Capstone Project Summary

Springboard Foundations of Data Science
By Jessie Huang
Mentor: Shmuel Naaman
October, 18, 2016

Background

Airbnb is a popular bed & breakfast option where homeowners (Hosts) can rent out their entire property or certain rooms (Listing) for travelers. With the increasing popularity of Airbnb coupled with dramatically increased rent prices in San Francisco, I'd like to understand the most important features that influence the listing price for San Francisco Airbnb listings and what a host can to do maximize their listing price.

Obtaining and Cleaning the Data

The dataset is provided by Inside Airbnb (http://insideairbnb.com/), an independent and non-commercial website. Inside Airbnb scrapes listing data from specific cities on Airbnb's site to analyze and explore how Airbnb is being used around the world.

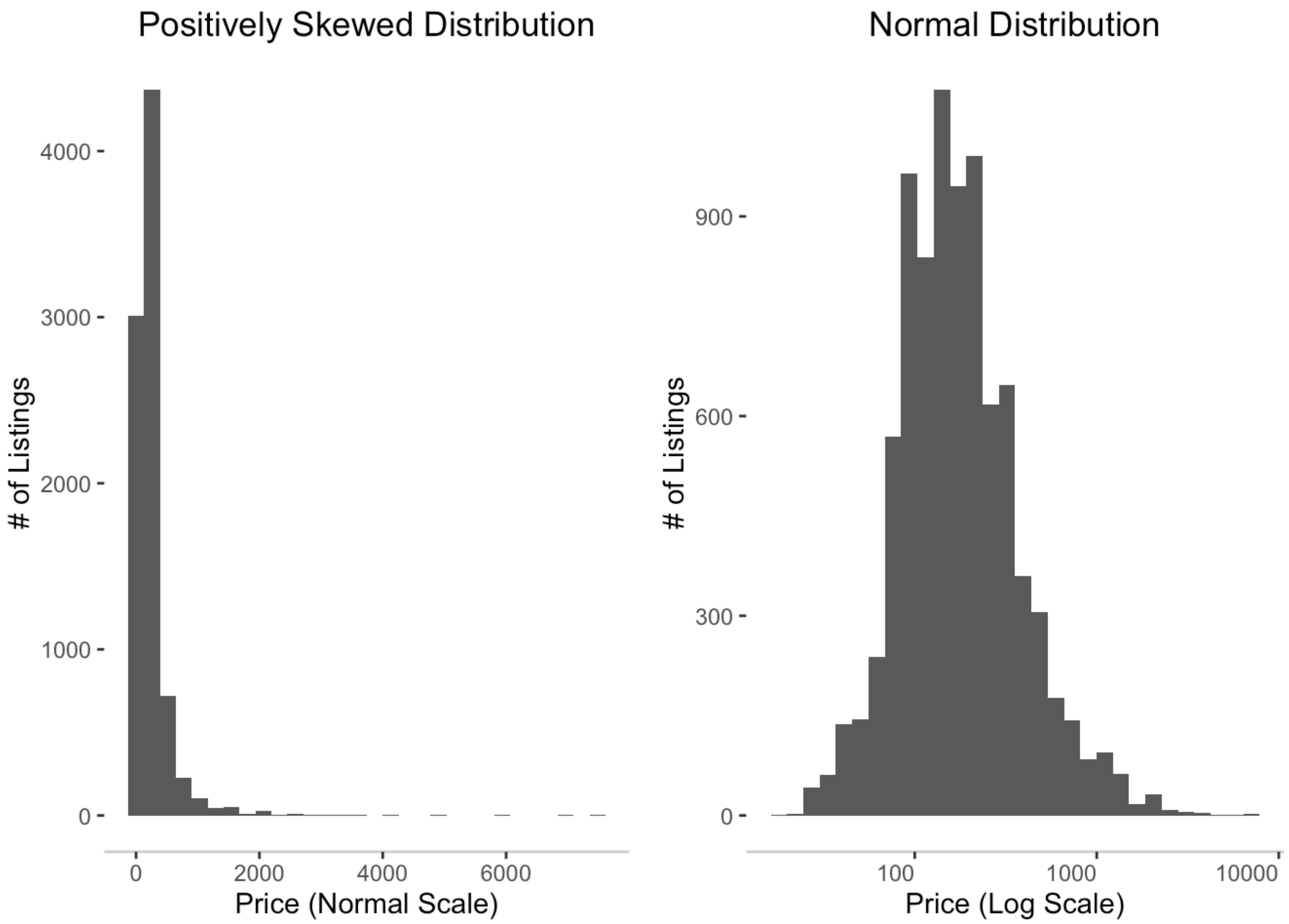
I began with a dataset that contains detailed information about each listing in San Francisco that was active at the time Inside Airbnb scraped Airbnb's site for San Francisco data on July 2, 2016. The raw dataset is 8,619 rows long and 96 columns wide.

After cleaning the data and removing irrelevant features from the dataset, there are 40 predictor variables to work with.

Data Analysis and Feature Selection

I used the linear model and correlation coefficient statistical tests to determine whether a feature is statistically significant to the price. The linear model is used when comparing two or more group means (categorical features) on a continuous dependent variable (log_price). For numerical features, I used the correlation coefficient to test for statistical significance. The correlation test is a normalized measurement of how two numerical features are linearly related.

However, a problem I ran into was the skewed distribution of the price feature. The linear model test can be used for features that have a normal distribution. After creating a new feature in the dataset that takes the price feature on a log scale, the distribution appears normal.



After the exploratory and statistical analysis stage, I ended up with the below list of features that were statistically significant and interesting to explore for the predictive model. The next step is to remove features that are closely correlated with each other and may cause overfitting or create bias in the results.

1. neighbourhood_cleansed
2. room_type
3. accommodates
4. host_listings_count
5. minimum_nights
6. is_dorm
7. bathrooms
8. beds
9. bedrooms
10. bed_type
11. number_of_reviews
12. reviews_per_month
13. review_scores_rating
14. cancellation_policy

Feature Engineering

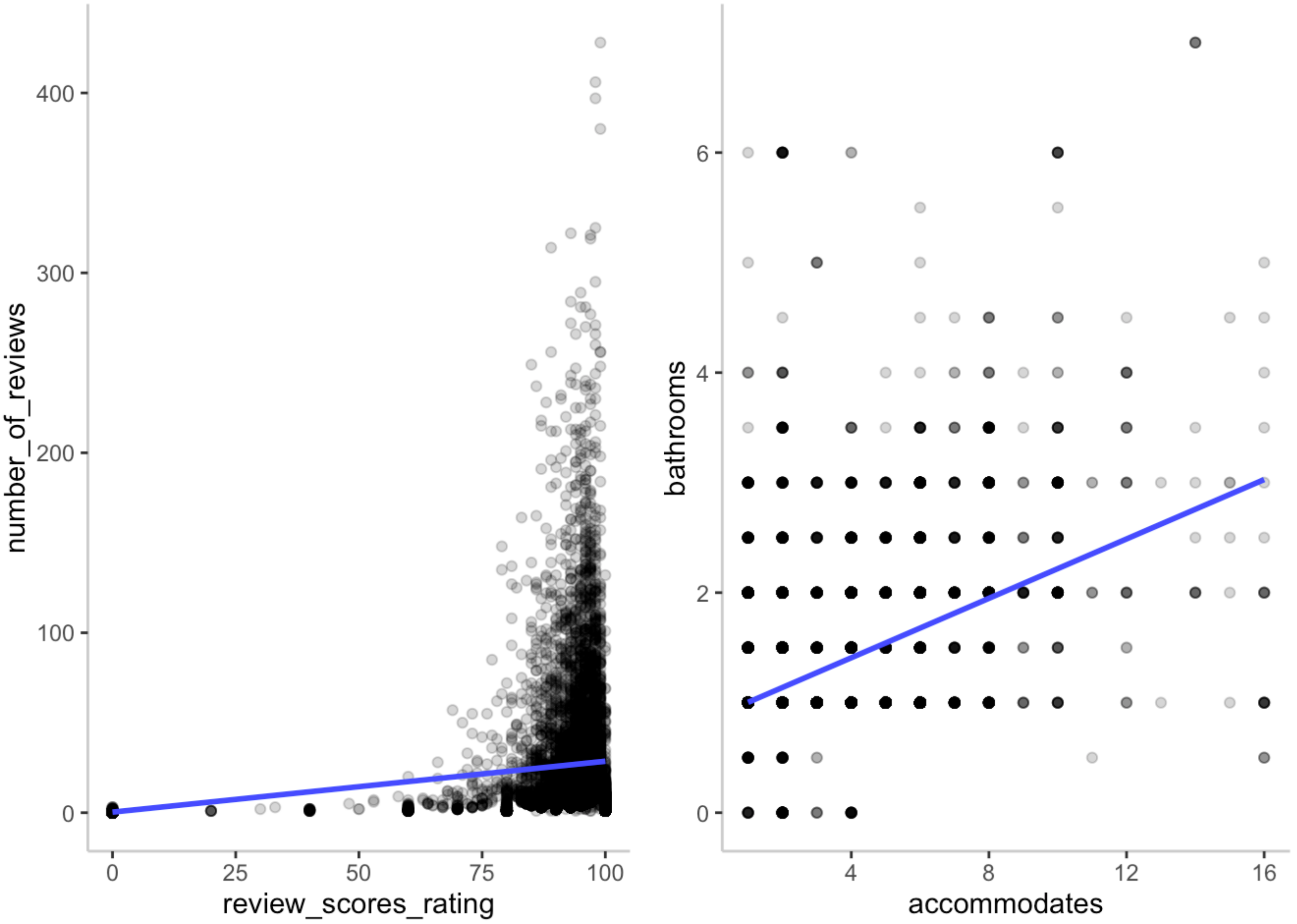
Some features were removed because they are too similar to each other. While they may improve the r-squared value, but is actually causing overfitting. Some examples are number_of_reviews, reviews_per_month, and review_scores_rating are most likely very closely related to each other. While including these do help with boosting the r-ssquared value, there is clearly some bias. This is confirmed because removing reviews_per_month only decreases the r-squared value by less than 0.01. I decided to remove reviews_per_month as it is too closely correlated to number_of_reviews where cor = 0.5671.

Similarly, the features: bathrooms, beds, bedrooms, and bed_type are likely very closely correlated to accommodates. After performing the correlation test on each of these features to *accommodates*, all of the features have a strong linear relationship where cor > 0.7, with the exception of *bathrooms* where cor = 0.4398.

The below scatterplot helps to confirm that we are not overfitting the model and there is little correlation between review_scores_rating/number_of_reviews and bathrooms/accommodates:

```
##
##  Pearson's product-moment correlation
##
## data:  listings$review_scores_rating and listings$number_of_reviews
## t = 30.887, df = 8586, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2970659 0.3351378
## sample estimates:
##       cor
## 0.3162291
```

```
##
##  Pearson's product-moment correlation
##
## data:  listings$accommodates and listings$bathrooms
## t = 45.375, df = 8586, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4225677 0.4566895
## sample estimates:
##       cor
## 0.4397873
```



Now that we have the final list of features to work with for predictive modeling, the next step is to understand which features actually have a high probability of influencing the listing price.

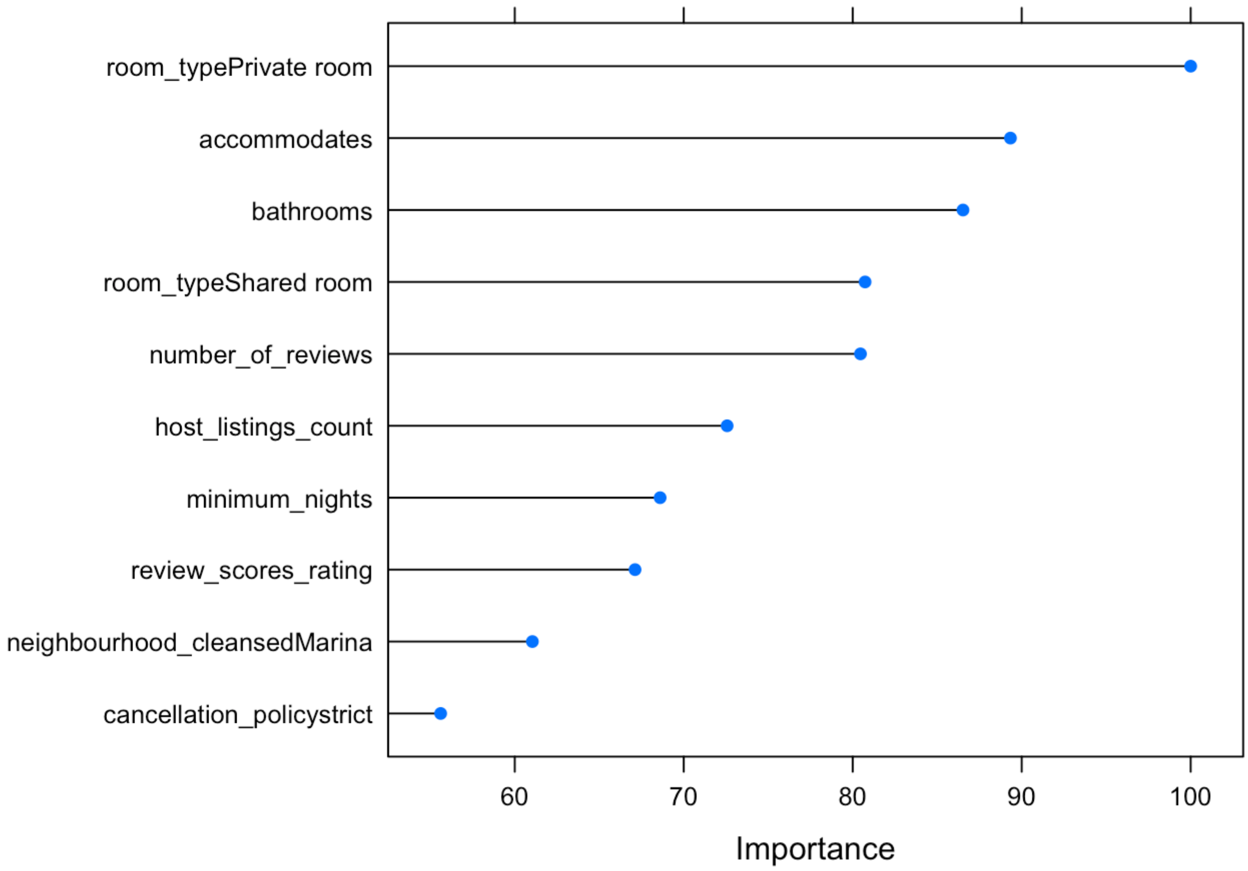
Predictive Modeling

In this seciton, we will fit and test various models to determine which model to use for future data on San Francisco Airbnb listings. To do so, the dataset will be cut in to a training set (80%) and testing set (20%). I will use the random forest and generalized boosted regression (GBM) models for predictive modeling on the training set. Based on the results of the models, I will take the highest performing model and verify its accuracy on unseen data with the testing set.

Random Forest Modeling

The random forest model is averaging multiple decision trees that are trained on different parts of the same training set. The goal is to overcome the overfitting problem of individual decision trees. An overfitted model fits the noise in the data rather than the actual underlying relationships among the variables. Overfitting usually occurs when a model is unnecessarily complex.

After feature engineering, the final iteration generates an r-squared value of 0.6330219 with a training size of 6,800 rows with the following features:



Based on the results of the random forest model, the top ten important variables in the model is given regarding each class of the feature. We can see that the listing price is influenced by the room type and features related to the listing accommodation size, which includes the number of people the listing can accommodate and the number of bathrooms available. The total number of reviews the listing has received also influences the listing price. However, we can see that the different San Francisco neighborhood breakdowns are also significant in influencing the price.

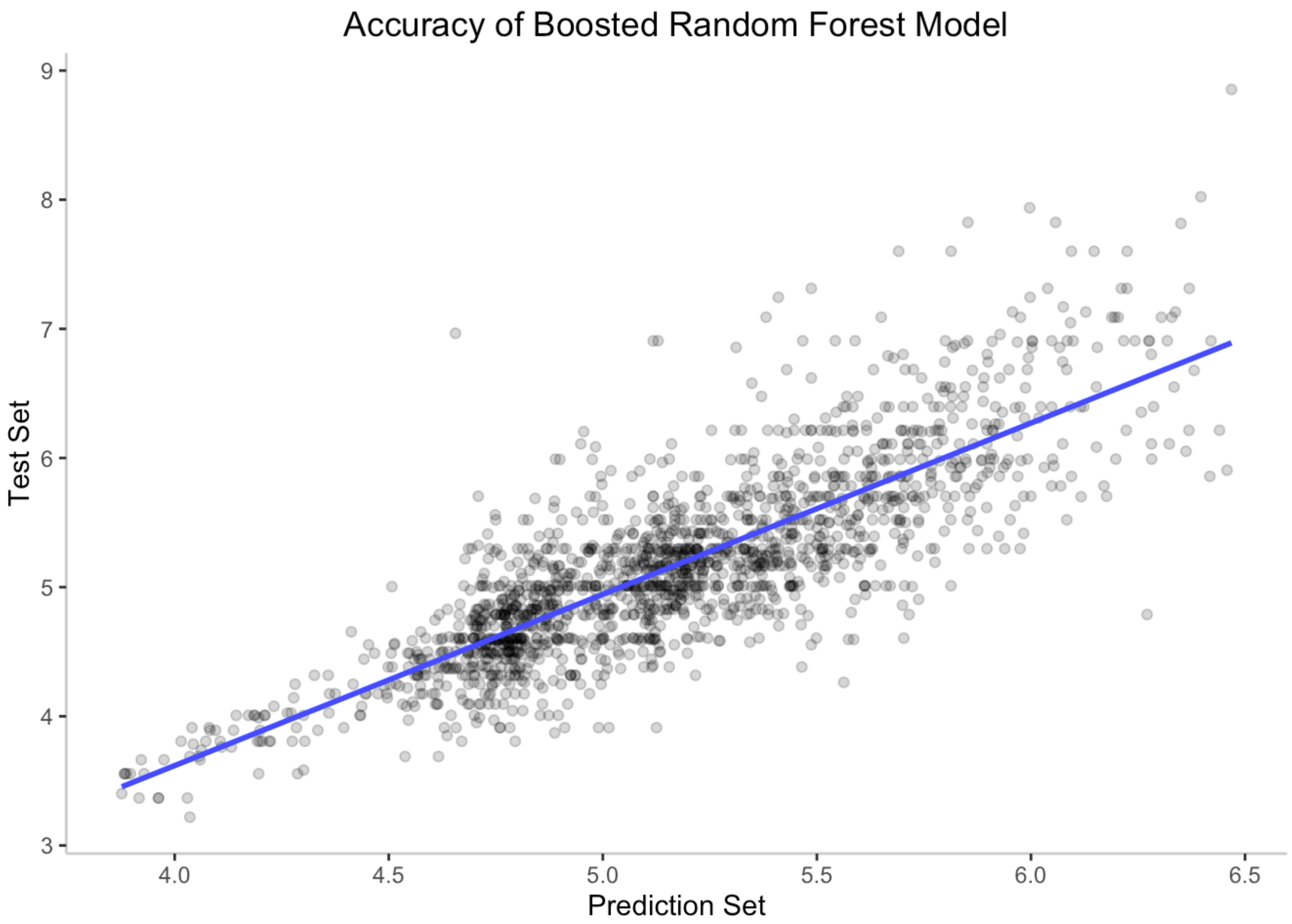
Additionally, the model returns an r-squared value of 0.6726716 for the training set and the r-squared value for the testing set is 0.6275013.

Because the dataset I am working with is quite small, the next step is to determine the validity of the random forest model to ensure that there was no bias in the way the training and testing data were split and to protect against overfitting. I will use the k-fold and grid search cross-validation techniques to evaluate the predictive models and also to tune the model parameters.

Random Forest Model Boosting: K-Fold and Grid Search Cross Validation

```
##
## Call:
## summary.resamples(object = results)
##
## Models: 50, 100, 150, 200, 250
## Number of resamples: 10
##
## RMSE
##       Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## 50  0.4363  0.4550 0.4595 0.4604  0.4665 0.4893    0
## 100 0.4371  0.4532 0.4593 0.4594  0.4642 0.4893    0
## 150 0.4355  0.4521 0.4588 0.4590  0.4646 0.4904    0
## 200 0.4350  0.4506 0.4596 0.4586  0.4636 0.4902    0
## 250 0.4362  0.4495 0.4592 0.4583  0.4632 0.4897    0
##
## Rsquared
##       Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## 50  0.6053  0.6528 0.6583 0.6653  0.6886 0.7139    0
## 100 0.6100  0.6574 0.6636 0.6686  0.6886 0.7169    0
## 150 0.6098  0.6587 0.6656 0.6699  0.6912 0.7182    0
## 200 0.6099  0.6594 0.6664 0.6709  0.6937 0.7197    0
## 250 0.6104  0.6587 0.6667 0.6711  0.6944 0.7209    0
```

After boosting the random forest model with k-fold and grid search cross validation, the r-squared values only improve slightly for both the training and testing sets. With the base model, the r-squared value for the testing set is 0.6275013. After tuning the model, the testing set r-squared value only improves to 0.6333874. We can also see from the scatterplot that there isn’t much difference.

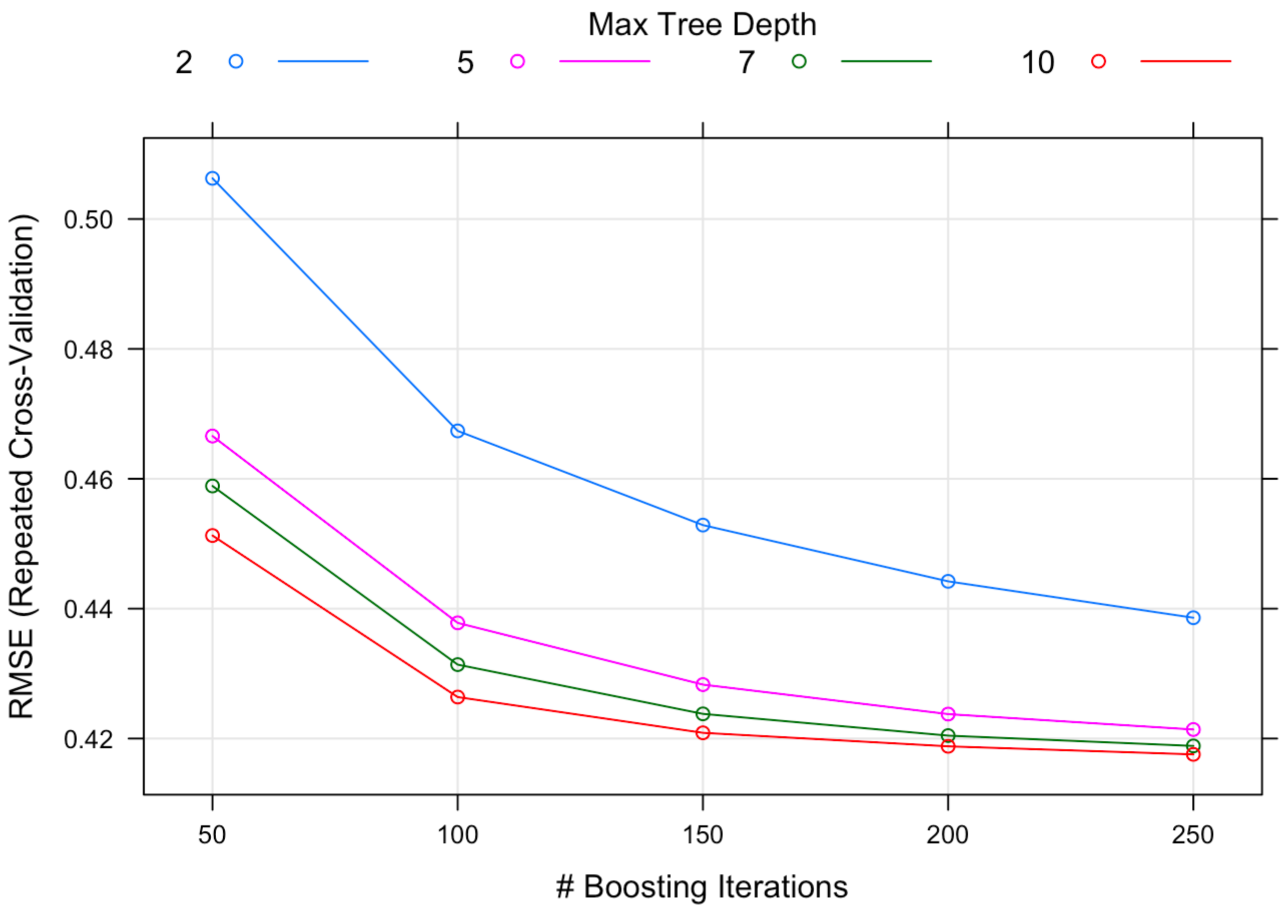


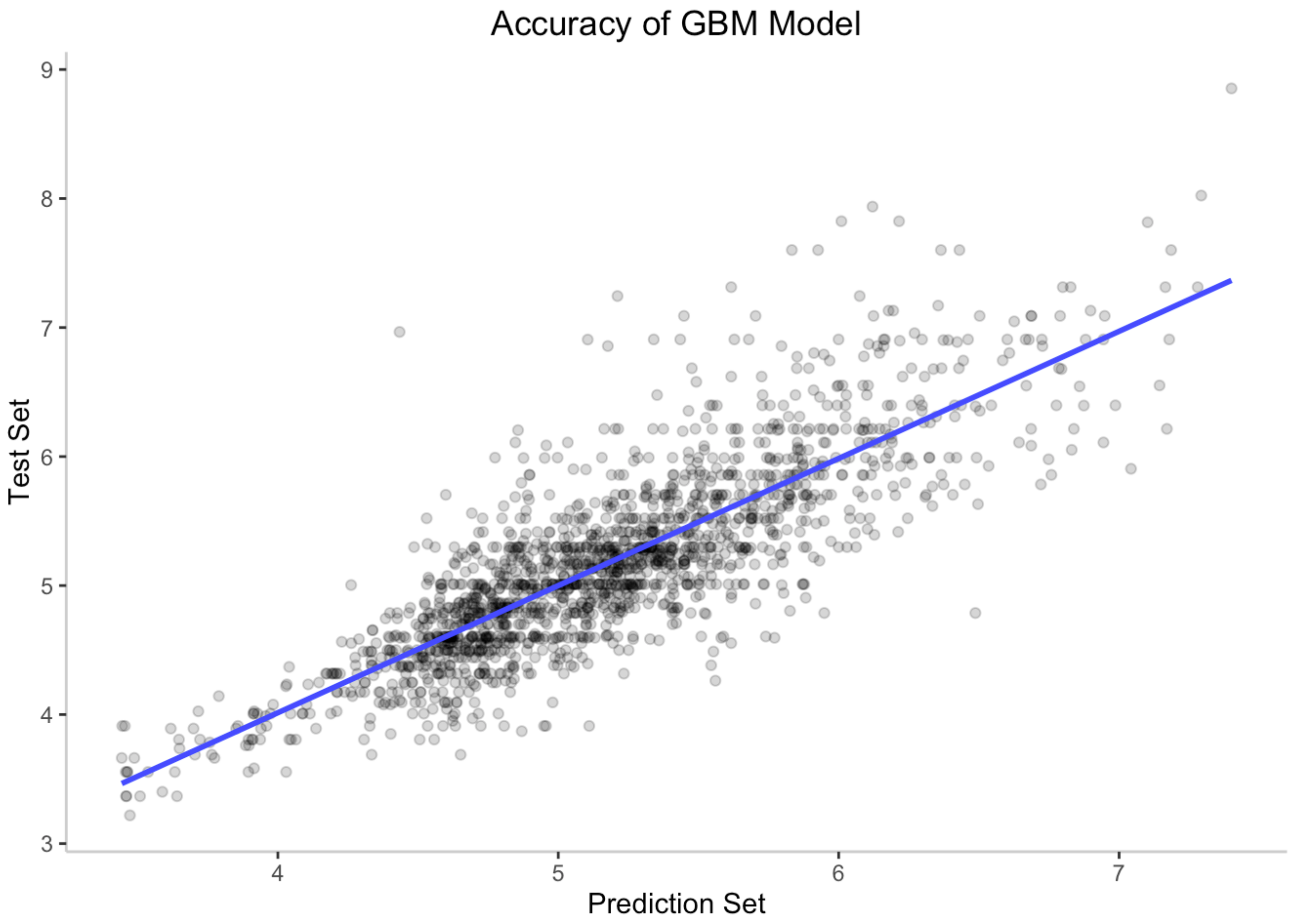
Generalized Boosted Modeling (GBM)

The next machine learning method that we try is the GBM model. This model fits to the data via boosted decision trees. GBM will train many decision trees sequentially, each time increasing the weight of data points predicted incorrectly the previous time and decreasing those that were predicted correctly. By combining the many trees, this produces a stronger predictor.

GBM: K-Fold and Grid Search Cross Validation

```
## Stochastic Gradient Boosting
##
## 6871 samples
##    9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 6184, 6183, 6184, 6185, 6182, 6185, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE       Rsquared
##    2                  50      0.5062636  0.5876091
##    2                 100      0.4673742  0.6217289
##    2                 150      0.4528664  0.6407134
##    2                 200      0.4442050  0.6520843
##    2                 250      0.4386009  0.6594164
##    5                  50      0.4665818  0.6297245
##    5                 100      0.4378124  0.6623628
##    5                 150      0.4283085  0.6742325
##    5                 200      0.4237667  0.6802034
##    5                 250      0.4213890  0.6835016
##    7                  50      0.4588906  0.6389040
##    7                 100      0.4313829  0.6707679
##    7                 150      0.4237993  0.6801691
##    7                 200      0.4204499  0.6847993
##    7                 250      0.4188586  0.6870795
##   10                  50      0.4512608  0.6491306
##   10                 100      0.4263867  0.6771808
##   10                 150      0.4208796  0.6842425
##   10                 200      0.4188025  0.6871583
##   10                 250      0.4175592  0.6890170
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.05
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were n.trees = 250,
##  interaction.depth = 10, shrinkage = 0.05 and n.minobsinnode = 10.
```




Accuracy of GBM Model

The GBM model performs much better than the random forest model, with an r-squared value of 0.685973. The final model plot tells that the overall error converge at around 250 trees. However, at even at 200 trees, tree depths of 5, 7, and 10 converge to approximately the same root-mean-square error (RMSE) so it's possible to speed up the algorithm by tuning the number of trees down to 200 for similar results.

Conclusion & Next Steps

The aim of this project is to build an accurate prediction model for San Francisco Airbnb listing prices based on what the host can offer in terms of their home and the host's history and reputation on the site. To achieve this, I applied statistical analysis to create a list of features to explore during the modeling phase and with feature engineering I narrowed the list down by eliminating the features that are too closely correlated and may cause overfitting. I then used the random forest and GBM models to train the dataset and selected GBM as the prediction model due to its relative higher accuracy in the cross validation.

During the predictive modeling stage, I found that:

1. The most important feature that influences the price is the number of people the listing can offer. Other features related to the accommodation size are just as significance in influencing the price. location of the listing in San Francisco.
2. The host's history on Airbnb is also important, as the number of reviews the listing has received also influences the listing price.
3. The neighborhood is surprising not as important as you would expect - it is likely because there are many neighborhoods and some neighborhoods do not have enough data.

However, we must remember some limitations:
1. The dataset is small, so there are probably some inaccuracies with the model.
2. We cannot conclude whether the statistically significant features help to increase the likelihood of a listing to being booked.
3. No booking data included in the dataset, so we cannot understand trends throughout the calendar year or most frequently booked listings.

For future extension of this project, it would be interesting to explore what helps a listing to be ranked higher based on its offerings and key words. Additionally, how does a host earn the badges that would encourage a potential guest to book the listing? Finding whether there's a pttern between review scores, key words, photographs would be interesting to understand what potential items helps a listing actually get booked after a host has priced their listing competitively.