Capstone 1: Milestone Report

Problem

With over 4 million listings in 191 different countries, Airbnb has become an amazing investment opportunity for people all around the world. While it is a remarkable achievement, the supply side continues to grow, which can make being an Airbnb host difficult. What can a host do to allow their listing to stand out and ultimately increase their revenue? This analysis aims to understand the most important factors for predicting the prices and ratings for every listing in San Francisco.

Client

Airbnb would benefit from using this analysis by being able to advise their hosts on the most important factors that lead to higher ratings and higher prices. By understanding the most important factors for price, a host could optimize their listing to increase their revenue. By understanding the factors for ratings, AirBnB's host would be able to create a better experience for Airbnb customers, who will be more willing to use the service again, while fostering positive communication around booking with AirBnB.

The Data - Inside AirBnB

The data was obtained from independent, non-commercial organization that utilizes public information compiled from the Airbnb web-site and is available to download on http://insideairbnb.com/get-the-data.html. The original dataset is a static 365-day snapshot from October 2, 2017 and includes 96 features on 8933 listings in San Francisco. These 96 features include 6 date/time features, 33 numerical features, 32 categorical features, 21 textual features, and 4 id keys.

Process

We first started out by exploring each column to remove the unneeded columns. Information we got rid of included:

- **Meta Information:**:
    - Unused keys:(last_scraped, scrape_id)
- **URLs:**
    - Links to pictures or the listings that will not be used in the analysis: listing_url, thumbnail_url, medium_url, picture_url, xl_picture_url, host_url, host_thumbnail_url, host_picture_url
- **Redundant Information:**
    - Since our analysis takes places in San Francisco, CA the following features all have the same value for each listings: city, state, market, smart_location, country_code, country, jurisdiction_names
    - In our analysis we use the features "zipcode" and "neighbourhood_cleansed" for understanding the impact location has on *Price* and *Ratings*. We don't need the following features, since the previously mentioned two sufficiently capture the information needed: latitude, longitude, neighborhood
    - is_location_exact refers to if the longitude and latitude are exact or within a mile radius of the listings true location. Since we are not using latitude and longitude in our analysis we can drop this one as well.
    - calculated_host_listings_count, host_total_listings_count, and host_listings_count all have the same values, so we chose to just keep calculated_host_listings_count for our analysis
- **Irrelevant Information About Host:**
    - These are features are text columns that most likely wouldn't have a strong impact on the model if we were to attempt to transform them: license, host_name
    - I doubt that a customer bases their ratings or how much they are willing to pay off of a host's location or their verifications, so we can drop: host_location, host_neighbourhood, and host_verifications
- **Calendar Information:**
    - The calendar is not very accurate, because it cannot tell the difference between a listing where a host has blacked out their own dates because they aren't renting or a listing with its dates blacked out because it has already been booked by customers. A lot of host will blackout dates until the time is closer, so the availability metrics don't necessarily represent how many days they have left to rent: availability_30, availability_60, availability_90, availability_365
- or Any column with 1 or less unique values

From there we had 56 columns remaining. These included our Response Variables (Price & Ratings), 24 Numerical columns, 11 Text columns, 16 categorical columns, and 4 date/time columns.

Numerical: Of the 24 numerical columns (26 with the response), all were loaded in as a string, and 7 of those were loaded in with special characters (Currency or Percentage). We used a loop to replace these special characters with an empty string, and then converted all 26 to a float.

- Now that the data has been transformed into a numerical type, we can find and remove any listings that are no longer active. The reason to get rid of these listing is that our goal is to find ways to increase the ratings and price of listings and if a listing is inactive then they cannot benefit from our analysis anyway. They are not a good representation of the population we are trying to help, so they may negatively affect our predictive performance. Using the column '*number_of_reviews*' we removed any rows that had less than 1 more review.
- Next we looked at the Null Values for each numerical column, so that we could either impute the missing values, transform the feature into something else, or know if we needed to remove it:
- **Square Feet**: The square feet column has over 97% of its rows missing. Instead of just removing the column, we transformed it to a column that states whether a listings has included square feet.
- **Weekly & Monthly Price**: These columns have a high amount of missing values. These columns represent a separate price (A discount) if you are to book by the over a Week / Month. We transformed these to binary values that say if there is a discount or not for a longer stay
- **Bedrooms / Bathrooms / Beds**: Since the column accommodates had no missing value and is highly correlated with all these columns, we imputed the missing values based on the median value of the respective columns that matched with the value for accommodates.
- **Review Scores**: review_scores_rating (One of our response variables) had no missing value, so we can use that to impute the missing values for the other review scores, since they are all highly correlated with ratings
- **Security Deposit / Cleaning Fee**: Security Deposit is most correlated with bedrooms(.32) and price(.32) and Cleaning Fee is most correlated with accommodates(.56), bedrooms(.59), beds(.56), and price(.51). We imputed cleaning fee first with median value grouped by bedrooms, so that we could impute the security deposit with the cleaning fee.

- **Host Response Rate**: We removed this column, because there was extremely small variation in the listings. 95% of listings had 100% response rate.

Date/Time Columns: We turned all four columns to the numerical type, so we can add them to our model. The data was scrapped on October 2, 2017, so all dates are relative to this date. We used the datetime.datetime and datetime.timedelta package to convert each column into days since the time scrapped.

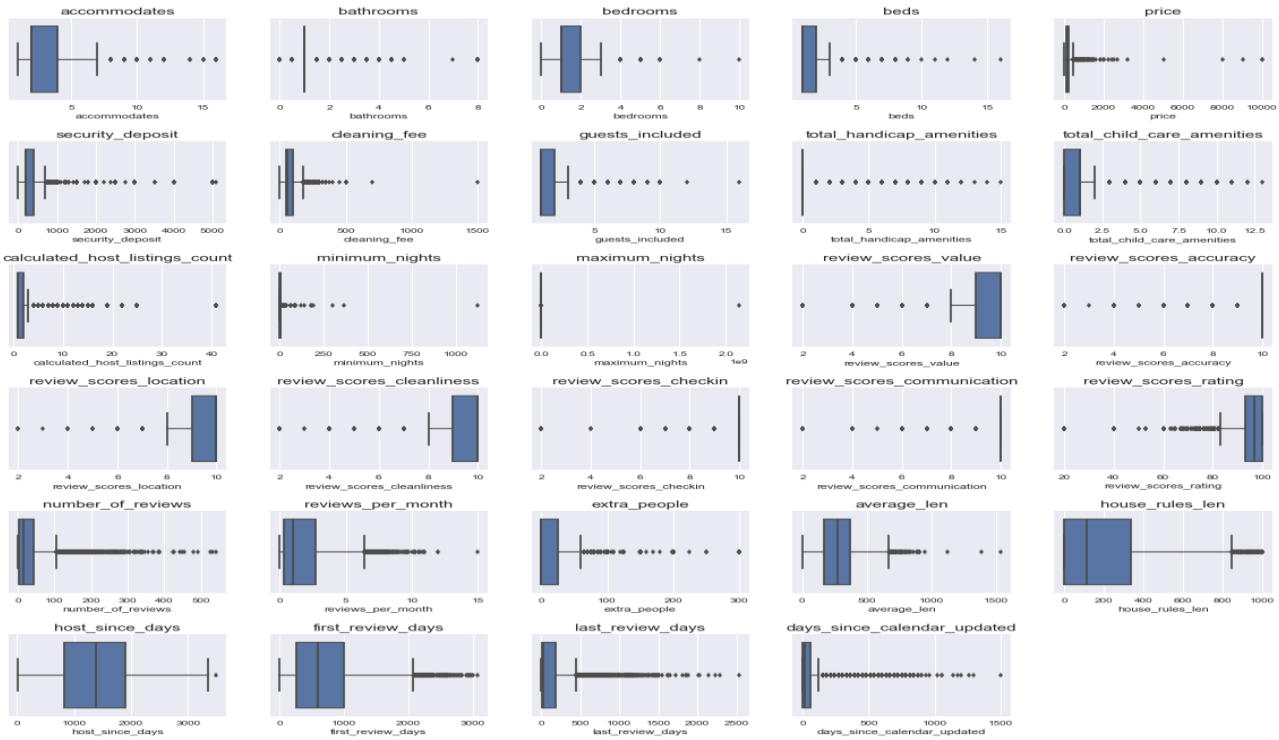
Text Columns: Since we can't impute for missing values in text columns, we just filled missing values with blanks ''. Then we calculated the average of the length of all the text columns, because there may be a relationship between the effort put into the listing summary, description, etc. with the ratings or price. We kept the length of the column *house_rules* and deleted the rest. The reason we kept this column is because if the list of rules is too long, then that may have a negative relationship with reviews. No one wants to have to worry the whole time about a bunch of rules while on vacation.

Categorical Columns: Our categorical data contains Binary columns, Multi-level columns, and a list within each row. We need to turn our categorical data from text into the categorical or boolean type.

- **Amenities:** The amenities column contains a list of all the amenities that a listing has. We then created dummy variables without removing the first column, since not having all the others doesn't imply you would have the removed amenity. In total there were 124 different amenities.
- **Binary Categorical Columns**: These columns were all *True/False* columns but loaded in as 't' or 'f'. We converted the text to 'True' or 'False', and then transformed it into boolean type.
- **Multi-Level Categorical Columns**: We changed these to categorical type and then checked from null values. Zip code and Host_response_time were the only ones to contain null values:
- **Host Response Time**: There were no easy ways to impute this column, so we used a naive method of backfilling the missing values.
- **Zipcode**: The neighbourhood_cleaned column had no missing values, so we created a dictionary of each neighborhoods most frequent value and then imputed the missing values by matching the dictionary neighborhood and its most frequent zipcode

Initial Findings

Numerical Features



We first constructed boxplots for all the numerical features, so that we can examine the outliers and skewness of each variable. Some things notice:

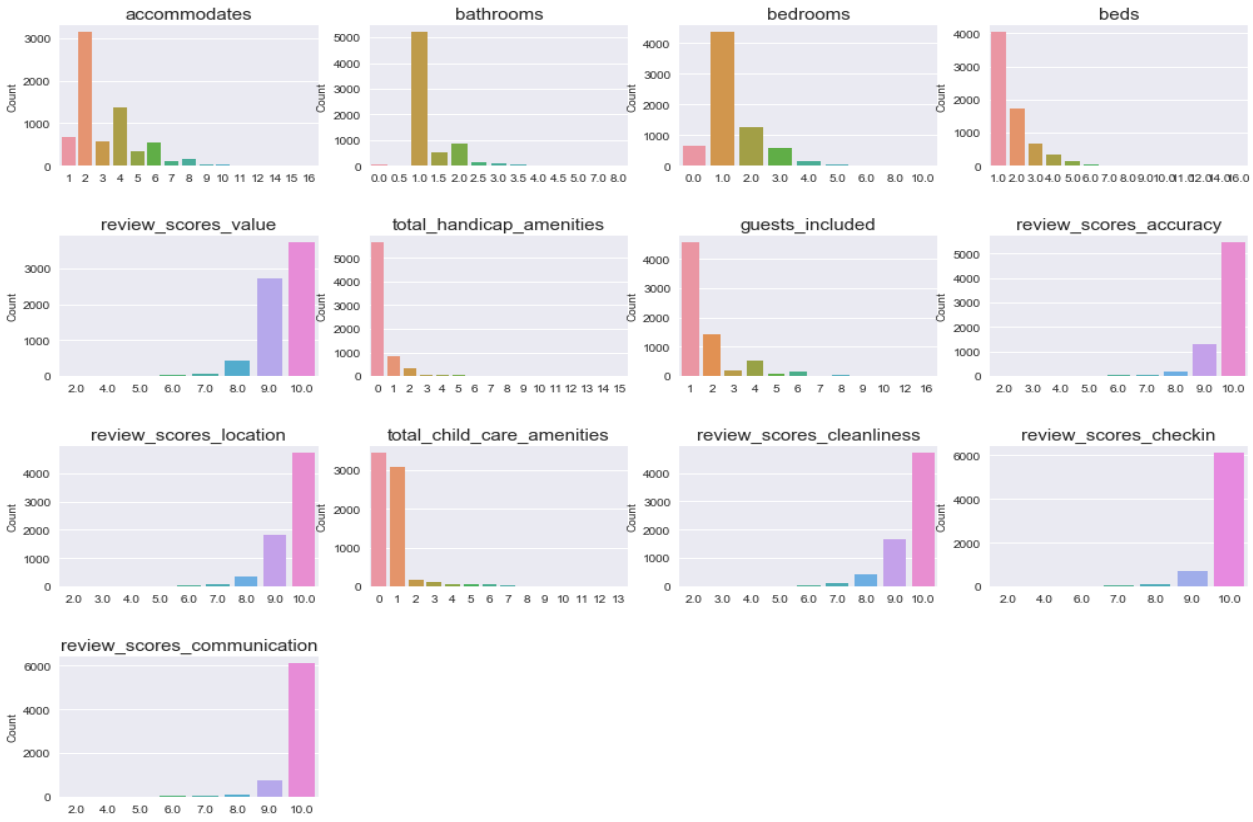
- All values are skewed, and most of them are skewed heavily right. This tell us that the majority of the listings in San Francisco are smaller listings, with some extreme unusual cases (such as 10 bedrooms).
    - Minimum nights and maximum nights have some extreme points, that appear to be errors, or the host chosen an arbitrary high number. (Maximum nights is over 2 billion). We removed these as the will only be harmful to our prediction model.
    - Price is also heavily skewed to the right, and since it is one of our response variables we will examine this more closely later.
- The features that are skewed left are the review scores. Most of them have an extremely small IQR as well. This is a common occurrence in ratings, where most

people give a perfect score unless there was some unusual negative event that occurred.

- Review scores for *cleanliness*, *value*, and *location* have the largest IQR range, which may indicate that people put more thought into these reviews and they may be stronger predictors.

We then separate the discrete numerical values and the continuous numerical values. For the *Discrete* numerical values, we created count plots to see how the values were distributed.

- It's clear that the columns beds, bathrooms, accommodates, and bedrooms have very extreme values. 99% of listings fall within half of the max beds, bathrooms, and bedrooms.



For the *Continuous* variables we constructed histograms to examine their distribution. We really just see how strongly everything is skewed. We should consider dropping more outliers, since they are so extreme.

Next we created a correlation heatmap of all the variables.

- We can see that Accommodates, Beds, Bathrooms, Cleaning Fee, and Guest included are all highly correlated with each other. May be useful to drop a few of these to improve our model's prediction.
- The review scores are all highly correlated with each other as well. We should consider dropping a few of these too. I think there is reason to keep them in as well. These may bring insights into what is the most important review a host can focus on to improve their overall rating.
- Days since first review and number of reviews are correlated, which makes since that the longer a host has been renting out the more reviews they would have. I assume that days since host isn't correlated because there are a lot of inactive listings from host that started and stopped Airbnb a long time ago. This could be because they chose not to comply with San Francisco's Airbnb regulations.

Categorical Features

We created bar charts for all 110 amenities. Only 56 of the 110 amenities applied to even 5% of the listings. These should be considered outliers it may be useful to drop these.

We then may a similar plot with all the True / False categories in our data. Has_square_feet, host_has_profile_pic, require_guest_profile_picture, and require_guest_phone_verification are very skewed and have a heavy majority of listings either True or False.

We then explore the Multi-Level Categorical features with count plots. It was clear the majority of listings were a house, apartments, or condominium with a real bed. The neighborhoods and zip codes were very well dispersed.

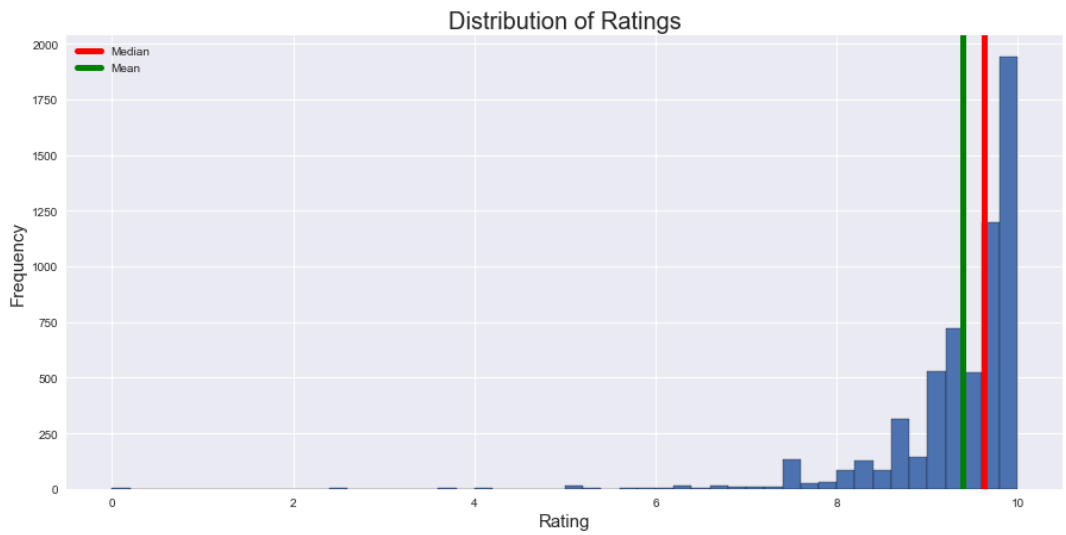
Price

From there we explore **Price**. We constructed a distribution plot that showed us the center of the distribution falls at $153 and the mean is at $210 with a right skew with a long tail. The Inter-Quartile Range for price was $105 and $250, yet the prices extend to around $10,000. The IQR rule for outliers shows us that there are 347 outliers for price and some of those outliers have a large influence on the entire dataset and need to be addressed. Over half of the listings in San Francisco are 1 bedroom, 1 bathroom, 1 bed, and accommodates 2-3 people, so it isn't surprising why the prices are right skewed. Some values seem to be outliers and should be removed from the dataset, so that they don't negatively affect the prediction models.

The characteristics of higher priced listings:

- The more a listings can accommodate / bedrooms / bathrooms / beds
- There are 37 different neighborhoods with a range in medians of $80 to $217. The Marina has the highest median price and seems to one of the most expensive places to rent an Airbnb
- Business Travel Ready listings are statistically significantly higher in price
- Non-Instant Bookable listings are statistically significantly higher in price

Ratings



The distribution and boxplot for listings ratings showed that there was a strong grouping above a rating of 9 or higher and that the data was heavily left skewed. People tend to give high ratings as long as their experience was positive. Of the 5000+ listings only 296 were considered outliers on the lower boundary IQR outlier rule.

Although the rating is one of the variables we are trying to predict, there are 7 total review scores. These are more specific ratings such as communication, location, etc. Looking at the correlation table of ratings, we see that the only significant relationship ratings has are the other reviews. Although they are correlated with each other I think it may be important to keep these as these could be a useful way to understand what customers appreciate more compared to the other review scores.

The characteristics of higher rated listings:

- The more a listings can accommodate / bedrooms
- 8 of the top 10 rated property types have less than 30 listings and 6 of those have less that 10 total listings. Also, the 132 listings that don't supply a couch, air mattress, pull-out sofa, or futon are higher rated on average than those that supply a bed. The unique property types and bed types have higher ratings.
- Having an Entire House / Apartment is higher rated than having a private room or having to share a room.
- The Business Travel Ready listings are higher rated
- The listings that are not Instant Bookable are higher rated

Modeling

Created Dummy Variables

Starting out I created dummy variables, while dropping the first column for all the categorical features. We dropped the first one because one level of the categorical feature becomes the reference group during dummy encoding for regression and is redundant.

Train/Test Split

Next, I created different splits of my data for both *ratings* and *price*. This way I can include rating as a factor for price & vice-versa. I split the data into 70% training set and 30% for testing.

Standardize the Features

I used the standard scaler from sklearn to standardize the training and test sets. Using variables without standardization can give the variables with larger ranges greater importance in the analysis. Transforming the data to comparable scales can prevent this problem. We only need to worry about this for models that use some form of a distance formula (Ridge, SVR, etc.) while others don't (Decision Tress, Random Forest, etc.). Since, it is not necessary to standardize our features for all models I created them as new variables of the train/test sets.

Custom Function - Reg_Model

I created a custom function that can either be used to perform cross-validation or perform a grid search for finding the optimal hyper-parameters (gridsearch=True).

- Cross-Validation: This allows me to plug in any estimator with scaled or unscaled data, split the training data into 5 splits, and obtain 5 validation scores.
    - It *prints* out the 5 r-squared validation scores, 5 training scores, the average r-squared validation score, the average train score, and the average MSE.
    - It *returns* the model with the training set fit to it for if we wish to make predictions and also the cross-validation model.
- Grid-Search: Allows me to add a range of values for whatever hyper-parameters I wish to test with the grid search.
    - It *prints* the average r-squared validation score, average test score, the best estimator and the best parameters.

    - It *returns* the grid search model

Custom Function - SFM

I created a function to plug in a model then perform feature selection with sklearn's SelectFromModel. This removes all features below a set threshold and is great getting rid of weak features that may be hindering our model's performance.

- The function *prints* the average r-squared validation score, the average test score, the average mean squared error, and the new shape of the training set.
- It *returns* the new X train and X test sets, so we can use it in the grid search / final model if it performs better than the model without it. It returns the SelectFromModel meta-transformer also.

Applying Fixes From EDA

Regression can be get thrown off very easily by extreme values, and this dataset has a quite a few extreme values. In our Exploratory Data Analysis, we made several observations of outliers and multicollinear features that we will remove before we continue.

- Removing observations with *extreme outliers.*:
    - We removed all observations that had a maximum night value above 1125 days. It is un reasonable to image anyone renting for that long, but over 44% of listings had a value of 1125 for their maximum night. My guess is this is the maximum value possible and the 13 observations over this were due to some error.
    - We removed all observations that has a minimum night value above 60 days because even if these were not errors, this is a very extreme case of listings that only represent .0001% of listings in San Francisco. It would be better to remove these, to prevent them from hurting our model.
    - We removed all listings with a price greater than $1000 a night. These are also extreme cases that do not represent the overwhelming majority of the listings in San Francisco and these really skew our data.
- Removed some *multicollinear features*:
- We chose to remove one of any pairs of features with a correlation of .75 or higher. I felt that these features were well represented by the other pair if the correlation was that high. Since, removing features does not always improve a model I saved the new training set to its own variables (X2_train and X2_test), so that we can try both the datasets on each estimator, then use grid search the best scoring one.

Price

Modeling

Baseline Estimator

Running a linear regression model, we obtained a baseline of -2.684e+17. Then applying the SFM custom function on the X2 features we improved the validation r-squared score 0.0674 with a standard deviation of 0.0165. This is great improvement, but still poor results. This will be our new baseline for comparing the rest of the models.

Results

For each model we used, we trained the model with the data from both the dataset with all the features and the dataset with the features / outliers we removed. Next, we performed our feature Selection on both models and then took the best scoring model from the four possibilities and then tuned the hyper-parameters for that chosen model. These were the results for each model:

- Ridge:
    - Average Validation Score: 0.6361
    - Validation Standard Deviation: 0.0361
    - Average Train Score: 0.6684
- Lasso:
    - Average Validation Score:0.6267
    - Validation Standard Deviation:0.0404
    - Average Test Score: 0.6747
- ElasticNet Score:
    - Average Validation Score: 0.6344
    - Validation Standard Deviation:0.0380
    - Average Train Score:0.6679
- Support Vector Regression:
    - Average Validation Score: 0.5864
    - Validation Standard Deviation: 0.0462
    - Average Train Score: 0.6442
- K-Nearest Neighbor Regression:

    - Average Validation Score: 0.3355
    - Validation Standard Deviation: 0.0133
    - Average Train Score: 0.4087
- Decision Tree:
    - Average Validation Score: 0.5487
    - Validation Standard Deviation: 0.0232
    - Average Train Score: 0.6450
- Random Forest:
    - Average Validation Score: 0.6161
    - Validation Standard Deviation: 0.0513
    - Average Train Score: 0.8873
- Extra Random Forest:
    - Average Validation Score: 0.6344
    - Validation Standard Deviation:0.0441
    - Average Train Score:0.9479
- AdaBoost:
    - Average Validation Score: 0.6076
    - Validation Standard Deviation:0.0388
    - Average Train Score:0.8090
- Gradient Boosting:
    - Average Validation Score: 0.6606
    - Validation Standard Deviation:0.0433
    - Average Train Score:0.8584
- eXtreme Gradient Boosting:
    - Average Validation Score: 0.6516
    - Validation Standard Deviation:0.0444
    - Average Train Score: 0.8929

The **3 best models** are:

- **Gradient Boosting:** It has the best validation r-squared score with relatively low variation.
- **eXtreme Gradient Boosting:** It has the second-best validation r-squared score also with relatively low variation.
- **Ridge Regression:** Even though it has the 5 best validation r-squared score, it has a lower variation then the third and fourth best scoring models.

The **Gradient Boosting model** is our choice with having a **Test Score** of: *0.6843*. This model explains around 68% of the variation in the model. Looking at the feature's

coefficients and feature importances, we get a better understanding of how features impact the price for a listing in San Francisco.

- The size of the listings (bedrooms, accommodates, bathrooms) are very important features for predicting price.
- Higher ratings tend to bring in higher prices for listings.
- Renting an Entire home/apt significantly impact the price
- Location of the Airbnb:
    - The best places to rent out an Airbnb to increase revenue are South of Market, Castro/Upper Market, and the Marina.
    - The worst locations to rent out an Airbnb in San Francisco are Parkside, Outer Richmond, and Twin Peaks.
    - Having a high "walk score" and being closer to the more populated spots in San Francisco is important for increasing revenue.
- The best amenities that a host can supply to increase their revenue are: Air conditioning, an indoor fireplace, patio or balcony, a hot tub, wireless internet, shampoo, a doorman, a microwave, and extra-pillows and blankets.
- The text length of a listing's ad is considered an important feature and used around 4% of the time when making predictions.

Ratings

Results

The results for each model:

- Ridge:
    - Validation Score:0.7295
    - Standard Deviation: 0.0627
    - Average Train Score: 0.7762
- Lasso:
    - Validation Score:0.7257
    - Standard Deviation: 0.06076
    - Average Train Score: 0.7633
- ElasticNet:
    - Validation Score:0.7247
    - Standard Deviation: 0.0597

    - Average Train Score: 0.7629
- Support Vector Regression:
    - Validation Score: 0.6980
    - Standard Deviation: 0.0580
    - Average Train Score: 0.7523
- K-Nearest Neighbor Regression:
    - Validation Score: 0.3409
    - Standard Deviation: 0.0595
    - Average Train Score: 0.4549
- Decision Tree:
    - Validation Score: 0.6118
    - Standard Deviation: 0.0990
    - Average Train Score: 0.7307
- Random Forest:
    - Validation Score: 0.7163
    - Standard Deviation: 0.0412
    - Average Train Score: 0.9366
- Extra Random Forest:
    - Validation Score: 0.7266
    - Standard Deviation: 0.0505
    - Average Train Score: 0.8699
- AdaBoost:
    - Validation Score: 0.6891
    - Standard Deviation: 0.0542
    - Average Train Score: 0.8124
- Gradient Boosting:
    - Validation Score: 0.7243
    - Standard Deviation: 0.0522
    - Average Train Score: 0.8621
- eXtreme Gradient Boosting
    - Validation Score: 0.7297
    - Standard Deviation: 0.0634
    - Average Train Score: 0.9083

The top models are:

- **Extra-Random Forest:** It has the third largest validation score, but a much lower standard deviation the top scoring model.
- **eXtreme Gradient Boosting:** It is the top validation score
- **ElasticNet Regression:** It has much less variation then most models

Model Selection

The eXtreme Gradient Boosting model is our choice for predicting ratings, with having a Test Score of: 0.7305. This model explains around 73% of the variation in the model.

Looking at the feature's importances and the features' coefficients we get a better understanding of how features impact the price for a listing in San Francisco. Some interesting insights on the relationship with feature and ratings are:

- The value a customer feels, the accuracy of the listings, and the cleanliness of the listing are the top 3 most important factors. A host should focus on these three factors if they want to have the largest positive impact on their ratings.
- No amenities had strong importances, yet there was a few that can make an impact (coefficients from ElasticNet).
- Having a kitchen, Hot tub, and Extra pillows and blankets have the strongest relationship with higher ratings
- Location / Neighborhood does not have a strong effect on ratings.
- The longer a host has been a host on Airbnb, (host_since_days, first_review_days,numer_of_reviews) the higher their ratings tend to be. This may be due to the host learning from the experience and providing better service over time.
- More expensive listings tend to have higher reviews.