Regression Mini Project
Sam Lee
STAT 486

I first looked at the data systematically by sorting out each column as a categorical or numerical column. I then cleaned some of the numerical columns through some basic cleaning techniques. If I encountered a column with textual information I thought about how to parse out information from that non-numerical column using feature engineering. One approach is how “positive” the neighborhood sounded. I took random samples of the neighborhood overview descriptions and found similar “positive” words that might sway consumer choice. I gave each entry a “positivity” score based on this analysis. Ultimately, the positivity score didn’t give much information about the price, so this was eventually excluded.

I am an economics major, so I also thought about the endogeneity of price: Since price is endogenous and dependent upon the equilibrium of demand and supply, we need some kind of instrumental variable to accurately predict how price changes when an exogenous shock happens to demand or supply. Unfortunately, this data set mostly contains factors related to demand-side (behavioral) economics, so it was challenging to take the endogeneity “out” of price. One approach I took was using dummies for location (fixed effects) and dummies for room type and apartment type. I also created two kinds of “recency scores” used as a potential IV for the demand for an apartment. One recency score was the difference (in days) between the most recent review and the last review left. I also created one for the host, using the host_since date as a baseline for comparison. I am assuming that these recency scores only affect price by affecting consumer behavior.

I decided to take a KNN-regression approach at first. I separated my categorical and continuous variables into two separate pipelines to be cleaned. Since I used KNN-regression I used a KNN-imputer for the numeric data. Something interesting I did was that in addition to including polynomial terms for the numeric data, I also included polynomial interaction terms between the numerical data and the categorical (binary/dummies) in the transformed pipeline. Due to the extensiveness of the model, I tried to use a randomized search cross-validation method to compare the effectiveness of different hyperparameters, but the model I constructed initially had too many parameters and observations so even with RSCV, it was taking hours to run, and then eventually forced a memory error. I looked at the correlation matrix between the X matrix and eliminated a good deal of variables that were uncorrelated with the price. I ultimately tried several KNN models changing the hyperparameters. I settled on n-neighbors = 10.

The best feature I created within the dataset was “neighborhood price” in an effort to resolve the spatial correlation between the prices. I calculated the haversine distance between every apartment and the corresponding vector of distances from the training data set. Neighborhood price was calculated as the average price of all properties that fell within an epsilon’s distance (I used epsilon = 1km as a hyperparameter) away from the individual property. This was the factor that had the greatest influence on price.

Due to the sheer number of parameters I was using to predict the price, I also decided to use lasso regression and ridge regression to test against the KNN-regression model so I could potentially eliminate colinearity and eliminate unnecessary predictors. However, these models yielded higher MAE than my KNN models.

Because of the number of parameters I was using was rather large (I was using a polynomial transformer of degree 3 on both the continuous and discrete variables), I had to fit my model using batch fitting, randomly dividing up the data into batches of 1000 points and fitting the data separately to avoid memory errors.

My out-of-sample MAE was 141.3