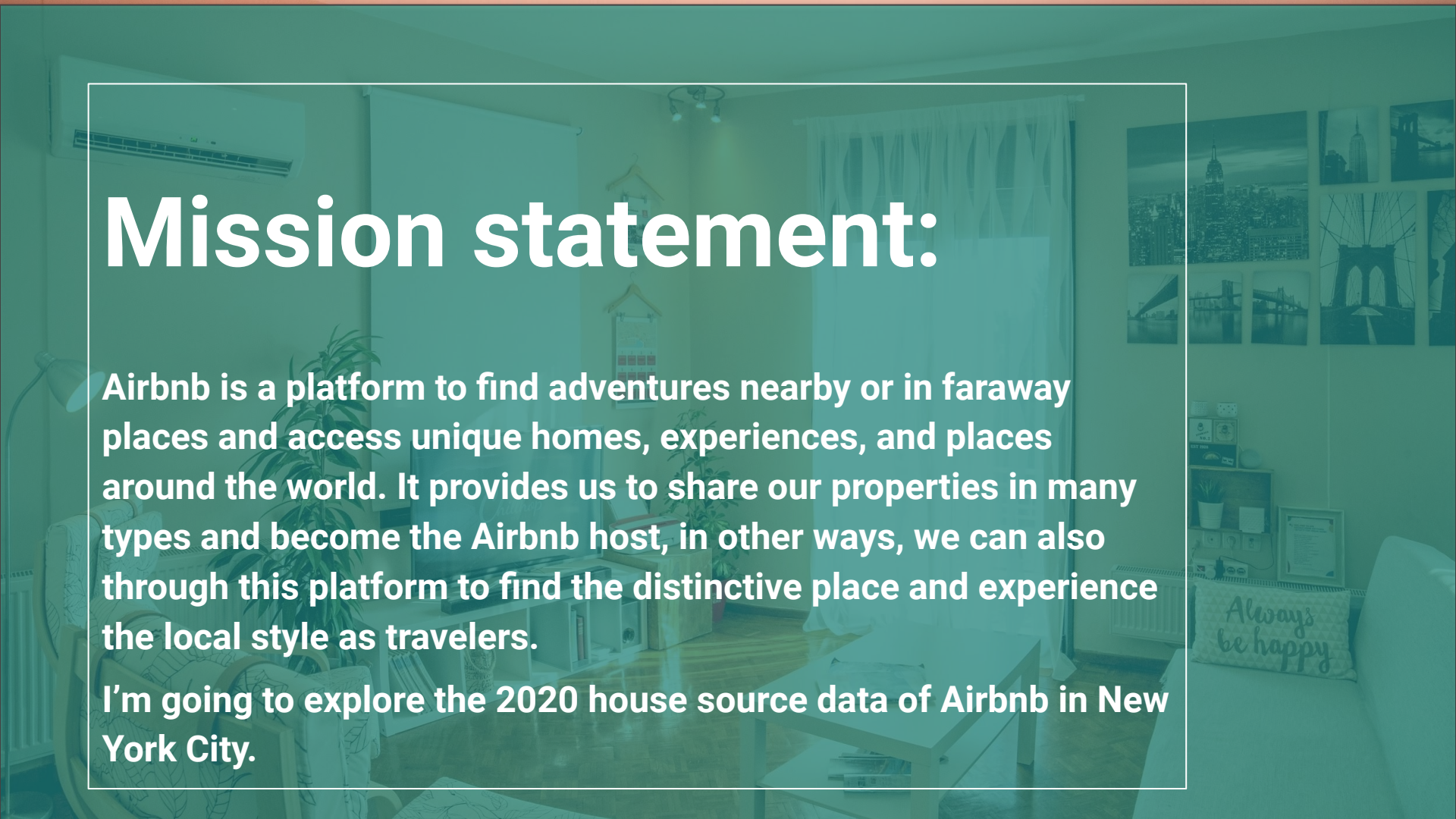

# Airbnb New York City

## Price Prediction

Chiahui Chen




# Mission statement:

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.

I’m going to explore the 2020 house source data of Airbnb in New York City.

# The problem


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.

Scenario of using pricing system

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

# Data Source & Cleaning

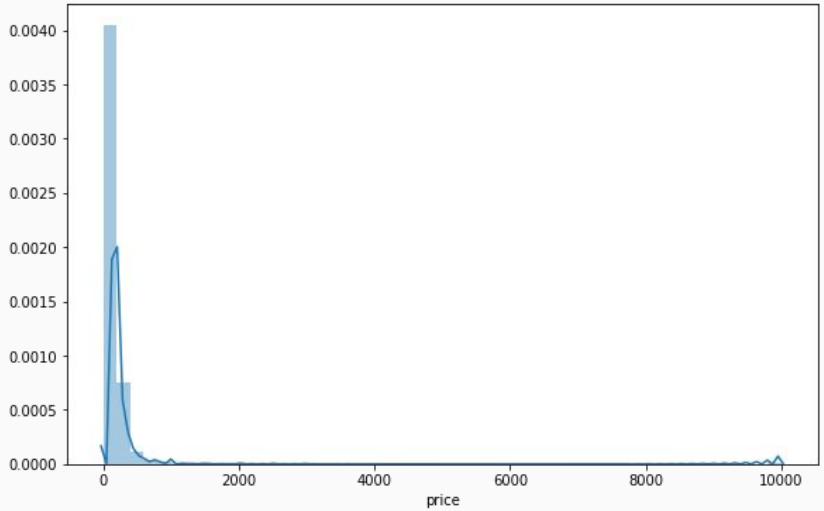

1. Data Source - Airbnb Open Data Source
   I'm going to use listing.csv to explore the house pricing. Before start the analysis, I manually filtered out the columns and only kept 34 independent variables for the analysis.

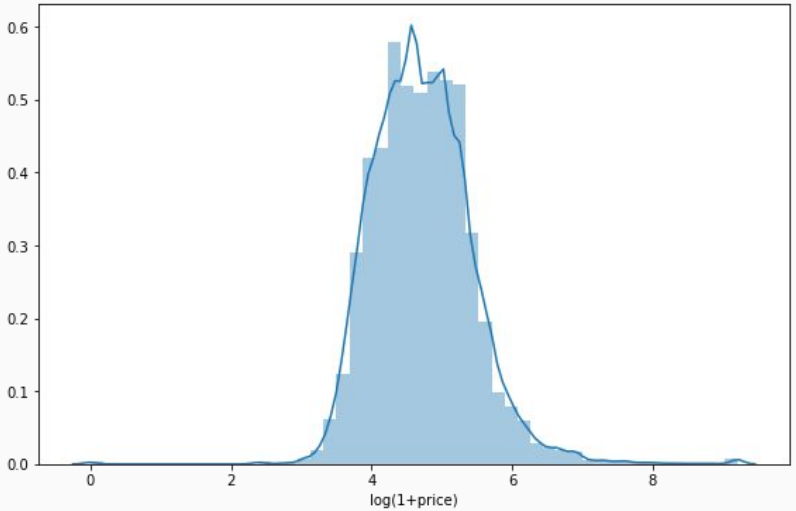
2. Missing Value Processing

3. Drop redundant data
   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
   The price distribution is skeld on the smaller side so I transferred the price to log(1+price) which is much closed to normal distribution.



# Feature selection & scaler

## 2. Categorical data transformation

Though onehot code is the popular way , it was more meaningful to transform the categorical data into hierarchy values in this problem.
Ex:
‘cancellation_policy’
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.

## 3. Unzipped Amenities Columns

To unzip the column, I scanned the whole dataset to create a set containing every single feature, then flattened the data and checked the correlation with price to get the top features.

## 4. Feature Scaler - Robust scaling

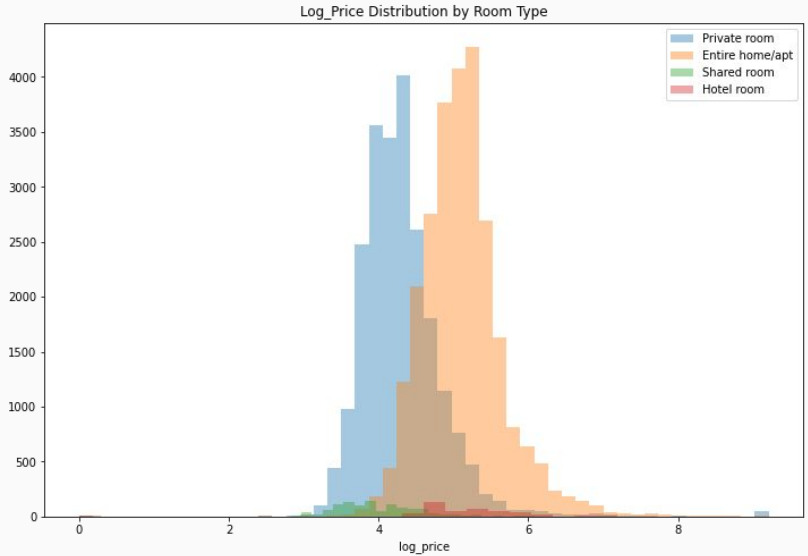
It is implemented to both compress the data on the sparse side and scaling the data on the dense side

| amenities |
| --- |
| {Internet,Wifi} |
| {TV,Wifi,"Air conditioning",Kitchen,"Paid park... |
| {TV,"Cable TV",Internet,Wifi,"Air conditioning... |

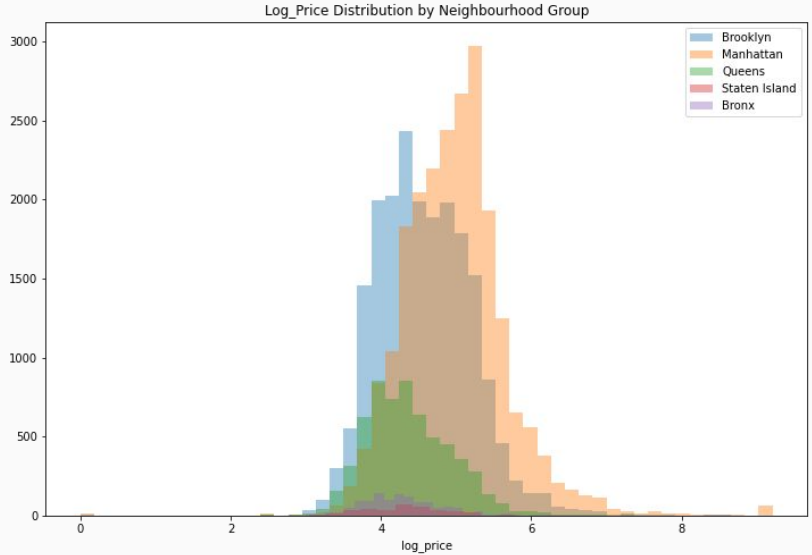
```
Family/kid friendly 0.17717748280424178
Paid parking off premises 0.10122791949871808
Iron 0.11107985389459528
TV 0.2619452838347856
Cable TV 0.20333477776334521
Bathtub 0.12358691689776173
Hair dryer 0.13149601478982287
Pets allowed 0.10992090959428745
Air conditioning 0.2041364740672369
High chair 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
```

# Data Exploration

Relationship betIen Room_Type and Log_Price




Relationship betIen location and price



# Predicting Modeling


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 | 0.0717 | 0.1524 |

2. Random Forest Regressor

I tried to used GridSearch to find the better parameter for optimizing the regressor, but the result was even worse.

Performance of the Random Forest Regressor.

| RMSE of Test Dataset | MSE of Train Dataset | MSE of Test Dataset | R2 |
|---|---|---|---|
| 0.4491 | 0.0231 | 0.1639 | 0.6753 |

# Continue Exploration with Foursquare Data

Whether the prediction will be better if we implement the data from Foursquare……..

# Exploration with Foursquare Data

1. Get New York Data from https://cocl.us/new_york_dataset

| | 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 |

2. Using Foursquare to Get Top 50 Venues in Each Neighborhood in the radius of 2000 meters.

| | 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 |

# Exploration with Foursquare Data

3. Assign Category Type to Venue Category

The original Venue Category contains 21 different groups but part of them have the similar purpose of action. 21 different groups were assigned to ‘Nearby_hotel’, ’Nearby_air_train’, ‘Nearby_bus_taxi_ship', ’Nearby_tour’.

| | 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 |

4. Define the Function to Calculate the Distance

5. Number of Nearby Venues for Each Airbnb Place

I calculated the number of venues under each type for each Airbnb place and the radius was within 500 meter.

| 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

Processing : assigning the independent variables and dependent variables, applying the data with Robust Scaler and split the data to train and test the dataset

Performance of the XGBoost model

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

# Conclusion

There is no significant improvement after we adding the Foursquare data.

Performance of the XGBoost model

| RMSE of Test Dataset | MSE of Test Dataset | MSE of Train Dataset |
|---|---|---|
| **0.390451** | **0.071667** | **0.152452** |

Performance of the XGBoost model with Foursquare Data

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