BIG DATA PROJECT REPORT

Analysis of New York AirBnB

Overview:

In this project, we applied a range of big data tools to explore some interesting data sets and derive insights from them. Ingest data, apply transformations, profile the data, summarize it, visualize it.

The objective of this project is to clean and visualize the Airbnb New York dataset to gain insights and answer questions about the listings, hosts, and locations. The dataset includes information such as listing ID, host ID, host name, neighborhood group, neighborhood, latitude, longitude, room type, price, minimum nights, number of reviews, and availability 365.

We aim to answer and visualize the few research questions like how reviews affect the price, which neighborhood is most likely to be instantly booked, how highly rated reviews impact the availability of the Airbnb and visualize various other patterns using our data set.

I. Introduction:

Airbnb, Inc. is a company based in the United States that operates an online marketplace for lodging, primarily homestays for vacation rentals, as well as tourism activities. Airbnb does not own any of the listed properties; instead, it profits from each booking through a commission. The company was established in 2008. Airbnb is a shortened version of the company's full name, AirBedandBreakfast.com. The popularity of Airbnb has grown exponentially in recent years, providing travelers with affordable and unique accommodations while also enabling hosts to monetize their extra space. As of 2022, Airbnb has over 7 million listings in more than 220 countries and regions.

In this project, we analyze a dataset that contains details on New York City's Airbnb listings. The data was obtained from Inside Airbnb, a non-profit collection of tools and information that enables you to investigate how Airbnb is actually used in places all over the world.

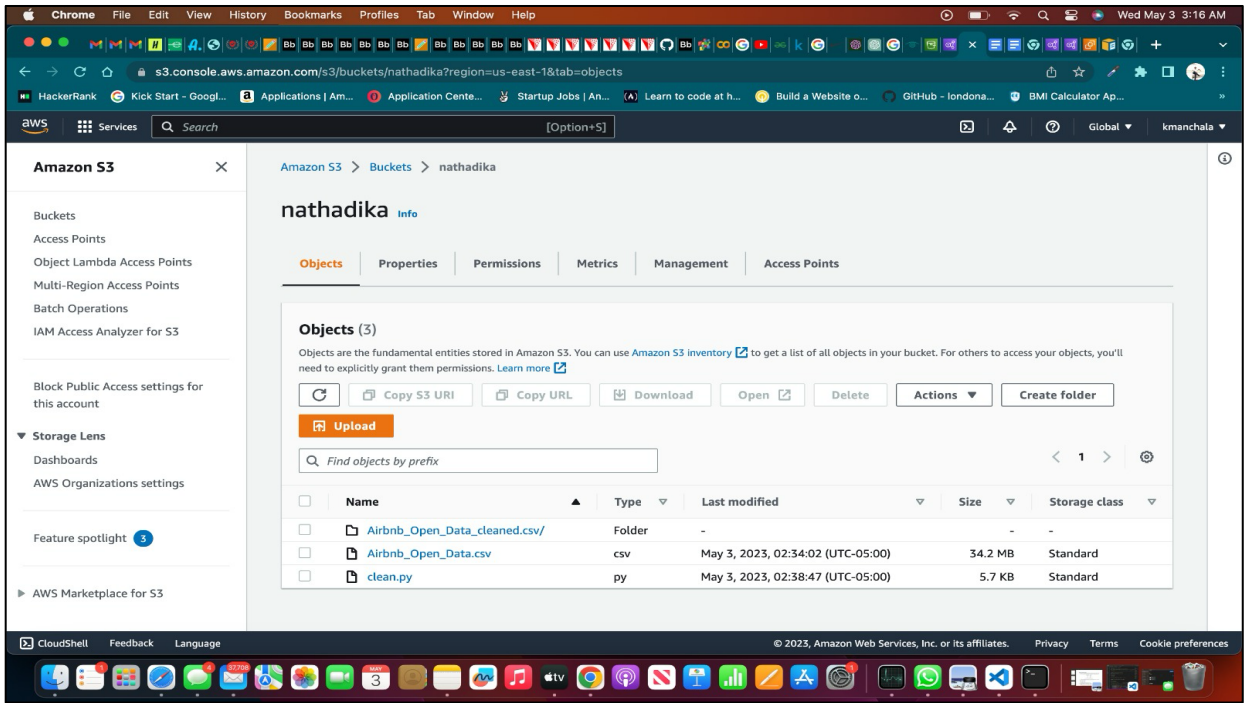
This analysis's goal is to examine and clean the data in order to learn more about the variables that affect Airbnb prices in New York City. To remove missing values, duplicates, and outliers from the data as well as to derive valuable insights from it, we will employ data cleaning techniques. In order to find patterns and trends in the dataset, we will finally visualize the data.

Project Tools:

- Amazon S3 Bucket
- Apache Spark
- Jupyter Notebook with Python

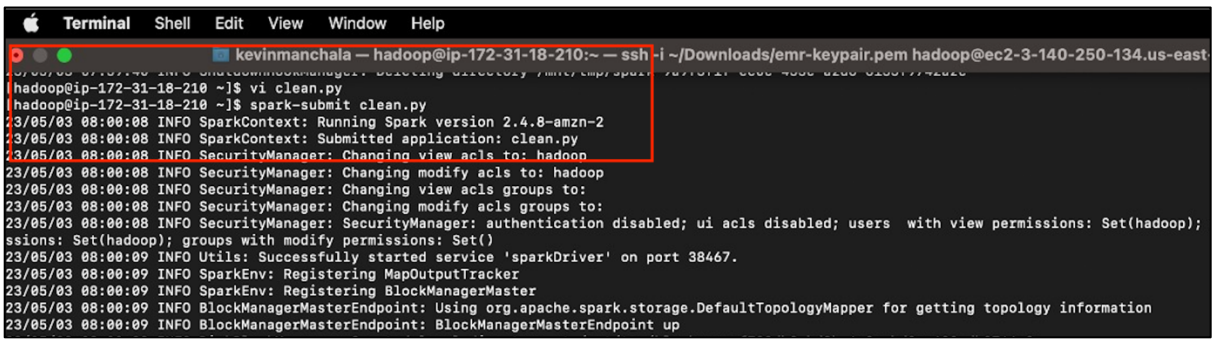
II. DATA OPERATIONS AND PROCESS FOR ANALYSIS:

- **Data processing :** We have processed the data into required fields using AWS S3 Bucket. We created a spark session and read the dataset (CSV's). After reading the data, we are using PySpark for data profiling the data to be used for data analysis. [4]

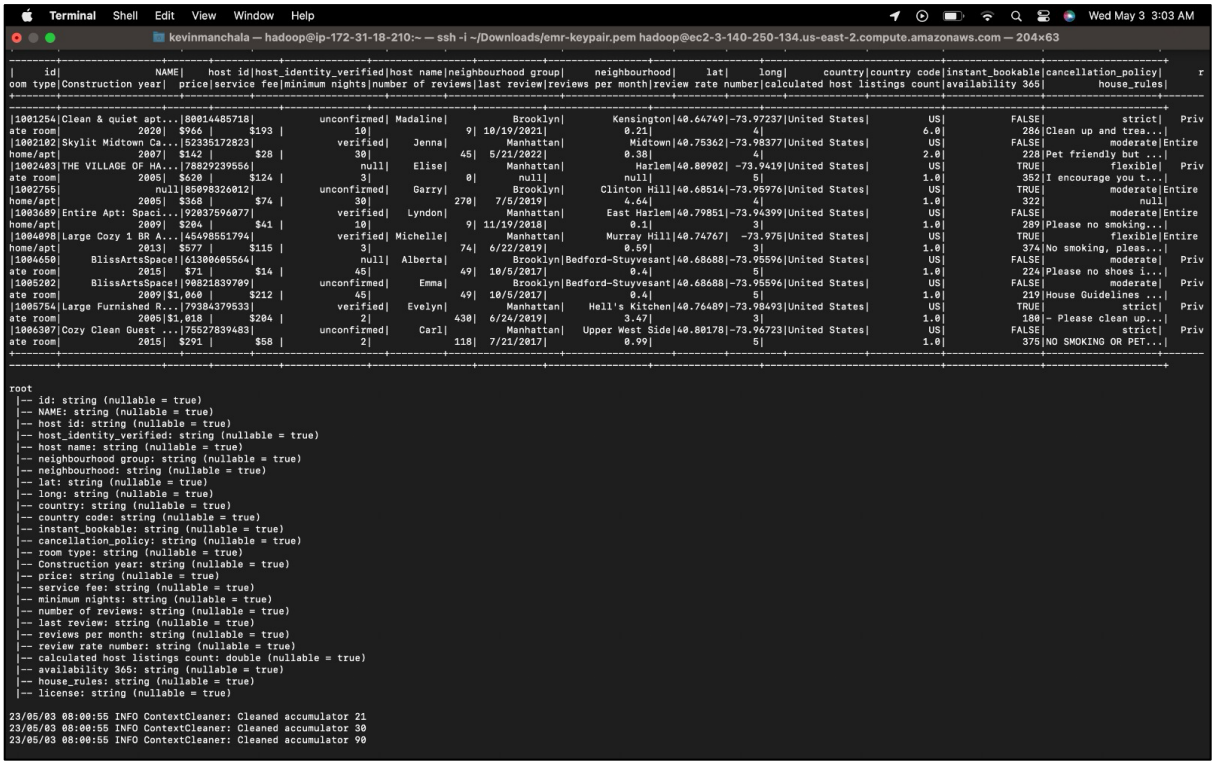


- **Data Cleaning :** We have performed cleaning on the dataset. We have removed duplicates, dropped unnecessary columns , renamed columns to remove spaces and make them more readable, removed numbers and punctuations from the 'NAME' column, leading and trailing spaces in string columns. Replacing $ and , in price column and service fee column and correcting and replacing wrong names. [1][2][3]

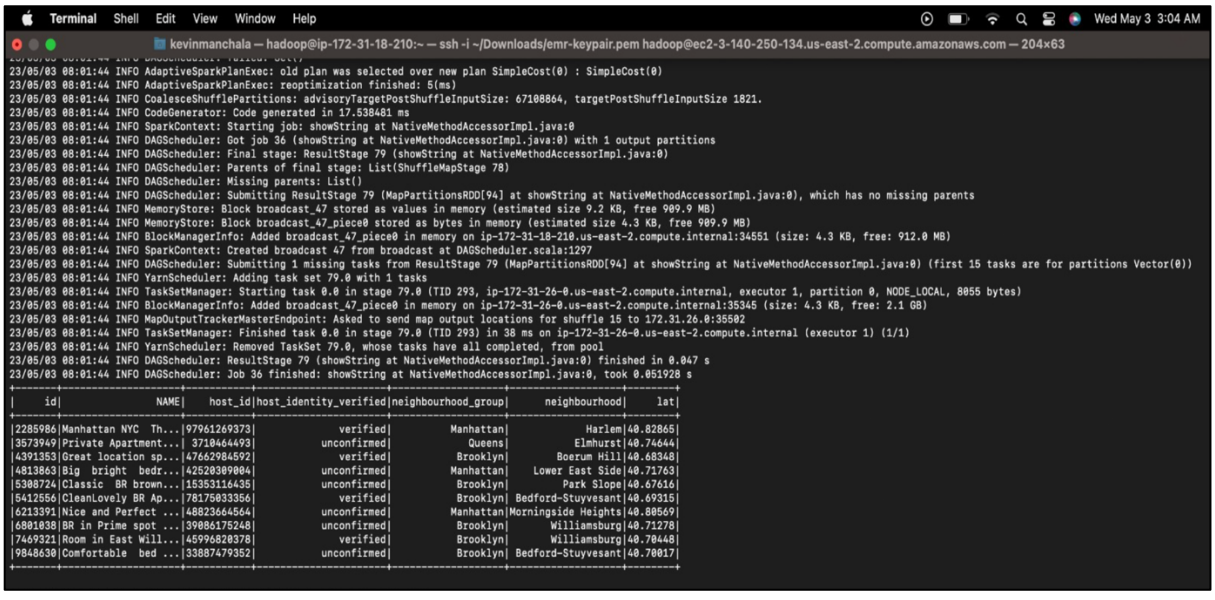
  - Running code in the EMR Terminal :



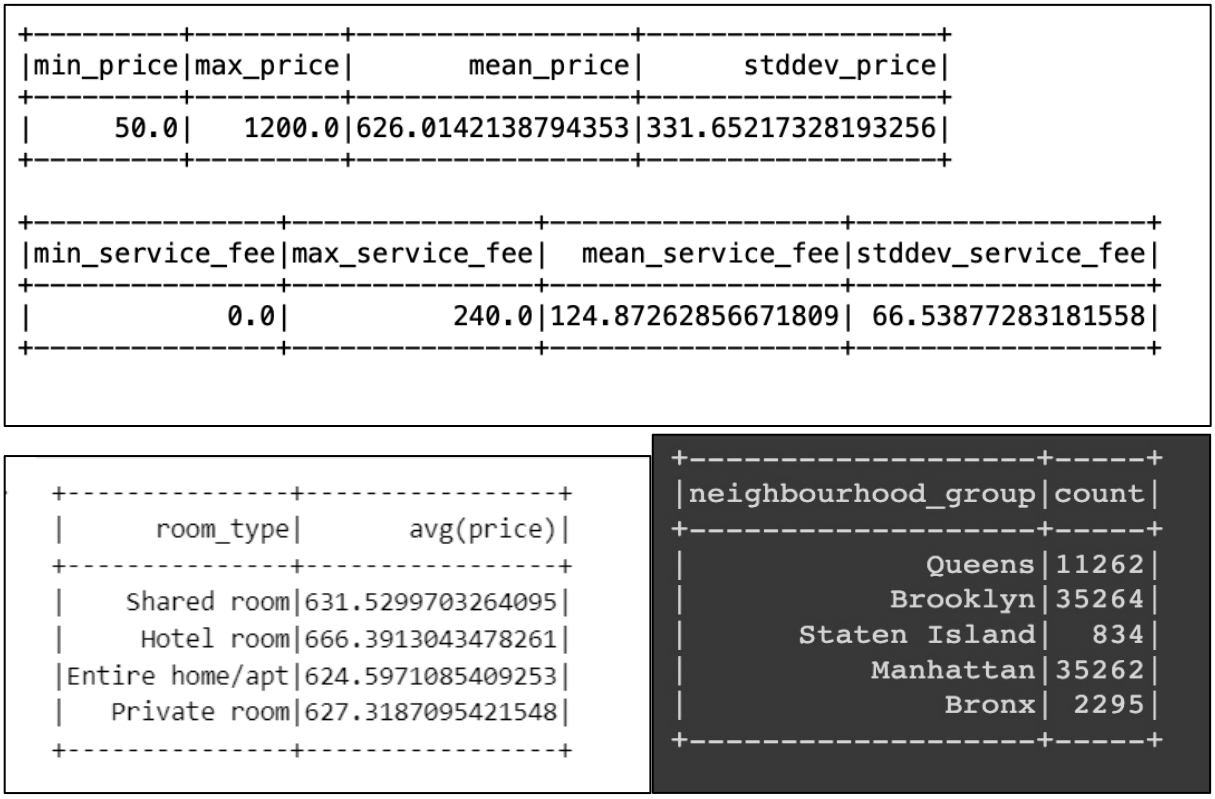
- Before cleaning the data. Below is the output of table in EMR.



- After cleaning the data. Below is the output of table (shown only 7 columns) in EMR



- **Data Profiling :** We performed data operations to change the formats of columns to read them in Spark. We renamed columns in the dataset to perform data operations and perform analysis. Handling missing or null values by replacing null values with min averages and mean of data to get the most accurate results. [1]



```
+---------+---------+-----------------+------------------+
|min_price|max_price|       mean_price|      stddev_price|
+---------+---------+-----------------+------------------+
|     50.0|   1200.0|626.0142138794353|331.65217328193256|
+---------+---------+-----------------+------------------+

+---------------+---------------+------------------+------------------+
|min_service_fee|max_service_fee|  mean_service_fee|stddev_service_fee|
+---------------+---------------+------------------+------------------+
|            0.0|          240.0|124.87262856671809| 66.53877283181558|
+---------------+---------------+------------------+------------------+
```

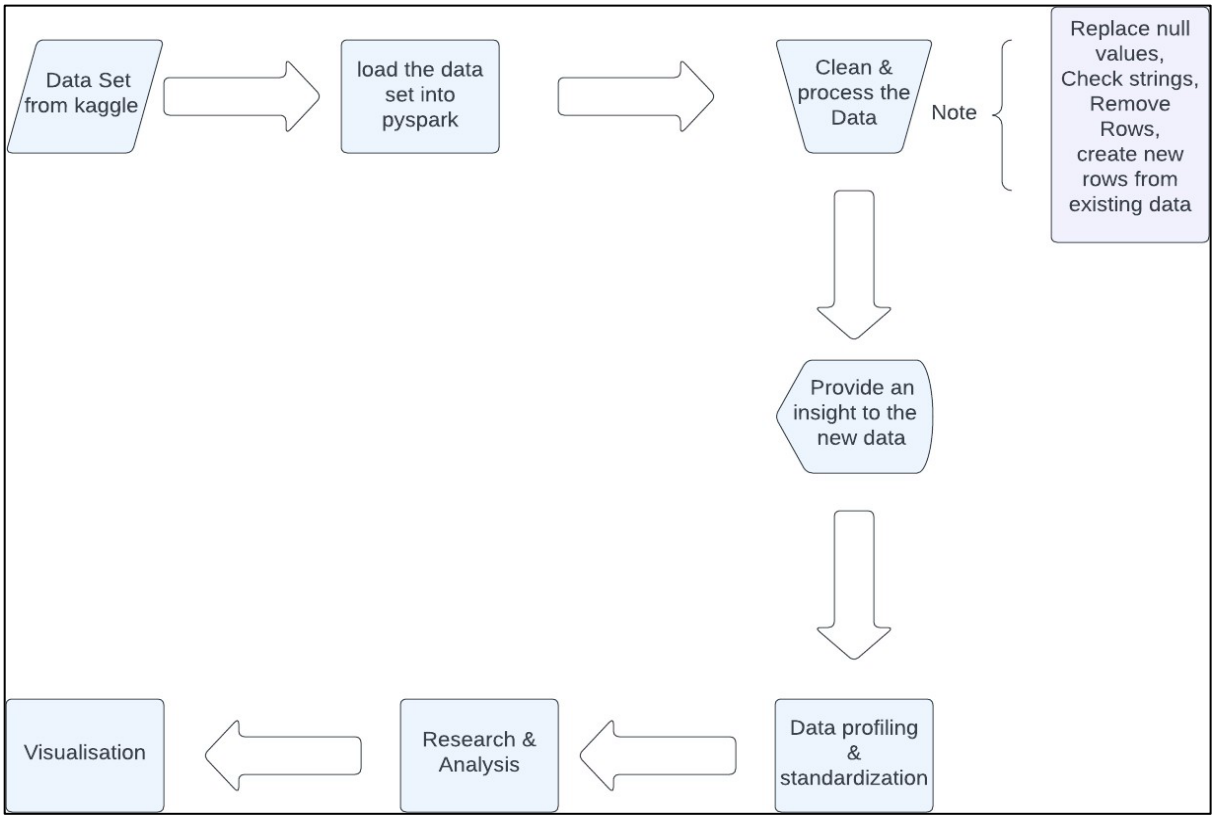
```
+---------------+-----------------+
|      room_type|       avg(price)|
+---------------+-----------------+
|    Shared room|631.5299703264095|
|     Hotel room|666.3913043478261|
|Entire home/apt|624.5971085409253|
|   Private room|627.3187095421548|
+---------------+-----------------+
```

```
+-------------------+-----+
|neighbourhood_group|count|
+-------------------+-----+
|             Queens|11262|
|           Brooklyn|35264|
|      Staten Island|  834|
|          Manhattan|35262|
|              Bronx| 2295|
+-------------------+-----+
```

- **Data Analytics :** We performed different operations on the fields to get the required analysis. We studied the data and made analysis of all the data.

- **Data Visualization :** We created Interactive Charts and Graph to represent the historical data and develop an analysis for the New York AirBnB data that have large commuters. We created data frames using PySpark and used those to plot graphs for the analysis.

Below is the Process Flow Diagram :



III. COMPARISION OF TOOLS:

PySpark: PySpark is a Python API for Apache Spark. It allows you to use Python to write Spark applications. PySpark supports a wide range of data processing operations, including SQL, machine learning, and graph processing. [3]

- Features :
  - PySpark is a Python-based API for Apache Spark. It allows you to use Python to write Spark applications.
  - It supports a wide range of data processing operations, including SQL, machine learning, and graph processing.
  - It is easy to learn and use, even for developers who are not familiar with Scala or Java.
- Benefits :
  - PySpark is a powerful and versatile tool for big data processing.
  - It is easy to learn and use, even for developers who are not familiar with Scala or Java.

- PySpark is well-supported by a large community of developers.

- Drawbacks :
  - PySpark can be slow for certain types of operations, such as batch processing.
  - PySpark can be memory intensive.

Hive: Hive is a data warehouse infrastructure built on top of Apache Hadoop. It provides a SQL-like interface for querying data stored in Hadoop. [9]

- Features :
  - Hive is a data warehouse infrastructure built on top of Apache Hadoop.
  - Hive provides a SQL-like interface for querying data stored in Hadoop.
  - Hive is easy to learn and use, even for developers who are not familiar with Hadoop or MapReduce.
- Benefits :
  - Hive is a powerful and scalable data warehouse infrastructure.
  - It is easy to learn and use, even for developers who are not familiar with Hadoop or MapReduce.
  - Hive is well-supported by a large community of developers.
- Drawbacks :
  - Hive can be slow for certain types of operations, such as interactive querying.
  - Hive can be memory-intensive.
  -

Scala: Scala is a general-purpose programming language that is designed for high performance and scalability. It is a compiled language, which means that it is faster than interpreted languages like Python. Scala is a functional programming language, which makes it well-suited for data processing tasks. [8]

- Features :
  - Scala is a general-purpose programming language that is designed for high performance and scalability.
  - Scala is a compiled language, which means that it is faster than interpreted languages like Python.
  - Scala is a functional programming language, which makes it well-suited for data processing tasks.
- Benefits :
  - Scala is a powerful and versatile programming language.
  - It is fast and scalable.

- It is well-suited for data processing tasks.

- Drawbacks :
  - Scala can be difficult to learn for developers who are not familiar with functional programming.
  - Scala can be memory intensive.

IV. OBSERVATIONS AND ANALYSIS:

AirBnB Dataset : We can see the top 10 rows of the dataset with attribute headers and its schema just after loading it through the file. [5]

id	NAME	host_id	host_identity_verified	neighbourhood_group	neighbourhood	lat
1210105	ALL ABOUT A VERY ...	42067422922	verified	Brooklyn	Fort Greene	40.69088
1244348	City Room Semi P...	39254540312	unconfirmed	Manhattan	Harlem	40.81156
1247110	HarlemHamilton He...	58292705877	unconfirmed	Manhattan	Harlem	40.82426
1249319	Entire Apt in Hea...	49315683762	verified	Brooklyn	Williamsburg	40.71534
1252080	Red Room for two ...	43465301578	verified	Brooklyn	Bedford-Stuyvesant	40.6798
1398992	Prime Williamsbur...	93988661910	verified	Brooklyn	Williamsburg	40.72059
1474105	Furnished Bedroom...	67288825352	verified	Manhattan	Harlem	40.80637
1575728	Lovely bdrm in P...	76284357302	verified	Brooklyn	Prospect Heights	40.67763
1638138	Lovely BR Midtow...	73838785546	unconfirmed	Manhattan	Midtown	40.75632
1688950	BR Village day...	60729843423	verified	Manhattan	Greenwich Village	40.7311

Schema of the dataset after we did couple of modifications to it.

```
root
 |-- id: string (nullable = true)
 |-- NAME: string (nullable = true)
 |-- host_id: string (nullable = true)
 |-- host_identity_verified: string (nullable = true)
 |-- neighbourhood_group: string (nullable = true)
 |-- neighbourhood: string (nullable = true)
 |-- lat: double (nullable = true)
 |-- long: double (nullable = true)
 |-- instant_bookable: string (nullable = true)
 |-- cancellation_policy: string (nullable = true)
 |-- room_type: string (nullable = true)
 |-- construction_year: integer (nullable = true)
 |-- price: double (nullable = true)
 |-- service_fee: double (nullable = false)
 |-- minimum_nights: integer (nullable = false)
 |-- num_reviews: integer (nullable = true)
 |-- last_review: string (nullable = true)
 |-- reviews_per_month: double (nullable = false)
 |-- review_rate_num: double (nullable = false)
 |-- calculated_host_listings_count: double (nullable = false)
 |-- availability_365: integer (nullable = false)
```

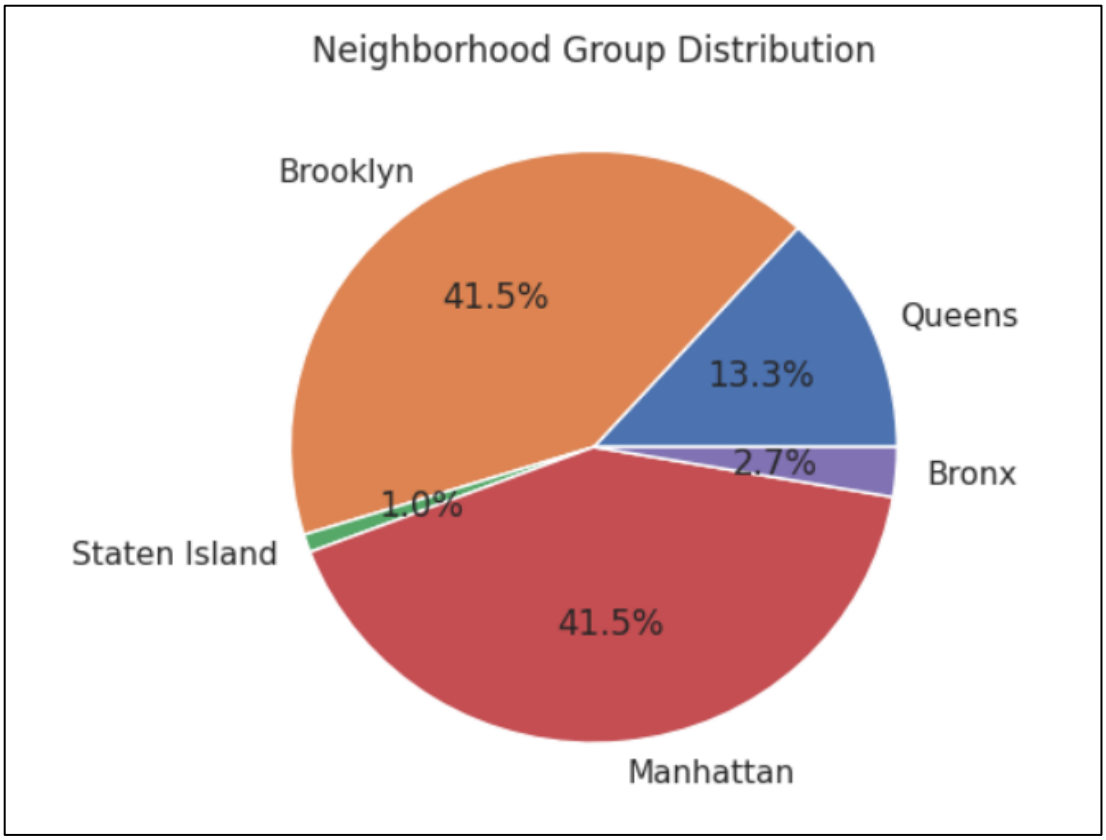
V. VISUALIZATION OF THE NYC AirBnB DATASET :

We used various visualization techniques to explore the relationships between different variables in the dataset. These visualizations helped us identify patterns and trends in the data and gain insights into the characteristics of the Airbnb market.

1. Neighborhood group distribution.

This is a pie chart showing the distribution of listings by neighborhood group, with the percentage of listings in each group labeled. [6]

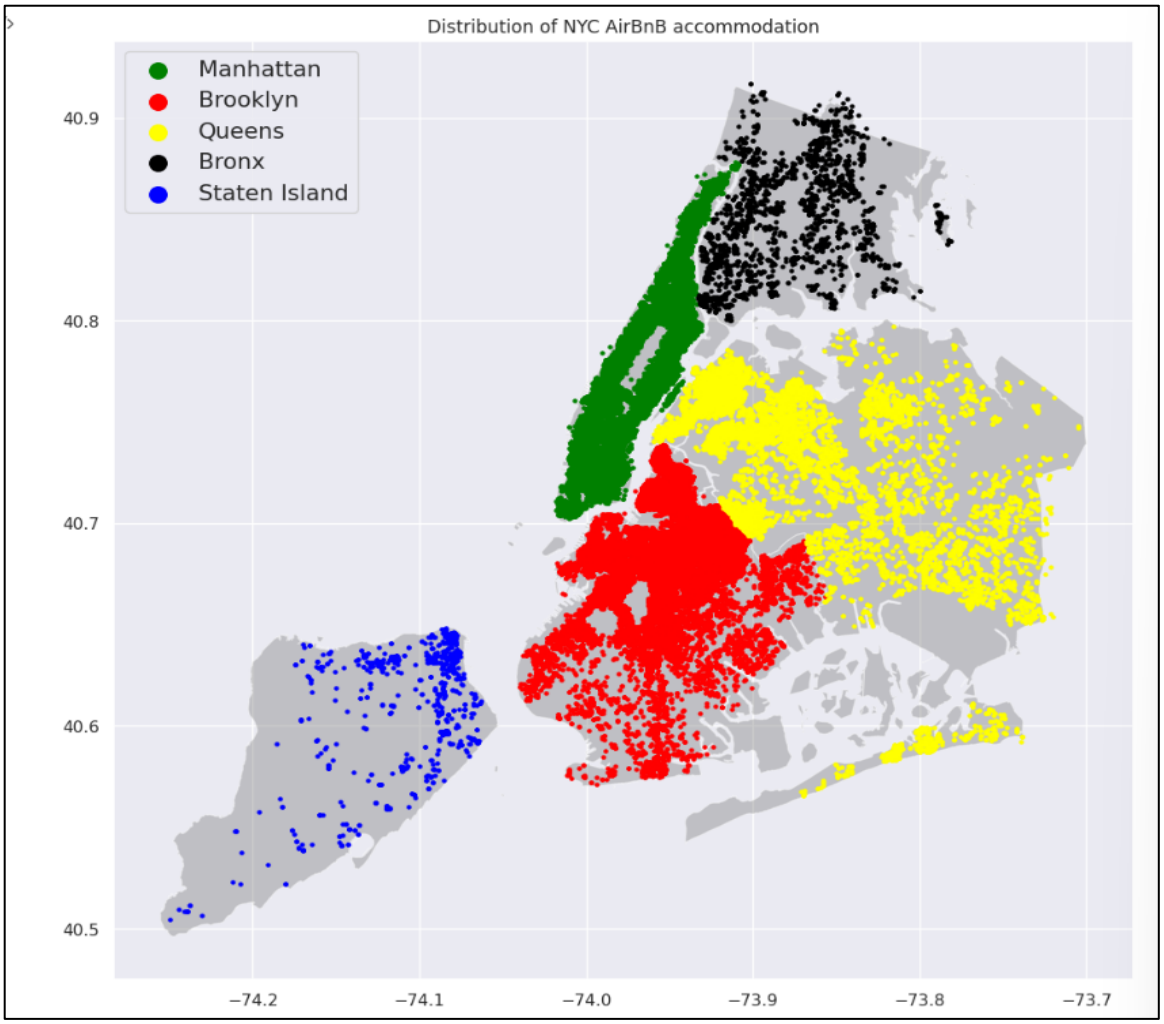
Brooklyn and Manhattan amount to 62 percent of the total listings precent in the New York however Staten Island proves to be lowest.



2. Density of Airbnb listings across neighborhoods.

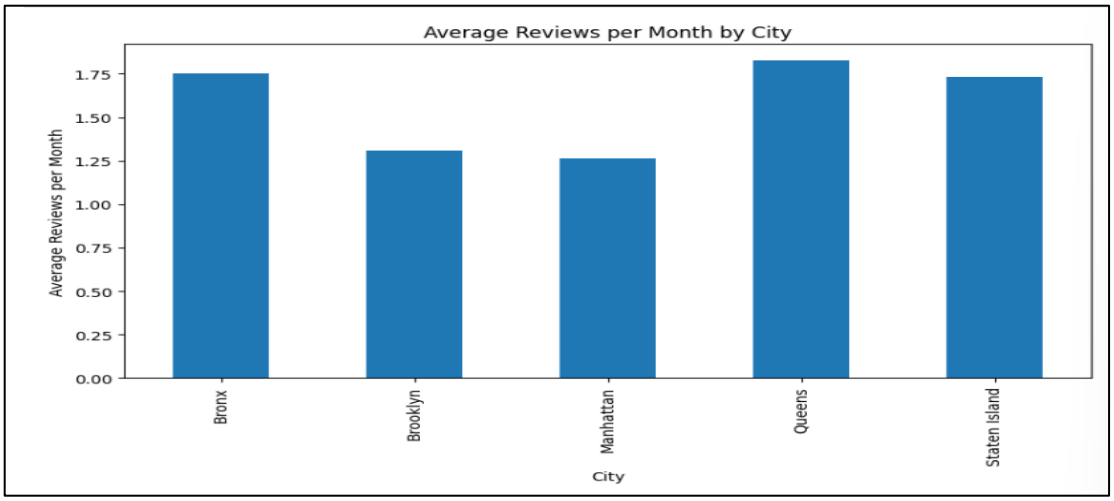
This graph is a 2D kernel density plot of the latitude and longitude coordinates of Airbnb listings using the seaborn library. The resulting plot shows the density of Airbnb listings across different neighborhoods in the chosen city, with warmer colors indicating higher density areas. The plot is given a title and displayed using the matplotlib.pyplot library. [6]

The next diagram shows the listing of New York according to the neighborhood type. [6]



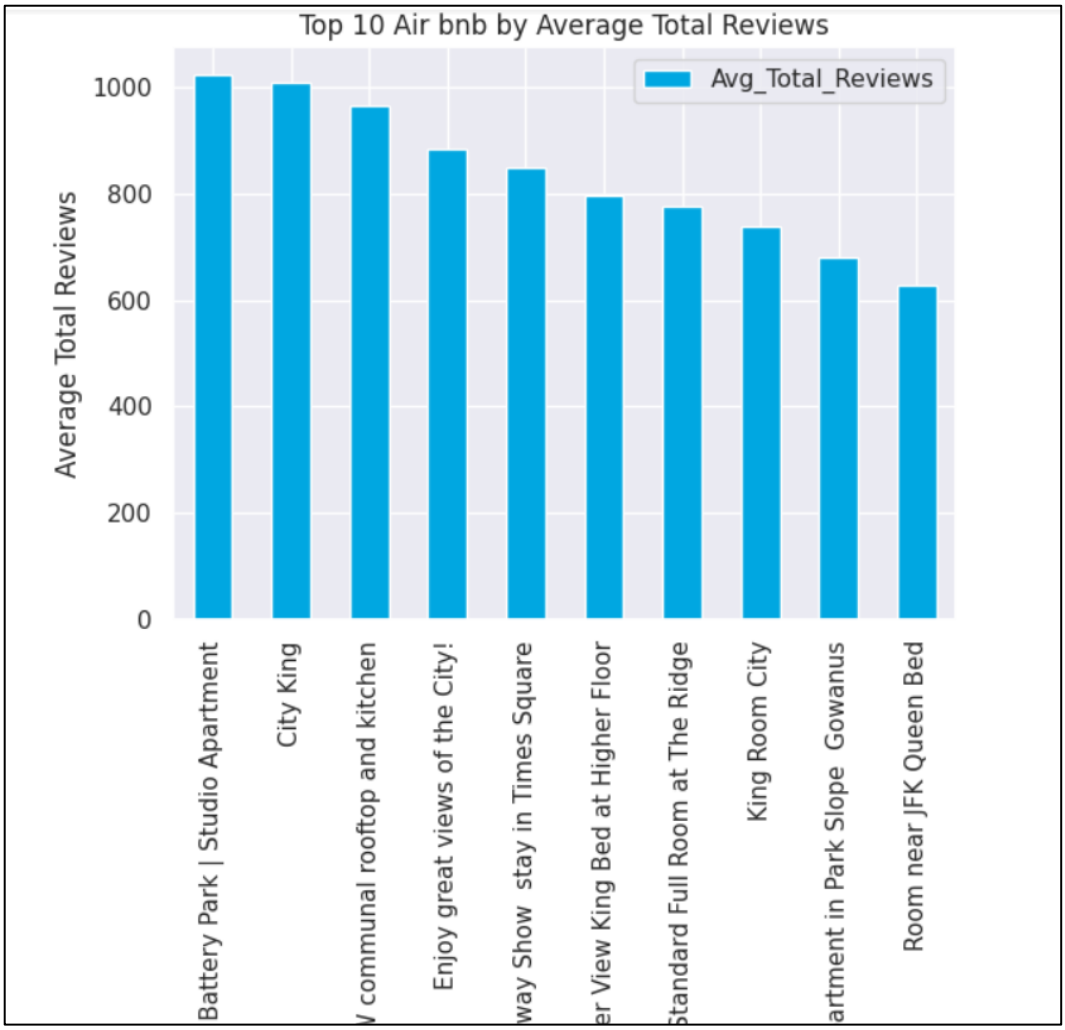
3. Reviews per month by city

The data shows the analysis to display the below graph is between neighbourhood_group and average reviews per month. This displays the average number of reviews per month for each city in the dataset. It helps to understand truly if a listing has been considered most popular throughout the time frame instead of seasonal holidays. [6]



4. Top 10 hosting places vs Average Total Reviews, Top 10 places here are taken based on the average total reviews for each place.

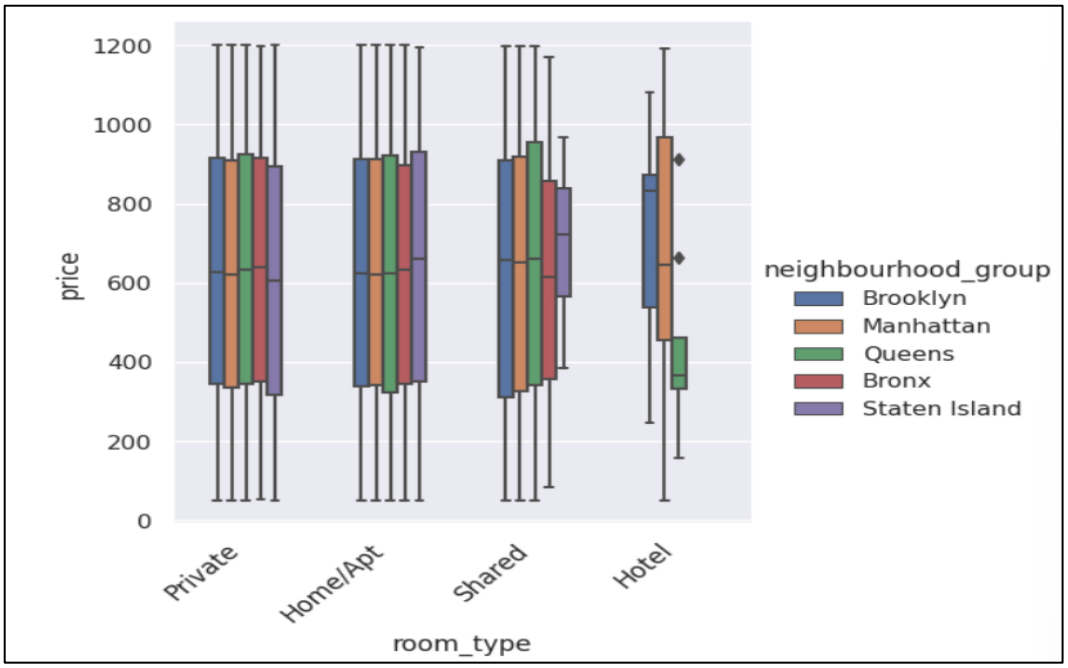
Below graph represent the top 10 hosting places completely based on the average reviews. The highest is the studio apartment from Battery Park. The average total reviews are 1000 plus . [6]



5. Price vs room type for different neighborhood.

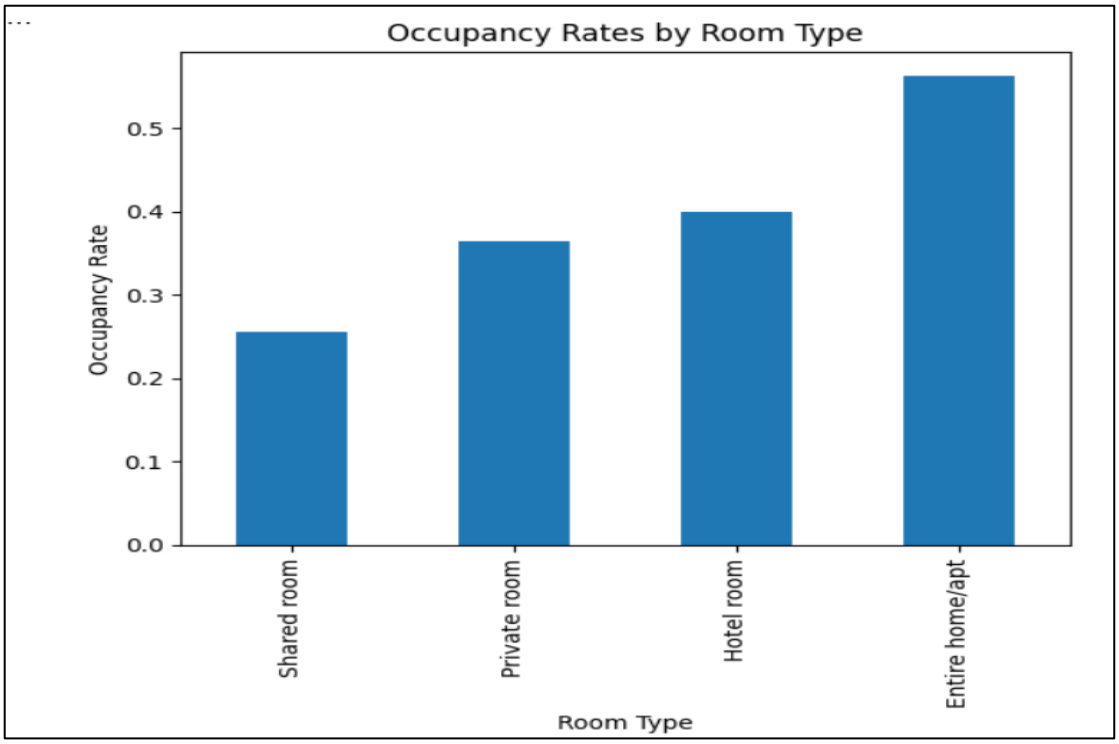
Showing the distribution of prices for different room types across various neighborhoods in a given dataset. The x-axis shows the room type , the y-axis represents the price, and the hue parameter groups the data by neighborhood. [6]

According to the data , private and apartment maintain similar min and max prices for all the neighborhoods . whereas Staten Island has the short range in shared rooms but also it has good ratings from the above radar plot. We can infer that cheapest hotel rooms can be found in queens



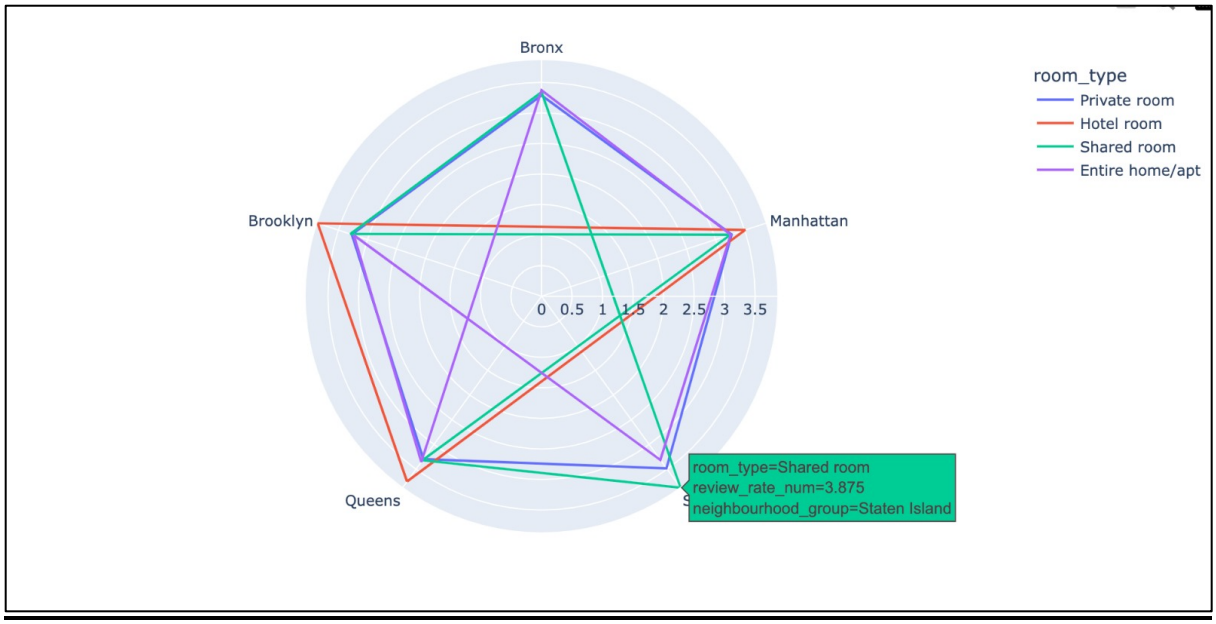
6. Occupancy rates by room type

The data shows the analysis to display the occupancy rates for each property type in the dataset. It calculates the occupancy rate for each listing based on the number of reviews and no of nights stayed, groups the data by property type, calculates the mean occupancy rate for each type. Number of reviews and nights stayed helps us to corelate that users prefer that listing based in the type of data we have. In this plot Entire home or apartment seems to be highest. [6]



7. Ratings per neighbourhood type

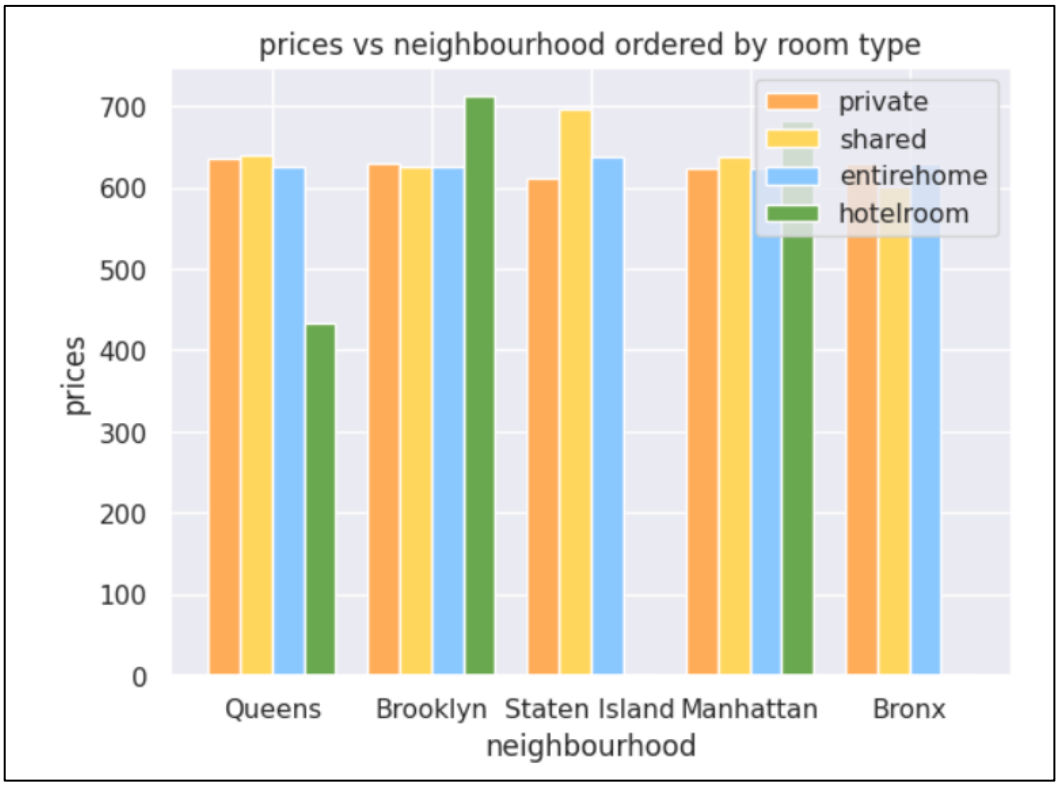
This radar plot helps us to figure out ratings against the neighbourhood type . As provided in the plot, shared room is highly rated only in Staten island, whereas hotel room is considered a popular opinion in Brooklyn and queens. The listings in Bronx seems to look like all the room types maintain an average rating of 3.0. Brooklyn has the highest rating of 4.0 for hotel rooms . [6] [7]



8. Prices vs neighbourhood ordered by room type.

The Data shows the analysis of the prices and the neighborhood ordered by room type in the bar plot shown below. [6]

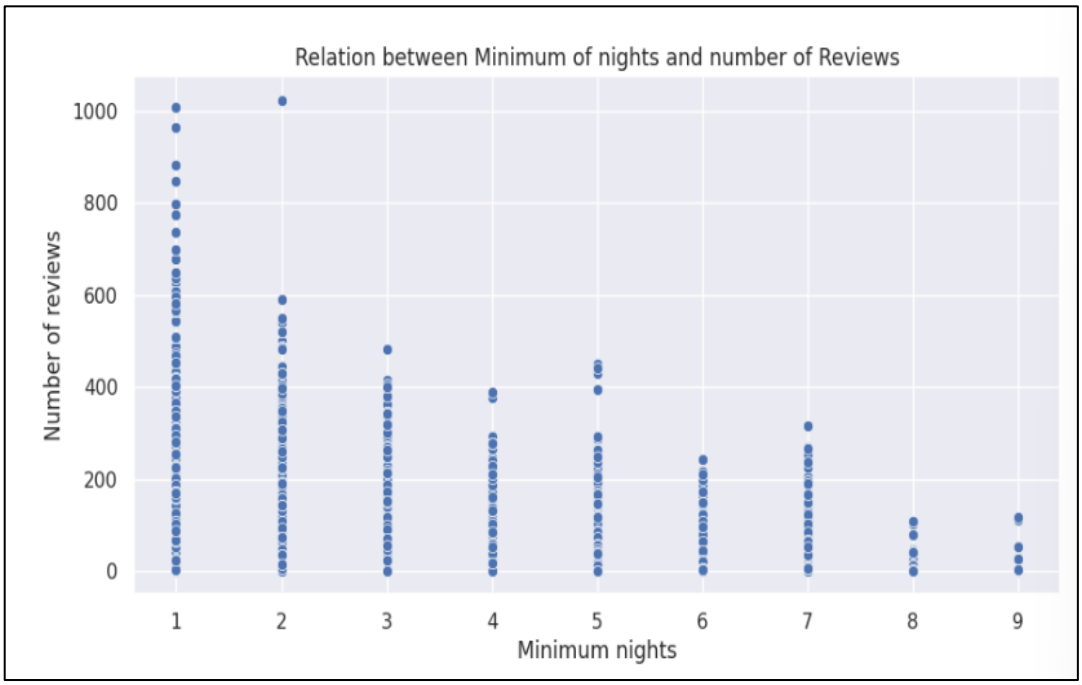
As we can infer from the plot , Brooklyn has the highest priced hotel room whereas queens has the lowest.



9. Minimum number of nights vs number of reviews.

This code removes outliers from a Pandas Data Frame by filtering values that fall outside the upper and lower bounds of the interquartile range (IQR), and then creates a scatter plot of the remaining data points, with 'minimum nights' on the x-axis and 'num_reviews' on the y-axis, to show the relationship between these two variables. The plot is also labeled with a title, x-axis label, and y-axis label. [6]

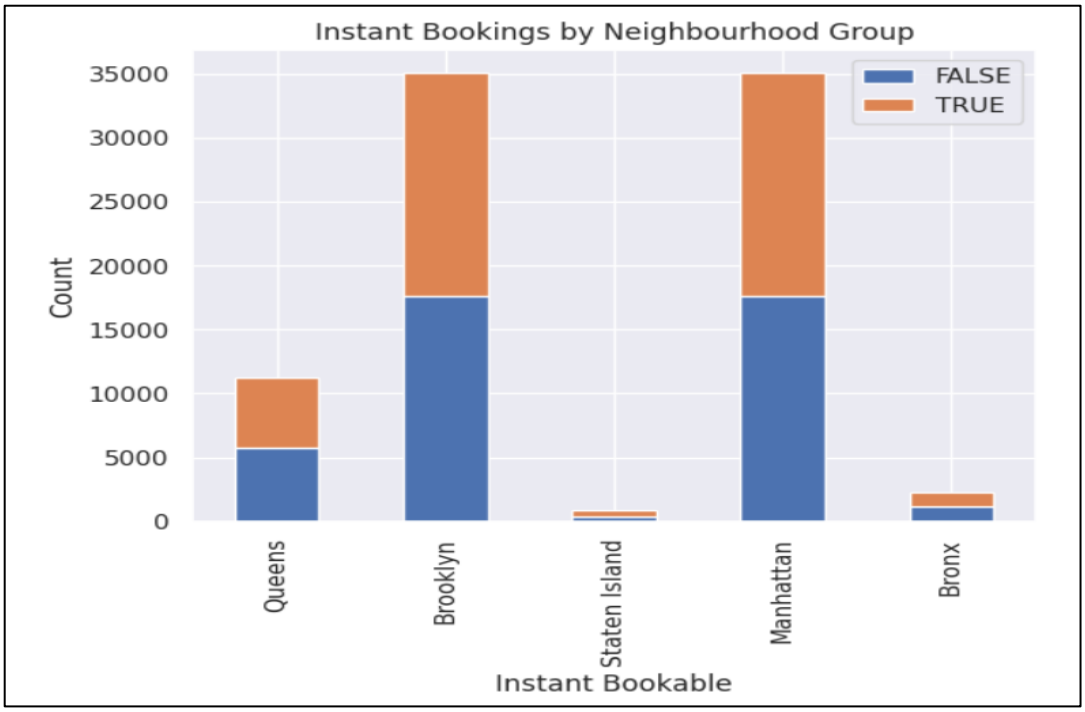
This graph infers that rooms that are stayed with lower number of nights are most lodged in. With minimum nights as 1 being the highest sort after



10. Instant bookings vs neighborhood group.

This diagram shows and groups data by neighbourhood group and instant bookable as true or false. X axis shows the total no of listings in the neighborhood , Y axis shows different neighborhood types. [6]

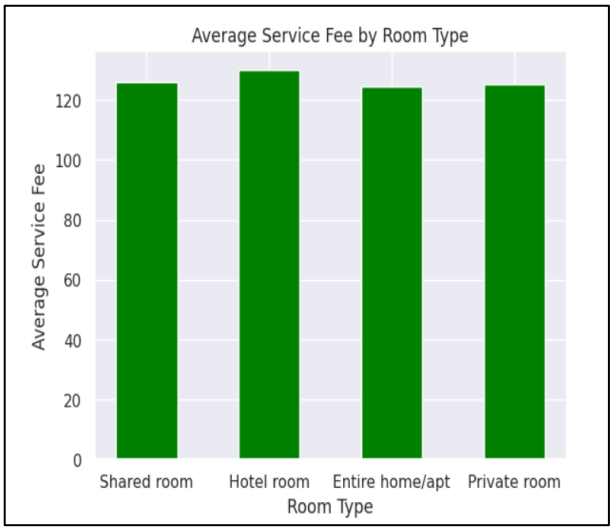
The graph infers out Manhattan and Brooklyn has equal proportion of instant bookable comparatively



11. Average service fee by room type.

The first bar chart to display the average review score by neighbourhood in the dataset. With the x-axis showing the neighbourhood group and the y-axis showing the average review score. [6]

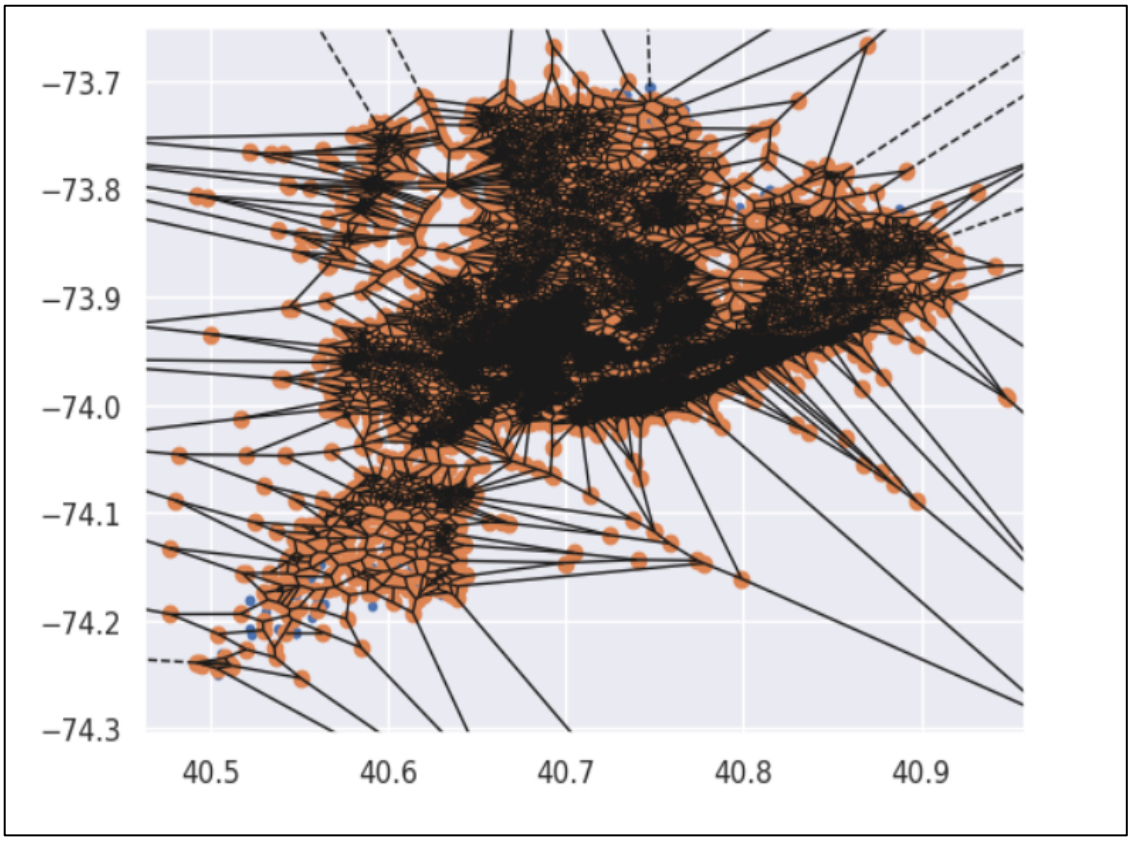
We can infer that service fee is charged highly in the hotel room as the hotel rooms usually provide more amenities compared to the rest of room type.



12. **Price vs number of reviews by room type** : Below is the price vs number of reviews scatter plot for every room type differentiated based on color. [6]

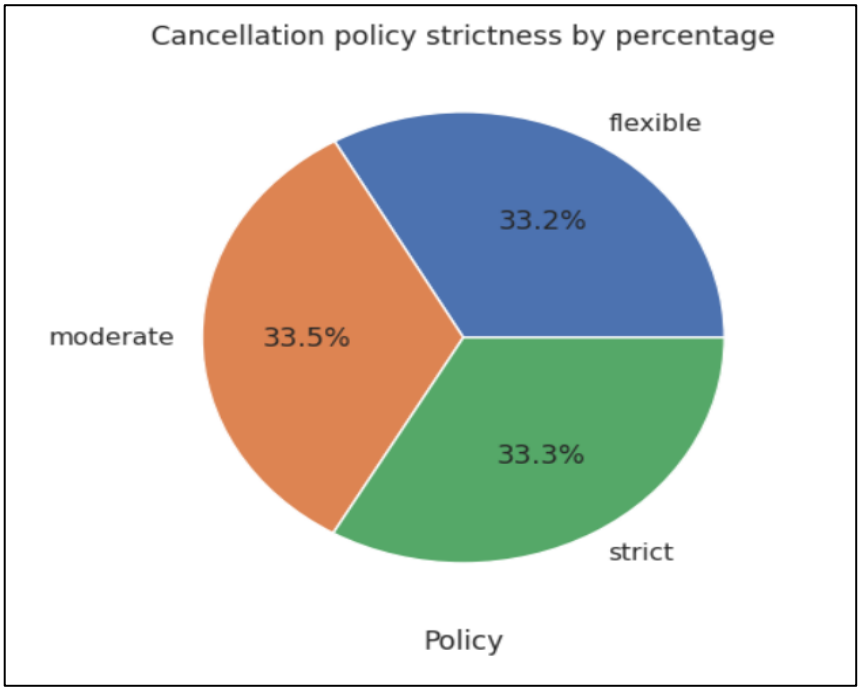
13. **Voronoi Diagram for distribution of data with latitudes and longitudes.** [6]

The Voronoi diagram helps to visualize the distribution of Airbnb listings across a geographic area. The Voronoi diagram is usually used for aviation purposes to find the nearest route to destination, however we can include this to get the closest attraction to your listing



14. Cancellation policy strictness.

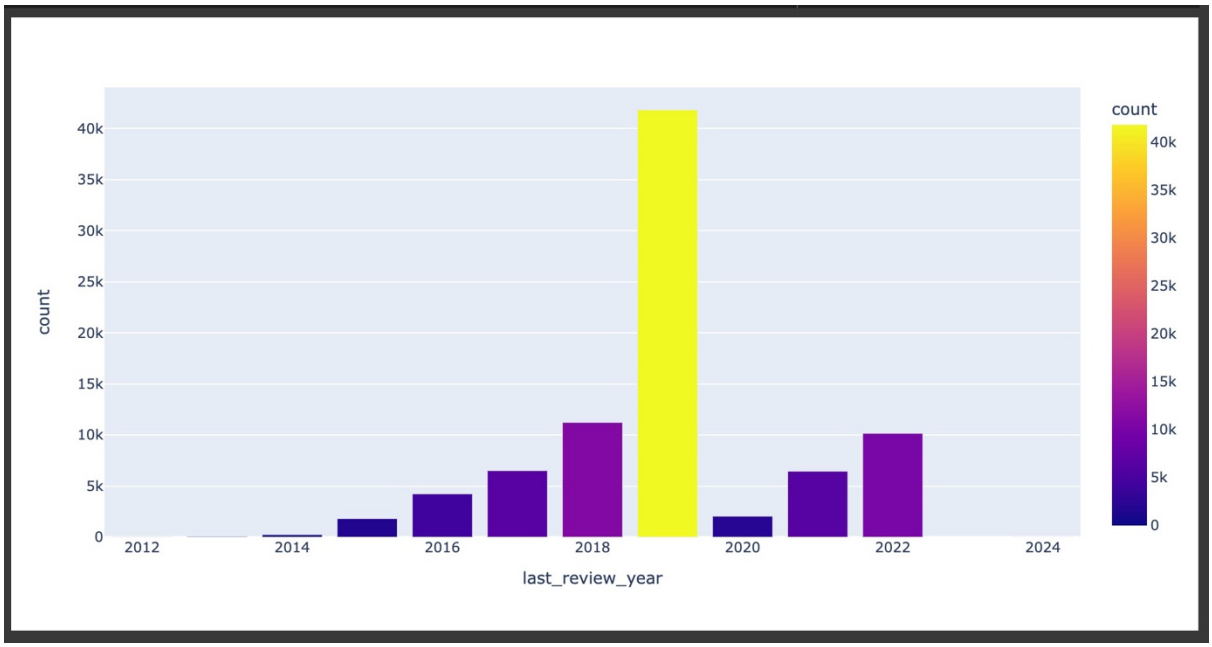
Below pie chart analyze the data by 'cancellation_policy' and counts the number of 'id' for each policy. The resulting chart displays the percentage of each policy type. [6]



15. Yearly based count :

This graph depicts yearly based number of reviews for the Air BnB's with x-axis as count of reviews and y-axis is year of the review.

We can infer the growing trends in booking the AirBnB until 2019 and with a sudden drop in 2020 probably due to covid and it started picking up in 2022.



VI. Conclusion :

In conclusion, this project involved cleaning and displaying a dataset of Airbnb listings from New York. We were able to deal with missing numbers, get rid of duplicates, and fix data inconsistencies thanks to the data cleaning procedure.

We were able to learn more about the characteristics of the postings and how they relate to different variables like location, property type, and price through exploratory data analysis and data visualization. The two most common types of real estate were apartments and homes, and we discovered that costs varied greatly depending on the location, number of bedrooms, and facilities provided.

We also discovered a number of additional potential future for this project, including user segmentation, sentiment analysis, time-series analysis, spatial analysis.

Overall, this project gave useful information about the Airbnb market's numerous trends and patterns. These insights can be used by both hosts and guests to improve their listings and make better educated selections when it comes to lodging.

VII. Future Scope :

Here are some future scopes based on the analysis and data cleaning done for this project:

1. **Time series analysis :** You can analyze the data over time to identify trends and seasonality in the Airbnb market. This can help hosts to better understand how to price their properties based on seasonal demand.

2. **Sentiment analysis :** Examining consumer satisfaction data to spot patterns and trends. This might aid Airbnb hosts in updating their listings and giving their visitors a better overall experience.

3. **Spatial analysis :** We can use geospatial techniques to analyze the data and identify how touristic locations and price are correlated. This can help to identify hotspots of Airbnb activity.

VIII. References:

[1] Python : https://docs.python.org/3/library/
[2] HDFS Architecture Guide : https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[3] PySpark : https://spark.apache.org/docs/latest/api/python/
[4] AWS S3 : https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html
[5] Data set : https://www.kaggle.com/airbnb-data-cleaning-visualization/
[6] Python Library :
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.violinplot.html
https://seaborn.pydata.org/generated/seaborn.pairplot.html
https://seaborn.pydata.org/generated/seaborn.lmplot.html
[7] Plotly : https://plotly.com/python/
[8] Scala Documentation : https://www.baeldung.com/scala/scaladoc
[9] Hive Documentation : https://cwiki.apache.org/confluence/display/HIVE