Technical Report for Price Prediction Modeling for Airbnb Apartments in Berlin, Germany

Ian Brandenburg (2304791)

GitHub Repository Link

1. Introduction

The objective of the project is to develop predictive modeling for Airbnb apartments in Berlin, Germany that accommodates two-to-six individuals. This project will employ the use of three models: OLS, Random Forest, and CART. The dataset being used is from Inside Airbnb, a site with detailed Airbnb data. The dataset used for this project on Berlin is from September 16th, 2023. The dataset contains 13,134 observations prior to the cleaning process. The data was uploaded to GitHub and called directly from GitHub in the Python code.

2. Data Preparation

The data was prepared and cleaned using Python. The first step that was taken was to get an overview of the data to determine the best steps needed in cleaning the data. From there, the data was cleaned using a function developed specifically for this dataset.

2.1 Cleaning

Prior to cleaning, the categorical variables were visualized to determine their values counts, as well as the total number of missing values across the variables being selected for research. There were a few discoveries that led to decision making in the cleaning process. The first decision made here was that the room and property type variables included categories relating to hotels. These categories were subsequently dropped from the study since this is focusing on apartments. Additionally, there were 67 different property types, which was a bit too broad. So, any property type with under 100 iterations was dropped from the study. Finally, there were missing values identified in the following columns: beds, review scores ratings, and host is a Superhost. All these missing values were converted to 0s.

The cleaning function developed for this project captures all these decisions and handles more complex variables. The amenities column was rather complex to wrangle but was done so by counting the frequencies of amenities that showed up throughout the dataset, and only selecting the top 15 by frequency count. These 15 amenities were then transformed into dummy variables for a more granulated perspective on the association between specific amenities and price.

The next step in the cleaning function was to remove unnecessary columns from the dataset. The dataset starts with 75 columns, many of which are not needed here. So, they were dropped. The price column was cleaned more strictly. Any missing values of 0 values were dropped from the dataset since price is the dependent variable. Furthermore, string replacement was conducted to remove the '$' symbol from the prices.

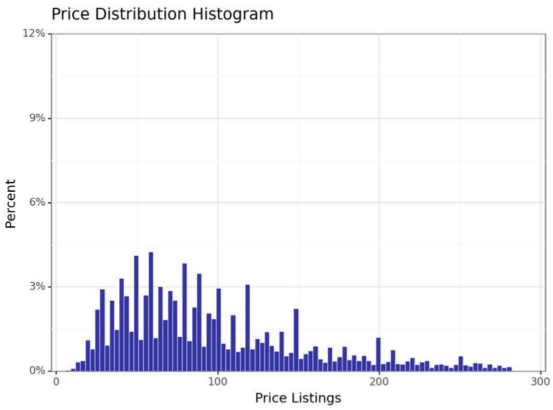
The dataset was then filtered to best fit the research objectives. Accommodates were filtered from two-to-four, while prices were filtered down to the inner-quartile range. This is justified by the few extreme values in price skewing the data heavily. Finally, hotel related categories were removed from the property and room type variables to focus on apartments.

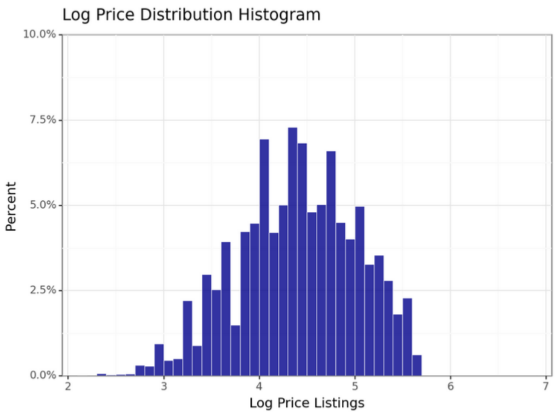
The property and room type variables were then transformed into dummy variables for a more detailed analysis. The property types contained 67 different types and were thus filtered down to only show property types with more than 100 occurrences. From here, the columns were renamed for organization purposes. Continuous variables were prefixed with “n_” and categorical variables with “f_”. Furthermore, the amenities columns were prefixed with “amenities_” and the other dummy variables were prefixed with “d_”.

Finally, missing values were handled by filling all numeric missing values with 0, and categorical missing values with “Missing”. Although most columns did not report missing values, this was for precautionary reasons to ensure that all missing values were managed.

The data was inspected post-cleaning to ensure all variable transformations were completed correctly, and that no mossing values remained. A total of 9,832 observations remained post-cleaning.

In the EDA, frequency of room and property types were reviewed to determine the distribution across these categorical types: “entire rental unit” was the most frequent property tape at 59% of the observations, while “entire home/apt” was the most frequent room type at 71% of the observations. Additionally, a distribution chart for prices was conducted to analyze the normality of price, with the results suggesting the data was relatively skewed. This resulted in the decision to conduct a log transformation of prices.





3. Training/Test Splitting

After the log transformation, the data was split into a training and testing set, with 70% going into the training set. Variables were split into predictor sets for incremental model testing. The following details how the variables were split up.

3.1 Basic Variables

These are foundational predictors that represent core characteristics of listings:

- **n_accommodates**: The number of guests a listing can accommodate. This is a direct indicator of the size and capacity of the property, which can be significantly associated with price.

- **n_beds**: The number of beds available. More beds could indicate a listing's ability to host more guests, hence being associated price.
- **f_property_type**: The type of property (e.g., apartment, house). Different property types have varying market values, potentially being associated with pricing.
- **f_room_type**: The type of room offered (e.g., entire home, private room). Entire homes usually command higher prices than private or shared rooms.
- **d_host_is_superhost**: Whether the host is classified as a "Superhost," which could signal higher quality or more desirable listings, potentially being associated with prices.
- **n_availability_365**: How many days a year the listing is available. Higher availability might indicate less demand, which could impact pricing strategies.
- **n_maximum_nights** and **n_minimum_nights**: Restrictions on bookings can influence a listing's appeal and thus possibly be associated with pricing.

3.2 Review Variables

These variables relate to guest feedback, an important factor in consumers' decision-making:

- **n_number_of_reviews**: A higher number of reviews can indicate popularity or longer presence on the platform, possibly correlating with higher trust and potentially prices.
- **n_review_scores_rating**: The average review score. Higher scores may allow hosts to charge more due to perceived higher quality or satisfaction.

3.3 Amenities Variables

A list that shows the presence or absence of various amenities (e.g., WiFi, kitchen), which are crucial for guest convenience and can significantly influence their willingness to pay.

3.4 Room Booking Types

Variables beginning with d_ likely indicate different booking policies or room types not covered by f_room_type, providing additional granularity on the listing's offering.

3.5 Interaction Terms

- **X1**: Interactions between property/room types and accommodations. These capture how the effect of property size varies by type, reflecting that some property types might be more valuable per guest accommodated.
- **X2**: Interactions involving host status and amenities with other features. These explore more nuanced relationships, such as whether Superhost status increases the positive effect of reviews or certain amenities enhance a listing's appeal based on host quality.
- **X3**: More complex interactions that aim to uncover deeper insights into how combinations of features, like property type with review scores or specific combinations of amenities, potentially being associated with pricing.

3.6 Predictor Sets

- **predictors_1**: A baseline set focusing on essential listing characteristics.
- **predictors_2**: An expanded set that adds reviews and amenities to the basic variables, aiming to capture a broader range of factors potentially being associated with price.

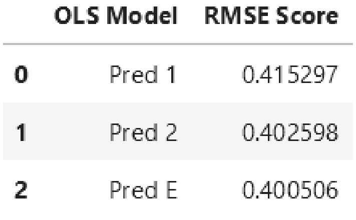
- **predictors_E**: The most comprehensive set, including all basic, review, and amenity variables, plus complex interaction terms, designed to capture the most nuanced associations with pricing.

4. Model Development

Three models were selected to be run for this project: OLS, Random Forest, and CART. Each of these will produce an RSME to determine the level of accuracy for each model. These models were programmed in Python and stacked into functions more for efficient code execution and simplicity.

4.1 OLS Model

A function was developed to the OLS model easily with the three sets of predictors. The function uses the log price, and joins the predictor set as the explanatory variables. This function produces the RSME and coefficients for the models.

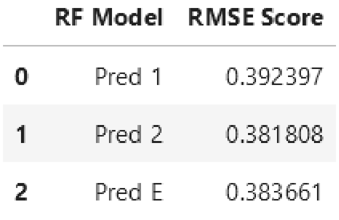


	OLS Model	RMSE Score
0	Pred 1	0.415297
1	Pred 2	0.402598
2	Pred E	0.400506

The second OLS model in this case was chosen as the most ideal, as the RSME is very close to the third OLS model, while the second is not as complicated. The OLS model does not require complex code for tabulation, and thus was completed quickly.

4.2 Random Forest

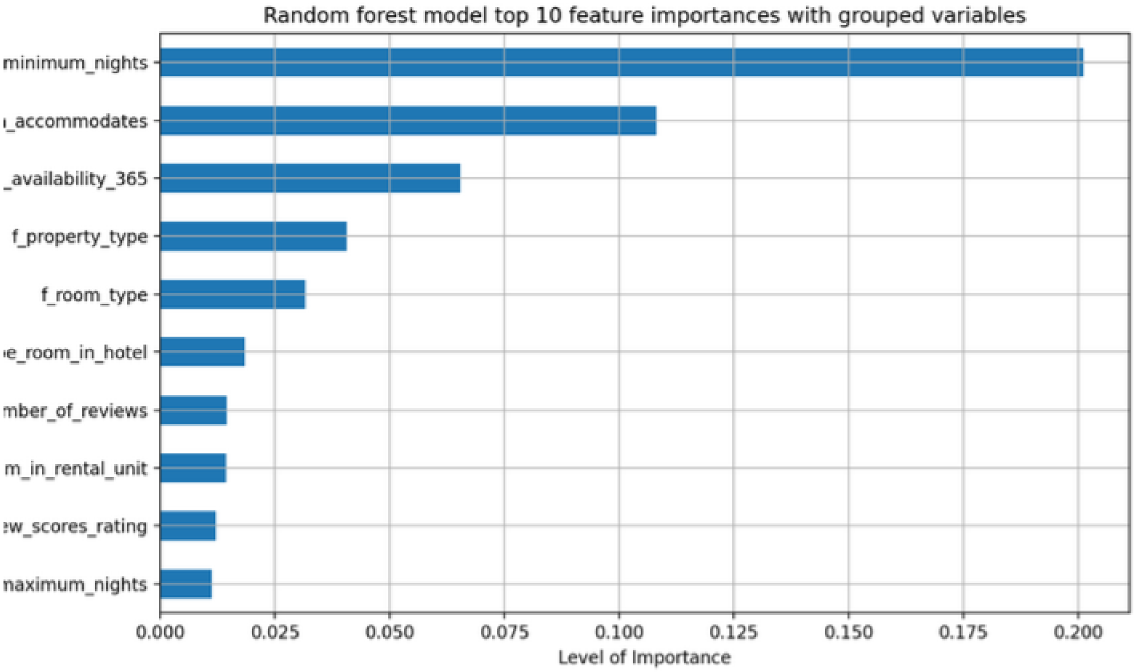
A function was created for the Random Forest model. This function produces several outputs. Initially, the function reviews the design matrices, and displays the shapes of X and y. From here, the function initializes the Random Forest regressor using the random state of 20240211. The 5-fold cross-validation results are displayed, which identified the best RSME. Three Random Forest models were run with 5-fold cross-validation to determine which set of predictors was the most optimal.



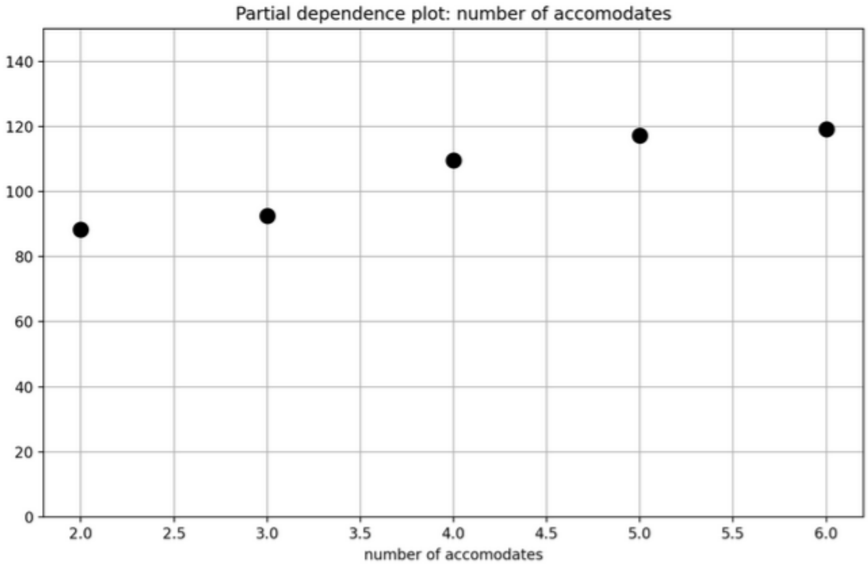
	RF Model	RMSE Score
0	Pred 1	0.392397
1	Pred 2	0.381808
2	Pred E	0.383661

The RSME for predictor set two, which was the selected Random Forest model to compare to the other models.

An additional function was developed for the model diagnostics. The purpose of this function was to streamline the process in determining the most important variables in the modeling, and to display diagnostics for a fitted Random Forest model. The parameters of this model include the fitted Random Forest model from the GridSearchCV, as well as the training dataset used to fit the model and the list of predictors used in the model. The outputs of this function are the feature importances, and the Random Forest model highest feature importances plot.

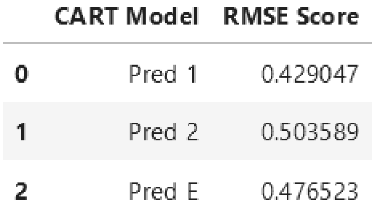


Additionally, a partial dependency plot was run showing the number of accommodates on the x-axis with the average log price on the y-axis. This demonstrated a positive association between the number of people accommodated and the price of the accommodation.



4.3 CART Model

A function was developed to run the CART model efficiently across the three sets of predictor variables. The function fits a CART model to the training set, and optimizes ccp_alpha using cross-validation, calculating the RMSE, and measures the execution time. The parameters of the function include the list of predictor variables and the training data set. The function returns the RSME of the CART model using the best ccp_alpha, as well as the best ccp_alpha value, and the execution time. All three predictor sets were run to identify which set of variables performed the best.

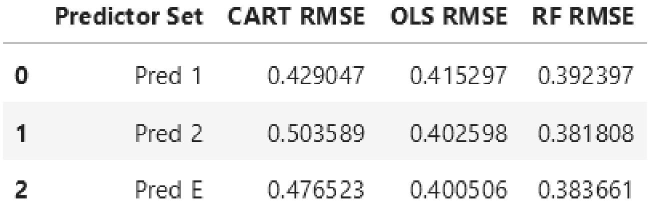


	CART Model	RMSE Score
0	Pred 1	0.429047
1	Pred 2	0.503589
2	Pred E	0.476523

The first set of variables that includes the basic variables performed the best, which is most likely due to the simplicity of the first prediction set. This contradicts the findings in the OLS and Random Forest models, which suggest that the second set of predictor variables is the most effective for a price prediction model.

5. Model Comparison and Conclusion

Of the three models executed, Random Forest performed the best. A code was built to develop a table comparing all three models, which can be seen below.



	Predictor Set	CART RMSE	OLS RMSE	RF RMSE
0	Pred 1	0.429047	0.415297	0.392397
1	Pred 2	0.503589	0.402598	0.381808
2	Pred E	0.476523	0.400506	0.383661

The best predictor set in the Random Forest model was the second set of variables, which included the basic variables, review variables, amenities, and room booking types. The feature importance suggests that the minimum number of nights bookable, how many people the listing can accommodate, and the availability of the accommodation are most important when predicting the price of Airbnb apartments in Berlin.