Anderson Nelson

APAN5200 – Frameworks and Method

Assignment: Kaggle Project

December 3, 2018

The dataset had numerous columns, and I quickly realized that there were variables that were not relevant to solving the problem of predicting prices for an Airbnb. I first started by forming assumptions on what are the variables that impact price. My hypothesis on the variables that impact customer service are supply, demand, locations, customers services, size in square footage, type of property and capacity. I removed about 40% of the columns because they were either duplicate, unnecessary, would cause my algorithm to be biased against certain groups of people or not relevant to the problem.

Data Cleaning:

Less than 1% of the entire dataset contained NA's and blanks on some of the columns. I tried a few different strategies to address including, median, mean, impute a constant variable such as 0 and even delete. Deleting is not possible on the scoring data, to complete a submission on Kaggle, the dataset had to have the same number of rows as the original dataset. Zip code, weekly and monthly price were the more difficult areas to address and were significant to the prediction.

Zipcode:

I assumed that there was a relationship between the neighborhoods and Zip code, and it turns out there was. Zipcode. I filtered Neighborhoods Group, and Zipcode from the broader dataset, and I was able to combine the zipcodes from both datasets (analysis and scoring) to ensure that I caputured all the occuring zicpcodes. I removed all the duplicates, and ranked zipcode by numerical order. That allowed me to see the neighborhoods clusters for each of the zip code. Since there were only 93 missing data in the scoring data, I joined the most recurring zip code into the scoring data by using the neighborhood group.

Weekly and Monthly Price:

Both datasets had missing data for the weekly and monthly price column, and they were both sizable amounts of the column, I tried a few different options:

1) Created a summary table that included neighborhood and average price and joined by neighborhood,

2) Calculated the mean of the entire column and applied the mean

3) Created a summary table that included the zip and the average price and joined by zip code

4) Segregated the weekly price into two datasets and created a training set and test set from the columns without NA. I then created a model and applied the prediction to the weekly_price columns, and repeated the same process in the monthly_price columns.

5) I also multiplied the price column by 7 to compute at the weekly price and by 30 at the monthly price in the analyis dataset. I used method 4 for the scoring data

```
> summary(dataset$weekly_price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  100.0   400.0   550.0   571.2   724.5   999.0   25307
```

Property Type:

A quick look at property type reveals that there are 35 unique property type. 98% of all the property types are concentrated into 5 types of properties. I renamed all the remaining as other.

Before

```
> str(dataset$property_type)
 Factor w/ 35 levels "Aparthotel","Apartment",..: 2 2 2 2 2 2 2 2 20 2 ...
```

After

```
> str(dataset2$property_type)
 Factor w/ 6 levels "Apartment","Condominium",..: 1 1 1 1 1 1 1 1 3 1 ...
```

Structure:

I discovered that the model's efficiency and accuracy is improved when categorical variables are converted into numerical variables I found that it can affect impact prediction accuracy. I spent some time addressing the structure of the data set to ensure that they are the appropriate types.

Price:

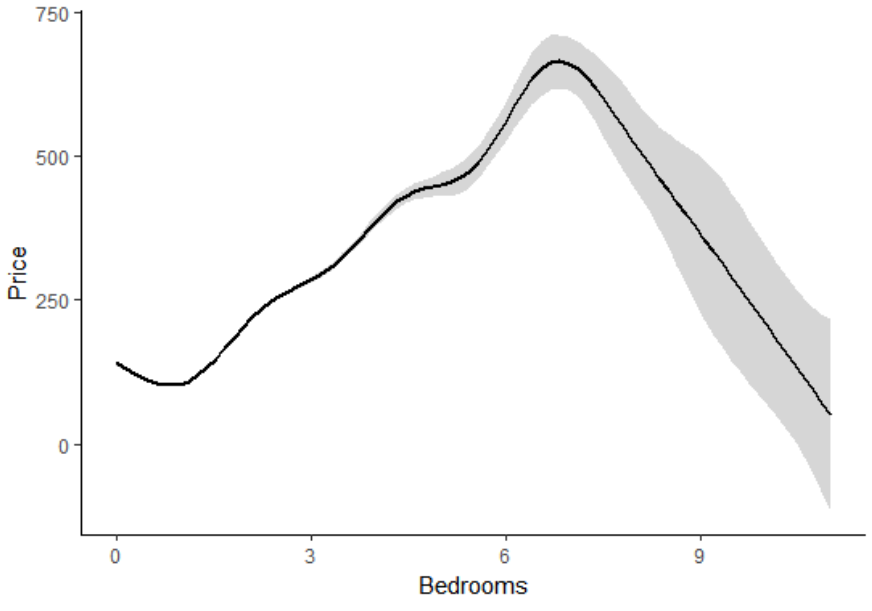
The below illustrate the distribution of the price column, an interesting observation is the range and the 0 price. It is unlikely that the seller would list their property for free. There are instances where that could occur, but that's most likely an outlier. The method that produced the best results for me was to remove the 0 occurrences.

```
summary(dataset2$price)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0      67     100     133     165     999
```

Data Visualization

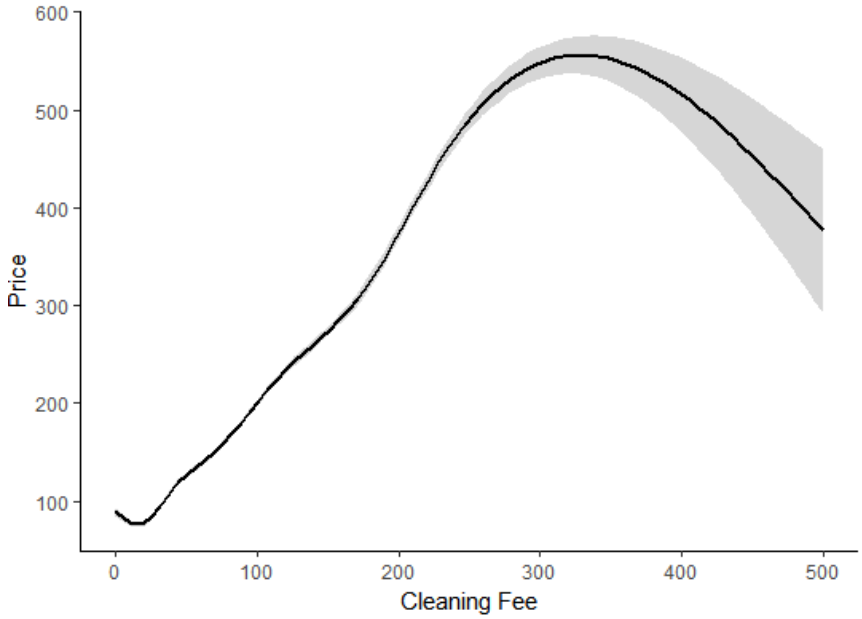
Price and Bedrooms:

I plotted price against the number of bedrooms and found a very interesting observation. The average price of an apartment increases as the number of bedrooms increases, hower the results peak at 7 bedrooms and the trends starts reversing.



Cleaning Fee:

Cleaning fee was one of the strongest predicters for my model, and I wanted to visualize the trends of cleaning fee, interestingly, it followed a similar pattern as price and bedroom.



Key Lesson Learned

1) Train time: The caret package allows you to test multiple variables in the model and visualize which parameters provided the best results. I tested~100,000 different combinations, and after about 8 hours the models was still computing. I realized that to compute this many variable is computationally expensive. In my best model, I tested multiple variables for one feature and utilized the best tune to test the subsequent variables.
2) Parameter Tuning: Not all parameter are created equal, some have a higher impact on the prediction than other. Focusingg on those parameters that have a higher impact can drastically change the results.
3) Be careful with predicting by using a prediction: I used predicted variables, weekly price, and monthly price, in my prediction, and while it was significant, I was able to get a better with the same model with it removed.
4) Use Google Cloud: Google cloud enables you to set up a virtual machine in the cloud, and install software such as R. The advantage is that you are able to run multiple models and compute models overnight and while your laptop is turned off.
5) Overfitting: Certain algorithm such as boosting can overfit the data. One of my boosting models had an RMSE of 0.7 and above 60 in my test set.
6) Cross Validations: The RMSE for a model that used cross-validation performed significantly better than models that didn't use cross validation, however, there is draw back, it is computationally expensive to perform.

Best Model

Throughout the kaggle competition, I submitted multiple models including linear regression, lasso, ridge, decision trees, random forest, extreme Gradient Boosting, Ada Boost, and Gradient Boost models. The model

that produced the best result was extreme Gradient Boosting. The caret package is a comprehensive package that streamlined the machine learning process. Here's how I approached parameter tuning,

I also tried to test all the parameters in a tuning grid and discovered that the time to complete this model was expensive which led me to isolate one parameter and test a range of predictors and use the best tune for my next independent tuning. My final tuning parameters had a best tune from all the previous models.

The final tuning parameter for my model were: ntrees: 6200, max_depth = 6, subsample = 0.5, eta= 0.1

What I would do differently.

1) **Be more organized:** Towards the end of the competition, I had multiple version of my models, and it became increasingly more difficult to keep track different version of my models. Also It would have helped if I had taken notes on the methods I attempted and recorded the result.

2) **Try different types of models:** Certain models have limitations I reached a point where each tuning improved the model performance by a marginal amount. The caret package has a variety of models that are available, I reached a point of diminishing returns for especially one method (extreme Gradient Boosting) and I should have examined another idea.

3) **Look at more packages:** There are numerous packages such as h20, zip code, etc that could have an impact on my model. Those packages streamline the process of transforming variable, work with text, longitude, and latitude.

4) **Explore amenities:** I didn't extract any insight from the amenities columns, but after observing some of the presentations, I realized that there was a lot of insight that could be extracted from that column.

5) **Combine different models:** Model prediction could have significantly improved by combing the results of my previous models.