Capstone Project 1

San Francisco Airbnb Project Milestone Report

Shirley Zhu

Problem Statement

This proposal is to develop an Airbnb listing model by machine learning that predicts the price of a listing of a property based on the location, size, amenities, etc.

My client will be Airbnb and its hosts. This model will help the Airbnb hosts to have a better prediction of the money they will make based on the size, location, cleaning fee, ect of their listing, they can use this prediction to set a reasonable price, which will maximize the profit in the long term, because a price that's too high will lower the renter's satisfaction and a price that's too low will lower the owner's satisfaction. Since Airbnb charges flat 10% commission from hosts upon every booking done through the platform, this helps with the profit of Airbnb as well.

This model will also help the hosts to know how the price should change if they decide to raise or cut the cleaning fee, to rent the property as the entire house or apartment or just one room, raise or cut the fee charged to extra person, etc.

The data were downloaded from this website: http://insideairbnb.com/get-the-data.html
I will use the San Francisco Airbnb listing data in the csv format, and manipulate them in python.

I would like to use machine learning and part of my data as training dataset to train my model to predict the price of the listings, use cross validation and gridsearch to evaluate and fine tune the parameters, and use a test dataset to provide an unbiased evaluation of a final model fit on the training dataset.

Data Wrangling

1. Data cleaning steps

- I used pandas pd.read_csv() method on the urls of the csv files at the Inside Airbnb website and got an overview of the data frame by using df.head(), I then got an idea of what the data frame looks like, how many columns it includes and what they are.

- I decided which columns are of the most interest of this project. Columns like neighbourhood, price, room_type, minimum_nights, availability_365, etc. are critical factors in this project.
- Since location is a supposedly primary determining factor in any kind of housing market, I grouped the data frame by different neighbourhoods and used aggregate functions such as value_counts, mean, median to know the total number of listings, average price, median price of the listings in a specific neighbourhoods. By using df.plot(), I could visualize the neighbourhoods with the most listing, the most expensive or cheapest neighbourhoods.
- Other exploratory data analysis includes grouping the data by different room types or minimum nights requirements and use aggregate function to find if there is any trend or relationship between different parameters.
- I used pandas.concat() to merge data frames that I am most interested in and get a new dataframe, and used scatter plot to see if the availability of a listing is dependent on the price of the listing.

2. There is no missing data for the data I collected so far. I will fill it with a NaN if there is any missing values.

3. Outliers - There were outliers in this dataset.

- In some neighbourhoods, there are only one or just a few listings. These data are still good for Exploratory Data Analysis, but they are not representative to study the trend of that area, or be usable for machine learning.
- In the minimum_nights column, there is a host that puts a minimum nights of 1 billion days, and a few others put over a thousand days. This caused the scale of the plot very hard to read, and these data are outside of 99% percentile. These outliers should not be included in the machine learning datasets.
- In the price column, there are a couple of listings with prices that are above $10,000. These data are also outside of the 99.9% percentile and make the plot very hard to visualize other useful information other than these data points. These outliers should not be included in the machine learning datasets.

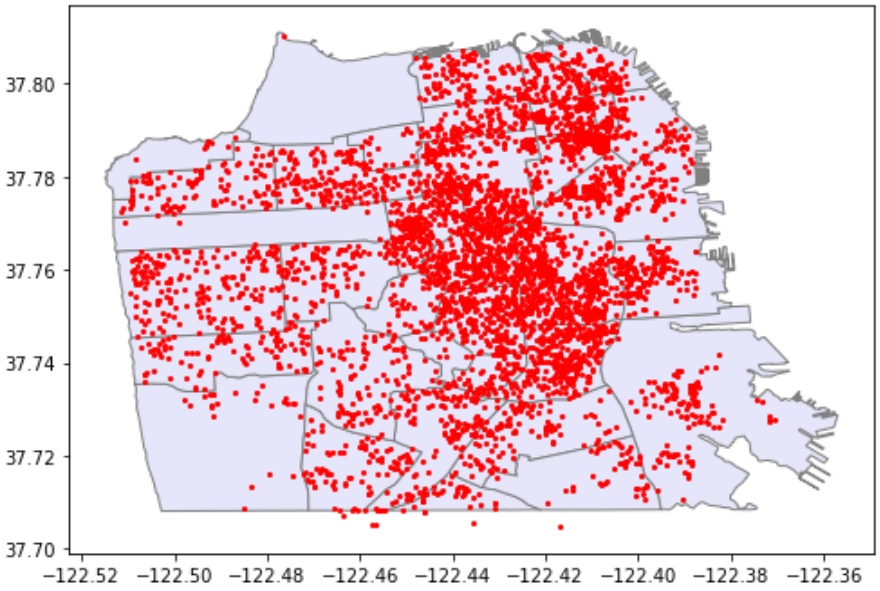
Exploratory Data Analysis

Methods:

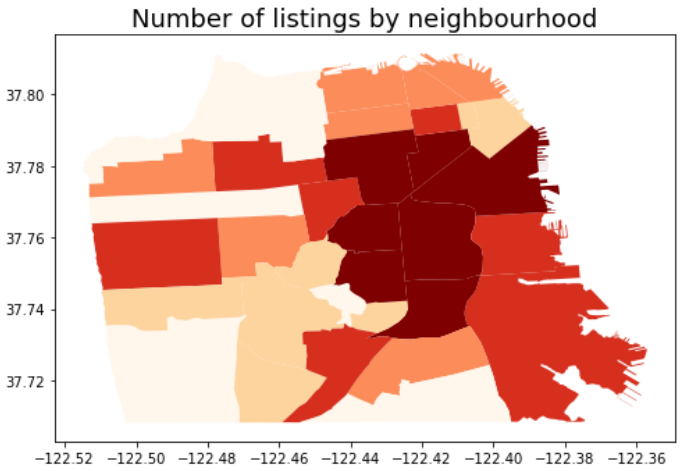
1. Load the data with geometry information of the neighbourhoods in San Francisco and create a GeoDataFrame from geodata.
2. Visualize all the listings in a map of San Francisco and visualize the number, average price, and popularity of listings by using color maps of different neighbourhoods.
3. Create a scatter plot of price vs. availability of different neighbourhoods and use bubble size to indicate the number of listings in each neighbourhood.
4. Define a function that returns the empirical cumulative distribution function (ECDF) which orders the data from smallest to largest in value and calculates the sum of the assigned probabilities up to that data point. Use histogram, boxplot and the ECDF to visualize the distribution of several features including minimum_nights, price, and reviews per month.
5. Use seaborn boxplot to visualize the relationship between number of bedrooms and price, minimum nights and popularity, price and minimum nights, etc. to find if there is a trend between these features visually.
6. Explore the relationships between cancellation policy and the price or the popularity by barplots.
7. Use bar plots to show the price and popularity of different property types.
8. Visualize the change of airbnb bookings reflected in the number of reviews by year, month and week.
9. Find the most used descriptive words in the names of the most popular listings (most reviewed) and the most expensive listings.

Code: the code of EDA and Inferential statistical analysis was in a jupyter notebook “SF Airbnb - Data Story.ipynb.

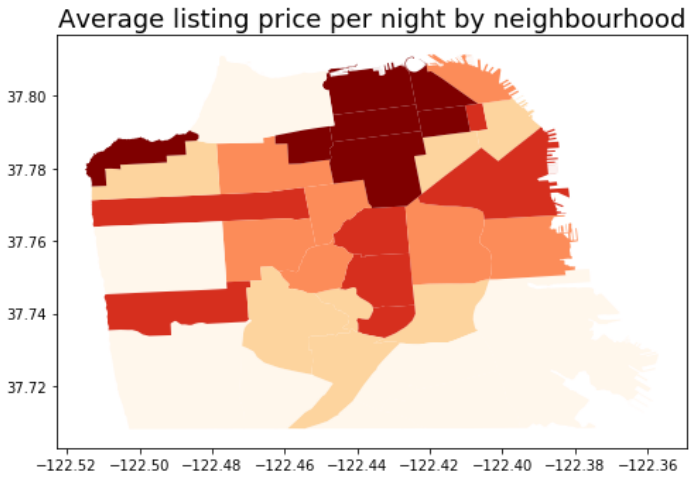
Summary of Findings:



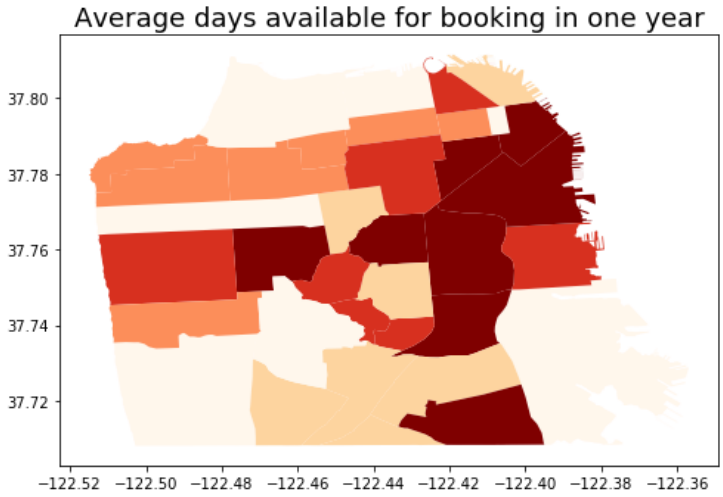
The map above shows the location of the Airbnb units in San Francisco. As the maps above show, Airbnb units appear to be concentrated in the northern and eastern neighborhoods of the city.



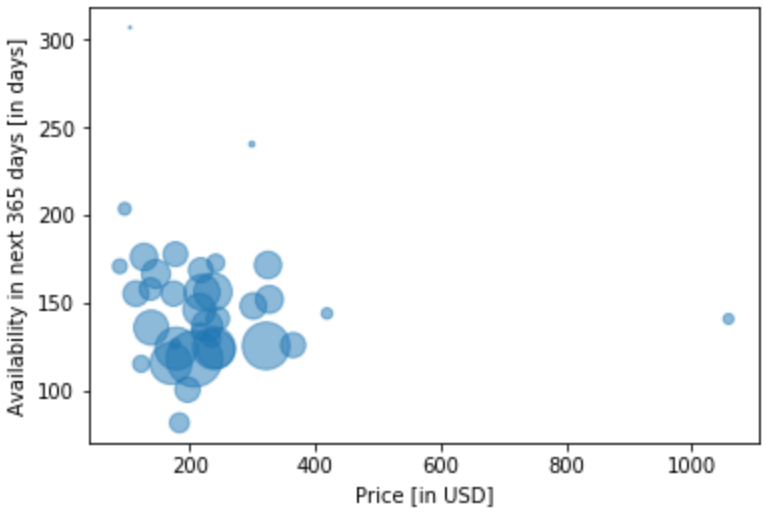
This map above demonstrates that most listings are concentrated in core neighborhoods that are close to Market Street.



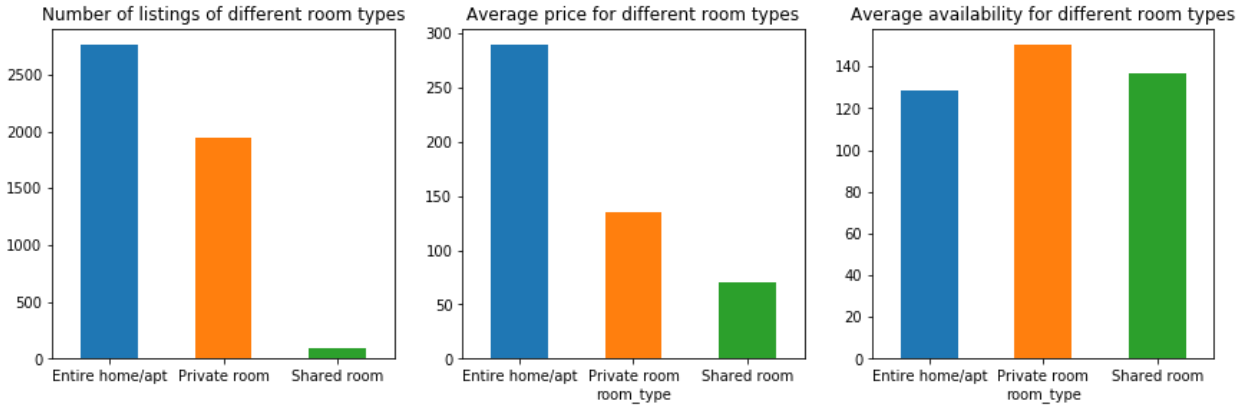
This map shows that most expensive listings are concentrated in north of Market Street and Seacliff area.



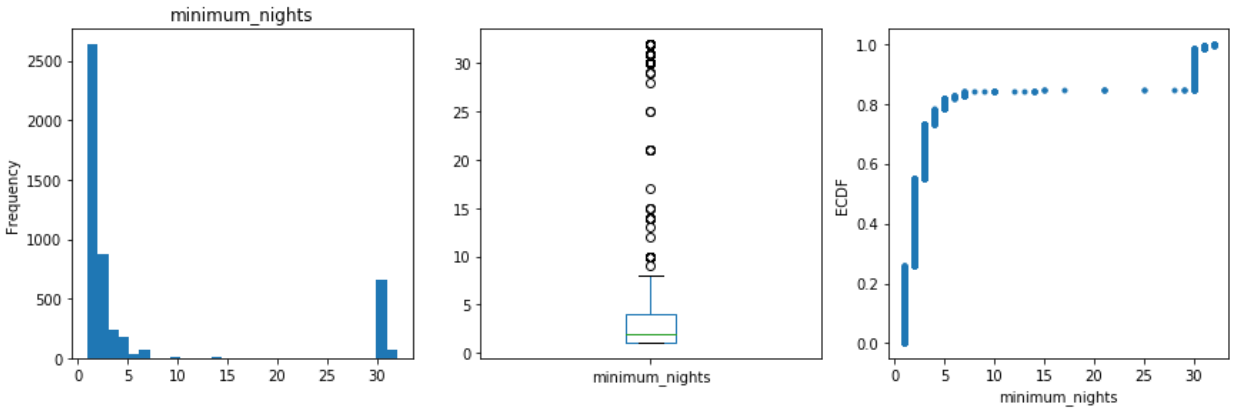
This map shows that the most popular neighbourhoods are those areas relatively cheap, but still close to the core neighbourhoods around Market Street. The most expensive neighbourhoods are relatively less desirable for a home stay with Airbnb.



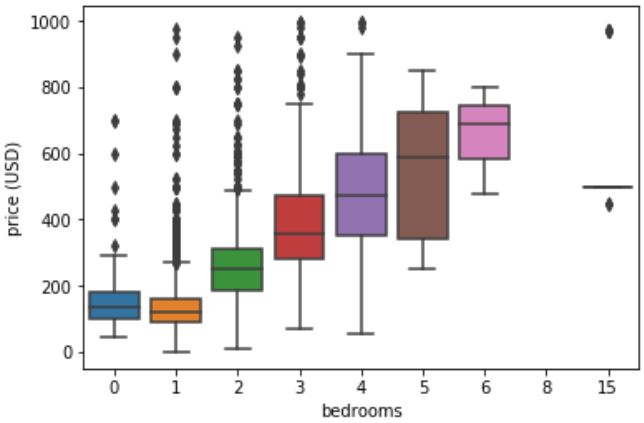
The plot above does not show a very clear trend between price and availability. There is a slight downward trend, meaning the neighbourhood with higher-end rentals are relatively more popular.



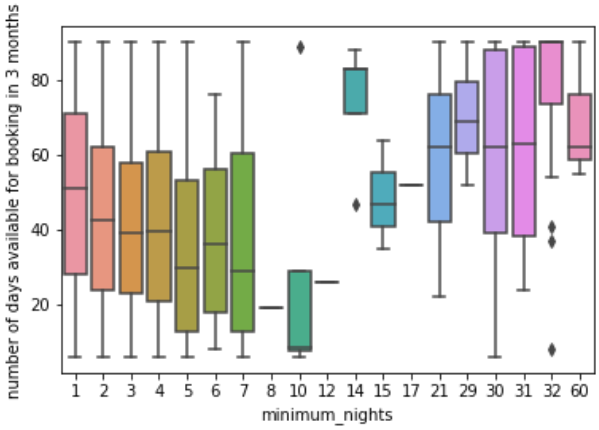
The plots above show that most people rent the entire home/apt out, and less people rent one of the rooms in their houses, and very few people share their rooms with strangers. The entire home types are most expensive, followed by private room and the shared rooms are the cheapest. The popularity are quite similar among different room types.



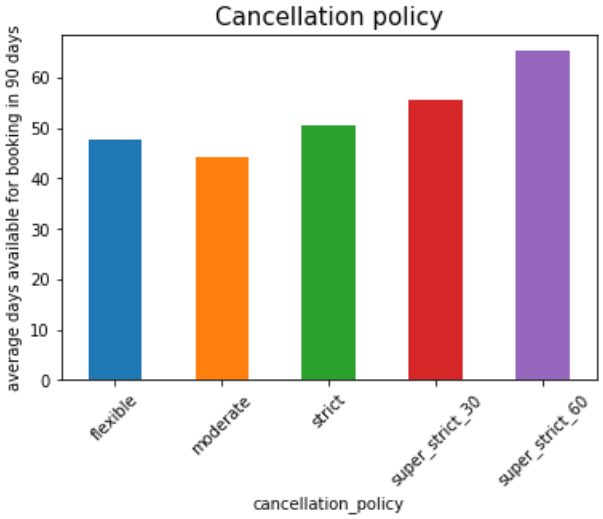
The above plots show that most people did not require a minimum nights more than one night. And there are over 500 listings have requirements of a minimum night of 30 days. The listings with more than 30 day minimum night requirements are outliers.¶



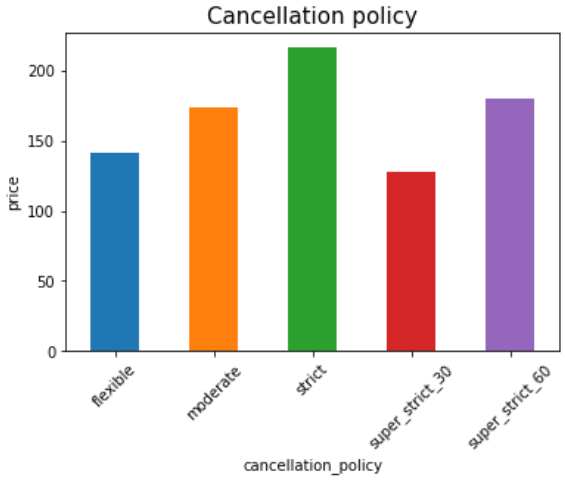
The plot above shows that the more beds of a listing, the more expensive. There are only one listing with 15 bedrooms. There is not much difference between a one bedroom apt and a studio.



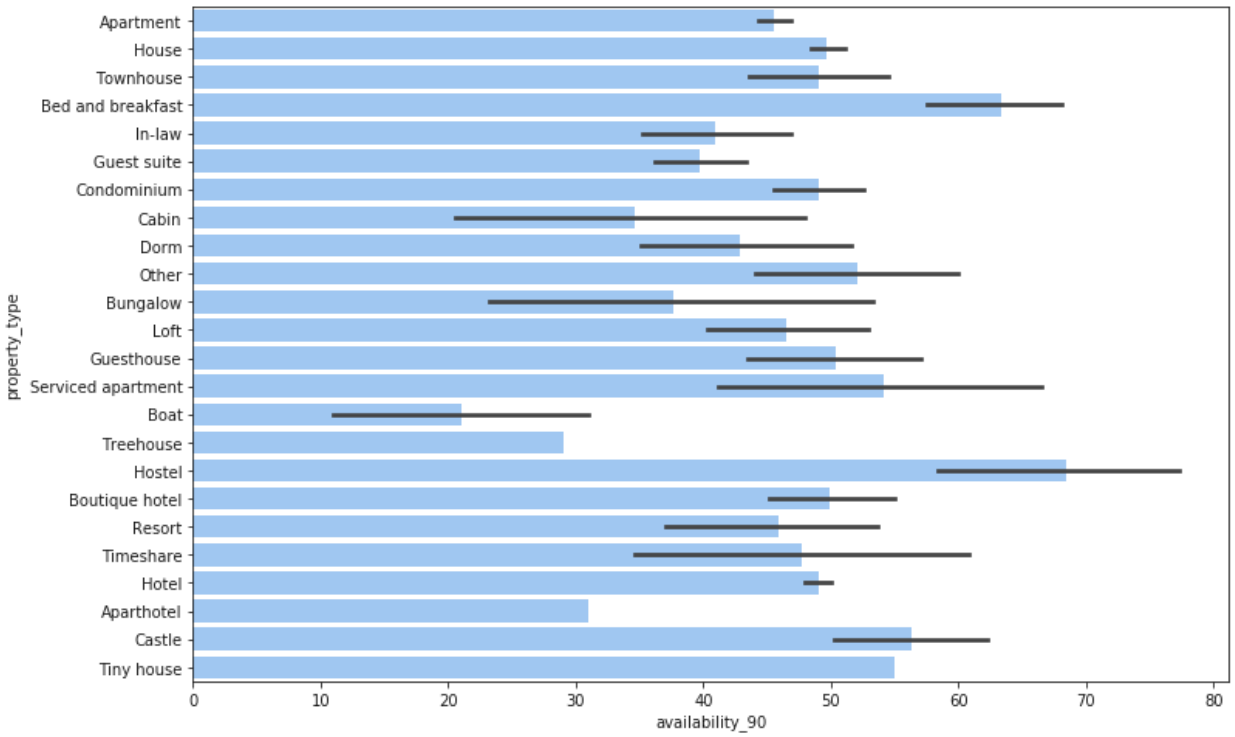
The plot above shows that listings with a minimum of more than 10 nights stay requirement are relatively less popular than those with less than 10 nights minimum stay. And among the listings with a less than a week of minimum nights stay requirement, the more minimum nights the host requires the guests to stay, the more popular the listing is.



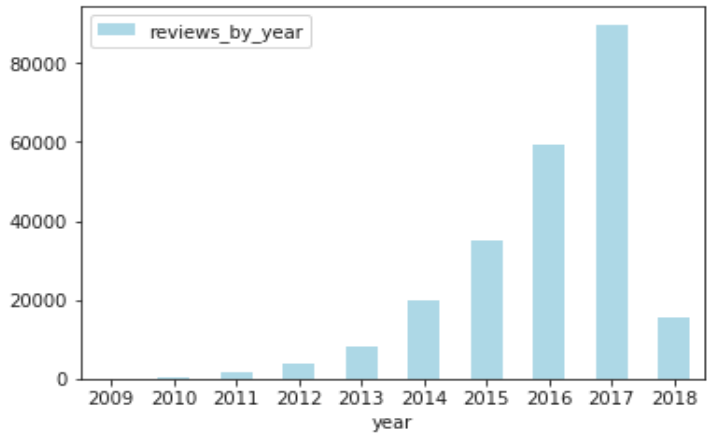
This plot shows that a flexible or strict cancellation policy does not greatly affect the popularity of a listing. A stricter cancellation policy decrease the booking rate a little bit. And setting a moderate cancellation policy helps with the booking rate.



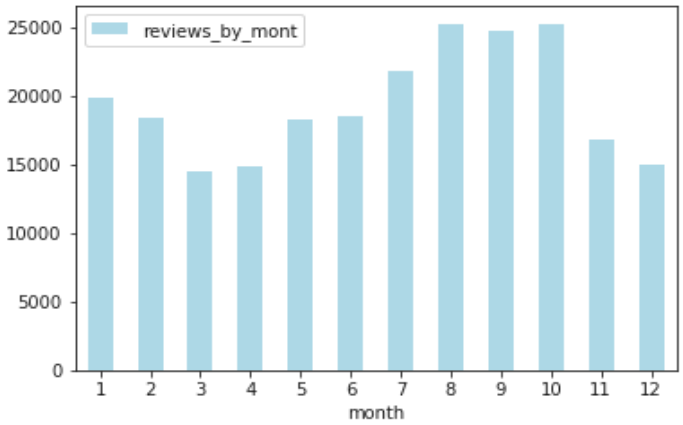
The plot doesn't show a general trend, but among the flexible, moderate, and strict categories, the higher the price, the more strict of the cancellation policy.



The bar plot shows that the most popular property types are boat, treehouse, cabin and bungalow, which means unique and fancy properties are very attractive and having high occupancy rates.



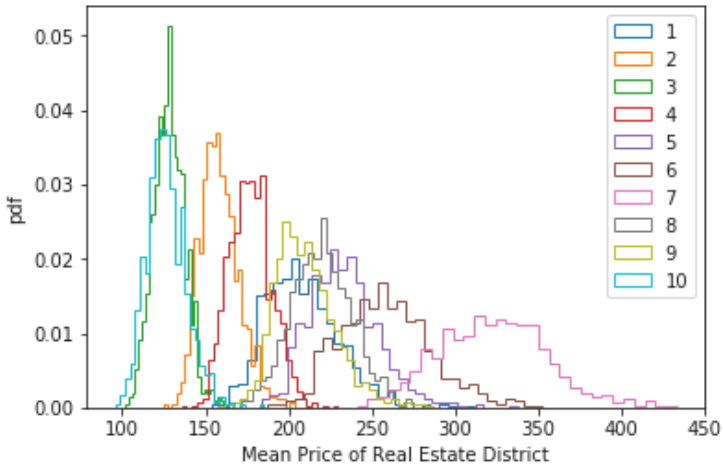
Airbnb stated in 2012 that 72% of guests leave reviews. So the reviews by year can represent the Airbnb stays increased a lot each year. This plot above shows the stays look like an exponential increase over the recent years, but still needs some regression to confirm that. When the data was scraped, it was still March in 2018.



The plot above shows the seasonality of airbnb stays. The peak months are August, September and October, and the low seasons are March, April, November and December.

Inferential Statistical Analysis

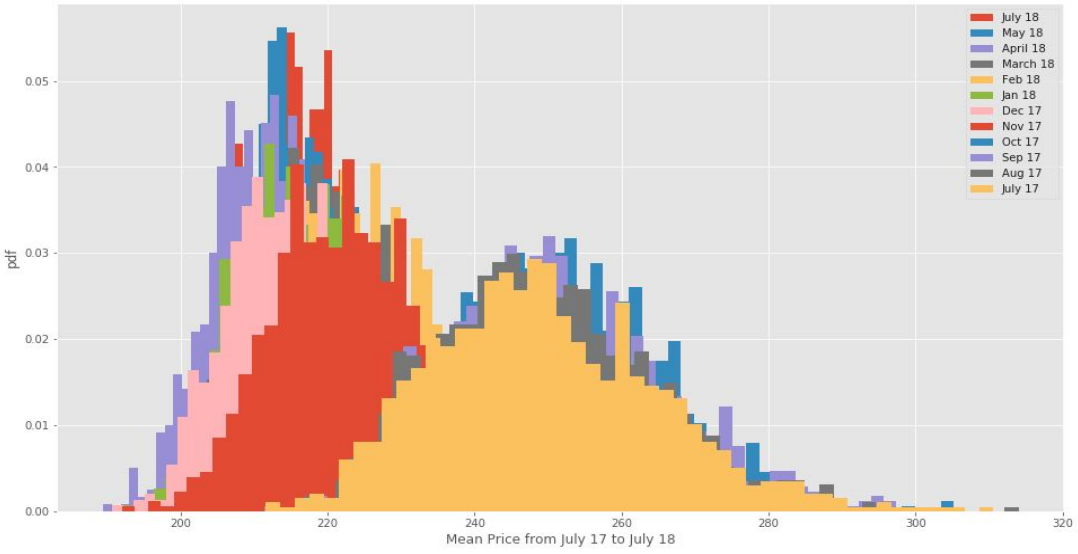
Some preliminary analysis shows that the price in each district is not normally distributed. However, according to Central Limit Theorem, the mean of the price in each district should be normally distributed. We can do ANOVA to test the difference of the means. ANOVA test is a generalization of the t-tests to more than 2 groups, which assumes that 1)the samples are independent; 2) each sample is from a normally distributed population in order for the p-value to be valid. And bootstrapping resampling was used to test the difference in mean price of each governmental district.

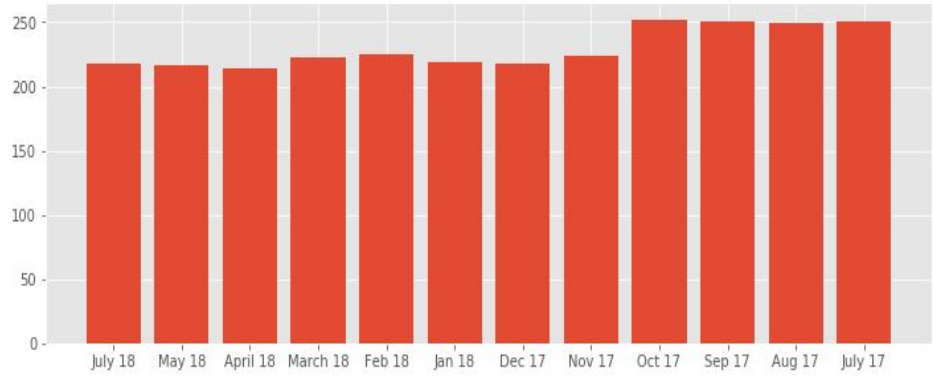


The above figure shows that District 10 (Outer Mission, Croker Amazon, Bay View, etc) and 3 (Ocean View and Lake Shore) have the lowest and second lowest prices, each of them has a mean price that is significantly lower than the rest of the districts, but they

are not quite different from each other. District 7 (Marina, Pacific Heights, Presidio, etc) has the highest price, statistically significantly higher than the rest.

Inferential statistical analysis was also used to evaluate the change in price with time. The one-way ANOVA tested the null hypothesis that airbnb listings in San Francisco have the same mean price each month from July 17 to July 18. The extremely low p-value indicated that the null hypothesis should be rejected, so there is some change over this year that is statistically significant (α=0.05). And bootstrapping resampling was also used to test the difference in mean price of each month.





As shown in the graphs, there are two groups of mean prices, there is no significant within each group. The mean prices of all listings in SF actually decreased from Oct 2017 to Nov 2017. Group 1 is July 2017 to Oc 2017, Group 2 is Nov 2017 to July 2018. The mean price does not significantly change since November 2017. The mean price decreased from about 250 USD to about 218 USD.

In-Depth Machine Learning Analysis

Data Preprocessing:

The code of the in-depth machine learning analysis was in a jupyter notebook file “SF Airbnb In-Depth Machine Learning Analysis.ipynb”.

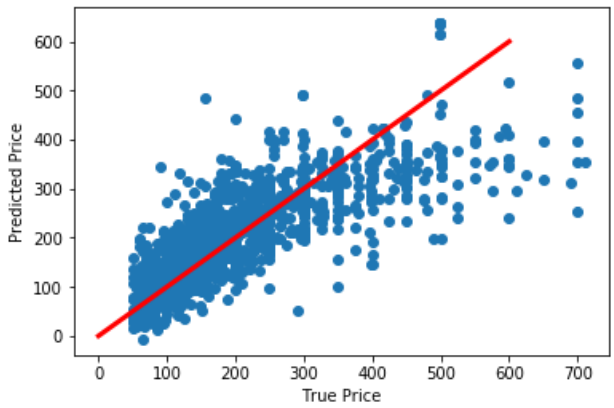
Steps taken to clean the data were different from the steps taken for the EDA. These are the data preprocessing steps that I took to prepare the data from machine learning:

1. Read SF listings from March 2018 csv file to a Pandas DataFrame.
2. Drop non-relevant, redundant, empty or not useful columns.
3. Convert the data types of some columns from “object” to “category”.
4. Convert the data types of some binary categorical columns further from “category” to “numerical” (boolean).
5. Convert the data types of some other columns from “object” to numerical by replace the special signs (“$”, “,”, etc) and change the dtype to “float”.
6. Replace the “NaN” in some columns with 0s or 1s, depending on the variables.
7. Add the “districts” column to the dataframe from another csv file to lower the number of different geographic locations of the listings.
8. Convert the categorical columns to numerical by using the get_dummies method.
9. Cap some columns’ extreme values with the 99% percentile.
10. Define the predictor variables and target variables. Only numerical features were used for now.

Supervised Learning:

There are 4557 observations and 81 features that are available. Train_test_split() was used to split the dataset into 70% training set and 30% test test.

1. Linear Regression. A simple linear regression model was created to fit the data. The R^2 for the training and test sets are 0.64 and 0.62 respectively, and the root mean squared error is $76.15 for the prediction of price. This figure shows that linear regression tends to underestimate the price for the expensive listings.



2. Lasso Regression. A simple Lasso regression was tried first to regularize the regression by selecting the most important features. The three features selected by lasso regression coefficient lasso.coef_ are “accommodates”, “cleaning_fee”, and “room_type_Entire home/apt”. The R^2 for both the training and test sets are low, 0.40 and 0.42 respectively, and the RMSE is high, $93.93.

   However, after hyperparameter tuning using GridSearchCV(), the R2 was increased to about 0.63 for the training and test set, and the RMSE was lowered to $75.19. The best alpha given by GridSearchCV() is 0.01 within the alpha space np.logspace(-2, 2, num=20).

3. Ridge Regression. Ridge regression would include every feature in the dataset while regularizing the parameters. Using GridSearchCV, the R^2 for both the training and test sets are are similar to lasso regression, which are 0.64 and 0.63 respectively, and the RMSE is high, $75.38.

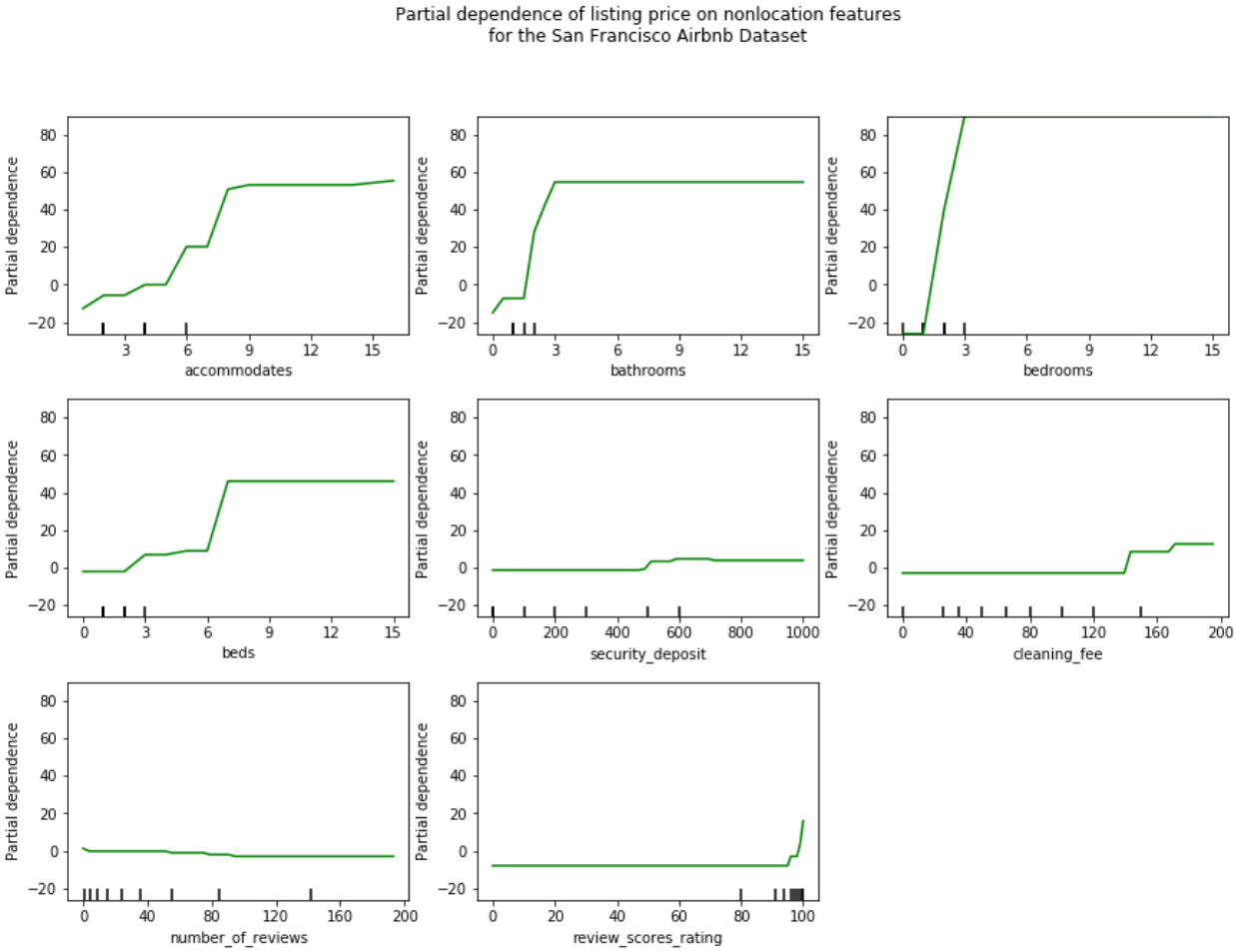
4. ElasticNet. ElasticNet is also linear regression with combined L1 and L2 penalty as regularizers. Here I also used GridSearchCV to tune the hyperparameters alpha and l1_ratio to get the best hyperparameters. The best l1_ratio is 1, which means that it is better to use a lasso regression solely instead of a ridge regression. And the RMSE for the ElasticNet regression here is exactly the same with the Lasso result. This indicated that a lasso regression is slightly better in predicting the price than a ridge regression here.

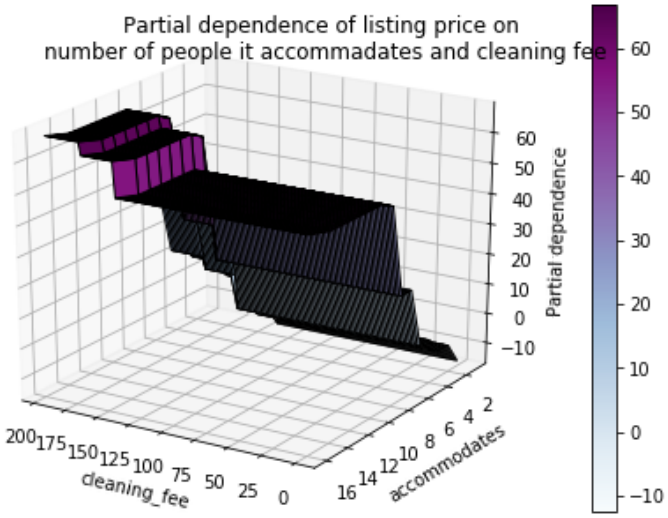
5. Gradient Boosting Machine. Unlike the four algorithms tested previously, GBM does not assume a linear relationship between predictor and target variables. This is a regression problem, so GradientBoostingRegressor() was used. Since there are many hyperparameters to tune, a RandomizedSearchCV() method was used instead of the GridSearchCV() to make the search faster. The

hyperparameters tuned include learning_rate, max_depth, min_samples_leaf, max_features, and subsample. Also, early stopping was used by specify the parameters including validation_fraction, n_iter_no_change and tol. This asks the regressor to set aside 20% of the training data as validation set for early stopping, and stop fitting if the scores for the set-aside data don't improve by at least 0.001 for the previous 5 iterations.

The R^2 for the training and test set are 0.80 and 0.71 respectively, and the RMSE is $66.83. This result is much better than all the linear regression models tested. With early stopping, the number of boosting stages performed is 47 versus much higher boosting stages without early stopping.

Partial Dependence Plots were plotted using the fitting result with the best parameters.



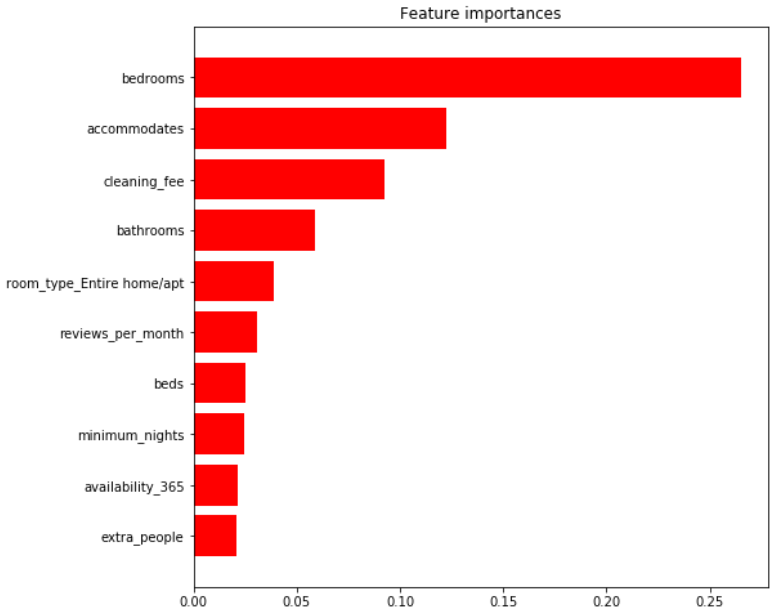


The above Partial Dependence Plots shows the listing price is in linear relationship with number of people it accommodates, bathrooms, bedrooms, security deposit, cleaning fee within a certain range of these variables. The listing price does not change much with the number of reviews much. And the listing price increases with review rating score. The two-way partial dependence plot shows the dependence of listing price on joint values of # accommodates and cleaning fee.

The Recursive Feature Elimination (RFE) method was used to rank the importance of all the features and select the most important features. It works by recursively removing attributes and building a model on those attributes that remain. The Recursive Feature Elimination selected three important features and ranked them according to their importance. RFE selected "accommodates" (the number of people it accommodates), "bathrooms", and "beds" as the three most important features to predict the price of the listing. Other important features include reviews per month, room type, cleaning_fee, guests_included, minimum nights, available days in the next two months, etc.

6. Random Forest Regression. This is another ensemble method like GBM. RandomizedSearchCV() was also used to search the best hyperparameters. The hyperparameters tuned include n_estimators, max_depth, min_samples_leaf, min_samples_split, max_feature. The R^2 for the training set is very high 0.92, but the R^2 for the test set is 0.70, slightly lower than the result got from GBM. And the RMSE $ 67.68 is higher than the sult got from GBM too, which is about $66.83.

Feature importance was also analyzed for the random forest regression result.



As seen from the above plot, the three most important features the random forest regression selected are bedroom, accommodates, and cleaning fee, which are slightly different from what were selected by other algorithms. Note that the bedrooms and number of people the house accommodates are highly correlated features.

In summary, in the in-depth machine learning analysis, I tested simple linear regression, lasso regression, ridge regression, gradient boosting machines and random forest to predict the listing price of SF Airbnb listing. GridSearchCV or RandomizedSearchCV were used to fine tuning the hyperparameters. So far, gradient boosting machines got the best result, 0.71 for the testing set R^2 and the lowest RMSE $66.48 with early stopping.

Conclusion

It is possible to use supervised machine learning to predict the Airbnb listing prices based on features such as the number of bedrooms, number of people the house can accommodate, how much the cleaning fee is, how much an extra people should pay, etc. There would be errors of course, but we can use the machine learning predicted price as a guidance for the owners to set the price for their listings. A reasonable listing price is important. If it is set too high or too low based on the size of one's house, the cleaning fee, or the room type, either the owner would not have the best return, or the customer would not think the stay is worth the money and leave bad review. A good business strategy for Airbnb would target at the double-win of both the owners and renters, so a machine learning predictive pricing guidance would be beneficial to all parties.