ORIE 5741 Final Report

How to find a well worth Airbnb room in New York

Sicheng Zhao(sz629), Boyuan Cui(bc594)

1. Introduction

1.1. Background

In New York City, there are hundreds of hotels to choose from, and as of September 2021, over 36,000 Airbnb rentals—both rooms to rent in occupied homes or entire apartments or townhouses. Like New York City itself, its hotels and Airbnb rentals are diverse, unique, and come with a sliding scale of price points from 1000 a night to under 100. However, currently, there is no convenient way for a new Airbnb host to predict the price of his or her listing. New hosts must often rely on the price of neighboring listings when deciding on the price of their listing. A system that may tell people which feature of an Airbnb room is important and how much does each type of room worth is in great demand.

1.2. Objective

This project analyzes Airbnb listings in New York City to better understand how different attributes such as bedrooms, location, house type amongst others can be used to accurately predict the price of a new listing that is optimal in terms of the host's profitability yet affordable to their guests.

Our model is intended to predict the price of this Airbnb room based on some characteristics, so it can help users to choose the most suitable room for them, thereby improving the happiness of travelers. Such a model can not only better help travelers choose Airbnb rooms in New York, but also can be applied to the entire hotel industry across the world to help hotels improve their pricing system.

2. Date Set

To reach this goal, we decide to build a prediction model based on the New York listing dataset and calendar dataset. The listing data contains 29142 rows and 96 features, including *the number of people accommodated, bedrooms, beds, review scores, host response rate,* and so on.

The calendar data contains 13469123 rows and 7 features, including *listing id, data, available, price adjusted price, minimum nights, maximum nights.*

2.1. Features and Data Pre-Processing

2.1.1. Listing Data

In the listing dataset, the features can mainly be divided into two groups, numerical data, and categorical data. Among them, numerical data accounted for 44 percent, categorical data accounted for 56 percent.

After we get the data, we start the cleaning process. In these features, we first drop all the features that have a missing rate larger than 50%, and for numerical features, we fill the missing values with Mean. Secondly, in all these features, many redundant features have less or no relation with the output feature, price. So, we drop all the irrelevant features such as *listing url, host id, picture url,* and so on, also we drop the text features since we do not want to do the natural language process. After finishing the pre-processing work, 36 features are left, including 28 numerical features and 9 categorical features.

2.1.2. Calendar Data

For Calendar Data, there are 7 features in total and since the number of null values is small, so we decide to drop those rolls. After that, we get a dataset with 13469123 rolls and 7 columns. In addition, since the price feature is presented in this style *$150*, we remove the dollar symbol and convert the string into a float for further process.

2.2. Exploratory Data Analysis

In this section, we do some exploratory based on the cleaned data. For listing data, we draw the geographic heatmap of the Airbnb prices, we have also compiled room prices by room type and by neighborhood. For the calendar data, we calculated the price trend of rooms in different months of the year, and we calculated the percentage increase of room prices on weekends.

2.2.1. Listing Data

We first draw the geographic heatmap of the Airbnb prices using the latitude and longitude to get an overall view. The map is shown below.

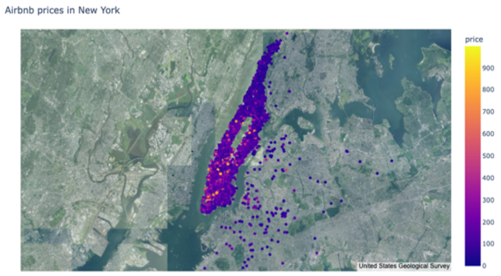

Figure 1. Heatmap of Price

We can see that most of the prices are concentrated under $400, several are over $700. The following figure shows more clearly.

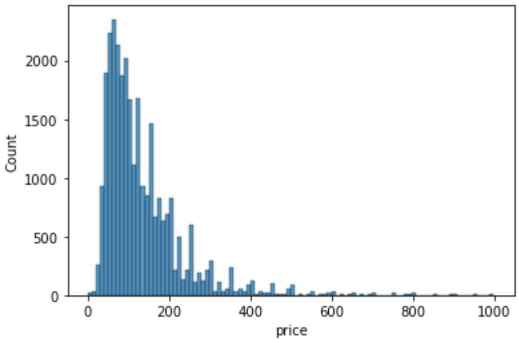

Figure 2. Price Distribution

We then draw the box plot for the room type and price to illustrate their relations. The lowest price of the three types of rooms is almost the same, however, the median and the maximum prices of the Entire home/apt are higher.

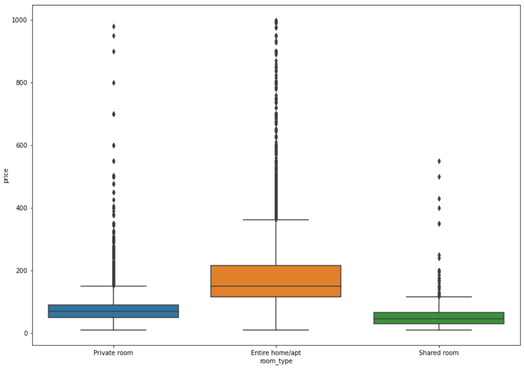

Figure 3. Price Distribution based on Room Type

In addition, we draw the average price for private rooms, shared rooms, and entire homes for each of the neighborhoods and list the top 10 and bottom 10.

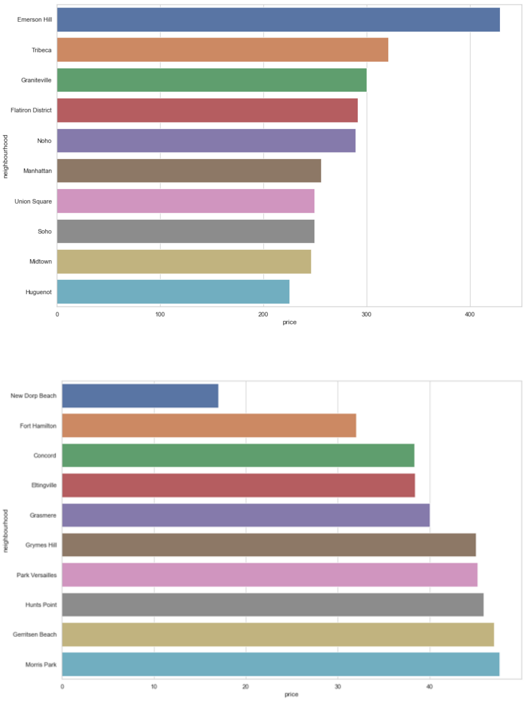

Figure 4. Average Price Rank for room types

From the chart above we can see that Emerson Hill is the neighborhood that has the highest average price and New Drop Beach has the lowest average price.

2.2.2. Calendar Data

For calendar data, our main goal is to find the relation between price and time, so we draw the price change over time. In addition, we calculate the price difference between weekdays and weekends.

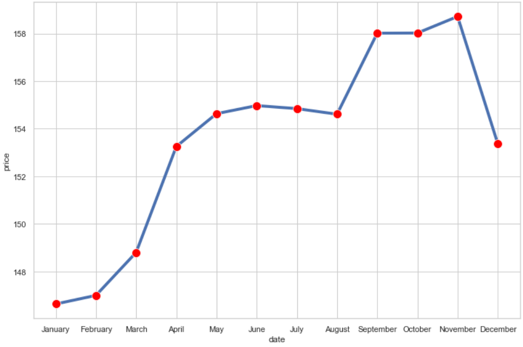

Figure 5. Price Change Over Month

We can find that the peak price is in November, and the lowest price is in January. In addition, we draw the price change over days.

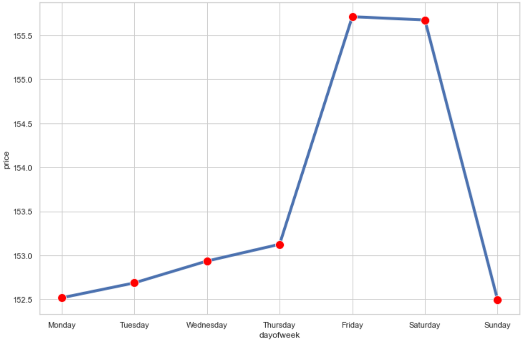

Figure 5. Price Change Over Day

Friday and Saturday are the two days that have the highest price which is about 2% higher than other days.

3. Methods and Models

Since we need to apply several models to the dataset, we encode discrete values into integers by using the one-hot encoding method. By doing this, we give a new binary variable for each unique value to remove the ordering between categories so that the result can have a better performance. Furthermore, we do the log calculation on the price to illuminate the influence of the outliers for better prediction. Finally, we split the dataset into the training set (80%), validation set (10%), and test set (10%) for further prediction.

In this project, RMSE and R2_score are used to evaluate and compare different Machine Learning Models.

The RMSE tells us how well a regression model can predict the value of the response variable in absolute terms while R2 tells us how well a model can predict the value of the response variable in percentage terms.

3.1. Base Model

Before trying various Machine Learning Models, it is important to set baseline performances based on simple heuristics or simple models. Accordingly, the K-Nearest Neighborhood Regression is the baseline model to compare the other Machine Learning Models.

As defined, the target is predicted by local interpolation of the targets associated with the k-nearest neighbors in the training set.

We train the KNN model and test it on the test set to get the RMSE and R2 scores. The result is shown in the table below.

Table 1: RMSE and R2 for KNN

Model	**Train**	**Test**	**Train**	**Test**
	RMSE	**RMSE**	**R2**	**R2**
KNN	0.427	0.527	0.555	0.341

Plus, to get a clear comparison of real data and predict data, we draw the True vs. Predict figure with an ideal trend line on it.

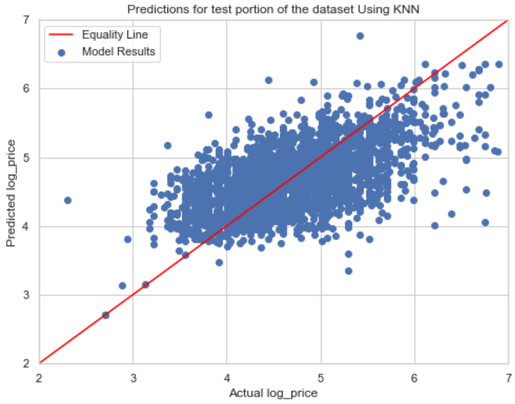

Figure 6. True vs. Predict for KNN

3.2. Linear Regression

After we tried the data on the baseline model, we can continue to apply another machine learning model to it. We first try the linear regression model since it is the simplest and widely used model. The results are shown below.

Table 2: RMSE and R2 for LR

Model	Train RMSE	Test RMSE	Train R2	Test R2
LR	0.347	0.372	0.720	0.712

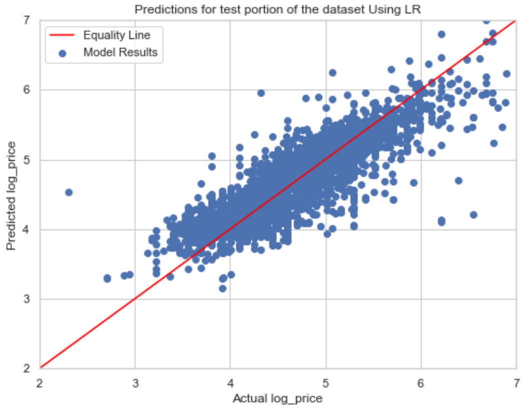

Figure 7. True vs. Predict for LR

3.3. Decision Tree Regression

After that, we try to apply Decision Tree Regression to our dataset. At the first time, we get the result as shown below, we can see that the rmse is low on the training set but high on the test set.

Table 3: RMSE and R2 for DT

Model	Train RMSE	Test RMSE	Train R2	Test R2
DT	0.008	0.492	0.999	0.424

From the results, we can conclude that the model is overfitting, so we do the Hyper-parameter tuning for the model. In order to get the best possible results from any Model, it is vital to determine the right combination of hyper-parameters to be used. This process is known as hyper-parameter tuning. It involves training the Model with different values for a set of parameters. The Model performance is then computed using the Validation Dataset. The parameter combination which yields the best performance is the one that will eventually be selected while comparing various models.

To determine the best possible values, the following different values of parameters were tried using Grid Search Cross-Validation.

Table 4: Hyperparameter for DT

Hyper-parameter	Values
Maximum Depth	[1, 20, 100]
Maximum Number of Features	[1, 5, 15, 20]
Maximum number of leaf nodes	[5, 50, 100]

The best estimator was determined to be with the following values of hyper-parameters, Max Depth=20, Max Features=20, and Max Leaf Nodes=100. After applying the best estimator, the rmse and r2 scores are shown below.

Table 5: RMSE and R2 for DT Tunned

Model	Train RMSE	Test RMSE	Train R2	Test R2
DT	0.435	0.443	0.541	0.535

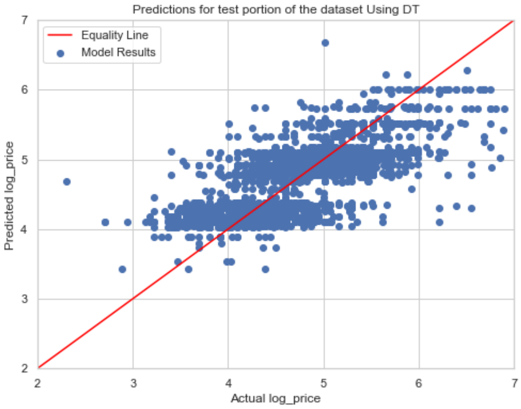

Figure 8. True vs. Predict for DT

3.4. XgBoost

XgBoost is an implementation of gradient boosting machines. It uses decision trees as base learners; combining many weak learners to make a strong learner. As a result, it is referred to as an ensemble learning method since it uses the output of many models in the final prediction. So, we think this model can make a better result in our prediction. The results are shown below.

Table 6: RMSE and R2 for XGB

Model	Train RMSE	Test RMSE	Train R2	Test R2
XGB	0.303	0.333	0.776	0.740

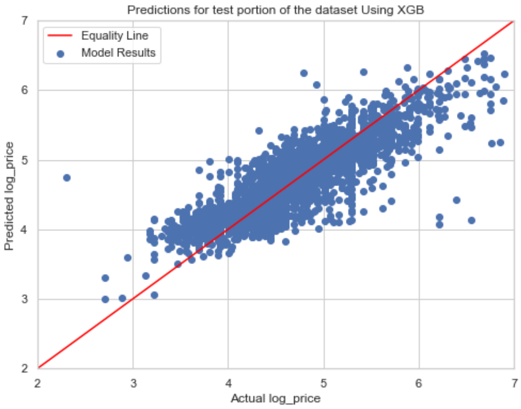

Figure 9. True vs. Predict for XGB

3.5. Random Forest

At last, we also apply the random forest model to see if it will have a better performance. To get the best parameter to prevent overfitting, we do the hyper-parameter tuning process again to get the best estimator. We use for loop to test the *min sample of leaf* and the best number of the *n_estimitors*. The following figures show our process.

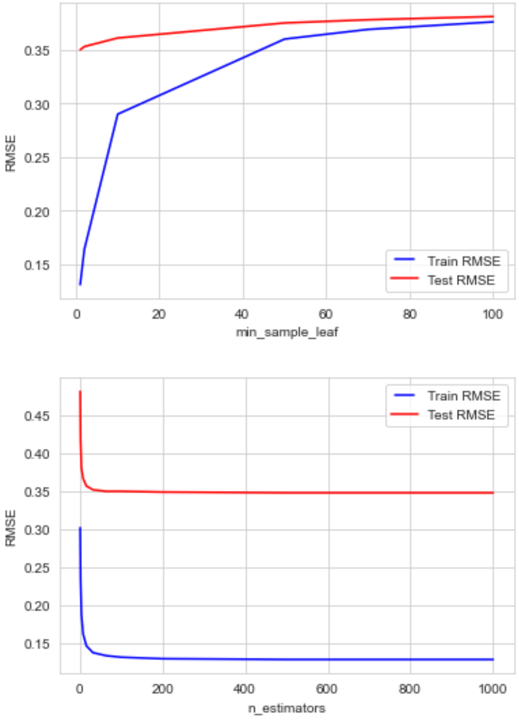

Figure 9. Tunning the number of leaf and estimator

The best parameter for our model is *n estimators = 500* and *min samples leaf=1*. The outcome of the model is shown below.

Table 7: RMSE and R2 for RF

Model	Train RMSE	Test RMSE	Train R2	Test R2
RF	0.128	0.348	0.960	0.712

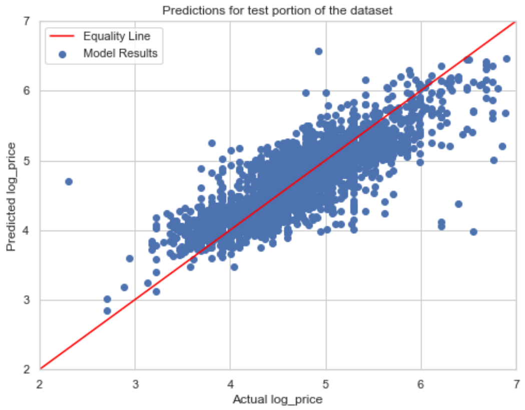

Figure 9. True vs. Predict for RF

4. Result Compare and Discussion

4.1. Model Compare

In this part, we mainly compare the previous results of different models and use multiple methods to figure out the contribution of different features in the dataset and analysis their influence on the prediction.

We put the results of all the models together and get a comparison table below.

Table 8: RMSE and R2 for 5 Models

Model	Train RMSE	Test RMSE	Train R2	Test R2
KNN	0.427	0.527	0.555	0.341
LR	0.347	0.372	0.720	0.712
DT	0.435	0.443	0.541	0.535
XGB	0.303	**0.333**	0.776	**0.740**
RF	**0.128**	0.348	**0.960**	0.712

As the baseline model, KNN performs the worst on both the Training and Test set. Linear Regression performs not badly and even better than Decision tree on the data, it may be because there are more numerical features in the data with strong linear relationships.

Furthermore, we can conclude that Random Forest performs the best on the Training set and XGBoost performs the best on the Testing set. From the table, the R2 score for the Random Forest model on the training data is 0.96 which means it has 96% accuracy and the R2 score for XGB is 0.74 on the testing data which means it has 74% accuracy in predicting the test data.

We can also draw the point to point compare based on randomly selected data to get a more direct knowledge of the performance of the Random Forest model and XGBoost model.

For example, we randomly choose 50 price points in the dataset and draw the real and predicted value on the test set in the same coordinate.

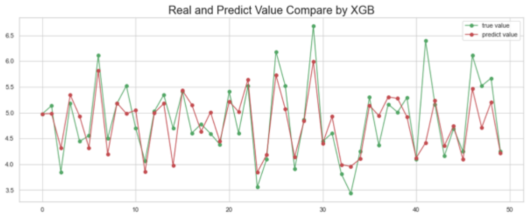

Figure 10. Real and Predicted Value by XGB

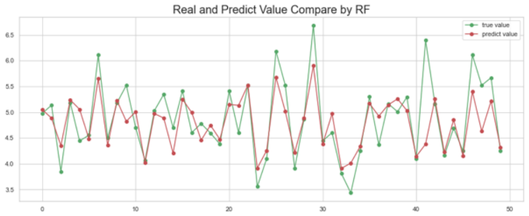

Figure 11. Real and Predicted Value by RF

We also calculate the maximum difference between predicted price and the real price, the result is 2.39 which is about \$11, so we can predict prices within \$11.

$$e^{(Max(Y_{predict}-Y_{real}))} \approx \$11$$

4.2. Feature Importance

After finding the best two models, we decide to go further into them. We calculate the feature importance based on Pearson Correlation to see the linear relationships between the input features and the target variables. The importance is shown below.

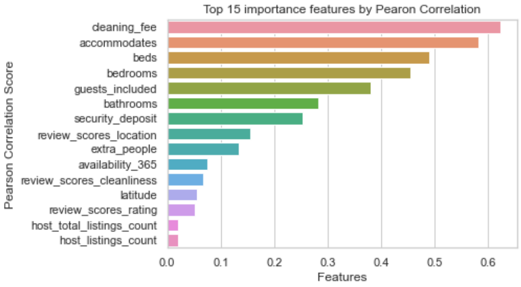

Figure 12. Top 15 important features by Pearson

However, 1/4 of our training and testing dataset are categorical features, in order to see the relationships between price and categorical outputs, we also calculate the importance based on the XGBoost and the Random Forest. The top 15 important features for XGB and RF are listed below.

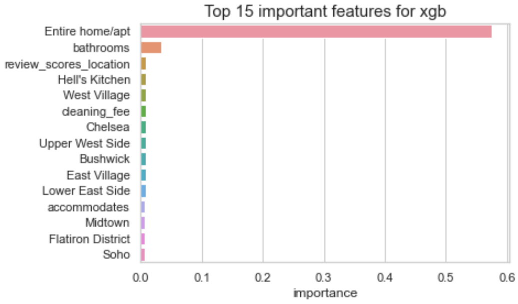

Figure 13. Top 15 important features for XGB

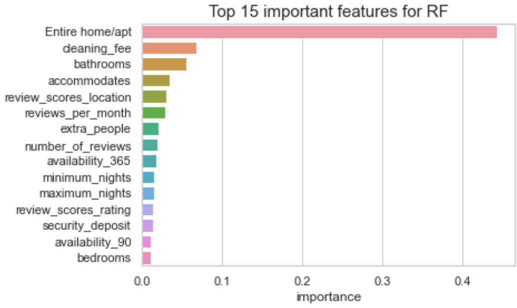

Figure 14. Top 15 important features for RF

As expected, the feature *Entire home/apt* rank top 1 in both models and its importance is way higher than the rest features. This result confirms our previous conclusion during the data exploration that the Entire home/apt has the highest average and maximum price. Other features such as *Bathrooms, Cleaning fees, Accommodates, Bedrooms, Guest included* are also important features when predicting the price.

In calculating the feature importance of the XGB, we find that neighborhood such as *Hell's Kitchen, West Village, and Chelsea* also ranks high in the list. This reflects that the price of these neighborhoods has a higher influence on the prediction of our model, so we decide to draw the list of the top 15 neighborhoods with the highest price to compare with this result. The figure is shown below.

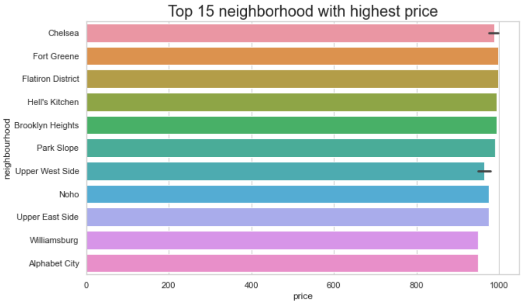

Figure 15. Top 15 neighborhoods with the highest price

We can see that *Chelsea, Hell's Kitchen*, and other neighborhoods with higher prices are also listed in Figure 13. In addition, we count the number of houses that have a price higher than $900.

Table 9: Houses with price over $900

Neighborhoods	**Numbers**	**Price in $**
Chelsea	3	999/995/975
West Village	2	950*2
Upper West Side	2	950/980
Hell's Kitchen	1	999
Flatiron District	2	999/999

This conclusion confirms that neighborhoods with higher prices have more influence on the prediction of our XGB model.

4.3. Improve Analysis

Furthermore, instead of just knowing the important features, we also want to know how these features will influence our prediction. We implement a python package call SHAP which helps us analyze the positive and negative contribution of each feature on the model prediction. We test the SHAP on XGBoost and come out with several results.

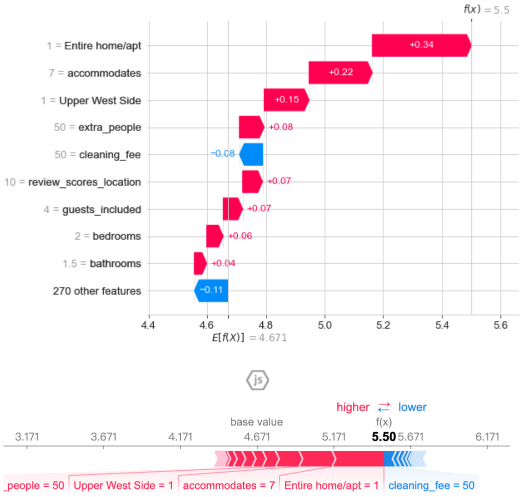

Figure 16. Contributions of the top features

The above two figure explanation shows features each contributing to pushing the model output from the base value (the average model output over the training dataset we passed) to the model output. The base value 4.671($106) is the means of all prices values which would the same for all instances. In addition, we can see features such as the *Entire home/apt, accommodates* has a positive contribution to the output of our model and *cleaning fee, room type private room, reviewing score, reviewing per month* has the negative contribution and they all determined the final prediction value to be 5.5 in log price which is $244 in regular price.

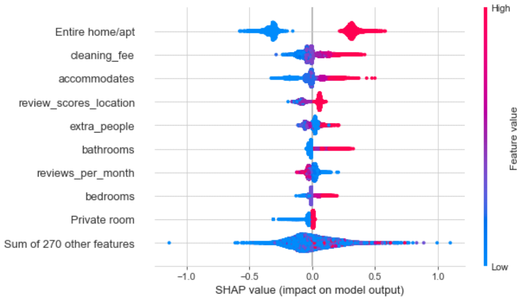

Figure 17. Sum of the contributions of the top features

The plot above sorts features by the sum of SHAP value magnitudes over all samples and uses SHAP values to show the distribution of the impacts each feature has on the model output. The color represents the feature value (red high, blue low). This reveals for example that a high *Entire home/apt, cleaning fee and accommodates* increases the predicted home price. The more *reviewing per month* the less the price will be which can probably interpret that a relatively cheap room may attract more people.

At last, we run the mean SHAP value to see the overall variable importance.

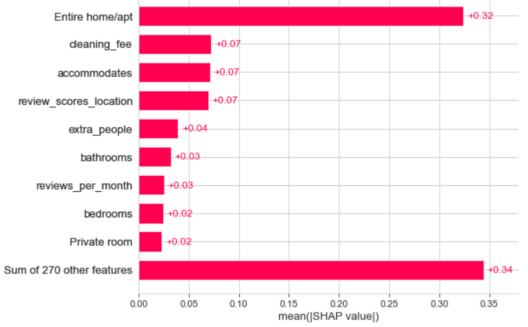

Figure 18. Overall variable importance by SHAP

We can see the top features are almost overlap with the previous results.

5. Conclusion and Future Work

5.1. Conclusion

From our analysis, we can see that XGB and Random Forest have better performance on the RMSE and R2 score because of the presence of categorical data, while the linear model also does not perform badly because most of the data is numerical data.

By comparing the different features, we can conclude that *Entire room/apt* has the greatest positive effect on the prediction results, followed by the *accommodates, Upper West Side and extra people.* However, *cleaning fee cleaning fee, room type private room, reviewing score, reviewing per month,* has a negative effect on the prediction.

Taking into consideration the initial hypothesis of whether we will be able to predict a new listing's price based on its features, we would say that we are able to predict price within $11. This is aligned with the project goals since using the initial price to start with, users can make any necessary changes based on different important features such as *Room types, cleaning fee, bedrooms, accommodates, reviewing score, days, and months* etc.

5.2. Ethical

5.2.1. Weapon of math destruction

After careful review of our entire modeling process and outcomes, we conclude that this project is unlikely to become a weapon of math destruction. These are three main reasons: the outcomes are clearly stated and easy to measure (just the listing price predictions); the modeling process is static and unidirectional (using relevant property information to predict its price) so there is no feedback loop; the predictions are only for reference purpose, so it is hard to infer any damaging effect to the society.

5.2.2. Fairness

There is no significant fairness concern in this project due to the dataset itself. As we check across all used features in our model, majority of them are numerical values reflecting objective information about such as user review frequency and geographic location on the property listings.

On the other hand, the remaining categorical features like room type and neighborhood are distinguishing factors by nature, as we can easily see that an entire house in downtown Manhattan will cost significantly more to rent than a shared room in Brooklyn suburban area. Since we are evaluating all features by the same methods to make predictions, there should be no apparent discrimination associated with our prediction models.

5.3. Future Work

Some features are not used in our dataset. These are majorly text-based features for example, Amenities (Text), Description (Text) and Reviews (Text). We may apply some natural language process on our dataset in the future to make the prediction more precise and robust.

Also, the location should play more important role in our model, so we may use some other methods to process the location data.

6. References

[1] http://insiderairbnb.com/get-the-data.html

[2] https://github.com/slundberg/shap