Chunyu Chen
Zongjie Yin

# Data Analysis For Airbnb

## Background

As you may already know, Airbnb is a widely-used website and app that serves as a platform for a two-sided market, catering to both landlords and renters. Landlords can use the platform to lease their vacant houses, while renters have the option of choosing Airbnb accommodations for their travels or when hanging out with friends, in addition to more traditional options.

## Purpose

The purpose of our project is to predict the daily price of a given house in New York City listed on Airbnb, based on provided information such as the number of bedrooms and bathrooms, location, and square footage of the house. Our aim is to maximize benefits for both landlords and renters by offering a relative price range for similar houses, allowing renters to have a wider range of affordable and higher-quality housing options, while also enabling landlords to earn more income in a shorter period of time.

## Data

The data used in our project was sourced from the Kaggle website at https://www.kaggle.com/c/airbnblala/data. The dataset contains 29,142 rows and 96 columns of data, with each row representing a single house listed on Airbnb in New York City. The columns contain three main types of information: basic housing information such as location and URL links, household information such as response times and listing house counts, and review scores such as overall ratings, accuracy, and cleanliness of the houses.

## Data Cleanning

First, we might want to see if there is any duplicated value in our data set. Luckily, we did not have any two houses with the same id, so we left all the data for our following analysis.

Aside from checking the duplicated data, we might want to see if there was any outlier in our data set. We found that there were a few rows of data which prices are less than 30 dollars while some are more than 500 dollars. After we checked the Airbnb website, it looked like those are actually not outliers because either it is a rudimentary house or it is a luxury house. Therefore, our team chose to leave those data. However, there are some rows of data which have prices with 0, which is not in a reasonable range of house price. After checking those houses in the dataset, we found 25 rows of data where the house price is listed as 0 dollars. Since our dataset contains 29,142 rows, our team chose to delete those 25 rows.

Then, we looked at the number of bedrooms and the number of bathrooms in the house, we found that there are thirty houses with 0 bedrooms but more than one bathroom. Since it makes no sense for a house with no bedroom but more than 1 bathroom, we chose to delete those thirty rows with unreasonable data.

After handling the outliers, we figured out that there were still many missing values in the dataset. Usually, there are several ways to deal with them: 1) delete the column if it is not informative; 2) delete the rows with missing values if they won't affect the results compared to the number of data points; 3) fill the missing values using the median, mean, or other relative methods.

Our team first filtered out the columns with over 50% missing values and found ten columns, including two price-related columns and eight uninformative columns such as URL links and house pictures. The response variable is the house price, so to predict it, the dataset should not include price-related columns such as weekly and monthly prices. Therefore, we deleted those ten columns.

Even after deleting columns with over 50% missing values, there were still some missing values in the dataset. The second important aspect of missing values is the location of houses. Since the zipcode is more precise than the city name, and the number of missing values for zipcode is lower than that of city name, our team used the zipcode to locate the city for each given house. This not only filled in the missing values but also corrected any misunderstandings resulting from city names in other languages.

Some of the other columns with missing values in our data set are either not important like the name of the house or redundant like the host location and host-about columns. Besides, there are some features with missing values that there is no way we can fill in like the summary, space, description, neighborhood overview, transit, access, interaction, house rules, host response time, host response rate, host neighborhood, and the neighborhood. For the state and market columns, we filled in with "NY" and "New York" since all of the houses are in New York City.

Finally, we filled in the beds column by making use of the mean value of the number of beds of the known houses with the same number of bedrooms and the room type because we think that they are highly related. For the security deposit column, we filled in the missing values by using the average security deposit of the known houses with the same location since we think the security deposit is related to its location. For the cleaning fee column, we filled in the missing values by using the mean cleaning fee of the known houses with the same number of bedrooms and bathrooms. The larger the houses, the higher the cleaning fee is.

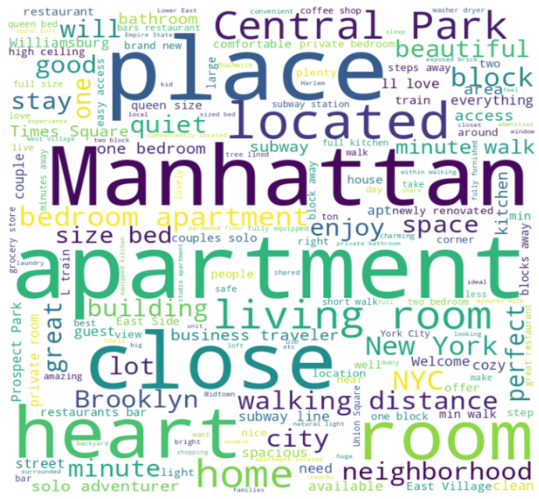
## Data Visualization

### Word Could:

After cleaning the dataset, there are a total of eighty-six columns. Thirty-two of these columns contain numerical variables, while the remaining fifty-four columns are non-numerical variables, such as long text, ordinal data, and nominal data. Thus, data visualizations of non-numerical variables are also important for investigating further information on house prices.

The most interesting plot for non-numerical variables is the word cloud. The size of words in the word cloud plot depends on their frequency in the given column. Therefore, important information is emphasized by larger word sizes. Below are the word clouds of the Summary column and Transit column in the dataset. Both plots have "Manhattan" in the middle, which seems to be the largest word in both plots. As readers look at these two plots, words related to location are emphasized multiple times. From these two plots, our team has decided to keep the location of the house as a factor for data analysis.

Summary Column




Transit Column


Our team also did word could for the amenity column and house rule column. The amenities column mentioned different features that are important for renters such as air conditioning, wifi… Also, the house rules column shows that smoking is one of the features that many landlords care about. Thus, all of four columns should be included in the data analysis process.

Chunyu Chen
Zongjie Yin

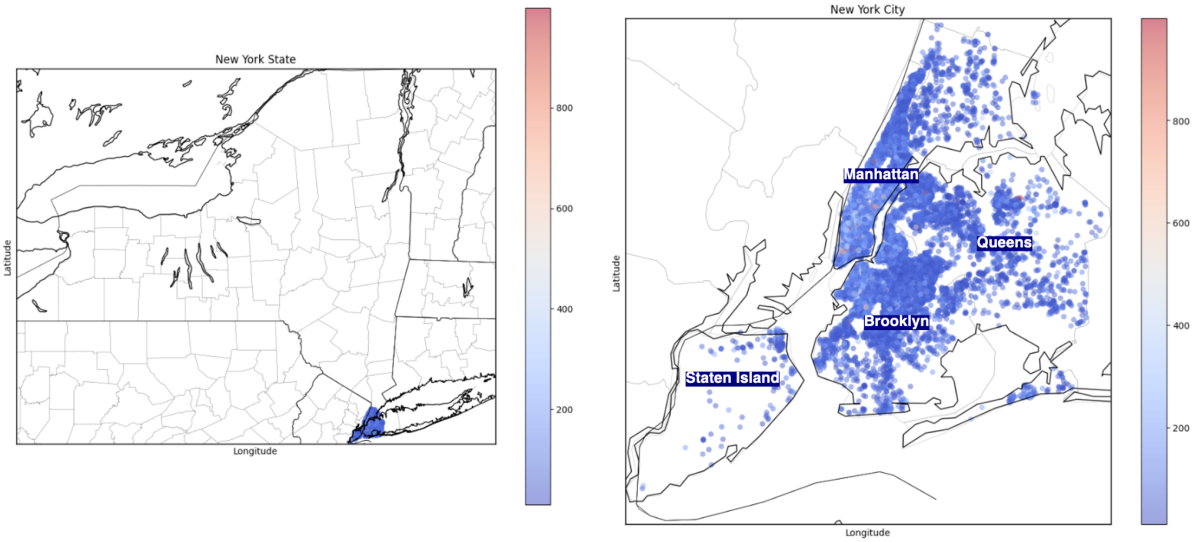
Amenities Column


House rules Column


**BaseMap:**

At the beginning, we drew the New York State map with a color bar indicating the prices, hoping to see the geographic and price distribution of Airbnb listings. However, it seemed that all of the houses were located in New York City. Therefore, we zoomed in and drew a map of New York City. We found that the houses were distributed around the five boroughs of New York City: Manhattan, Queens, Brooklyn, Bronx, and Staten Island, with the majority located in Queens, Manhattan, and Brooklyn. In terms of price distribution, most of the houses were priced below $200, while some in Manhattan were over $600, as shown on the color bar. This is reasonable since Manhattan is the most prosperous place in New York City, as always.

Price V.S. Location of House

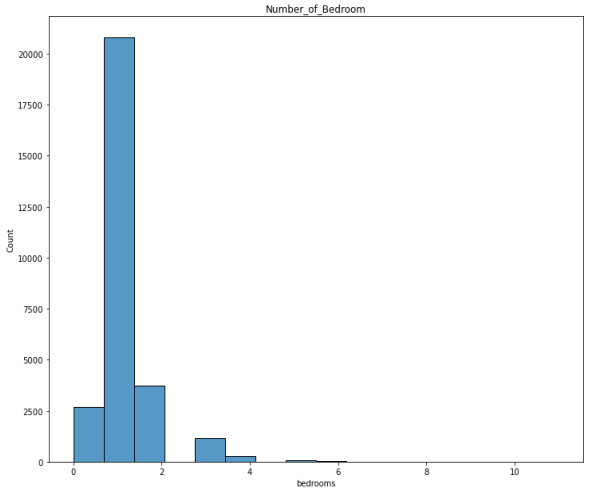


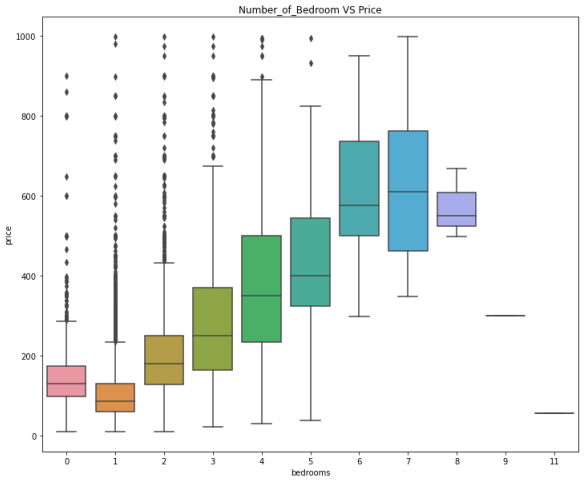

**Boxplot and Histogram**

**Number of Bedrooms**

We first examined the number of bedrooms within the houses using a histogram, and then looked at the relationship between the number of bedrooms and the price of the houses using a box plot. We found that most of the houses had one bedroom, which makes sense given the high cost of land in New York. Looking at the box plot, we observed that as the number of bedrooms increased,

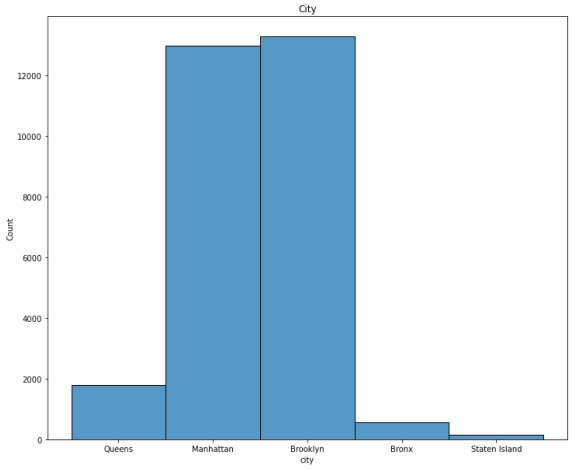
the median price of the houses also increased, except in the case where the number of bedrooms was 0, which could be explained as a studio apartment.

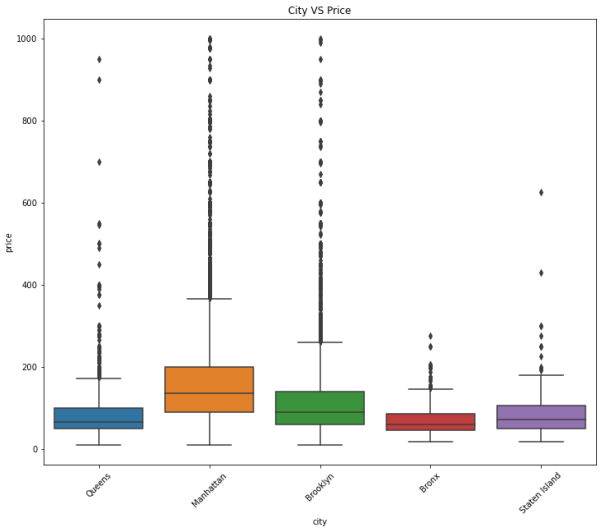



## City

We also examined the geographic distribution of the houses using a histogram and the distribution of prices using a box plot. We found that the highest number of houses listed on Airbnb were in Manhattan and Brooklyn, with Queens following behind. Regarding the prices, we observed that the median price of houses in Manhattan was significantly higher than in Brooklyn, Staten Island, and Queens. This was not surprising since Manhattan is the most prosperous borough in New York City.

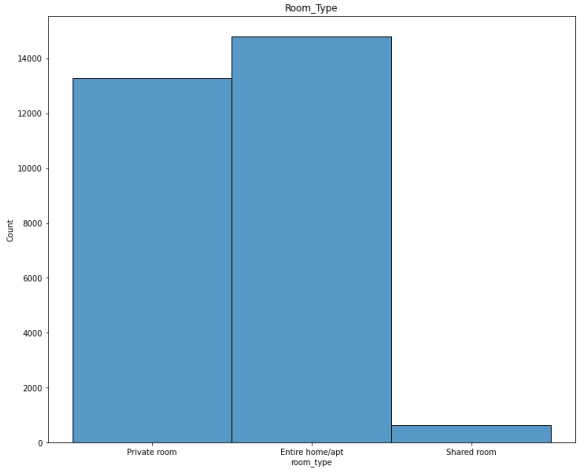


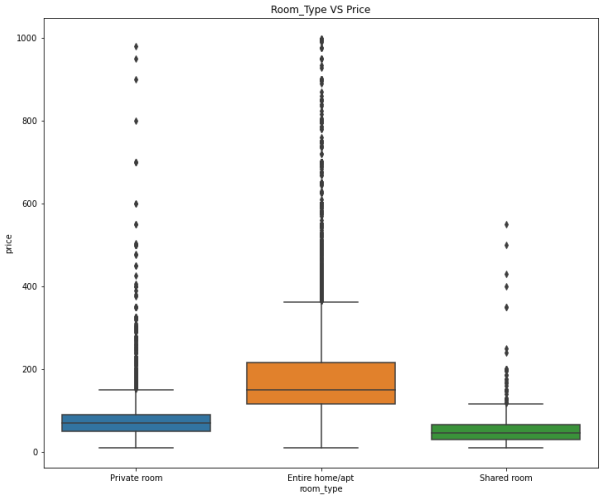




## Room Type

When examining the histogram of room types, we found that most of the rooms listed on Airbnb were either entire rooms or private rooms. Furthermore, when looking at the box plot of room types, we observed that the median price of houses with private rooms was higher than that of entire homes/apartments or shared rooms.

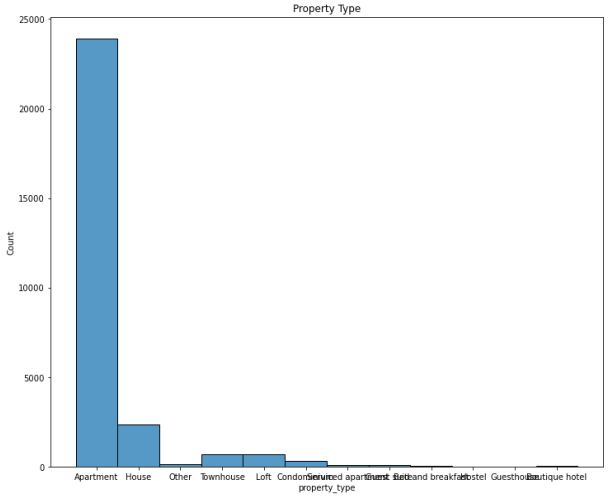
Chunyu Chen
Zongjie Yin

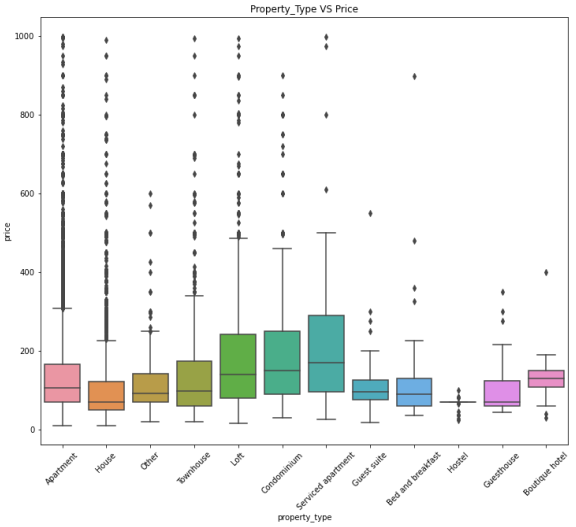



## Property Type

When examining the histogram of property types, we found that most of the listings on Airbnb were apartments. Furthermore, when looking at the box plot of property types, we observed that the median price of serviced apartments was higher than other types of properties. This is reasonable since the price of a serviced apartment includes services that other types of properties may not have.

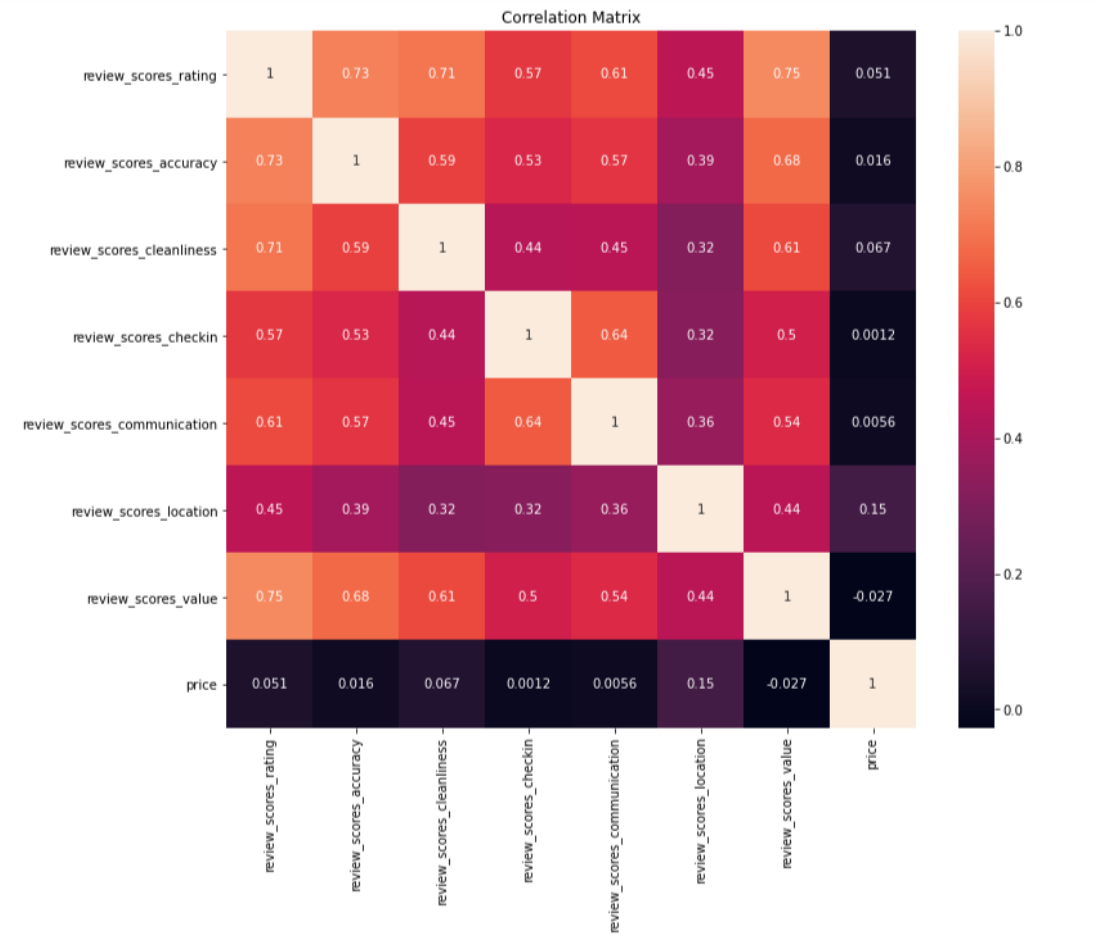



## Correlation Matrix

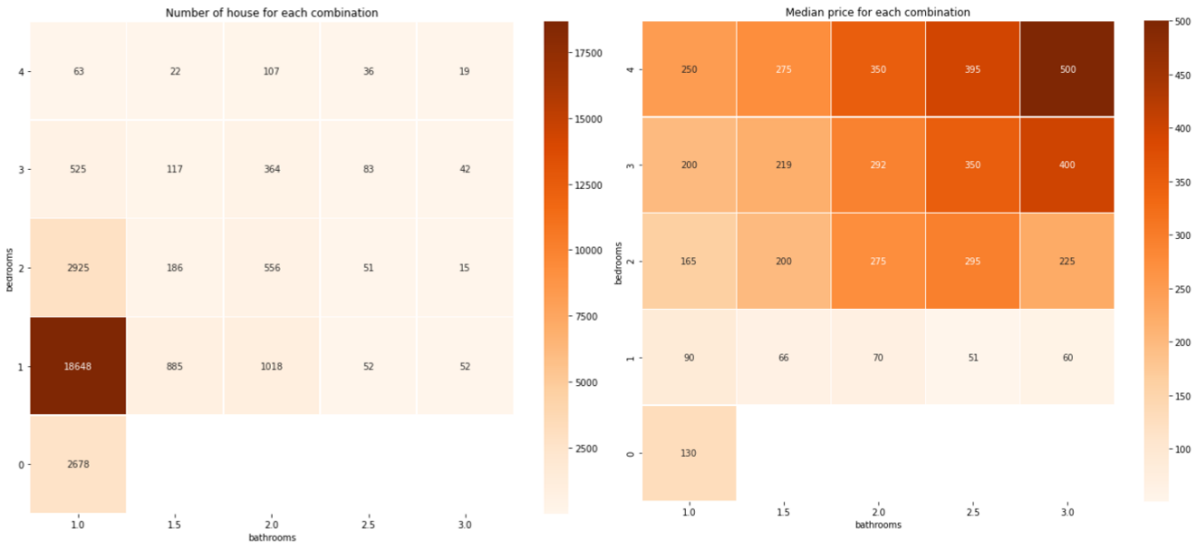
To investigate the correlation between review score and house price, we created a correlation matrix. We found that most of the review score columns had a strong positive correlation coefficient of 0.5. Including the price column, we observed that review score location had the highest correlation coefficient of 0.15, which is not surprising given our previous findings in the box plot. However, we were surprised to find a negative correlation coefficient between the review score value and the price of the house, as higher review scores usually indicate higher prices.

Chunyu Chen
Zongjie Yin




**Heatmap**

We created two heatmaps to examine the number of houses and the median prices for each combination of the number of bedrooms and bathrooms in the houses. In the left heatmap, we observed that the most common setting was 1 bedroom and 1 bathroom, followed by studio apartments (0 bedroom with 1 bathroom) and 2 bedroom with 1 bathroom setups. On the other hand, in the right heatmap, we observed that the median price of houses increased with the number of bedrooms and bathrooms, which is not surprising and makes perfect sense.




**Statistical Methods and Analysis**

Chunyu Chen
Zongjie Yin

**Model Set Up**

In the machine learning section of our project, we compared the results across different models. To start, we divided our dataset into an 80% training set and a 20% test set. We used mean squared error (MSE) to evaluate the performance of each model, including train MSE and test MSE. To ensure accurate results and meaningful model comparisons, each model was run at least 300 times to obtain the mean train MSE and test MSE.

**Linear Regression Model**

Since the linear regression model is good for interpretation, our team used it as the first model to apply on the Airbnb dataset. In order to prevent overfitting, we chose to include variables as matrices when applying the linear regression model. Initially, we only included numerical columns in the model, which contained a total of thirty-two columns. The training mean squared error (MSE) was 5494.708 and the test MSE was 6003.124.

Additionally, we tested the null hypothesis of each column, which stated that each variable associated with the given column has no relation with the y variable (house price). The report indicated that six columns had a p-value over 0.05. A p-value over 0.05 means the variable is not statistically significant, and there is not enough evidence to reject the null hypothesis. Therefore, these six numerical variables were excluded from the linear regression model. After removing these columns, the training MSE was 5496.487 and the test MSE was 5996.889, which was lower than the original model.

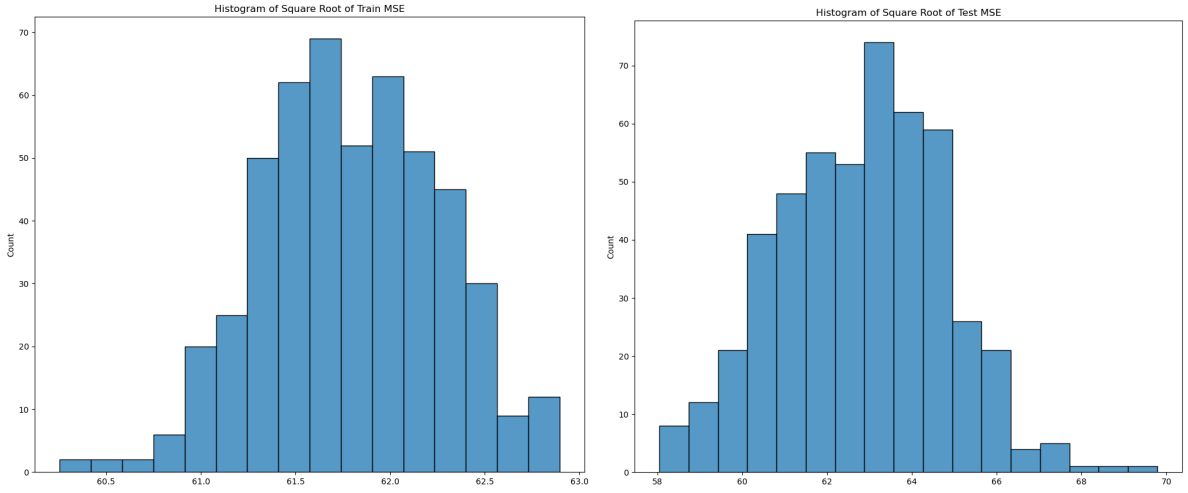
Afterwards, our team converted boolean variables from true and false to 1 and 0 as a vector in the matrix. The training MSE was 5461.320 and the test MSE was 5968.068. Four columns had P-values over 0.05 and were deemed not statistically significant. After deleting these four columns, the training MSE was 5462.479 and the test MSE was 5968.241.

For categorical data, our team used one-hot encoding to convert the data into several vectors with 1 as an indicator for true and 0 as false for one category. There were four categorical data in total: property_type, room_type, bed_type, cancellation_policy, and host_response_time. After including these four features, the training MSE was 4836.958 and the test MSE was 5352.153, which significantly improved the model's prediction.

Multi-categorical data was converted into several categories vectors with 1 as an indicator for true and 0 as false for one category using many-hot encoding. The host_verifications column in the dataset was a set data column that was used in this encoding and included in the linear regression model. The training MSE was 4830.112, and the test MSE was 5344.259.

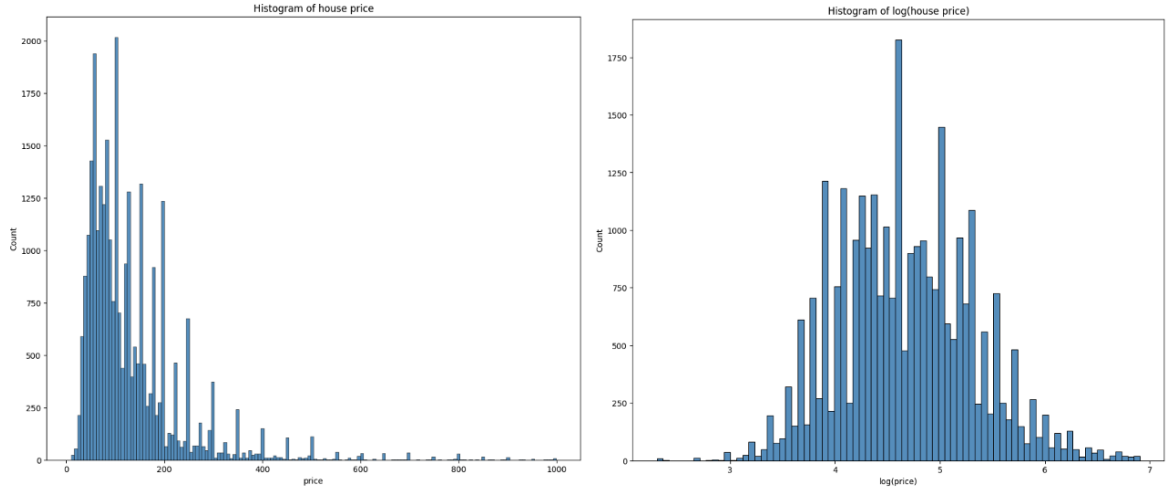
Lastly, location is one of the most important features, and our team used zip code as a location feature. After including the zip code, the training MSE was 3869.359, and the test MSE was 4363.496, which decreased significantly.

Thus, the final linear regression model excluded features with P-values over 0.05, and the distribution of the training MSE and test MSE is shown in the histogram. The median square root of the training MSE was around 61.75, and the median square root of the test MSE was around 63. This means using this linear regression model to predict house prices will provide an error around 63 dollars.




**Log Linear Regression Model**

The linear regression model assumes that the response variable follows a normal distribution. However, the histogram plot on the left-hand side shows the distribution of the real house prices, which is not close to a normal distribution. The distribution of house prices is right-skewed. To address this issue, our team applied the logarithm function to the real house prices and plotted the histogram again, which is shown on the right-hand side. The distribution of the log-transformed house prices now fits a normal distribution.




Thus, our team uses the log function of house price. In order to prevent overfitting, we chose to include variables as matrices when applying the linear regression model. Initially, we only included numerical columns in the model, which contained a total of thirty-two columns. The training mean squared error (MSE) was 0.206 and the test MSE was 0.209. After removing these columns, the training MSE was 0.206 and the test MSE was 0.209.

Afterwards, our team converted boolean variables from true and false to 1 and 0 as a vector in the matrix. The training MSE was 0.204 and the test MSE was 0.206. Four columns had P-values over 0.05 and were deemed not statistically significant. After deleting these four columns, the training MSE was 0.204 and the test MSE was 0.206.

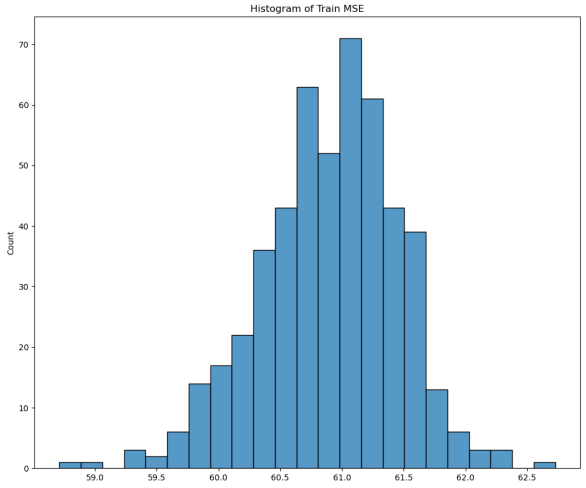
For categorical data, our team used one-hot encoding to convert the data into several vectors with 1 as an indicator for true and 0 as false for one category. There were four categorical data in total: property_type, room_type, bed_type, cancellation_policy, and host_response_time. After including these four features, the training MSE was 0.153 and the test MSE was 0.154, which significantly improved the model's prediction.

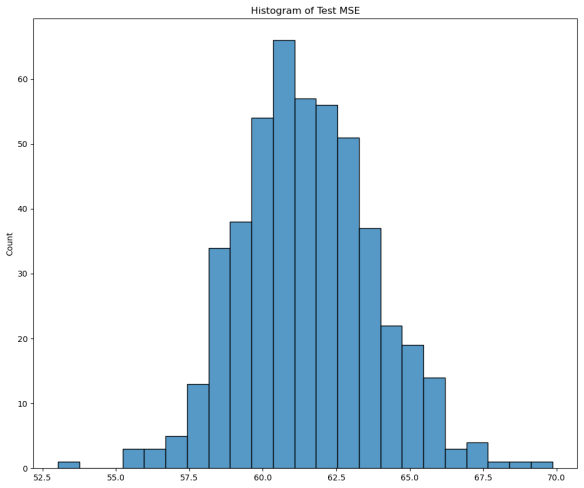
Multi-categorical data was converted into several categories vectors with 1 as an indicator for true and 0 as false for one category using many-hot encoding. The host_verifications column in the dataset was a set data column that was used in this encoding and included in the linear regression

model. The training MSE was 0.152, and the test MSE was 0.154.

Lastly, location is one of the most important features, and our team used zip code as a location feature. After including the zip code, the training MSE was 0.106, and the test MSE was 0.112, which decreased significantly.

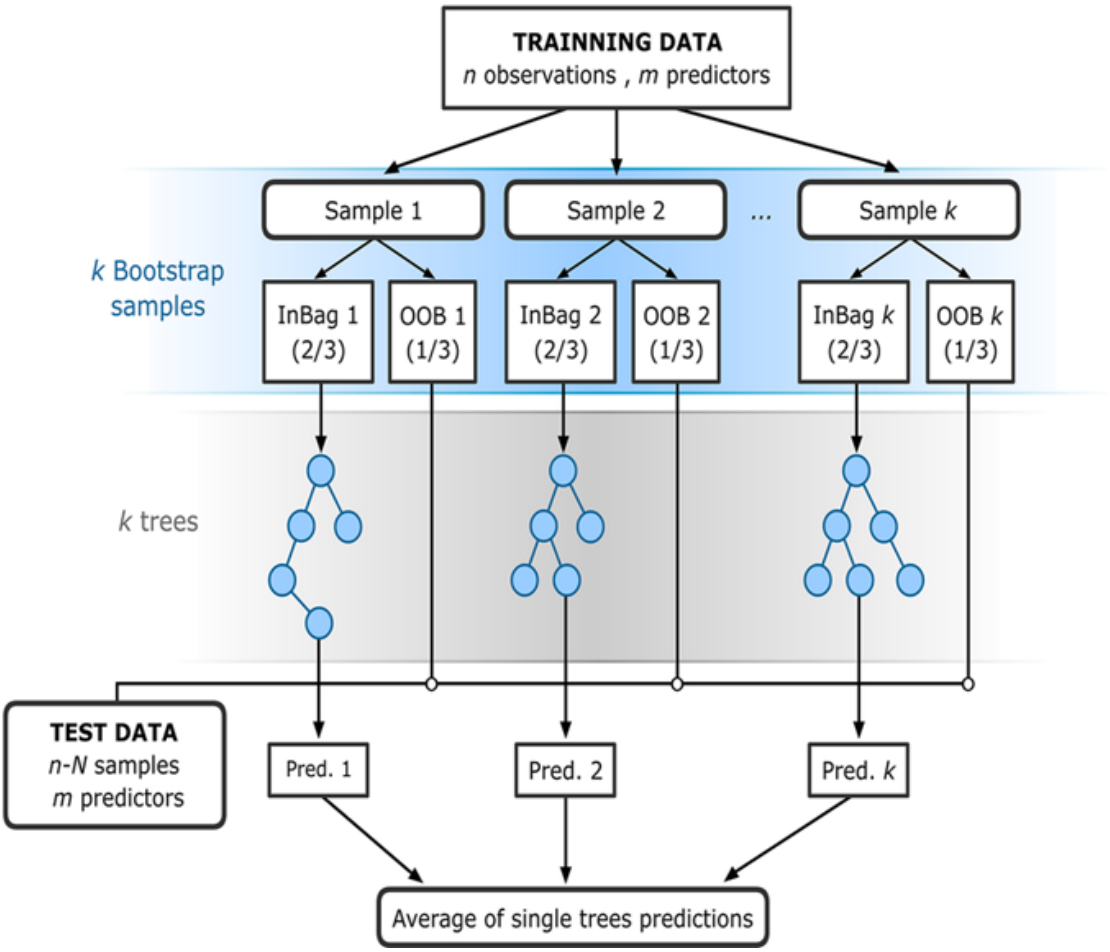
The final linear regression model excluded features with P-values over 0.05, and the distribution of the training MSE and test MSE is shown in the histogram. Compared to the previous model, our team changed the MSE back to the same as before. The median square root of the training MSE was around 61, and the median square root of the test MSE was also around 61. This means that using this linear regression model to predict house prices will provide an error of around 61 dollars which is smaller than the previous model.




**Random Forest**

We then considered using the most classical tree-based ensemble learning method: Random Forest. The mechanism behind Random Forest is the decision tree. It combines the output of multiple decision trees to reach a single result.




During each split, decision trees can use several metrics such as entropy and Gini index to measure the quality of the split. In this project, we chose to use the Gini index, which is a commonly used metric that measures the impurity or purity of the split.

The formula of the Gini Index is as follows:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

where,

'pi' is the probability of an object being classified to a particular class.

The reason why we wanted to try this method was that it could avoid overfitting since it contains a lot of trees, it was flexible since it works for both regression and classification problems, it was based on a parallel running mechanism, and it was stable due to the large number of trees.

We first used the same covariates as in linear regression, then took advantage of the cross-validation technique to ensure the stability and robustness of the model. Furthermore, we tested different combinations of hyperparameters, including the number of estimators, maximum depth, maximum features, minimum sample leaf, and minimum sample split, to get the best result with the lowest MSE. Finally, we achieved an MSE of 3608 with the hyperparameters set to the number of trees: 1000, minimum sample split: 2, minimum sample leaf: 1, maximum depth of tree: 50, and maximum features as the square root of the total number of features.

```
Best hyperparameters: {'n_estimators': 1000, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 50}
Mean squared error: 3608.0033399500694
```

Random Forest can be computationally intensive, especially with large datasets and high numbers of trees. The algorithm can also be considered as a black box, as it can be difficult to understand how the model arrives at its predictions. However, there are techniques such as feature importance measures that can help in understanding the importance of different features in the model.

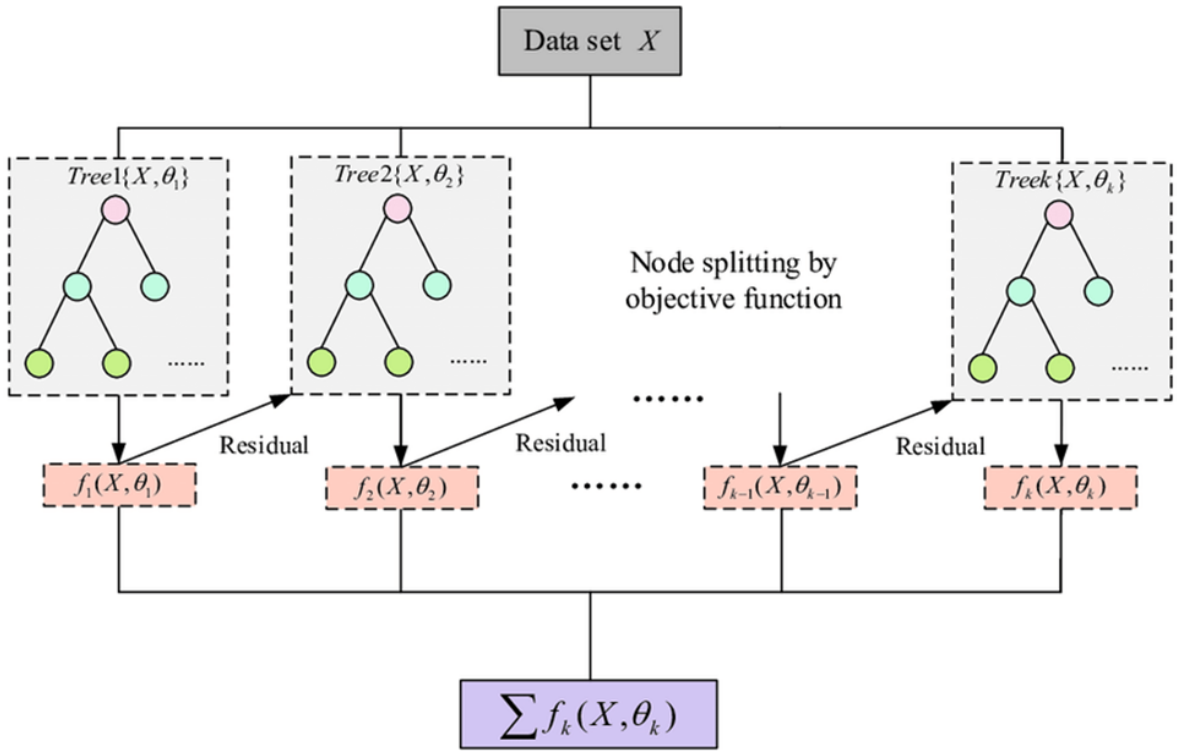
**XGBoost**

XGBoost is known for its speed and performance as it optimizes the gradient boosting algorithm to achieve high accuracy and fast training. It can handle missing values, regularization, and parallel processing. XGBoost also includes the feature importance ranking and early stopping capabilities, which make it a preferred choice for many machine learning tasks.

To use XGBoost for our project, we first defined the input variables and target variable, as we did in previous models. Then, we applied cross-validation to tune the hyperparameters, such as learning rate, maximum depth, minimum child weight, and gamma. We used the mean squared error (MSE) as the evaluation metric for the model.

After running multiple iterations with different hyperparameters, we achieved the lowest MSE of 3521 with learning rate of 0.1, maximum depth of 8, minimum child weight of 10, and gamma of 0.1.

One of the main advantages of XGBoost is its ability to handle large datasets and provide high accuracy with fast training time. It also has a well-designed system for regularization, which helps prevent overfitting. However, its black-box nature makes it difficult to interpret the results and understand the underlying relationships between variables.



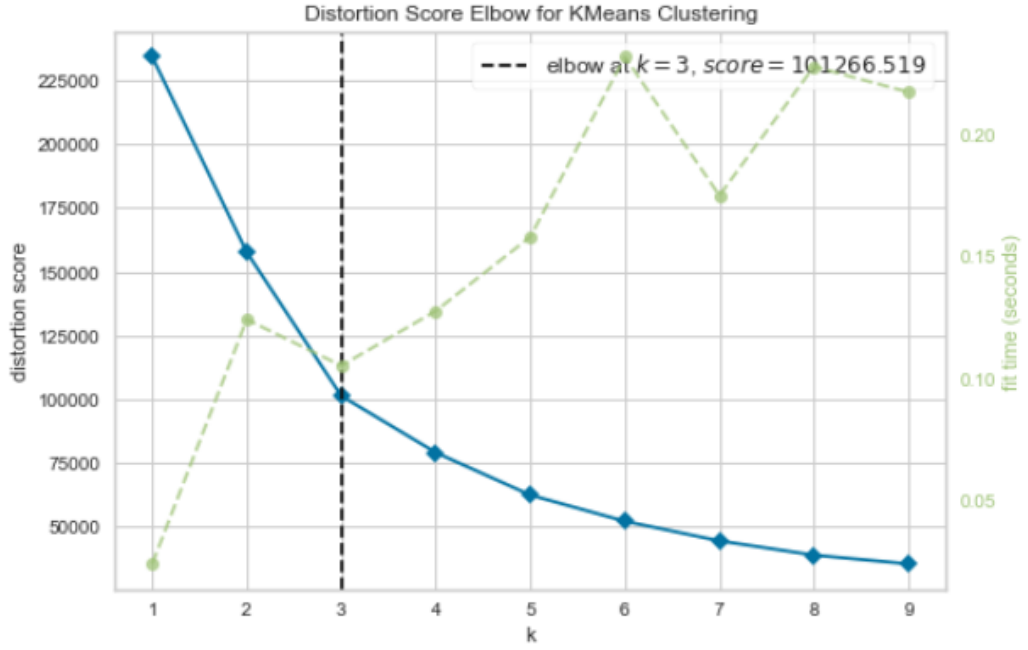

The reason why we wanted to try this method was that: 1. It could avoid overfitting since it has a customized regularization term; 2. Its loss function could be customized; 3. Less feature engineering was required. We first put the same covariates as in Random Forest, then took advantage of the cross-validation technique to ensure the stability and robustness of the model. Furthermore, we tested different combinations of hyperparameters such as the number of estimators, maximum depth, and learning rate to get the best result with the lowest MSE. Finally, we achieved an MSE of 3196 with the hyperparameters number of trees: 1000, maximum depth: 5, and learning rate: 0.1.

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best parameters: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 1000}
MSE: 3196.9645474860213
```

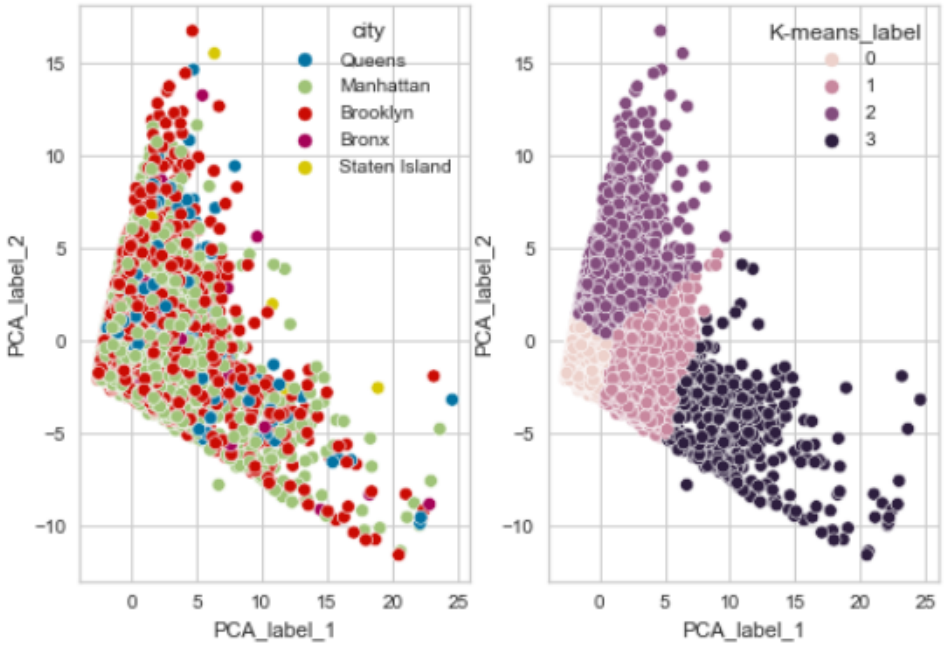
Even though it took about 2 hours to run the whole calculation, it improved the MSE a lot compared with Random Forest. However, the black box algorithm, complex hyperparameters, and difficult interpretation are some of the main disadvantages for XGBoost.

**PCA and K-means**

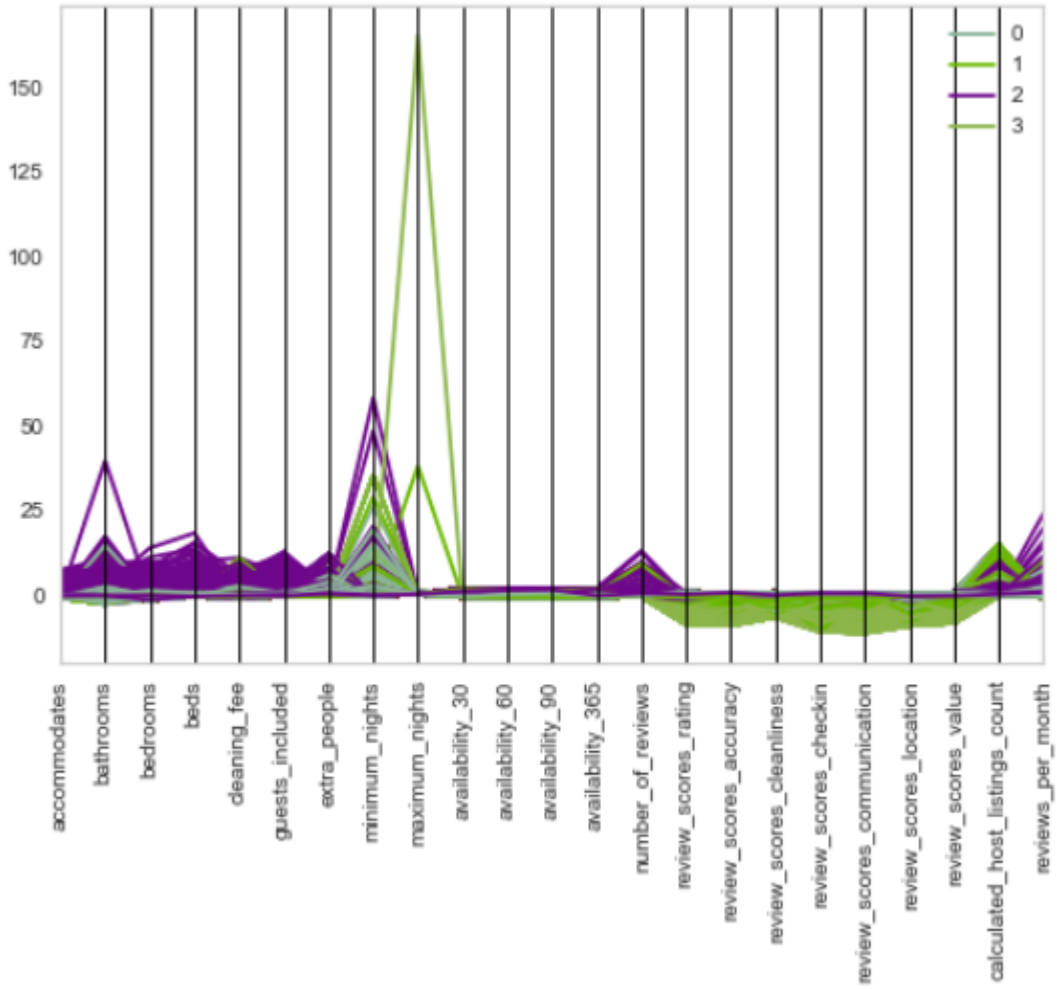
In order to visualize the clustering pattern of the houses, we first standardized all the columns values. Then we used the technique of PCA to get the first two "important" variables, which were just the linear combination of some of the "most variant" covariates. Finally, we utilized the K-means method with different numbers of clusters. Here, we used the Elbow method to find the optimal k value. After drawing the Distortion Score Elbow for K-means Clustering graph, we got the optimal number of k as 4 with the distortion score of 101266.519 (the sum of the Euclidean squared distance from the centroid of the respective clusters), which was actually not a good clustering.




So, we drew the K-means graph with k = 4, and we got a cone shape. We then compared it with the original data set split by different cities. These two graphs implied that there was no clear clustering pattern that fit the whole data set.




Finally, we drew a graph to try to see the characteristics of each cluster resulting from the K-means. We found that cluster 2 had a high level of the number of bathrooms, bedrooms, beds, accommodates, cleaning fee, guests included, extra people, minimum nights, reviews, reviews_per_month, availability_30, availability_60, availability_90, and availability_365, while cluster 3 had a high level of maximum_nights (which I think might be dirty data). Cluster 1 had a low level of review_scores_rating, review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_location, and review_scores_values, which might represent those "bad" houses.




## Conclusion

Going through the whole paper, we first took advantage of dimension reduction technique PCA and clustering method K-means and tried to figure out the pattern for different clusters. Unfortunately, after looking at the K-means graph, there was no clear pattern distinguishing between different houses. However, from the location vs price boxplot, we could see that the median price for Manhattan houses was higher than others. Besides, we could see that the entire house/apt has a higher median price from the room type vs price boxplot. Obviously, the number of bedrooms and bathrooms affected the price of the house from the heatmap and boxplot. Furthermore, the property type played an important role in influencing the price of the house, as serviced apartments have the highest median price from the property vs price boxplot. Interestingly, almost all the review score-related information has a positive correlation with the price of the house, except for the review score values from the correlation heatmap. We then mainly included all this information with some of the other columns to fit the Linear, Log Linear Regression, Random Forest, and XGBoost, respectively. The results we got were as follows.

| Model | Linear Regression | Log Linear Regression | Random Forest | XGBoost |
|---|---|---|---|---|
| MSE | 4363 | 3721 | 3608 | 3196 |

Fortunately, we made progress with different machine learning methods. Among these covariates, we found that the zip code column had a significant impact on the price of the house as it decreased the MSE significantly in the Linear Regression Model. Other than that, property_type, room_type, bed_type, cancellation_policy, host_response_time which are five columns that using one-hot encoding also reduce the MSE significantly in the Linear Regression Model. Moreover, what

caught our attention was that Random Forest and XGBoost showed significant improvements compared to Linear Regression, which could imply that there were some interaction effects between the features of the houses.

Labor Division

We are evenly distributed all the work.