Predicting Austin Airbnb Prices

Ammada Tuy
Sprinboard Data Science Intensive
Capstone 1 Final Report

1 Introduction

Airbnb is the one of the leading platforms in providing lodging experience. Within Airbnb's business logic, pricing is likely the most important feature for hosts and customers. Ensuring fair pricing directly affects booking activities. Studying the reasonable forecast and fair suggestion of prices of Airbnb listings can have huge real-life values and may generalize to other applications as well.

According to Bizjounals, Austin is Airbnb's biggest market in Texas, accounting for roughly 30 percent of the 1.5 million arrivals in the state in 2017. Large events such as the Austin Marathon, South by Southwest and Austin City Limit attracts a steady stream of visitors year-round. With plenty of listings and booking activities, Austin serves as a great example for the study of Airbnb pricing.

In this project, I build a price prediction model of Airbnb listings and make comparisons between different methods. The data is from Inside Airbnb's website. It covers all Austin listing details, calendar details, review details and associated geolocation information collected on September 19th, 2019. I used traditional machine learning methods (linear regression, ridge regression, lasso regression, support vector regression, random forest regression and gradient boosting) to output the predicted prices of listings.

2 Data Cleaning

The data was published in the form of multiple csv files.

- listings.csv - Detailed listings data for Austin
- calendar.csv - Detailed calendar data for listings in Austin
- reviews.csv - Detailed reviews data for listings in Austin
- neighbourhoods.geojson - GeoJSON file of neighbourhoods of the city

I used Python to import the csv files into my Jupyter notebook to clean and explore the data.

2.1 Feature Selection

The first step for cleaning was to shortlist the important features because including too many may lead to slow computation or wrong predictions due to excess noise. The listings.csv file which had the most data contained 11339 records and 106 different features. I checked for duplicate records but there were none. I analyzed each feature and decided which ones to keep for further analysis. For example, I dropped the features related to host such as host_name, host_since, etc. because they are less likely to be useful in determining prices.

The original data had two different neighborhood columns, one with numeric zip codes and one with names. Since more than one neighborhood name was linked to a zip code, I went through each zip code to determine which neighborhood name it had the most of. This information became the final neighbourhood column.

I also determined what data from the calendar.csv and reviews.csv files were important for my analysis. Using the calendar data, I created a new column to contain the number of days each listing was booked for the year and then merged it with the listings data. This feature was useful to explore which neighborhoods had the highest percentage of bookings and other insights. I did not take any features from the reviews.csv file but I used it as a reference while investigating missing values that were related to reviews.

I also created a new column called bedroom_bath_ratio which represents the ratio of bathrooms to bedrooms in a given listing. I did this after I filled in all the missing values for bathrooms and bedrooms. There were some records that had zero bedrooms and one or more bathrooms which I flagged as -1. The ratio would not work for these because there needs to be at least one bedroom. I also set the value to zero if the record had zero bedrooms and zero bathrooms.

After dropping features, merging the data and creating new columns, 16 features remained.

- listing_id – unique identifier for the listing
- neighbourhood – neighborhood represented as names
- zip_code – areas of the city represented as zip codes
- latitude – measurement location north or south of the equator
- longitude – measurement location east or west of the Prime Meridian
- room_type – type of listing space, ie., entire home or room
- accommodates – number of allowed guests
- bathrooms – number of bathrooms
- bedrooms – number of bedrooms
- beds – number of beds
- price – average price of the listing per night
- minimum_nights – minimum amount of nights needed to book a listing
- number_of_reviews – total number of reviews made by renters
- review_scores_rating – the rating determined by reviews
- number_of_bookings – number of days the listing has been booked over the year
- bedroom_bath_ratio – the ratio of bathrooms to bedrooms in a given listing

In general, the cleaning and preparation step also involved fixing structural issues, handling missing data and managing outliers.

2.2 Structural Issues

There were a couple of structural issues I had to reconcile. I converted the price feature from object type to float type. Since there were four options for the room_type feature, I converted it from object type to categorical type. Since there were several options for neighbourhood, I converted it from object type to categorical type.

2.3 Missing Values

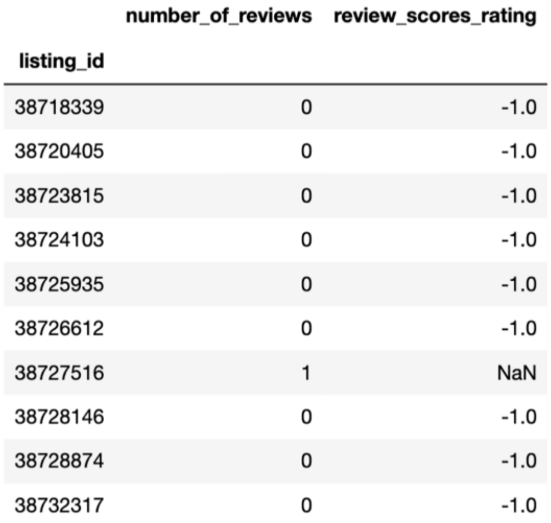
There were four features which had missing values. These were bathrooms, bedrooms, beds and review_scores_rating.

Bathrooms had 18 missing values. Since the number of bathrooms is typically correlated to the number of bedrooms, I used the non-missing bedrooms value to determine the median of all bathrooms of all records with the same number of bedrooms. For example, if the record contained a missing bathrooms value, I would look to see if there was a non-missing value of bedrooms. If the value of bedrooms was two, I would calculate the median value of bathrooms from all records containing two bedrooms.

Bedrooms had seven missing values. Since the number of bedrooms is typically correlated to the number of bathrooms, I used the non-missing bathrooms value to determine the median of all bedrooms of all records with the same number of bathrooms.

Beds had 11 missing values. Since the number of beds is typically correlated to the number of accommodates, I used the non-missing accommodates values to determine the median of all beds of all records with the same number of accommodates.

The review_scores_rating feature had 23% missing values which was relatively large. I realized that many of the missing values were a result of zero counts in the number_of_reviews feature. For those missing values, I flagged them as -1. Figure 1 shows an example of this.



	number_of_reviews	review_scores_rating
listing_id		
38718339	0	-1.0
38720405	0	-1.0
38723815	0	-1.0
38724103	0	-1.0
38725935	0	-1.0
38726612	0	-1.0
38727516	1	NaN
38728146	0	-1.0
38728874	0	-1.0
38732317	0	-1.0

Figure 1: Flagged missing review_scores_rating if number_of_reviews was zero

Interestingly though, there were a handful of records that had more than zero counts in the number_of_reviews as well as missing review_scores_rating. After analyzing the detailed reviews data, I realized that those records had automated review postings related to cancellations, which indicated that there were no reviews made by a renter yet. I also flagged these missing values with -1. Figure 2 shows the records that had the automated review comments.

Figure 2: Flagged review_scores_rating if there was an automated cancellation comment

2.4 Handling Outliers

I identified the features that had outliers by using the describe method and by plotting the data using the Plotly library. I decided whether keeping them was necessary for my overall analysis. I inspected each numerical column and looked for significant differences. I inspected the gaps between the min and 1st/10th percentiles, and the max and 90th/99th percentiles, as these made outliers more apparent.

As an example, I have included a graph (figure 3) to analyze the outliers for the number of bathrooms. It is obvious in the graph that the majority of the data lies below four bathrooms.

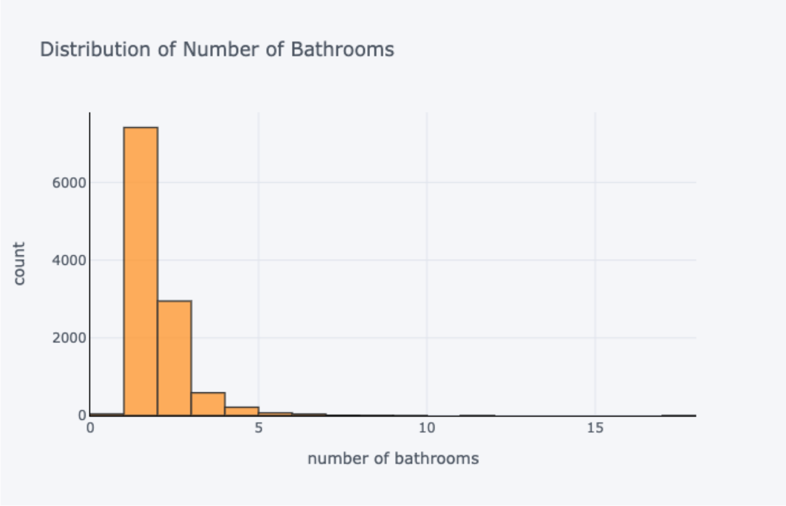


Figure 3: Bar plot of the distribution of the number of bathrooms

There were also several outliers in accommodates, bedrooms, beds and price. I made a note of these outliers. They were not necessarily bad data so I did not drop them from the main analysis.

There were also a handful of records with a price of zero which seemed incorrect. Perhaps this was the price at the time of the data scraping occurred and the host changed it soon after or maybe they were new listings. Since the amount was relatively small, I removed those six records altogether.

With these changes completed, the final dataset contained 11,333 records and 16 features.

3 Exploratory Data Analysis

Exploratory data analysis was a critical process of performing the initial investigations on the cleaned Airbnb data to discover patterns, spot anomalies and to check assumptions with the help of summary statistics and graphical representations.

Before I dove into plotting, I came up with a set of pre-defined questions that I believe would be interesting and relevant to analyze. I focused mainly on identifying the distributions of the different features and how they related to each other and to price.

As an Airbnb host, one of the first questions I would ask is about the price distribution of all listings. The average price is $391 a night. This number is much higher than the median of $145, due to several outliers of very high price points. The most expensive Airbnb listing was priced at $17,999, although the majority of listings were well under $300 a night.

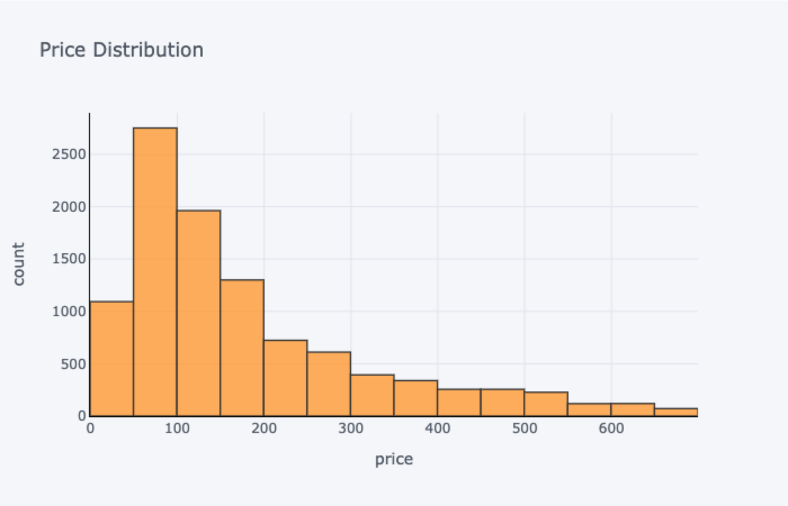


Figure 4: Bar plot of the distribution of price

Neighborhoods tend to have their own identities – different shops, food, activities and people. Due to the uniqueness of each one, I assumed there would be differences in the price ranges. I compared the price distribution of each neighborhood and found that Westlake Hills and Barton Creek were the most expensive areas, with several listings priced over $5,000 a night. Travis Heights, East Downtown and Old West Austin were on the higher end as well, with several listings over $2,500 a night.

There are four different room types – entire home/apartment, private room, shared room and hotel room. Due to the uniqueness of each room type, I assumed there would be major differences in the pricing. Entire home/apartment and hotel room prices are likely to be higher due to more space, privacy and amenities. Private room is literally a private room in a host's home. Shared room is a room in a host's home which you may share with a stranger. It is similar to a hostel. From my analysis, I found that most hotel rooms ranged between $175 to $550 and entire home/apartment ranged between $100 to $300. Private rooms ranged between $35 to $90 and shared rooms ranged between $20 to $50. Private rooms and shared rooms are considerably more affordable.

I discovered that between the different room types, private room get booked the most (as shown in figure 5). I was not surprised. They are much more affordable than entire home/apartment and you get some privacy. Hotel room get booked the least. This made sense because Airbnb is known as a platform utilized by mostly private home owners, not hotel companies.

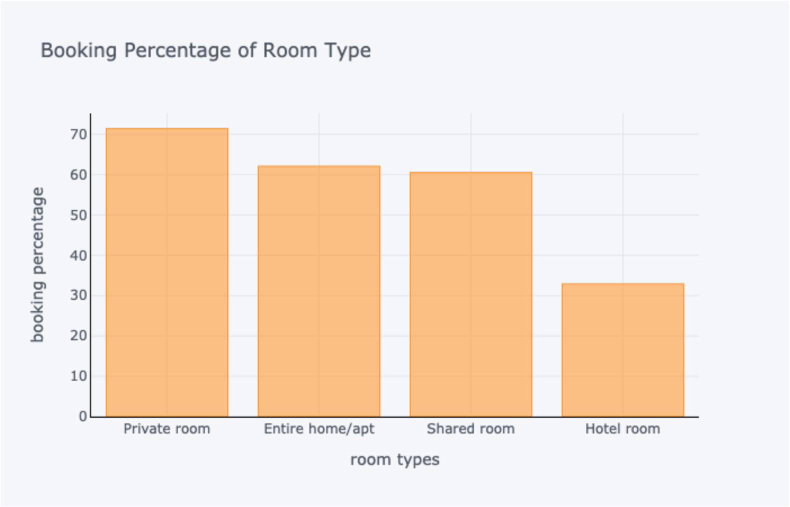


Figure 5: Bar plot of the booking percentage of room type

As the number of bedrooms, bathrooms, beds and accommodates of a listing increased, I expected the price to increase as well. I analyzed the price distribution of unique values within each feature. What I found was true – the price range typically increased as the values got larger. Listings with one bedroom, a half bathroom, one bed or the possibility to accommodate three people were booked the most.

I analyzed the review scores rating against price and didn't see an obvious correlation. What I did notice though, was that listings above $7,000 a night only had ratings above 60. There were several listings that were rated lower than 70 but were priced less than $1,000. Generally, most of the listings were above 90.

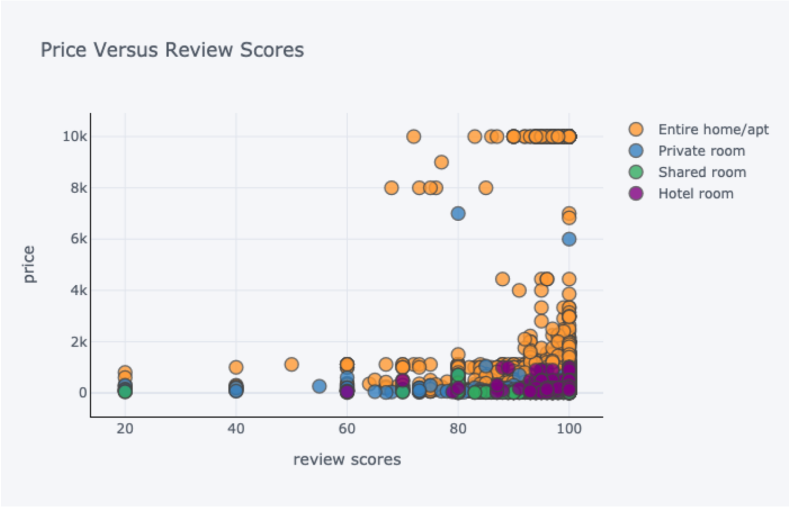


Figure 6: Scatter plot of price against review scores rating of different room type

Although exploratory data analysis gave me valuable insight, I needed to apply statistical inference to properly analyze and draw the appropriate conclusions.

5 Statistical Inference

Statistical inference is a critical process of diving deeper into the relationships observed in the data exploratory analysis step. The price feature is particularly significant in terms of predicting optimal Airbnb prices. I identified which pair of features were strongly correlated to price and looked at significant differences between subgroups in the data that may be relevant.

5.1 Correlation Coefficients

The correlation coefficient is a statistical measure of the strength of the relationship between the relative movements of two features. I used this calculation to determine the correlation and significance between price and various features. A positive correlation coefficient means that both features move in the same direction and a negative correlation coefficient means they move in opposite directions. A correlation coefficient near zero means there is no correlation.

Bedrooms had a positive correlation coefficient of 0.52, which means as the number of bedroom increases, price typically increases and vice versa. Bathrooms had a positive correlation of 0.51. Beds had a positive correlation of 0.41. Accommodates had a positive correlation of 0.55. All four features had a p-value of zero meaning they were statistically significant.

I also calculated the correlation coefficients for the features review scores rating, bathroom to bedroom ratio, number of bookings and number of reviews. I found that these features did not have any correlation to price because the correlation coefficient was near zero. These values were statistically significant because their p-values were close to zero.

5.1 Analysis of Variance

The Anova method, also known as the Analysis of Variance, is used when one wants to compare the means of a condition between two or more groups. This will test if there is a difference in the mean somewhere in the model, but it does not tell where the difference is, if there is one. If there is a difference, the Tukey HSD post-hoc test can be used to figure that out.

Figure 7 is a visual representation of the price distribution by room type - shared room, private room, hotel room and entire home/apartment. Based on the graph, it seems that pricing of hotel room and entire home/apartment is significantly different than that of private room and shared room. To dig deeper and to get a more precise answer to this question I used the Anova and the Tukey post-hoc test.

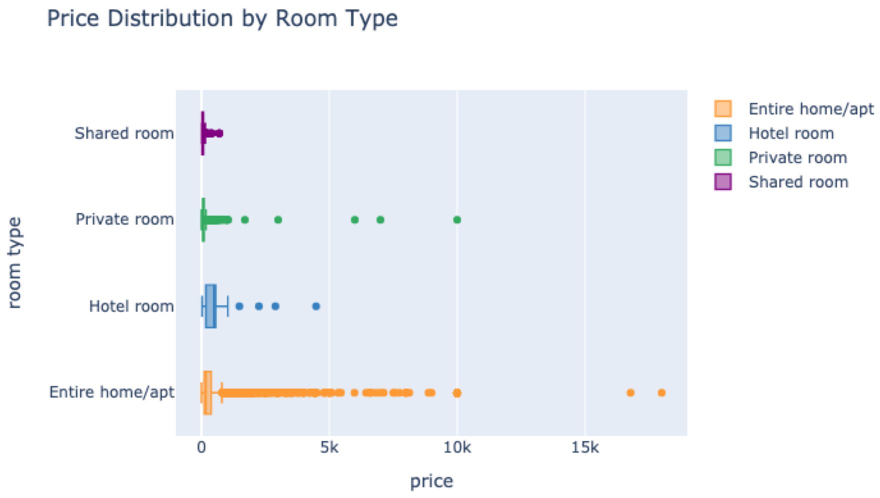
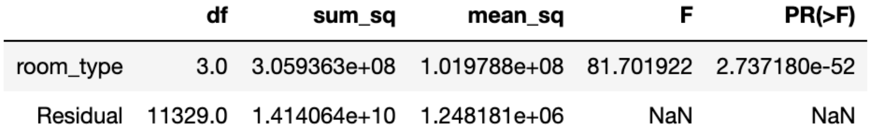


Figure 7: Box plot of the price distribution of different room type

I first applied the Anova method on room type and found that the F-statistic was 81.70 and the p-value (PR(>F) in figure 8) was near zero, meaning there was a difference between two or more different room types. Since there was a difference, I used the Tukey post-hoc test to determine where those differences were.



	df	sum_sq	mean_sq	F	PR(>F)
room_type	3.0	3.059363e+08	1.019788e+08	81.701922	2.737180e-52
Residual	11329.0	1.414064e+10	1.248181e+06	NaN	NaN

Figure 8: Result from Anova method

The Tukey post-hoc test compared the mean price of each room type against the mean of other room types, and determined if the difference was significant. I found that the mean price for entire home/apartment is significantly different than the price for private room and shared room. On average, private room and shared room is $378 to $413 cheaper than entire home/apartment. Price for hotel room is also significantly different than private room and shared room, about $400 to $435 more expensive. There was not much of a difference between entire home/apartment and hotel room, and between private room and shared room.

In the exploratory data analysis step, I observed huge differences in price ranges between neighborhoods so I looked at how neighborhoods compared in price. I used the Anova method to compare each neighborhood against the others. I found there were a handful of neighbourhoods that were usually significantly more expensive than the rest. These neighbourhoods were Westlake Hills, Barton Creek, Highlands, Gracywoods and Old West Austin. The prices ranged between $284 to $1,000 for more than 39% of the other neighbourhoods. Among these five neighbourhoods, Westlake Hills and Barton Creek were the most expensive, at least more than $476 for more than 77% of all neighbourhoods. The cheapest neighbourhoods were Cherry Creek, Crestview, Lamplight Village, MLK & 183, SW Williamson Co., Scofield Ridge and University Hills.

Since I saw a positive correlation of 0.52 for bedrooms, I decided to use Anova on the feature. Studios (zero bedrooms) and one bedroom were significantly cheaper than 50% of higher number of bedrooms. Two and Three bedrooms were significantly cheaper than 44% of higher number of bedrooms. Four bedrooms were significantly cheaper than 19% of higher number of bedrooms.

Since I saw a positive correlation of 0.55 for accommodates, I decided to use Anova on the feature. Listings with one, two or three accommodates were significantly cheaper than 46% of other numbers of accommodates. Listings between four to nine accommodates were significantly cheaper than at least 23% of higher number of accommodates. Listings with 18 accommodates were significantly more expensive than 31% of listings with accommodates of more than 18, specifically listings that can accommodate between 21 to 32 people.

6 Modeling

Before modeling I completed a few more data cleanup steps. I dropped features I didn't need such as listing_id, zip_code, latitude and longitude. I used one-hot encoding which transformed the categorical features into numbers so the models can understand whether or not a particular observation falls into one category or another. Since the distribution of price is heavily skewed to the right, I applied log transformation to make it less skewed. I also scaled all of the numeric features for more uniform and fair influence for all weights.

6.1 Methods

Once the data was cleaned, I proceeded with traditional machine learning. Since the target feature is continuous, I fitted regression models to the dataset. The first model I ran was linear regression. Since this is one of the simplest regression models, I used it as my baseline model to compare the other models against. Linear regression works nicely if the data is linear.

Next I ran the following models to predict price:

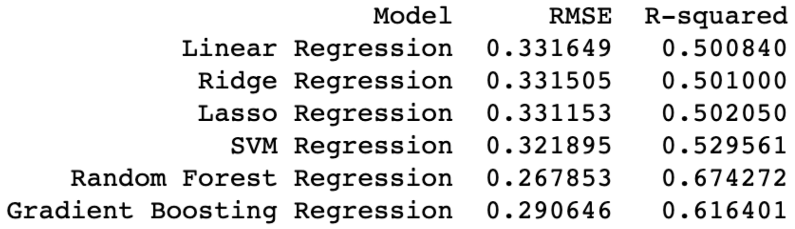
- ridge regression and lasso regression – regularization techniques to penalize the magnitude of coefficients of variables. The key difference is in how they assign penalty to the coefficients. Unlike ridge regression, lasso has the ability to shrink many coefficients to zero if they are not relevant.
- random forest regression and gradient boosting regression – both ensemble methods that aggregates many decision trees. Random forest builds trees in parallel while gradient boosting does it in a serial manner, where each tree tries to correct the mistakes of the previous. Good for non-linear data.
- support vector regression – can transform the data into a higher dimensional space, in which the data becomes linearly separable.

To perform the modeling, I used the most popular machine learning library in Python, Scikit-Learn. It has built-in functions for all the machine learning algorithms I used and a simple, unified workflow. From this library, I used a combination of randomized search and grid search with 3-fold and 5-fold cross validations to get the best optimal model parameters. Finding the best hyperparameters are important because they directly control the behavior of the training algorithm and have a significant impact on the performance of the model that is being trained. I chose to do cross validation over the classic hold-out method so that I can have better confidence in my prediction accuracy. Performing K-folds cross validation will allow me to loop through all the data and get the average score from all loops.

6.2 Results

I evaluated and compared the performance of the models using two different metrics, root mean squared error (RMSE) and r-squared. The RMSE value measures the magnitude of the margin of error. The r-squared value is a measure of how close the data are to the fitted regression line. Ideally, lower RMSE (closer to zero) and higher R-squared (closer to one) values are indicative of a better model.

Figure 1 displays the scores for each of the different models I performed. The baseline linear regression model had an RMSE of 0.33 and R-squared of 0.50. The best performing model was random forest regression. This model had an RMSE of 0.27 and r-squared of 0.67. Though random forest regression was the best performing model, it only scored 0.06 higher in terms of r-squared and 0.02 lower in RMSE than gradient boosting. Figure 2 and 3 shows a visual comparison of the models.



```
                      Model      RMSE  R-squared
          Linear Regression  0.331649   0.500840
           Ridge Regression  0.331505   0.501000
           Lasso Regression  0.331153   0.502050
             SVM Regression  0.321895   0.529561
   Random Forest Regression  0.267853   0.674272
Gradient Boosting Regression  0.290646   0.616401
```

Figure 1: Summary of RMSE and r-squared scores of all models

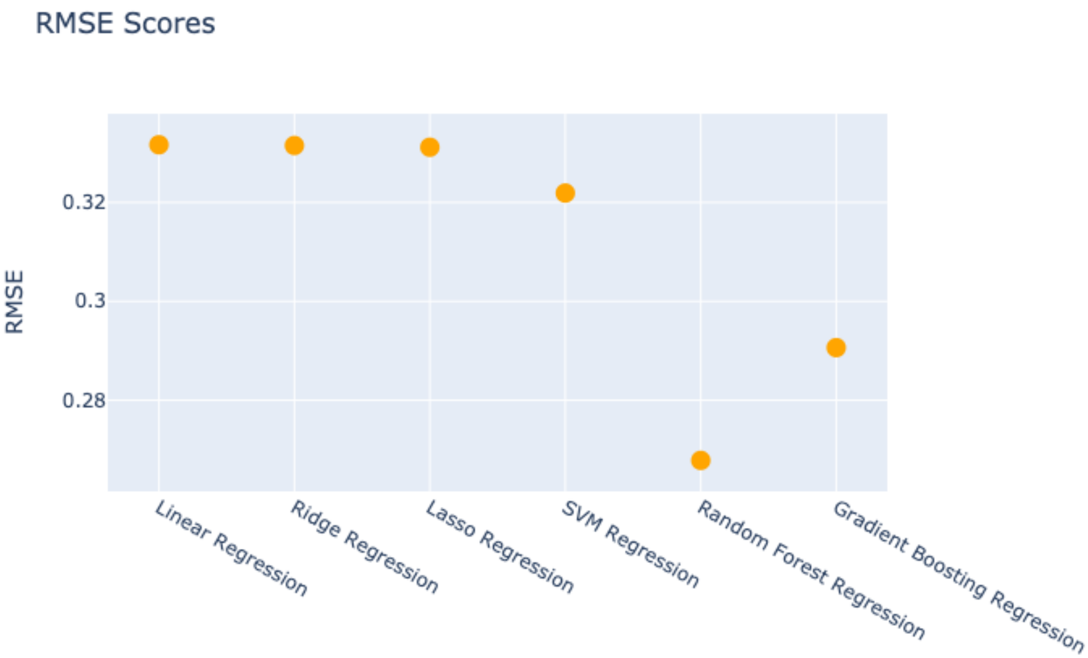


Figure 2: Dot plot of RMSE scores for all models

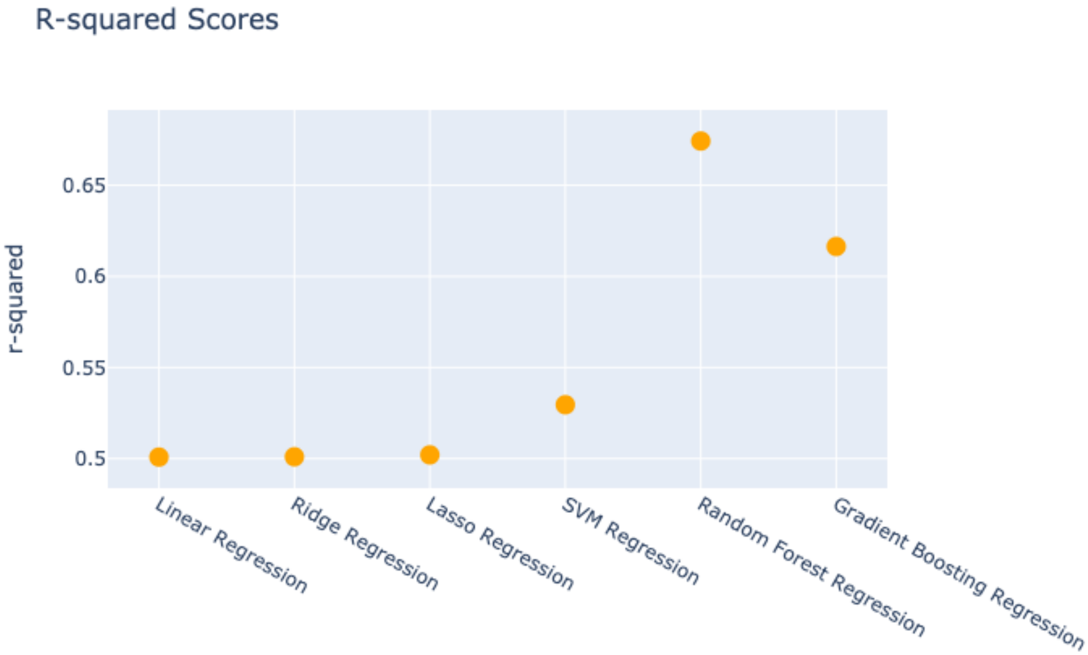


Figure 3: Dot plot of r-squared scores for all models

Gradient boosting had the second best score but training time was significantly more than all of the models. It took about 416 minutes. Random forest regression took about 42 minutes. The rest of the models took less than 10 minutes each.

6.3 Recommendations

Based on my findings, I recommend the following:

- Having more bedrooms, bathrooms, beds and accommodating more guests will allow a host to charge at a higher premium.
- Certain neighborhoods can charge more than others. For example, Westlake Hills and Barton Creek have listings over $5,000 a night.
- Having a listing that is either entire home/apt or private room will have greater chances of getting booked.
- Listings that are private room or shared room are significantly cheaper than listings that are entire home/apt or hotel.
- Hosts should try to maintain a high review scores rating. Listings at higher prices generally do not have scores below 70.

6.4 Future Exploration

I believe there is more improvement to be gained from the models. With more time I would have tried using other feature engineering methods and remove features that show signs of multi-collinearity. I would also try other machine learning algorithms. Perhaps neural networks would have shown an improvement. Incorporating calendar data provided by Inside Airbnb may have also helped, instead of using just the average price. Rates are known to fluctuate higher over holidays and popular events. The ability to track seasonal trends might lead to a better model. I would also like to see the transferability of my model when applied to other cities.

References

"Get the Data." *Inside Airbnb*, http://insideairbnb.com/get-the-data.html. September 19, 2019.

Salazar, Daniel. "Airbnb's most popular places to stay in Austin include ultra-modern guesthouse, tiny home." *Bizjournals*, 30 Jan 2018, https://www.bizjournals.com/austin/news/2018/01/30/airbnbs-most-popular-places-to-stay-in-austin.html.

Source code available at https://github.com/ammadatuy/Springboard_Capstone_1.