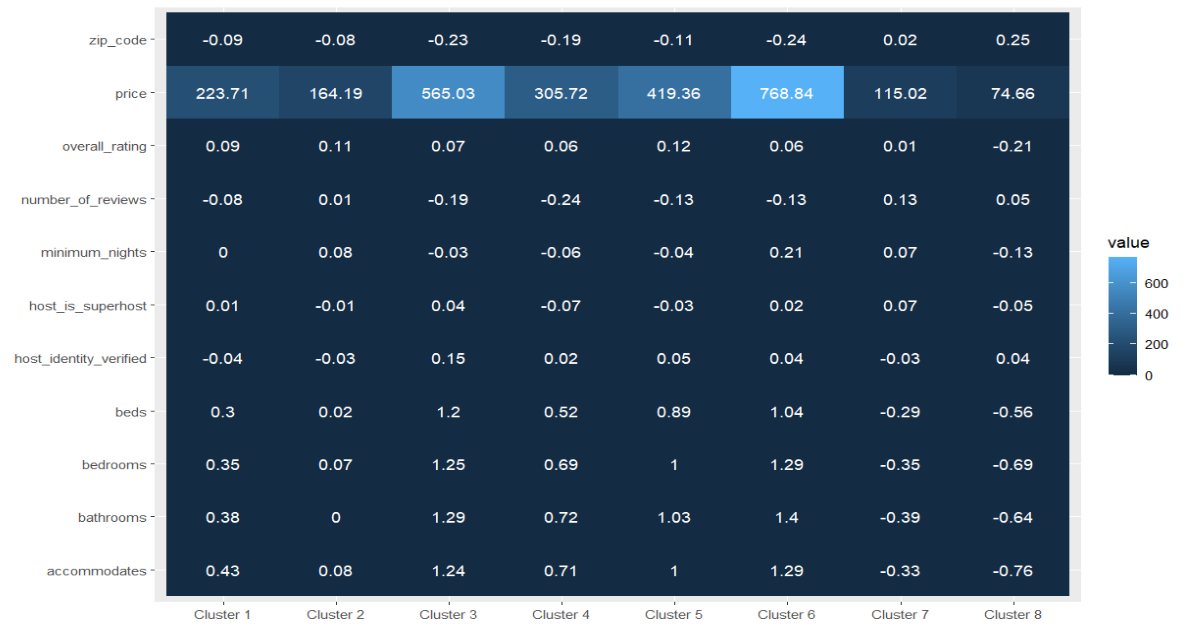
Sage Gray  
Dr. Liu  
BAS 474  
05/07/2025

Airbnb Price Prediction

# Clustering

The best value I found was clustering with 8 centroids, and I created the following centroids table:

  
Fig 1. Centroids table for k = 8

The two most valuable clusters were clusters 6 and 8. Cluster 6 has the highest price average of $768.84 and had above average bedrooms, bathrooms, and number of people it accommodates. Cluster 8 had the lowest price average of $74.66 and had a below average rating, number of beds, bedrooms, bathrooms, and number of people they accommodate.

# Predictions

I tried several models for predicting, including Linear, Polynomial and Radial Kernel Support Vector Machines, Glmnet, Logistic Regression, and Ensemble before landing on Random Forest as my final prediction model. The model was off by about $101.74 on average, give or take $2.34. Using the 1 SD rule, Ensemble was the only model about as accurate as Random Forest. I don’t think I would recommend moving forward with this model for the price suggestion feature because a $100 plus or minus $2 error is almost half of the mean pricing on these locations. I don’t think it’s accurate enough to be useful. The features that seem the most valuable are the number of people the space accommodates, and the number of bedrooms and bathrooms, because the top two clusters in terms of price have above average values and the lowest two price clusters had below average values in all three of those features. I did not find any standout host qualities that affected price, as they all seemed to be scaled and clustered out to be within .1 of each other. Going forward, some potential aspects of the location that could be used to improve predictions could be “Distance from Local Landmark” or “Walkable Area” or “Bikeable Area”. These could help to show how the price would change for people who wish to see the surrounding area and what they would be able to pay.

# Technical Report

## Clustering

To perform the clustering I needed to remove some of the identification data that wasn’t adding anything to my analysis. These columns were mostly unique identifiers, such as “image url” or “host name”, or columns that were mostly unique, such as “host total listings” or “host response rate”. Then I normalized all the values and turned non-numeric values into leveled factors. I searched for the smallest within sum of squares values with cluster centers between 2 and 20. I created the following elbow plot:

A line graph with numbers

AI-generated content may be incorrect.

As can be seen, the Sum of Squares drops off dramatically between one and 2, then slowly levels out. I decided to try k values of 7, 8, and 9 before landing on 8 due to its different pricing values and distinguishing features in the chosen predictors. I used KMeans clustering because I thought the price ranges could be dropped based on the average values of the commodities each home could have had.

## Price Prediction

For predictions, several columns had missing values, and I needed to decide how to deal with them. For Bathrooms, Bedrooms, and Beds, I decided to replace the values with the median of the columns. Most homes only have 1 or 2 bedrooms and bathrooms, and it seemed likely that the missing values could have fit with the average house. The “overall\_rating” column had missing values because the listings had 0 reviews. Replacing the rating with 0 could cause these places to have an unfairly low score, which could skew the data lower. I decided it would be best to replace those missing values with the mean of overall\_rating. The missing values in “host\_acceptance\_rate” didn’t have an obvious reason, so I replaced them with the median.

Predicting numeric values such as price requires linear regression, so I started with a vanilla model and worked through linear regression, Random Forest, an Ensemble model, and three support vector machine models: Linear, Polynomial, and Radial. I tuned the hyperparameters for each model with grid expansion and landed on the top three models of Random Forest, Ensembled Random Forest, and Polynomial Support Vector Machine. Random Forest had an RMSE of 104.0482, Ensembled Random Forest with an RMSE of 104.057, and Poly SVM had an RMSE of 114.8272.

A Random Forest model creates a series of deep decision trees and averages them together to create the most accurate prediction. It takes different sets of predictors from the same training data to aggregate the most important aspects. The Random Forest model I landed on had the tuned hyperparameters of mtry = 6, meaning it pulled 6 predictors for each tree per iteration.

# Prediction Leaderboard

