & Market Growth Data. https://www.matthewwoodward.co.uk/work/airbnb-statistics/#:~:text=Airbnb%20Corporate%20Statistics-,Airbnb%20Key%20Statistics%202022,million%20listings%20on%20the%20platform](https://www.zotero.org/google-docs/?Ht077D)

[William Thomas Thackway, C. J. P. (2021, June 4). *Airbnb during COVID-19 and what this tells us about Airbnb’s Impact on Rental Prices*. Urban Findings. https://findingspress.org/article/23720-airbnb-during-covid-19-and-what-this-tells-us-about-airbnb-s-impact-on-rental-prices](https://www.zotero.org/google-docs/?Ht077D)

**9 Appendix**

* Correlation table with log price (Top 7)
* Heatmap of all predictors