THE UNIVERSITY OF SYDNEY

QBUS6810
Statistical Learning and Data Mining
Semester 1, 2022
Regression Project: Airbnb Pricing Analytics

Team members

Yuke Wang	**SID:510210139**
Leya Yang	**SID:510174886**
Qiao Zhu	**SID:470081428**
Yizhong Lim	**SID:510484268**
Jiaqi Zheng	**SID:500355097**

Table of Contents

1. Introduction

With the rapid development of society, Airbnb as a rental platform, has become a popular travelling-style stays for most of travellers. Since it could provide more comfortable and personalised experience than normal hotel of short-term living experience. Related to Airbnb, the report aims to perform a deep data analysis of Airbnb's price in Sydney and issues affected it by generating several statistical models to predict the prices. While the most accurate-predicted model will be selected which is stack model, and there are three important factors including 'accommodates', 'bedrooms', and the host locating area. Consequently, based on these three factors, suggestions about best investment ideas and beneficial ideas will be provided to the Airbnb's hosts and investors.

2. Overall problem and objectives

As a role of data analyst in the company targeting the Airbnb market, the project task aims to analyse the related data and finally come up with advising service to our hosts, property managers and real estate investors. While there are three factors which will influence the price most. And these will also be figured out after modelling.

The dataset contains all prices of accommodations and factors in Airbnb platform. To begin with, the exploratory data analysis is applied for giving a first glance of the whole dataset. By generating the correlation between variables and price as response variables, there are some variables which may not be important for analysis, have been delated as well during the feature engineering stage.

After finishing the data processing, exploratory data analysis and feature engineering, the better understanding of dataset could help to perform several predicted models. In this report, the models used include linear regression, ridge regression, decision tree, gradient boosting, XGBoost, and model stacking. The most ideal model will be decided based on the Kaggle competition scores, and it is model stack. In the end, three quantitative insights, which are essential to the price, are selected for giving our investors and hosts professional and accurate advice.

3. Data understanding and pre-processing

This project includes two datasets, training data and testing data. For training set, there are 12941 instances and 61 features. For testing set, there are 5547 instances and 60 features, which is not include response variable "price". As we aim to predict Airbnb price, we name feature "price" as response variable, merge two datasets as one for further analysis convenience, the dataset now is 18488 instances and 60 features.

3.1 Useless types

detect_types function from dabl package was used to detect useless type features. 98.76% of data in "neighbourhood_group_cleansed", "has_availability" and "calculated_host_listings_count_shared_rooms" are 0, so we remove these un-useful features before dealing with data forms.

3.2 feature forms processing

Firstly, we remove unit sign from variables, including the dollar sign for response variable "price" and percentage sign for "host_response_rate" and "host_acceptance_rate".

For string variables, “bathrooms_text” describe the bathroom types, if it is half-bath or private, we encode as 0.5. Since we want to keep the “description” variable, here we implement the sentiment analysis - AFINN method to measure the description score based on each word. AFINN is an English word those words scores range from minus five (negative) to plus five (positive) (Nielsen, 2018).

for categorical variables, since there are too many unique types in “property_type”, so we combine the “room_house_type” and “property_type” and summarize into 13 categories. For “host_location” variable. Because we aim to predict Airbnb in Sydney, we use key words “Sydney”, “AU”, “Australia” to exclude the host that outside Australia, filling outside location as 0. There are 1143 host which is not located in Australia, since the proportion is small, may not impact the prediction. And for Boolean variables, we convert to numeric as false to 0 and true to 1. Additionally, for numerical variable “host_since”, we convert it into days.

3.3 Dealing with missing values

There are 23 features that contains missing values, the table shows the missing ratios in percentage for each variable. We drop the column "license" which has the most missing values. For text type columns 'host_response_time', 'host_neighbourhood', 'neighbourhood', 'description_score', 'host_location', we fill the missing values with “none”.

For review related numeric columns, we fill them by their mean score. Among them, because the length of the review period may be one of the factors affecting the price, we calculate the difference between 'last_review' and 'first_review' to get a new column called 'review_gap' in days and replace its missing values with its mean score. So the 'last_review' and 'first_review' therefore can be deleted.

For some specific numeric columns, for example, missing in 'beds' may indicates the room number is 0. Therefore, for other numeric columns we fill them directly as 0.

	Missing Ratio (%)
license	0.616778
host_response_time	0.604662
host_response_rate	0.604662
host_acceptance_rate	0.563014
host_neighbourhood	0.422544
neighbourhood	0.377434
review_scores_value	0.297544
review_scores_checkin	0.297274
review_scores_location	0.297274
review_scores_accuracy	0.296949
review_scores_communication	0.296463
review_scores_cleanliness	0.296354
first_review	0.270392
last_review	0.270392
review_scores_rating	0.270392
reviews_per_month	0.270392
bedrooms	0.070803
beds	0.038998
minimum_nights_avg_ntm	0.000054
maximum_nights_avg_ntm	0.000054
minimum_minimum_nights	0.000054
maximum_minimum_nights	0.000054
minimum_maximum_nights	0.000054
maximum_maximum_nights	0.000054

Table 1 Missing ratios in each feature

4. Feature Engineering

Before exploratory data analysis, we firstly check the distribution of response variable “price”. Figure 1 shows an obviously right skewed, most data are in range 50 to 230 AUD and only small amount Airbnb has price over 1000 AUD. To mitigate data skew, we take the logarithm. The log density plot in Figure 2 shows data close to normal distribution.

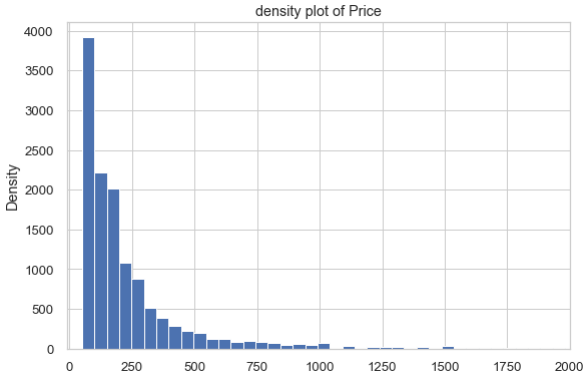

Figure 1 price distribution

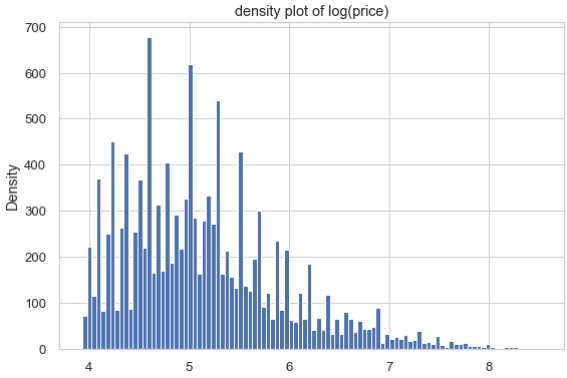

Figure 2 log(price) distribution

For predictors, we made regressions plots for all the features. We found that bedrooms, beds and accommodates are highly similar shown in Figure 3, which are right skewed. Additionally, these features and bathroom_text are showing an increasing linear trend, which indicates these features have linear regulation with price.

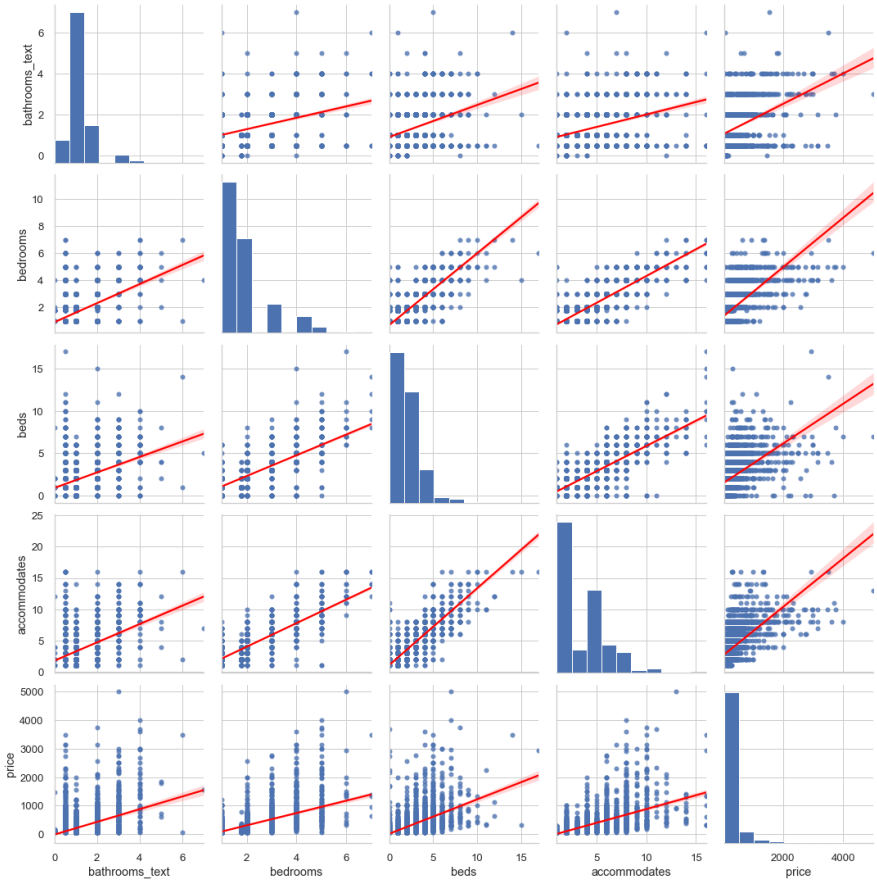

Figure 3 linear regression of bathroom_text, bedrooms, and beds

For “amenities” and “host_verifications” columns, the more items that host described, we think better the services they offered. We count the number of items with code to encoding these descriptive

words into numbers. Using the same idea, host_response_time column was encoding with a score ranged from 1 to 5, which higher the score, faster the host response to the customers inquiries.

Next, target encoding method was used for “room_house_type”, “neighbourhood” and “neighbourhood_cleansed” columns, which means the types of different house were converted into the average price that belongs to the same type. This way, we could better process data that has too many types.

5. Exploratory Data Analysis

By applying exploratory data analysis, the main purpose is to analyse data through visual technique. The following section will perform several graphs to evaluate the relationship between variables and the corresponding price as response variables.

Firstly, to get the quick random insight of data, it is important to know the types of room in the dataset. The bar chart below shows the amounts of house in different types. Most of the types for rent in the Airbnb is entire home or apartment. As Airbnb is as a platform mainly provided accommodation for people in travel. While the private room is listed in the top3 position as well.

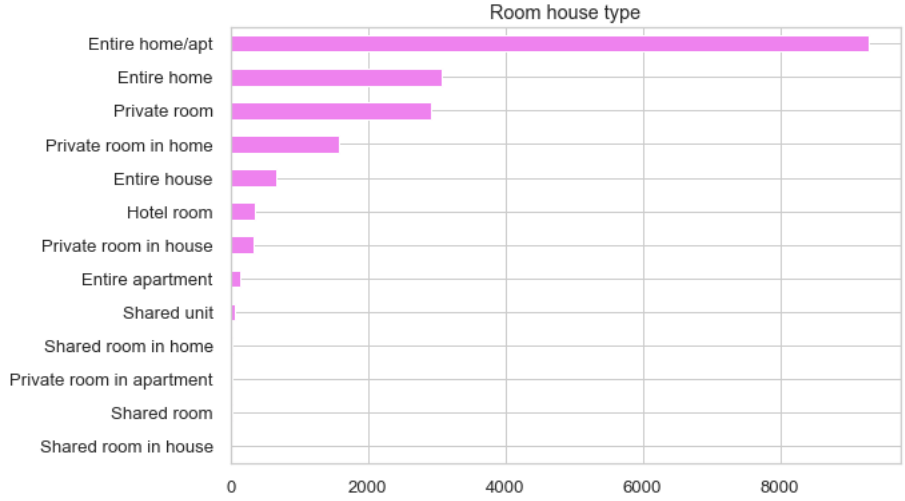

Figure 4 Room hose type counts

Also, it is considered that the review rating maybe influences the price level, higher rating maybe could reflect better services provided by the owners. It may have some correlation with price. The scatter plot below shows the relationships between the rating and price. It seems that most of the data in training set has high rating between score 4 to 5. There is also high price in low rating. It needs more information to investigate these two variables.

Figure 5 Review rating score relate to price

Besides, the number of amenities may be important for Airbnb bookers, as convenience could be less time consumption for travellers. As most of the accommodation's types are entire home/apt and private room, the amenities data are chosen from the two groups. And the two scatter plots below show that here is no clear trend or signal reflect that the price have any relationship between amounts of amenities. While there is various price level in different amounts of amenities. Hence, the factor may be little important to price.

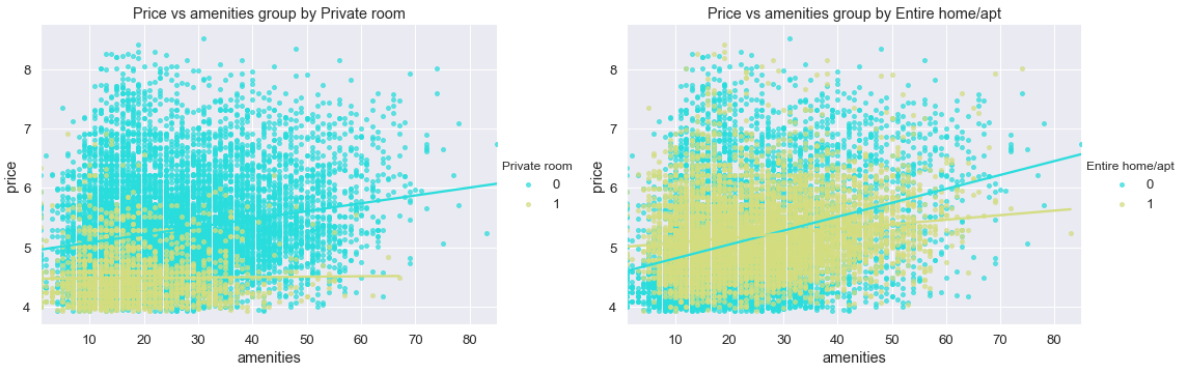

Figure 6 price with amenities (group by private & entire room)

Since some of these are features which do not have strong correlation will be eliminated after finishing the analysis. And it is necessary to generate the heatmap to enhance overall situation of correlation. Therefore, the heatmap of correlation between each variable has been shown below.

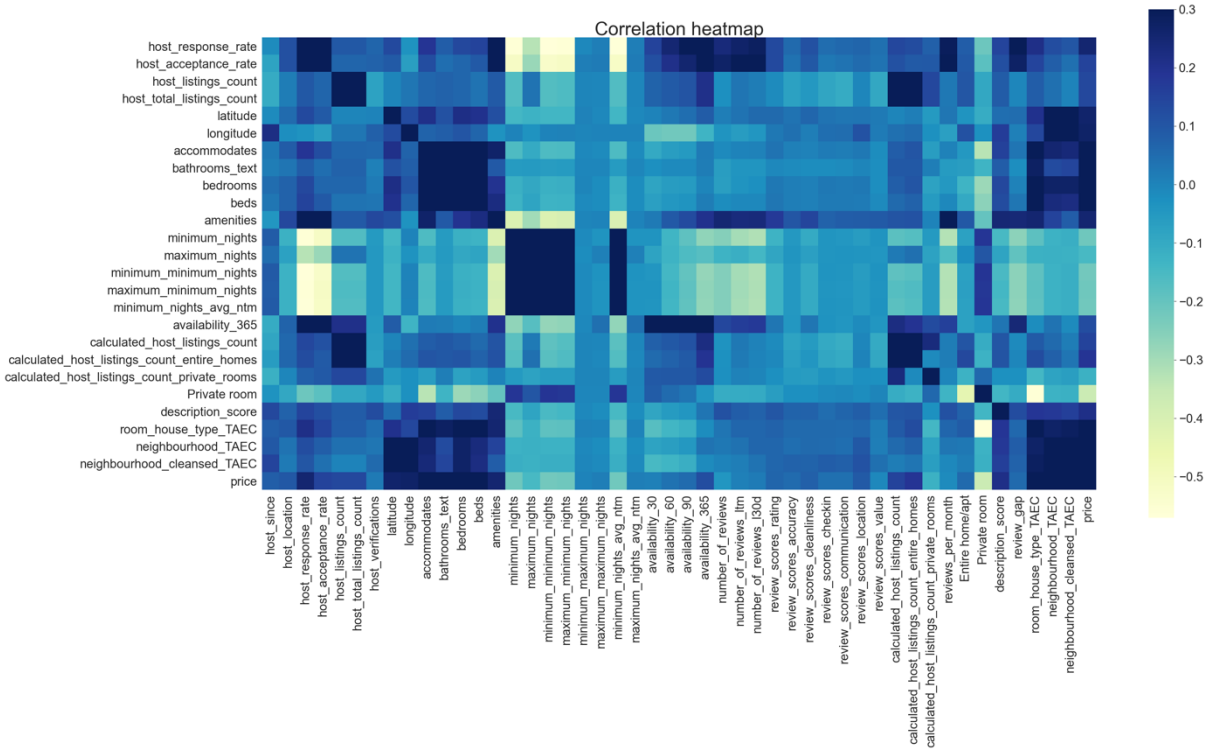

Figure 7 Correlation heatmap on each feature

From the heatmap, the variables including latitude, longitude, accommodations, bedrooms, room_house_type and neighbourhood have larger correlation, which are larger than 0.3, with price since their colour are darker. Consequently, the plot of large correlation variables will be discussed below.

Firstly, the boxplot is the relationship between number of accommodations and price. While the trend of price increases with larger amounts of accommodations. Once the amount is over 11, the price will not rise obviously.

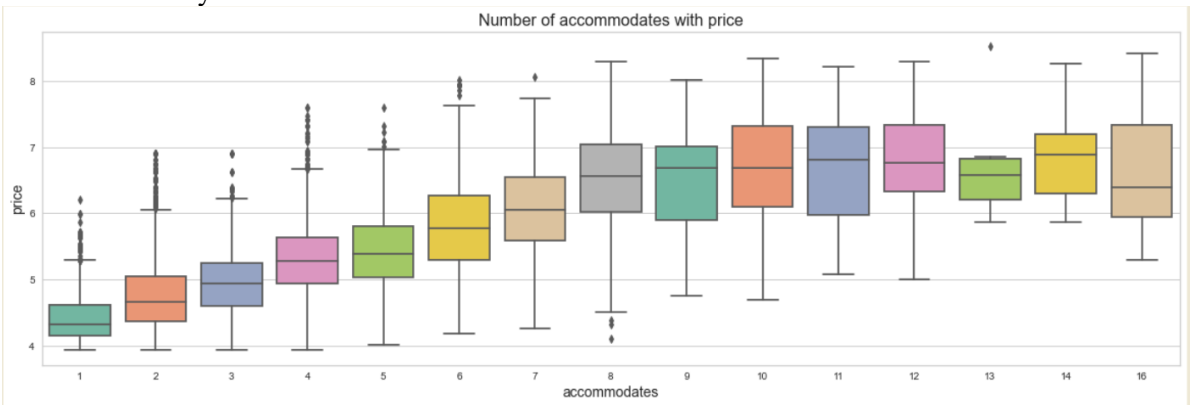

Figure 8 box plot of accommodates

Secondly, the graph below consists of latitude and longitude corresponding to the price. While there are some relationships between the price and the geographic location of the accommodations.

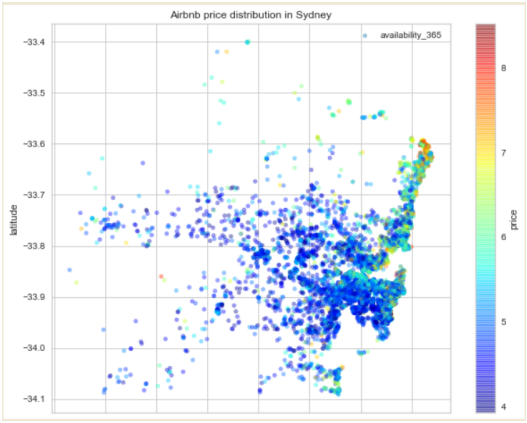

Figure 9 Sydney Airbnb locations

After enhancement of exploratory data analysis, the modelling for the data set will be conduct. Once finishing the modelling and choosing the best model, the relationship and important factors will be more convinced.

6. Methodology

We split the data in train.csv again into two datasets for training and testing and then use them to build and improve models for further testing. After that, we upload the results of these model runs and get the scores on Kaggle. Specifically, the models we uploaded are Ordinary Least Squares regression (OLS), Ridge, Decision tree, Gradient boosting, XGBoost and stack models. And since OLS, XGBoost and stack models won the highest Kaggle scores among linear models, non-linear models, we will mainly focus on discussing these three candidate models.

6.1 Linear Model- Ordinary Least Squares

As mentioned above, OLS gained a better score in Kaggle (0.47392) than the Ridge regression model (0.87286), so we will describe it in detail. OLS is "a common technique for estimating coefficients of linear regression equations which describe the relationship between one or more independent quantitative variables and a dependent variable (simple or multiple linear regression)" (Xlstat, n.d.). It is a simple and generalized model that dashes and can provide the basis for subsequent model optimization by different methods. However, this model also contains some limitations. For example, outliers can have a powerful impact (Sciencing, n.d.). Nevertheless, we still choose OLS as the benchmark model because of its attractive characteristics.

We import the LinearRegression library to build the model to introduce easy-to-use code using *sklearn.linear_model import LinearRegression.* Then, we establish the OLS model to fit the data set. In addition, we use the code *fit_intercept = True* to make sure that the intercept was taken into account.

After making the predictions, we visualise the feature's importance after making the predictions to demonstrate the model's results. For this purpose, we use the code *from tutorial9 import plot_feature_importance* to gain insight into our dataset's important feature column, and the results will be shown below.

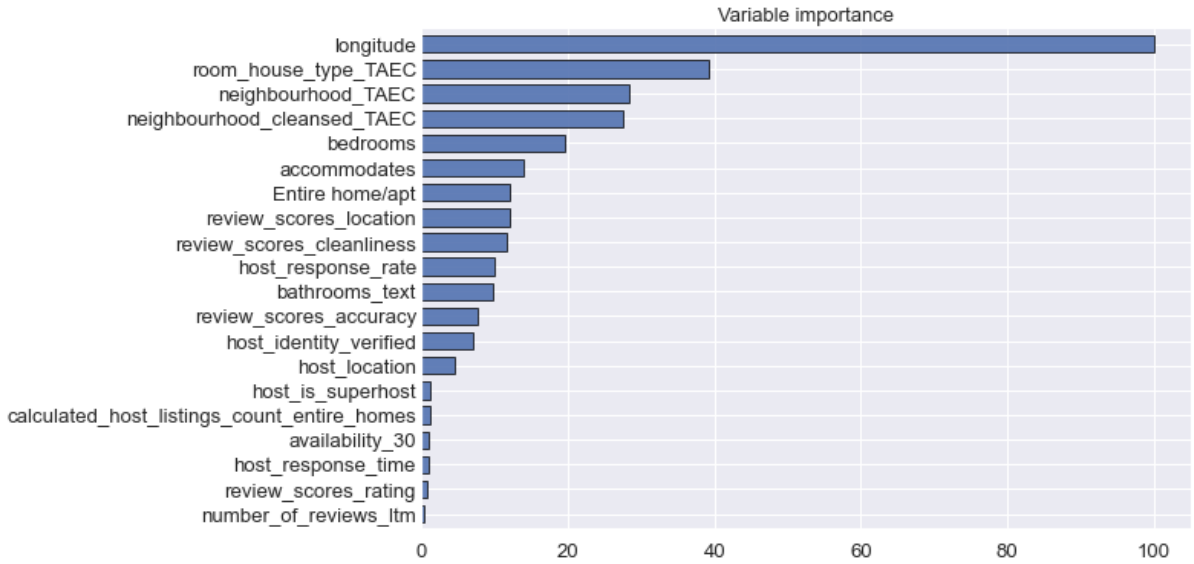

Figure 10 OLS's Feature important plot

As shown in the figure, according to the OLS model we built, longitude has the most significant influence among all the features.

6.2 Best Non-linear Model-XGBoost Model

6.2.1 Model fitting

Based on our private score for each model in Kaggle, we select XGBoost (0.40825) as our best non-linear model. 'Extreme Gradient Boosting (XGBoost)' is a well-known gradient boosted tree supervised algorithm to predict target variables accurately (AmazonSagemaker, 2022). This boosting technique follows the ensemble principle and gathers a set of weak models to improve the model prediction (Pathak, 2019). The motivation behind using this model is that it can handle structured datasets on either regression or classification problems which satisfies our Airbnb regression analytics project. Besides, the core algorithm of XGBoost is parallelisable, so it is feasible to learn on a large dataset as our dataset without sacrificing speed.

First, we start by importing the xgboost library by using *import xgboost as xgb* to develop our model. Next, since our project is a regression problem, we use *XGBRegressor( )* from the xgboost library and fit the regressor, *xgbst = xgbst.fit(X_train,y_train)* to 80% of our training set and make a prediction on the remaining 20% of the test set by using y_predict= xgbst.predict(X_test).

We will also use the same code mentioned in the OLS part to visualise the feature's importance after making the predictions to show the results.

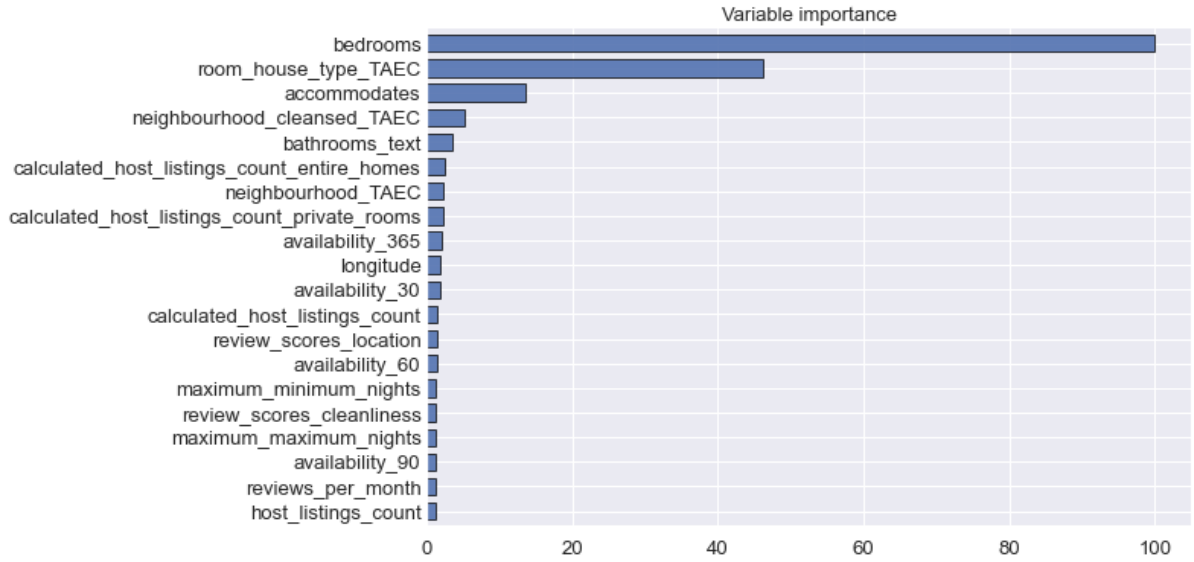

Figure 11 XGBoost's Feature important plot

Based on the results shown in Figure 2, bedrooms have the highest feature important score among other features. On the other hand, longitude, considered the most influential in the OLS model, is considered much less critical, while room_house_type_TAEC has the second most decisive influence in both models.

6.2.2 Hyper-parameter tuning

For model refinement, we implement grid search for hyper parameters tuning. That is, for each iteration, test all the possible combinations of hyperparameters, by fitting and scoring each combination separately. The tuned hyper parameters are included:

- Max_depth: the maximum depth of each tree, often values are between 1 and 10. Since deeper depth result more complex model and easier to overfit, here we set the max depth as 10 to looking for optimal one.
- n_estimators: the number of trees/estimators are built within the ensemble model. Although the larger number of trees will have more accurate prediction because of the nature of the Gradient Boosting algorithm, the larger n is require longer running time and may lead to overfit. Therefore, 500 estimators are used.
- learning_rate: the step size that use to prevent overfitting, here we set as 0.1.

The best parameters output is {'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 500}. Compare with model before adjustment, the RMSE decreases 0.0061 and the R^2 becomes 0.7324, an increase of about 0.0080. The Kaggle score is improved 0.5%.

6.3 Stack Model

The stack model gained the highest score (0.40339) in the Kaggle competition among all the models we uploaded, so it is considered the best one. Brownlee (2020) explains that model staking is a kind of ensemble machine learning method that specifies a meta-learning algorithm to learn how to best gather the predictions from different models. The motivation for developing this model is that it does enhance the overall accuracy of the models we use.

The core structure of the stacking model involves two levels of models, the level-0 model, and the level-1 model. The level-0 model can apply more than two base models to fit training data and compile the prediction. We integrate models, including of OLS model, ridge model, decision tree

model, gradient boosting and XGBoost model, into the level-0 model by using *StackingCVRegressor* from *mlextend.* We then apply *stack.fit( )* to fit the training data. The level-1 model, also known as the meta-model, aims to produce the final model stack prediction. It uses the prediction made by the base model as its training input and learns how to combine it perfectly. Since there are no restrictions on selecting the algorithms for the meta-model, we prefer the linear regression method as it is more appropriate for predicting numerical data. However, the stacking approach does come with the potential issue of data leakage may trigger overfit risk as we use the full validation set to fit this model.

7. Results

To examine the quality of the model, we will introduce four metrics: root mean squared error (RMSE), mean absolute error (MAE), represent squared error (MSE), and R^2.
Each metric stands for a different meaning. Root mean squared error is "the square root of the mean of the square of all of the error" (Science Direct, n.d.). And Absolute Error refers to the difference between the measured and actual values, while MAE is the average value of absolute error (Statistics How To, n.d.). Then we use the MSE, which is the average amount of the square of the difference between actual and the measured values (Deval, 2020). And R^2 "represents the coefficient of how well the values fit compared to the original values." (DataTechNotes, n.d.).

Although their meanings may vary widely, they can all be used to evaluate the difference between the results predicted by a model and the data set. The values of these four metrics for the six mods we built are shown below.

	RMSE	R^2	MAE	MSE
Linear regression	0.4507	0.6723	0.3431	0.2031
Ridge Regression	0.7873	-0.0002	0.6244	0.6198
Decision Tree	0.5858	0.4462	0.4330	0.3432
Gradient Boosting	0.4140	0.7234	0.3111	0.1714
XGBoost	0.4072	0.7324	0.3033	0.1658
Model Stack	0.4010	0.7405	0.2986	0.1608

Table 2 Metrics for the models

Based on the result in Figure 3, we can rank each model by evaluating their RMSE, which is the primary metric used for this project. The top two XGboost and Gradient Boosting models have similar scores of 0.4072 and 0.4140. It is worth mentioning that both models are followed the concept of boosting algorithm. After, the ranking is followed by OLS, decision tree, ridge regression and model stack. However, it is always crucial to compare the results with the data that are not from the training set as we expect our model can be generalized to unfamiliar data. From the Kaggle results, the stack model has an outstanding performance of 0.40339, which makes it stand out from other models. Hence, we are confident in selecting it as our final model. Besides, ridge regression (0.7873) and decision tree (0.5858) are underperformance as their score are not good as our OLS (0.4507) benchmark model. In a nutshell, two models out of six fail to satisfy our requirements, and further study is recommended to improve the results.

8. Data Mining

To decide what kind of hosts could be regarded as the best host, the method is to discuss three quantitative main insights from the models above. Since the variables all relate to the price of Airbnb. It is considered that the 'best host' discussed in the report is the host who generate the high revenue and price of the accommodations. Base on three factors affecting Airbnb prices, further suggestions will be illustrated to related host and real estate investors in the end of the report.

Based on the feature importance of the well-performed models, there are three factors including 'bedrooms', 'accommodates' and 'longitudes' which are top significant to the Airbnb price. Hence, the following paragraph will analyse these three factors.

Firstly, for bedrooms, the boxplot demonstrates the relationship between the numbers of bedrooms and its related price. Most of numbers of bedrooms are below seven, little number of bedrooms which is ten with low prices. As number of bedrooms increase, the prices level grows respectively. It shows that the price with highest mean mainly distributes in 4 to 6 bedrooms. Whereas the box of 1.78 bedrooms is the mean of bedrooms dealing with missing data. Hence, it is recommended that number of 4 to 6 bedrooms will result in relatively higher Airbnb price.

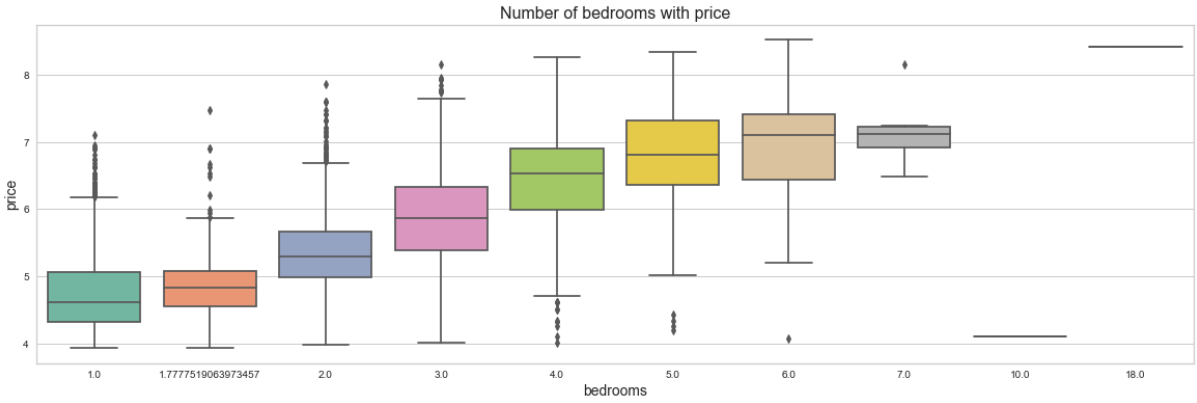

Figure 12 boxplot of number of bedrooms with its price

Secondly, accommodate is also an important factor which alters the price. The accommodate represents the maximum capacity, could be essential for the Airbnb guests.

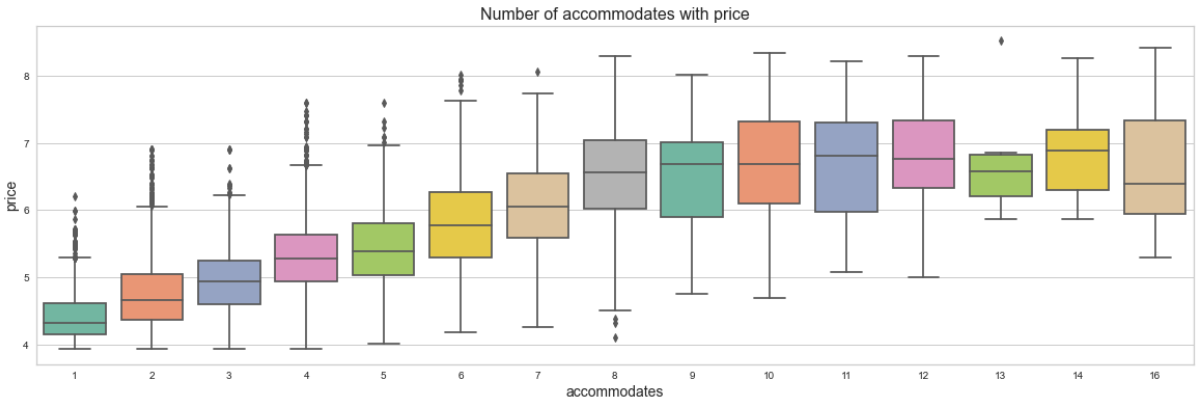

Figure 13 boxplot of accommodates with price

The boxplot shows that the price level increases with growing accommodates as well. While the price reaches maximum level with range of accommodates in 8 to 12. The accommodates number in 16 also corresponds to large price but its price means is a little lower. It might be the reason that there are some accommodations which are full of too many people in lower price per person. Hence, by

considering this, the number of accommodates recommended is 9 to 12. Since the range of price fluctuations in number of 8 is a little bit large compared to others.

Thirdly, the longitude, as a factor about geographical location, is seen as another significant issue for investors and owners of Airbnb. To discuss this variable, longitude, latitude, and neighbourhood regions are also related. Hence the analysis below will combine these factors.

Figure 14 Airbnb in Sydney's location and price

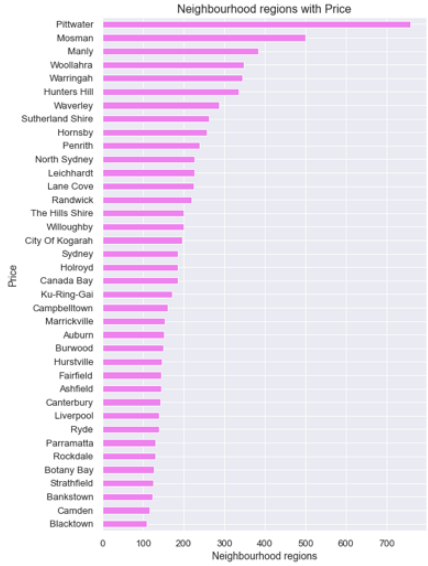

Figure 15 boxplot of neighbourhood regions with price

The graph shows that most of the accommodations in lower price are in the southeast region of Sydney, whereas the pricing increase when the region moves to northeast region. From the left graph, the top-price region is Pittwater and Mosman, it is reasonable since the Mosman is a wealthy port suburb which wealthy locals frequently visit with wonderful view. Hence, it is recommended that the north-east and up-north region is in high-value of investment.

In conclusion, based on the three main factors 'accommodates', 'bedrooms', and the host locating area. The accommodations with 4 to 6 bedrooms in around 10 accommodate in the top-north or northeast region in Sydney could be the most beneficial for real estate investors and the Airbnb owners.

9. References

Amazon Sagemaker. (2022). *XGBoost Algorithm*. Amazon Web Services. https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html.

Brownlee, J. (2020, April 27). *Staking Ensemble Machine Learning with Python*. Machine Learning Mastery. https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/.

DataTechNotes. (n.d.). *Regression Model Accuracy (MAE, MSE, RMSE, R-squared) Check in R*. https://www.datatechnotes.com/2019/02/regression-model-accuracy-mae-mse-rmse.html

Deval, S. (2020, August 8). *Mean Squared Error – Explained | What is Mean Square Error?* . Great learning https://www.mygreatlearning.com/blog/mean-square-error-explained/

Nielsen, F. Å. (2018, 2 19). Finn Årup Nielsen's blog. Retrieved from afinn Addressing “addressing age-related bias in sentiment analysis”: https://finnaarupnielsen.wordpress.com/tag/afinn/

Pathak, M. (2019, November 8). *Using XGBoost in Python Tutorial*. DataCamp. https://www.datacamp.com/tutorial/xgboost-in-python#what.

ScienceDirect. (n.d.). *Root-Mean-Squared Error*. https://www.sciencedirect.com/topics/engineering/root-mean-squared-error

Sciencing. (n.d.). *The Disadvantages of Linear Regression*. https://sciencing.com/disadvantages-linear-regression-8562780.html

Statistics How To. (n.d.). *Absolute Error & Mean Absolute Error (MAE)*. https://www.statisticshowto.com/absolute-error/

Xlstat. (n.d.). *Ordinary Least Squares Regression (OLS)*. https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols