Airbnb - Pricing Model - Summary Report

Phakphum Jatupitpornchan

This brief aims to provide a brief summary of the implemented prediction process and obtained results. The main goal of the prediction models is to predict the price of the Airbnb apartments in Bangkok in September 2023 which can accommodate 2-6 people.

Code for the analysis can be found in the following link:https://github.com/PhakphumJ/DA3-phdma/tree/main/Assignment%202

Task 1 - Building, Selecting, and Evaluating the Models

Sample Design

I build the prediction models using data on Airbnb listings in Bangkok in March 2023. The whole sample includes both apartments and condominiums. I include condominiums to increase the sample size. Condominiums are more similar to apartments than other types of properties such as houses or villas. The sample only include listings that can accommodate 2-6 people. 20% of the apartments in the whole sample are set to be the hold-out set.

Label Engineering

The target variable is daily price of the listing in Thai Baht (THB). I use the level of the price as the target variable in model building without any transformation.

Although the price may have a non-linear relationship with the important predictors, I choose to capture this by using quadratic terms of the predictors instead.

Feature Engineering

Feature Selection

I use almost all the variables in the dataset as features. The variables that are excluded are those without hints of being relevant for price prediction. For example, *host_name*, *latitude*. For the full list of the features, please see the technical report.

Feature Transformation

Apart from transforming categorical variables and amenities into dummy variables, I also create a few new variables which may be useful for prediction. For example, I created variable indicting the time since the last review and variables capturing whether the accommodation is near a rail transit station. Variables indicating the number of bathrooms are also created.

Dealing with NAs

There are a few variables with NAs. The reasons for the missing values maybe from the hosts not providing the information or the listings/hosts have not gotten any requests/reviews yet. Examples of variables with NAs are *host_response_rate*, *review_scores_rating*, and *description*.

The details of the approach to deal with NAs are in the technical report.

Model Building

I consider three main types of models: linear regression, random forest, and bagging. Linear regression is simple and offers high interpretability, random forest offers high-quality prediction. Bagging is chosen to demonstrate the gain from decorrelating the trees.

The details of each model will be discussed in the following sections.

OLS

8 linear regression models are considered. 5-fold cross-validation is used to select the best model. The one that with the lowest RMSE is chosen.

The selected model is Model 2. The predictors in the model are number of people that can be accommodated, number of beds, number of bedrooms, type of room, type of property, and number of bathrooms.

Details of the models considered are in the technical report.

Random Forest

Next, I build a random forest model. 5-fold cross-validation is used to tune the parameters (*mtry*: Number of variables randomly sampled as candidates at each split). To save computation time, I only use 200 trees.

$mtry = 20$ produces the lowest RMSE.

Bagging

I build a bagging model with 200 trees. 5-fold cross-validation is used to tune the parameters (*minsplit*: Number of minimum observations in the terminal nodes).

The optimal *minsplit* is 6.

Evaluation and Diagnostics

Before evaluating the model. Each model is re-estimated using the whole work set.

Then, the models are evaluated on the hold-out set. The RMSE/Mean(Price) of the models on the hold-out set are presented in Table 1. While the OLS performs reasonably well, is significantly outperformed by the bagging model. However, the random forest comes out on top. It performs better than the bagging model because the benefit from decorrelating the trees. Nonetheless, the difference is small.

Table 1: RMSE/Mean(Price) of The Models on The Hold-out Set

	OLS	Random Forest	Bagging	Mean price
Small apt.	1.04	0.77	0.80	1846.50
Large apt.	0.65	0.49	0.51	3480.42
All apt.	0.88	0.66	0.68	2307.35

Looking at the performance across the size of the apartments, the random forest model also performs the best for all sizes. One interesting observation is that all of the three models perform significantly worse for the small apartments. The performance of the OLS for small apartments is 60% worse than the performance for large apartments (RMSE/Mean(Price) increases by 60%). The random forest model and bagging model both performs 57% worse for small apartments than for large apartments.

I extract the the table showing the results of random forest model in the case study and show it in Table 2.

Table 2: Performance of Random Forest Model in the London Case Study

	RMSE	Mean price	RMSE/price
Small apt.	28.53	62.3	0.46
Large apt.	62.11	144.6	0.43
All apt.	42.36	88.8	0.48

Comparing with the results in Table 1, models built in this exercise are not as successful as the models in the case study, especially for the small apartments. This might be because of the sample design, sample size, or number of trees. However, I believe that a major factor is the difference in the features. In Bangkok's dataset, there is no information about cancellation policy, which is shown to be an important feature in the case study. Another potentially important difference in feature is the time since the first review. For this exercise I opted for the time since the last review instead.

Task 2 - Predicting on the Live Data

I take the three models and predict the price of the apartments in the Bangkok in September 2023 which can accommodate 2-6 people. The performance of the models are presented in Table 3.

Table 3: RMSE/Mean(Price) of The Models on Live Data

	OLS	Random Forest	Bagging	Mean price
Small Apt.	4.13	4.06	4.18	2552.30
Large Apt.	0.96	0.66	1.00	3852.46
All Apt.	3.15	3.06	3.19	2912.05

The order of the performance is the same as in the hold-out set. The random forest model performs the best, followed by the bagging model, and the OLS model. However, all models perform significantly worse than in the hold-out set. This is true across all sizes of the apartments, especially for small apartments. This might be because the relationship between the price and the features changes from March to September. March is the low season in Thailand while September is the high season. Another reason might be the extreme outliers in the price of the apartments in September which I discuss in the technical report.

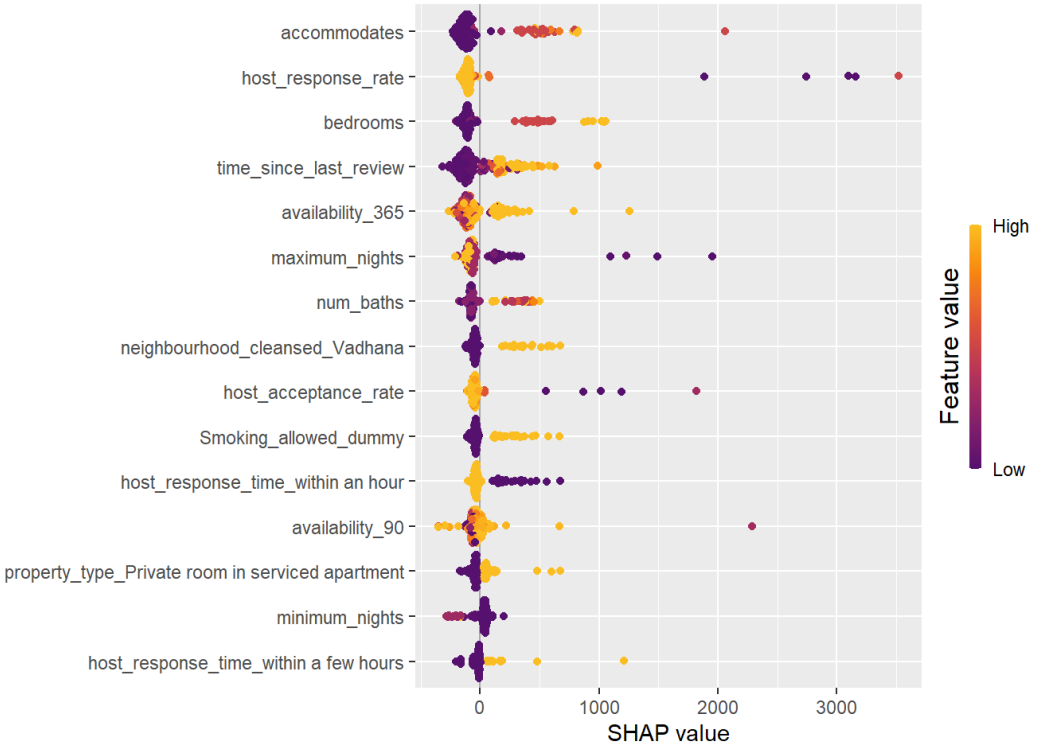
Task 3 - Explaining the Predictions from ML Model.

In this summary report, the distribution of SHAP value of the top features in the random forest model is presented by using a beeswarm plot (using observations in the hold-out set). It shows how the values of the features affect the predicted price of each observation.

accommodates is shown to be the most important feature. The effect of *accommodates* is also consistent with the usual expectation. The more people that can be accommodated, the higher the predicted price.

For most features, the effect is consistent with the usual expectation. For example, having number of beds lower than the average decreases the predicted price. Or, when smoking is allowed, the predicted price is higher.

Figure 1: Beeswarm Plot



However, there are some features that have opposite effect from what we would expect. These are *host_response_rate*, *time_since_last_review*, and *host_response_time_within an hour*. I expected that when the time since the last review is longer, the prospect guests would be less likely to trust the reviews and the predicted price would be lower. However, the plot shows that the predicted price is higher when the time since the last review is longer. I discuss further interpretation of these results in the technical report.