NYC Airbnb's: Travel Stay Recommendations

1st Siju Chacko 2nd Mythri Shivakumar 3rd Rebecca Abraham

I. Problem Statement

New York city is a big center for tourism, people traveling from different states to people coming in from other countries. Being able to find an affordable and good stay is the number one factor during travel. Airbnb's are one such form of stay that provides short and long-term options. It provides people with a much more affordable stay along with providing an authentic view of the city and providing other local experiences.

However, there are a lot of concerns about the availability of good accommodation among the different Airbnb's. Some of the issues being customer experiences, the availability of listings, booking rates, period of stay and number of customer reviews.

The objective of this project is to see how these above-mentioned issues affect the Airbnb selection. We will compare and understand how different aspects such as locations, costs, availability and reviews affect the overall choice of an Airbnb, and the different trends associated with it.

II. Potential of the Project

In this project we aim to provide information for two groups of people, one being the hosts that provide the Airbnb service, and second the customers or guests who utilize the service.

We hope to gather information about the different aspects of Airbnb's and what factors affect the selection of an Airbnb and how these factors can help the hosts understand what type of rooms to provide and what location or cost is the most favored.

In turn the insight gained can be used to increase/enhance the Airbnb selections where the hosts understand what most customers are looking for in terms of rooms, being it a single room, an entire house or a shared space. This can encourage the hosts to increase the service in certain areas based on the information gathered in this project.

Along with the hosts the information collected can be provided to customers/users to be able to get a view of the good options or what factors to pay attention to when choosing an Airbnb, in terms of location, cost, different trends in the availability of listings or period of stays.

Overall, this information can benefit the Airbnb community by understanding customer needs and allowing hosts to provide better service catered to the guests and hence having more flexible stay options, bringing in more travelers/tourism that can access more affordable options and in turn increase economic growth of the city.

III. Data Sources

Dgomonov. New York City Airbnb open data. Kaggle. (2019, August 12). Available at: https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data

IV. Data cleaning/Processing

A. Cleaning Step I: Dropping Duplicates

The dataset is raw; in this case, there are many possibilities for it to contain duplicate data. The reason we must do that the final model is not biased in the case where duplicate data count is more than unique data. We do this step by using the $drop\_duplicates$ method.

B. Cleaning Step II: Handling missing values

The dataset is vast and there is no certainty that all the cells are filled with values. Often, the data is collected manually, and several fields can be ignored or can be a result of manual error. Having such missing values in the data can cause accuracy issues and cause the model to be biased. Therefore, it is necessary to handle these missing values properly. In the case of our dataset, the following methods were employed:

- Checking the count of missing values: $.isnull().sum()$
- Filling object data type missing values with a placeholder: In our dataset, fields - name, and host name, had missing values, and filling them with a statistical value would not work and dropping of them would also not be very beneficial, so placeholders like 'No Name Provided' and 'Unknown Host' has been put in place using $.fillna()$ method.
- Filling number datatype missing values: First, we notice that the $last\_review$ feature is categorized as int while it belongs to $datetime$, so after conversion we fill the missing value with forward fill. We do this as an assumption that the dataset is chronological, and forward-filling would be slightly accurate. We do this using $.fillna(method =' ffill')$. Another reason we do this is because we have over 10000 missing values in this column and dropping all those rows would potentially reduce the accuracy of our model. Next, we fill $reviews\_per\_month$ with 0 if it is a missing value as this also has over 10000 missing values in this column, and dropping all those rows would potentially reduce the accuracy of our model.

C. Cleaning Step III: Standardize Object features

When data is collected, there are several possibilities for extra spaces to make it to the string, but in the model, it will be treated new value, to avoid that and to group all the fields that are the same together we standardize the strings by stripping the white spaces and then to maintain uniformity, we convert all the strings into their title case.

D. Cleaning Step IV: Handling outliers within columns

Now when we try to make inferences from the columns and values, we will be unable to make proper conclusions if there are outliers, so for each number column, we are finding the outliers based on quantiles and then clipping the outliers to the low bound or the high bound depending on their difference,

E. Cleaning Step V: Valid Geographical Points

This cleaning step was focused on geographical data cleaning, since our data has a latitude and longitude column and our data is geographically located in NYC. We control the latitude and longitude parameters to match the geographical requirement and remove any listings outside this area.

F. Cleaning Step VI: Text Correction

This cleaning step was focused on correcting potential typos in one of the columns which dealt with identifying the specific borough in nyc which is very important information for a potential customer. We made sure to look for and replace any common typos we find in the neighbourhood group column so the customer can view the listings error free.

G. Cleaning Step VII: Date Corrections

This cleaning step was focused on creating a better view of the database, we wanted the user to separately see the month and year that the last review was written for a particular listing. This way a user can determine for themselves if they should be trusting that older listing or consider another place that has a more recent review.

H. Cleaning Step VIII: Feature Encoding

In this step of data processing/cleaning the step is to use feature encoding. Lot of the data when collected has categorical data such as strings that can be represented in a numerical form, that helps with processing the data better when run through algorithms. Here in our data we have converted the room type into codes $0, 1, 2$ in place for (Entire Home/Apt, Private Room and Shared Room that can used for easier reference as well as during graphical representations. In our data we have used the Ordinal encoding from sckit-learn, that helps keep the dimension of the dataset small compared to using one-hot encoding. Further this step can help with model training as numerical data will make the training of the model faster.

I. Cleaning Step IX: Feature Aggregation

In this step the data is further processed to aggregate the price and minimum_nights column to give the Total Price. This step helps with focusing on two from among the many features by combing them into one that can further reduce the compute time as well. In this case with our data the Total Price can be considered instead of comparing price and minimum_nights individually and hence save memory usage and simplify our view of the data. Along with that grouping the data such as into bins will help identity trends or patterns in the problem or even detect outliers not seen in raw data.

J. Cleaning Step X: Normalization of Data

In this last step of data cleaning we have normalized the numerical data. This step converts the features into equal weighted values all on the same scale. This processing helps with faster convergence of the model and increased generalization and accuracy. Also provides better visualization of the data as they are on one scale hence easier identification of patterns or trends.

V. Exploratory Data Analysis (EDA)

In this section we hope to present the data in terms of certain features that can help gather information that can educate the customers and hosts about the patterns in Airbnb options. Some of the features that we hope to dive into from our dataset are price, minimum_nights, Total_price, calculated_listings, neighbourhood groups and room_types

A. EDA Step I: Visualizing the Distribution of price per night

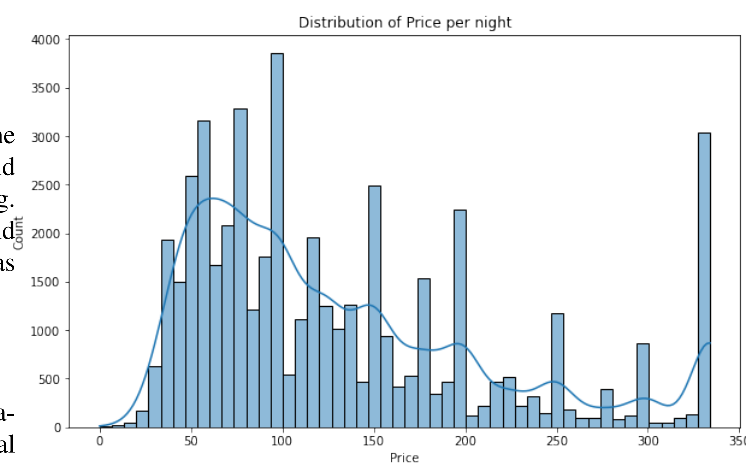

Figure 1. Distribution of Price Per Night

From the dataset, we notice that price has an effect on the amount of people renting the property, so we first visualize the distribution of price so that any new host will know which price range would be ideal for high customer rate. From this we know that when the price is < 100 per night, we have a higher customer traction, but oddly we notice that apart from prices less than 100 the next peak we notice when the price is a little shy of 350

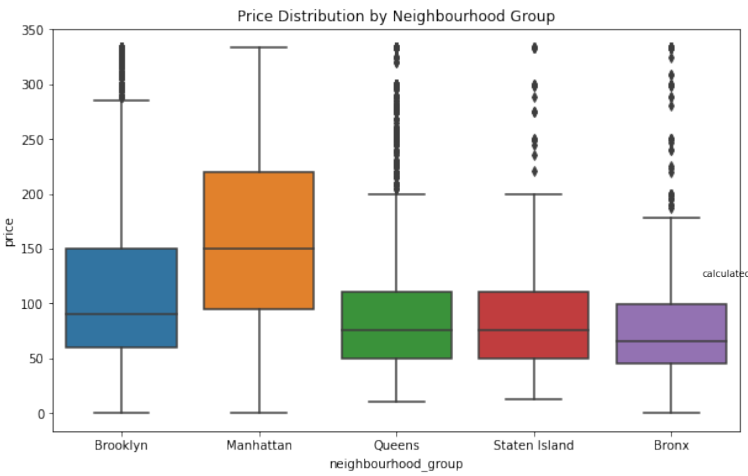

Figure 2. Price By Neighbourhood Group

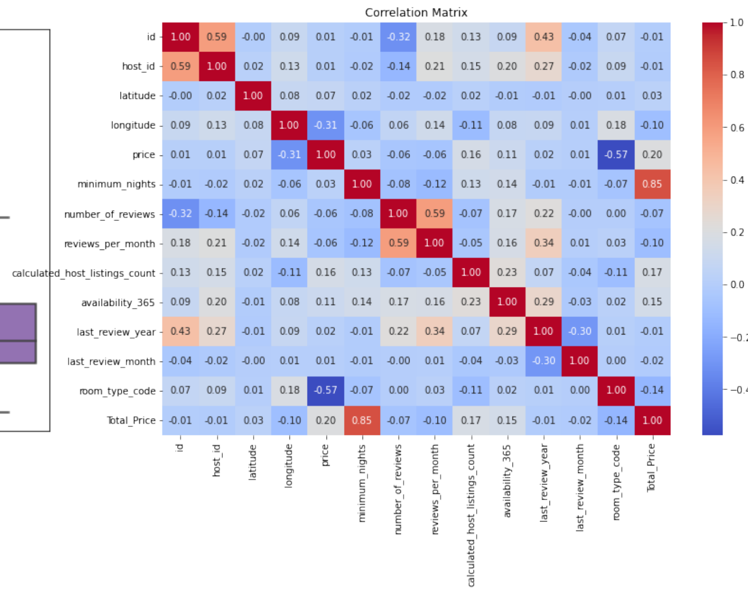

Figure 3. Correlation Matrix of the Features

B. EDA Step II: Visualizing the price distribution across areas

After our above visualization the host must know in which area what the trend of price is so they can accordingly price their accommodation and in order to do so we will plot the graph for the same. Here we notice that the price per night is almost the same across Queens, Staten Island and Bronx and this also majorly falls in the 125 and below range which falls in line with our price distribution. We also notice that Manhattan has relatively higher pricing while Brooklyn lies in the middle zone.[Fig. 2]

C. EDA Step III: Correlation Matrix

We notice that there are several feature columns present, but it is necessary to know what relationships exist between all of them, so to do so we plot a correlation matrix. From our current visualizations we are trying to find trends in price, so we look at the relationship of price with other features and we take the threshold as 0.07, we compare the absolute value of all with the threshold and we find: Features with low correlation with price below 0.07: ['minimum_nights', 'last_review_year', 'host_id', 'id', 'last_review_month', 'number_of_reviews', 'reviews_per_month'], so dropping those rows in the correlated dataset. [Fig. 3]

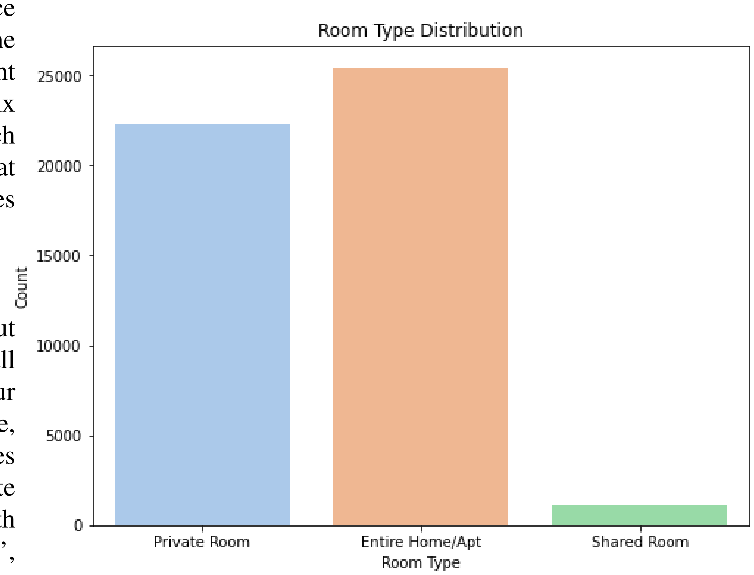

Figure 4. Room Type Distribution

D. EDA Step IV: Visualizing Room types

If a host is planning to put their house up, it is necessary to know what kind of houses are already present so they can also understand if their house is desired. From the above distribution we notice that shared rooms are mostly not preferred. There is a very small difference between a private room and entire home, but it will be beneficial for the host if they have an entire home or apartment.[Fig. 4]

E. EDA Step V: Visualize Reviews with Price

From the dataset it's easy to see that there are varying prices and it's hard to tell which one is the best for your money. So using a comparison between number of reviews and price allows us to make this decision easier. From the graph it's easy to see that the $50 - 100$ price has the most concentrated amount of reviews and the increased price doesn't correlate to a better place according to reviews. So a customer can be sure that it's good and they are not pressured to spend more money.[Fig. 5]

F. EDA Step VI: Neighbourhood Popularity on Reviews

When looking at the dataset it's hard to look through so many options and know which area is the best. Luckily there is a column for neighbourhood group which breaks it down by area but we still don't know which area that people are most likely to want to go to. That's why we have a graph for

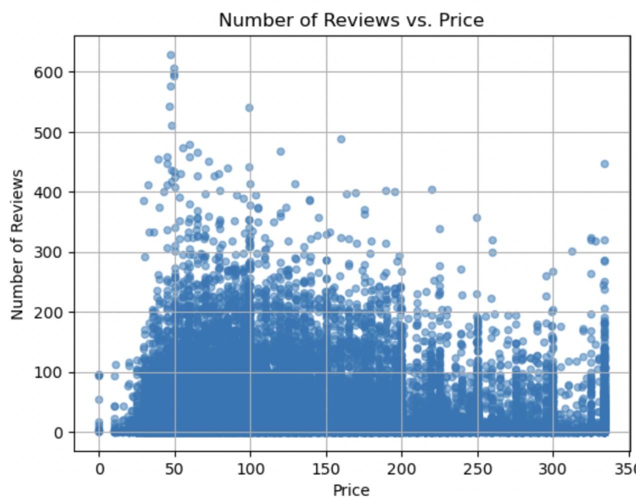

Figure 5. Distribution by Reviews and Price

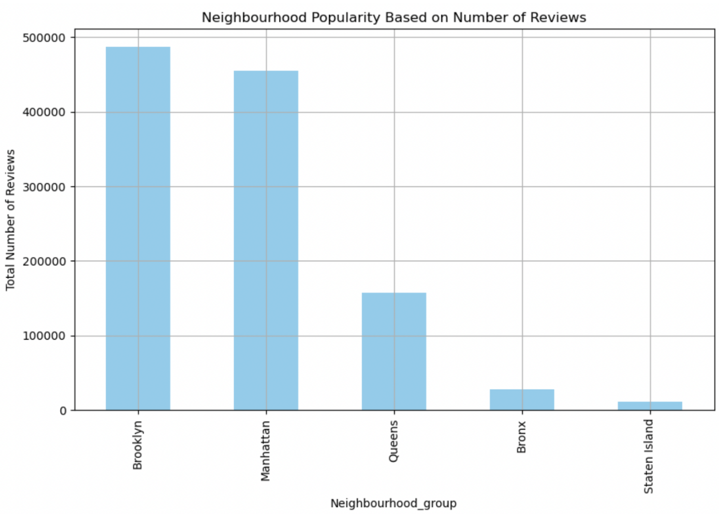

Figure 6. Neighbourhood Popularity Based on Number of Reviews

the neighbourhood popularity based on neighbourhood group and number of reviews. Based on the data, Brooklyn and Manhattan were a lot more popular than the other 3 boroughs. With this information people that weren't sure of where to go to or just aren't knowledgeable about NYC can make a more educated decision.[Fig. 6]

G. EDA Step VII: Geographical Visualization

When looking at the dataset it's hard to visualize where all the different listings are located, I know some websites provide a map which nicely show the customer where each listing is located geographically. That's why I decided to take the provided latitude and longitude and plot all the different listings on a graph so customers can see where most listing are located and more easily identify how many options they have for different locations. From the graph we see that most listings are located in the center of the NYC area. So we can

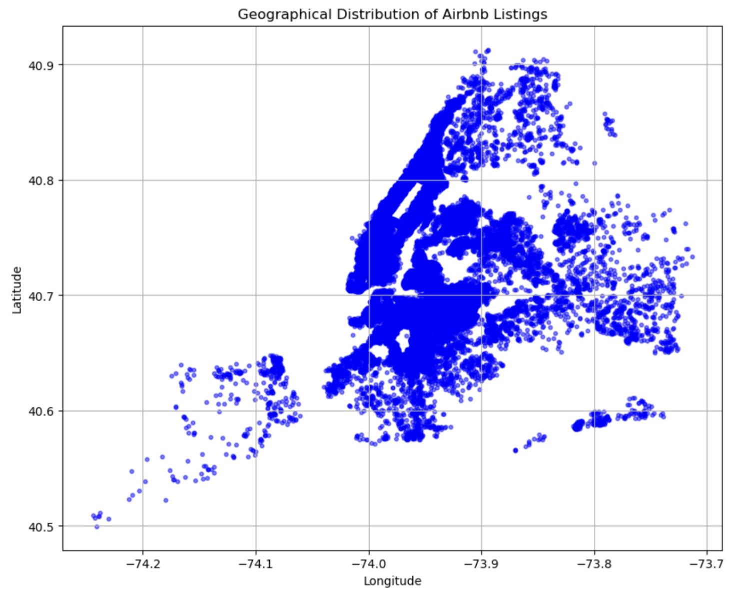

Figure 7. Geographical Distribution

make a conclusion that it will be harder for customers to get listings further away from the center but more chance to get something towards the center.[Fig. 7]

H. EDA Step VIII: Neighbourhood Price vs listings

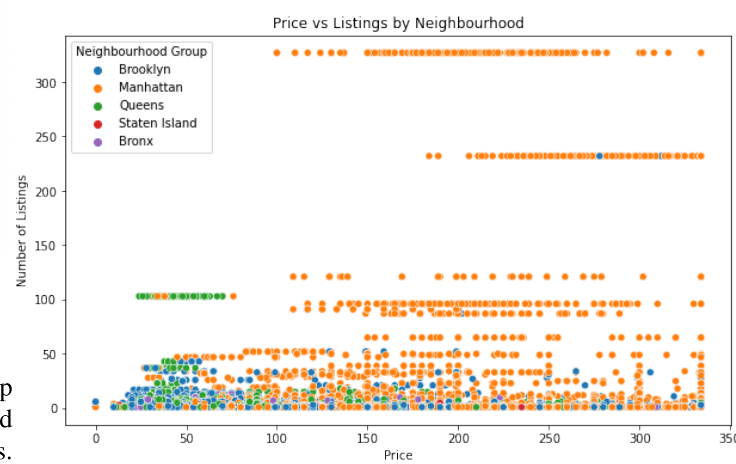

Figure 8. Price vs Listings in Different Neighbourhoods

In this graph visualization we have selected three features from the dataset, the price, number of listings and the neighbourhood group. We can see here that based on the Neighbourhood groups, Manhattan has the most number of listings of Airbnbs. We can also understand that most of the listings start at the price of $50 and go up all the way to $300 plus. From this we can understand the trend that Manhattan is a favoured neighbourhood for Airbnbs in NYC.

When considering the hosts who put up listings for stay we can see in the graph that it is comparatively higher in Manhattan. And then we can see the next following neighbourhood is

Brooklyn. And this can be concurred with the graph of Neighbourhood vs Price in Fig. 2 above that shows Manhattan first followed by Brooklyn then Queens, Staten Island and Bronx. We also learn that Bronx is least favoured area of stay and has the most limited listings. From the data plotted we see the most listings for all neighbourhoods fall in the range of 50 to 100 and then a few in the range of 230 to 300 plus.[Fig. 8]

I. EDA Step IX: Room Type by Neighbourhood vs Total Price

This graph is plotted based on the room types in terms of the total price and the neighbourhood. In this visualization too we can see how Manhattan is the most sought out area. Even based on the total price we can see that even though the price for the minimum nights is pretty high in Manhattan travellers still opt for that area a lot more compared to the other areas.

Another point to be noted is how based on room types the most chosen type is an Entire Home/Apt. This is common in all the neighbourhoods where an Entire house is much more preferred stay for a lot of people traveling to NYC. We can agree with this data based on our initial count plot[Fig. 4] as well that shows the distribution of room type.

From this we can gather the understanding that when new hosts want to put up new listings of a room the most preferred and sought after choice among the customers being it any neighbourhood, is the Entire Home/Apt. And if possible to have the listing in the Manhattan area which another favoured location.[Fig. 9]

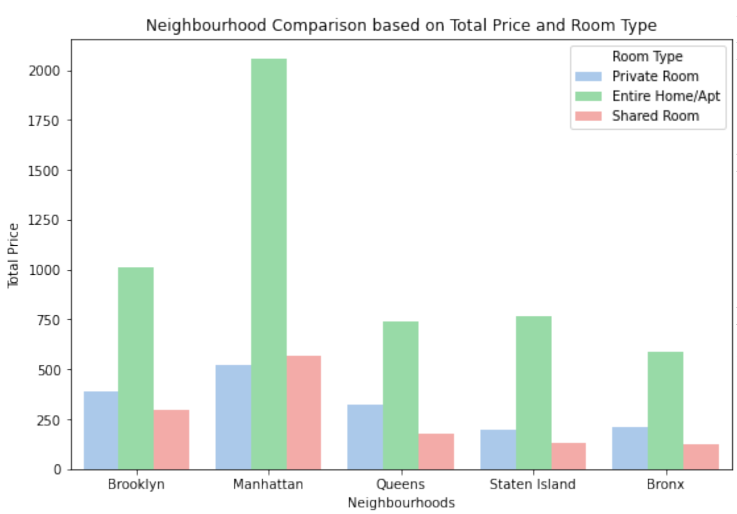

Figure 9. Room Type Distribution Based on Neighbourhoods vs Total Price

J. EDA Step X: Distribution of Last Reviews based on Room Types

In this last graph we are trying to gather some information on how the date of the reviews for the rooms have changed with years. Use this visualization to understand if there is much relation to how customers have further given reviews over the years. As we can see in the graph all three room types seem

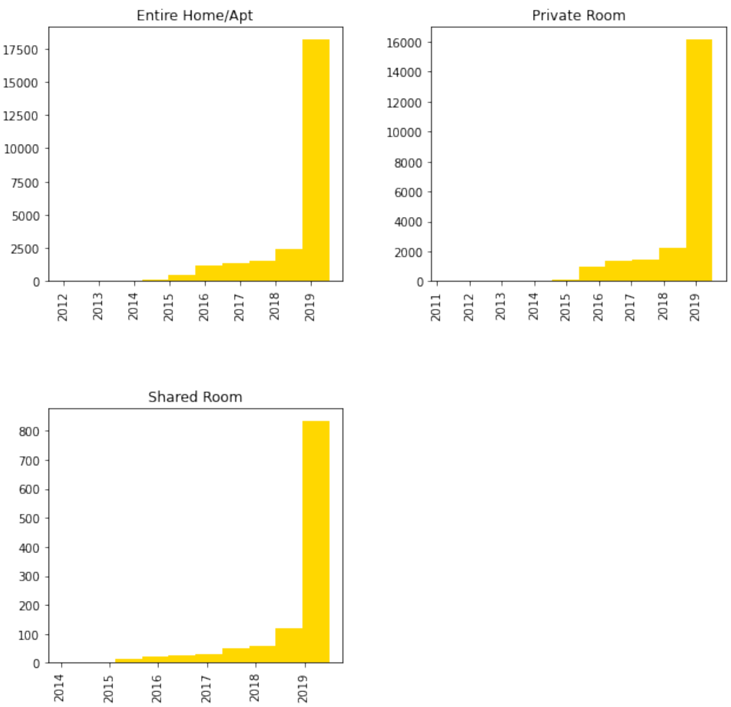

Figure 10. Room Type Distribution Based on Neighbourhoods vs Total Price

to have a similar growth with the number of last reviews they have received.

However we can notice that after 2018 and entering into 2019 the reviews have peaked. And based on our previous graphs Fig. 4, Fig. 6 and Fig. 9 we can see that Entire Home/Apt and Private Room have a lot more reviews compared to the Shared Room.

From the graph we can even though less there is some correlation between the last review received to the room type.[Fig. 10]

VI. Conclusion of Data Pre-processing and EDA

Finally to conclude our findings and information gathered so far we can state in two main points that can benefit customers and hosts:

- First that Manhattan is the most sought out area to find an Airbnb. Also one of the good neighbourhoods for new hosts to list their houses or apartments.
- Second is most customers opt for the Entire Home/Apt as their choice of room type. This shows that hosts interested in putting more listings would benefit if it is an Entire House.

Phase 2

VII. Algorithms/Visualizations

A. K-Means Clustering

K-Means is an unsupervised machine learning algorithm. It is used to cluster data in groups based on similarity. We chose

K Means clustering for the Airbnb listings dataset because it is a straightforward method for grouping listings. We have used the K Means Clustering Algorithm to form the clusters of pricing range for the listings. In order to do so, we have grouped the listings based on their latitudes and longitudes. We use the price, latitude and longitude as the features of the model. Following this the data is scaled as K Means is sensitive to magnitude. Because of this scaling, we get to see that the data now contributes equally to the clusters. We assigned three clusters that we have finalized to depict affordable, mid-range and luxury. To the data frame we add a column with the label of the cluster id. We also map the cluster names and id to print the count of listings in each cluster.
$Luxury$: 22388
$Affordable$: 13926
$Mid-Range$: 12581

To visualise this, we use the scatter plot, along the latitude on y axis and longitude on the x axis. [Fig. 11]

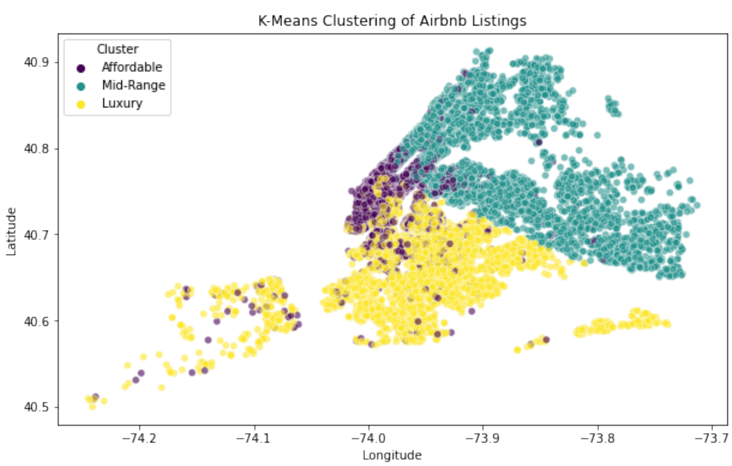

Figure 11. K-Means Clustering

B. Decision Tree

Next choose the Decision Tree model for this problem because it is a powerful model for classification tasks. In this case, we want to predict the room type of an Airbnb listing based on the features that we have: price, number of reviews and availability. This will provide a clear representation of the decision-making process based on the value of a feature, which makes it easy to explain how the model reaches its predictions. Each of the decisions is presented in the node. I limited the depth of the tree to 5 to avoid over fitting and defined the split and leaf. to prevent the tree from creating small and unreliable nodes. We also calculate the accuracy of the model. Then with the classification report, we can look at the precision, recall and f1 score along with the report. This will provide us with a better understanding of how the model performs across different room types. [Fig. 12]

```
Accuracy: 0.81
                 precision    recall  f1-score   support

Entire Home/Apt       0.81      0.88      0.84      5029
   Private Room       0.81      0.78      0.79      4509
    Shared Room       0.90      0.08      0.15       241

       accuracy                           0.81      9779
      macro avg       0.84      0.58      0.59      9779
   weighted avg       0.81      0.81      0.80      9779
```

Figure 12. Decision Tree Metrics

To visualize this, we will plot the decision tree at each step as well.[Fig. 13]

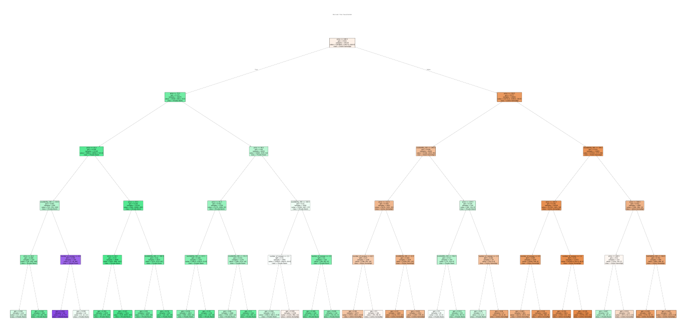

Figure 13. Decision Tree

C. Support Vector Machines (SVM)

The Support Vector Machine (SVM) algorithm is an excellent choice in regards to classification for features including but not limited to room types. It performs particularly well in predicting Airbnb room types based on features such as Latitude, Longitude, and Reviews Per Month by utilizing kernel functions like the radial basis function (RBF) to find the optimal hyperplane. The RBF kernel, `StandardScaler`, and default hyperparameters were used for training and tuning the model. The `accuracy_score` was employed to measure the overall performance of the SVM. A confusion matrix was also plotted to emphasize the model's power in distinguishing different types of rooms. A well-performing SVM should yield a balanced confusion matrix, indicating good classification capability. The classification report presents metrics such as precision, recall, and the F1-score, which can be used to gauge the correctness of the model in classifying the room types.[Fig. 14]

D. XGboost

XGBoost is an ensemble learning technique where a set of relatively weak learners combine to form a robust classifier. It efficiently handles large datasets and complex relationships among features, making it a good fit for predicting room types based on features such as neighborhood group, host listings count, and availability. The model was trained and tuned using `OrdinalEncoder`, label encoding, default hyperparameters, SVM Accuracy: 0.52

	precision	recall	f1-score	support
Entire Home/Apt	0.67	0.58	0.62	5082
Private Room	0.59	0.45	0.51	4465
Shared Room	0.05	0.41	0.09	232
accuracy			0.52	9779
macro avg	0.43	0.48	0.41	9779
weighted avg	0.62	0.52	0.56	9779

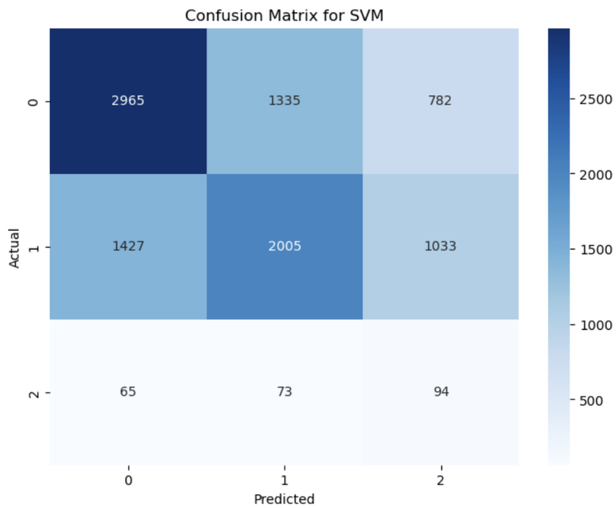

Figure 14. SVM

XGBoost Classifier Accuracy: 0.64

	precision	recall	f1-score	support
0	0.66	0.68	0.67	5082
1	0.61	0.61	0.61	4465
2	0.62	0.11	0.19	232
accuracy			0.64	9779
macro avg	0.63	0.47	0.49	9779
weighted avg	0.64	0.64	0.63	9779

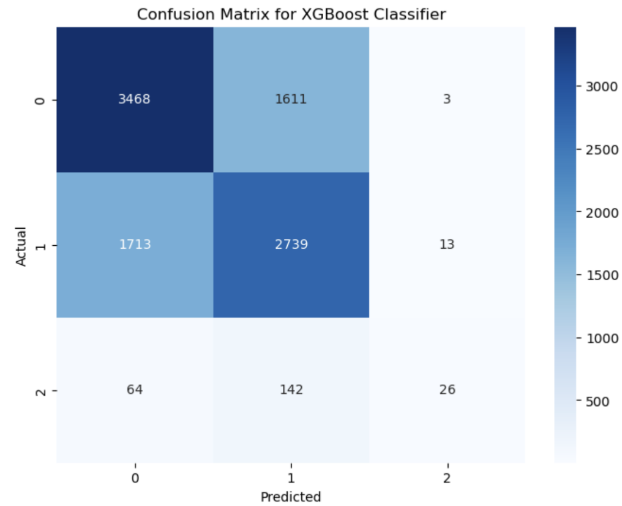

Figure 15. XGBoost

and a stratified train-test split. The accuracy score here refers to how precisely XGBoost can predict the room type. In this case, the confusion matrix provides insights into potential overfitting or underfitting issues. The classification report gives further details on precision, recall, and F1-score performance.XGBoost's feature importance allows one to determine which factors play the most crucial role in predicting the room type. This insight enables hosts to optimize various factors to attract more guests, while guests and travelers can gain an understanding of which types of rooms are most available in each neighborhood.[Fig. 15]

E. Naive Bayes

Naive Bayes is a probabilistic classifications algorithm. It helps with classification tasks. It helps predict by calculating the probabilities of each given features. We decided to choose this model here by focusing on the three features $price$, $calculated\_host\_listings\_count$ and $neighbourhood\_group$. Using the price we decided to split it into two separate categories: Low and High, using the median as the threshold. Then the type of naive Bayes we chose is Gaussian Naive Bayes(Reference 1), since our target feature is price with continuous values. Once the model is trained we are able to see that $calculated\_host\_listings\_count$ influences the likelihood of the listing being classified as Low or High. To further understand how the features affect the models performance we have calculated some of the metrics like accuracy, precision and recall. Below metrics shows even though the accuracy is not as high its still above 50 and our precision and recall values are better. This shows the model is effective for these features used and they do have do have a role in creating or affecting the trends. [Fig. 16] To further

Naive Bayes Accuracy: 0.53

	precision	recall	f1-score	support
0	0.52	0.98	0.68	4941
1	0.77	0.07	0.13	4838
accuracy			0.53	9779
macro avg	0.64	0.52	0.40	9779
weighted avg	0.64	0.53	0.41	9779

Figure 16. Naive Bayes Metrics

visualize what the price categories look like for in terms of the number of listings here is a graph. We can see clearly that there is more spread for listings in the high category and also taking into considering it does matter about the neighbourhood here. We also can understand that though there is approximately equal count of listings however with a few more in the high category.[Fig. 17]

F. Logistic Regression

Since we can tell that classification is a good model to use with our dataset we decided to see how our other features behave. In the next model we have used Logistic Regression. Now Logistic Regression is most often used for binary classification tasks. Its shows how a given features belong to a particular class. This will also help understand the relationship between our features. Also some of the features of our dataset can be

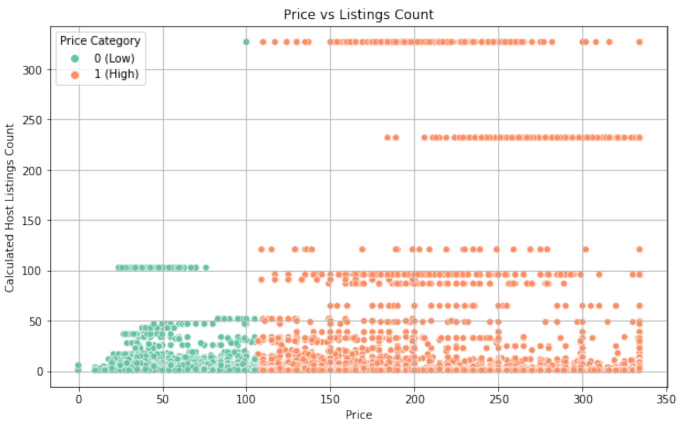

Figure 17. Naive Bayes Model Visualization

clearly linearly separated. For this model hence we decided to select the features *neighbourhood_group*, *price* and the *room_type* as a binary category where `Entire Home/Apt` is one category and `Private Room` and `Shared Room` as another category. From this model we hope to understand what the likelihood of the room type chosen by users/travellers will be `Entire Home/Apt`. Based on the training and prediction of values with our model we have noted some metrics to see how our features affect the classification.[Fig. 18]

```
Losgistic Regression Accuracy: 0.81
              precision    recall  f1-score   support

           0       0.78      0.86      0.82      4750
           1       0.85      0.77      0.81      5029

    accuracy                           0.81      9779
   macro avg       0.81      0.81      0.81      9779
weighted avg       0.82      0.81      0.81      9779
```

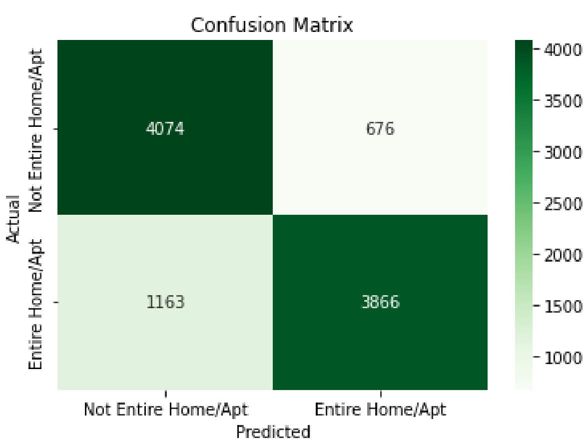

Figure 18. Logistic Regression Metrics

From the metrics we can see how we have a good accuracy score. Along with that our precision and recall are. Along with that the confusion matrix shows how the `Entire Home/Apt` has a high value, but we also notice how `Not Entire Home/Apt` has a higher value compared, however, this is since its combined category of two room types i.e. `Private Room` and `Shared Room`. Another point to be noted is the metrics is based on *neighbourhood_group* as well and we can see how still `Not Entire Home/Apt` is still preferred and this concurs with our initial graphs in EDA.

This is a more visual understanding of our model below are two graphs that show how `Not Entire Home/Apt` with the price range of 110-240 is a much more chosen option as in boxplot.[Fig. 19]

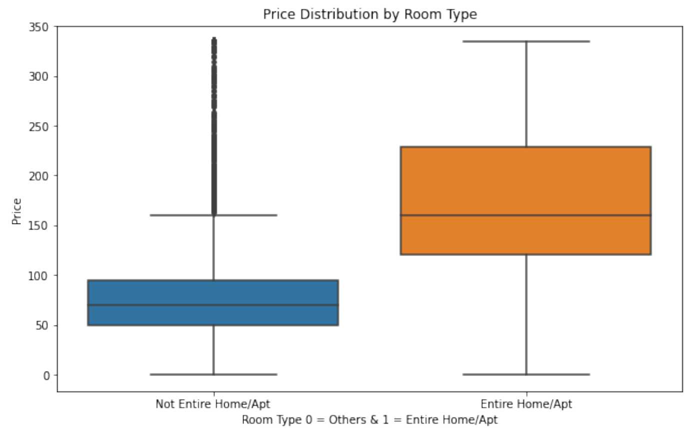

Figure 19. Logistic Regression Model Visualizations

Another graph that shows the distribution of the predicated values to show the models confidence and the level of predictive capability. we can the significant peak near 1.0 suggesting that model is confident in classifying many as`Entire Home/Apt`. [Fig. 20]

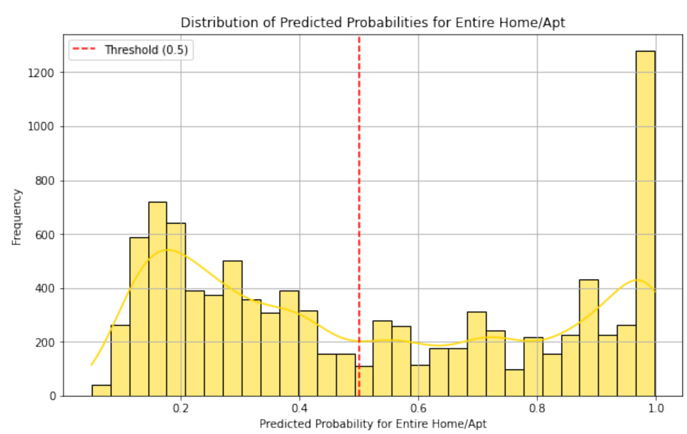

Figure 20. Logistic Regression Model Visualizations

All these show how even based on model used we can see and suggest to travellers/user that `Entire Home/Apt` is a great option and most sought after choice.

REFERENCES

[1] Dgomonov, New York City Airbnb Open Data, Kaggle. 2019, August 12. Available at: https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data

[2] XGBoost documentation. XGBoost Documentation - xgboost 2.1.1 documentation. (n.d.). Available at: https://xgboost.readthedocs.io/en/latest/

[3] Parth Shukla, Analytics Vidya. Naive Bayes Algorithms: A Complete Guide for Beginners. 2024, March 21. Available at: https://www.analyticsvidhya.com/blog/2023/01/naive-bayes-algorithms-a-complete-guide-for-beginners/#h-what-is-naive-bayes-algorithm

[4] Ajitesh Kumar, Analytics Yogi, Recommender Systems in Machine Learning: Examples. 2024, September 16. Available at: https://vitalflux.com/recommender-systems-in-machine-learning-examples/

[5] Scikit Learn. Naive Bayes. (n.d.). Available at: https://scikit-learn.org/stable/modules/naive_bayes.html

[6] Scikit Learn. KMeans. (n.d.). Available at: https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html

[7] GeeksforGeeks. K means Clustering – Introduction. 2024, August 29. Available at: https://www.geeksforgeeks.org/k-means-clustering-introduction/

[8] Wikipedia. Decision tree learning. (n.d.). Available at: https://en.wikipedia.org/wiki/Decision_tree_learning

[9] Scikit Learn. Decision Trees. (n.d.). Available at: https://scikit-learn.org/1.5/modules/tree.html

[10] GeeksforGeeks. Decision Tree in Machine Learning. 2024, March 15. Available at: https://www.geeksforgeeks.org/decision-tree-introduction-example/

Peer Evaluation Form for Final Group Work

CSE 487/587B

- Please write the names of your group members.

Group member 1 : Siju Chacko

Group member 2 : Mythri Shivakumar

Group member 3 : Rebecca Abraham

- Rate each groupmate on a scale of 5 on the following points, with 5 being HIGHEST and 1 being LOWEST.

Evaluation Criteria	Group member 1	Group member 2	Group member 3
How effectively did your group mate work with you?	5	5	5
Contribution in writing the report	5	5	5
Demonstrates a cooperative and supportive attitude.	5	5	5
Contributes significantly to the success of the project .	5	5	5
TOTAL	20	20	20

Also please state the overall contribution of your teammate in percentage below, with total of all the three members accounting for 100% (33.33+33.33+33.33 ~ 100%) :
Group member 1 : 33.33
Group member 2 : 33.33
Group member 3 : 33.33