Predicting an Airbnb's Price:

A Look at Airbnb Data from the City of Chicago and the Significant Features Associated with Price

STAT:4540 – Fall 2019

Matthew Naples, Joel Northrup,
Rhedt Roelandt, Calvin Skalla, Cory Skeers

© Remo Daut

Contents

1 Introduction

The goal of this project is to use statistical learning methods in order to predict the nightly price (named price) of an Airbnb. Also, it is to find the relationships between price (or log(price)) and predictors. For the accomplishment of the former goal, we were willing to sacrifice interpretability for predictability. We used a general additive model to accomplish this goal. As for accomplishing the goal of understanding relationships, we decided to make a separate set of linear regression models. We realize that this may lack predictive performance due to its more stringent assumptions, but we were willing to sacrifice some of that in order to understand the relationship between price and its predictors. Of course, we would not rely on linear regression to understand the relationship between price and our predictors if the model utility in prediction suffered too greatly. The predictors for each model will be discussed in depth as we walk through the details. Different models yielded different sets of variables. So, by providing multiple data visualizations to easily understand the data, we conclude that there is in fact influential responses to predict the price of an Airbnb.

Data Cleaning and Feature Engineering

There were many variables as candidates…

```
FALSE  [1] "host_response_time"
FALSE  [2] "host_response_rate"
FALSE  [3] "host_is_superhost"
FALSE  [4] "host_has_profile_pic"
FALSE  [5] "host_identity_verified"
FALSE  [6] "latitude"
FALSE  [7] "longitude"
FALSE  [8] "property_type"
FALSE  [9] "accommodates"
FALSE [10] "bathrooms"
FALSE [11] "bedrooms"
FALSE [12] "beds"
FALSE [13] "security_deposit"
FALSE [14] "cleaning_fee"
FALSE [15] "guests_included"
FALSE [16] "extra_people"
FALSE [17] "minimum_nights_avg_ntm"
FALSE [18] "maximum_nights_avg_ntm"
FALSE [19] "number_of_reviews"
FALSE [20] "review_scores_rating"
FALSE [21] "review_scores_accuracy"
FALSE [22]
"review_scores_cleanliness"
```

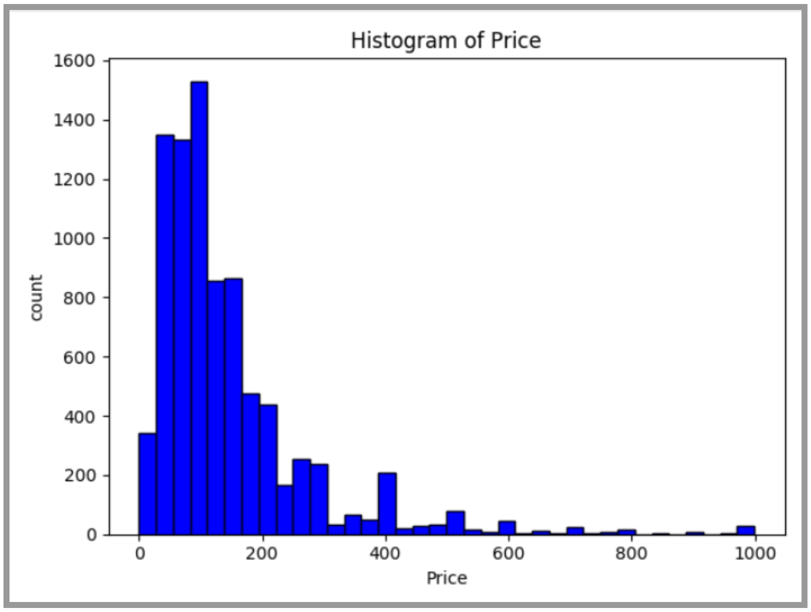
```
FALSE [23] "review_scores_checkin"
FALSE [24]
"review_scores_communication"
FALSE [25] "review_scores_location"
FALSE [26] "review_scores_value"
FALSE [27] "requires_license"
FALSE [28] "instant_bookable"
FALSE [29] "cancellation_policy"
FALSE [30]
"require_guest_profile_picture"
FALSE [31]
"require_guest_phone_verification"
FALSE [32]
"calculated_host_listings_count"
FALSE [33]
"calculated_host_listings_count_ent
ire_homes"
FALSE [34]
"calculated_host_listings_count_pri
vate_rooms"
FALSE [35]
"calculated_host_listings_count_sha
red_rooms"
```

```
FALSE [36] "reviews_per_month"
FALSE [37] "crib"
FALSE [38] "balcony"
FALSE [39] "dryer"
FALSE [40] "kitchen"
FALSE [41] "pool"
FALSE [42] "air conditioning"
FALSE [43] "oven"
FALSE [44] "washer"
FALSE [45] "essentials"
```

```
FALSE [46] "wifi"
FALSE [47] "hot tub"
FALSE [48] "gym"
FALSE [49] "toilet"
FALSE [50] "capPC1"
FALSE [51] "reviewScoresPC1"
FALSE [52] "reviewsPC1"
FALSE [53] "listingCountPC1"
FALSE [54] "listingCountPC2"
```

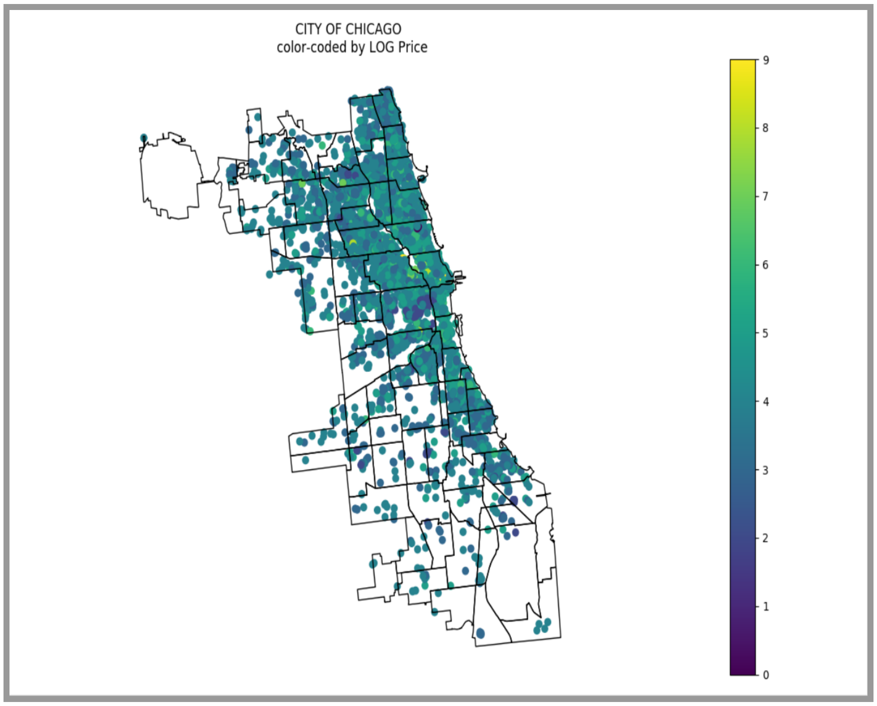
Much of the data available required minor cleanup – stripping out dollar signs from numerical data, converting simple data such as profile picture to a binary “hasProfilePic,” etc. Null values were handled on a case-by-case basis. For some features, such as “cleaning fee,” a null value has been assumed to be a zero and replaced by such. For others, such as “review scores rating,” a null value has been assumed to be null for lack of information, and has been replaced by the mean value of that feature over all collected data. For instance, the mean for one dataset’s “review scores rating” is 95.074. For observations where this rating is missing, the null value has been replaced by 95.074, assuming a neutral response. Treating this as a baseline, then, our model assumes that a rating above 95.074 (for this dataset) to be significant in one direction, while below this to be significant in the other direction. This results in a feature-set which has been standardized but not normalized by scale, so that such normalization can be handled on a per-model basis.

Several data features, such as host response rate, are categorical in nature, with Airbnb providing a dictionary of possible outcomes. These are retained as categorical features, but analysis and experimentation may lead to a quantitative approach where such data has a clear ordinal nature.



Lastly, the “amenities” data has been treated in two separate manners, with the two approaches yielding similar results as of yet. Amenities are also provided as a dictionary of possible inclusions, the final result containing any combination of zero to all amenities.

In one method, the data cleaning process is used to create one binary feature for every amenity. The other method simply condenses all amenities into a single quantitative variable which represents the total number of amenities listed. As both these approaches have thus far yielded similar results, the latter is preferred for its ability to retain accuracy while helping reduce the total number of features the model requires. Additionally, it allows for more potential scaling of the model approach, as regions which offer different amenities (or which would require language translation) may be directly compared. This “number of amenities” approach is used in the final GAM methodology.



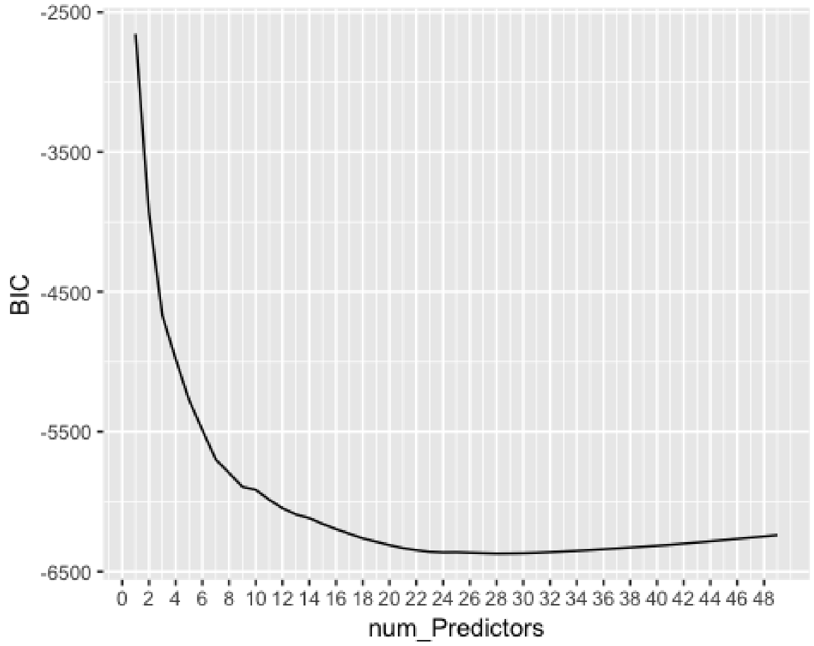
We observed that plotting the histogram of price displays a distribution that is highly right-skewed. To rectify this, we performed a log transformation on the data. By doing this, we were able to spread the smaller values of price while simultaneously compressing the larger values, effectively removing the skew and approaching normality. This made it more likely to return usable outputs once we ran on our models on the transformed data.

2 Methods

Linear Regression Models

After data cleaning and feature engineering, the first series of modeling efforts included linear regression, polynomial regression, and linear regression using lasso and ridge shrinkage methods. For linear and polynomial regressions basic 80-20 set validation approaches were used, while lasso and ridge linear regressions used 10-fold cross validation for fitting. Lasso, ridge, and polynomial regression methods didn’t help much in predicting price. Classic linear regression models appeared to work just as well in predicting as the models with regularization and polynomial terms. Also, classical linear regression offers the benefit of simpler interpretation of parameters.

We split the data into an 80/20 train/test split, and log transformed price. To select a proper model,we used a backward selection (BIC) using the regsubsets() method. Here are the following BIC's:



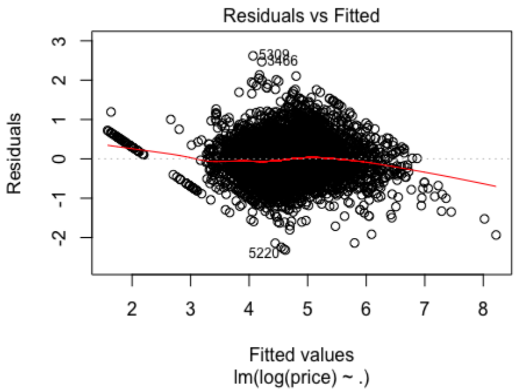
The model with the minimum BIC was the 28-covariate model. The summary for this model can be found in the appendix.

Model I

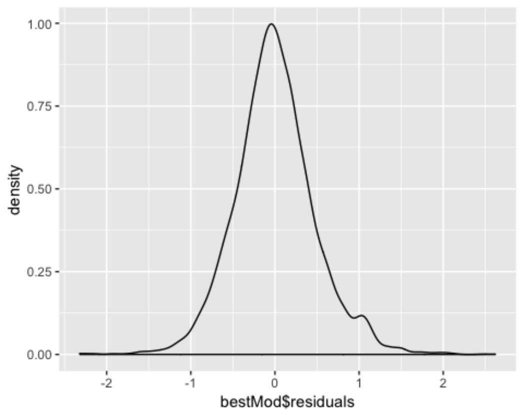
$$Log(Price) = \sum_{i=1}^{28} \beta_i \hat{x}_i + \varepsilon$$

This linear regression equation demonstrates the

prediction of price based off of the model with 28 variables



The model does have a fairly high R-squared. However, some assumptions do appear to be imperfect. For example, the tails of the residuals are heavy. Also, while the whole of the data suggests homoscedasticity holds, there appears to be a small pocket of correlated values - those which yield a small predicted log(price).

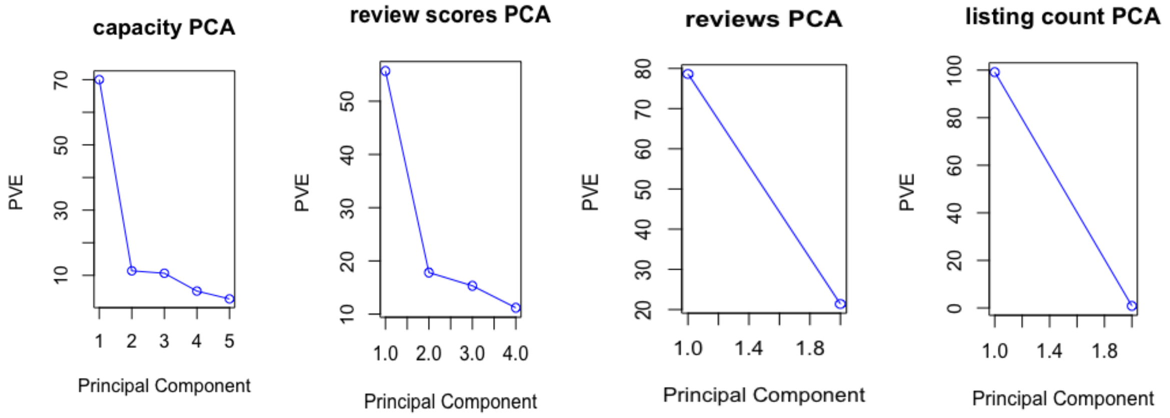


The model may be okay to predict outcomes, but multicollinearity exists, meaning the slope estimates aren't quite reliable. We would like to understand some relationship between the response variable and covariates as opposed to just predicting the response using the covariates. To remedy multicollinearity, we opted to use PCA.

To carry out the PCA, we first analyzed the correlation matrix and noted all variables that had a correlation>.3. The following 4 groups of variables seemed to be correlated with each other:

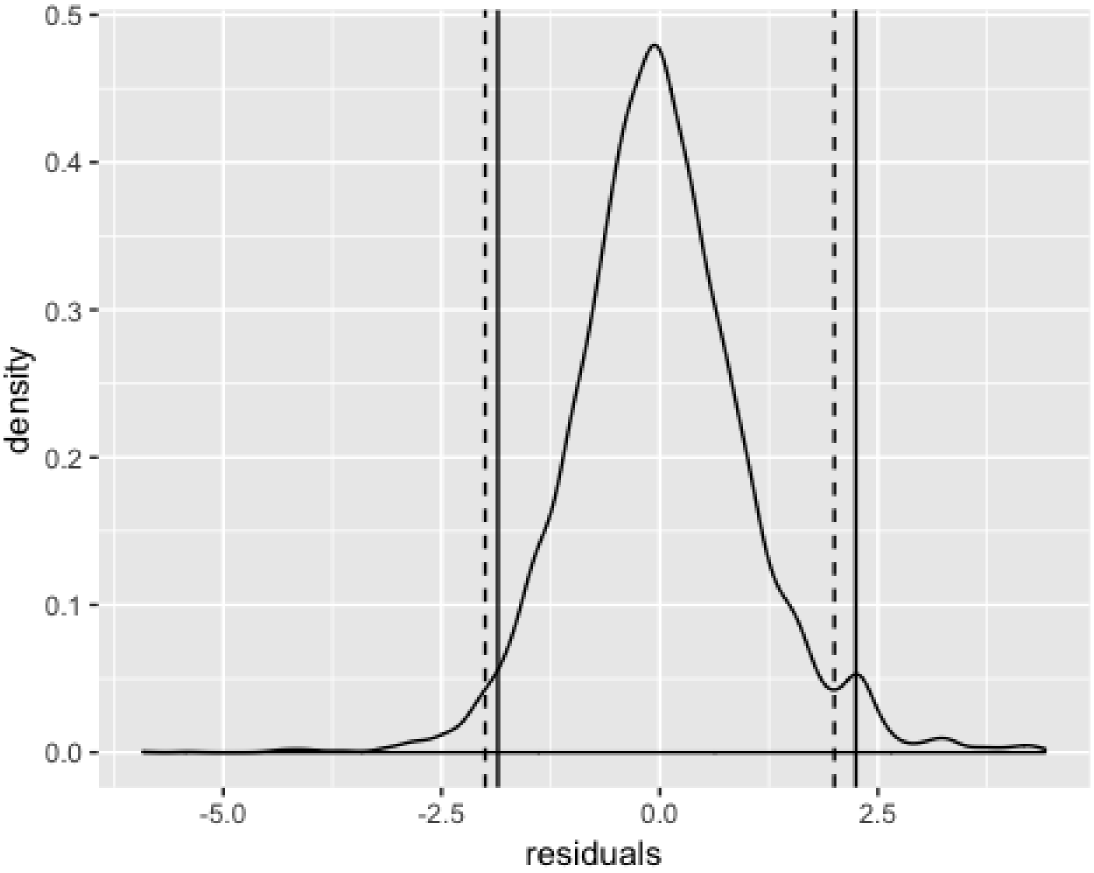
```
FALSE [1] "accommodates" "bathrooms"    "bedrooms"     "beds"
FALSE [5] "cleaning_fee"

FALSE [1] "review_scores_cleanliness" "review_scores_checkin"
FALSE [3] "review_scores_location"    "review_scores_value"

FALSE [1] "reviews_per_month" "number_of_reviews"

FALSE [1] "calculated_host_listings_count"
FALSE [2] "calculated_host_listings_count_entire_homes"
```

Thus, a PCA was performed on each of the 4 groups. Below are the plots showing Percentage of Variance Explained (PVE) for each analysis:



Each analysis yields a first principal component that captures 56 and 99 percent of total variability in its respective analysis. So we used these first principal components from all of the PCAs (except for the final PCA, where we pulled the first 2 principal components as 56% doesn't capture enough information in the data), as well as the variables left over from the previous regression model and performed best subset selection (since p has been reduced and is computationally feasible).

PCA did fix multicollinearity. The intercepts should be a bit more reliable then. However, I say that with caution. The distribution of residuals still has slightly heavier tails than a normal distribution. Below is the summary for the model with the lowest BIC, and below that is a distribution of the standardized residuals. The dotted lines represent ±2 sigma. The true lines represent the true .25th quantile and 97.5th quantile. A slightly heavier tail means that we may be underestimating the true sigma term. If that's the case, we may therefore be slightly underestimating the standard error of each slope estimate. The output summary for this model can be found in the appendix.



Model II

$$Log(Price) = \sum_{i=1}^{10} \beta_i \hat{x}_i + \varepsilon$$

This linear regression equation demonstrates the

prediction of price based off of the model, with 10 variables

Next, we validate the models, testing each of them on the test set. Here's the test error for the small model:

```
## [1] 0.3416791
```

Translated into units of price, the root MSE equals $130.06.

For the large model:

```
mean((log(heldOutData$price) - predict(bestMod, newdata=heldOutData)) ^ 2)

## [1] 0.2827925
```

Again, translated into units of price, the root MSE equals $121.34.

As is quite apparent, both models perform sufficiently well.

Generalized Additive Model

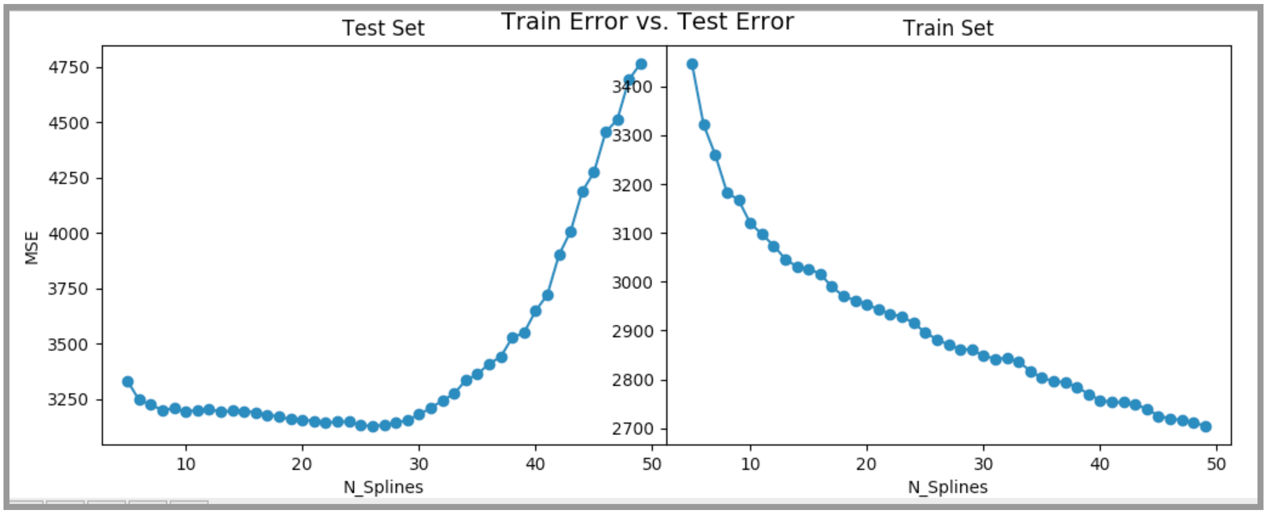
Due to the sparsity of available data on the low- and high-price extremes, and in order to introduce further flexibility into the model, a generalized additive model (GAM) was implemented using univariate splines across 44 features. Repeated set validation with an 80-20 split was used in order to determine the optimal number of splines (resulting in the optimal flexibility), which occurred with approximately 21 splines.

$$Log(Price)_i = \beta_0 + \sum_{j=1}^{44} f_j(x_{ij}) + \varepsilon_i$$

This equation explains the generalized additive model showing

that each variable receives their own function

Fitting a dataset with high-price outliers beyond the 95th percentile removed, this 21 univariate-spline GAM approach is able to approximate between 55 and 62% of the variance in AirBnB rental pricing.



Train and test MSE are plotted for a range of splines. After an initial dropoff of MSE in the test set, as expected as bias decreases, it levels off and, after around 21 splines, increases sharply due to high flexibility leading to overfitting of the training data. As expected, in the training set the MSE continues to decrease well past this point.

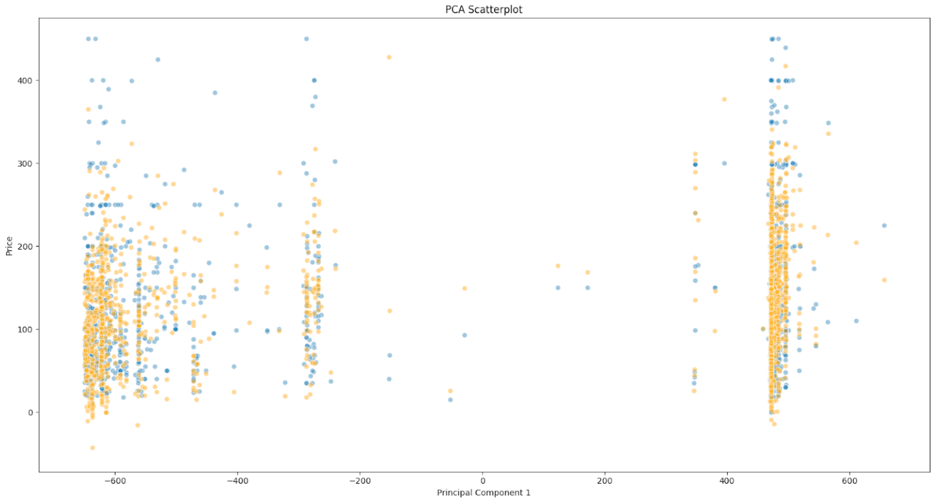
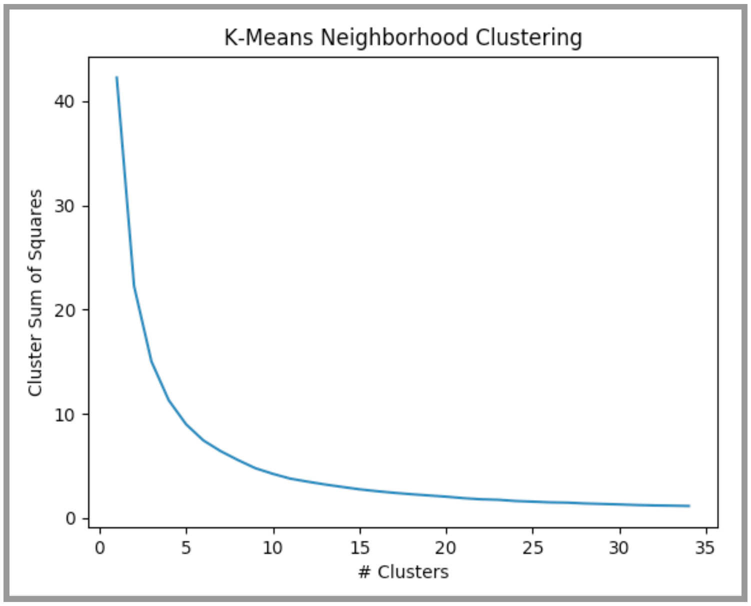


Figure: Using PCA to visualize price predictions (orange) against true prices (blue) based on a principal component which accounts for approximately 72% of the variance in the 44 features used with the Generalized Additive Model. Where many data points are available, the model performs well, creating dense clusters around the true prices. It tends to miss the higher-priced anomalies in these clusters, while overestimating price in less represented feature-sets.

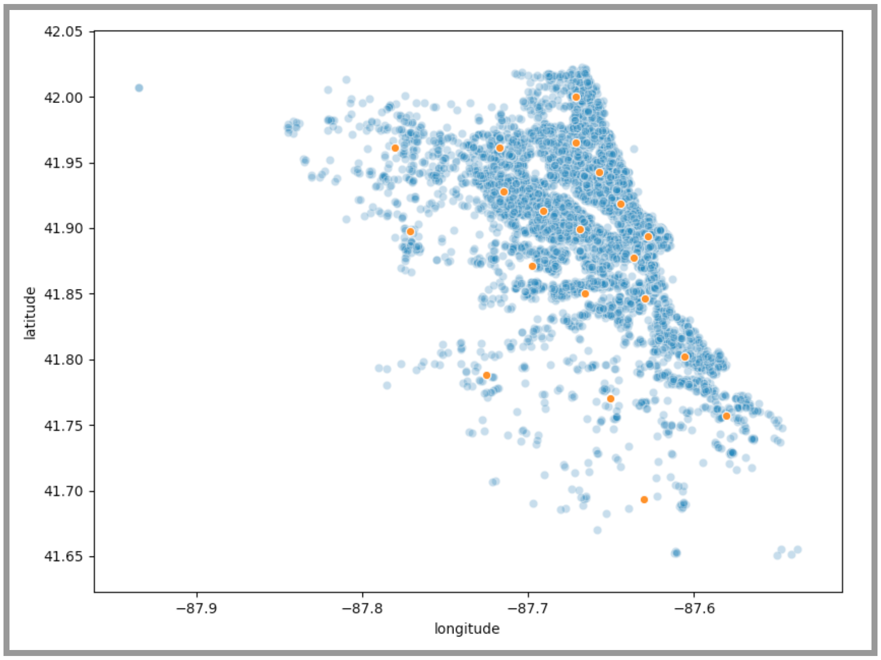
Handling of Locational Data

Our intuition is that the AirBnB rental's location plays a role in determining its optimal price, however working off of defined neighborhoods presents possible bias, as well as limits the modeling approach from future scaling.

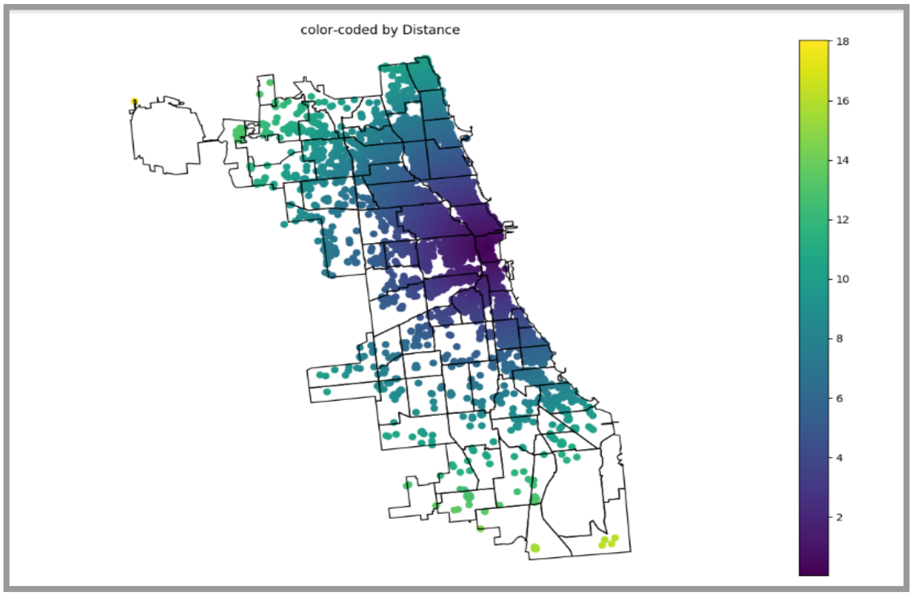


Rental longitude and latitude data are available, and when treated as separate quantitative variables performed well for individual cities. It again introduces a problem of scaling of the modeling approach. When applied to smaller regions, the lack of variability in lat/long reduces their effectiveness as price predictors, while when applied to larger regions, their dependence becomes a greater issue.

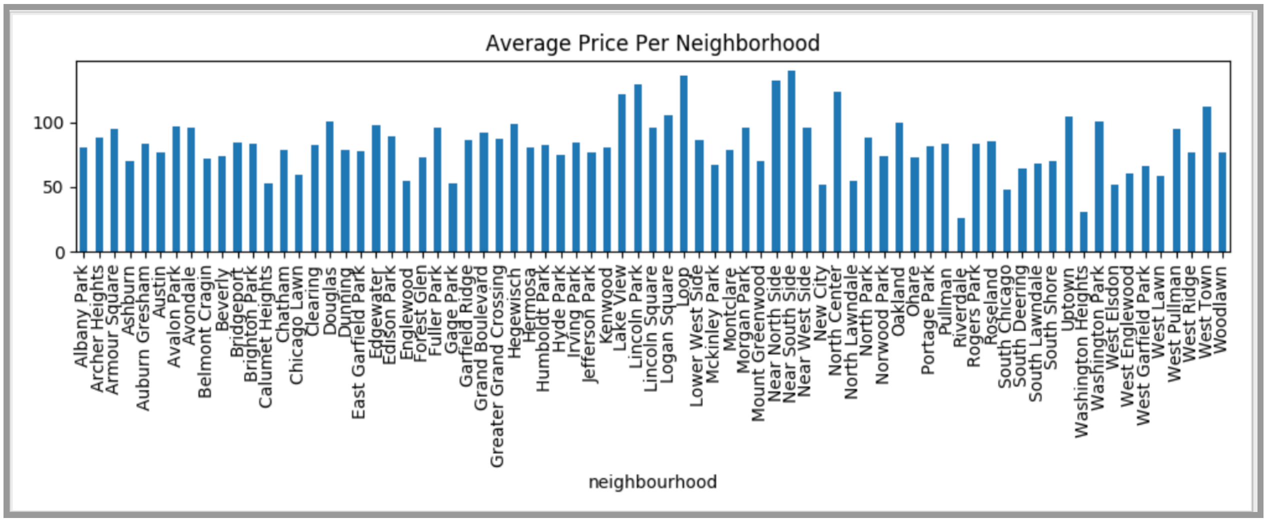
In order to lessen this impact, we have devised a k-means approach to transform the quantitative lat/long variables into a single, k-level categorical variable. In testing, a K of 8 appears adequate to match the performance of using lat/long directly for a large city-sized region, while a K of 20 performed better than using lat/long. (Figure Top: Cluster sum of squares, where K = 8 occurs near the "elbow" of the bend and performs similarly to unmodified long/lat, though further performance is achieved up to around K=20.



Bottom: 20 cluster centroids depicting neighborhoods (orange) overlaid upon all Chicago AirBnB rentals in the dataset.)



This same approach is likely to scale well both to a neighborhood level (single Chicago suburb for instance, given adequate observations) or upward to state, country, or global models. The primary limitation is that determination of the k-means neighborhoods requires access to the full dataset prior to model fitting, increasing opportunity for introducing bias as well as further limiting the transferability of a model fitted in one location to another (Chicago to New York City, for instance) without additional fitting.



An additional approach we performed was to use the given latitude and longitudinal data to find the Euclidean distance to the center of Chicago for each Airbnb . Our results surprised us in that there was very little correlation between 'distance to center' and price. As is apparent in the graphs below, prices tend to stay constant depending on the 'distance to center' when grouped by neighborhood.While the distance from the center of Chicago was not overly significant in predicting the price for these Airbnbs, we would expect the distance from the center to play a greater role in predicting prices in cities such as Berlin, Germany & Paris, France.

3 Conclusion

In conclusion, our analysis revealed some surprising results. Despite the fat-tailed distribution of Price in our Chicago Airbnb dataset, with the various methods employed, we are confident in the ability of our model to predict the price of an Airbnb. By first taking the log of the prices, we learned that there are, in fact, significant features that exist that greatly aid in the prediction of an airbnb's nightly price. We were also able to reveal the many relationships that are present between the price of an Airbnb and its predictors. In accomplishing our goal, we made a measure of sacrifice of the interpretability of our model for predictability. This was accomplished using a general additive model. We then decided to make a separate set of linear regression models in order to accomplish the goal of better understanding relationships. We understood the several drawbacks to this approach include a lacking predictive performance, but we were willing to sacrifice some of that in order to understand the relationship between price and its predictors. Of course, we would not rely on linear regression to understand the relationship between price and our predictors if the model utility in prediction suffered too greatly. Through statistically analyzing airbnb data, we found certain features that were influential in predicting the price.

4 Appendix

Python Code

The code for this project can be found at the following two locations:

https://github.com/coryskeers/airbnb_proj
https://github.com/lawntek/Statistical_Learning

References

Cox, Murray. (2018). *Chicago, Illinois, United States - Get the Data*. Chicago : Inside Airbnb

James, G., Witten, D., Trevor, H., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York : Springer

R Model Summary Outputs

The R Model Summary outputs from the Methods section can be found here:

```
FALSE 
FALSE Call:
FALSE lm(formula = log(price) ~ ., data = newD[c("price", bestNames)] %>% 
FALSE     filter(price > 0))
FALSE 
FALSE Residuals:
FALSE     Min      1Q  Median      3Q     Max 
FALSE -2.3168 -0.2901 -0.0184  0.2685  2.6179 
FALSE 
FALSE Coefficients:
FALSE                                                Estimate Std. Error t 
value
FALSE (Intercept)                                   2.448e+02  1.333e+01  
18.359
FALSE latitude                                      1.516e+00  1.261e-01  
12.029
FALSE longitude                                     3.476e+00  1.750e-01  
19.867
FALSE accommodates                                  1.014e-01  4.416e-03  
22.951
FALSE bathrooms                                     6.545e-02  1.080e-02   
6.060
FALSE bedrooms                                      4.843e-02  9.500e-03   
5.098
FALSE beds                                         -4.496e-02  6.029e-03  
-7.457
FALSE cleaning_fee                                  1.640e-03  1.523e-04  
10.767
FALSE extra_people                                  2.049e-03  2.306e-04   
8.885
FALSE minimum_nights_avg_ntm                       -6.369e-04  1.856e-04  
-3.432
FALSE maximum_nights_avg_ntm                        6.982e-05  1.107e-05   
6.310
FALSE number_of_reviews                            -7.648e-04  1.229e-04  
-6.221
FALSE review_scores_rating                          6.173e-03  1.749e-03   
3.529
FALSE review_scores_cleanliness                     7.200e-02  1.103e-02   
6.530
FALSE review_scores_checkin                        -5.443e-02  1.514e-02  
-3.595
FALSE review_scores_communication                  -6.400e-02  1.643e-02  
-3.894
FALSE review_scores_location                        1.242e-01  1.136e-02  
10.936
FALSE review_scores_value                          -9.201e-02  1.242e-02  
```

```
-7.408
FALSE instant_bookableTrue                          5.374e-02  1.202e-02
4.470
FALSE calculated_host_listings_count               -1.335e-02  2.344e-03
-5.697
FALSE calculated_host_listings_count_entire_homes   1.597e-02  2.466e-03
6.473
FALSE calculated_host_listings_count_private_rooms -3.430e-02  2.618e-03
-13.101
FALSE calculated_host_listings_count_shared_rooms  -5.258e-02  4.990e-03
-10.539
FALSE reviews_per_month                            -4.475e-02  3.489e-03
-12.826
FALSE cribTrue                                      7.355e-02  1.971e-02
3.732
FALSE balconyTrue                                  -4.952e-02  1.462e-02
-3.388
FALSE dryerTrue                                     1.180e-01  1.750e-02
6.745
FALSE kitchenTrue                                   2.464e-01  3.258e-02
7.564
FALSE washerTrue                                    8.763e-02  1.385e-02
6.329
FALSE                                              Pr(>|t|)
FALSE (Intercept)                                   < 2e-16 ***
FALSE latitude                                      < 2e-16 ***
FALSE longitude                                     < 2e-16 ***
FALSE accommodates                                  < 2e-16 ***
FALSE bathrooms                                    1.43e-09 ***
FALSE bedrooms                                     3.52e-07 ***
FALSE beds                                         9.93e-14 ***
FALSE cleaning_fee                                  < 2e-16 ***
FALSE extra_people                                  < 2e-16 ***
FALSE minimum_nights_avg_ntm                       0.000603 ***
FALSE maximum_nights_avg_ntm                       2.97e-10 ***
FALSE number_of_reviews                            5.22e-10 ***
FALSE review_scores_rating                         0.000420 ***
FALSE review_scores_cleanliness                    7.03e-11 ***
FALSE review_scores_checkin                        0.000327 ***
FALSE review_scores_communication                  9.95e-05 ***
FALSE review_scores_location                        < 2e-16 ***
FALSE review_scores_value                          1.43e-13 ***
FALSE instant_bookableTrue                         7.95e-06 ***
FALSE calculated_host_listings_count               1.27e-08 ***
FALSE calculated_host_listings_count_entire_homes  1.02e-10 ***
FALSE calculated_host_listings_count_private_rooms  < 2e-16 ***
FALSE calculated_host_listings_count_shared_rooms   < 2e-16 ***
FALSE reviews_per_month                             < 2e-16 ***
FALSE cribTrue                                     0.000192 ***
FALSE balconyTrue                                  0.000709 ***
FALSE dryerTrue                                    1.65e-11 ***
FALSE kitchenTrue                                  4.41e-14 ***
FALSE washerTrue                                   2.63e-10 ***
FALSE ---
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
FALSE 
FALSE Residual standard error: 0.4829 on 6969 degrees of freedom
FALSE Multiple R-squared:  0.6121,  Adjusted R-squared:  0.6106 
FALSE F-statistic: 392.8 on 28 and 6969 DF,  p-value: < 2.2e-16
```

```
FALSE 
FALSE Call:
FALSE lm(formula = log(price) ~ ., data = newD[c("price", bestNames2)])
FALSE 
FALSE Residuals:
FALSE     Min      1Q  Median      3Q     Max 
FALSE -3.1696 -0.3267 -0.0233  0.2981  2.3637 
FALSE 
FALSE Coefficients:
FALSE                          Estimate Std. Error t value Pr(>|t|)    
FALSE (Intercept)             4.527e+00  1.254e-02 360.972  < 2e-16 ***
FALSE extra_people            3.434e-03  2.420e-04  14.189  < 2e-16 ***
FALSE maximum_nights_avg_ntm  9.311e-05  1.215e-05   7.663 2.07e-14 ***
FALSE balconyTrue            -9.181e-02  1.609e-02  -5.705 1.21e-08 ***
FALSE kitchenTrue             3.441e-01  3.568e-02   9.644  < 2e-16 ***
FALSE washerTrue              1.664e-01  1.493e-02  11.144  < 2e-16 ***
FALSE capPC1                  1.862e-01  3.626e-03  51.355  < 2e-16 ***
FALSE reviewScoresPC1        -2.678e-02  4.516e-03  -5.930 3.17e-09 ***
FALSE reviewsPC1              8.769e-02  5.236e-03  16.747  < 2e-16 ***
FALSE listingCountPC1         7.014e-02  4.805e-03  14.598  < 2e-16 ***
FALSE listingCountPC2        -2.033e+00  5.020e-02 -40.502  < 2e-16 ***
FALSE ---
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
FALSE 
FALSE Residual standard error: 0.5362 on 6987 degrees of freedom
FALSE Multiple R-squared:  0.5206,  Adjusted R-squared:   0.52 
FALSE F-statistic: 758.9 on 10 and 6987 DF,  p-value: < 2.2e-16
```