Airbnb Price Prediction

Andy Dai, Zhe Zhou, Loreto Villalba Rubio

Introduction

- Provide an analysis of Airbnb, specifically at New York City area
- To mine the data and uncover interesting observation about the different hosts and areas
- Examine the data, analyze outliers and clean the data
- Price variation based on different factors

Data Exploration

Variables in 2019 Airbnb New York City booking data set:

- Neighborhood
- Host name
- ID number
- Customer name
- Location by Latitude and Longitude.
- Room Type
- Price difference
- Host’s availability
- Days stayed and reviews

Preprocess the dataset:

- Tested out the “price” and “availability_365” column to find out anomalies
- Removed the data that had either of these (N/A or 0) as part of cleaning the data since the price and the number of days when listing is available for booking will not be zero

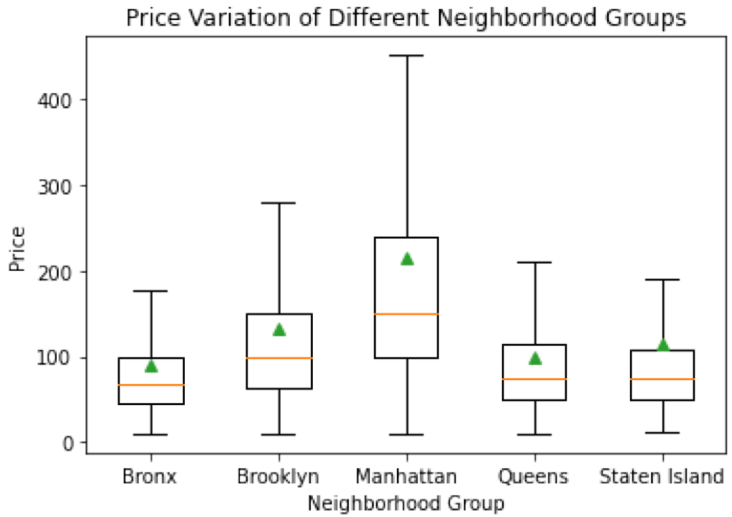
Price vs. Neighborhood

- Group the dataset by the column “neighbourhood”
- Remove any group that size is less than 5
- Filter down to median price in each group
- Print out top 5 and bottom 5 neighborhood based on the price of the Airbnb in that neighborhood

neighbourhood	price
Concord	34.5
Castle Hill	39.0
Hunts Point	40.0
Corona	40.0
Tremont	41.0

neighbourhood	price
Tribeca	309.0
Flatiron District	299.0
NoHo	250.0
Midtown	225.0
West Village	218.0

Price vs. Neighborhood

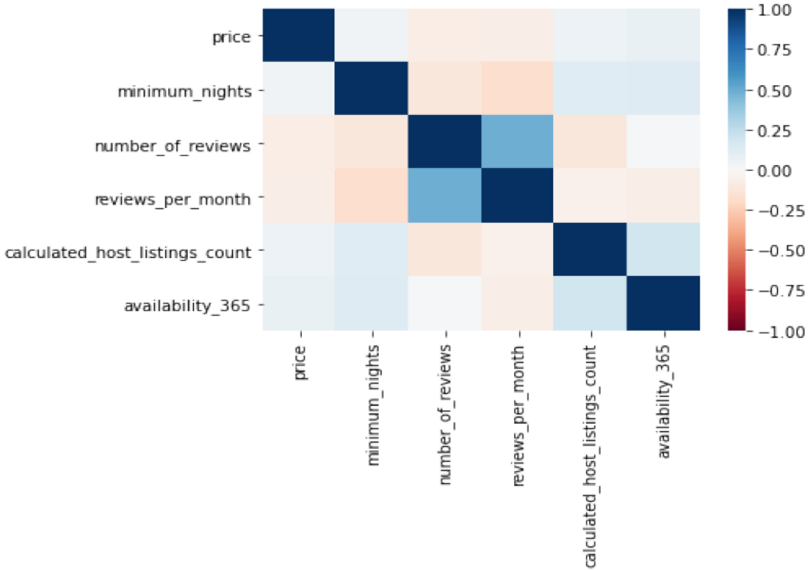


- We used boxplot for all neighborhood groups that are created
- We eliminated any luxury Airbnb booking which may affect the outcome as outliers
- According to the chart, Manhattan has the most expensive Airbnb among all neighborhood groups (also highest median and mean price)
- The Airbnb price and price variation will increase in Manhattan

Correlation Analysis

To determine the correlation we created a heat map:

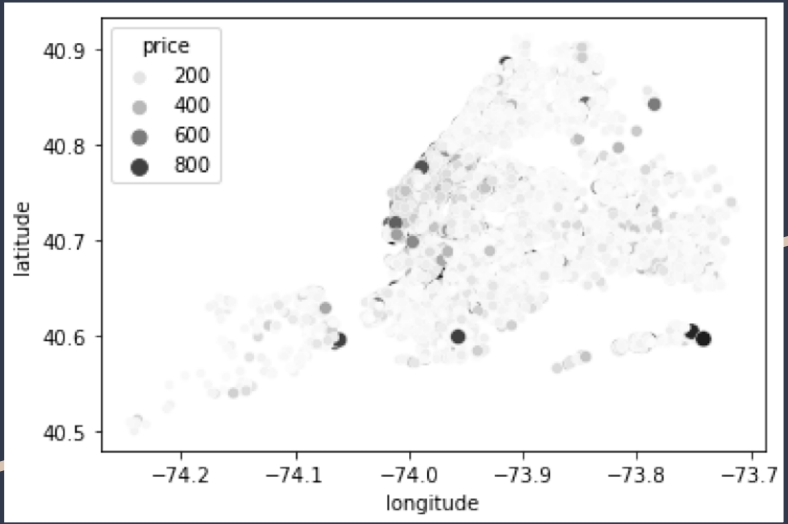
- Using “Price”, “Minimum_nights”, “Number of reviews”, “Reviews per month”, “Host listings count” and “Availability_365”
- Darker color implies the absolute value of the correlation coefficient is closer to 1 or -1



According to the heatmap, find correlation between “price” and other variables:

- the “number of reviews” and “reviews per month” have a negative correlation to “price”
- “minimum nights”, “calculated host listings count” and “availability” have a positive correlation to “price”

Price vs. Longitude & Latitude



We want to figure out the association between longitude & latitude and price:

- Remove any data whose price is greater or equal to $1000
- Create a scatter plot where x value is the column "longitude", and the y value is the column "latitude"
- The color and size of the points bases the value of "price" (The higher the price, darker the point will be)

According to the scatter plot, we can figure out in which neighborhood groups price will be higher:

- Manhattan has the most expensive booking among all neighborhood groups

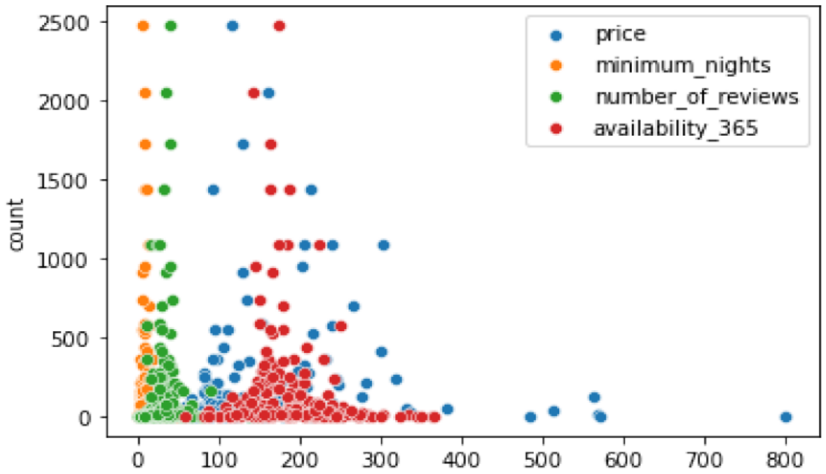
Busiest Area

We also want to consider whether the busiest area will have impact or relation on neighborhood pricing:

- We found that there's no linear relationship between price, min nights, number of reviews and availability using the scatter plot
- If the price of an area is higher, more hosts want to make listings there. Also, more reviews imply a higher number of customers
- From the data, Bedford-Stuyvesant is the busiest area cause it has the highest number of listings. With Average price of $115.35 and mean # of reviews of 40

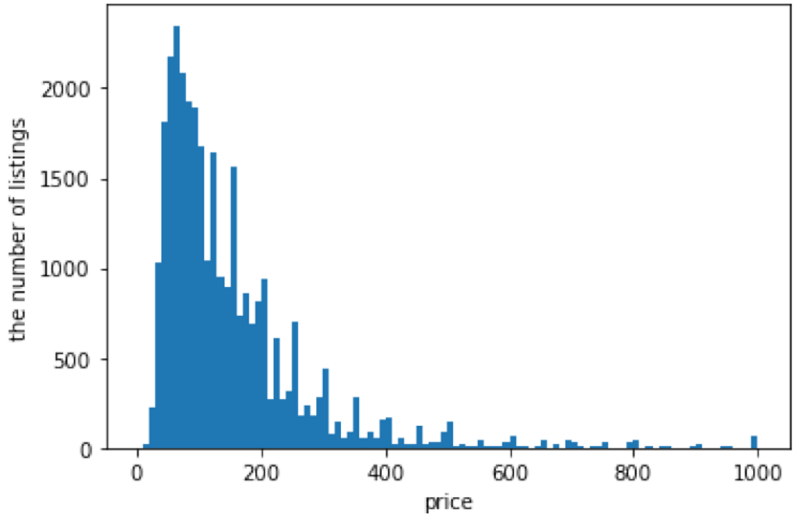
neighbourhood	count	price	minimum_nights	number_of_reviews
Bedford-Stuyvesant	2478	115.354722	6.350686	39.598870
Williamsburg	2051	161.171136	7.753291	35.203315
Harlem	1734	129.643022	7.662053	38.734141
Bushwick	1447	91.409122	6.852799	31.941949
Hell's Kitchen	1446	213.183264	9.569848	31.527663

neighbourhood	availability_365
Bedford-Stuyvesant	174.545198
Williamsburg	142.779132
Harlem	163.369666
Bushwick	162.447132
Hell's Kitchen	188.009682



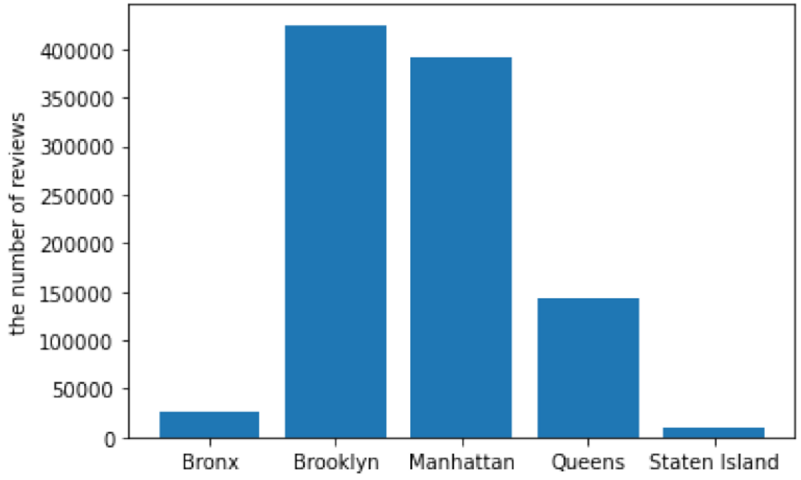
Interesting Finding

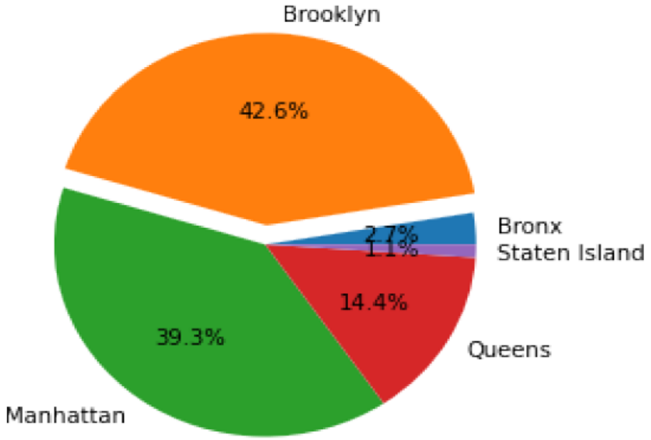
According to the data, prices of most listings are between $0 and $200, with a very small proportion of all listings in the dataset will have their price of more than $200



Interesting Finding

According to the data “the number of reviews”, Brooklyn has the highest number of reviews, with Manhattan right behind it at 39.3% of the market share in NYC





Classification of Airbnb Price Prediction

In this part, we decide to build a random forest regression model to do the classification of Airbnb Price Prediction

Package and Function used in Python:

- “DecisionTreeClassifier( )” in **sklearn.tree**

Variables used to fit the regression model (selected from previous analysis):

- “price”
- "latitude"
- "longitude"
- "room_type"
- "reviews_per_month"
- "calculated_host_listings_count"
- "availability_365"

Since some variables are categorical, we need to convert them into numeric values before fitting the random forest regression model

For “price”, according to mean value in “price” variable, we decide to separate it into 3 types:

- low (0): less than $100
- medium (1): between $100 and $200
- high (2): above $200

In the random forest model, we don’t need to consider if a categorical variable is nominal or ordinal. However, the function require numeric input, so we need to use numeric values to represent each type in categorical variables.

For “room_type”, use 0 to represent shared room, 1 to represent private room, and 2 to represent entire home/apt.

For the random forest regression model, we need to decide the value of depth. The goal is to maximum the accuracy and make depth as small as possible.

The algorithm on the right side fit the model by different values of depth from 2 to 10, and pick the depth that has a significant effect on the accuracy of the regression model.

So, if the accuracy of the model increase 0.01, we will consider this increase as significant.

In the end, the algorithm tells the best depth is 6, and the accuracy of the model is 58.92%

```
# Find optimal depth for random forest regression model
depth = 0
accuracy_score = 0
for i in range(2, 10):
    CLF = DecisionTreeClassifier(criterion="entropy", max_depth=i)
    CLF = CLF.fit(X_train, y_train)
    y_pred = CLF.predict(X_test)
    if metrics.accuracy_score(y_test, y_pred) - accuracy_score > 0.01:
        depth = i
        accuracy_score = metrics.accuracy_score(y_test, y_pred)

# Model evaluation
CLF = DecisionTreeClassifier(criterion="entropy", max_depth=depth)
CLF = CLF.fit(X_train, y_train)
y_pred = CLF.predict(X_test)
print("Depth:", depth)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

```
Depth: 6
Accuracy: 0.589242053789731
```

Python Codes & Results

Regression of Airbnb Price Prediction

Package and Function used in Python:

- “LinearRegression( )” in **sklearn.linear_model**

Variables used to fit the linear regression model (selected from previous analysis):

- “price”
- "neighbourhood_group"
- "room_type"
- "minimum_nights"
- "number_of_reviews"
- "reviews_per_month"
- "calculated_host_listings_count"
- "availability_365"

Since some variables are categorical, we need to convert them into numeric values before fitting the linear regression model

For "neighbourhood_group", since it’s a nominal variable, so it cannot be converted to numeric values directly. We replace it by four variables “Brooklyn”, “Manhattan”, “Queens”, and “Bronx”, and let 1 represents “Yes”, 0 represents “No”. Note that, all 0 represents “Staten Island”.

For “room_type”, we consider it as a ordinal variable, so we can convert it to numeric values directly. We use 0 to represent shared room, 1 to represent private room, and 2 to represent entire home/apt.

```python
# Using Cross Validation to fit the model
coef = []
MSE = []
for i in range(100):
    # Seperate dataset
    train, test = train_test_split(data_P9)

    # Fit model without Cross Validation
    LR = LinearRegression()
    y = train.iloc[:,0]
    X = train.iloc[:,1:train.shape[1]]
    LR.fit(X, y)
    coef.append([LR.intercept_] + LR.coef_.tolist())

    # Calculate MSE and RMSE
    y_test = test.iloc[:,0]
    X_test = test.iloc[:,1:test.shape[1]]
    y_pred = LR.predict(X_test)
    MSE.append(np.mean(np.square(y_test - y_pred)))

coef_ = np.asarray(coef).mean(axis = 0)
MSE = np.mean(MSE)
RMSE = np.sqrt(MSE)

var_name = ["Intercept"] + list(data_P9.columns)
for i in range(len(coef_)):
    print(var_name[i] + " " + str(coef_[i]))
print()
print("MSE:", MSE)
print("RMSE:", RMSE)
```

Below is the coefficients of the fitted multiple linear regression model:

```
Intercept -63.532004107849254
price 25.291611580649814
Brooklyn 90.99603093800376
Manhattan 5.127678358742941
Queens -6.704862674747806
Bronx 108.37827900137567
room_type -0.1458467155796938
minimum_nights -0.26612008211416066
number_of_reviews -4.503830767598092
reviews_per_month -0.14234354945251831
calculated_host_listings_count 0.17797504411992743
```

Mean Square Error (MSE) for this linear regression is 58517.31; Root Mean Square Error (RMSE) for this linear regression is 242.90

Python Codes & Results

Discussion

Deficiency in the classification and regression of Airbnb price prediction

- RMSE of the linear regression model is around 250, so errors will be large when using this model to predict the price
- Not perform detailed variable analysis before fitting the model, so some variables may not very significant to price
- Not consider the interaction between different variables
- For classification, classifying randomly will make a 33% accurate prediction, so the accuracy of the random forest regression model may not be acceptable
- Only do classification and regression on price, clustering may be a better approach method

Thank You !