Technical Report: Building Price Prediction Models For Copenhagen Apartments

Shahana Ayobi

2023-02-07

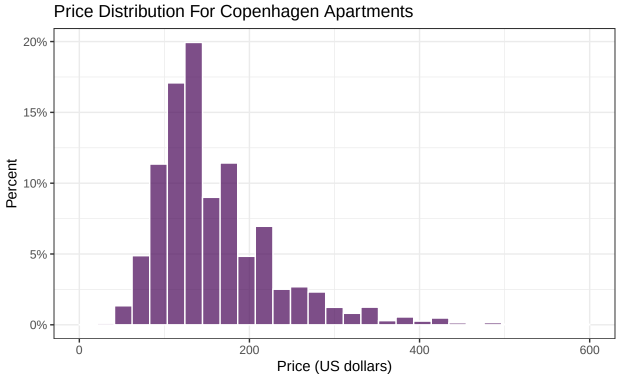
Introduction

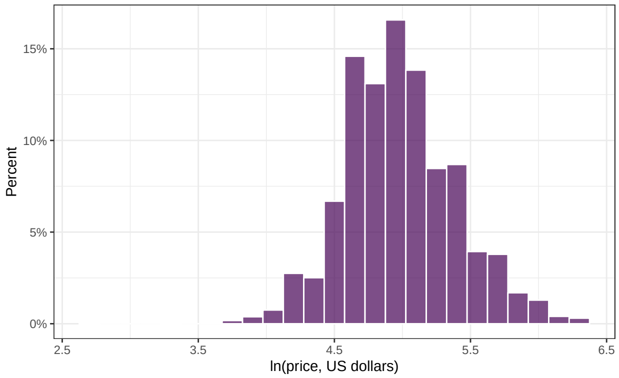
This report's objective is to give a thorough explanation of the price prediction model. The primary objective of this project is to assist a business in setting a price for brand-new flats that have not yet hit the market. The data is gathered from Inside Airbnb, which can be obtained here, to develop a price prediction model for a business operating small and mid-size apartments hosting two to six guests in Copenhagen, Denmark. Five price prediction models—OLS, Lasso, Cart, Random Forest, and GBM, Gradient Boosting Machine—are produced as a result of data cleaning, munging, and analysis. GBM therefore displayed the best prediction outcome with a 65.38 USD RMSE. The most important predictor features are the neighborhood, reviews per month, review scores rating, number of accommodates, property type, the number of bathrooms, and the number of beds. Other factors, such as the number of days since the first review, are also significant predictors. The project's ultimate goal is to complete a better prediction model as measured by relative RMSE values.

Data Cleaning

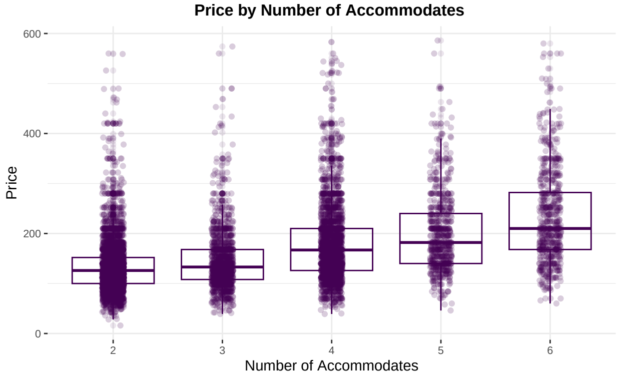
The original data set consists of a single data table with 75 columns and 13,820 observations. The following are some of the crucial columns: ID, price, reviews per month, number of accommodates, property type, amenities, and other features specific to the host and rental property. The information relates to one-night rental rates for the period of December 29, 2022. Price per night, per person, expressed in Danish Krone, is the desired variable which is then converted to US dollars by by multiplying it to market exchange rate of 0.14. Because working with Tidy data tables is easier, the original data set needs to be cleaned because it contains a lot of information. At first, I removed columns for the host's website, the host's profile photo, the house rules, notes, and other information. Because this project does not include these columns as a target. Additionally, transforming the "amenities" column into binary variables makes up a large portion of the transformation procedure used to convert the original data into a tidy data table. Only dummy variables for amenities that were important for the analysis have been created such as having TV, coffee maker, wifi, heating, microwave and so on. More information on the data cleaning can be found on the code. Additionally, following the initial stage of cleaning, follows data preparation. The following phases make up the process of preparing data for analysis: determining the types of the variables, getting rid of duplicates, and dealing with missing values. The chosen data collection, Airbnb Copenhagen, includes binary variables like all amenities and host attributes along with numeric values like price and factor variables like neighborhood and property type. Thus, I generated factor variables for predictors like neighborhood and property types.

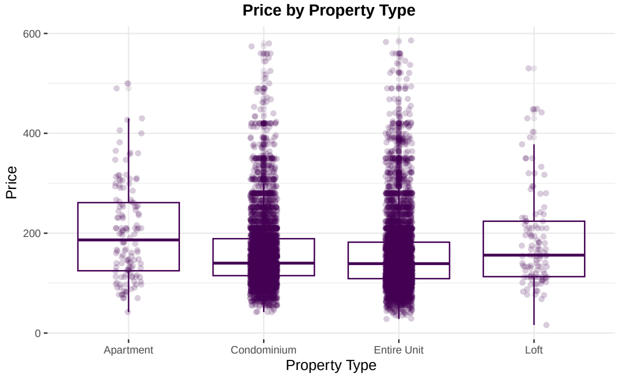
Filters The data was filtered in accordance with the project's primary objective of predicting apartment prices for units that fall between 2 and 6 accommodates. Price per night included extreme values exceeding 1000 USD per night, which comprised fewer than 1% of the observations, therefore price was filtered to less than 600 USD and the observation where price is missing was dropped. Additionally, it is essential to examine prices and the log of price distribution because the project's objective is to create a price prediction model. As shown below the price distribution is skewed with a long right tail while log price distribution is close to normal. However, prediction is carried out on price per night for model simplicity.





Additionally, because the primary objective of this study is to develop price predictions for small and mid-size flats, the following categories for property types are filtered for the analysis: entire home or apartment, entire serviced apartment, entire condominium, entire loft, entire rental unit. According to the dictionary, there are various sorts of apartments, including lofts, condominiums and a rental unit referring to different types of apartments. The data set demonstrates the existence of a single room type. The figures below show the number of guests, and property type, as well as the mean prices for each which shows with increased number of accommodates price increases. While the property types apartment and lofts are higher priced than the two other categories. Also, room type variable have been dropped from the data It shows that there is only one room type across the entire data set.





New Variables The second step in the process is to create new meaningful variables, such as the number of days since the first review, which is calculated by subtracting the date of the first review from the date the data was scraped and generating square, and cubic functional forms of the number of days since the first review and square form number of accommodates.

I have also grouped some numerical variables, such as the number of bathrooms. This variable contained data such as half bathroom. Another variable was created as a factor of bathrooms, with four cuts of 0, 1, 2, and 10. This means combining all of the variables in the aforementioned groups. Another example is the number of reviews for 0 to 51 and 51 and higher.

Other variables with missing values were addressed as follows: the first assumption is that each apartment has at least one bathroom; The missing number of beds was placed equal to number of accommodates, assuming the apartments having single beds. It is assumed that the minimum number of reviews is one. To indicate missing values in each predictor, flags were created.

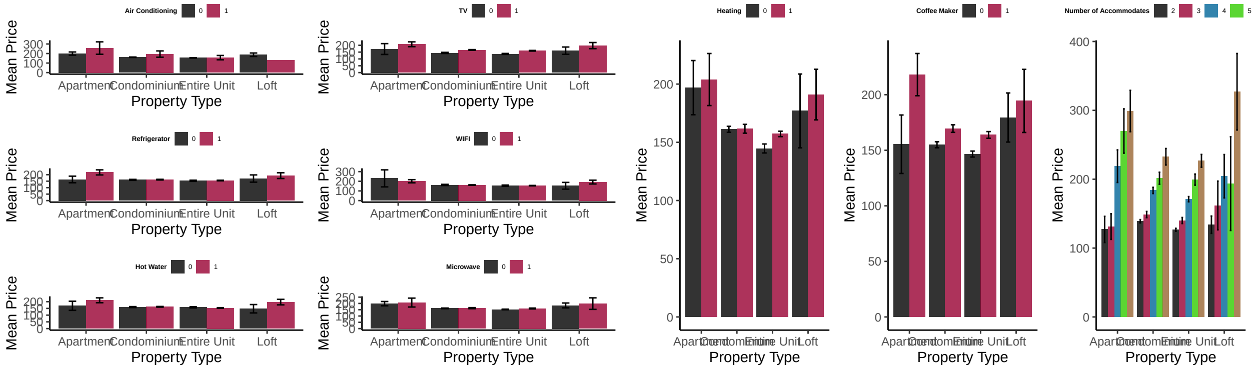
As a result of data munging and preparation, the cleaned dataset has 9,244 observations and 54 variables.

Data Analysis and Feature Engineering

Feature engineering entails deciding on the type of predictor variables to include as well as the functional forms of predictors and potential interactions. The predictors are grouped as follows:

- **Basic variables** include the main predictors such as the number of accommodated, number of beds, property types, number of days since the first review, and its flag variable.
- **Basic addition** This includes key factorized variables like neighborhoods minimum nights.
- **Review Variables** include important guest review predictors such as review score rating and its flag indicating missing values, and factored total number of reviews.
- **Polynomial level** is made up of squared terms for number of accommodates as well as squared and cubic terms for days since the first review.
- **Dummies**, which included binary values for all amenities.

Finding the right interactions is the next step in the process. The plots below were used to visualize price changes across each interaction.



Based on the plots above, I created three types of interactions: X1 which includes property type multiplied by the number of accommodates. X2 contains property type, air conditioning, refrigerator, WIFI, coffee maker, microwave and hot water dummy variables and X3 includes property types times neighborhood, accommodates times neighborhood groups, and all amenities.

Modeling

The best model provides the best prediction in the live data. The original cleaned data is split into two random parts by 20% to 80% ratio in order to avoid over-fitting. The Holdout set includes 20% and the rest 80% is work data set. Then, the work dataset is split into train and test datasets and 5-fold cross validation is run on the train dataset. Then the best model is chosen based on the lowest average of 5 CV RMSE result. Eight basic OLS regression models from simplest to the most complex one were utilized to find the best model for our analysis. 5-fold cross-validation RMSE suggests that Model 7 regression has a better performance and it has the lowest RMSE test value of 59.2 dollars with 61 variables. The table is as following:

Table 1: Evaluation of 8 Cross Validated Models

model	coefficients	BIC	R2	RMSE_train	RMSE_test
(1)	1	82951.52	0.1695	65.8328	65.8492
(2)	9	82694.77	0.2056	64.3890	64.5059
(3)	24	81635.12	0.3239	59.3987	59.6419
(4)	27	81645.76	0.3254	59.3341	59.6141
(5)	30	81631.52	0.3291	59.1700	59.4797
(6)	58	81752.45	0.3407	58.6582	59.2983
(7)	61	81761.75	0.3422	58.5891	59.1922
(8)	105	82016.17	0.3544	58.0468	59.5269

Models

OLS Model

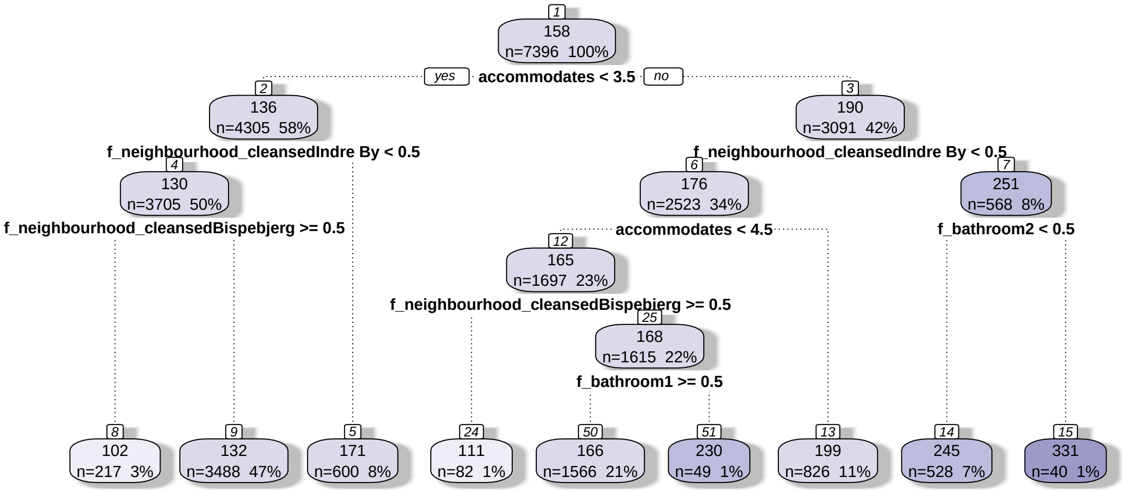
The OLS model, which has a CR RMSE of 59.2742 an R-squared of 32.7% out performs GBM model with 105 predictors.

LASSO

LASSO is an algorithm that fits a model by shrinking coefficients, some of which are reduced to zero by the addition of a penalty term. After performing 5-fold cross validation to determine the best value for lambda. Before running the models, the lasso tuning parameter is set, which acts as a weight for the penalty term versus the OLS fit. As a result, the strength of the variable selection is determined. The tuning parameter, Lamda, is set to a value 0.01 to 0.25 and the best tuned lambda with lowest RMSE is chosen as 0.01. LASSO outperforms all models in Holdout set with RMSE value of 57.6185 dollars and 5 fold cross validated average RMSE of 59.0655 dollars which is lower than GBM model.

CART

CART also refereed as regression trees is an algorithm that has no formula; the goal is to produce a set of predictor bins. The bins are divided into smaller bins by this algorithm, and no functional forms or interactions are provided for CART.The CART model displays a 62.268 USD RMSE which is the highest among all models, thus, CART does not work well for our prediction.



Random Forest

Random forest is an ensemble method in which the results of multiple predictive models are combined to generate a final prediction. For this model, I used basic variables, basic additions, reviews, and dummies with a minimum nod size of 50. The RMSE for the Holdout set is 58.497 dollars while the cross validated RMSE is the lowest among all models with value of 58.79 dollars.

GBM

GBM is the final model for this case study. In terms of relative RMSE for 5-fold cross validation, it under performs OLS, LASSO, and Random Forest of 59.3699 dollars. The Rsqaured is also lower than OLS and LASSO with value of 32.54%.

Result The Random Forest model has a relatively better performance for 5-fold cross validated work set, while LASSO works best for holdout set, as can be seen from the table of models below. The data's 5-fold cross validation RMSE is 58.79 dollars for Random Forest which is the lowest among all models, which is 0.2735 dollars less than the RMSE for LASSO which is negligible. However, LASSO gives lowest RMSE for holdout set and the highest R-squared with value of 33.21% despite penalizing for interaction terms and shrinking some coefficients to zero. Considering external validity and model performance for live data, LASSO is chosen as the best model that has relative better performance in both sets.

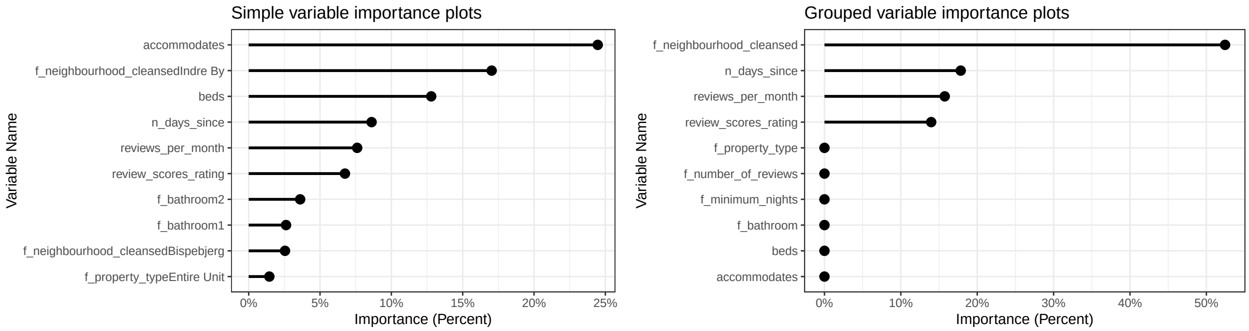
Table 2: Evaluation of Models

	CV RMSE	Holdout RMSE	CV Rsquared
OLS	59.2742	57.6412	0.3270
LASSO	59.0655	57.6185	0.3321
CART	62.2688	61.4069	0.2581
Random forest	59.9218	58.1490	0.3182
GBM	59.3699	57.8733	0.3254

Diagnostics

LASSO is an algorithm that fits a model by shrinking the coefficients and even shrinks some of them to zero is chosen as the best model. Diagnostic tools, on the other hand, can be used to uncover information about the patterns of association that drive prediction. Some examples are as follows:

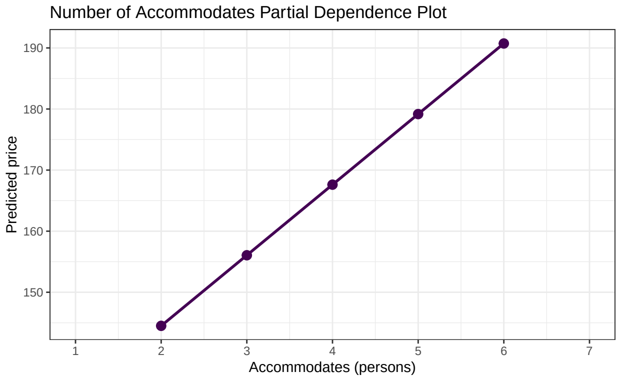
Variable Importance Plot depicts the average importance of fit when an x variable or set of x variables is used. Plot of variable importance for The top ten most important variables show that number of accommodates and beds, Indre By neighborhood, review variables are the most important. The importance of grouped variables reveals that neighborhoods, reviews, and days since first review are important.

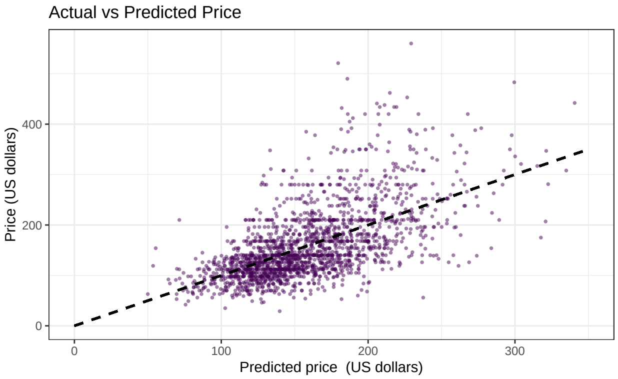


Partial Dependence Plot depicts how average y varies for different x values in relation to all other predictor variables. The partial dependence plot is based on the holdout set's predictors. The partial dependence plot for the number of accommodates and the price shows that the price rises as the number of accommodates increases.

Performance Across Subsamples Examining the fit in different subsamples can provide information about the prediction's external validity.

Actual vs Predicted Price Another post-prediction diagnostic is a comparison of predicted and actual prices. The graph below shows that prediction performs better for lower prices than for higher prices. Howeve, for prices above 200 dollars, the model seems to not fit well.





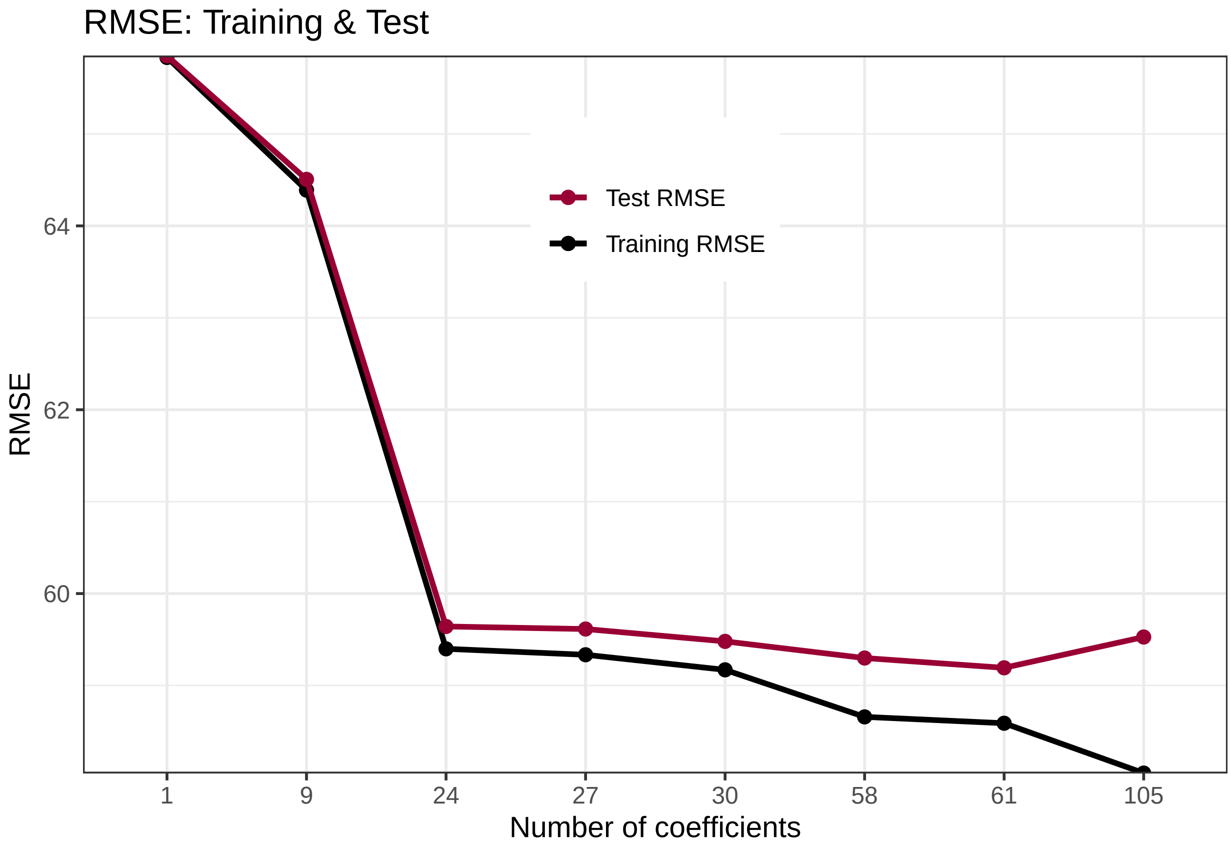
Conclusion

The purpose of this report was to develop a more accurate model for predicting Airbnb prices in Copenhagen for small to mid-size apartments. Five models were depicted in order to compare and contrast their performance. The basic LASSO model with lambda value of 0.01 , which highlights meaningful characteristics about the nature of Airbnb apartments in Copenhagen was the best model with a 59.065 dollars RMSE, while it performed best in the holdout set with a 57.618 dollars RMSE. While OLS came in second place with 57.64 dollars RMSE for holdout set. In this study, simple models such as LASSO and OLS outperformed complicated models such as GBM and CART. Therefore, LASSO is chosen as the best model; however, results can be different depending on the dataset. The neighbouhood, days since first review, property type, number of bathrooms, the number of accommodations, review scores rating and reviews per month are key price drivers based on post prediction diagnostics.

Appendix

Table 3: Copenhagen Airbnb apartment price prediction Models

Model	Predictors
M1	Num of accommodates
M2	M1 + number of beds + property type + number of days since first review + room type
M3	M2 + Num bathrooms + Neighbourhood group + host reponse rate + reviews per month + reviews scores rating, flag review_scores rating", number_of_reviews
M4	M3 + squared termof guests + squared and cubic terms of number of days since first review
M5	M4 + property type and number of guests interaction + property type and room type interaction
M6	M5 + property type interaction with dummies as air conditioning, TV, wifi, coffee_maker, microwave, hot_water
M7	M6 + all other amenities
M8	M7 + all other amenities, Neighbourhoods interacted with property type


RMSE: Training & Test
Test RMSE
Training RMSE
RMSE
Number of coefficients