GitHub: https://github.com/szilvasipeter2000/Data-Analysis-3/tree/main

DA3 Assignment 2

Peter Szilvasi

2024-02-10

Introduction

Our objective in this analysis is to create an effective pricing strategy for the Client's new apartments. Departing from a conventional market pricing, we will build a price prediction model based on Inside Airbnb's dataset in Melbourne.

Data

Source and Overview

Initially we started by downloading the dataset and uploading it back to GitHub. The source is Inside Airbnb, and the chosen date was 2023.09.04. This is not the most up-to-date data, but it is the last one that contains information about amenities of the apartments.

Initial filtering includes dropping columns which are mainly NAs, empty lists, URLs, useless IDs, or variables that are non-relevant for the business case, such as host information. This means that initially the dataset includes 23,185 observations and 33 columns.

Data Cleaning

The downloaded data was very dirty and required many hours of cleaning. The main cleaning process was done in Python, while the categorization of amenities into amenity groups was done in MS Excel.

During the cleaning process, the price and bathroom variables were converted into numerical columns, and 73 dummy variables were created to incorporate all of the 3946 unique values of the original amenities column. This process involved a lot of micro decisions on how to group these values. In the end these were the top categories with their respective unit counts:

dummy	soap_shampoo	tv	oven	stove	soundsystem	fridge	wifi	pool	kidfriendly
count	1384	447	374	363	302	285	212	56	38

Feature Engineering

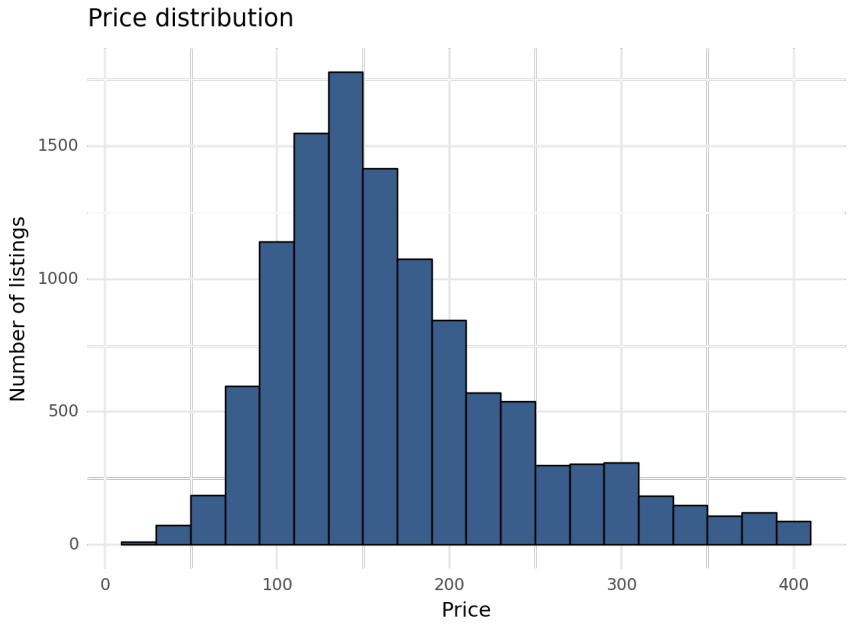
The feature engineering process included creating flags and imputing for not very important variables that were missing rows on the hundreds, or even thousands scale. These columns were mainly the review scores for certain aspects, such as cleanliness, check-ins, or location.

We also pooled or grouped property types, bedrooms, bathrooms, minimum nights, number of reviews, and availability. This was done to reduce the dimensionality of the data and to ensure that there are enough data points for each category.

Exploratory Data Analysis (EDA)

The most important decision of the EDA was to decide where to cut off the prediction variable's values. The distribution of the price variable is strongly skewed to the right, meaning that the more we cut-off from its higher values, the more accurate our model will be at the end.

Our decision was to restrict the values at the 90th percentile, which is 400 dollars.



LASSO Model for Variable Selection

After the cleaning and filtering of our data, we still had over 10,000 observations so it was time to build our models. As a variable selection method, we used LASSO to check which Airbnb features should be used for the prediction models, as well as which interactions to choose. The running of the LASSO model required the use of 395 variables, including interactions. As a result, we got the theoretically best performing OLS variables and we saw that the only significant interaction was between room type and property type.

Model Selection

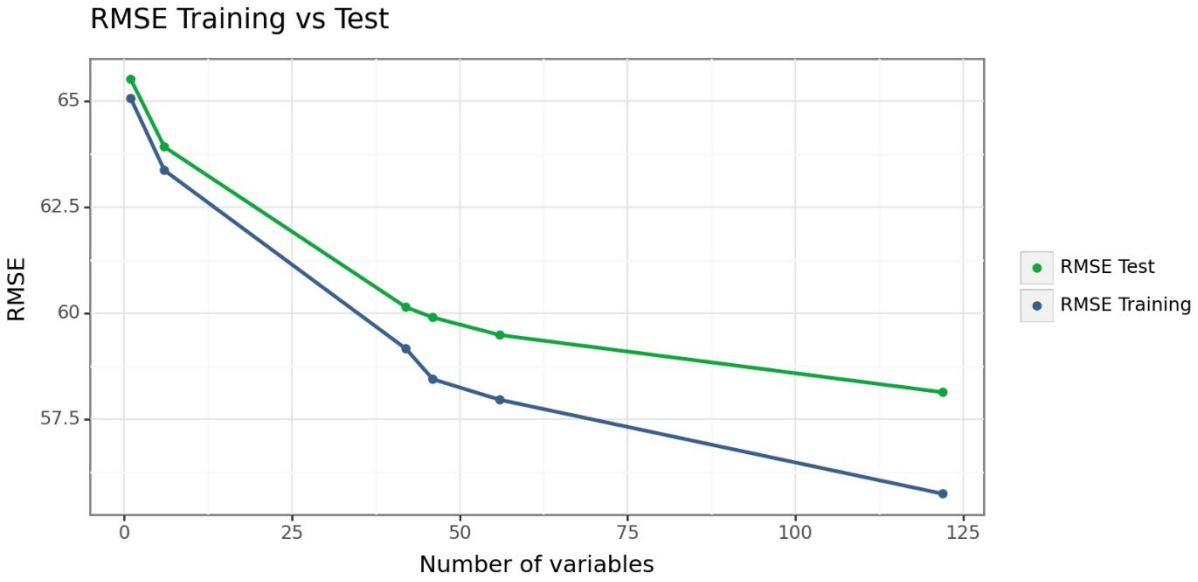
We created 4 Models for this part of the analysis. 2 was based on the theoretically best performing OLS with Lasso using Grid search, and a Cross-Validation OLS Regression. The remaining two was a Random Forest and a Gradient Boosting Model. At the end we will compare the 4 with their respective RMSE values and decide which we wish to use. For all the models we applied the 80-20 rule for the training and holdout set.

LASSO and CV OLS

These 2 were fairly similar as both are using the same variables and interactions, with the same linear modelling method, however LASSO uses a penalty term to penalize for the inclusion of unnecessary variables.

The results were as expected. The LASSO Model outperformed the theoretically profound OLS Model with an RMSE of 56.7 against the OLS's 58.13. On the RMSE vs Test graph, we can see the improvements made by adding extra variables. Even the simplest model only has an RMSE of around 65. There is a significant increase from model 2 to 3, where we add bathrooms, neighborhoods, availability throughout the year and minimum nights.

An interesting part when adding the dummies, the Test and Training RMSE starts to diverge from each other. This could be a sign of early overfitting. The training data learns the patterns too well, capturing noise in the process. Even after this, the most complex model is still the best performing with almost 125 variables.



Random Forest

The theoretical recommended number of variables for the Model is 11.53 so we set the max features to be maximum 12. It used the same basic set of variables as the previous models.

Its performance was slightly better than the previous 2 models, with an RMSE of 55.69. It also gave the feature importances for the model. The most important variable is bedrooms. If we were to neglect this variable, our prediction accuracy would decrease by 8.37%. Accommodates, number of bathrooms, number of beds are following closely, just as reviews per month, with values from 7.93% to 4.17%. Also including the dummy variables was a good choice as starting with whether the Airbnb has outdoor opportunities helps the model with 2.52% prediction accuracy. Before reaching 100% cumulative importance almost all the 73 dummy variables are listed. Thus, we can safely say giving up on up-to-date data in fact made the model more accurate.

Gradient Boosting Model

The GBM emerged as the top-performing model in the analysis, achieving the lowest RMSE of 52.73. This performance can be attributed to its fine-tuned parameters we applied: Max depth of 5, 10 max features, 20 minimum sample splits, and 300 number of estimators.

Model Performance as Conclusion

When deciding which model to use we looked at the following table, which is sorted by performance.

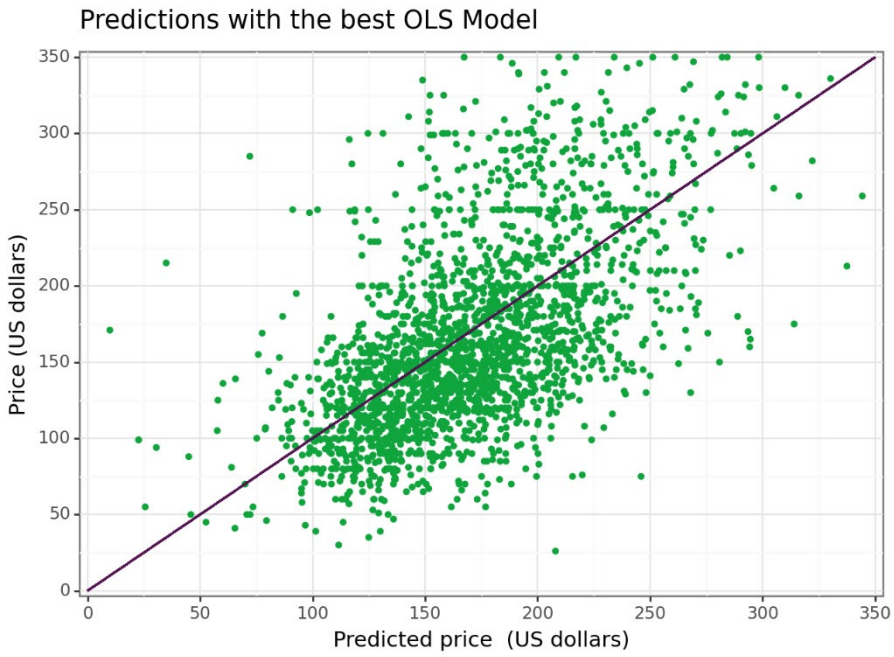
Table 1.	Model	RMSE	Running Time
0	Gradient Boosting Model	52.731020	6.43 s
1	Random Forest	56.030508	48.8 s
2	Lasso OLS	56.701704	3 m 5.9 s
3	OLS CV	58.144098	11.9 s

As seen in Table 1, our GBM Model emerged as our preferred choice due to its outstanding performance in minimizing the mean squared errors and the running time of the model. The Lasso took the longest time, however it was an essential

step in the variable selection process so without the Lasso, there is no GBM. We can also mention as conclusion that the number of bedrooms, accommodates, bathrooms, beds, and reviews per month are the most important features of an Airbnb.

Limitations and Room for Improvement

Note, that these models are predicting with a price range that is quite wide. If we were to narrow it down to 80th percentile instead of 90, we would get much lower RMSE values.



On this graph, we can see that if we were to restrict our price variable to less than 250 or even lower, our mean square errors would drastically decrease. As for those values the predicted values are way closer to its actual value, while above 250, the predictions start to become lower from their actual values.

Just to see, these would be our values if we were to restrict our price column at the 80th percentile, meaning that if price couldn't be higher than 293 dollars.

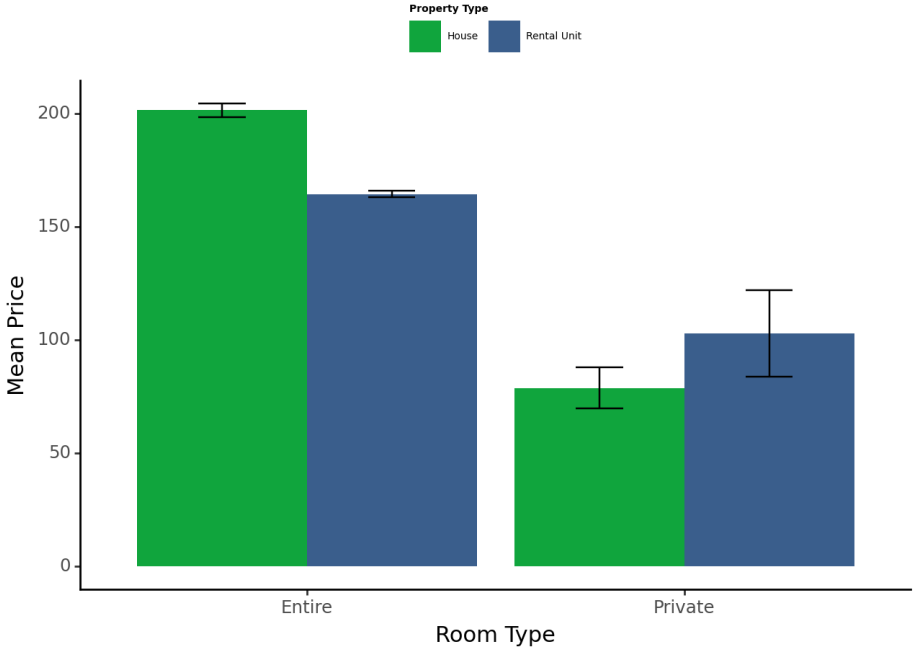
	Model	RMSE	Running Time
0	Gradient Boosting Model	40.642169	6.23 s
1	Random Forest	42.841533	46.4 s
2	Lasso OLS	43.187603	2 m 59 s
3	OLS CV	44.155009	11.2 s

Our RMSE values would drastically decrease. The model performance order wouldn't change. Even with this filtering of the price variable, we still have above 10,000 observations, personally as a Data Analyst I feel like restricting the prediction variable at its 80th percentile is too strict.

However, whether the Model should use this price range or the previous one, should be the decision of the Company we are helping, as they know if they want to be above the price of 283 dollars or not.

Appendix

Interaction between Room Type and Property Type



LASSO Regression