Assignment 2: Technical Report

Alima Dzhanybaeva

Introduction

In this project, I tried to find the best price prediction model for mid-size apartments in Los Angeles. The results for 5 models (OLS, LASSO, CART, Random Forest, GBM) showed the lowest RMSE of 73.69 for *Random Forest*, suggesting the best predictive performance.

Cleaning data

The original dataset for Los Angeles includes 40438 observations and 75 variables, and was scraped from Airbnb on December 7th 2022.

Dropping variables:

- Firstly, the variables that won't be helpful for the prediction of prices (i.e. 'listing.url', 'scrape_id', 'description', 'picture_url' etc.) were dropped.
- After the check for missing values the following variables that are assumed to be not that significant and included a lot of NAs were also dropped: 'host_neighbourhood', 'neighbourhood' ('neighbourhood_cleansed' will be used instead), 'bathrooms' ('bathroom_text' will be transformed and used instead), 'bedrooms', 'last_review', 'host_response_time', 'host_response_rate', 'host_acceptance_rate'.

Filtering:

- *Price*: The 'price' variable was filtered for values that are more than 0 (as it does not make sense when the place is rented fo free) and less than 800 (800 is 95% percentile).
- *Property type*: The goal of the project is to predict prices for apartments, therefore, the dataset was filtered for following property types: "Entire condo","Entire loft", "Entire serviced apartment", "Entire home/apt", "Entire rental unit".
- *Number of accommodates*: As the mid-sized apartments are the focus of this assignment, the 'accommodates variable' was filtered for values between 2 and 6.

Transformation of the variables:

- Values for the variables 'host_is_superhost', 'host_has_profile_pic', 'host_identity_verified', 'has_availability', 'instant_bookable' were changed from TRUE|FALSE to 1|0.
- All the text from 'bathrooms_text' was deleted so the variable can be converted to numerical, and the name was changed to 'bathrooms'.
- Factoring variables 'property_type', 'neighborhood_group_cleansed', 'neighborhood_cleansed'.

Creation of new variables on the basis of existing ones:

- *Amenities*: (1) the original variable was transformed into the list, (2) the unique values were observed, (3) dummy variables were created for the features that were the most frequent and are assumed to be significant for prices (TV, WiFi, hot tub, gym, sound system, dishwasher, pool, balcony, fridge, stove, coffee machine, good view, hair dryer, air conditioner, breakfast, wardrobe). The creation of the dummy variable on the example of 'TV' is presented in the code below:

```
df$d_fridge <- sapply(df$amenities,
                      function(x) ifelse(length(grep("fridge", x, ignore.case = TRUE)) > 0 |
                                                              length(grep("refrigerator", x, ignore.case = TR
```

- *New factor variables*: (1) 'f_bathroom' with 3 categories: 0-1, 1-2, 2-5; (2) f_number_of_reviews with 3 categories: 0-1, 1-51, 51-maximum number of reviews; (3) f_minimum_nights with 3 categories: 1-2, 2-3, 3-maximum value for 'minimum nights'.
- *Functional forms*: (1) 'accommodates2' (number of accommodates squared); (2) 'ln_accommodates' (log of the number of accommodates); (3) 'ln_accommodates2' (log of the number of accommodates squared); (4) ln_beds (log of the number of beds); (3) ln_number_of_reviews (log of the number of reviews).
- *Number of days since the first review and its functional forms*: (1) creating 'days_since' variable by subtracting 'first_review' from 'calendar_last_scraped'; (2) functional forms of 'days_since': 'ln_days_since', 'ln_days_since2', 'ln_days_since3', 'n_days_since2', 'n_days_since3'.

Imputation of values:

- For missing values in 'review_scores_rating', 'reviews_per_month', 'n_days_since' '0' was imputed, as NAs in these variables were found for observations that had 0 reviews.
- For missing values in 'bathrooms' median value was imputed.
- For missing values in 'beds' it was assumed that the number of beds is equal to number of accommodates.

After cleaning the data we are left with 11890 observations and 59 variables.

Feature engineering

The following groups of variables were defined:

- *Basic variables*: number of accommodates, property type, number of beds, number of days since the first review.
- *Factor variables*: factor variables for the number of bathrooms, neighbourhood group.
- *Reviews*: factor variable for number of reviews, rating, number of reviews per month.
- *Polynomials*: squared term of number of accommodates, sqaured and cubic terms of the days since the first review.
- *Dummy variables*: dummy variables for amenities, and such as

- *Interaction terms*: (X1) (property type * number of accommodates), (property type * neighbourhood group); (X2) on the basis of the graphs (Appendix: Graph1) it was decided to include the following interaction terms: (property type * (air conditioner, pool, hot tub, dishwasher, TV, Balcony, gym. good view); (X3) (number of accommodates * neighbourhood group), (property type * all dummy variables).

In the table below you can get familiar with 8 models and the variables that were added to each of them:

Model	Predictors
M1	# of Accommodates
M2	M1 + Property Type + Number of Beds + Number of Days Since the First Review
M3	M2 + # of bathrooms + Neighbourhood group + Reviews per Month + Review Score Rating + # of Reviews
M4	M3 + # of Accommodates Squared + Squared and Cubic Terms # of Days Since the First Review
M5	M4 + (Property Type * Number of Accommodates) + (Property Type * Neighbourhood Group)
M6	M5 + (Property Type * several Amenities Dummies)
M7	M6 + all Dummy Vars
M8	M7 + (# of Accommodates * Neighbourhood group) + (Property Type * all Dummy Vars)

Models evaluation

The dataset was randomly divided to working set (80% of observations) and hold-out set (20% of observations):

```
# Create a holdout set

smp_size <- floor(0.2 * nrow(df))
set.seed(12345)
holdout_ids <- sample(seq_len(nrow(df)), size = smp_size)
df$holdout <- 0
df$holdout[holdout_ids] <- 1

#Hold-out set
data_holdout <- df %>% filter(holdout == 1)

#Working set
data_work <- df %>% filter(holdout == 0)
```

Additionally, in order to estimate how well models are performing on a new, unseen data 5-fold cross-validation was implied.

The results of 8 OLS models are presented in the table below:

Model	N predictors	R-squared	BIC	Training RMSE	Test RMSE
(1)	1	0.1376757	111464.4	84.66793	84.65799
(2)	6	0.1843692	110980.7	82.33904	82.38185
(3)	12	0.2597859	110112.7	78.43036	78.57805
(4)	15	0.2633928	110093.8	78.23495	78.42164
(5)	24	0.2653805	110150.5	78.11666	78.43121
(6)	56	0.3023260	109952.8	76.08310	76.83552
(7)	67	0.3183255	109832.9	75.19032	76.08590
(8)	102	0.3257105	110049.9	74.70709	76.40225

Even though **Training RMSE** is the lowest for the *Model 8*, the difference between *Model 7* is negligible. Moreover, **BIC** and **Test RMSE** are the lowest for *Model 7*.

Therefore, *Model 7* model will be used for price prediction with **OLS** and **LASSO**.

And for **CART**, **Random Forest**, and **GBM** basic variables, factor variables, reviews, polynomials, and all dummy variables were used.

Results

According to the results presented in the table below, **Random Forest** showed the best performance with the lowest values for **CV RMSE**, **Holdout RMSE** and the highest value for **CV R-squared**.

	CV RMSE	Holdout RMSE	CV R-squared
OLS	76.06700	75.94949	0.3046565
LASSO	75.89403	75.91083	0.3076597
CART	79.19440	79.92913	0.2467668
Random forest	73.68993	73.42125	0.3502718
GBM	74.17428	74.29148	0.3397116

Diagnostics

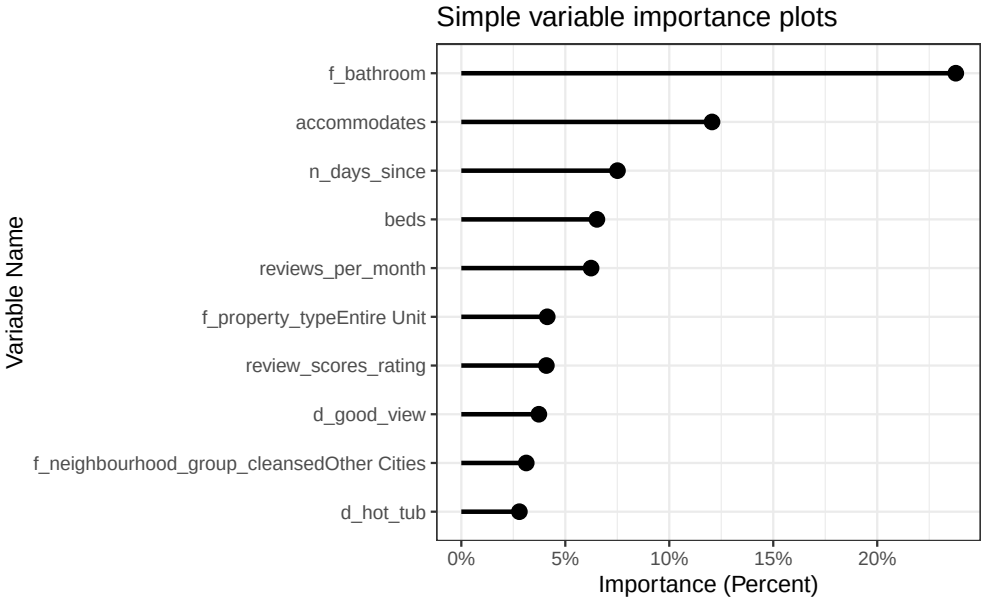
As **Random Forest** is a "black box" method it is quite hard to understand the inner workings of the model and the relationship between the input variables and the predictions, as it is represented by a combination of many decision trees and not by mathematical formula like in linear regressions.

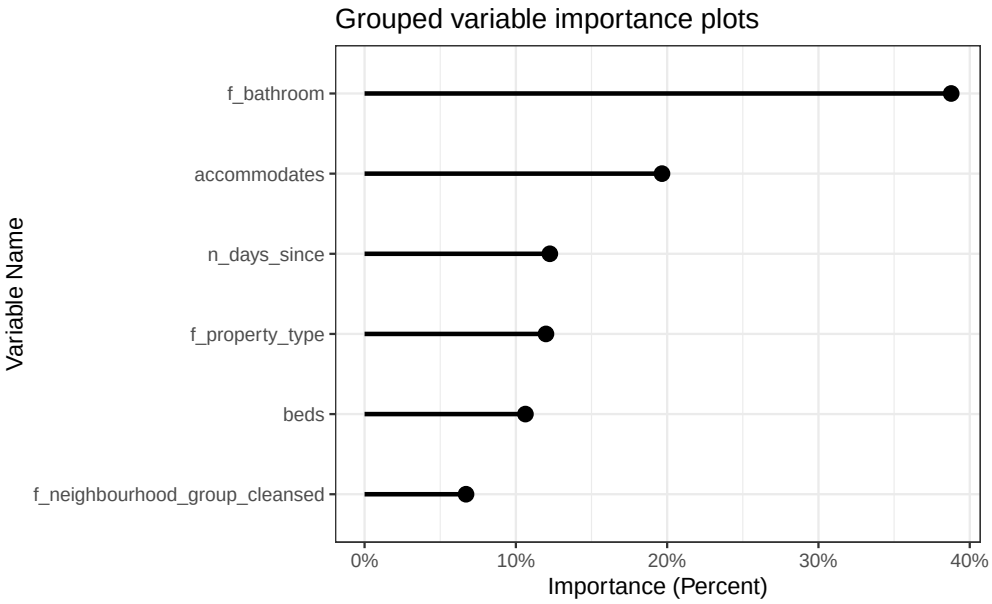
Nevertheless, we can try to uncover the obtained patterns with the variable importance plot, partial dependence plot, comparison of the measures for different subsamples, and plot for actual vs predicted prices

Importance of the variables

It is noticeable that 'f_bathroom' and 'accommodates' are the two most important variables, as they together account for around 30% in the 'Single variable importance plot' and 60% in the 'Grouped variable importance plot'.

Another important variables are: 'n_days_since', 'f_property_type', 'beds', 'f_neighbourhood_group_cleansed', and 'reviews_per_month'.

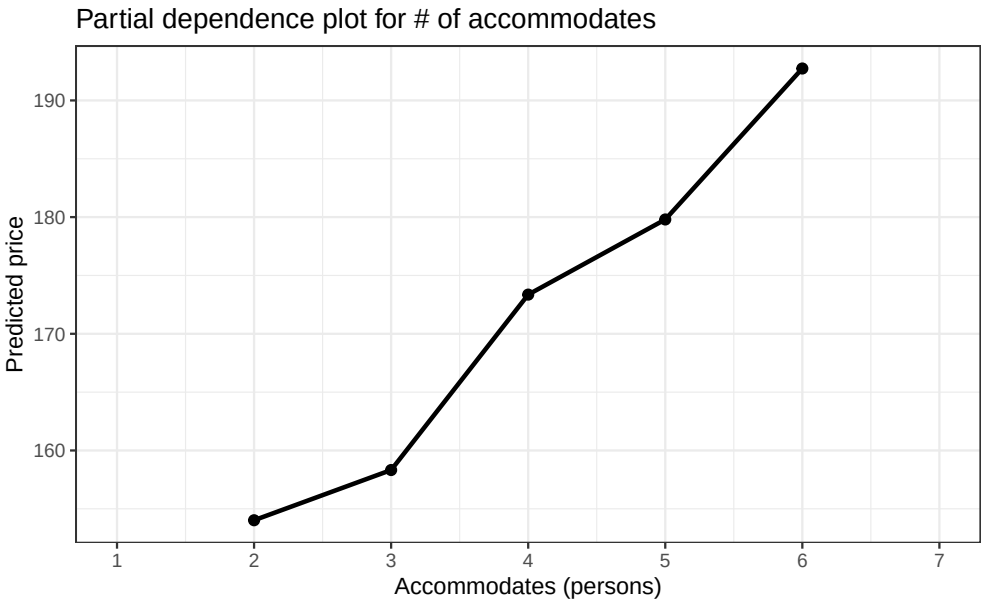


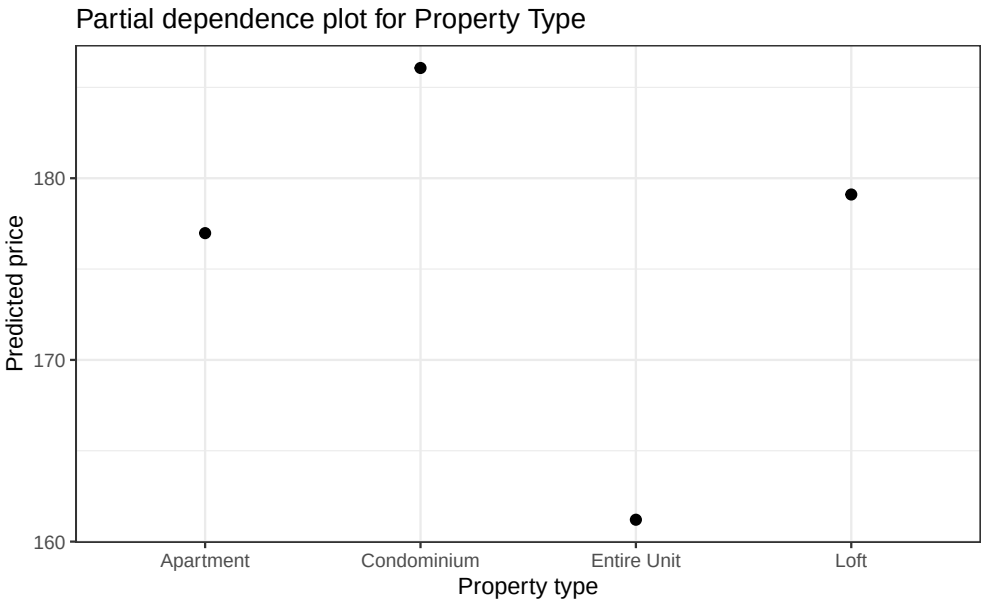


Partial dependence plot

As we can see from the partial dependence plot for the number of accommodates, as the number of guests that can check in into apartment rises the predicted price also increases.

As for the partial dependence plot for property type, the lowest predicted price is observed for 'Entire Unit', while the highest is for 'Condominium'.





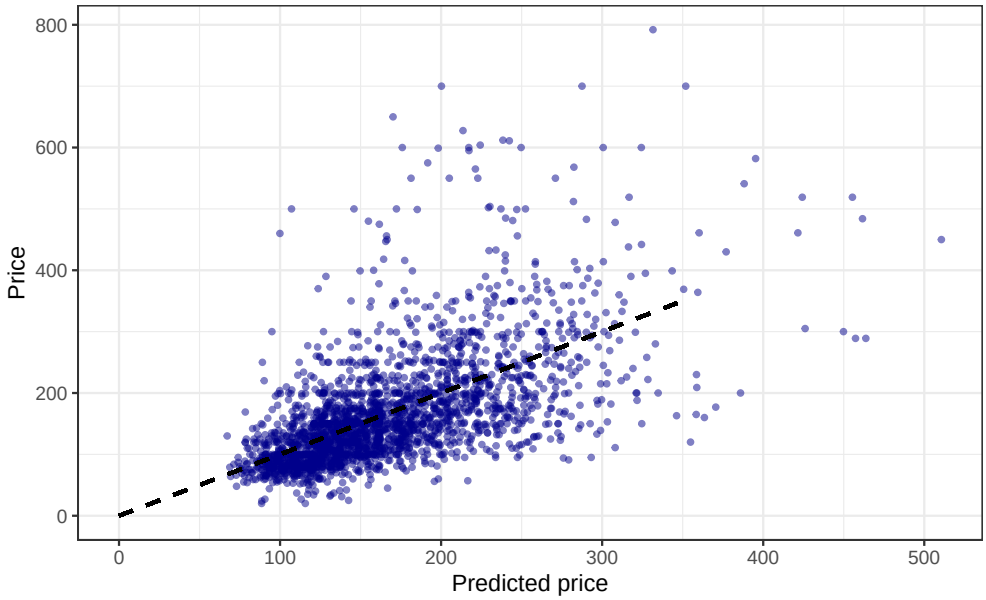
Performance across different sub samples

From the table below we can observe, that **Random Forest** does a better job with predicting prices for 'small apt', "Entire Unit', and 'City of Los Angeles', while the measures are worse for 'bigger apt', 'Condominium', and 'Other Cities'.

Var.1	RMSE	Mean.price	RMSE.price
Apartment size			
bigger apt	88.78435	203.6750	0.4359118
small apt	56.77949	135.8250	0.4180341
Type			
Apartment	89.23683	200.0930	0.4459767
Condominium	95.39489	220.3160	0.4329913
Entire Unit	66.45633	155.1145	0.4284340
Loft	87.30834	173.5243	0.5031477
Neighbourhood			
City of Los Angeles	67.75441	153.6120	0.4410749
Other Cities	80.58797	186.1039	0.4330268
Unincorporated Areas	74.93016	173.5000	0.4318741
All	73.42125	167.3818	0.4386453

Actual VS Predicted prices

The figure below implies that the model is performing better for cheaper apartments.



Conclusion

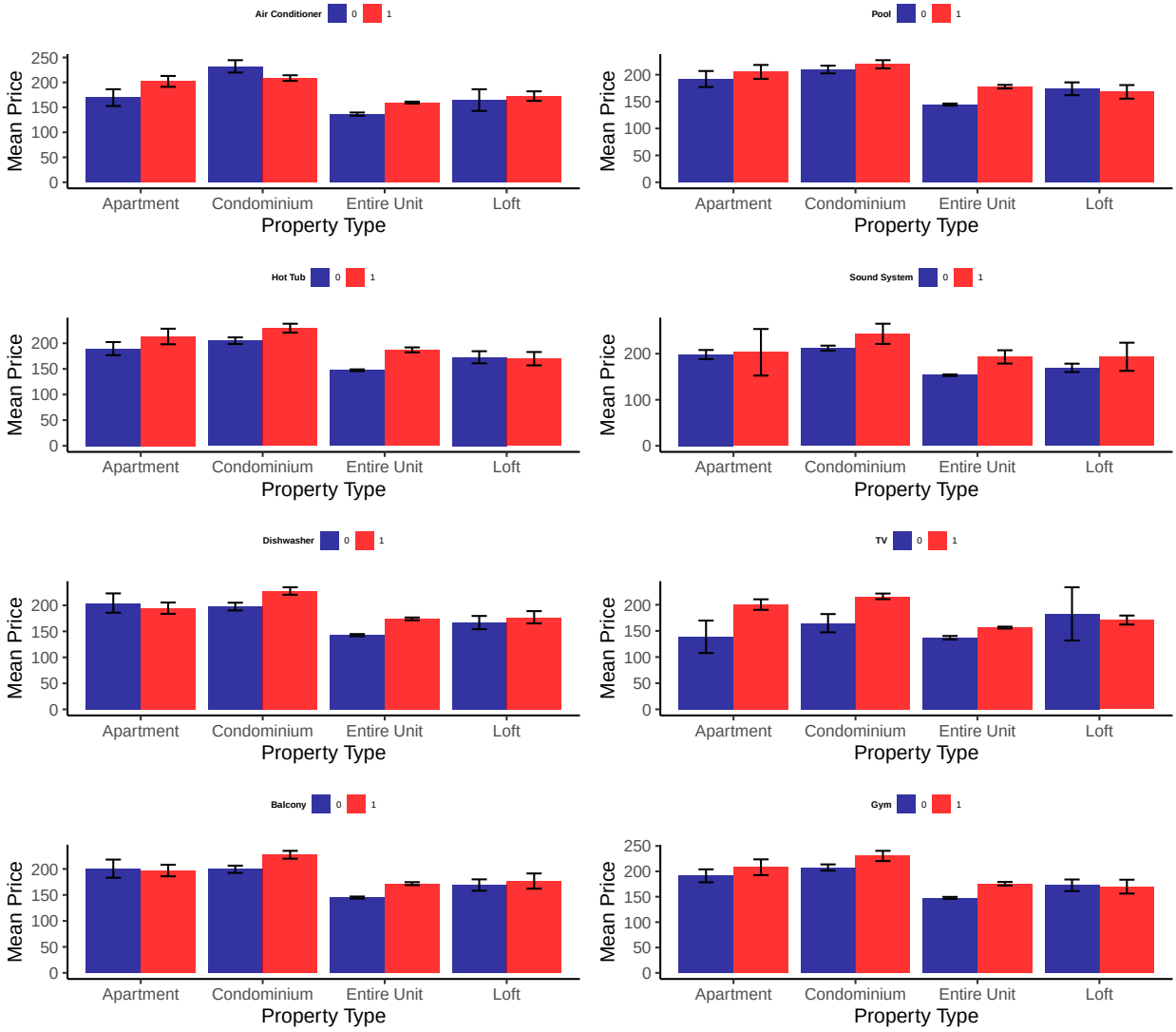
This project tried to find the model that will predict prices for mid-size apartments in Los Angeles. The obtained values **CV RMSE**, **Holdout RMSE** for all 5 models (OLS, LASSO, CART, Random Forest, GBM) showed that **Random Forest** has the best performance.

Using variable importance plots it was found out that 'f_bathrooms' and 'accommodates' have the biggest influence on the prices for apartments. The high importance of these two variables is not surprising, as their higher values suggest the bigger apartment size.

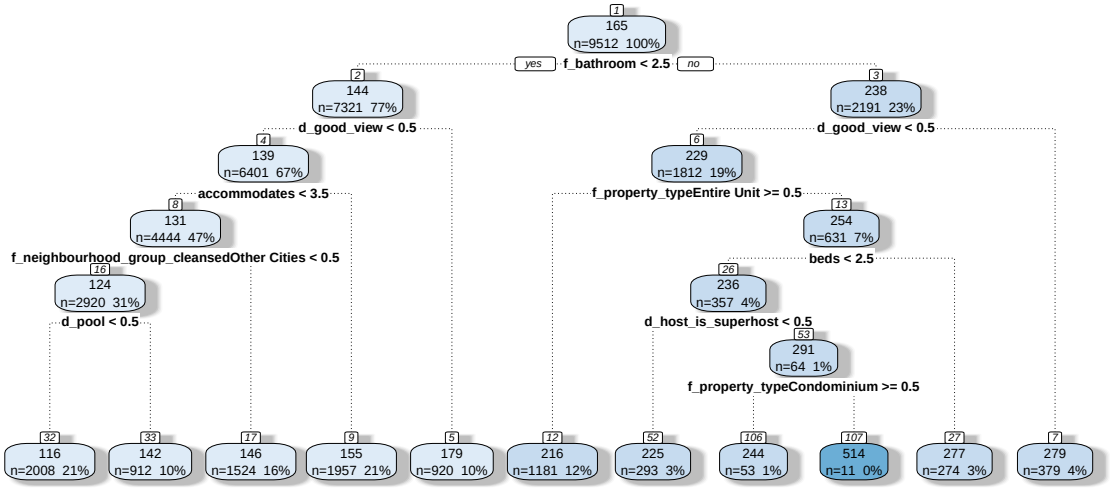
Additionally, other diagnostics showed that the model performs better for cheaper and smaller apartments. This is may be due to the fact that the dataset doesn't include variables that actually capture the features that make apartments higher in value.

Appendix

Graph1: Interaction terms



Graph2: CART


1
165
n=9512 100%
yes
f_bathroom < 2.5
no
2
144
n=7321 77%
3
238
n=2191 23%
d_good_view < 0.5
d_good_view < 0.5
4
139
n=6401 67%
6
229
n=1812 19%
accommodates < 3.5
f_property_typeEntire Unit >= 0.5
8
131
n=4444 47%
13
254
n=631 7%
f_neighbourhood_group_cleansedOther Cities < 0.5
beds < 2.5
16
124
n=2920 31%
26
236
n=357 4%
d_pool < 0.5
d_host_is_superhost < 0.5
53
291
n=64 1%
f_property_typeCondominium >= 0.5
32
116
n=2008 21%
33
142
n=912 10%
17
146
n=1524 16%
9
155
n=1957 21%
5
179
n=920 10%
12
216
n=1181 12%
52
225
n=293 3%
106
244
n=53 1%
107
514
n=11 0%
27
277
n=274 3%
7
279
n=379 4%