Qbus6810 GroupAssignemnt
Group 57

1. Business Understanding

As the largest networked hospitality service in the world, Airbnb has been offering the complete independence to its hosts when it comes to pricing their properties on its online platform for short-leasing. There is a wide range of information displayed on the listing that makes a property attractive to the guests. With the increasing number of hosts using Airbnb, it is essential for the hosts to find an optimal price for their properties to remain competitive.

It is intuitive to believe that factors such as the location, interior, number of bedrooms and review of property might have significant impacts on its price. However, exactly how do factors impact the pricing is unknown to most of the hosts and therefore, getting insights into this mechanism is imperative for them.

This report has reached beyond that and aims to develop a price recommendation engine for Airbnb hosts to predict nightly prices of their listings based on advanced machine learning techniques.

The problem formulation consists of three steps. First, collect and process the data of Airbnb listings including various features in a specific region, in this case, it is the Northern beaches council area of Sydney. Second, build and train models using machine learning methods based on feature engineering and selections. Moreover, at last, we will recommend a pricing model for the client supported by analysis and evaluation rather than only choosing the best performing model.

2. Data Processing

In order to clean and prepare the data sets for further analysis, we started by checking them for missing values. The number of missing data in both training and test sets is shown in table 1.

	Train	Test
Security deposit	160	256
Cleaning fee	119	194
Review score rating	128	173
Review score accuracy	129	175
Review score cleanliness	128	175
Review score check-in	129	176
Review score communication	128	176
Review score location	129	176

Review score value	129	176
Reviews per month	120	163
bathrooms	0	2

Table 1. Number of Missing Values in Data Sets

Then we fill in the missing data with different methods depending on the scenarios. For security deposit, the mean is put in for listings that are either apartment or house while zero is put in for the rest of the listings. For cleaning fee, we calculate the mean of cleaning fee for private room and entire home/apt and fill them to the missing data accordingly. Regarding all the review score related missing values, the mean of each feature is filled in, and the missing value in review per month is put in with zero. Finally the bathroom missing value in test set is filled in with the mean number of bathroom.

In addition, when a listing's number of beds is zero, we replaced it with its bedroom number, and when the bedroom number is zero, we replaced it with its number of Beds.

3. Exploratory Data Analysis

After collecting and cleaning all the data, the Exploratory Data Analysis (EDA) is undertaken as the first step, which is a significant process where we can gain deep understanding and insights towards the target and features. The data are presented with graphics and charts to make the distributions of each variable clear and intuitive. In this case, some critical information, potential issues and interesting patterns about data are discovered.

Response

Firstly, the target feature Price is extracted from the dataset and analyzed by plotting the histogram, as shown at the left panel in figure 1. It can be easily found that the shape of the price distribution is right-skewed, with the central location just above 100.

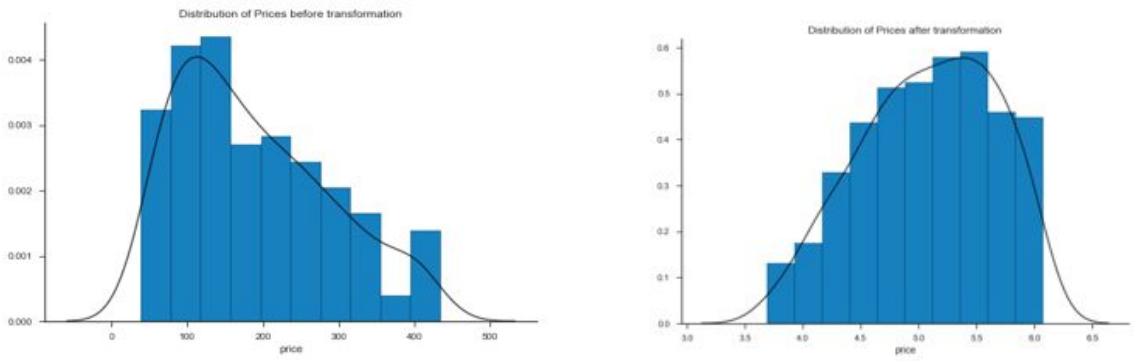

Figure 1. Comparison of property prices before and after log transformation

To make the data a better approximation of the normal distribution and therefore improving the validity of the associated statistical analysis, the log transformation is applied to the prices. The transformed distribution is illustrated at right panel in figure 1, with the skewness of prices changed from 0.604 to

-0.289, which is closer to a normal distribution. To take account of the issue of invalid results of log zero, we manipulate the data of prices plus 1.

Features

As for the features, we firstly classify variables into different types of categorical and dummy. In this case, there is a majority of categorical variables, and considering the complex pricing for a property (i.e., 5-bedroom house and 2-bedroom house can have the same level of price due to other factors), the box plotting can be the optimal graphic tool to observe and describe the macro tendency that a particular feature influences the target variable. The relationships between prices and dummy variable cancellation_policy and categorical variable accommodates are selected as examples and illustrated in figure 2.

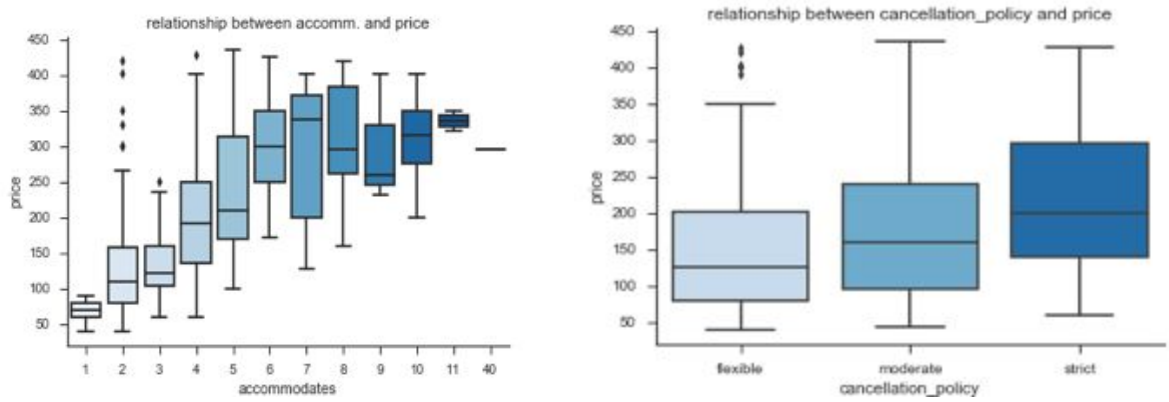

Figure 2. Box plots of accommodates and cancellation_policy against prices

In the left panel of the figure 2 above, it can be observed that there is a generally upward relationship with the number of accommodates increasing, while the same direction appears in the right panel demonstrating the stricter the property's cancellation policy, the higher the typical prices. However, we can observe that there are some outliers appearing at the boxes of '2 accommodates' in left panel and 'flexible' in right panel, which will be discussed in the feature engineering part later. There are also features of reviews ratings that are based on tenants' subjective comments to the property. We need to pay special attention to those review, and the review_scores_rating is selected as the example to be illustrated:

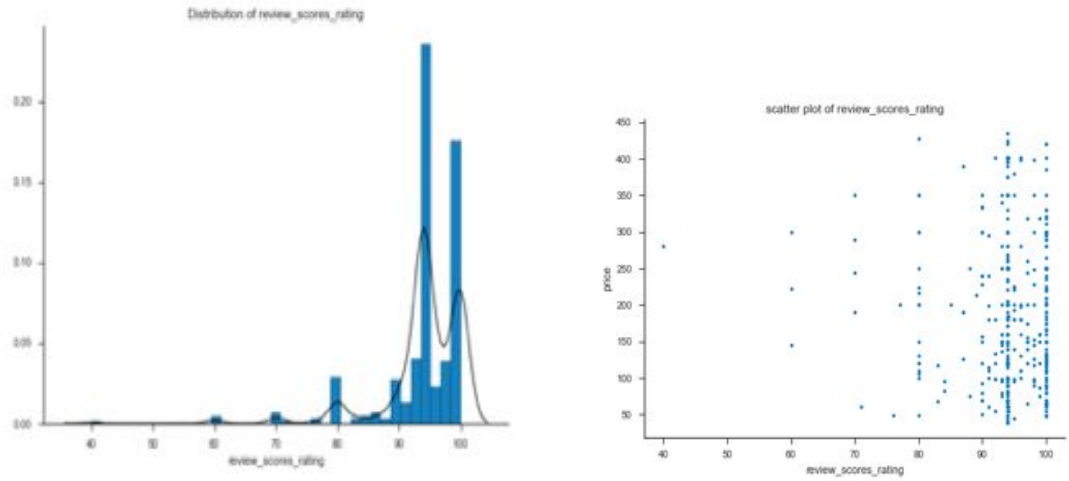

Figure 3. Distribution and scatter plot of the review_scores_rating

The left panel of figure 3 indicates the distribution of the review_scores_rating is significantly left-skewed, which proves the truth that tenants tend to comment under a decent stay experience. Furthermore, it can be observed that in the right panel above, the dots at right corner are much denser than the dots at right top, which reveals that at same rating level properties with lower price tend to obtain more reviews, thereby reflecting the change of the price levels.

Feature engineering

Feature engineering is one of the most effective means to improve the performance of the model, in which some of the features can be transformed and utilized more efficiently. In this case, we create two new features, measure outliers, and inspect the effect of transformation to features.

For the better understanding of the connection between pricing pattern and property characteristics, we create a new variable bedroom per bathroom by doing dividing, as we believe if each bedroom is equipped with their own bathroom, the property can have a price premium. Figure 4 illustrates the distribution of the created feature bedrooms per bathroom.

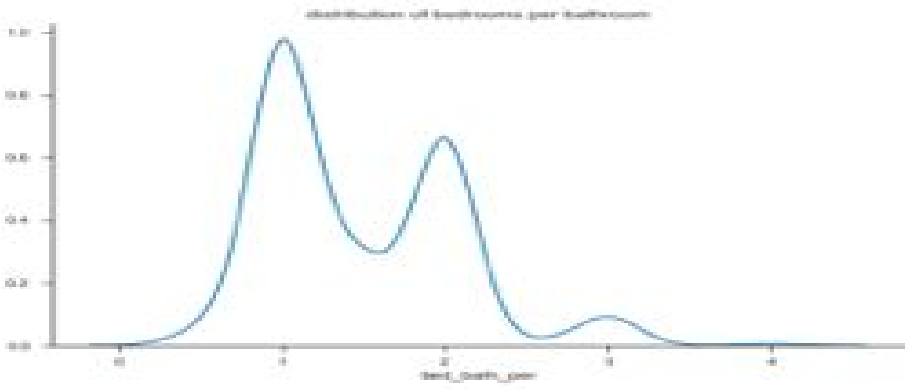

Figure 4. Distribution of the created feature bedrooms per bathroom

In the figure 4, it is noticeable to observe that less popular the property is when a bathroom serves more residents (i.e., more bedrooms). Therefore, we believe this feature makes sense and add this new feature into the model evaluation.

The other feature we have transformed is the location because the price of a property tends to be higher if it nears to shopping malls, tourist attractions, or transport junctions. In this way, we utilize the original features Longitude and Latitude and import the module of geocoding to convert them into a new feature Location. In addition, we use the KNN to find neighbor centre and zone each input into different neighbors like Manly, Palm Beach, and Narraweena by computing the distance of property towards a particularly central location.

As for the outliers in some features like 40 accommodates for a property with two bedrooms and one bathroom, we attempted to delete it as we believe it was misinputted, but the truth is the result with those outliers remaining is better.

In addition, there are features like host_is_superhost, property_type, and cancellation_policy whose results are categorical. To assist manipulating and fitting the numerical data in the regression model, we transform all of them into the dummy variables with the value of 0, or otherwise 1.

Lastly, we investigate the skewness for each feature and is trying to transform positively and negatively skewed data to approach an approximately normal distribution by applying log and square transformations respectively. However, the results are not that dissatisfactory due to the volumes and special attributes of some features. In the end, we decide to remain features intact, and the skewness of some features are shown in Table 2.The skewness of some features .

Variable	Skewness
Host_is_superhost	-2.613
Host_ total_listing	8.095
Host_identity	-0.148
Latitude	1.179
Longitude	-0.769
property_type	-0.373
room_type	-1.206
Accommodation	6.835
Bathrooms	1.756
Bedrooms	0.937
Beds	1.566

Security_deposites	4.878
Cleaning_fee	1.487

Table 2.The skewness of some features

Overall, in the feature engineering, two new features bedrooms per bathroom and location have been created as they are regarded as contributory to the success of the model performance later, but the effectiveness of those features needs further research.

4. Methodology

4.1 Assumptions

4.1.1 Correlations

We use the following correlation heatmap to check degrees of inter-associations, and the figure 6 shows the correlation between all the variables each other from strong to weak inner relations with the color from light to dark. According to the figure, features from the room_type to cleaning_fee seem highly correlated with the change of prices. Therefore, those features should be the optimal features set. However, due to less than 30 features provided, and the truth which has been proved by several attempts indicates the more features our models covered, the more accurate our prediction can be. In this way, all the variables and newly-added featuk mres are included in our variable selection to be considered into the model evaluation. Thus, we still incorporate all the features to avoid missing information and we have checked that the collinearity problem is not nominated in our regression model.

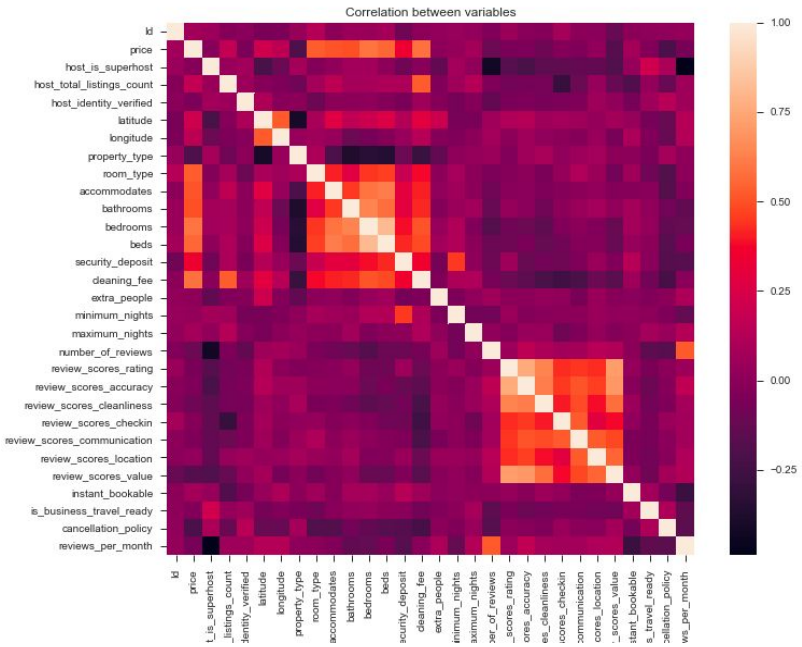

Figure 6. Heatmap

4.1.2 Non-linearity

Considering the importance of all variables, those relationships are presumed to be captured using complex models and simple linear regression model may not perform well due to its inability to capture non-linearity. This diagram above also affect our model selection decision in the following part.

4.1.3 Homoskedasticity

Additionally, we consider the potential time effects which will violate the constant variance assumption of the error distribution. Thus, the logarithm of prices has been applied to eliminate the heteroskedasticity of time-series data.

4.2 Train model and variable selection

After cleaning up the data and completing the functional engineering, we then select the appropriate models to predict the price of clients' properties listed on Airbnb. We start our variable selection using principal component analysis and find the subset of significant variables cannot explain and predict the price accurately. Hence, the decomposition of host features will lead to models inability to capture complex relationships. The deterministic factors in our model are then based on all the characteristics and the additional features we create.

In this report, we consider several models including three ensemble models which are random forests, Xgboost, and LightGBM, and two interpretable models which are RidgeCV and Elastic net. The best performing models from the two categories will be explained in detail.

Further, the training data is randomly partitioned into a training and validation set with a proportion of 85% and 15% to assess the learning performance of the algorithms. The validation set is not used in the training or test phase, which allows us to validate the predictive accuracy of each model and prevent overfitting on the training data. The training set is then trained to build our model, and the generalizability of the mapping function will be tested using the allocated test data.

We initialize building our learning function using ordinary least squares and find the linear relationship cannot perform well regarding to capture complexity. Then we try other models and select the best performed and most interpretable function.

4.3 Prediction accuracy

To identify the well-performed model, we use Root Mean Square Error (RMSE) to evaluate model performance. When processing and presenting the RMSE, we assume that errors are unbiased and also randomly distributed. RMSE is the square root of the average of the sum of squares of the difference between our prediction and the true value, which shows how the prediction residuals are spread out. Comparing the forecast with the known value from the test set, we list the below table of validation scores from the Kaggle submission for each model in an ascending order and we round the results to 2 decimals.

$\widehat{y_i}$ = *forecast value*
y_i = *observation* (*known value*)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Model	RMSE
Xgboost	59.38
Random Forest	64.38
RidgeCV	69.32
Elastic Net	85.92
LightGBM	194.44

Table 3. RMSE

By observing the RMSE above, we identify a simple model Elastic net and one advanced model Xgboost to do further analysis.

4.4.1 Elastic Net Model

As a simple model, elastic net has great interoperability. When multiple features correlated with each other, elastic net is useful to solve the limitation of LASSO and Ridge method (Muniain & Ziel, 2018). Our case has approximately 30 variables and the heatmap shows existence of high correlations between several variables. If we directly apply LASSO, only one variable will be chosen as the main predictor. Therefore, other information contained by remaining variables will be ignored resulting in low prediction accuracy. This method is very similar to OLS but with quadratic and linear penalties.

Ridge uses L2 as the penalty term which will limit the size of the coefficient vector. Its penalty added equivalent to the square of the magnitude of coefficients, represented as:

$$\sum_{j=1}^{p} \left|\beta_J\right|^2$$

L1 is used as the penalty by LASSO whose penalty term imposes sparsity on coefficient. It will add penalty equivalent to the absolute value of magnitude of the coefficients, represented as :

$$||\beta||_1 = \sum_{j=1}^{p} \left|\beta_j\right| .$$

Elastic net adds a quadratic part to penalty, represented as follow:

$$\hat{\beta} = arg_{\beta} min(||y - X\beta||^2 + \lambda_2||\beta||^2 + \lambda_1||\beta||_1) .$$

Elastic net then performs variable selection and shrinkage regularization, where coefficients of irrelevant variables will be shrinked to zero.

4.4.2 Use Elastic Net to predict Airbnb price is unreliable

According to the result of RMSE 85.92, the elastic net is poorly performed and the standard deviation of the error is widely spread out, which potentially indicates the model is unable to capture most of the

relations between the variables, while lower value of RMSE represents better fitness of the data. We believe this poor performance is contributed by the existence of nonlinear relation between variables and this kind of association is out of the coverage that can be fitted by the Elastic net (Zou & Hastie, 2005). Moreover, elastic net uses cross-validation to select λ1 and λ2 in sequential order instead of simultaneously identifying λ1 and λ2, the cross-validation selection may cause "double shrinkage problem". Therefore it still leaves out some relations (Li & Lin, 2010). Therefore this model is inappropriate for this price predicting case.

4.5.1 Build Xgboost

Due to the dissatisfactory result from Elastic Net, we start to move from linear relation to more complex relation. Xgboost is called eXtreme Gradient Boosting which implements Gradient boosting framework. It formalizes using a more formal model to control overfitting for better performance (Yang & Bath, 2018). Boosting can help to convert the weak regressor into strong regressor by adding up models. Boosting makes the new model adapt to the predicted residuals obtained from the previous model and then minimizes the loss when adding the latest prediction. Since Xgboost can handle almost all linear and nonlinear instances, we can't visually see the relationships in the data after processing the data. With the parallelized tree boosting algorithms, Xgboost can solve the problems efficiently and flexibly. In this case, we chose to use tree boosting because in most of the situations tree boosting can contribute high performance for prediction mining in both classification and regression (De'Ath, 2007).

The following functions show the algorithms of adding weak learners adaptively and iteratively to the previous one where the objective is to minimize the residuals of the previous function until reaching the strong learner (Möller etc., 2016). Moreover, the loss function accounts for the inaccuracy of previous predictions and a regularization term to punish the complexity of the model.
We need to decide new function to add in tth iteration:

$$\begin{gathered}\widehat{y}_i^{(0)} = 0 \\ \widehat{y}_i^{(1)} = f_1(x_i) = \widehat{y}_i^{(0)} + f_1(x_i) \\ \widehat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \widehat{y}_i^{(1)} + f_2(x_i) \\ \cdots \\ \widehat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \widehat{y}_i^{(t-1)} + f_t(x_i)\end{gathered}$$

Following is the square loss and the residual from previous iteration:

$$\begin{aligned}Obj^{(t)} &= \sum_{i=1}^{n} (y_i - (\widehat{y}_i^{(t-1)} + f_t(x_i)))^2 + \Omega(f_t) + const \\ &= \sum_{i=1}^{n} (2(\widehat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2 + \Omega(f_t) + const\end{aligned}$$

4.5.2 Use Xgboost to predict Airbnb price

The intercorrelations between variables can be resolved by Xgboost automatically, which inherits favorable robustness from decision trees (Friedman, 2001). Moreover, the inherent boosting model deals

with imperfectness of data concerning monotone transformations, outliers, missing data and irrelevance of input variables. Hence, this mapping function estimates the host's price most effectively.

In this case, there is a relatively small group of hyperparameters in our learning algorithms. Therefore GridSearch Cross Validation is used to tune the parameter. The automated search uses a grid of possible values to assess on the cross-validated sample and provide the score for the regressor, which is the log loss function of Xgboost. We test max_depth of 5 in steps of 2 and n_samples_split from 250 to 1500 in steps of 250, where the process is iterated until the maximum number of trees has been reached. Further, learning rate, subsample, and number of estimators are optimized.

4.5.3 Feature importance

Then based on the process that we have done, we can export a feature importance diagram. The figure 7 demonstrates the important aspects of the functional relationship, and the variables explain most of the variations are latitude, cleaning_fee, longitude, reviews_per_month and accommodates in descending importance order. These factors can be concluded as the most influential drivers of predicting the Airbnb listing price in our algorithms.

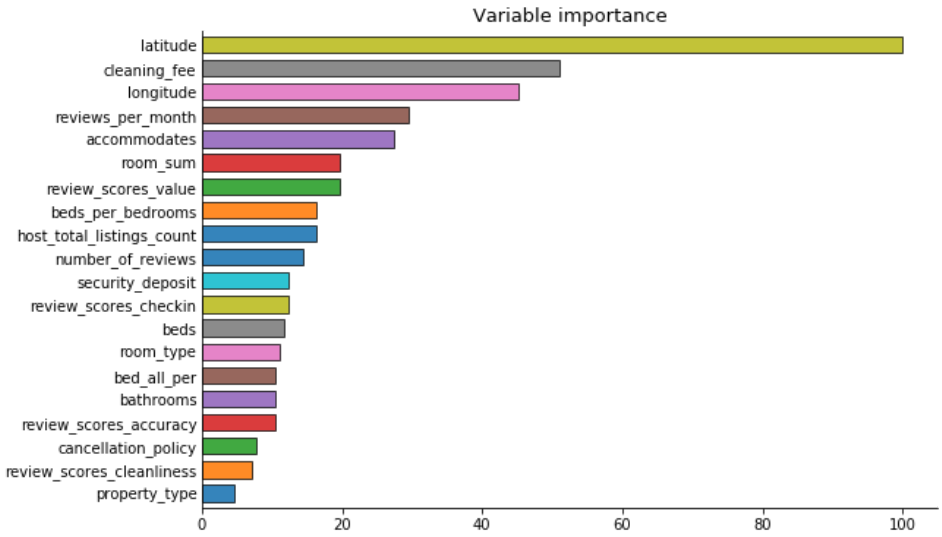

Figure 7. Variable importance

4.5.3 Interpretations of Xgboost

We want to interpret and understand how the input variables will lead to different hosts price levels and which specific input variables or a set of essential variables have most influences in variations of our predictions as well as the nature of the dependence of property price on those inputs (Friedman, 2001). However, the interpretability of a small decision tree approximation is diminished in the context of boosting algorithms, and we need additional tools such as partial dependence plot to support an obvious interpretation of our “black box” prediction method due to the high dimensionality arguments. In a nutshell, Xgboost predict Airbnb listing prices most effectively with a validation score of 59.38, which is the lowest RMSE compared to the other models.

4.6 Analysis of limitations

The first limitation is that we need more detailed dataset and features provided in order to improve the model performance. For example, during the feature engineering phase, we have recognized that occupancy rates will have a significant impact on booking status, also, the seasonal effects and the built tiem of a property are determinant to its pricing. However, there are no clues about these factor in the database. The tourist season usually causes house prices to fluctuate (Choi,Jung,RyuDo Kim & Yoon, 2015), and the occupancy rate will give us a clearer understanding of which houses are popular choices. Generally, popular choice owners will pay more attention to the price setting to attract or protect their own economic interests (Ikkala & Lampinen, 2014).

Meanwhile, even with a very high flexibility, Xgboost may have some problems concerning interpretability. Like all black box models, the processes and calculations in the model are difficult to analyze, making it difficult to explain the resulting price model to the customer (Bort Escabias, 2017). However we chosen tree to improve the interoperability a little.

We also notice that the hosting price is usually related to the target consumer group. In this case, we are unilateral from the perspective of the host. However, in actual situations, the needs of consumers are also essential for the formulation of prices. For example, some consumers only need one room, while others have specific requirements for the living experience. Their sensitivity to the same variables is likely to be different, and the factors they care about will also affect the price they can accept.

5. Evaluation

The criteria used to evaluate model performance and accuracy is the RMSE score. The RMSE table in the previous section shows the RMSE scores for the five models we have chosen on the validation set of the test data. Elastic net (85.92) and LightGBM (194.44) have relatively poor performance compared to the other three models. Random forest (64.38) and Ridge CV (69.32) perform considerably average in between, while Xgboost (59.38) has the lowest RMSE score and therefore is the best performance model.

With the assumption we made in methodology, the poor performance of elastic net can be explained by the inability to capture non-linearity. The result of LightGBM is unsatisfying and way below our expectation. We suspect it is because LightGBM is more sensitive to hyperparameters and requires a more sophisticated process of tuning parameters. LightGBM uses loss guided tree that will promote the overfitting situation. During our process we may lack control of overfitting and the tuning parameter. Therefore in this case our Light GBM model can not provide a good result.

For Random forest, we believe it is because the effort from feature engineering has been obscured by the process of bagging and building trees. Probably because that usually random forest doing good will small number of informative variables, and in this case variables group is a bit large and has inner correlation between them.

And in terms of Ridge CV, its moderate performance may be caused by the fact that there is a specific group relationship in the variables, and the interrelated variables have a significant impact on the price forecast.

Overall we choose Xgboost as the optimal model for price prediction. Though the scope of this report could be stretched out by adding text sentiment analysis on reviews and it will enrich the insights of customer segmentation and make up for the limitations we find in the methodology.

6. Conclusion

Based on the business context, we have built an optimal pricing model using processed data and feature engineering. With the validation score and analysis we are able to conclude that the best performing model for Airbnb price prediction is Xgboost with all the features we proposed in the report. In the meantime, model of linear relationships is not recommended for pricing in this case. Despite our best efforts, there are still limitations of this pricing model due to uncertainty of data processing and lack of information. However, with a deeper data mining and appropriate adjustments, this model will have improved performance and therefore is the recommendation we make for our client.

References

Bort Escabias, C. (2017). *Tree Boosting Data Competitions with XGBoost* (Master's thesis, Universitat Politècnica de Catalunya).

Choi, K. H., Jung, J. H., Ryu, S. Y., Do Kim, S., & Yoon, S. M. (2015). The relationship between Airbnb and the hotel revenue: in the case of Korea. *Indian Journal of Science and Technology*, *8*(26).

De'Ath, G. (2007). Boosted trees for ecological modeling and prediction. *Ecology*, *88*(1), 243-251.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Ikkala, T., & Lampinen, A. (2014, February). Defining the price of hospitality: networked hospitality exchange via Airbnb. In *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing* (pp. 173-176). ACM.

Li, Q., & Lin, N. (2010). The Bayesian elastic net. *Bayesian Analysis*, *5*(1), 151-170.

Möller, A., Ruhlmann-Kleider, V., Leloup, C., Neveu, J., Palanque-Delabrouille, N., Rich, J., ... & Pritchet, C. (2016). Photometric classification of type Ia supernovae in the SuperNova Legacy Survey with supervised learning. *Journal of Cosmology and Astroparticle Physics*, *2016*(12), 008.

Muniain, P., & Ziel, F. (2018). Probabilistic forecasting and simulation of electricity prices. *arXiv preprint arXiv:1810.08418*.

Yang, H., & Bath, P. A. (2018, June). Prediction of Loneliness in Older People. In *Proceedings of the 2nd International Conference on Medical and Health Informatics* (pp. 165-172). ACM.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301-320.