Airbnb in New York

Yuyang Li
Brown University Data Science
Github: https://github.com/mikelyy/Airbnb-in-New-York

Figure 1 Airbnb Tag

1. Introduction

1.1 Project Task

The goal of this project is to build different machine learning models to predict the prices of Airbnb in New York, given the datasets from Kaggle. It is an interesting topic because nowadays more people use airbnb to expand on traveling possibilities. Staying in airbnb gives people more personalized experiences. It is a regression task. The target variable is price.

1.2 Related Work

From Kaggle Kernels, I have gone through some projects completed by other people. In Dgomonov's "Data Exploration on NYC Airbnb", he uses the data for security, business decisions and guiding marketing initiatives, etc. I get some ideas of how to implement exploratory data analysis and feature engineering for the dataset. In Andrew W's "Smart Pricing with XGB, RFR + Interpretations", he uses the data to build and train a smart pricing model. The target variable is price. I learn some feasible regression models. See reference in section 6.

1.3 Dataset Description

There are in total 16 features and 48895 rows in the original dataset. Features name, host_name, last_review and reviews_per_month have missing values with 0.0327%, 0.0429%, 20.5583% and 20.5583% respectively. After performing data cleaning and transformations, there are in total 14 features. The description of features is shown below.

Name: returns positive if it has at least one of four frequent words (bedroom, privat, apart, brooklyn), negative otherwise. (categorical)

Neighbourhood_group: represents the location of airbnb in New York. (categorical)

Neighbourhood: represents the area of airbnb in New York. (categorical)

Latitude: represents the latitude coordinates of airbnb in New York. (continuous)

Longitude: represents the longitude coordinates of airbnb in New York. (continuous)

Room_type: represents the listing space type. (categorical)

Price: represents the price of airbnb in New York. (continuous)

Minimum_nights: represents the minimum amount of nights. (continuous)

Number_of_reviews: represents the number of reviews. (continuous)

Reviews_per_month: represents the number of reviews per month. (continuous)

Calculated_host_listings_count: represents the amount of listing per host. (continuous)

Availability_365: represents the number of days when listing is available for booking. (continuous)

Race: represents ethnicity of the host. (categorical)

Time: represents the difference between each last_review date and standard date (‘2019-07-09’). (continuous)

2. EDA

2.1 WordCloud

Figure 2 WordCloud shows frequent words in name

2.2 Histogram

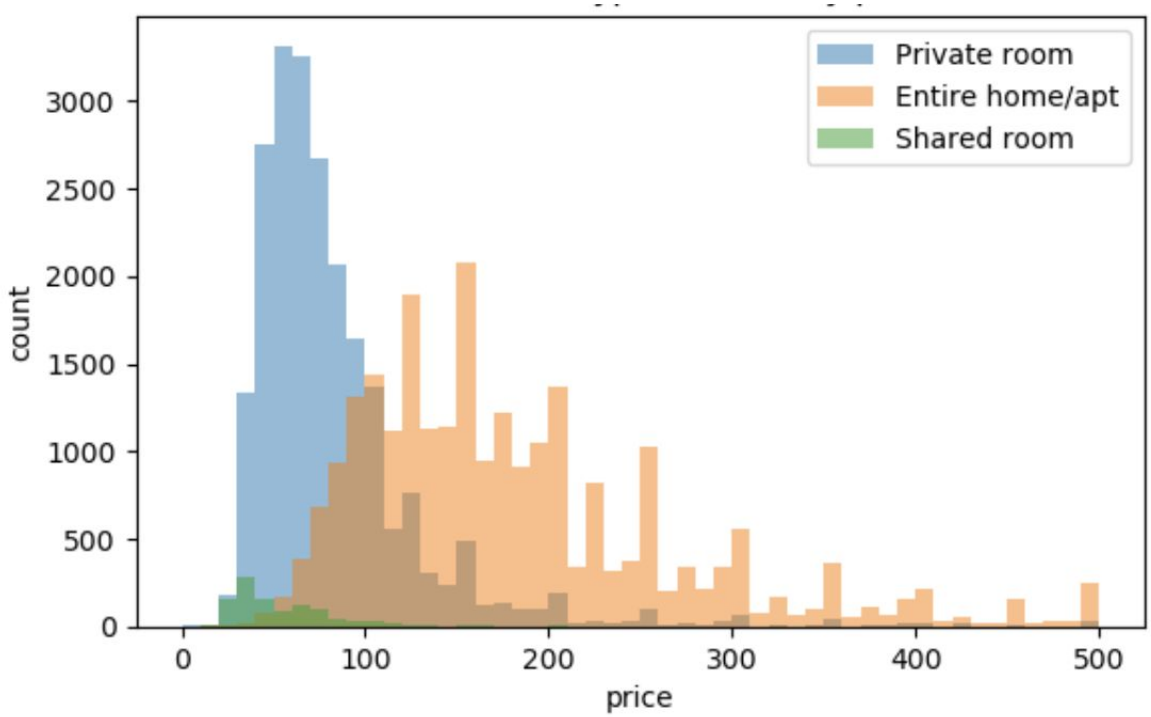

Figure 3 Histogram shows relationship between counts of each type of room and price

2.3 Heatmap

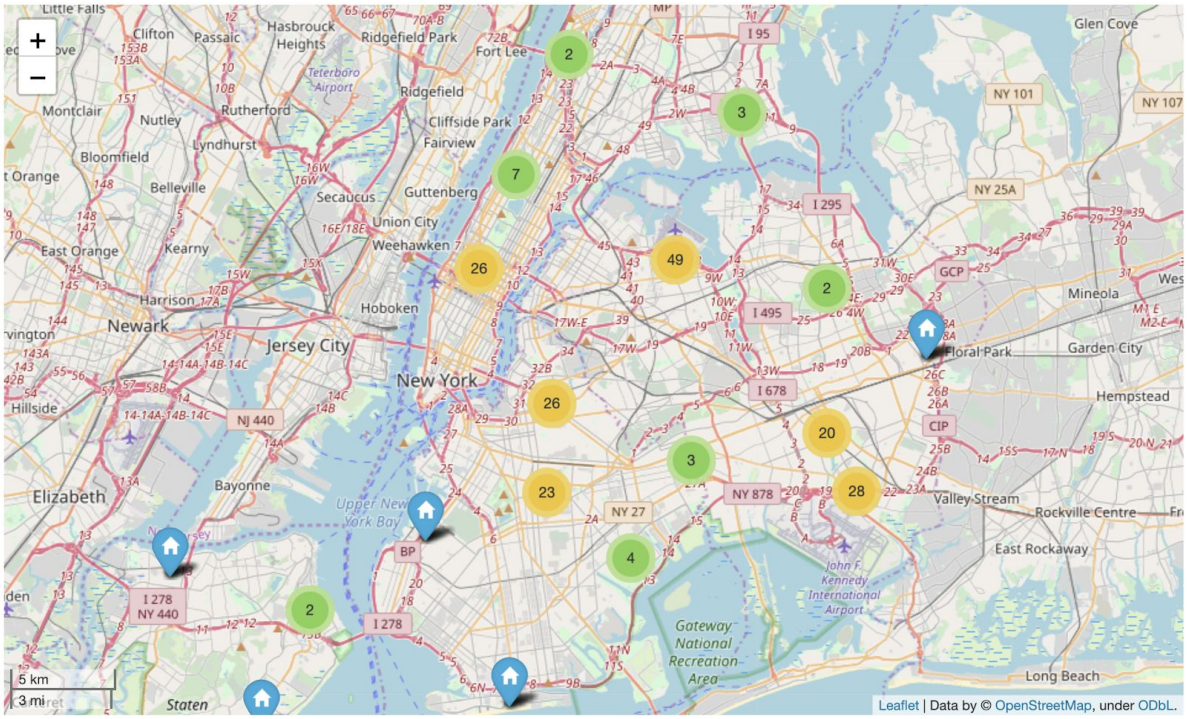

Figure 4 Heatmap shows airbnbs with top 200 reviews_per_month in New York

2.4 Correlation Matrix

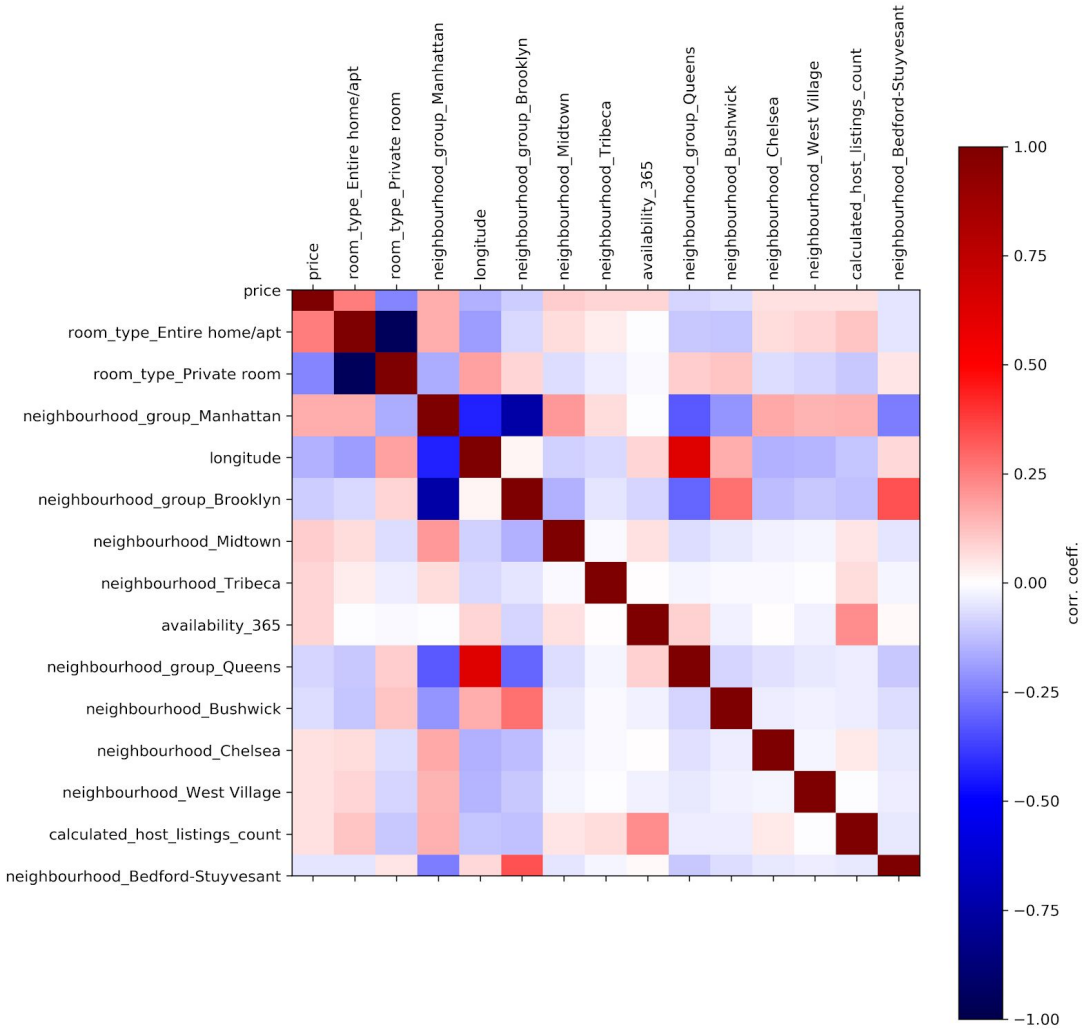

Figure 5 Correlation Matrix shows top 15 correlated features

3. Methods

3.1 Data Preprocessing

I dropped id and host_id, because for id, it represents listing id which is meaningless. For host_id, it is meaningful because one host could have more than one listing. However, calculated_host_listings_count represents the amount of listing per host. So I dropped host_id.

I replaced the missing values in both reviews_per_month and last_review with 0, because by intuition, if there is no reviews per month, it means the reviews per month is 0. And if there is no last review, it means the date for last review does not exist.

Besides, I used '2019-07-09' as the standard date, calculated the number of days between each last_review date and the standard date, stored the new values into new feature which is called time and dropped the original feature last_review. What's more, I also performed NLP technique on name by categorizing each row to positive if it contains at least one of the four words (bedroom, privat, apart, brooklyn), and negative otherwise. I also used package Ethnicolr on host_name by categorizing each name based on its ethnicity and stored them into new feature race.

For data preprocessing, I used OneHotEncoder to categorical features neighbourhood_group, neighbourhood, room_type, name and race. I used MinMaxScaler to continuous features availability_365, latitude and longitude. And I used StandardScaler to continuous features minimum_nights, number_of_reviews, reviews_per_month, calculated_host_listings_count and time. For the target variable price, no label processing is applied. After preprocessing, there are in total 244 features with 48895 rows.

3.2 Machine Learning Pipeline

I tried different machine learning models such as Lasso Regression, Ridge Regression, Random Forest Regressor, Support Vector Regressor and KNeighborsRegressor.

	Paramters	Values
Lasso Regression	alpha	np.logspace(-4,4,num=8)
Ridge Regression	alpha	np.logspace(-6,6,num=21)
Random Forest Regressor	max_depth & min_samples_split	(3,5,7,9) & (2,4,6,8,10)
Support Vector Regressor	gamma & C	(1e-3, 1e-2, 1e-1) & (20, 40, 60)
KNeighborsRegressor	n_neighbors	20,40,60,80,100

Figure 6 Table shows parameters and values for different models

For Lasso Regression, I tuned parameter alpha and tried the values from np.logspace(-4, 4, num=8). For Ridge Regression, I tuned parameter alpha and tried the values from np.logspace(-6, 6, num=21). For Random Forest Regressor, I tuned max_depth and tried the values 3,5,7,9, and I tuned min_samples_split and tried the values 2, 4, 6, 8, 10. For Support Vector Regressor, I tuned gamma and tried the values 1e-3, 1e-2, 1e-1, and I tuned C and tried the values 20, 40, 60. For KNeighborsRegressor, I tuned n_neighbors and tried the values 20, 40, 60, 80, 100.

I used R2 score to evaluate model performance. I used R2 score because it is easily understood. The R2 score for baseline model is 0. The closer the R2 score is to 1, the better the model is.

For the machine learning pipeline, I used k fold cross validation. I obtained the test size to be 20% and chose k folds to be 5. And I also set the random state to be ranged from 23*1 to 23*10. Finally, I calculated the mean and standard deviation of R2 score to measure the uncertainties.

	Mean	Standard Deviation
Lasso Regression	0.118	0.027
Ridge Regression	0.117	0.027
Random Forest Regressor	0.212	0.131
Support Vector Regressor	0.165	0.097
KNeighborsRegressor	0.139	0.07

Figure 7 Table shows the mean and standard deviation of R2 score for each model

4. Results

4.1 R2 Score

The R2 score for baseline model is 0.

	R2 score
Lasso Regression	0.117
Ridge Regression	0.118
Random Forest Regressor	**0.212**
Support Vector Regressor	0.165
KNeighborsRegressor	0.139

Figure 8 Table shows the R2 score for each model

By comparing R2 score, Random Forest Regressor performs the best with R2 score of 0.212, while Lasso Regression performs the worst with R2 score of 0.117 among my five models.

4.2 Global Feature Importance

I calculated the global feature importance and the result is shown below.

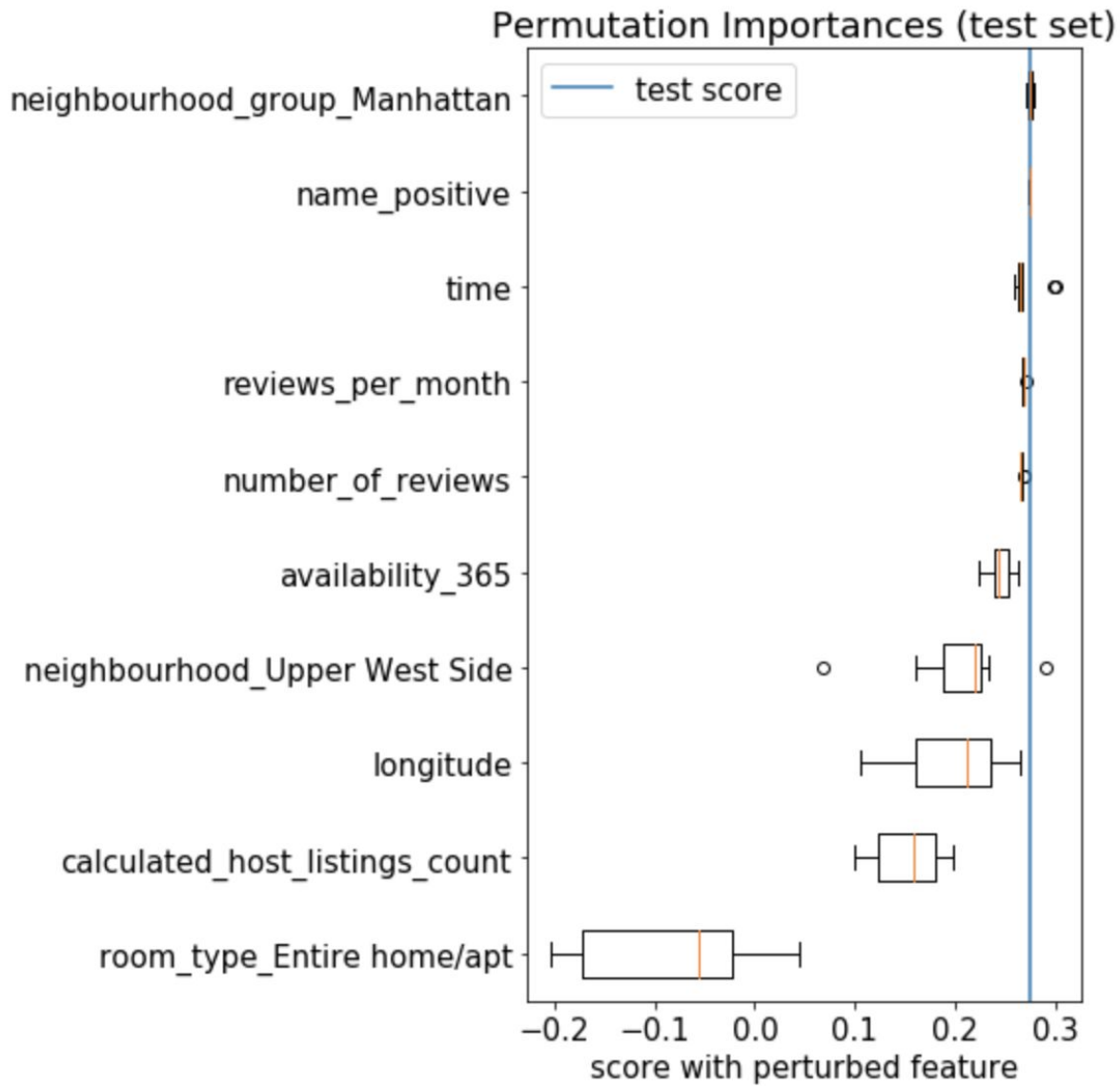

Figure 9 Permutation test for global feature importance from Random Forest Regressor

It shows that room_type_Entire home/apt is the most important feature, which makes sense, because it is an important factor when people book the airbnbs. An entire home or apartment gives people more room and flexibility and people would feel more comfortable and private with an entire apartment. Both calculated_host_listings_count and availability_365 are relatively important as well. It is reasonable because when people book the airbnb in New York, the number of days that an airbnb is available does matter. For example, if a person plans to go to New York in December and he would like to book the airbnb, he could not choose the airbnbs that are not available during the winter periods. And for calculated_host_listings, I would say if a host has more listings, he would probably be more experienced in designing the airbnb layout, which could make people tend to book that host's airbnb with a higher probability. Overall, the results I get fit into a human context.

5. Outlook

There are several ways to improve my model. I could tune other parameters and try different values. What's more, I can use other types of models such as neural networks. Another thing I might try is to implement log transformation on the target variable price during preprocessing, because the range of target variable is large and its distribution is somewhat skewed. I could collect more data and features like whether or not there is public transportation close to airbnb to improve model performance.

6. References

Dataset: **https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data**

Andrew W, Smart Pricing with XGB, RFR + Interpretations, 2019/09
https://www.kaggle.com/jrw2200/smart-pricing-with-xgb-rfr-interpretations

Dgomonov, Data Exploration on NYC Airbnb, 2019/07,
https://www.kaggle.com/dgomonov/data-exploration-on-nyc-airbnb