Predicting Airbnb Price Listings in the United States

Problem Statement

Airbnb is an American company that operates an online marketplace that offers housing for rent, for the purposes of vacations or brief stays. It is a very popular way for Americans to rent houses for a short period of time, and is easily accessible online or via mobile app. Home owners can post their listings on Airbnb independently, but it is important to ensure prices offered are fair and competitive in order to keep Airbnb on top. Therefore, the goal of this project is to create a model that can predict prices based on the facilities (e.g. the type of house being rented out) and the location of any given listing. The model can be a good way to keep price listings in check by Airbnb.

Dataset

We obtained data containing information on Airbnb price listings in the United States in the year 2020; the dataset was obtained from Kaggle. It was contained in a .csv file and turned into a DataFrame. The columns of the DataFrame before cleaning are as follows:

1. **Id** - The listing id or house id
2. **Name** - The name of the listing or house
3. **Host_id** - the id number of the person listing the house
4. **Host_name** - the name of the person listing the house
5. **Neighbourhood_group** - the neighbourhood group the house is in
6. **Neighbourhood** - the neighbourhood the house is in
7. **Latitude** - coordinates of the house
8. **Longitude** - coordinates of the house
9. **Room_type** - the type of room being offered in the listing
10. **Price** - the price of the listing
11. **Minimum_nights** - the minimum amount of nights the house can be rented
12. **Number_of_reviews** - the number of reviews on the house from previous renters
13. **Last_review** - the date of the last review on the house
14. **Reviews_per_month** - how many reviews a particular house gets each month

15. **Calculated_host_listings_count** - how many listings the host has on Airbnb
16. **Availability_365** - the availability of a listing throughout the year
17. **City** - the city the house or listing is located in

The original dataset contains 226,030 rows and 17 columns.

Data Wrangling and Cleaning

The first glaring problem in the dataset was the amount of missing values for each variable (column in the dataset). A majority of columns, such as the price, room_type, and city had no missing data. However, some columns had a very large amount of missing data, such as the neighbourhood_group column with 115,845 rows of missing data. The last_review and reviews_per_month columns also each had 48,602 rows of missing information. That's quite a lot of missing data, and given that the dataset was very large, the best option was to simply remove all missing values from the dataset.

Removing all missing values from the dataset left us with a DataFrame containing 85,144 rows and 17 columns, which is still a lot of information to work with.

Exploratory Data Analysis

We initially explored the distributions of each column in the dataset to get a better understanding of the data. Some columns, like latitude and longitude, had a bit of an odd distribution, as they only took on certain values. This was the first indication that these columns could be dropped from the data.

We then go to find out more about the distribution of listings in each city; that is, how many different cities are covered by this dataset, and how many listings are in each city. The analysis reveals that most listings are in New York City, and that there are only 5 U.S. cities included in this dataset: Hawaii, Seattle, New York City, Los Angeles, and Rhode Island. This information is shown in Figure 1 below.

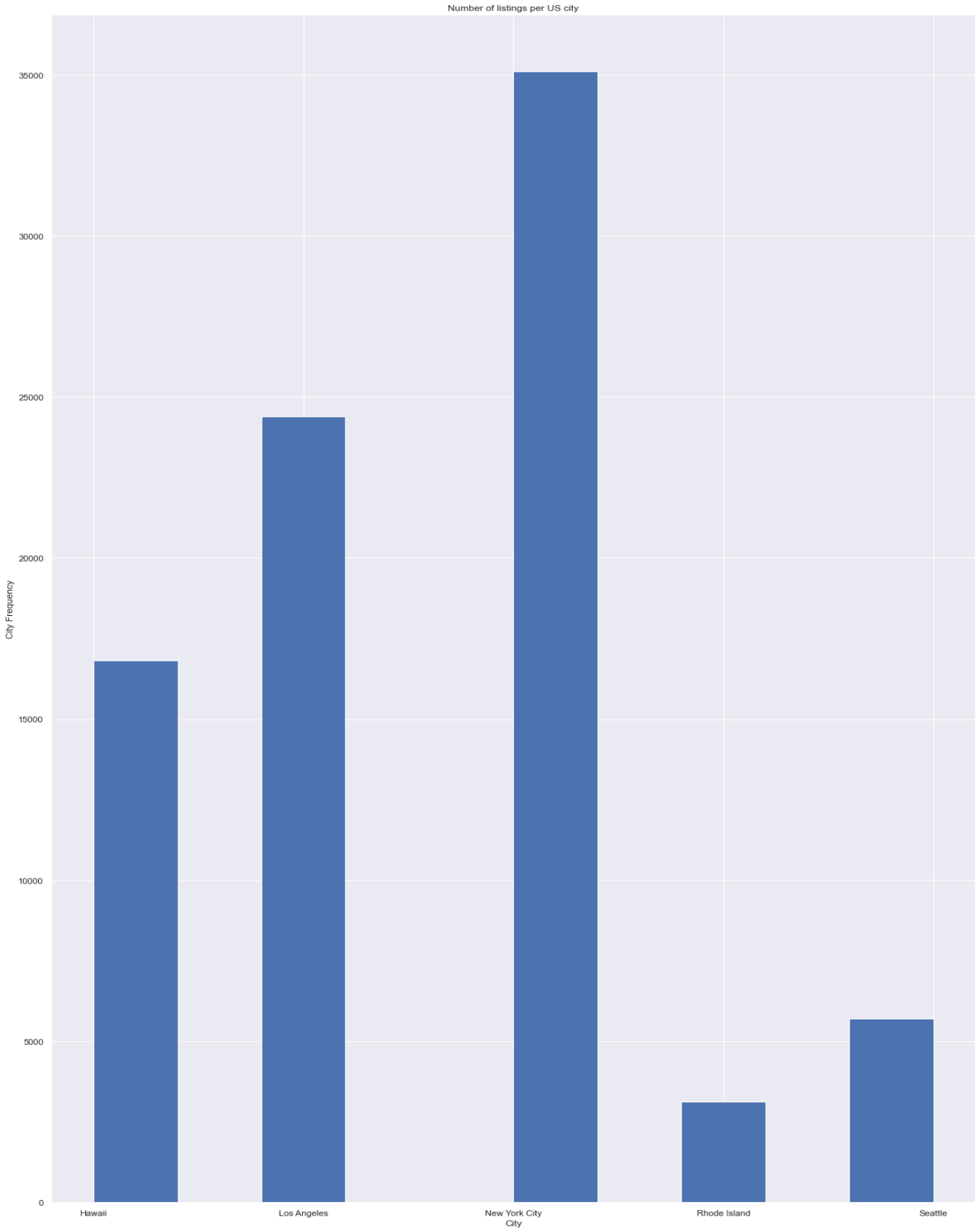
Number of listings per US city
City Frequency
0
5000
10000
15000
20000
25000
30000
35000
Hawaii
Los Angeles
New York City
Rhode Island
Seattle
City

Figure 1

Next, we explored the correlation between each column in the dataset.
The purpose of this step is to check whether any two columns are highly correlated with one another. If they are, this could interfere with our model's predictions, because we are trying to predict the price of a listing based on its features. We would need to eliminate any problematic columns that could be affecting the price, or other features in this dataset, or we could scale the data so that the high correlation is accounted for.

It was found that some columns did indeed have a high correlation with one another. For instance, the id and host_id columns had a high correlation with one another. This makes sense, because the id of the listing would belong to a certain host with a host id. That is, a particular homeowner with a certain id will have that same id for any of the listings they have on Airbnb. The id and host_id columns had a correlation of 0.56.

The latitude and longitude columns were also highly correlated. This also makes sense, since the latitude and longitude together pinpoint the exact coordinates of an Airbnb listing. These two columns had a correlation 0.83.

Finally, the number_of_reviews and reviews_per_month columns were highly correlated with one another. Logically, the total number of reviews for a listing would be the sum of the reviews it gets each month. The more it gets each month, the more total reviews it has. These two columns had a correlation 0.75.

So, these correlations make sense. However, there is a chance that they impair our model's ability to predict the price of a random listing given to it based on certain features, like room type or city. Therefore, it is probably best to exclude these columns from our model. This helps us narrow down which features we want to focus on as we proceed with our modeling: longitude and latitude will most likely not be as helpful in price prediction as, for instance, the city the listing is located in. id and host_id are also not very likely to be helpful in predicting the price of a listing. We may want to keep the reviews columns, because they could be helpful in price prediction. For example, if a listing has a lot of reviews, specifically good reviews, it would be viewed by potential renters as more trustworthy than a listing

with no reviews at all. The price could go up based on how trustworthy a listing is, and that is mainly decided by the reviews.

We created a heatmap of the correlations between each of the columns in the DataFrame. Light red means a weak or no correlation, and the darker red indicates a strong correlation between two columns. This heatmap, which was created with seaborn, is shown below in Figure 2.

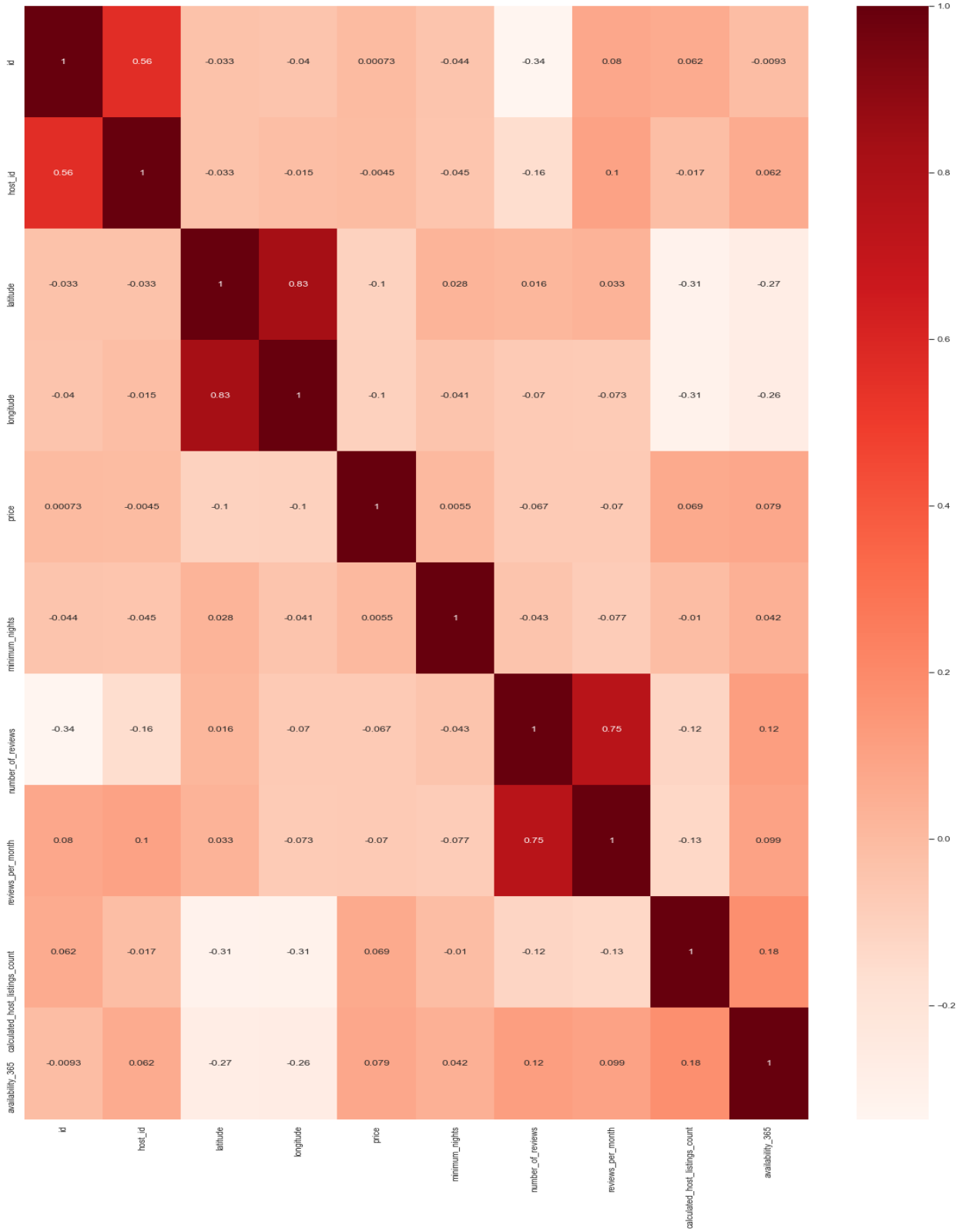
id
host_id
latitude
longitude
price
minimum_nights
number_of_reviews
reviews_per_month
calculated_host_listings_count
availability_365
1 0.56 -0.033 -0.04 0.00073 -0.044 -0.34 0.08 0.062 -0.0093
0.56 1 -0.033 -0.015 -0.0045 -0.045 -0.16 0.1 -0.017 0.062
-0.033 -0.033 1 0.83 -0.1 0.028 0.016 0.033 -0.31 -0.27
-0.04 -0.015 0.83 1 -0.1 -0.041 -0.07 -0.073 -0.31 -0.26
0.00073 -0.0045 -0.1 -0.1 1 0.0055 -0.067 -0.07 0.069 0.079
-0.044 -0.045 0.028 -0.041 0.0055 1 -0.043 -0.077 -0.01 0.042
-0.34 -0.16 0.016 -0.07 -0.067 -0.043 1 0.75 -0.12 0.12
0.08 0.1 0.033 -0.073 -0.07 -0.077 0.75 1 -0.13 0.099
0.062 -0.017 -0.31 -0.31 0.069 -0.01 -0.12 -0.13 1 0.18
-0.0093 0.062 -0.27 -0.26 0.079 0.042 0.12 0.099 0.18 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Figure 2

We then take a closer look at the distributions of each of the columns in the DataFrame, using distribution plots available in seaborn. At this point, there are 85,144 rows and 17 columns in the DataFrame. The distributions of each column indicated that there were still too many data points to consider. More importantly, it revealed that almost every column had significant outliers present. Outliers can significantly hinder our model's ability to predict the price of an Airbnb listing, so it is important to remove them. We trimmed the data to exclude outliers present in each column, using a variety of methods that depend on the distribution of a particular column. Columns that had a skewed distribution were trimmed by removing outliers using the IQR method; this is done by calculating the first quartile (Q1) and the third quartile (Q3), calculating the difference between quartiles, and removing all points that lie above the maximum range, defined as Q3 + 1.5*IQR. Columns with a normal distribution were trimmed by removing outliers lying outside the 99th percentile of the distribution.

After removing outliers from all of the columns, the trimmed dataset contained 62,808 rows and 17 columns. We removed quite a few rows (over 20,000), but still had a lot of data to work with for our modeling.

Model Selection

Three models were chosen to predict prices for this dataset: Linear Regression model, a Random Forest Regression model, and a K-Nearest Neighbor Regression model. After preprocessing the data, we only keep the following columns as our predictor variables:

1. Minimum_nights
2. Number_of_reviews
3. Reviews_per_month
4. Calculated_host_listings_count
5. Avaialability_365
6. DM_Hawaii

7. DM_Los Angeles
8. DM_New York City
9. DM_Rhode Island
10. DM_Seattle
11. DM_Entire Home/apt
12. DM_Hotel room
13. DM_Private room
14. DM_Shared room

Columns 1-5 are the numeric variables we chose to keep for the model. These columns were scaled using skearln's StandardScaler. Columns 6-14 are dummy variables created during preprocessing to represent the categorical features we want to use in our model to predict price. The five different cities that have listings, as well as the four different room types offered were deemed to be the most relevant categorical price predictors, so dummy variables were created for them, so that they can be evaluated using the three models mentioned above. These 14 columns are our X variable, which will be used to predict price. The price, which is not included in the scaling or in these columns, is our target feature; the variable we want to predict. Therefore, price was stored separately as our y variable. We created training and test splits of this data in order to train the linear regression , RF, and KNN regression models and predict price.

Before choosing the best performing model, we first tuned the hyperparameters of each model to get its best performance on this data. We tuned the hyperparameters of each of the three models using GridSearchCV from sklearn's model selection. It was found that the best k for the linear regression model was k=11, with a maximum of k=14. For the Random Forest model, it was found that the best number of trees is 784(n_estimators). This is a very high number, which suggests high accuracy, but makes our code run very slowly. The best number of neighbors for the KNN Regression model was 9 neighbors. This is also shown using an Elbow Curve, which graphs all possible values of K (number of neighbors) and their respective sum of squares. The idea is that the best K lies at the elbow of the curve, which is where the SS begins to get lower and lower. This is shown below in Figure 3.

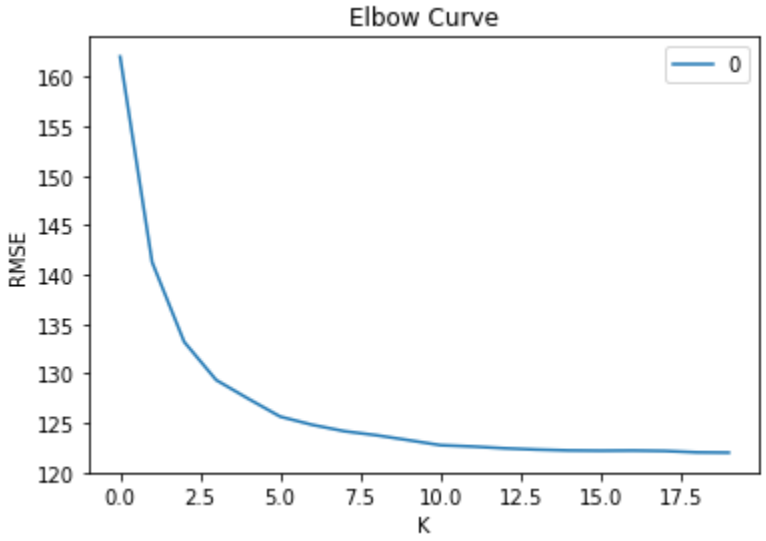

Figure 3

After hyperparameter tuning, we cross validate each model using sklearn's cross_val_score feature. We then calculated the R2 score and mean absolute error of each model to evaluate performance. The mean absolute error of the KNN and RF Regression models were the highest, meaning they had the highest accuracy when predicting price. The Linear Regression model consistently had the lowest mean absolute error. The RF model had the lowest variability of its mean absolute error, suggesting it was the most reliable, while the KNN model had the highest variability of all three models. The mean absolute error of the KNN was highest at $75.2, while the RF model had a mean absolute error of $75.44. This means the KNN model was predicting price 0.22 cents more accurately than the RF model. The linear regression model's mean absolute error was $77.61. The average price across all listings is $157.9, which means all three models performed better than if we simply guessed price by taking the average of all listings by about $80.

Given these assessment results, it is clear that the choice of modeling is between the KNN Regression model and the Random Forest Regression model. Which one to choose is difficult to say without further hyperparameter tuning. If we consider the bias-variance

tradeoff, an argument can be made to choose the RF model, because it had much lower standard deviation of the mean absolute error, meaning it would predict price with less variability. The accuracy in price prediction between the KNN and RF models were very close; only $0.22 difference. Therefore, the suggestion is to proceed with the RF model over the KNN model. Another reason an argument can be made for the RF model over the KNN model is because as we add more and more listings, the RF model would scale well with the data, while the KNN would take too long to accurately predict price. The comparisons of mean absolute error, and its standard deviation, for each of the three models can be seen in Figure 4 below.

Model Name	Mean absolute error	Standard Deviation
KNN Model	75.2	1.59
Random Forest Model	75.44	1.18
Linear Regression Model	77.61	1.4

Figure 4

Notably, the linear regression model revealed that the most important features in predicting price were as follows:

- Hotel room

- Entire room/apt
- Private room
- Shared room
- Rhode Island
- Hawaii
- Availability_365

This suggests the type of room offered in a listing was the strongest indicator of what the price should be, with hotel rooms having the highest coefficient, meaning hotel rooms were generally more expensive than other room types. In terms of city, listings in Rhode Island were most expensive. Rhode Island also had the least amount of listings, as we discussed in Figure 1. It makes sense that the rarer the listing, the more expensive it is, as the demand would greatly exceed the supply of an Airbnb room in Rhode Island. In terms of numeric features, the listing's availability throughout the year was the best predictor of price.

Further Research and Recommendations

Recommendations on how to use this model:

- The model can be used by Airbnb to ensure that new price listings are accurate and competitive enough based on the type of room being offered and location of the listing.
- The model can be used by independent homeowners who are looking to rent their home with the goal of being more competitive than a big company like Airbnb.
- The model could be used by developers who aim to rent out their homes or condos and want to decide what type of houses to build.

Further research:

- The model should be expanded to include every city in the United States.

- An ensemble model could be created to expand to every city and make price predictions.
- The model can also be expanded to include more years than 2020 alone. It could be expanded to about 5 or 10 years. This would allow us to analyze the trends in rental prices over time.