Airbnb dataset description

Mohammad Javad Aghaie

Contents

Task1: Handling Missing Data & Outliers

Missing Data

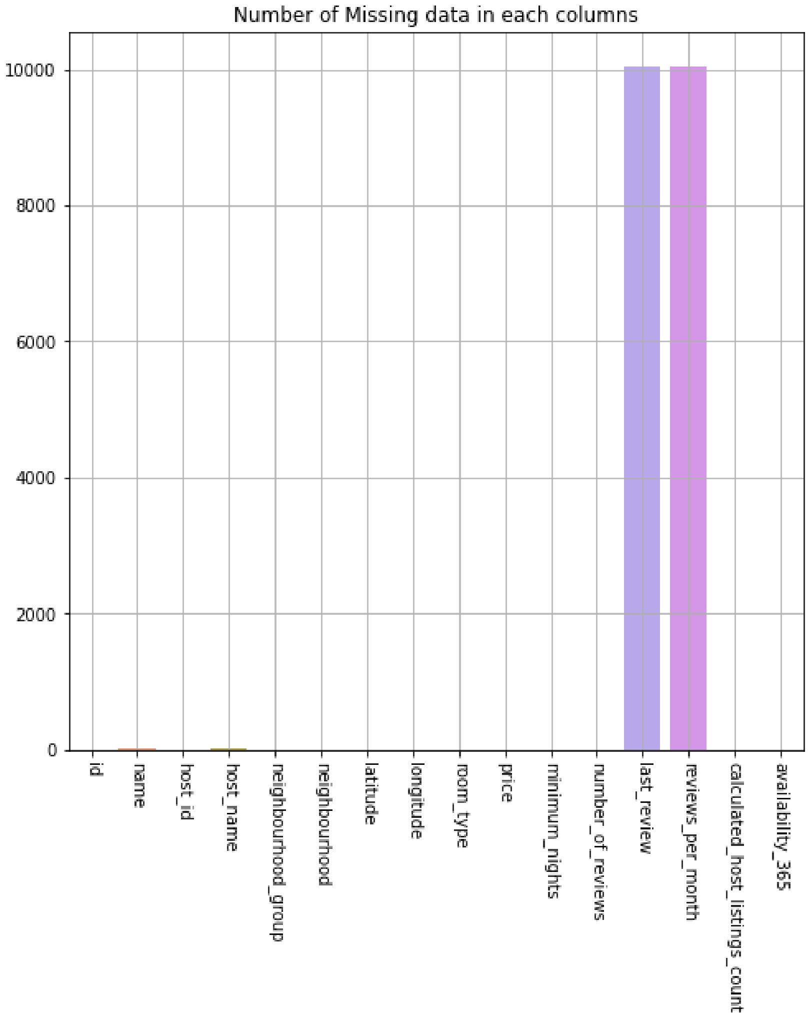


Figure 1: Bar Plot of Number of Missing data

This dataset contains 16 attributes (columns) and 48895 samples (rows). It contains some missing data. As it can be seen in Figure 1, four features (columns) have missing data. "Last view" and "reviews per moths" has the most missing data. There are many ways to deal with missing data. We can omit the samples but it also deprives us from valuable information other features gives us. As a result, I have used Panda's imputation with "pad" method. Another famous and powerful method for quantitative values is KNN.

Outliers

For Dealing with outliers, we need to diagnose them. On the other hand, because our goal is to predict Price, we find outliers based on Price feature (column). There are many ways for finding and handling. I used IQR due to its simplicity. I plotted Price in a boxplot (Figure 2). As it can be seen, there are a lot of outliers. First, we need to find Q1 and Q3 which are related to first and third quartiles, respectively. The IQR is difference between Q1 and Q3. With the help of IQR, we can find upper and lower bounds. As a result, we can find outliers and delete them.

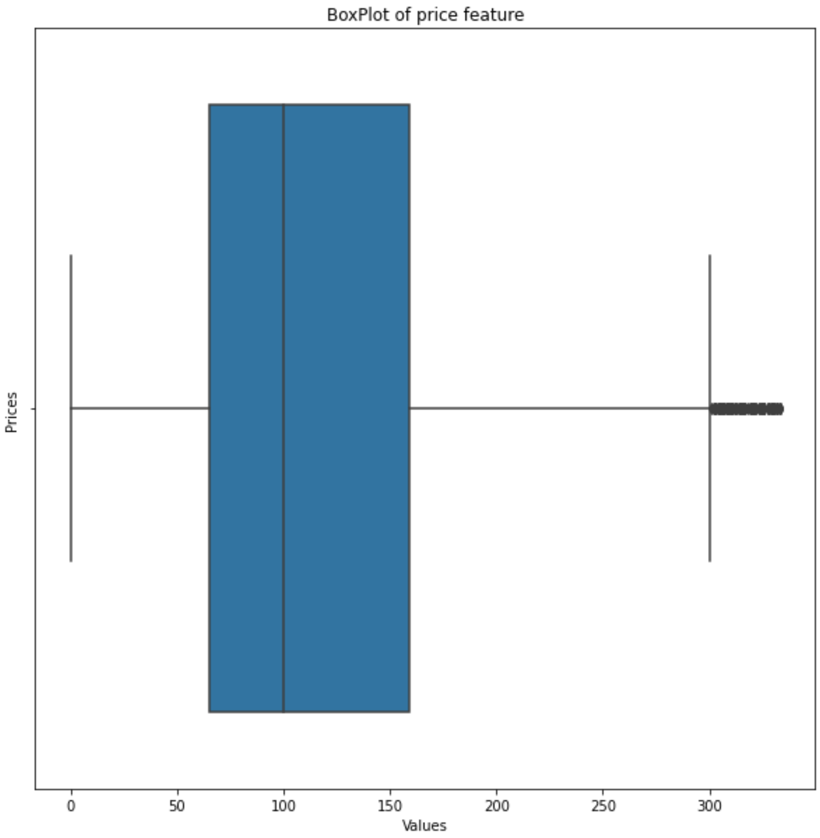


Figure 2: Box Plot of price Feature

Task2: General Information

Histogram

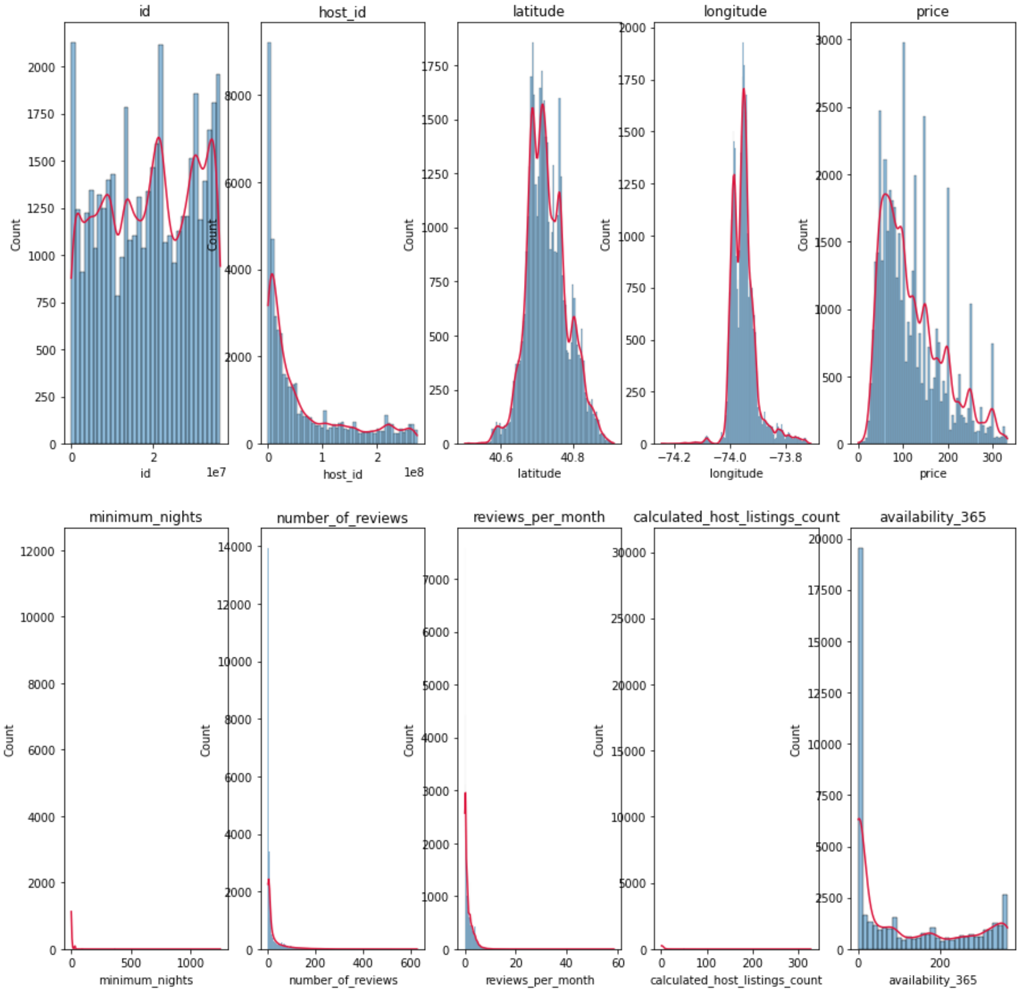


Figure 3: Histogram of quantitative features

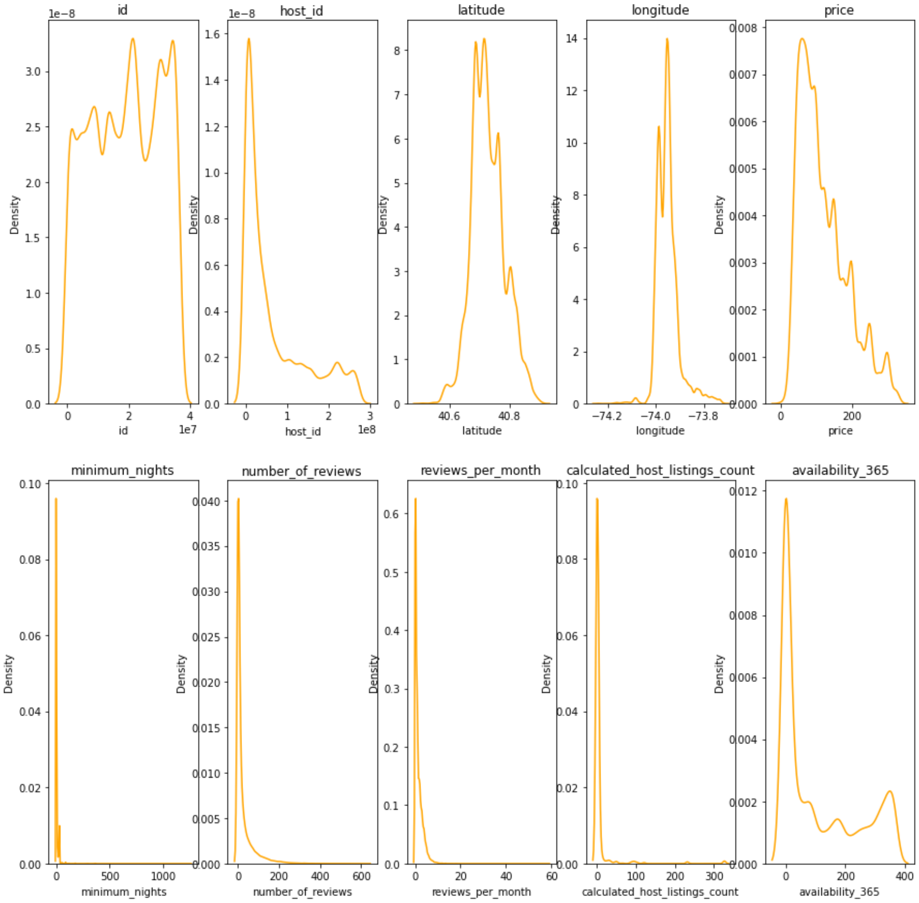
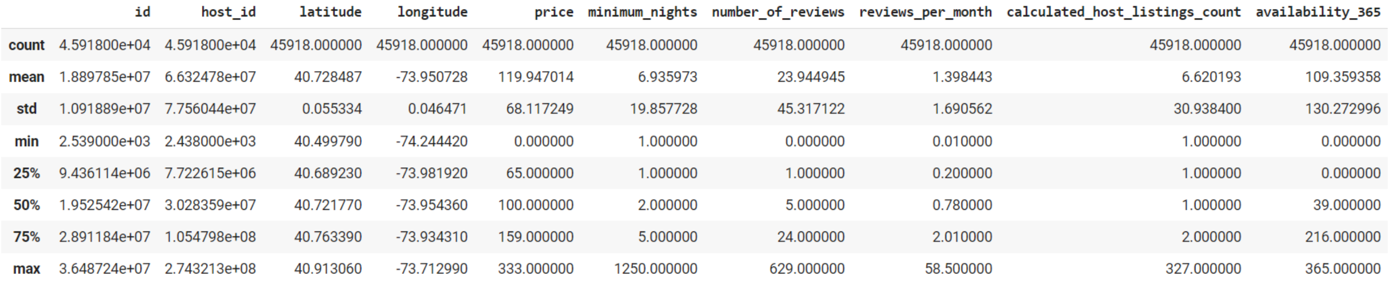


Figure 4: kde plot of Data



	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	4.591800e+04	4.591800e+04	45918.000000	45918.000000	45918.000000	45918.000000	45918.000000	45918.000000	45918.000000	45918.000000
mean	1.889785e+07	6.632478e+07	40.728487	-73.950728	119.947014	6.935973	23.944945	1.398443	6.620193	109.359358
std	1.091889e+07	7.756044e+07	0.055334	0.046471	68.117249	19.857728	45.317122	1.690562	30.938400	130.272996
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	0.010000	1.000000	0.000000
25%	9.436114e+06	7.722615e+06	40.689230	-73.981920	65.000000	1.000000	1.000000	0.200000	1.000000	0.000000
50%	1.952542e+07	3.028359e+07	40.721770	-73.954360	100.000000	2.000000	5.000000	0.780000	1.000000	39.000000
75%	2.891184e+07	1.054798e+08	40.763390	-73.934310	159.000000	5.000000	24.000000	2.010000	2.000000	216.000000
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	333.000000	1250.000000	629.000000	58.500000	327.000000	365.000000

Figure 5: Data description

It can be seen in Figure 3 and Figure 4 that four features have Normal Distribution but with skewness and kurtosis. Following is more information about them:
The skewness of **number_of_reviews** is 3.63, and the kurtosis is 18.84.
The skewness of **minimum_nights** is 21.94, and the kurtosis is 884.54.
The skewness of **calculated_host_listings_count** is 8.43, and the kurtosis is 77.03

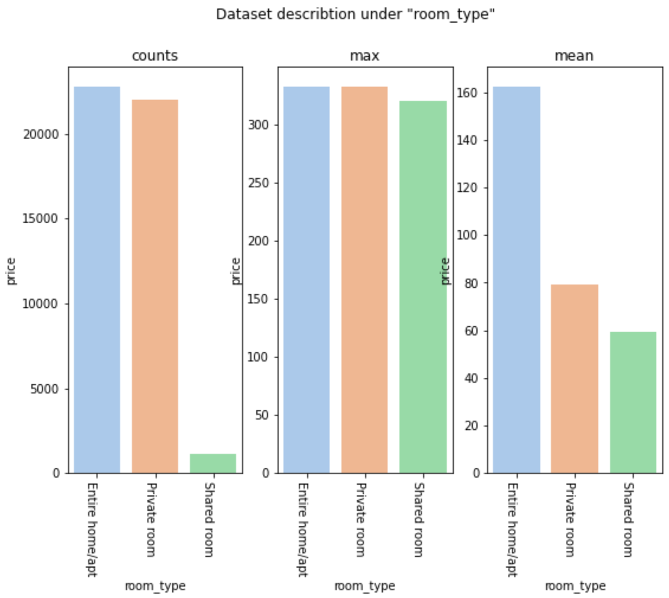
All skewness is positive, which means the distribution's right tail is longer than the left.

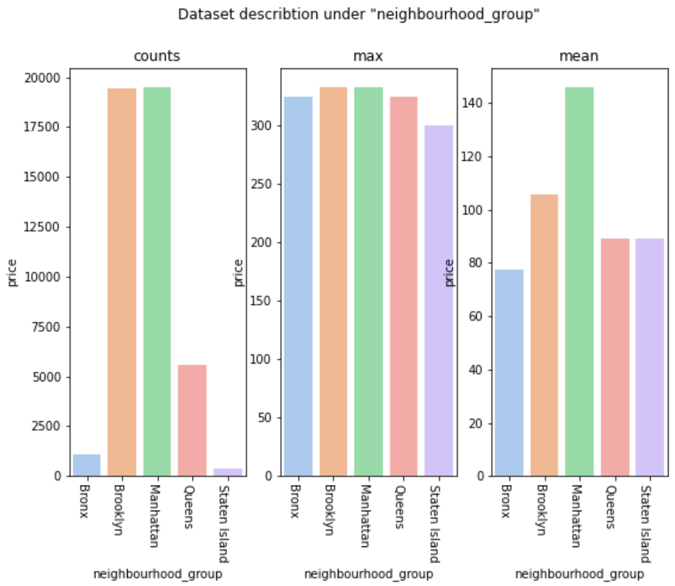
On the Other hand, Price has multi-modal distribution, and availability_360 has bio-modal.

Boxplots

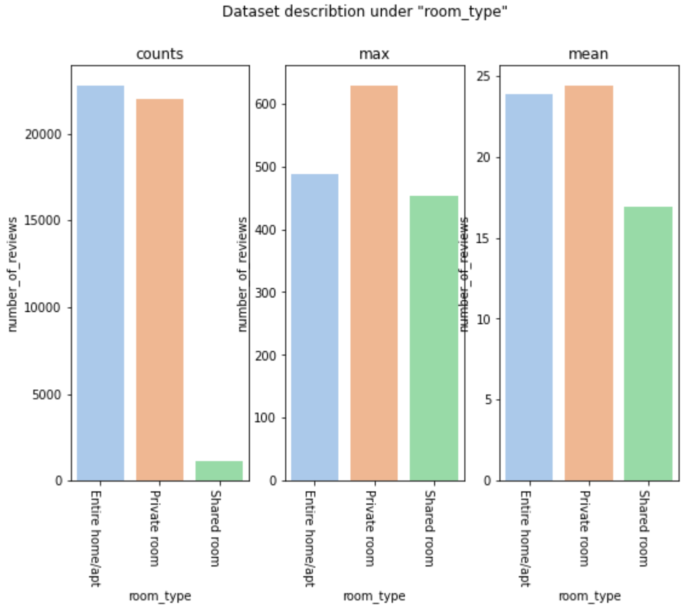
In this section, we are going to see the information of one feature under other feature conditions. Here again, because Price is the most critical feature, we are going to talk about it more.

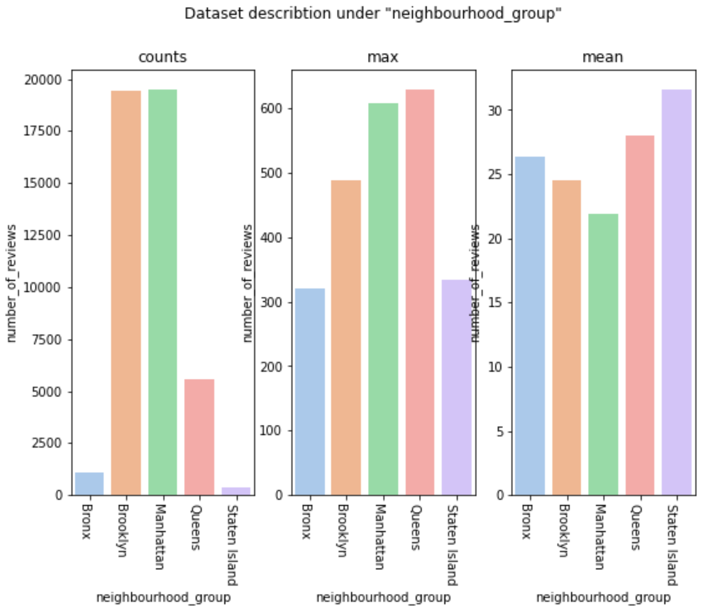
There are three categorical features that may have important information. First, we use bar plot to get some information about the general information about Price in different features. As it can be seen in **Error! Reference source not**





found., Entire home has the most Price and also the most demands. About neighborhood group, Both Brooklyn





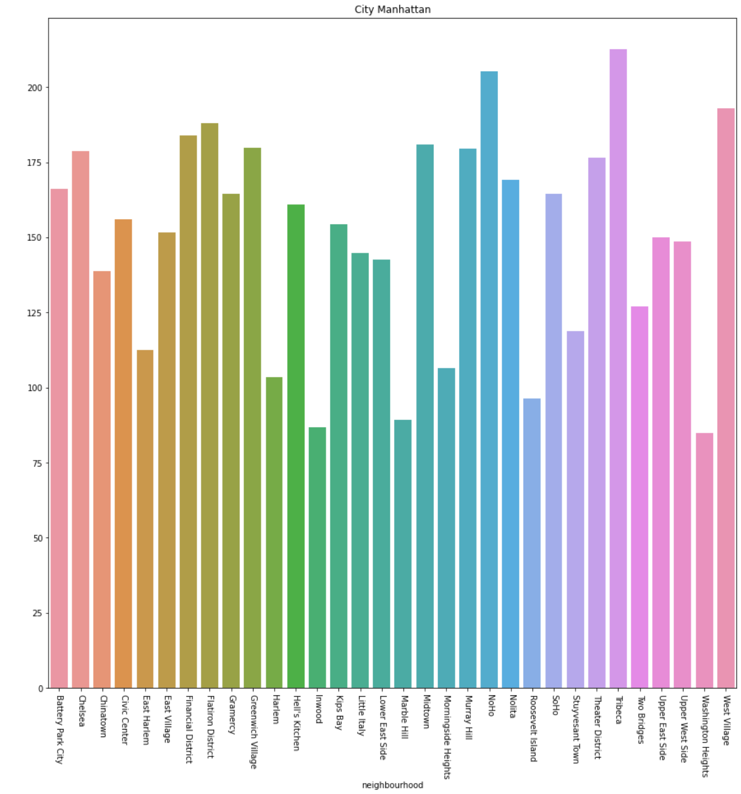
and Manhattan have the highest Price and there are the most demands for them. In average, Manhattan is the most expensive city and most demands. For better understanding, we can use reviews here, too.

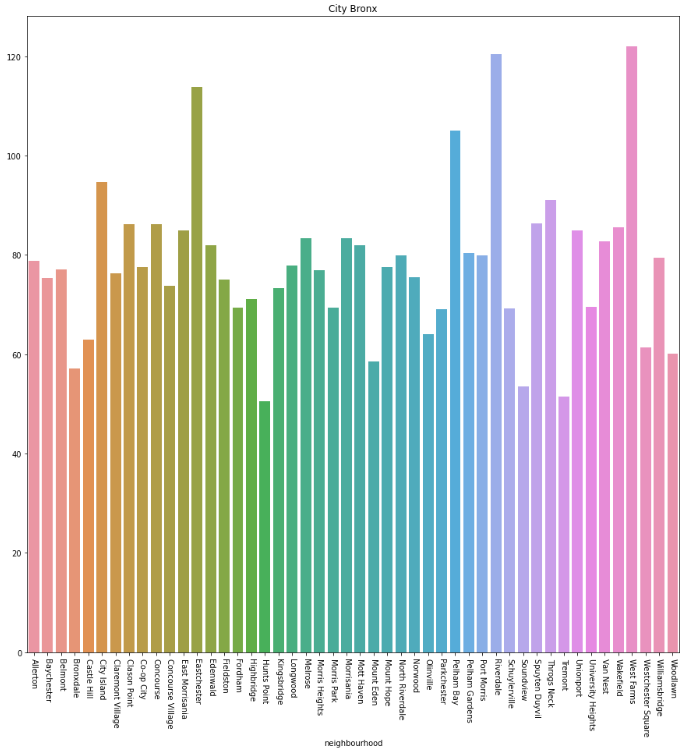
In Figure 5, we can see private room has more reviews.

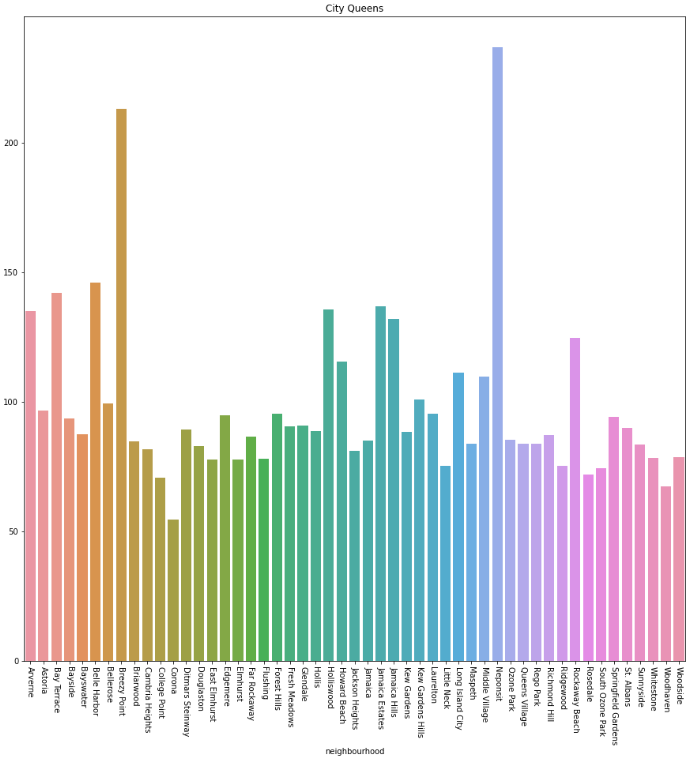
Each group has different areas. Due to time consuming, we ignored this part.

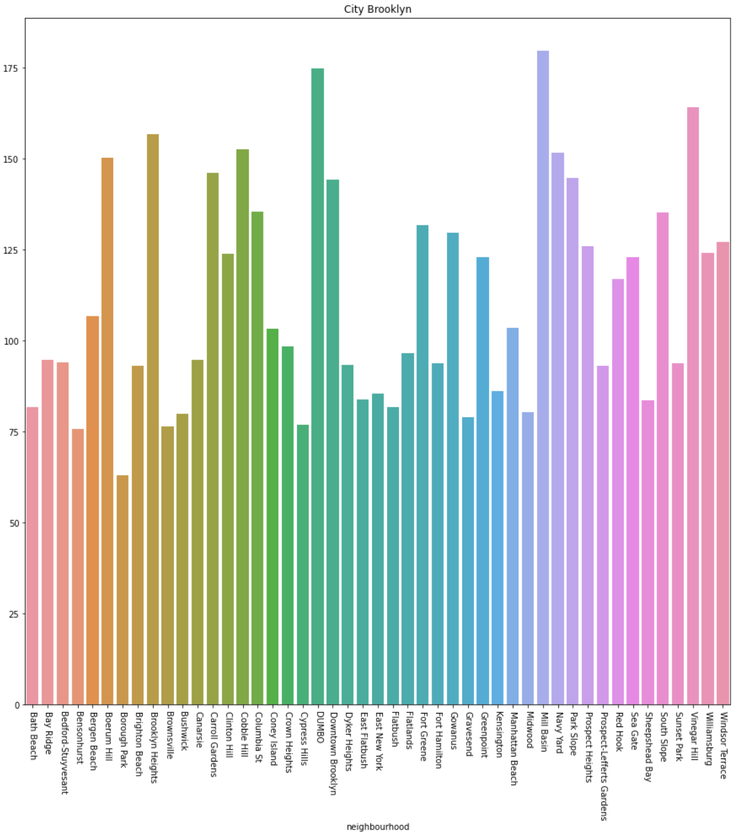
Figure 5: Review information under room type condition

Figure 6: Reviews information under neighborhood condition


City Manhattan
0
25
50
75
100
125
150
175
200
Battery Park City
Chelsea
Chinatown
Civic Center
East Harlem
East Village
Financial District
Flatiron District
Gramercy
Greenwich Village
Harlem
Hell's Kitchen
Inwood
Kips Bay
Little Italy
Lower East Side
Marble Hill
Midtown
Morningside Heights
Murray Hill
NoHo
Nolita
Roosevelt Island
SoHo
Stuyvesant Town
Theater District
Tribeca
Two Bridges
Upper East Side
Upper West Side
Washington Heights
West Village
neighbourhood


City Bronx
0
20
40
60
80
100
120
Allerton
Baychester
Belmont
Bronxdale
Castle Hill
City Island
Claremont Village
Clason Point
Co-op City
Concourse
Concourse Village
East Morrisania
Eastchester
Edenwald
Fieldston
Fordham
Highbridge
Hunts Point
Kingsbridge
Longwood
Melrose
Morris Heights
Morris Park
Morrisania
Mott Haven
Mount Eden
Mount Hope
North Riverdale
Norwood
Olinville
Parkchester
Pelham Bay
Pelham Gardens
Port Morris
Riverdale
Schuylerville
Soundview
Spuyten Duyvil
Throgs Neck
Tremont
Unionport
University Heights
Van Nest
Wakefield
West Farms
Westchester Square
Williamsbridge
Woodlawn
neighbourhood


City Queens
0
50
100
150
200
Arverne
Astoria
Bay Terrace
Bayside
Bayswater
Belle Harbor
Bellerose
Breezy Point
Briarwood
Cambria Heights
College Point
Corona
Ditmars Steinway
Douglaston
East Elmhurst
Edgemere
Elmhurst
Far Rockaway
Flushing
Forest Hills
Fresh Meadows
Glendale
Hollis
Holliswood
Howard Beach
Jackson Heights
Jamaica
Jamaica Estates
Jamaica Hills
Kew Gardens
Kew Gardens Hills
Laurelton
Little Neck
Long Island City
Maspeth
Middle Village
Neponsit
Ozone Park
Queens Village
Rego Park
Richmond Hill
Ridgewood
Rockaway Beach
Rosedale
South Ozone Park
Springfield Gardens
St. Albans
Sunnyside
Whitestone
Woodhaven
Woodside
neighbourhood


City Brooklyn
0
25
50
75
100
125
150
175
Bath Beach
Bay Ridge
Bedford-Stuyvesant
Bensonhurst
Bergen Beach
Boerum Hill
Borough Park
Brighton Beach
Brooklyn Heights
Brownsville
Bushwick
Canarsie
Carroll Gardens
Clinton Hill
Cobble Hill
Columbia St
Coney Island
Crown Heights
Cypress Hills
DUMBO
Downtown Brooklyn
Dyker Heights
East Flatbush
East New York
Flatbush
Flatlands
Fort Greene
Fort Hamilton
Gowanus
Gravesend
Greenpoint
Kensington
Manhattan Beach
Midwood
Mill Basin
Navy Yard
Park Slope
Prospect Heights
Prospect-Lefferts Gardens
Red Hook
Sea Gate
Sheepshead Bay
South Slope
Sunset Park
Vinegar Hill
Williamsburg
Windsor Terrace
neighbourhood

In the end we can see kde plot of Price and total reviews in different cities under different room type.

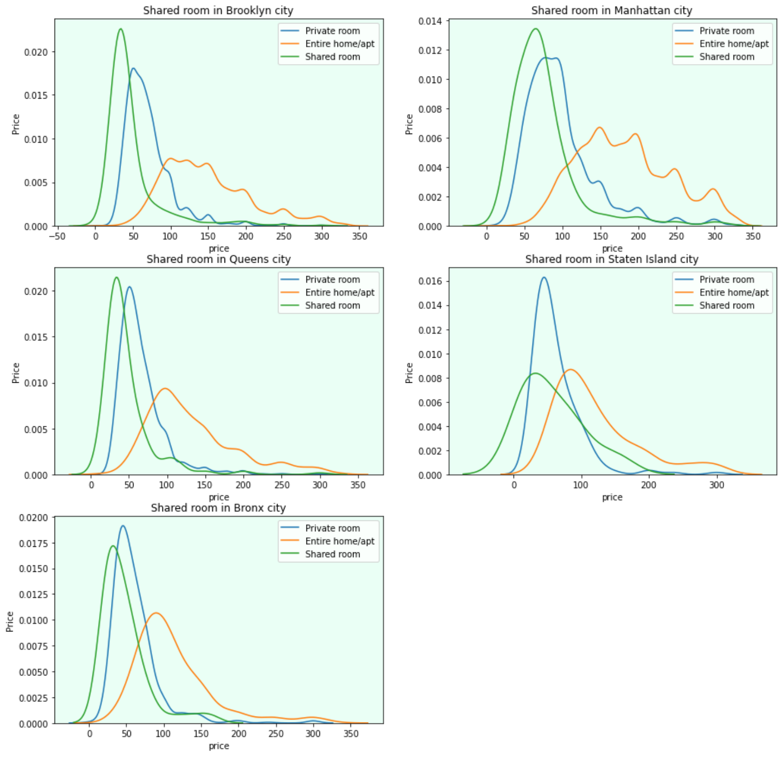


Figure 8: kde plot of price under different condition

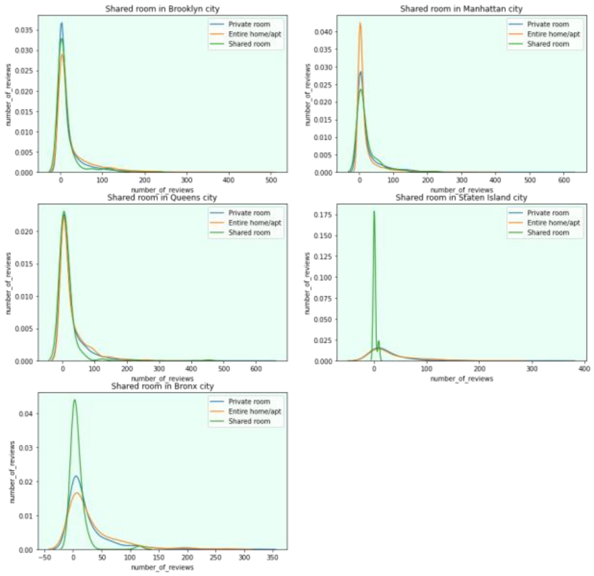


Figure 7: kde plot of number of reviews under different conditions

Task 3: Most Reviewed

Q: **If the number of comments for an ad is considered an indicator of the number of customers, it is desirable to find the owners of the ad who have the most customers and investigate the reasons for it.**

First, we find the maximum review. Sample number 11759, with 629 views, has the most views. Previously, we checked two categorical features. This sample is in **Queens** and has a **Private room**. Its Price and its number of reviews per month are near the average. For its availability_365 is near the mean of the next peak.

Task 5: Statistical hypothesis testing:

A

H0: **People spend an average 60 $ for a room**

H1: **People don't spend an average 60 $ for a room**

From the histogram of Data, it seems that people spend an average of 60 dollars for a room. Due to the non-normal distribution, we use the Wilcoxon test. We use 0.05 as alpha. The test result is:
t stat: 188.5827386334005
p_value : 0.0

reject null hypothesis

B

H0: **Neighborhood group doesn't have an effect on Price**

H1: **Neighborhood group has effects on Price**

For making sure that the neighborhood is important, we use the upside-down hypothesis. For this, we need to divide data into different neighborhood groups. The result would be :

statistic is 1507.27 & p-value is 0.0
reject null hypothesis

C

H0: **People who apply for an Entire home pay 20% more than the rest.**
H1: **People who apply for an Entire home pay 20% less than the rest.**
As it was obvious from previous plots, an Entire home is more expensive, but how much? I checked it by this hypothesis.
statistic is -169.64363251226828 & p-value is 0.0
reject null hypothesis

D

H0: **Neighborhood group has an effect on the number of reviews**.
H1: **Neighborhood group doesn't have an effect on the number of reviews.**
Let's check some features' effects on reviews.
f stat : 16.82124151547754 , p_value : 6.461275942530628e-11
reject null hypothesis

E

H0: **The number of reviews has an effect on Price.**
H1: **The number of reviews doesn't have an effect on Price.**
The most important hypothesis is this one. We will check the effect of reviews on Price.
f stat : 251.4445095533491 , p_value : 0.0
reject null hypothesis

Task 6: Model Generation

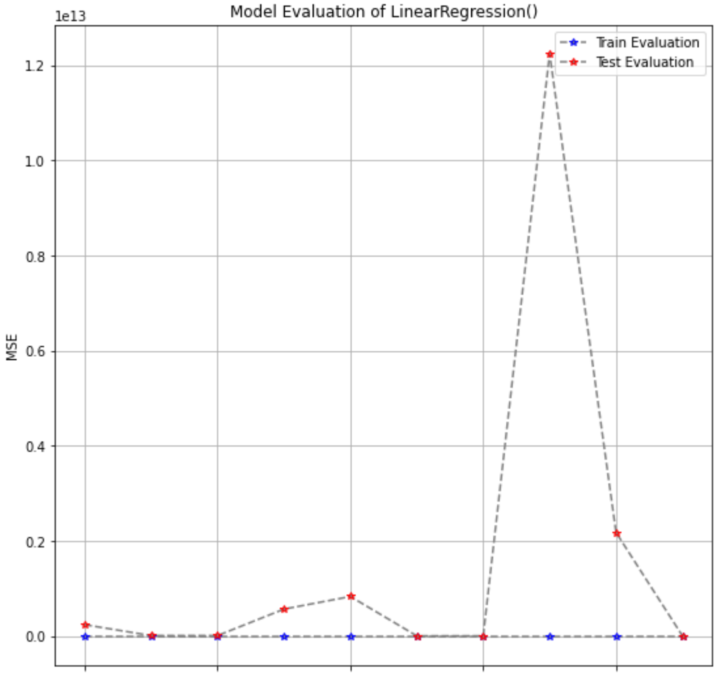
For predicting Price, I decided to choose Linear Regression. There are many other methods, but I have used the simplest one. In the first step, I deleted some unimportant features such as names. Then, we should handle categorical data. For this, we use the dummy method and add the result to the input data (here is x). we also need to delete the price feature from x and create y with that. Then, we split data to train and test. We use test data as validation and use train data as train-test. We also use min max scaler. This makes Data be in 0 and 1 interval. We should note that dummy feature doesn't need this action. Then we use 10-Fold cross validation in train data in order to preventing overfitting. The following we have train, test and validation evaluation of model. (Due to model simplicity (low complexity), evaluations are not good ☹)

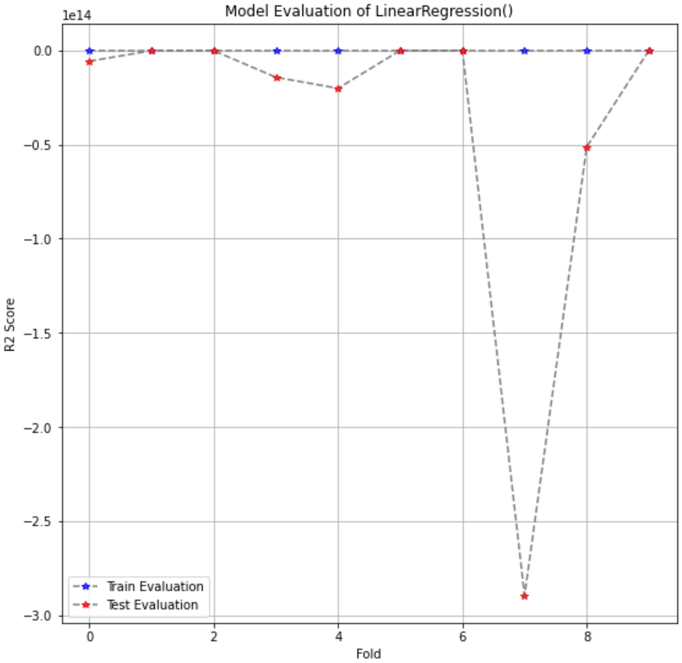
Output for validation data:

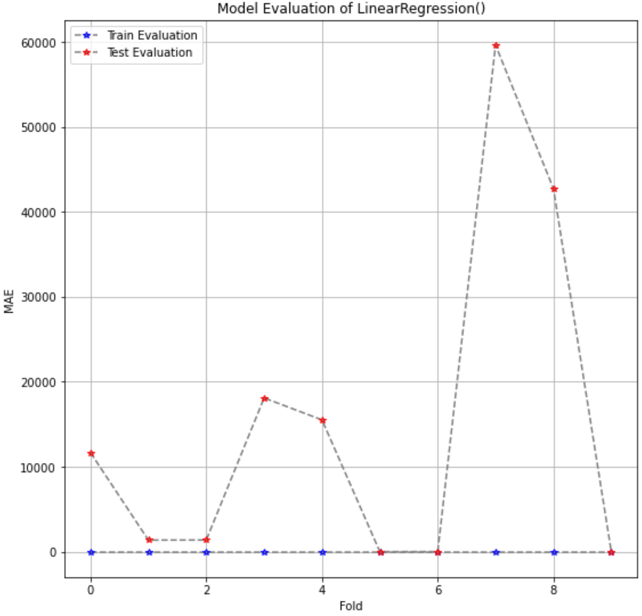
MSE value for validation is 28952776223.61

MAE value for validation is 2246.0

R2-score value for validation is -675550438311.28


Model Evaluation of LinearRegression()
1e13
MSE
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Train Evaluation
Test Evaluation
0
2
4
6
8
Fold


Model Evaluation of LinearRegression()
1e14
R2 Score
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
Train Evaluation
Test Evaluation
0
2
4
6
8
Fold


Model Evaluation of LinearRegression()
Train Evaluation
Test Evaluation
MAE
60000
50000
40000
30000
20000
10000
0
0
2
4
6
8
Fold