New York_Airbnb Price Prediction

Chiahui Chen

Introduction

1. Background

Airbnb is a platform to find adventures nearby or in faraway places and access unique homes, experiences, and places around the world. It provides us to share our properties in many types and become the Airbnb host, in other ways, we can also through this platform to find the distinctive place and experience the local style as travelers.

In this Notebook, I'm going to explore the 2020 house source data of Airbnb in New York City.

2. Problem

Pricing is an essential part of Airbnb business. A clear and meaningful pricing system will benefit the business both from the house owner side and the customer side. In this project, I'm going to predict the daily price for the house/apartment/room on airbnb. Besides, I also did the research on the improvement of prediction after implementing the Foursquare Data.

The prediction can be used in the business scene including but not limit to:

1. New house price suggestion
2. Existed house price scanning
3. Market strategy making
4. House segmentation

Data Source

Most listing information in different locations can be found in the Airbnb Open Data Source. There are three datasets in each city and I'm going to use listing.csv to explore the house pricing. In the original csv file, there are over 100 columns, in order to get the usable and reasonable variables, I manually filtered out the columns and eventually kept 34 independent variables for the analysis.

Data preprocessing - Data cleaning

Data cleaning is implemented in the several steps.

1. Missing Value Processing

When checking the data quality, I found the data missing exists in several columns.

```
Out[246]: Unnamed: 0                          0
          id                                  0
          neighbourhood_cleansed              0
          neighbourhood_group_cleansed        0
          zipcode                           437
          latitude                            0
          longitude                           0
          property_type                       0
          room_type                           0
          accommodates                        0
          bathrooms                          72
          bedrooms                          121
          beds                              528
          bed_type                           12
          amenities                           0
          square_feet                     49849
          price                               0
          guests_included                     0
          minimum_nights                      0
          maximum_nights                      0
          availability_30                     0
          availability_60                     0
          availability_90                     0
          availability_365                    0
          number_of_reviews                   0
          review_scores_rating            12006
          review_scores_accuracy          12040
          review_scores_cleanliness       12026
          review_scores_checkin           12053
          review_scores_communication     12038
          review_scores_location          12056
          review_scores_value             12057
          cancellation_policy                 0
          reviews_per_month               11030
          dtype: int64
```

I considered ‘beds’ as a crucial input in the first place, so I use the formula “beds = (accommodates + 1) // 2)” to fill the Null value. Under our investigation and data understanding, proper default value is given to the columns “bathrooms”, “bedrooms” and “bed_type”.

For the eight attributes related to review, I cannot judge the feature importance at this step. Therefore I will fill the Null value with the mean value of the group.

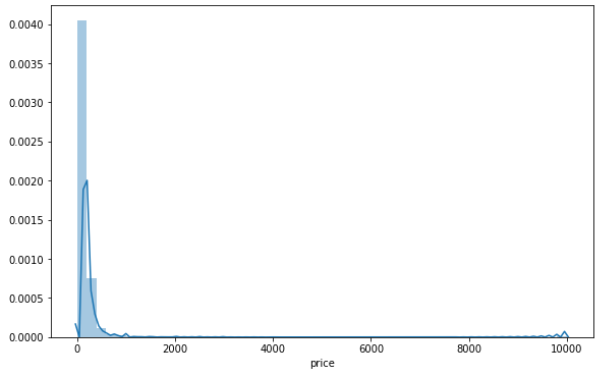
2. Drop redundant data

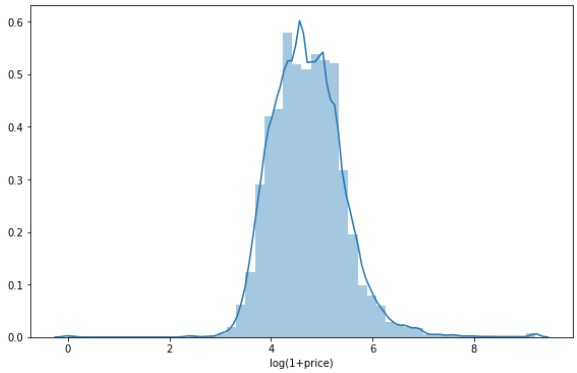
Despite the data quality being fairly high, I decide to drop redundant data to keep the data precise. In this step, ‘square_feet’ is dropped because of a large proportion of

missing data, plus no idea on how to refill them. Besides, ‘zipcode’ is dropped because it is repeated with the latitude and longitude.

Feature selection

1. Price transformation





According to the data analysis on the price, the distribution is skeld on the smaller side. Holver, by transforming the price into its log value, the distribution shows more like a normal distribution, which is more acceptable for a regression model.

2. Categorical data transformation

There are seven categorial metrics in the datasets. Although one hot code is one the easiest way to flatten the data, it will add too many columns in the dataset (where I want to leave those quotes to ‘Amenities’). I found it was meaningful to transform the categorical data into hierarchy values in this problem.
Take ‘cancellation_policy’ as an example, the unique values include 'flexible', 'moderate', 'strict_14_with_grace_period', 'strict', 'super_strict_30', 'super_strict_60', which I can use our prior knowledge to sort them based on strictness. To realize the transformation, I can make a dict like
cancellation_policy_dict={'flexible':1,'moderate':2,'strict_14_with_grace_period':3,'strict':4, 'super_strict_30':5,'super_strict_60':6}
I can transform other categorical data in a similar way.

3. Unzipped Amenities

Amenities is an interesting column which contains the available assets and features for the room/house such as ‘Cable TV’, ‘Pets Allold’, ‘Air Conditioner’ and so on. Those features will relate to the house price in our common sense. To unzip the column, I scanned the whole dataset to create a set containing every single feature. I then flattened the data and checked the correlation with price to get the top features. The result is like:

```
Family/kid friendly 0.17717748280424178
Paid parking off premises 0.10122791949871808
Iron 0.11107985389459528
TV 0.2619452838347856
Cable TV 0.20333477776334521
Bathtub 0.12358691689776173
Hair dryer 0.13149601478982287
Pets allowed 0.10992090959428745
Air conditioning 0.2041364740672369
High chair 0.10275908993191595
Children's books and toys 0.10052901994436188
Shampoo 0.11587834590427842
Lock on bedroom door -0.1876099956038007
Pack 'n Play/travel crib 0.1340300044728197
```

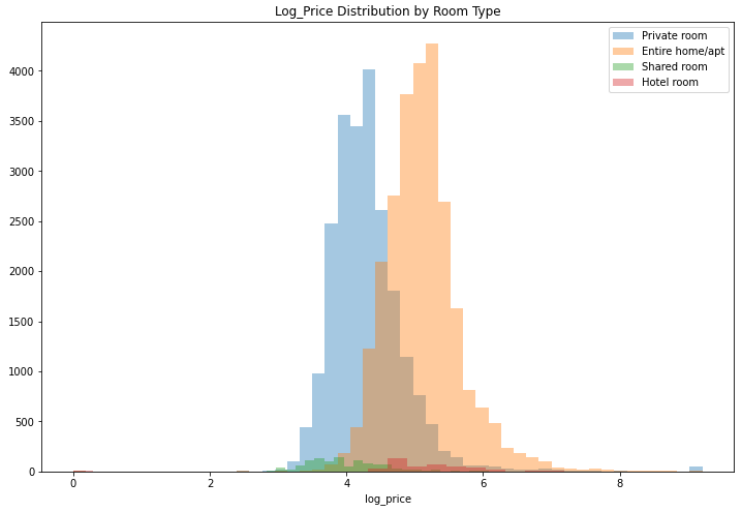
Feature Scaling

1. Robust Scaling

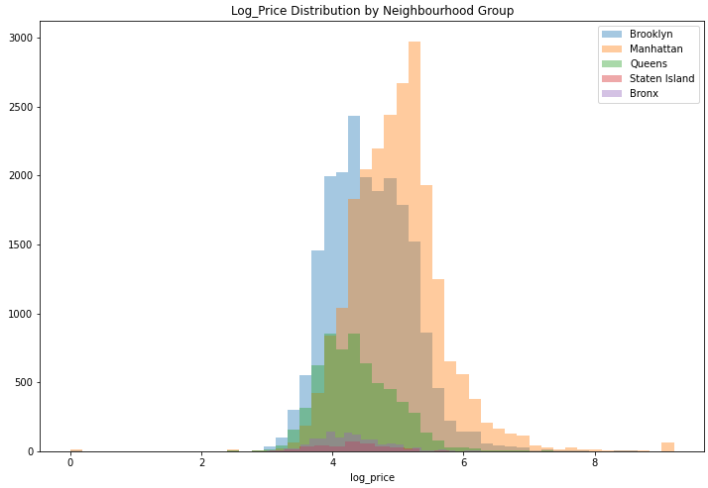
Scaling is a necessary step to keep each variable distributed in a proper and comparable interval before the modeling step. In this project, standard scaling is not recommended as I observed that data is mostly distributed skeld on one pole. Instead, Robust scaling is implemented to both compress the data on the sparse side and scaling the data on the dense side, which is proved to work in the modeling.

Exploratory Data Analysis

1. Relationship betlen Room_Type and Log_Price
In the figure below, I can obviously find that there is a significant difference in price range among these room types and the peaks of the distribution among each type are Hotel Room >= Entire home/apt > Private Room > Shared Room. In the next section, the result of this exploration could help us to process the categorical data.

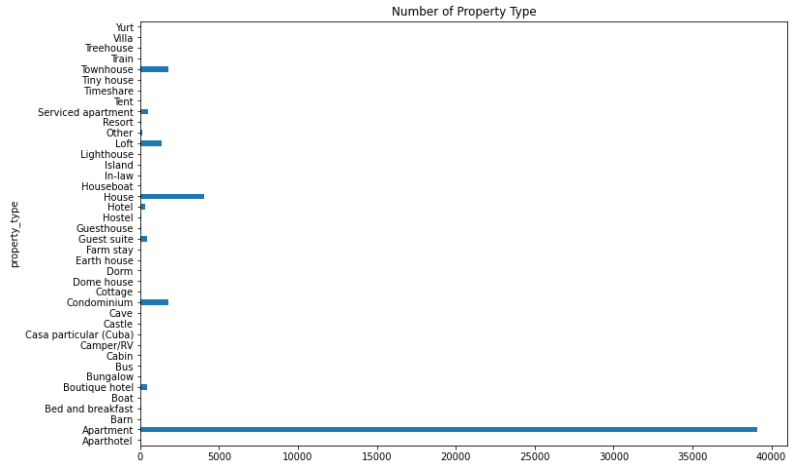


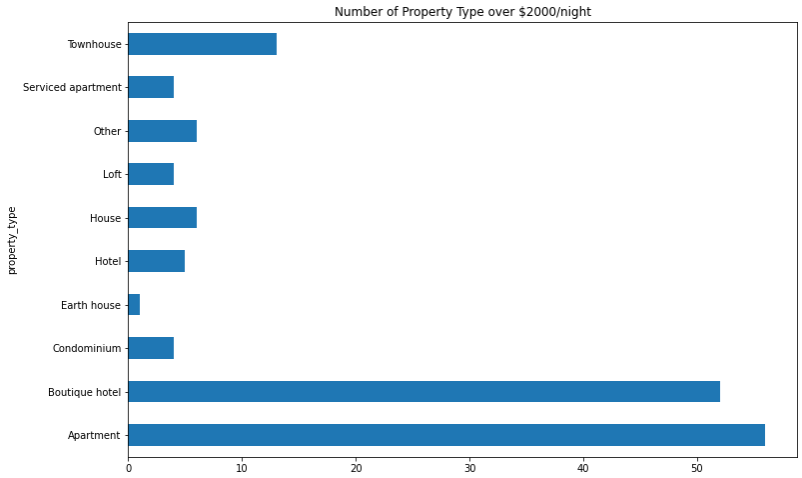
2. Relationship betlen location and price
In the figure below, I can assert that the location would affect the house pricing and the properties in Manhattan tend to have the higher price.



3. Relationship betlen Property Type and Price

It is widely accepted that less than 25% of property types would have a higher housing price and boutique hotels can be labeled as the luxurious place. I don’t specify the apartment as the expensive place since the base number of apartment type is way larger than boutique one.





Predictive Modeling

There are many regression models that can be used to predict the housing price and I'm going to apply XGBoost and Random Forest Regression models to our dataset.

1.XGBoost
I set n_estimator to 3000 times to train the model iteratively with early_stopping_rounds at 150 times based data size.

Performance of the XGBoost model.

RMSE of Test Dataset	MSE of Test Dataset	MSE of Train Dataset
0.390451	0.071667	0.152452

2. Random Forest Regressor

RMSE of Test Dataset	MSE of Train Dataset	MSE of Test Dataset	CV_rfr_r2
0.4491	0.0231	0.1639	0.6753

I also used GridSearch to find the better parameter for optimizing the regressor, holver, the result was even worse than the previous one.

```
{'max_depth': 9,
 'max_features': 13,
 'min_samples_leaf': 6,
 'min_samples_split': 2,
 'n_estimators': 10}
```

Top 10 Important Features to Affect the Housing Price

room_type	**0.266747**
longitude	**0.123649**
accommodates	**0.112937**
latitude	**0.080912**
bedrooms	**0.075502**
neighbourhood_group_cleansed	**0.043307**
bathrooms	**0.040523**
minimum_nights	**0.028554**
reviews_per_month	**0.019660**
maximum_nights	**0.018640**

Continue Exploration with Foursquare Data

1. Get New York Data from https://cocl.us/new_york_dataset
2. Load the json

	Borough	Neighborhood	Latitude	Longitude
0	Bronx	Wakefield	40.894705	-73.847201
1	Bronx	Co-op City	40.874294	-73.829939
2	Bronx	Eastchester	40.887556	-73.827806

3. Using Foursquare to Get top50 Venues in Each Neighborhood

I used Foursquare Explore to find the top 50 venues in the radius of 2000 meters , holver, I limited the category of the venues in the list below since the search range is too wide and it will exceed the limitation of requesting the calls on Foursquare .

 # 4deefb944765f83613cdba6e, Historic Site
 # 4bf58dd8d48988d12d941735, Monument / Landmark
 # 4bf58dd8d48988d164941735, Plaza
 # 4bf58dd8d48988d181941735, Museum
 # 4bf58dd8d48988d184941735, Stadium
 # 4bf58dd8d48988d1ae941735, University
 # 4bf58dd8d48988d1f9941735, Food & Drink Shop
 # 4d4b7105d754a06379d81259, Travel & Transport
 # 4bf58dd8d48988d1fa931735 Hotel

Implement New York Data to Airbnb Data

1. Selecting the usable columns from the Foursquare dataset

	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Enterprise Rent-A-Car	40.896091	-73.865364	Rental Car Location
1	Metro North - Woodlawn Train Station	40.895591	-73.862814	Train Station
2	Quality Rent A Car	40.892660	-73.854805	Rental Car Location
3	Bx16 Bus Stop	40.898490	-73.854182	Bus Station
4	Bee Line 42 MTA NYCT BX39 MTABus BXM11 (White ...	40.898214	-73.854533	Bus Station

2. Assign Category Type to Venue Category

After examining the Foursquare dataset, I found that Venue Categories are too detailed and many of them indicate the same purpose of action, thus I simply created four types to classify the venue categories.

```
venue_category_list =['Rental Car Location', 'Train Station', 'Bus
Station',
      'Metro Station', 'Bus Stop', 'Hotel','Boat or Ferry',
      'Tourist Information Center','Travel & Transport', 'Pier', 'Motel',
      'Bike Rental / Bike Share','Resort', 'Heliport',
      'Hostel','Bed & Breakfast', 'Light Rail Station','Airport Terminal',
      'Port', 'Cruise', 'Taxi Stand']
type_dict =
{'Nearby_hotel':['Motel','Hotel','Hostel','Bed & Breakfast','Resort'],
 'Nearby_air_train':['Train Station','Travel&Transport','Heliport','Metro
Station','Airport Terminal','Light Rail Station'],
 'Nearby_bus_taxi_ship':['Rental Car Location','Bus Station','Bus
Stop','Boat or Ferry','Pier','Bike Rental / Bike Share','Port', 'Cruise',
'Taxi Stand'],
 'Nearby_tour':['Tourist Information Center']}
```

	Venue	Venue Latitude	Venue Longitude	Venue Category	Venue Category Type
0	Enterprise Rent-A-Car	40.896091	-73.865364	Rental Car Location	Nearby_bus_taxi_ship
1	Metro North - Woodlawn Train Station	40.895591	-73.862814	Train Station	Nearby_air_train
2	Quality Rent A Car	40.892660	-73.854805	Rental Car Location	Nearby_bus_taxi_ship
3	Bx16 Bus Stop	40.898490	-73.854182	Bus Station	Nearby_bus_taxi_ship
4	Bee Line 42 MTA NYCT BX39 MTABus BXM11 (White ...	40.898214	-73.854533	Bus Station	Nearby_bus_taxi_ship

3.Define the Function to Calculate the Distance

Before calculating the distance betlen airbnb places and venues, I defined the function to transform the distance from longitude and latitude.

4. Number of Nearby Venues for Each Airbnb Place

It’s time to combine the airbnb data with the foursquare data. First of all, I created the four columns in the Airbnb dataset with the name of types above. Then, I calculated the number of venues under each type for each Airbnb place and the radius was within 500 meter. The cost of calculating here is considerable and took a lot of time as I’ll so that’s why I only set the radius at 500m.

Indoor fireplace	Nearby_hotel	Nearby_air_train	Nearby_bus_taxi_ship	Nearby_tour
0	0	3	2	0
0	112	0	0	0
0	0	3	0	0
0	37	0	2	0
0	1	2	0	0

XGBoost Prediction after Adding Foursquare Data

Before using the XGBoost model, I processed the data including assigning the independent variables and dependent variables, applying the data with Robust Scaler and split the data to train and test the dataset.
And now, it is time to reveal whether the Foursquare Data would help with the prediction.

RMSE of Test Dataset	MSE of Test Dataset	MSE of Train Dataset
0.3904625	**0.068508**	**0.152461**