2020
S2

Airbnb Pricing Analytics

QBUS6810 GROUP11

Group Member: Mingchuan Dong 490360796

Ryan Lin 490082876

Wency Jin 460373186

Lily Li 490457731

Kaiyu Xiao 490072176

1.0 Introduction

In this project, we are given a dataset regarding Airbnb rentals in Sydney. Our goal is to maximise the income of hosts, property managers, and real estate investors. We aim to present 3 models (best linear model, best nonparametric model and the model stack) which enables the company to generate pricing suggestions for hosts and to assist owners and investors in estimating future Airbnb rental income. To achieve the goal, we set three steps, data cleaning, modelling and model evaluation. As for the data cleaning, our group selected 54 variables from the original data for analysis. Then we classified these variables into 4 sections: categorical, numerical, text, and location to be processed separately. In the modelling step, our group tried to build as many models as we could to see how different model perform on the test data. In the end, after continuous trials of the model, we decided to use the stacked model to predict the price. And from this model together with exploratory analysis, our group found three insights worthy of attention. First, we found out that the more information contained in the text section, the higher price an Airbnb probably has. The underlying reasoning may be that as a host provides more information regarding it listing, it is more likely to attract people and they would like to stay in that listing, and thus resulted with a higher price. Second, the bedroom number and number of accommodation acceptable is also related to price such that the more beds or number of accommodations a host can offer, the higher price the listing would probably have. Third, the location is another key insight. It was found that for listings that are closer to the coast, the price may also be higher.

2.0 Data processing and exploratory data analysis

First, our group observed that the only difference between test data and train data is that test does not have the price column. They have the other same variables. At the same time, our group formulated three principles before processing data. First, everyone cannot delete the row of test data, which will confuse the test results when uploading to Kaggle. Second, the variable processing of train data and test data should be performed at the same time, so as to ensure that the model built from the train data can be predictable for the test data. Third, our group does not want to delete the number of rows of train data. Because we observe that the price in the train data has no missing value, all train data in each row are valuable. And after we classify the data into different sections to be processed separately, deleting a row with one section of data, will cause merging to be difficult later.

2.1 Check dirty float type variables and useless type variables by dabl

To begin our data processing, we started by checking the dirty float type of variables in the data set by using dabl. Only the zip code is considered as dirty. Most of the zip code should be four digits, but there is some other data type such as `NSW 2127`. We will mention how zipcode would be cleaned later in the location part. Secondly, we observed the variables that are flagged as useless by dabl. All values in the `experiences offered` variable is `none`. All values of `requires license` and `is business travel ready` are `f` and that `require guest profile picture`, `require guest phone verification` contains more than 99% of `f` values. These variables will not have a large enough impact and predictivity within the model as they contain similar information. Thus, we deleted these variables from further analysis.

2.2 Inspecting multicollinearity

Next, we considered the problem of multicollinearity and checked whether there is a high correlation between the existing variables. A high correlation will cause multicollinearity in the model which may violate assumptions for certain models. We screened out two variables with a correlation greater than 0.9 and only kept one variable within the dataset. Through this step of filtering, we have deleted 11 variables: `host total listings count`, `calculated host listings count`, `calculated hosting count entire homes`,

`minimum minimum nights`, `maximum minimum nights`, `maximum nights avg night ntm`, `maximum nightum nights max`, `maximum nightum nights max`， `av Availability_90`.

2.3 Cleaning predictors

From dabl we found that price is a free string variable, however, it should be a numerical variable. Thus we delete the `$`and `,` symbol and convert the price variable into `float64`. The same problem occurred for `host acceptance rate` and `host response rate`, thus we deleted `%` and converted them into `float64`. Thirdly, we convert the `host_since` variable into datetime variable. It is also noted that all these changes were applied to both train data and test data.

3.0 Compute for missing values and feature engineering

Before the feature engineering and computing for missing values, our group split the data frame into 4 different parts, categorical data, numerical data, text data, and location data. Our group will do feature engineering separately.

3.1 Categorical

After inspecting the data, we found that the variables "first review", "last review", and "host response time" contains some missing values. For "first review", we dropped the variable because we think it does not provide further information for the data. For "host response time", we filled it with a new category "Not Mention". For the variable "last review", firstly extracted only the year information of it. Then, we categorised it into three different categories, 2019 – 2020, before 2019 and unknown. Here we filled in the unknown for the null values. We manipulated the variable in this way first because we think that computing 0 for null values does not make sense. Furthermore, we think that for last reviews, it is more important about whether the host has been reviewed recently rather than the actual year. For "host verifications", we replaced text information with the number of host verifications completed. Similar methods were applied for "amenities", where we replaced it with the number of amenities that the host provides. For "property type", we found that the variable of "property types" has the problem of a sparse label, thus we merge all classes with little count into "other" category. For "room type", we found that the number of "Hotel room" and "Shared room" is small, thus we merge them into the "other" category. Finally, we convert categorical data into dummy variables to meet the model building requirement.

3.2 Numerical variables

For dealing with numerical data, we firstly dropped 'square feet', 'monthly discount' and 'weekly discount' due to too many missing values existing (i.e. more than 90% of these values are missing). Then, we dropped 'minimum maximum nights' to avoid multicollinearity. The reason of dropping 'host id' is that we consider it as identification and that we think the only possible useful information we could get from it is based on the number of digits, the earlier host may have fewer digits. However, this information is the same as 'host since', thus we dropped it. As for missing values in 'bathrooms', 'bedrooms', and 'beds', we computed the missing values based on the other two categories. For example, we found that the median number of 'bathrooms' for 1 or 2' bedrooms' is 1, so we filled in null values for 'bathrooms' based 'bedrooms'. For null values in `clean fee` and `deposit fee` we filled in null values for 0 as we consider missing value with these variables represents the hosting do not require a clean fee or deposit fee. For all the variables regarding reviews, we filled in null values by using IterativeImputer of other columns. However, it is worth mentioning that such a computing method may possibly result in overfitting as we computed these values based on other variables in the dataset and used it again to predict the response variable. For 'host since', we extracted the year information as we consider day and month information not to be very important. Lastly, after inspecting `host response rate` we found that there are actual values of 0. We think that the

`host response rate` is calculated by the percentage of response. For example, if there are 10 messages and the host replied 10, the response rate would be 100, however, if there are 10 messages and the host did not reply to any, the rate would be 0. For null values, it could be that there is just no message for the host, thus computing null values with 0 might miss subtle information. Thus, after careful consideration, we decided to categorise `host response rate`. This is because we believe that the difference between 1% in `host response rate` may not have a huge impact on price, however, the larger difference may do. Thus, we categorised `host response rate` into bins of 20s and filled in the unknown for null values.

3.3 Text

In the text section, it contains ten variables `name`, `space`, `description`, `neighborhood overview`, `notes`, `transit`, `access`, `interaction`, `house rules` and `host about`. First, after inspecting the data, we found that the variable `name` contains the location of the listing most of the time, thus doing NLP on the name might result in multicollinearity problems with other variables such as local name. Furthermore, words in the variable `name` are generally also in the variable description; thus, we dropped the `name` column from further analysis. Then we also found that description is generally just a collection of all other test variables. Thus, we decided to perform NLP on description, and for the other variables, we will compute them as 0 and 1, to capture whether the host wrote information or not. However, for the variable space, we found that different host may have written positive or negative comments, thus simply converting them into 1 and 0 may miss important information. Thus, we will handle this variable by word cloud analysis and regular expression. Here we fill in `None` for null values first.

3.4 Location

In the location set, we first fill in the missing data in location. In this step, we fill in the data by finding the corresponding `zipcode` by the variable `neighbourhood`, when there is missing data for `zipcode` and `neighbourhood` we filled in the `neighbourhood` with `neighbourhood cleaned` and found the corresponding zip code of that `neighbourhood`. When converting the zipcode data from object format to int format, we check the data in the zipcode and found that in addition to the normal four-digit number, there are several wrong formats in zipcode, such as 2094.0, 2216/n2216, NSW2025, 2037 2020. We corrected these incorrect formats by querying other correct zipcode data. In the next step, we perform zipcode data classification, which helps us to explore the data of different regions in EDA. According to the Local Government Areas of Greater Sydney, we divide zipcode into 9 categories, namely City & Inner West, Eastern, Endeavour, Greater Western, Harbourside North, Northern Beaches and Upper North. The remaining remote area zipcodes are classified as Other as the eighth category. In this case, the area we are exploring is the Greater Sydney area, and the postcode of the Sydney area should not be greater than 3000, but a Victorian address and a Queensland address appear in the data set, so these two are additionally classified into the ninth category, 'Unknown'.

4.0 EDA and more Feature engineering

4.1 Response variable

The first step of our EDA is to inspect the response variable. We plotted the distribution of price and found that it is highly positively skewed. Thus, we computed log transformation on price and plotted its distribution.

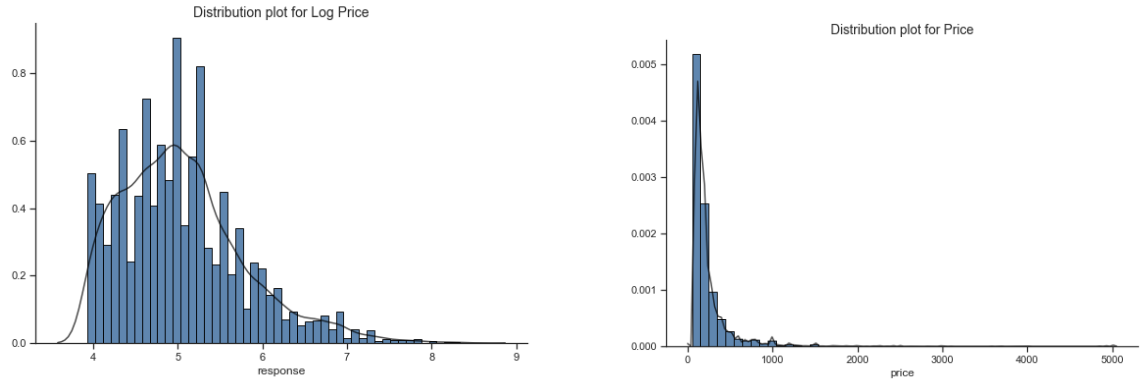


Figure 4.1.1 and 4.1.2

4.2 Numerical

As for the analysis of numerical data, we firstly drew a picture to see the correlation between each variable to roughly understanding the relationship between them (fig 4.2.1). We also found that variables `accommodates`, `bedrooms`, `beds`, `bathrooms`, `guest included`, `host listing count`, `availability 361` and 'calculated host listings count private rooms' are correlated with the response variable with a correlation above 1 in absolute values. Then we drew distribution diagrams (fig 4.2.2 in appendix) and scatter plots (fig 4.2.3 in appendix) with the response variable for all variables to inspect the distribution of these variables and its relationship with response variables. From the distribution plots, we found that for some of the variables, it is highly skewed, thus would require transformation. From the scatter plots, we discovered that 'reviews_score_rating' has an interesting relationship with the response, the higher rating related to higher response rating. It means that the owners of higher reviews score rating are more willing to the response. Besides, the relationship between deposit fee and cleaning fee is similar. The majority number of fees distribute in a fixed interval and the lower price of fee usually relates to the higher response. People who own a house requiring a lower extra fee will be more willing to response customers. We also checked probable outliners in our variables. We found that there for `maximum_nights`, there are two values of 10000 and 9999. We think that these two values are errors as 10000/365 is around 27 years which does not make sense; thus, we will drop these values later when merged with other variables. Furthermore, there are 4 outliners in 'bathrooms', 'bedrooms' and 'beds', after careful inspection regarding the data, we considered two of them as error and dropped it. Lastly, we drew 3 diagrams (fig 4.2.4, 4.2.5 and 4.2.6) to show the relationship between average price and 'beds', 'bedrooms' and 'bathrooms' to estimate whether there will be a relationship between them. We found that a greater number of beds is related to a higher price. This is the same for the number of bedrooms and bathrooms.

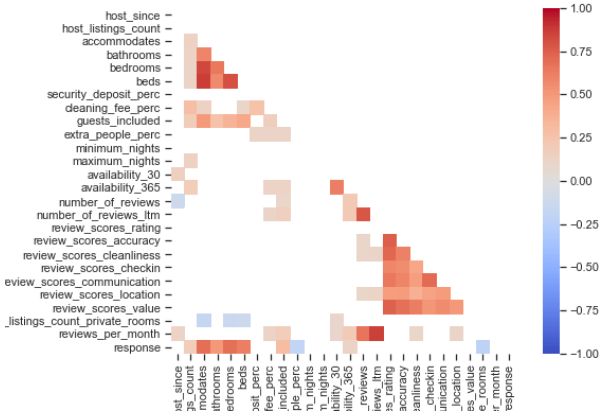
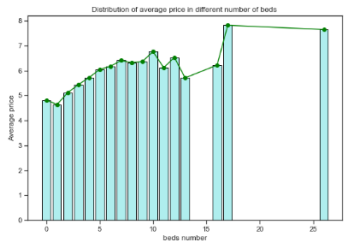


Figure 4.2.1



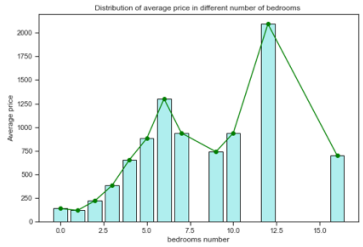


Figure 4.2.4, 4.2.5 and 4.2.6

4.3 Categorical

For "host response time", "within a day" has the highest average rental price at 267.17, while "Not Mention" and "within an hour" has the lowest average rental price, at 202.41 and 215.76 (Table 4.3.1). By inspecting the boxplot (fig 4.3.1), we could see that the distribution is quite similar for all the categories, where "within a day" has the lowest variance that would be the reason why "within a day" would have the lowest average rental price. For "host is superhost", the mean and median of the rental price for superhost are slightly lower than the normal host, and the variance is larger (Table 4.3.2). This is quite odd here, however, we did not find the underlying reasoning behind it. For "host verification", the correlation between "host verifications" and the rental price is about 0.0113, and the average price does not show any obvious trend (fig 4.3.2). Therefore, we think "host verifications" may not affect the rental price. For "is location exact", the mean rental price for the exact location would be higher than those are not exact (Table 4.3.3). It is really interesting that on average people would like to pay more to rent a house without a clear address. For "amenities", the correlation between "amenities" and the rental price is about 0.1247, and we could find an increasing trend of rental price about the number of amenities in Figure 4.3.3. Therefore, the more amenities host could provide, the higher rental they may receive. In fact, most Airbnb houses have many kinds of amenities, but their host does not put them into their rental description. For "cancellation policy", we found the problem of a sparse label with the cancellation policy, thus we categorised it into three categories, "strict", "flexible", and "moderate". From Table 4.3.4, we could find that Strict cancellation policy would have the largest mean and median of the rental price. Although flexible cancellation maybe for favourable for people, it is not associated with a higher price. This could be due to that strict cancellation policy may be related to listings that are more popular and higher price. Hosts who own more popular Airbnb houses would have more concern about the cancellation of guests. This is because their houses are so popular that it needs a booking in advance, and if a customer cancels, it may result that there is no guest to check-in during this time, which would lose the money they could have been earned. Therefore, more popular houses are more likely to take a strict cancellation policy. For unpopular houses owners, they do not have so many advance reservations, thus cancelling may lead to loss of profit much lower than those with strict cancellation.

host_response_time	count	mean	std	min	25%	50%	75%	max
Not Mention	4472	202.41	222.87	51	80	131	231	5000
a few days or more	185	261.38	362.63	51	74	121	275	2999
within a day	674	267.17	371.06	51	85	144	250	4000
within a few hours	836	264.19	391.84	51	89	149	249	3580
within an hour	4464	215.76	259.27	51	100	150	220	4013

Table 4.3.1

host_is_superhost	count	mean	std	min	25%	50%	75%	max
f	9019	218.56	270.65	51	89	144	231.5	5000
t	1612	214.93	263	51	100	150	223	4000

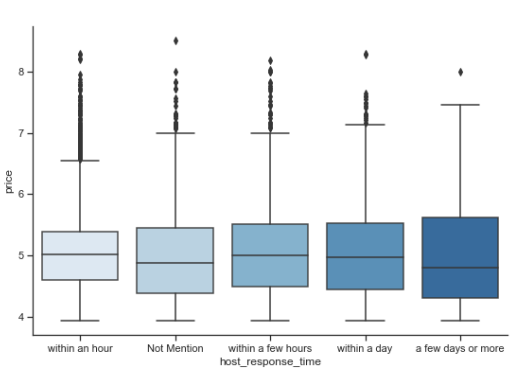
Table 4.3.2

is_location_exact	count	mean	std	min	25%	50%	75%	max
f	2602	220.13	319.42	51	85	140	220	5000
t	8029	217.32	251.21	51	92	147	235	4013

Table 4.3.3

cancellation_policy	count	mean	std	min	25%	50%	75%	max
Strict	4752	269.53	326.1	51	115	170	290	4013
flexible	3213	176.4	216.63	51	74	115	199	5000
moderate	2666	176.32	187.07	51	86	129	199	2414

Table 4.3.4



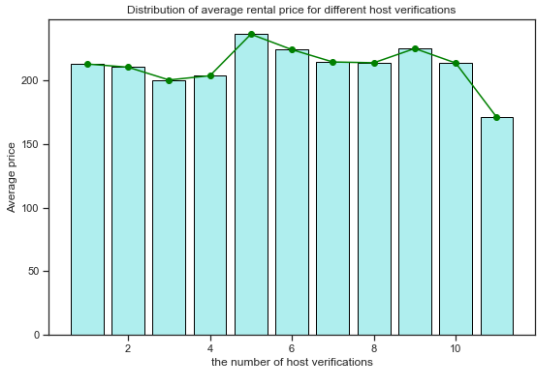


Figure 4.3.1 and Figure 4.3.2

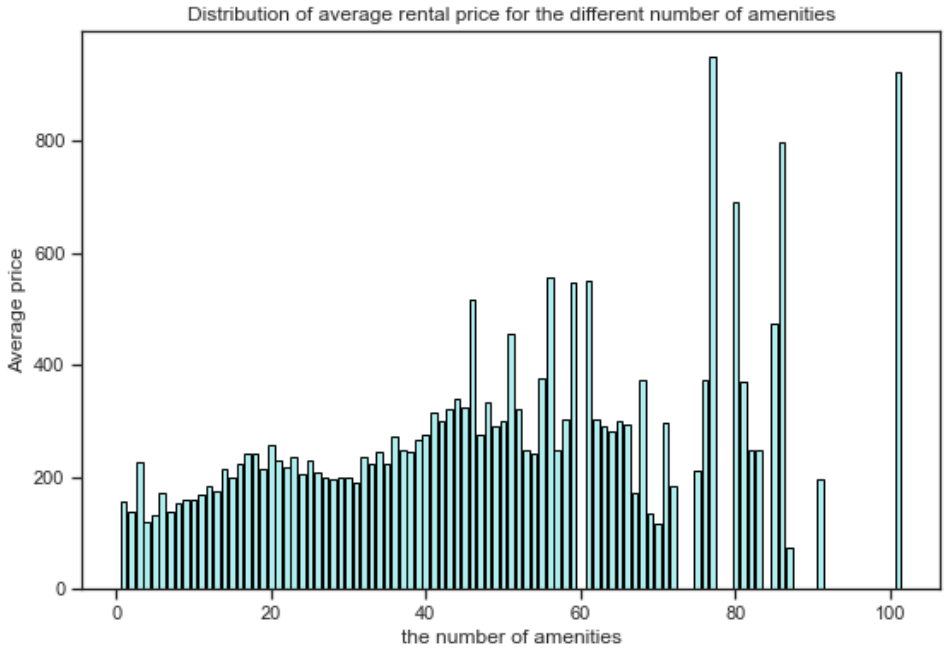


Figure 4.3.1

4.4 Text

Since there are 10635 prices from the train set, we found the first third price range and did a word cloud for the top first third space. The high price listings normally contain the words such as `modern, enjoy,

waterfront, luxury, stylish, living, new, designed, brand, styled, beachside, lounge, huge, garden, prime, balcony, security, house`. So, for word analysis on `space` by using a regular expression, if the same words appear in the `space` sentence, we would classify it as high price words. Our group also did a word cloud for the lowest first third space. We found the interesting thing is that the lowest price Airbnb does not contain any negative words. Mostly, `welcomed, guests, necessary` are more frequently in low price space range.

Figure 4.4.1 Word cloud for space split by top third (left) and bottom third (right) price range

For the description, our group also found an interesting thing after generating a word cloud. In the description, we found that most host or owners would like to describe how many bedrooms in the house; how many minutes away can be achieved where or whether this is an apartment or house. Maybe this information can dress more attention form the customers. For all other boolean variables computing above, we plotted a bunch of boxplot chart. By observing these charts and descriptive statistics. We found if there is a value in these variables (e.g. rules, interaction etc.), houses are generally higher on price.

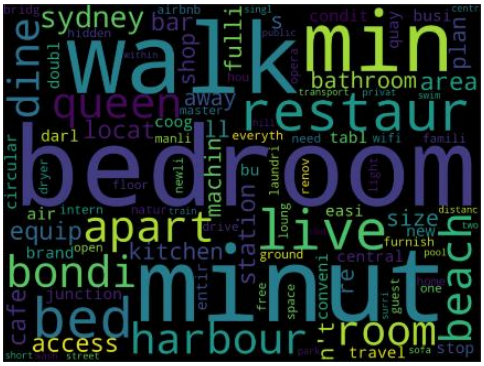


Figure 4.4.2 Word cloud for description

4.5 Location

In the exploratory data analysis of geographic location, we selected the first 1000 rows of data in geographic location as a sample to observe the impact of geographic location on price. We found that in the above map, the density of points represents the central tendency of the house/apartment, and the darker the colour of the points, the higher the price of the house. From this we can find that in the Sydney city central (CBD), the southeast coastal area and the northern coastal tourist area have a relatively high concentration of houses, and the prices are relatively high, showing dense dark purple points on the map. There are fewer houses in inland areas, and the prices are much lower than those in coastal areas and CBD areas. Overall, Sydney's Airbnb rentals density and prices are decreasing from the coast to the inland. In terms of geographic location, City & Inner West, Eastern, Harbourside North and Northern Beaches are the top four

in house prices. This also shows that Airbnb is indeed higher in the CBD, coastal areas and northern tourist areas.

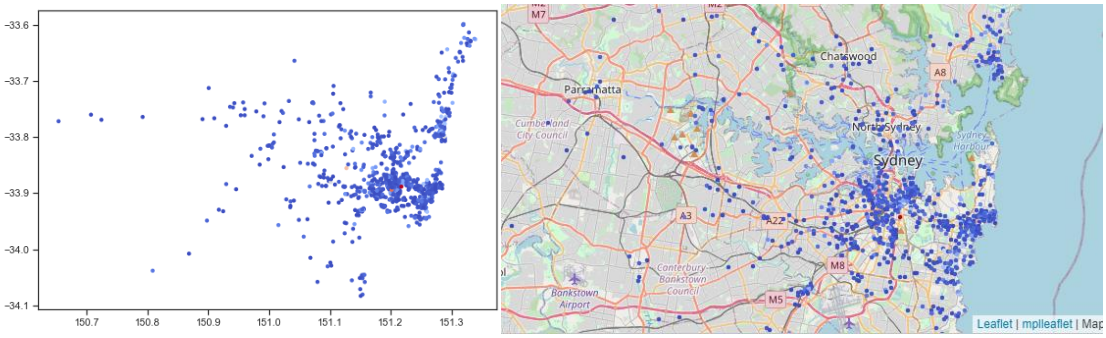


Figure 4.5.1 and 4.5.2

5.0 Methodology (Three model)

Linear Regression

Linear regression is a technique to analyse the relationship dependent variables and independent variables. Our team chose a linear model based on three assumptions. First, the housing price has a linear relationship with other variables. Second, the dependent variable values are independent of each other. Third, the housing price as the dependent variable conforms to a normal distribution. The main reasons for choosing a linear regression model are as follows: First, the process of modelling does not require very complicated calculations, and it runs fast even with a large amount of data. Second, the understanding and explanation of each variable can be given according to the coefficient. The linear regression model can intuitively express the relationship between independent variables and dependent variables. The parameters in the linear regression model objectively express the importance of each variable in the prediction, so the linear model has good explanatory properties. In this case, we import the plot_coefficients function and use plot_coefficients to find out what are the 20 variables with the largest coefficients in x_train. According to the plot_coefficients, longitude and latitude take the largest coefficients in the linear regression model, which means that geographical location is the most important variable in linear regression models. Furthermore, we also make residual plots. From theory, if the points within the residual plots are roughly symmetrically distributed near the x-axis with roughly constant variance (constant variance), it may be reasonable to use linear regression. In our residual plots, it can be seen that our residual map matches the residual map that assumptions of linearity hold.

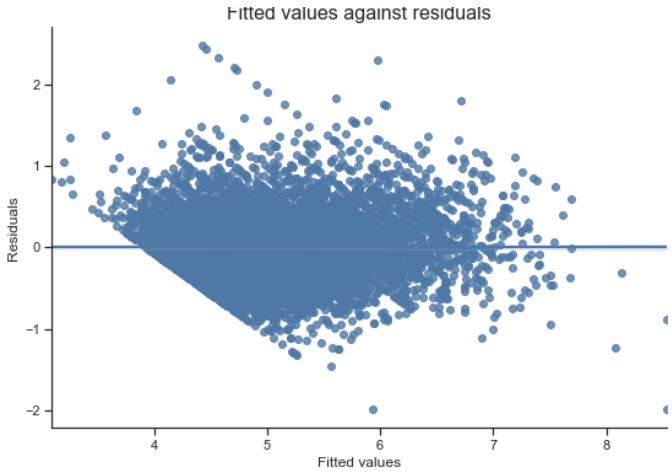


Figure 5.0.1

5.1 Lasso

The main reasons for choosing a lasso regression model are as follows: First, Considering that there may be an approximately linear relationship between the independent variables of our data, in this case, it will have a great impact on the regression analysis, so we choose lasso regression, which can reduce the effect of collinearity through adding penalising terms to the regression optimisation function. This is very necessary for our data set. There are too many similar independent variables in Airbnb's data set. Second, the advantage of the Lasso over ridge regression is that it performs better in compressed variables such that its penalised coefficients more. For ridge regression, it only decreases the coefficient in order to penalise, however, for Lasso, it sets the coefficient of some variables to zero. In the Airbnb data set, Lasso can help us directly clean up variables that the algorithm considers useless. This helps to see more effective independent variables in predicting Airbnb's housing prices. Although the least square estimation is an unbiased estimation, it often has a large variance when there are too many variables. Although Lasso is a biased estimate, while introducing a certain bias, it may be able to greatly reduce the variance of the estimate, thereby reducing the overall MSE. Here we have limited sample information of Airbnb, so we want to use limited information to estimate too many coefficients, at this time, Airbnb's information is likely to be insufficient, so use Lasso to shrink variables is necessary to improve the estimation effect. In this part, lasso regression shrinks 146 coefficients to zero, because these variables are not important, and 26 variables are selected and kept in the model. In this case, according to the plot_coefficients of lasso regression, different from the linear regression model, accommodates has the largest coefficients, and the number of bedrooms has the second-largest coefficients, and then longitude as location variables get the third largest coefficients. As a result, accommodates is the most important variable in lasso regression models.

5.2 XGBoost Model

XGBoost is an ensemble machine learning algorithm based on the decision tree, which could use the gradient boosting framework. The assumption of XGBoost is that encoded integer values for each input variable have an ordinal relationship. Our team decide to choose this model because its artificial neural networks tend to outperform all other algorithms or frameworks in predicting problems with unstructured data, and it is used widely in Kaggle competitions. Besides, the implementation of XGBoost is very fast, which would be very suitable for large sets of data like our Airbnb data. As for model fitting, our group decide to adjust these four parameters of XGBoost Model, learning rate, the number of estimators, max depth of decision tree and subsample. Learning rate is the rate at which the model learns each time. Low learning rate would more accurately find the optimal solution, but it would cost a long time. Max depth of the decision tree depends on the complexity of a model, and increasing the max depth may result in overfitting. Subsample is the subsample ratio of the training instances, which is set to avoid overfitting. Our group firstly processed a randomised search to get the best parameters of the learning rate, the number of estimators, max depth of decision tree and subsample. The search range of learning rate is 0.005 to 0.1, the search range of the estimators' number is 100 to 2501, the search range of decision tree max depth is 2 to 8 and the search range of subsample is 0.5 to 1. After searching, we found that the best learning rate for our model is 0.0065, best max depth is 7, the best number of estimators is 2029, and best subsample is 0.7723. Then we train the XGBoost model based on these optimal parameters. We found that "bedrooms" has the highest feature importance in this model, and "accommodates", "cleaning fee perc", "longitude" are also important features.

5.3 Model stacking

Model stacking is to combine predictions from multiple learning algorithms as our final model. Comparing with the other models, this model usually achieves the best generalisation performance. The first step of this model is to predict all the initial training data into M models. Then the predictions getting from these

M models will be combined into a matrix. This matrix is new training data. Next, using new training data to fit a linear regression model and make a prediction on test data later. By using this model, different models are combined, and the final model will be more complex, and the performance will be usually better. Our group choose to combine six models we used before. It performs linear regression of our 6 models according to different proportions. Finally, we can see that the predicted values obtained by different models have different proportions in our final model, but their overall proportion is 1. From this result, we got our model that Gegre Rdidge accounts for 53.82%, enet accounts for 530.54, Lasso accounts for -531.00, rf_search accounts for -25.61%, xbst accounts for 86.94% and lgb accounts for 31.69%. These 6 models account for different proportions in the final model, and the combination of their different proportions forms the final model. The reason for choosing this model is that we hope to combine our separate models and get a better performance one. Usually, model stacking will perform better than the separate model. Meanwhile, the MSE of model stacking is the smallest one within our models. It means that this model has the best prediction ability comparing other models we selected. For forecasting the housing price in Airbnb, this model could refit a model to make a secondary prediction of the predicted values obtained by the previous models. This step may allow us to get the more accuracy predicted housing price from our independent variables. Furthermore, although this model stacking may be hard to interpret, its predictability is much higher, thus for the objective of the report, that is to help the Airbnb company to generate pricing suggestions for hosts and to assist owners and investors in estimating future Airbnb rental income, we think that predictability is more important than interpretability here.

6.0 Model validation

Model Name	Linear regression (benchmark)	Lasso regression model	XGBoost	Lightgbm	Model stacking
RMSE	0.38253	0.3817	0.31954	0.32099	0.31759

7.0 What are the best hosts doing? (Data Mining)

First of all, we found from the dummy variable of the text that as long as the homeowners add content to the `neighborhood_overview, notes, transit, access, interaction, house_rules and host_about` variables, the average house price will be higher than those homeowners who do not write content. Interesting things is that even for the house rule, our group initially guessed that the content of the house rule might reduce the house due to the increase in the restriction on the use of the house. Then we found that it was actually contrary to our conjecture through modelling. In the model, although the importance of each of these variables is low, they are also data that cannot be underestimated when they are accumulated. From this phenomenon, we infer that, for homeowners, the easiest way to increase housing prices is to fill in each housing information carefully. Even if there is no content that can be filled in, the host can fill in some welcome information lines statement. Because we have observed that some of the `house rules` are no real restrictions, but some welcome statements show the host's love for customers. This method does not need to increase any output for the homeowner, and at the same time, it is possible to let customers know their renting house through more words. It is the easiest way to achieve the purpose of attracting tourists. Similarly, for the content of the description, we found that the content that can be filled in this variable is very wide. Some people fill in the information about bedroom information, and some fill in the distance to the nearest station, or even the nearest restaurant. For this description section, our group recommends that the homeowner fills in the information more detailed the better. The more content included, the more beneficial it is to the increase in house rental prices.

According to our XGBoost model, we could find that "bedrooms" and "accommodates" are the most important features. Therefore, our group focus on analysing the relationship between Airbnb rental price,

the number of bedrooms and the number of accommodates. We found that the correlation between bedroom number and price is 0.6836, and the correlation between accommodate number and price is 0.6953, which means they are highly correlated with the price. The following tables show the basic description in the different number of bedrooms and accommodate. As can be seen, both tables show an increasing trend. For bedroom number, the average rental price for 1 bedroom is the lowest, while the house with 6 bedrooms has the highest rental price when removing the influence of outlier from the picture. For the number of accommodate, 1 accommodate has the lowest average price, and there is the highest average price of 10 accommodate. To sum up, bedroom and accommodate are both about the number of guests, so our group think an increase in the number of occupants would increase rental income for Airbnb hosts.

bedrooms	count	mean	std	min	25%	50%	75%	max
0	641	138.39	112.5	51	95	121	150	1700
1	5476	118.37	72.24	51	70	100	149	1207
2	2690	219.45	140.39	51	141.5	194	250	2414
3	1003	382.86	291.12	51	199	301	450	3953
4	582	654.84	519.11	51	321	500	799	3700
5	196	880.69	681.93	63	450	700	1000	5000
6	32	1300.94	804.1	301	830.25	1190	1562.5	3968
7	7	938.71	518.27	327	583.5	999	1100	1878
9	1	740	NaN	740	740	740	740	740
10	1	938	NaN	938	938	938	938	938
12	1	2094	NaN	2094	2094	2094	2094	2094
16	1	700	NaN	700	700	700	700	700

Table 7.1

accommodates	count	mean	std	min	25%	50%	75%	max
1	717	84.47	43.99	51	60	70	95	386
2	4348	117.45	71.18	51	74	100	140	1000
3	734	150.29	84.73	51	100	130.5	179	1000
4	2200	215.21	145.98	51	138	180	250	1725
5	616	260.94	190.12	55	150	199	301	2001
6	976	343.53	265.39	51	186	255	400	2414
7	224	416.21	257.27	68	237.75	350	500	1892
8	447	634.42	540.88	74	260	496	799	3953
9	97	656.69	627.27	125	287	496	799	4000
10	158	905.7	794.79	65	370.5	700	1100	5000
11	26	663.15	583.27	80	211.5	400	983	2187
12	43	746.47	584.66	77	372	500	931	2999
13	5	585.6	407.61	249	388	500	500	1291
14	15	778.93	509.76	268	338.5	500	1190	1800
15	3	179	120.53	60	118	176	238.5	301
16	22	840.45	646.53	170	325.75	700	1105	2495

Table 7.2

In order to study the relationship between the number of occupants and the rent even further, our group decided to analyse the influence of bed number that is a variable related to the number of occupants. From the following table, it is a basic description of a different number of beds. We could find that although there are some small fluctuations, the rental price also shows an increasing trend with the rising of bed number. Therefore, it also indicates that the house which could live more occupants could receive more rental. Increasing the number of bedrooms and beds both could raise revenue, but the cost of adding a new bedroom would be much higher than buying a new bed. Our group would like to advise the Airbnb hosts that you could choose to add more beds in one bedroom to increase the occupant number, which may increase revenue with the lowest cost. However, it does not mean that the host could put 5 or 6 beds in one room, which may cause a decrease in the living experience of guests and then receive some negative comments.

beds	count	mean	std	min	25%	50%	75%	max
0	199	154.05	148.43	51	80	115	159	1500
1	4998	117.35	71.97	51	70	100	144	1000
2	2513	195.37	144.18	51	121	162	229	2001
3	1323	283.35	247.39	51	150	208	315	3953
4	791	408.16	393.01	51	179	299	500	3100
5	424	561.15	494.6	51	249	394	750	4013
6	199	634.5	533.93	60	301	450	775	3700
7	82	787.95	603.19	100	350	517	1000	2999
8	54	817.31	917.85	77	350	576.5	1000	5000
9	15	663.2	344.91	265	419.5	680	789.5	1500
10	11	1082.73	791.42	301	595	857	1350	2999
11	12	579.92	478.25	170	273.5	419	712.5	1878
12	4	823.5	526.12	275	504.5	759.5	1078.5	1500
13	1	301	NaN	301	301	301	301	301
16	3	606.67	465.76	280	340	400	770	1140
17	1	2495	NaN	2495	2495	2495	2495	2495
26	1	2094	NaN	2094	2094	2094	2094	2094

Table 7.3

We classified these houses by zip code and checked their median house price. The most expensive area is Northern Beaches, where is boarding the sea, and the price is 204.5. Besides, the second-highest housing prices are City& Inner West and Eastern, and the price is 144. Meanwhile, Eastern is a region very close to the sea, City and Inner West is the central area of Sydney. So, it could be inferred that the housing price in the city centre and near the sea is relatively high.

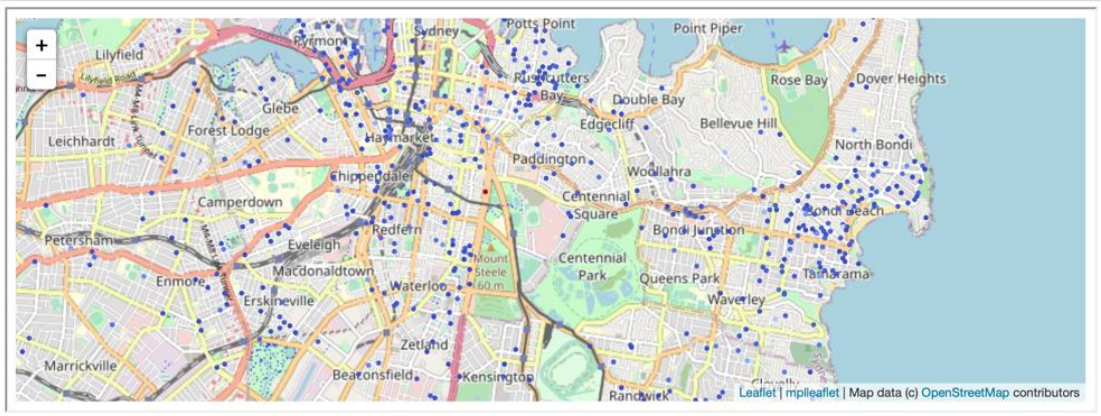


Figure 7.1

We also used latitude and longitude to build a map to show housing price distribution in Sydney. The darker point on the map indicates the higher housing price. It is easy to find that the higher price places are concentrated in the city centre or distributed along the coastline such as the Hay market and Bondi beach. The result is consistent with the result we got from the zip code classification above. Consequently, we could make a prediction that the house on the Airbnb near the city centre or the sea will have a higher price..

Appendix

Fig 4.2.1

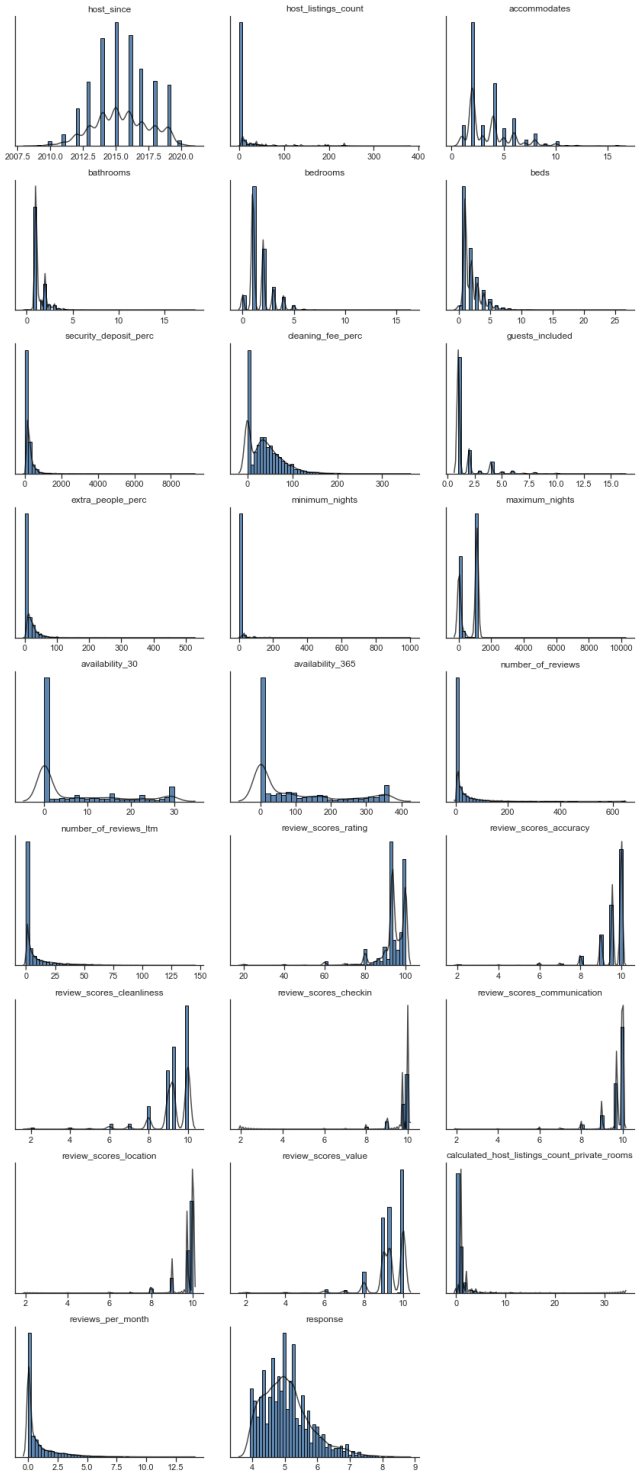
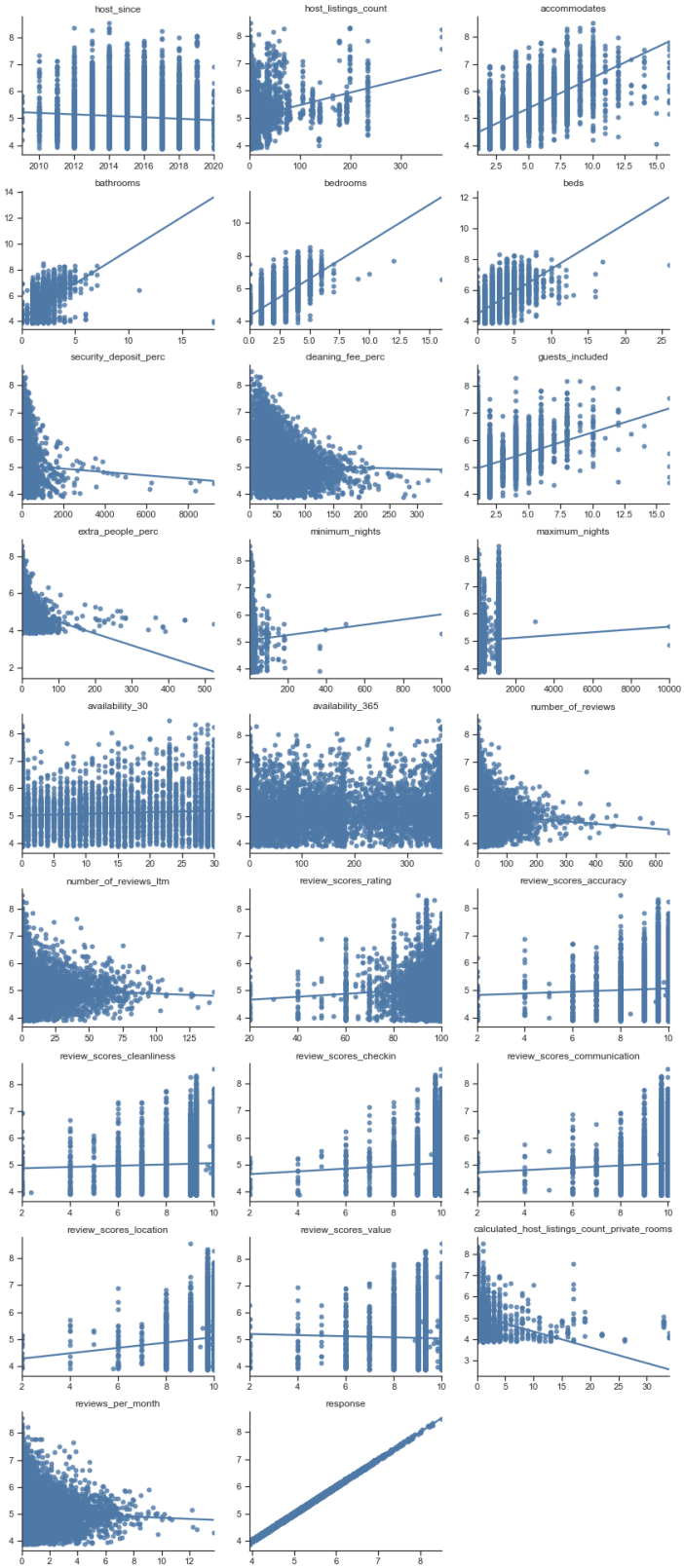


Fig 4.2.3


host_since
host_listings_count
accommodates
bathrooms
bedrooms
beds
security_deposit_perc
cleaning_fee_perc
guests_included
extra_people_perc
minimum_nights
maximum_nights
availability_30
availability_365
number_of_reviews
number_of_reviews_ltm
review_scores_rating
review_scores_accuracy
review_scores_cleanliness
review_scores_checkin
review_scores_communication
review_scores_location
review_scores_value
calculated_host_listings_count_private_rooms
reviews_per_month
response