Project Report

On

Analyzing Airbnb data for New York City

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Engineering (Data Science)

by

Devansh Jain

(20001016015)

under the supervision of

Dr. Naresh Chauhan

Department of Computer Engineering

J. C. BOSE UNIVERSITY OF SCIENCE & TECHNOLOGY, YMCA

FARIDABAD-121006

May 2023

CANDIDATE'S DECLARATION

I hereby certify that the work which is being carried out in this Project titled **"Analyzing Airbnb data for New York City"** in fulfillment of the requirement for the degree of Bachelor of Technology in **Computer Engineering (Data Science)** and submitted to "**J. C. Bose University of Science and Technology, YMCA, Faridabad**", is an authentic record of my own work carried out under the supervision of **Dr. Sapna Gambhir**.

Devansh Jain

20001016015

CERTIFICATE

This is to certify that the work carried out in this project titled **"Analyzing Airbnb data for New York City"** submitted by **Devansh Jain** to "**J. C. Bose University of Science and Technology, YMCA, Faridabad**" for the award of the degree of Bachelor of Technology in **Computer Engineering (Data Science)** is a record of bonafide work carried out by them under my supervision. In my opinion, the submitted report has reached the standards of fulfilling the requirements of the regulations to the degree.

Dr. Naresh Chauhan
(Mentor)
Professor Department of
Computer Engg.
J. C. Bose University of Science and Technology, YMCA, Faridabad

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

INTRODUCTION

Airbnb is considered one of the biggest hotel chains in the world. And it does not own a single hotel room!

The company became successful by connecting travelers who need a place to stay with the so-called hosts, people who are willing to rent their places. In the Airbnb platform, it is possible to book everything from a shared room in a house with other people to an entire apartment or hotel room.

Founded in 2008, Airbnb has already hosted over 300 million guests and aims to reach 1 billion by the time it turns 20, in 2028.

The company also makes a lot of its data available for free. Through the Inside Airbnb website, anyone can have access to a great amount of information about Airbnb operation in the most important cities in the world.

PROBLEM IDENTIFICATION

NEW YORK CITY

New York City is the most populous city in the United States, with over eight million inhabitants, and it is the center of the largest metropolitan area in the world by urban landmass, the New York metropolitan area. Considered the cultural, financial, and media capital of the world, New York is the home of the United Nations Headquarters. The city is composed of five boroughs: Brooklyn, Queens, Manhattan, the Bronx, and Staten Island.

With numerous famous attractions, such as Central Park, Times Square, the Brooklyn Bridge, and the Empire State Building, New York receives over sixty million visitors every year which generated an all-time high $61.3 billion in overall economic impact for New York City in 2014. Such numbers make New York City a huge Airbnb hub and a great topic for our analysis.

OBJECTIVE

In this project, we will work with a dataset of New York City properties advertised on the platform. This dataset contains information about the prices, locations, reviews, room types, hosts, and more for over 50,000 rooms.

Our main goal is to take some insights from the data, such as the most common room types, locations, and how the prices vary depending on the room type and the location of the property.

To accomplish this goal, we'll need to go through the following steps:

- Getting and exploring the data;
- Cleaning the data;
- Analyzing the data.

Python and its powerful libraries will be our tool to get this job done!

CHAPTER 2: DESCRIPTION AND METHODOLOGY

FLOW OF WORK :-

Step 1. Exploratory Data Analysis

Step 2. Data cleaning/ removing outliers

Step 3. Analyzing the data and Correlations

Step 4. Visualization using scatter plots and leaflet maps

Step 5. Drawing Conclusions

HOW TO CARRY FORWARD ?

THE DATA

We'll begin by importing:

- pandas for data manipulation;
- seaborn and matplotlib for data visualization;
- folium to deal with geographical data.

Next, Read the data into a DataFrame.

Before we perform any analysis, we'll first see what our dataset looks like. These are the variables it contains:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50246 entries, 0 to 50245
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              50246 non-null  int64
 1   name                            50228 non-null  object
 2   host_id                         50246 non-null  int64
 3   host_name                       50235 non-null  object
 4   neighbourhood_group             50246 non-null  object
 5   neighbourhood                   50246 non-null  object
 6   latitude                        50246 non-null  float64
 7   longitude                       50246 non-null  float64
 8   room_type                       50246 non-null  object
 9   price                           50246 non-null  int64
 10  minimum_nights                  50246 non-null  int64
 11  number_of_reviews               50246 non-null  int64
 12  last_review                     39216 non-null  object
 13  reviews_per_month               39216 non-null  float64
 14  calculated_host_listings_count  50246 non-null  int64
 15  availability_365                50246 non-null  int64
dtypes: float64(3), int64(7), object(6)
memory usage: 6.1+ MB
```

Displaying its first 5 rows :-

```
ny.head()
```

Python

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	c
0	2060	Modern NYC	2259	Jenny	Manhattan	Washington Heights	40.85722	-73.93790	Private room	100	1	1	2008-09-22	0.01	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	3	48	2019-11-04	0.38	
2	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	318	2020-04-26	4.66	
3	5099	Large Cozy 1 BR Apartment In Midtown East	7322	Chris	Manhattan	Murray Hill	40.74767	-73.97500	Entire home/apt	200	3	78	2019-10-13	0.58	
4	5121	BlissArtsSpace!	7356	Garon	Brooklyn	Bedford-Stuyvesant	40.68688	-73.95596	Private room	60	29	50	2019-12-02	0.38	

Investigating null values :-

```
(round(ny.isnull().sum() / ny.shape[0] * 100, 2)).sort_values(ascending=False)
```

Python Python

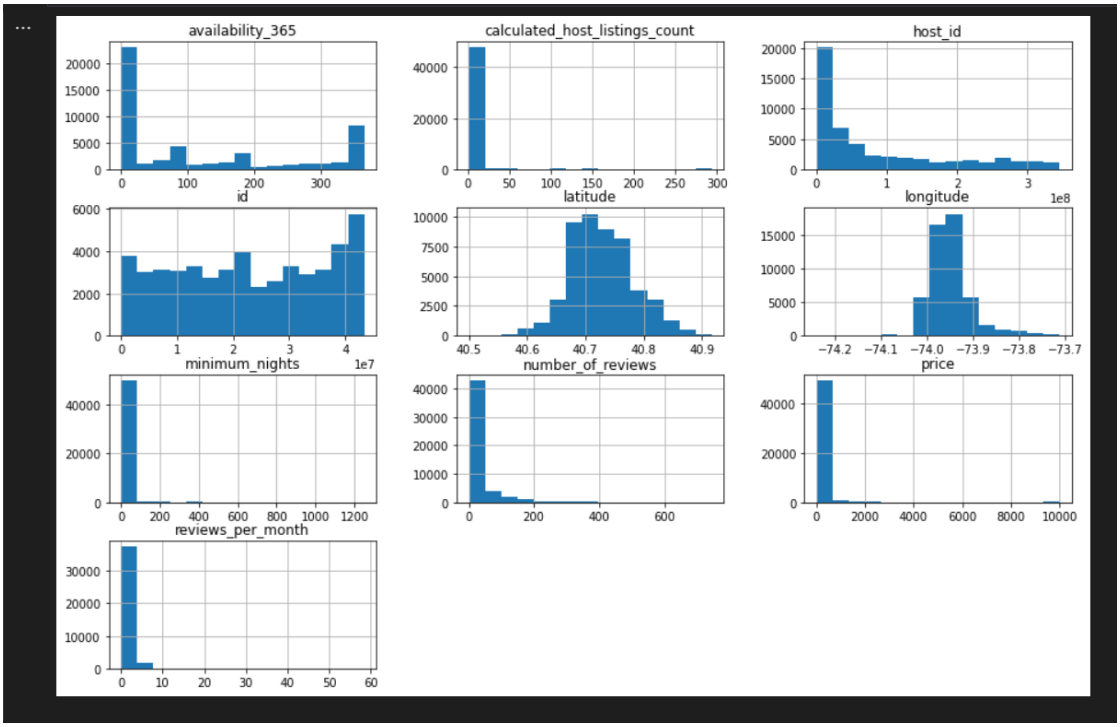
```
reviews_per_month                21.95
last_review                      21.95
name                              0.04
host_name                         0.02
availability_365                  0.00
calculated_host_listings_count    0.00
number_of_reviews                 0.00
minimum_nights                    0.00
price                             0.00
room_type                         0.00
longitude                         0.00
latitude                          0.00
neighbourhood                     0.00
neighbourhood_group               0.00
host_id                           0.00
id                                0.00
dtype: float64
```

Good news! Only the last_review and reviews_per_month columns contain a significant amount of null values. As both of these columns are not the focus of our analysis, this will not be a problem. Later in this project, we will drop them.

The name and host_name columns also contain null values, but this also will not affect our project since we are not performing any analysis on them. Also, the number of null values is irrelevant.

Variable Distribution

We'll now plot some histograms in order to see the distribution for each variable and start looking for outliers.



Looking at the histograms, we can notice that some important variables like price and minimum_nights our poorly distributed. In order to better identify these problems, let's see more statistics about the dataset using the describe method.

```
ny[['price', 'minimum_nights', 'number_of_reviews', 'reviews_per_month',
    'calculated_host_listings_count', 'availability_365']].describe()
```

	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
count	50246.000000	50246.000000	50246.000000	39216.000000	50246.000000	50246.000000
mean	163.130777	7.912968	24.410978	1.091541	7.046292	121.786530
std	421.687803	21.472286	48.609159	1.425768	28.428846	140.634991
min	0.000000	1.000000	0.000000	0.010000	1.000000	0.000000
25%	68.000000	1.000000	1.000000	0.160000	1.000000	0.000000
50%	104.000000	3.000000	5.000000	0.510000	1.000000	65.000000
75%	175.000000	5.000000	24.000000	1.570000	2.000000	249.000000
max	10000.000000	1250.000000	746.000000	58.430000	294.000000	365.000000

It is easy to see that some values do not make sense. Let's look at the price column for instance. The average price is 163.13 and75175, however, the maximum price is 10,000.

The same happens in the minimum_nights column, where the maximum value is 1,250! How can someone expect to have their place booked if

the visitor has to stay at least three and a half years? It makes absolutely no sense!

Values like these distort reality and any analysis we attempt to perform. Now we'll have to deal with them.

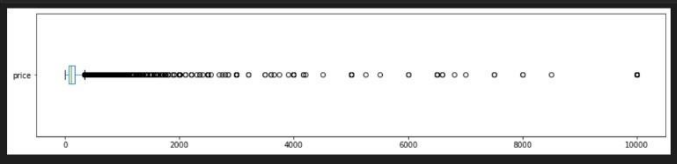
Removing Outliers

We'll plot boxplots for each of these columns so we can take a closer look at their distribution.
Also, let's see how many and what percentages of prices are equal to 0 and 500.00 and the percentages of minimum nights that are over 30 nights.

```
# Displaying the box plot
ny['price'].plot(kind='box', vert=False, figsize=(15,3))
plt.show()

# Calculating the number and the percentage of prices that are equal to $0 and over $500.00
print('Values over $500.00: ')
print(len(ny[ny['price'] > 500]))
print('{:.4f}%'.format((len(ny[ny['price'] > 500]) / ny.shape[0]) * 100))
print('\n')
print('Values equal to $0: ')
print(len(ny[ny['price'] == 0]))
print('{:.4f}%'.format((len(ny[ny['price'] == 0]) / ny.shape[0]) * 100))
```

Python



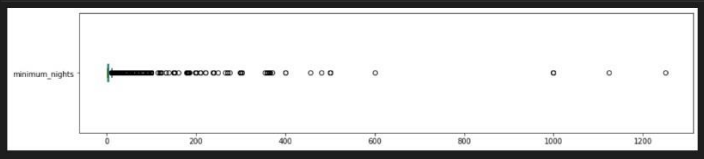

```
Values over $500.00:
1139
2.2668%


Values equal to $0:
26
0.0517%
```

```
# Displaying the box plot
ny['minimum_nights'].plot(kind='box', vert=False, figsize=(15,3))
plt.show()

# Calculating the number and the percentage of rooms with the minimum nights value over 30
print('Values over 30 nights: ')
print(len(ny[ny['minimum_nights'] > 30]))
print('{:.4f}%'.format((len(ny[ny['minimum_nights'] > 30]) / ny.shape[0]) * 100))
```

Python




```
Values over 30 nights:
772
1.5364%
```

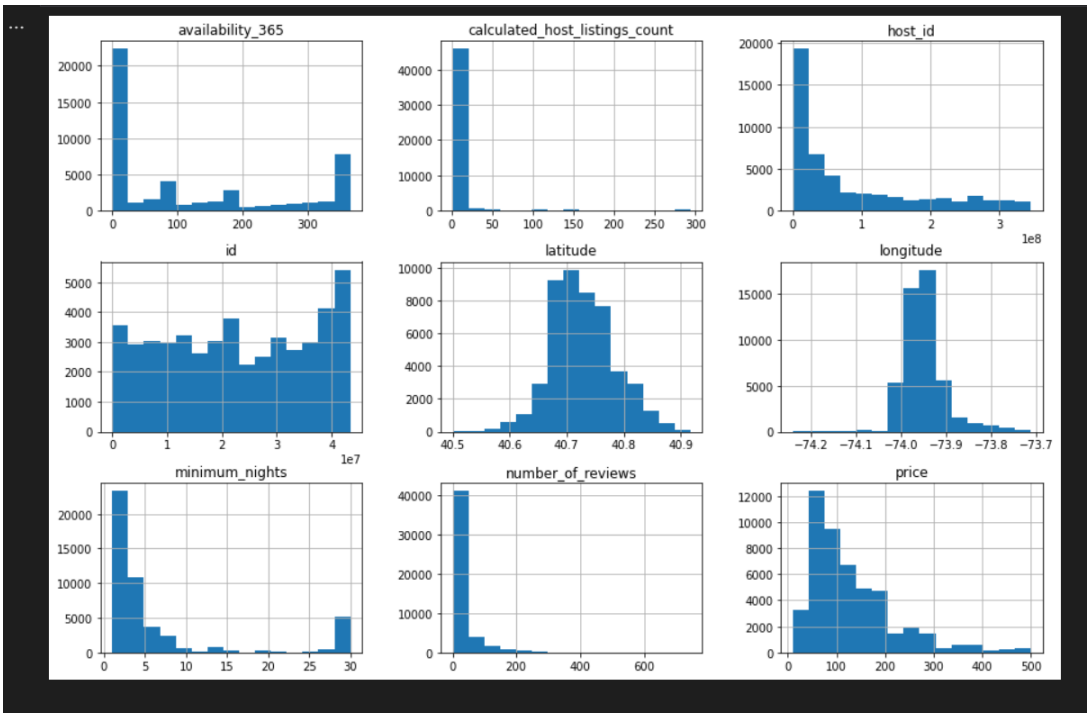
We can see that only 2.27% of the price column is above $500.00 and only 1.54% of the values in the minimum_nights column is above 30. We have 26 elements with price zero as well.

Taking into consideration, as we said earlier, that 75% of these columns' values are below $175.00 and 5 nights, respectively, it is reasonable to lose roughly 3.8% of the data in order to make it more realistic. Therefore, we'll create a new dataframe, ny_clean, that contains only the rows in which the price is more than 0 and less than 500, and the minimum nights is no more than 30.

Also, probably some columns fulfill both these requirements, which means that we are losing even less than 3.8% of the dataset.

After we create the new dataframe, we'll drop
the reviews_per_month and last_review columns as we said earlier in the project.

Finally, let's see if the histograms look better.



Correlation Heatmap

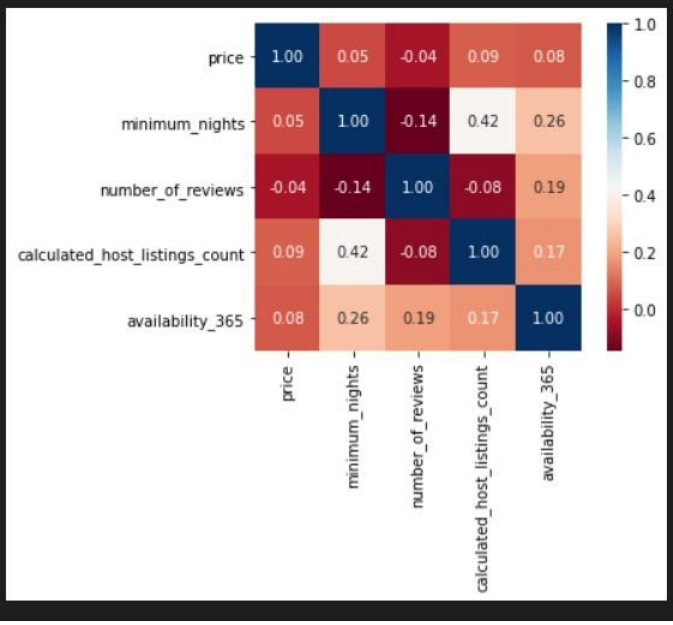
Now let's see if there's any correlation between the numeric variables in the dataset.

We'll first create a correlation matrix using the corr method and then we will take advantage of the heat_map function from seaborn to visualize this matrix.

```
corr = ny_clean[['price', 'minimum_nights', 'number_of_reviews', 'calculated_host_listings_count', 'availability_365']].corr()
corr
```

	price	minimum_nights	number_of_reviews	calculated_host_listings_count	availability_365
price	1.000000	0.050787	-0.043062	0.086891	0.076512
minimum_nights	0.050787	1.000000	-0.143330	0.415113	0.259879
number_of_reviews	-0.043062	-0.143330	1.000000	-0.081395	0.194047
calculated_host_listings_count	0.086891	0.415113	-0.081395	1.000000	0.173000
availability_365	0.076512	0.259879	0.194047	0.173000	1.000000

```
sns.heatmap(corr, cmap='RdBu', fmt='.2f', square=True, linecolor='white', annot=True)
```



CHAPTER 3: HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and software requirements of any project are must to be satisfied, so that the virtual environment can be set up on any machine to run the project. So, in this section the software and the hardware requirements are discussed completely.

Software Requirements

Code Editor – VS Code/ Jupyter Notebook

OS – Windows 10

Language – Python

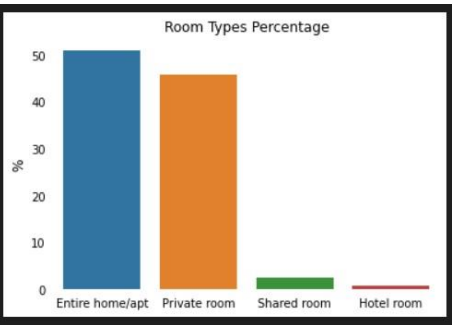
Libraries – pandas, NumPy, Seaborn, matplotlib, folium

Hardware Requirements

a. A computer
b. Minimum System Requirements: -
    i. O.S. -Windows 10/11
    ii. Quad-core i5 CPU minimum
    iii. 8 GB RAM
    iv. 256GB SSD

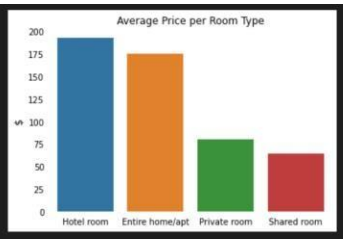
CHAPTER 4: RESULT ANALYSIS

Individual Analysis



Entire apartments and private rooms dominate the Airbnb market in New York City. Hotel rooms are basically nonexistent.

Now let's see the average prices for each of these room types.

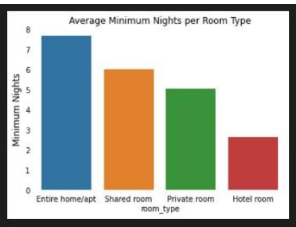


Hotel rooms are the most expensive type of room, on average. This might happen for two reasons:

- Hotels are naturally more expensive because of the number of employees and the services and options available to the guest, such as room service and parking spots;
- There are fewer hotels in the dataset, which distorts the average.

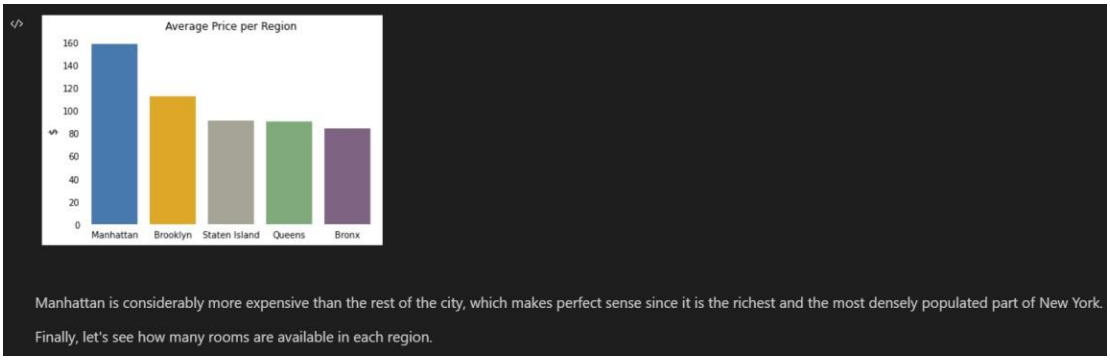
Other than that, it is reasonable to think that entire apartments are more expensive than private rooms and that private rooms are more expensive than shared rooms.

Now, let's take look in the average minimum nights for each room type.



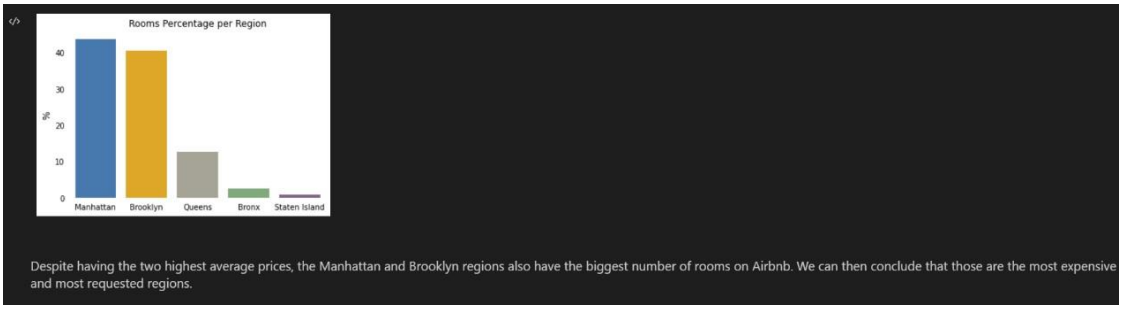
Locations

We'll now investigate prices in different regions of the city. For this, we'll use the `neighbourhood_group` instead of the `neighborhood` column because it divides the city into five major regions instead of lots of small neighborhoods.



Manhattan is considerably more expensive than the rest of the city, which makes perfect sense since it is the richest and the most densely populated part of New York.

Finally, let's see how many rooms are available in each region.

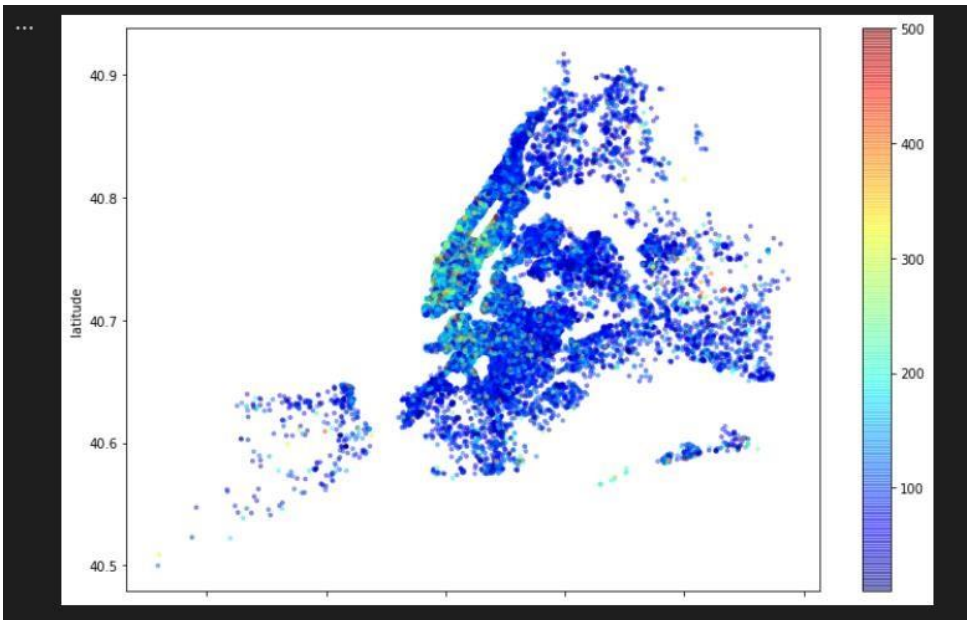


Despite having the two highest average prices, the Manhattan and Brooklyn regions also have the biggest number of rooms on Airbnb. We can then conclude that those are the most expensive and most requested regions.

Geographical Data

Now that we already have some information about the price distribution for rooms in New York City, let's visualize this data geographically and try to determine which points of the city present higher and lower average prices on Airbnb.

First, we'll create a scatter plot using the latitude and longitude columns of our dataset.



But it is not easy to see the city in charts like these. In this specific case, as New York is one of the most famous cities in the world, we can see some patterns. For example, we can identify the island of Manhattan and we can see Central Park, the white square in the middle of the island.

With this in mind, it is easier to see that the prices in Lower Manhattan are higher than most of the other regions on the map. The region around the Brooklyn Bridge presents some higher values and we can see some very high values on the east side of Queens too.

We can have a general idea of what is going on mostly because we are familiar with the city's geography. If that was not the case, it would've been harder to take any insights from this plot.

To improve this visualization and plot a real map, we'll work with folium from now on.

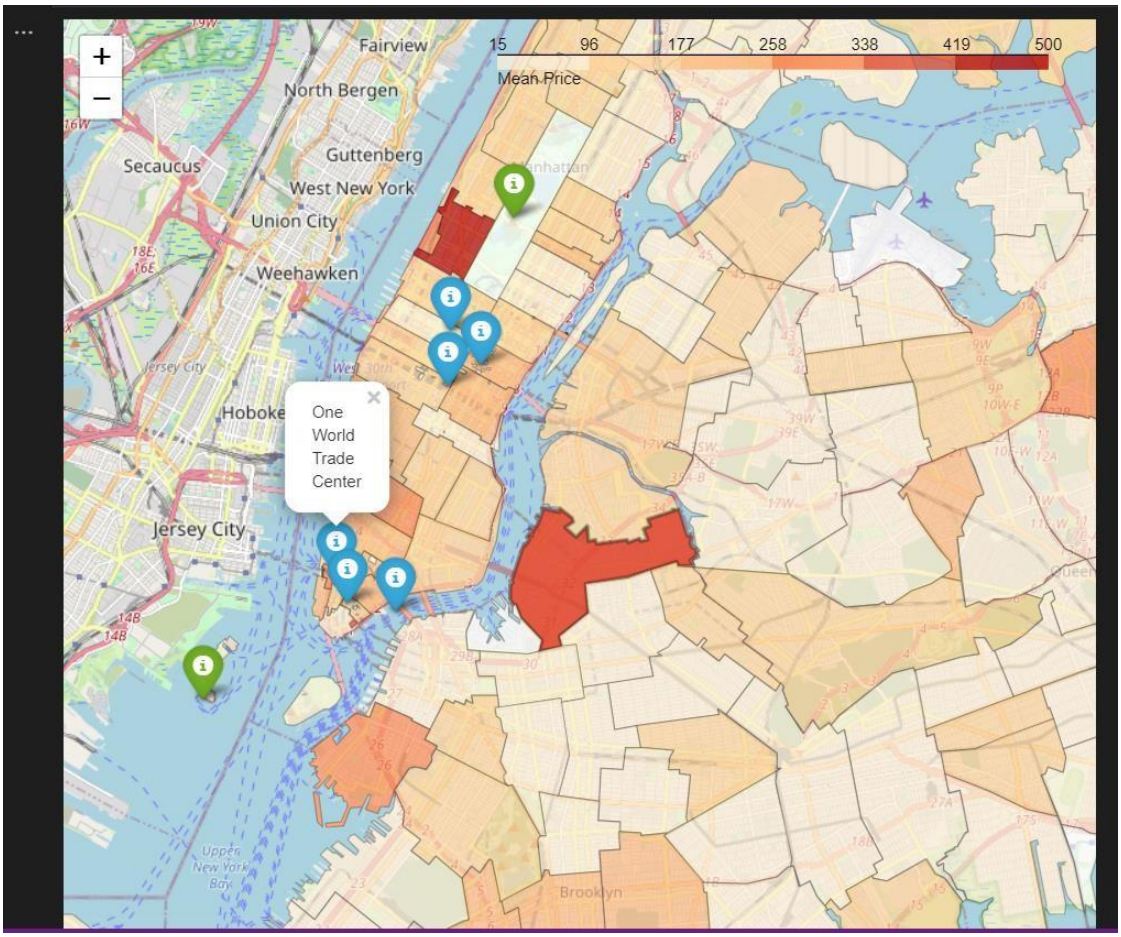
FOLIUM LEAFLET (FINAL OUTPUT MAP)

Folium is a library that makes it easy to visualize data that’s been manipulated in Python on an interactive leaflet map.

Plotting the Map with Folium

Finally, we'll plot the map. To do this, we'll follow these five steps:

- Create a figure;
- Create the map;
- Create the Choropleth layer;
- Create some markers on the map pointing to some famous New York attractions;
- Display the map.



CHAPTER 5: CONCLUSION

Now we have a good visualization of New York City Airbnb prices.

As expected, Lower Manhattan is a very expensive region of the city and it is also where the most famous attractions are located, which certainly influences the high prices by increasing the demand for rooms.

The Bronx and Staten Island are low prices regions. Each of them has one more expensive area, but this might be related to the small number of rooms in these regions, which can distort the average price, as we saw earlier in the project.

Brooklyn and Queens are very mixed prices regions. Although the bigger part of these regions consists of low price areas, they also have a significant amount of expensive neighborhoods. The areas closer to Manhattan tend to be more expensive, for instance. Some areas in the south of Brooklyn also contain some higher prices as well as the east side of Queens, as we mentioned when we discussed the scatter map.

EXPECTED OUTCOMES

During the project, we performed some interesting analysis of the New York City Airbnb data and managed to answer some questions, such as:

- What kind of room is more common in New York City Airbnb?
- What is the price difference between different types of rooms?
- What are the most expensive regions to stay in New York?

We could also see how to use Python to go from a text file to a complete interactive map.

To accomplish such goals, we went through major data manipulation steps, such as exploring, cleaning, analyzing, and visualizing data.

With all that said, the conclusions are:

- Private rooms and entire apartments are the most common room types;
- Hotel rooms and entire apartments are usually more expensive than private and shared rooms;
- Over 80% of the rooms are located in Manhattan and Brooklyn, which are also the most expensive regions;
- Yes, if you want to stay close to the major attractions of the city you'll probably expend more money.

BRIEF PROFILE OF STUDENT

Name : Devansh Jain

Roll No : 20001016015

Branch : Computer Engineering (Data Science)

Email Id : 20001016015@jcboseust.ac.in

Brief About Project:

The Project revolves around Analyzing a large dataset of Airbnb. The objective of the project is to find out the useful insights out of this huge data and build a model so as to give to front-end workers to make this model available to clients in friendly mode. We use various scatter plots and folium interactive map to accurately interpret our data and form conclusions required for Airbnb business intelligence.

Future Scope:

The project includes giving out analysis results only. The project can be extended to including a front-end part also that may include fully working application or a website to use these analysis results and give effective results to user's queries.