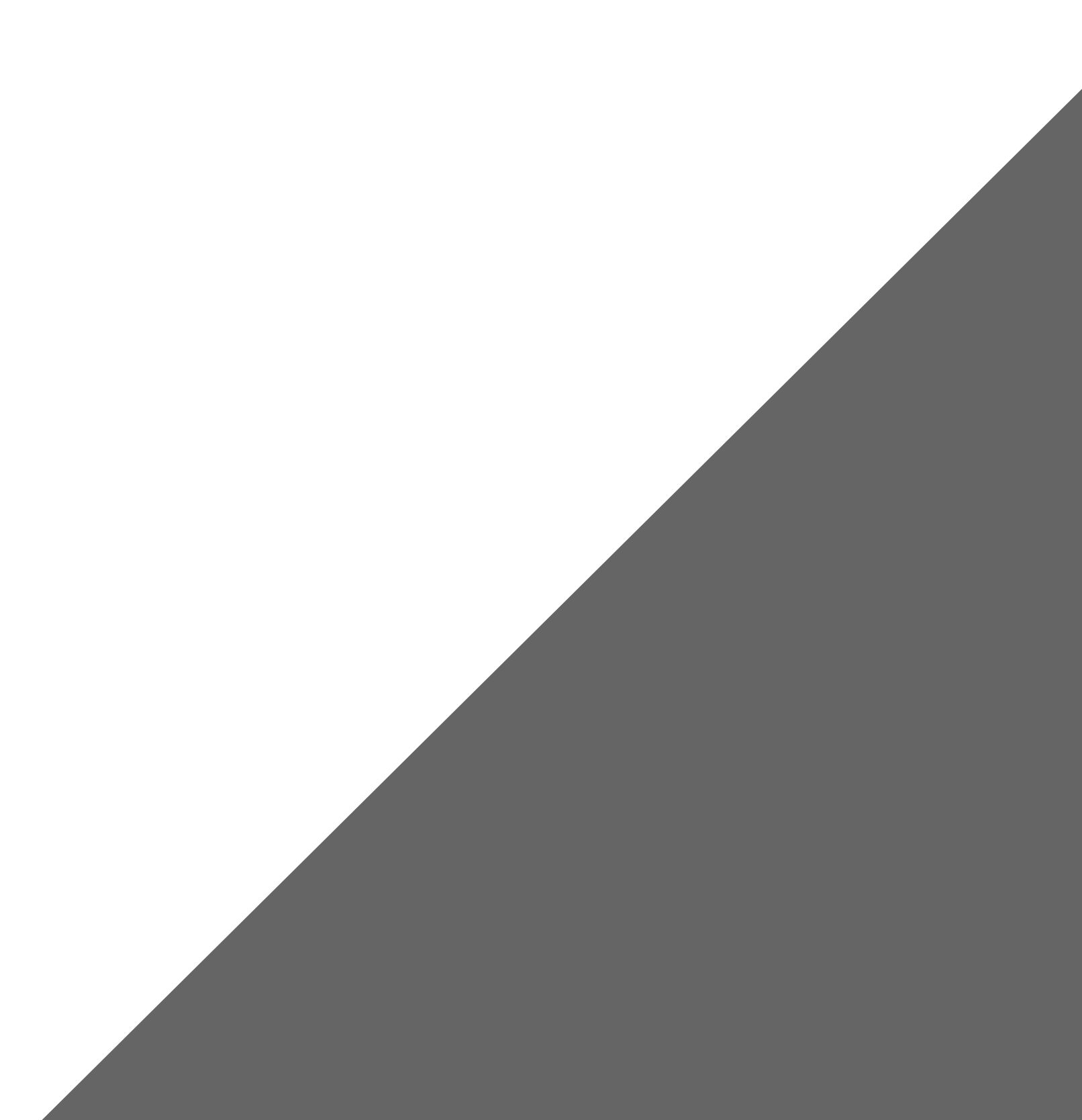
Airbnb

Data Analysis



**Introduction:**

In this project we're going to study Airbnb business in New York City of different hotels. We'll explore the data and fetch insights and recommend business decisions based on those insights. We'll computer this for different hotels and compare if one is doing good what’s the reason behind it or if other is doing bad what’s the reason behind that as well. We'll try to figure out the patterns between different features and how each feature is impacting the target variable (price). From this analysis clients are going to benefit from. If we provide them all the information about areas of hotels, type of services they provide and prices of multiple hotels in that area then client will be able to easily and confidently select the required hotel based on its services and client can get better environment according to his or her needs.

**Business problem:**

1. **Customer can understand Price Distribution And Traffic in each Area to make a decision.**

* Which areas have highest price and what is the traffic ratio in that area based on number of reviews?
* Is there is correlation between traffic and price distribution?
* Which option customer should avail if he/she wants to get a cheaper hotel but better quality?
* Which option customer should avail if quality is a priority as compared to price?

1. **Customer can Understand Relation Between Service Type Provided compared to Price in Each Area**

* Is Service type impacting on price?
* Areas those provide quality services, how they vary in terms of price for each room type?
* Which service type is cheap with better quality and also available in particular area?
* Best locations of each area with quality services?

1. **Tips For Host For Defining Listing (Name) for Its services.**

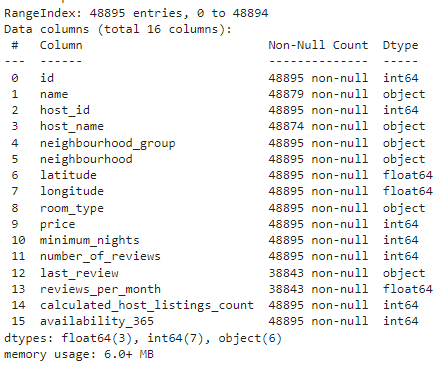
* Most Searched words or terms for airbnb?
* What is the reason behind it?

1. **Tips for Host For defining Service Type based on Customer Search.**
2. **What are the features which are impacting the price?**

**Data Preprocessing:**

(48895, 16)

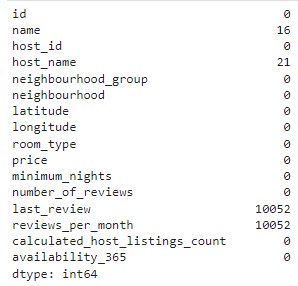
* The above output is showing tuple
* The first value is showing the number of records (rows) in the data set and second is showing the number of columns in the dataset
* So we have total 488985 numbers of rows and 16 numbers of columns (features) in our dataset.



The above output is showing some basic information about data.

* Name of the columns and not null values of the column and data type of the columns
* And also summary of it. how much float integer and string columns we have in our dataset.
* What’s the memory usage of dataset? How much memory, data is occupying.

**Check if there any null values in the dataset.**



After loading the dataset in and from the head of AB\_2019\_NYC dataset we can see a number of things. These 16 columns provide a very rich amount of information for deep data exploration we can do on this dataset. We do already see some missing values, which will require cleaning and handling of Nan values. Later, we may need to continue with mapping certain values to ones and zeros for predictive analytics.

There are null values in our dataset:

* last-review and review per month columns have 10k plus null values.
* hotel name and host name have few null values

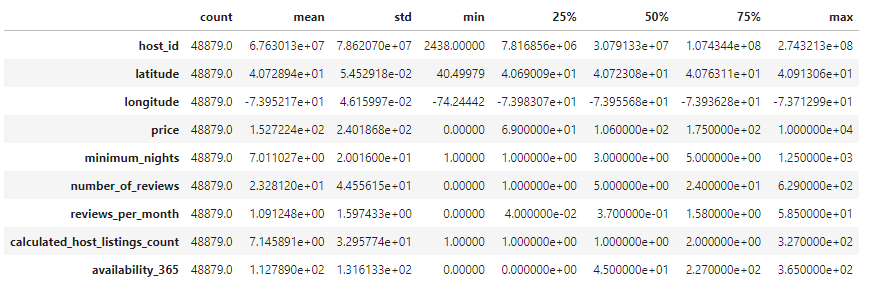
**Drop unnecessary columns:**

In our case, missing data that is observed does not need too much special treatment. Looking into the nature of our dataset we can state further things: columns "host\_name" are irrelevant and insignificant to our data analysis, columns "last\_review" and "review\_per\_month" need very simple handling.

To elaborate, "last\_review" is date; if there were no reviews for the listing - date simply will not exist.

In our case, this column is irrelevant and insignificant therefore appending those values is not needed. For "review\_per\_month" column we can simply append it with 0.0 for missing values; we can see that in "number\_of\_review" that column will have a 0, therefore following this logic with 0 total reviews there will be 0.0 rates of reviews per month. Therefore, let's proceed with removing columns that are not important and handling of missing data.

* id will not contribute in our analysis, host-name is not necessary because we have host id also.
* last\_review is also not important because we still have a lot of reviews information and it's also having a lot of null values.
* Filled the null values of reviews per month column with 0.

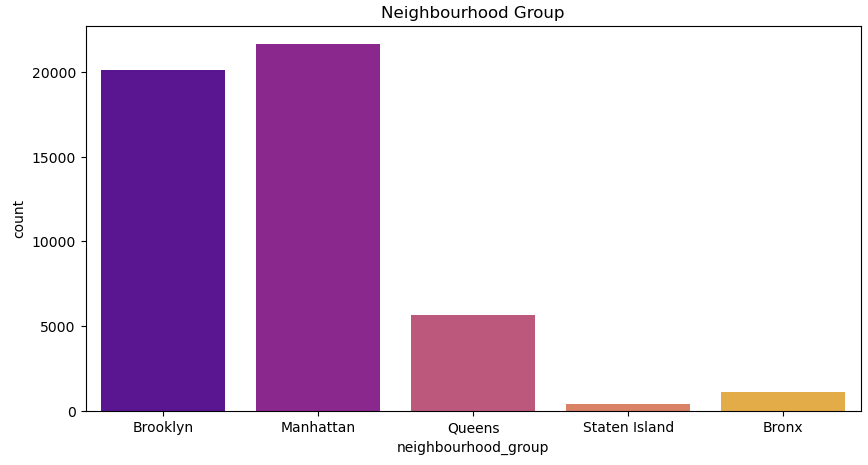


1. **Data Size and Availability**: The data contains 48,879 records for various Airbnb listings.
2. **Host Information (host\_id)**:
   * The average host\_id is approximately 67.63 million, with a standard deviation of 78.62 million.
   * The minimum host\_id is 2,438, and the maximum host\_id is 274,321,300.
3. **Geographical Information (latitude and longitude)**:
   * The average latitude of the Airbnb listings is around 40.73, with a small standard deviation of 0.055.
   * The average longitude is approximately -73.95, with a standard deviation of 0.046.
   * The latitude and longitude range indicates that the data primarily represents listings in a specific geographical area.
4. **Price (price)**:
   * The average price of the Airbnb listings is $152.72, with a relatively high standard deviation of $240.19.
   * The minimum price is $0, indicating that there might be some listings with no charge (e.g., promotions, free stays).
   * The maximum price is $10,000, suggesting that some listings are significantly more expensive than others.
5. **Minimum Nights (minimum\_nights)**:
   * The average minimum nights required for booking is approximately 7, with a high standard deviation of 20.02.
   * The minimum number of nights required is 1, while the maximum is quite high at 1,250.
6. **Reviews Information (number\_of\_reviews and reviews\_per\_month)**:
   * On average, each listing has received 23.28 reviews.
   * The number of reviews per listing ranges from 0 to 629.
   * The average number of reviews per month for each listing is around 1.09, with a high standard deviation of 1.60.
   * Some listings have a significantly higher number of reviews per month, reaching up to 58.50, indicating popular properties.
7. **Calculated Host Listings (calculated\_host\_listings\_count)**:
   * On average, each host has 7.15 listings, but with a relatively high standard deviation of 32.96.
   * Some hosts have as few as 1 listing, while others have up to 327 listings.
8. **Availability (availability\_365)**:
   * On average, listings are available for booking 112.79 days of the year.
   * There is a wide range in availability, with some listings being available throughout the year (365 days), while others have no availability (0 days).

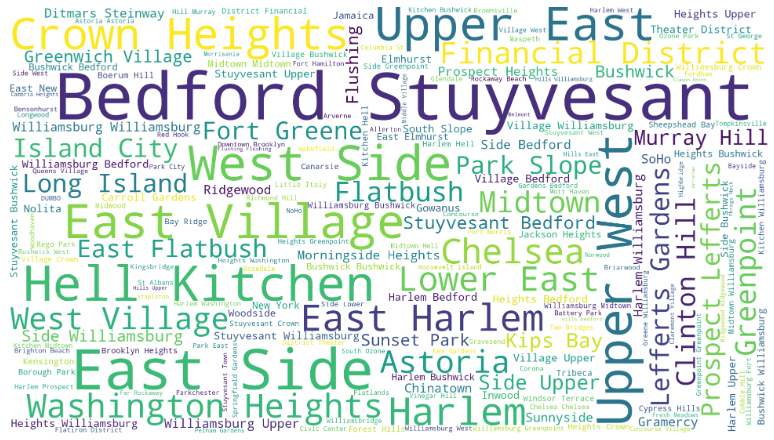
**Exploratory Data Analysis:**

Now that we are ready for an exploration of our data, we can make a rule that we are going to be working from left to right. The reason some may prefer to do this is due to its set approach - some datasets have a big number of attributes, plus this way we will remember to explore each column individually to make sure we learn as much as we can about our dataset.

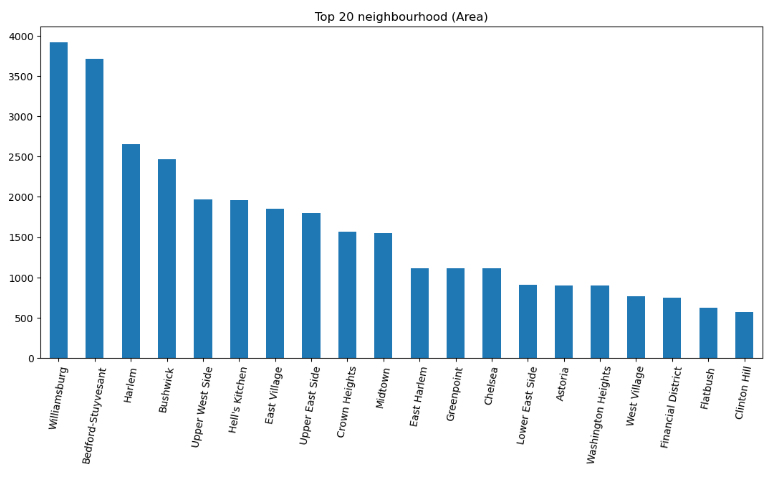
**Univariate Analysis:**

****

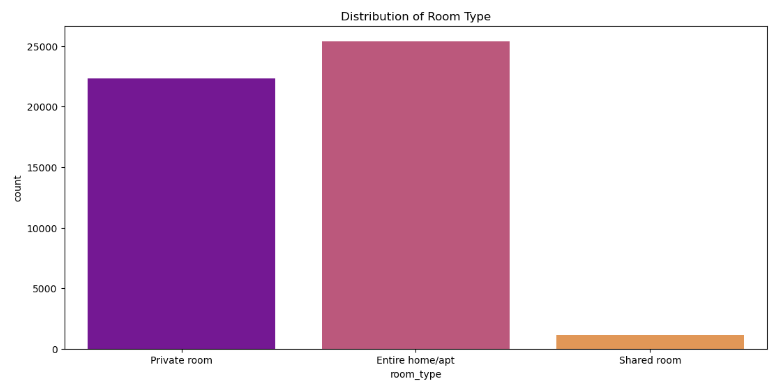
* neighbourhood\_group column's mean location.
* We have total 5 neighborhood groups.
* Manhattan is the most occurring group and on second it's Brooklyn.
* It mean majority of the hotels are at Manhattan and Brooklyn



* neighborhood columns mean specific area at which hotels are.
* we have total 221 unique areas having airbnb hotels
* Let's check the top 20 neighborhood (area)

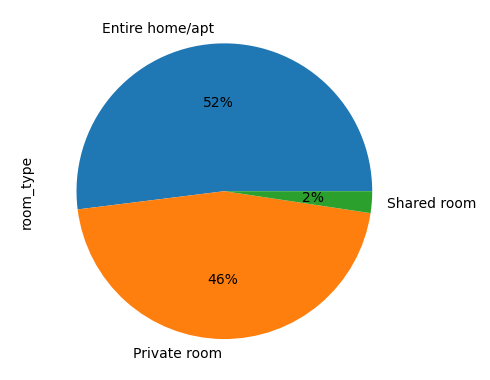


* The above bar graph is showing top 20 areas where most hotels are.
* On x axis there are area's name and on y-axis there are number of hotels in that area.
* The bar is in sorted order based on the number of hotels
* We can see that Williamsburg and Bedford-Stuyvesant are the top areas having more hotels.

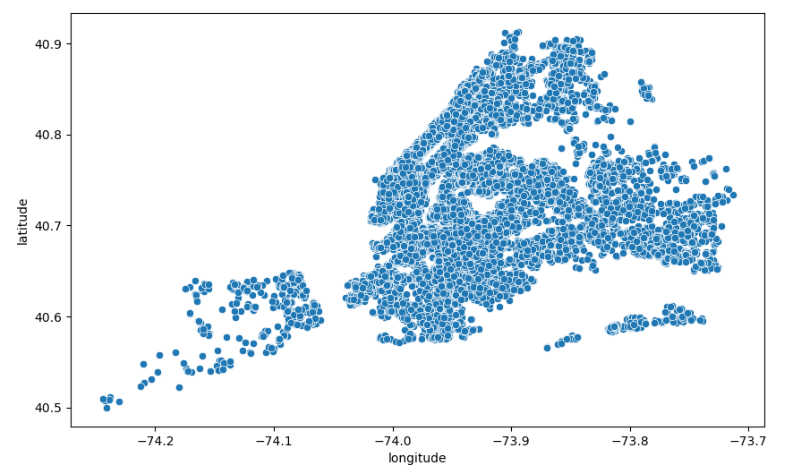


There are three room types:

* Private Room
* Entire home
* Shared room



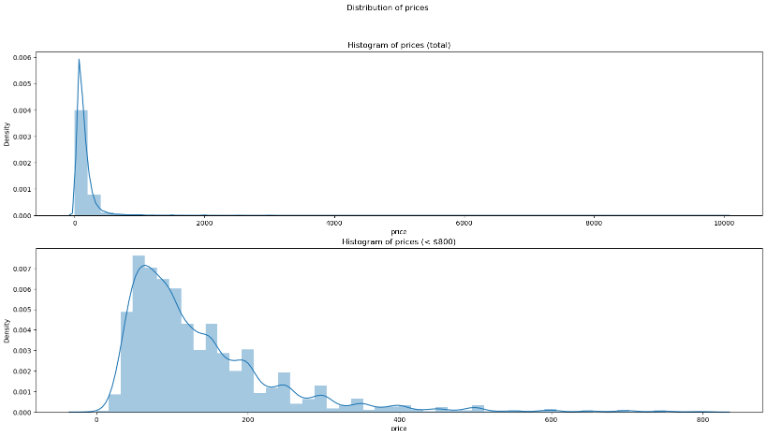
* Majority of the hotels are offering Entire home for families
* On second they are offering Private room
* On third shared room, people usually don't prefer shared room.



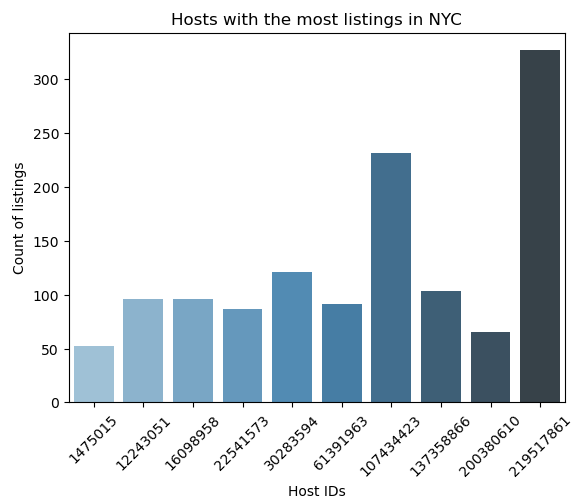
Distribution of hotels over multiple areas.

**What is the global price distribution?**

;

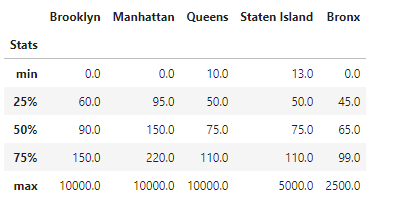


**Host id vs Listing**

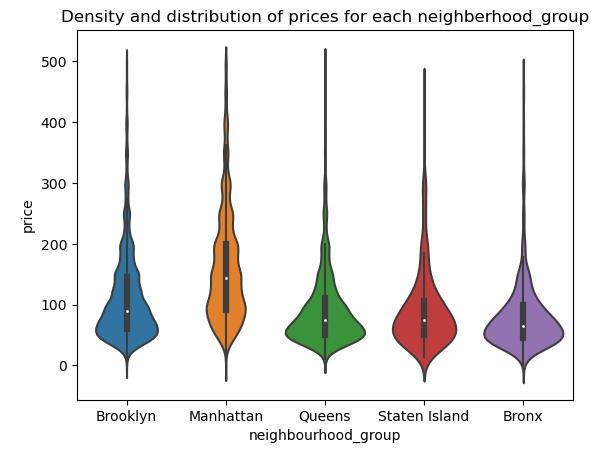


Interesting, we can see that there is a good distribution between top 10 hosts with the most listings. First host has more than 300+ listings.

**Neighborhood groups vs price:**



we can see from our statistical table that we have some extreme values, therefore we need to remove them for the sake of a better visualization.



**Customer can understand Price Distribution And Traffic in each Area to make a decision.**

Great, with a statistical table and a violin plot we can definitely observe a couple of things about distribution of prices for Airbnb in NYC boroughs.

* First, we can state that Manhattan has the highest range of prices for the listings with $150 price as average observation, followed by Brooklyn with $90 per night. Queens and Staten Island appear to have very similar distributions, Bronx is the cheapest of them all.
* This distribution and density of prices were completely expected; for example, as it is no secret that Manhattan is one of the most expensive places in the world to live in, where Bronx on other hand appears to have lower standards of living.

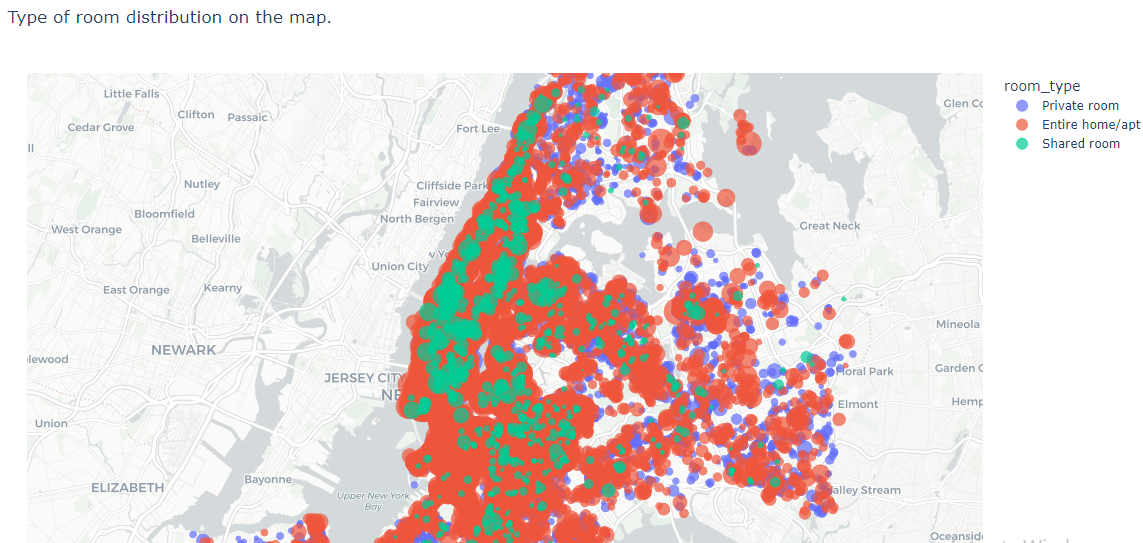


* if we compare Manhattan & Brooklyn
  + Manhattan has the highest price range 150 dollars but Reviews are lesser then Brooklyn
  + Brooklyn has 90 price range But have more Reviews than Manhattan
* If we compare Queen and Staten Island Both have same price but Queen has more review as compared to State Island. It means more customers visited Queen as compare to Staten Island Area

#### Traffic is impacting the price for a fact

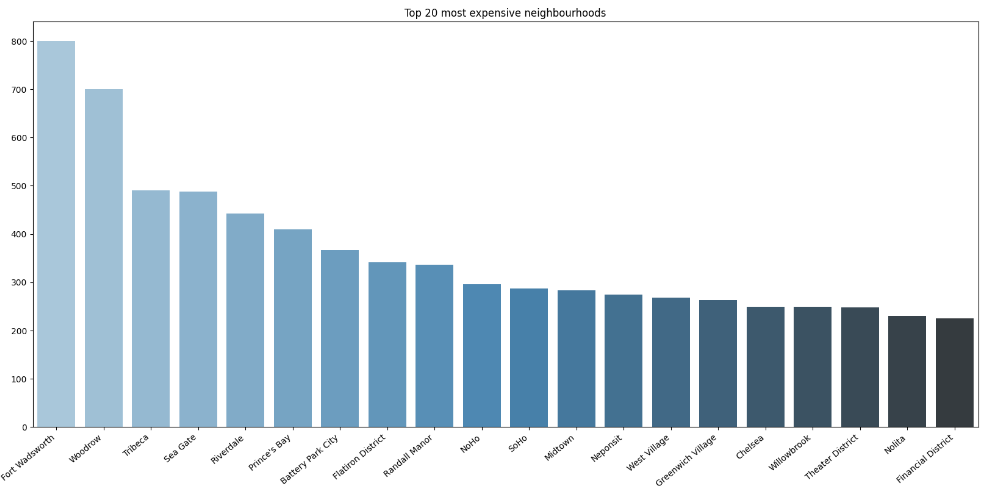
#### Scatter map of most expensive Airbnb rents (>700$)

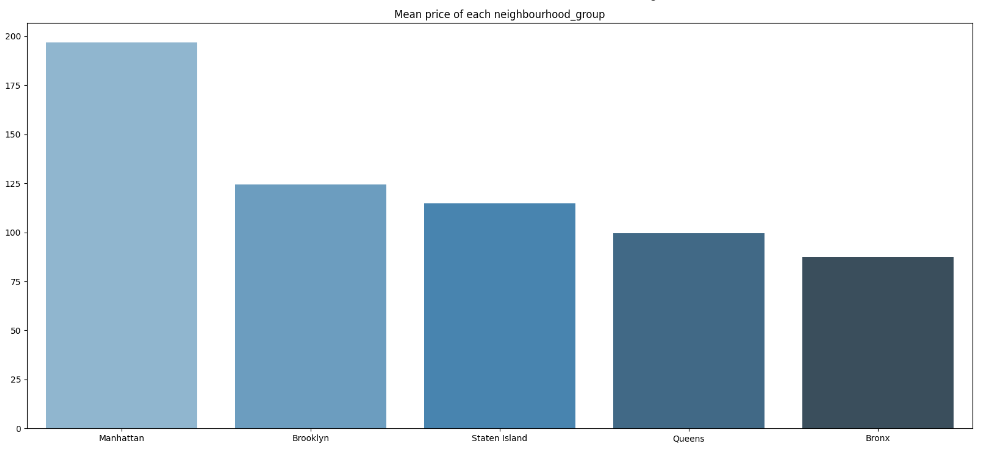




**Price VS Neighbored**

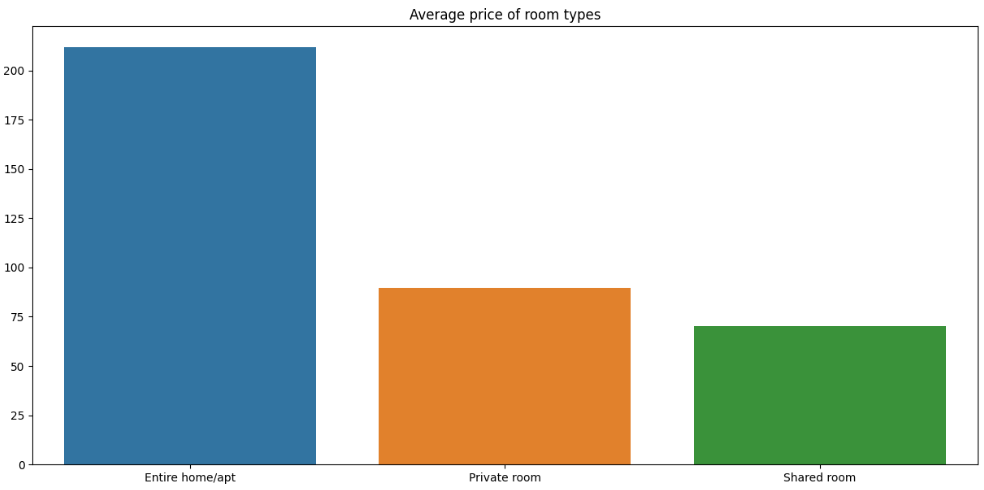
Let's take a look at the spatial distribution of prices (Which are the most expensive zones?).

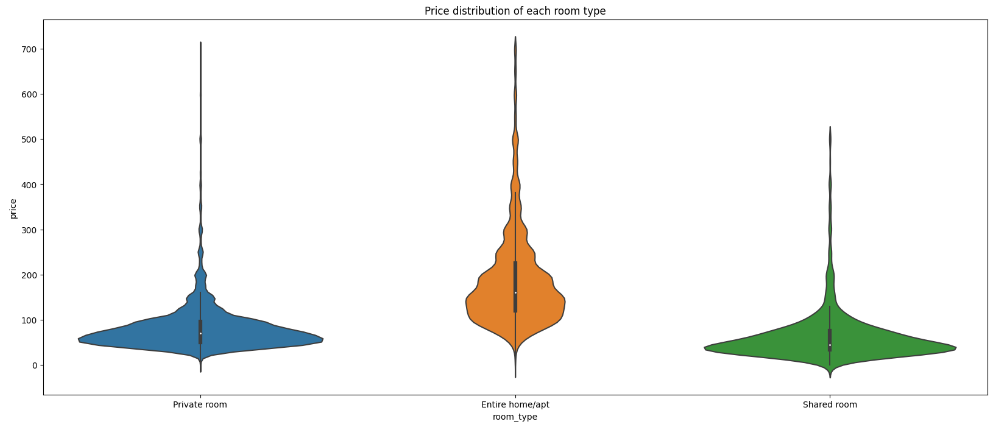




**Price vs Room type**

* From the previous plots we can see that the most expensive rooms tend to be located in the Manhattan zone.
* We also can study the prices are depending on the room type.





* We can see that Entire apartments are more expensive as compared to private rooms and it's making sense. And the average price is $150.
* The private room's average cost is 70 to $90.
* And Shared room's average cost is 40 to 60 $.

**Customer can Understand Relation Between Service Type Provided compared to Price in Each Area in this conclusion**

* The price distributions of each room type in the 5 different NY zones look similar, the most notable difference is on their means.
  + Manhattan is the most Expensive Area and second is Brooklyn Although Brooklyn has more traffic as compared to Manhattan based on number of reviews.
  + But here Again, On Average Manhattan is charging more for each Room Type as compared to Brooklyn
  + Queen has much more traffic based on review as compared to State Island which has very low traffic. It means most of the time customers visited Queen as compare to Staten Island Area. May be that's the reason the average prices of queen is slighter higher then Staten Island.
  + Bronx is cheapest and have more traffic as compared to Staten Island
* The price distributions of the room type's point that, in general, shared and private rooms have similar prices (less deviation). On the other hand, entire apartments have more variability (and are more expensive, obviously).
* We don't have data about the properties, but we can guess that other variables like square ft. or being near a metro station affect the price.

#### if customer wants a cheaper house but have better traffic then Bronx is a good to go as compared to Staten Island.

#### More Reviews can point to quality as well So if a Customer wants Quality but not at the highest price So Brooklyn will be a good option.

#### prices vs number of nights:

#### Percentile 95 of minimum\_nights: 30.0

#### Mean of minimum\_nights: 7.011027230507989

#### Mode of minimum\_nights: 1

#### 21.PNG

#### We can see that are a high number of rooms with minimum\_nights=0 and some of them excessively high (1200). As explained by Dgomonov, this can be due to, at the time the data was gathered, the rooms were not available, or bad data. However, analyzing the distribution, we see that the normal value is around 2 and the highest ones tend to be around 30 nights.

#### 22.PNG

#### Section conclusions

#### We can see several things:

#### Entire apartments are the most expensive ones (obviously)

#### Shared rooms tend to be more in the city centre.

#### The price distributions of each room type in the 5 different NY zones look similar, the most notable difference is on their means.

#### The price distributions of the room type's point that, in general, shared and private rooms have similar prices (less deviation). On the other hand, entire apartments have more variability (and are more expensive, obviously). We don't have data about the properties, but we can guess that other variables like square ft. or being near a metro station affect the price.

#### When looking at the Popularity/Price plot, we can see that rooms with more reviews tend to be cheaper (although the Entire apartment class is more noisy than the others). This can be explained as: the more expensive is an apartment, the less people stays there and therefore, the less reviews. This would be interesting to use in a model to predict prices or the popularity of a room.

#### It's generally cheaper to stay in rooms between 14 and 28 nights.

#### Usually, the minimum required nights to stay in a room is around 2.

#### Multivariate Analysis

#### 23.PNG

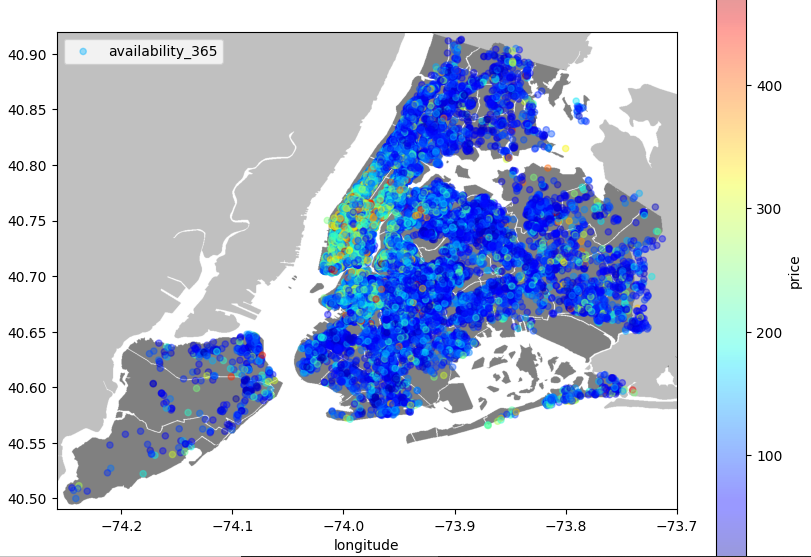
Amazing, but let' breakdown on what we can see from this plot.

* First, we can see that our plot consists of 3 subplots - that is the power of using catplot; with such output, we can easily proceed with comparing distributions among interesting attributes. Y and X axes stay exactly the same for each subplot, Y-axis represents a count of observations and X-axis observations we want to count. However, there are 2 more important elements: column and hue; those 2 differentiate subplots. After we specify the column and determined hue we are able to observe and compare our Y and X axes among specified column as well as color-coded. So, what do we learn from this?
* The observation that is definitely contrasted the most is that 'Shared room' type Airbnb listing is barely available among 10 most listing-populated neighborhoods. Then, we can see that for these 10 neighborhoods only 2 boroughs are represented: Manhattan and Brooklyn; that was somewhat expected as Manhattan and Brooklyn are one of the most traveled destinations, therefore would have the most listing availability.
* We can also observe that Bedford-Stuyvesant and Williamsburg are the most popular for Brooklyn borough, and Harlem for Manhattan.

**Best locations of each area with quality services?**

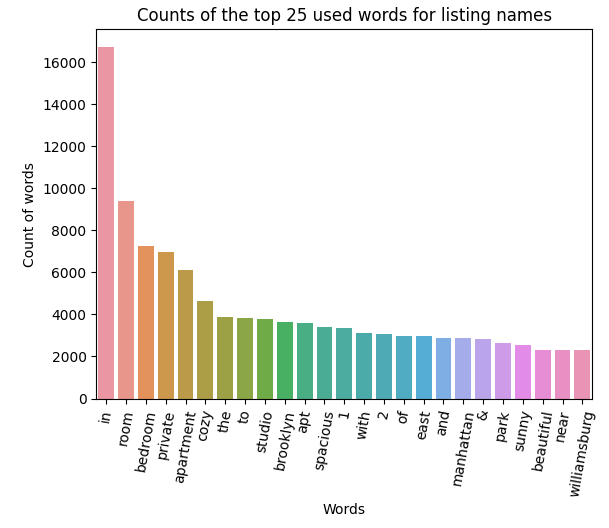
Bedford-Stuyvesant and Williamsburg are the most popular for Brooklyn borough, and Harlem for Manhattan.

**map (latitude longitude) vs availability 365 and price:**

****

Fantastic! After scaling our image the best we can, we observe that we end up with a very immersive heat map. Using latitude and longitude points were able to visualize all NYC listings. Also, we added a color-coded range for each point on the map based on the price of the listing. However, it is important to note that we had to drop some extremely high values as they are treated as outliers for our analysis.

**Hotels/Room Name Description:**

****

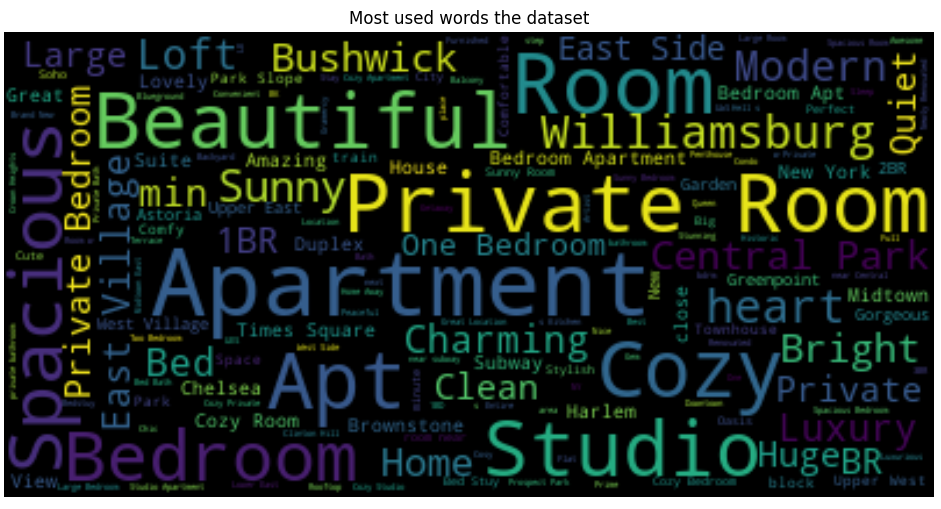
**Tips For Host For Defining Listing (Name) for Its services.**

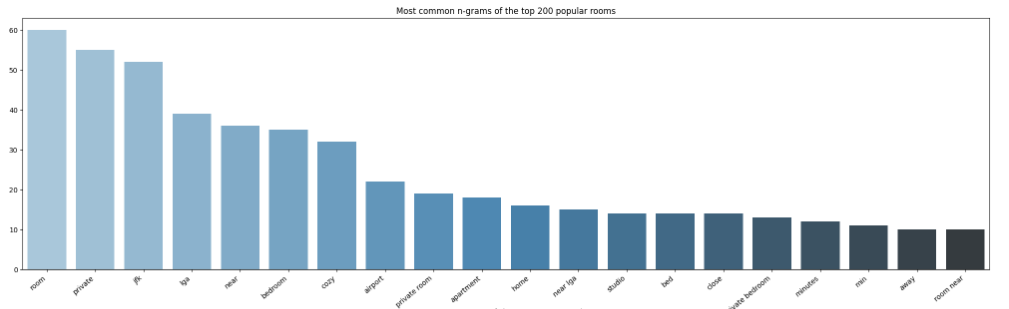
Most Searched words or terms for airbnb?  
What is the reason behind it?

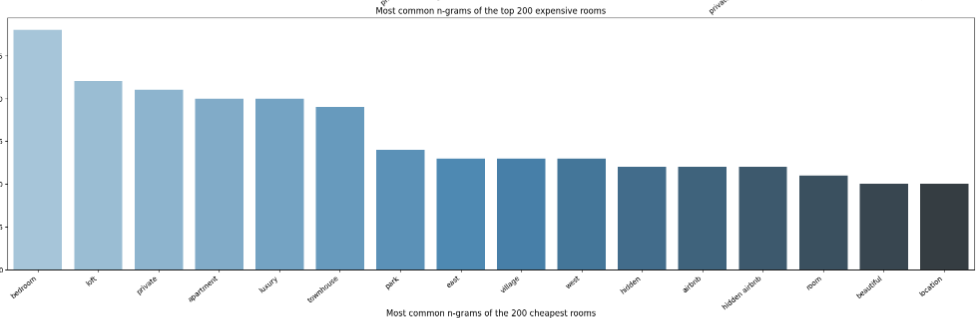
We can observe that finding out and going over top 25 used listings' name words - we are able to see one clear trend.

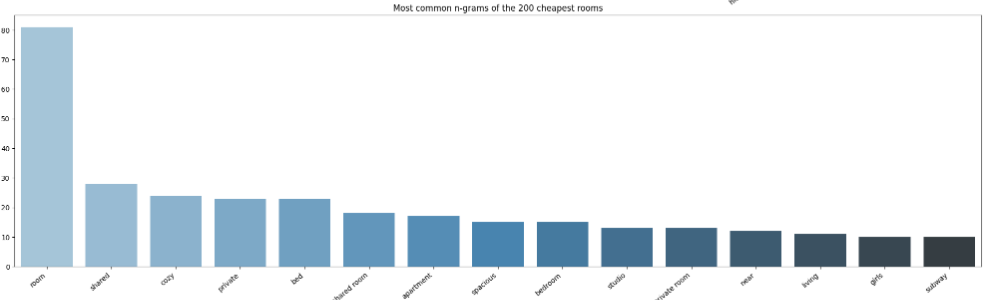
* It shows that hosts are simply describing their listing in a short form with very specific terms for easier search by a potential traveler. Such words are 'room', 'bedroom', 'private', 'apartment', 'studio'.
* This shows that there are no catchphrases or 'popular/trending' terms that are used for names; hosts use very simple terms describing the space and the area where the listing is.
* This technique was somewhat expected as dealing with multilingual customers can be tricky and you definitely want to describe your space in a concise and understood form as much as possible.

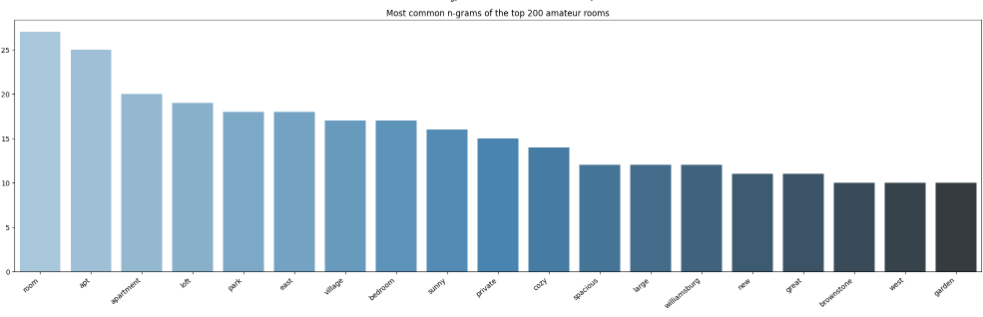
**Comparison Analysis:**











**Tips For Host For defining Service Type based on Customer Search**

We can see differences when analyzing the word distribution of the rooms:

* The words of the most expensive rooms reinforce our previous hypothesis (the zone in which the room is located affect the price) as the most used words include central/central park/townhouse.
* The most popular rooms are the ones which are near to airports or 'minutes from' places. This would way that if we had an accessible room (near transportation), its chances to be popular would be increased.
* The amateur rooms tend to be sold in other ways in comparison to the professional ones. They tend to be posted with concepts as "cozy", "spatious", "great" (comfort) whereas the professional ones are sold in a more objective way (not using valorations).
* The cheapest rooms also sell subjective terms. We can also suppose a certain overlap between amateur and cheap ones (from the distributions above, the amateur ones tend to be cheaper).

**What are the features impacting price?**



* P value 0.000 means the null hypothesis is true and is therefore very statistically significant.
* A low p-value (usually less than 0.05) indicates that there is a significant relationship between the categorical variable and the target variable "price".
* **Area, Host, and room Type** have strong correlation with price, And we have already observed it in our previous analysis.

So the features which are impacting the price **are Area, Host, Room Type**.

**Predictive Analysis:**

**Modeling**

Our purpose is to predict the price based on the given dataset. Since price is a continuous variable we need to do a regression task. Therefore, we will use the Regression models of the traditional models such as KNN, SVM.

We are not after accuracy, because we are doing a regression task, not a classification task. Our main metric will be mean squared error and we will try to minimize it with some methods such as hyper parameter tuning.

Almost in every model we have sections such as

Before Hyper parameter Tuning( We are trying model with default

* Train model with the default parameters
* See the error and r2 score
* Compare with other models and results

After Hyper parameter Tuning

* Determine the parameter grid
* Do a grid search to get best model parameters.
* Compare with the default model and other models.

**Data Preparation:**

* I have dropped all the unnecessary columns like host\_id, name, latitude, longitude, neighborhood, number of reviews and reviews per month.
* I have performed encoding to categorical variables.
* I have performed normalization on the data set to avoid over fitting.
* 80% data will be used for training and 20% will be used for testing.

**Model Training:**

What has been done in this section:

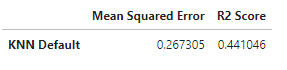
1. Trained in Various Models
2. In Each model, we tried to minimize the mean\_squared\_error with using hyper parameter tuning and cross validation.
3. Showed results in each model.
4. Model comparison, (which model performed best)



**K NEAREST NEIGHBOUR**

**KNN** can be used for ***regression***. But there is a better model which is KNeighborsRegressor which is used for regression tasks. Therefore, we will try *KNeighborsRegressor* for our price prediction goal.

**BEFORE HYPERPARAMATER TUNING:**

****

**HYPERPARAMETER TUNING FOR KNN**

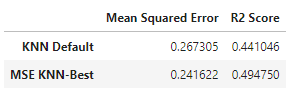
Since we have hyperparamaters such as number of nearest neighbor, or distance metric, we will apply hyperparamater tuning to see if we can decrease the error. Let's test it by:

* Number of Neighbors(1, .... ,10)
* LDistance Function (1: Euclidean, 2: Manhattan)

Cross Validations:

* And since we split the data by 80 training, 20 test data. We will do 100 / 20 = 5 cross validation splits.

We will use GridSearchCV which searches the model with all possible combinations with using cross validation. It calculates all the scores and finds the model which performed best

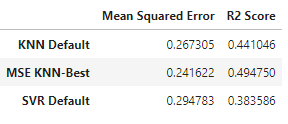


We Can see a improvement in the best model in both of the metrics.

**SUPPORT VECTOR MACHINE**

LinearSVR is support vector machine for regression problems. Therefore we will use first default parameters and we will apply hyper parameter tuning. In most regression cases LinearSVR perform faster and more accurate results.

Source: [SkLearn- LinearSVR](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html)



The numbers are slightly worse than the KNN with best performance. Numbers are not bad, but let's see if we can improve the performance with hyper parameter tuning.

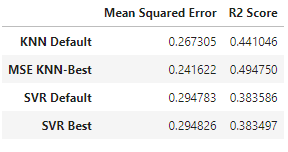
**APPLY HYPERPARAMETER TUNING AND CROSS VALIDATION**

The parameters for the Linear SVR:

* **loss:** Specifies the loss function (Either L1 Loss or l2 loss)
* **C:** Regularization parameter. ('C': [0.1, 1, 10, 100, 1000])
* **dual:** dual or primal optimization problem. (Either True or False)

What we are going to do:

* Use grid search to fit all those parameters
* In each iteration use 5 cross validation points

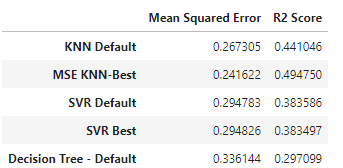


It can be seen that **KNN performed better mostly** (we can see it also from MSE scores). **Hyper parameter tuning for SVR does perform better than the default model.** But there is not much difference. The reason might be that there is no much difference between models in terms of parameters and the difference is not enough to get a huge difference.

**DECISION TREE**

Decision Tree can be used for classification and regression. Here, we use Decision Tree for regression.

**BEFORE HYPERPARAMETER TUNING**



Decision Tree is the worst so far

**APPLY HYPERPARAMETER TUNING AND CROSS VALIDATION**

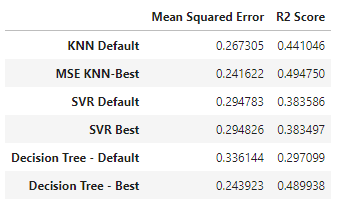
Parameters that is used for Decision Tree Regressor:

1. **max\_depth**: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.
2. **min\_samples\_leaf**:The minimum number of samples required to be at a leaf node.
3. **min\_samples\_split**:The minimum number of samples required to split an internal node.

[Reference to explanation of parameters](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html)

What we are going to do:

1. Use grid search to fit all those parameters
2. In each iteration use 5 cross validation points

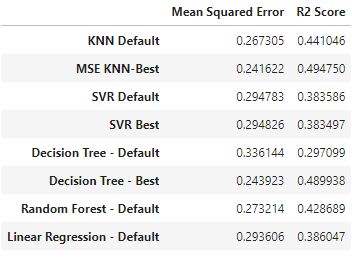


**RANDOM FOREST**

Random Forest is supervised machine learning algorithms which can be used for regression.

Hyper parameter tuning takes too much time, approximately at least 40 minutes, to finish for random forest model. Therefore, we didn't apply Hyper parameter tuning.

Actually, default random forest gives the best solution among the models that we tried so far. However, it has some performance issues compared to other models.



**RESULTS & DISCUSSION**

Random Forest model gives us the best score in R2 as well as MSE. However, as we described in Random Forest section, running time of the Random Forest model is slightly more than other models. Neural Network performed similar to Random Forest and two of them performed best among the other models.

If we do not have time restriction, KNN or Neural Network can be used. Both give us noticeable performance and accuracy. If we had more data, neural network would perform better than the current performance

**CONCLUSION**

1. We trained the data in several models with hyper parameter tuning to see best parameters for each model.
2. In most of the models, they performed similar but neural network performed best.
3. The errors is slightly better than the Kaggle Results.

**WHAT CAN BE DONE TO INCREASE ACCURACY**

* There could be more data.
* Models can be trained in lots of parameters to see which one is best. Since it took so much time, we did not train in every possible of combination.

**Summary:**

This Airbnb ('AB\_NYC\_2019') dataset for the 2019 year appeared to be a very rich dataset with a variety of columns that allowed us to do deep data exploration on each significant column presented. We have tested several hypothesis against the data, and the conclusions obtained are listed at the end of each section, however, here are the most interesting ones (IMO).

* First, we have found hosts that take good advantage of the Airbnb platform and provide the most listings; we found that our top host has 327 listings.
* After that, we proceeded with analyzing boroughs and neighborhood listing densities and what areas were more popular than another. Next, we put good use of our latitude and longitude columns and used to create a geographical heat map color-coded by the price of listings.
* Further, we came back to the first column with name strings and had to do a bit more coding to parse each title and analyze existing trends on how listings are named as well as what was the count for the most used words by hosts. Lastly, we found the most reviewed listings and analyzed some additional attributes.
* For our data exploration purposes, it also would be nice to have couple additional features, such as positive and negative numeric (0-5 stars) reviews or 0-5 star average review for each listing; addition of these features would help to determine the best-reviewed hosts for NYC along with 'number\_of\_review' column that is provided.
* Overall, we discovered a very good number of interesting relationships between features and explained each step of the process.
* NYC shared rooms tend to be grouped in the city centre, maybe because there are thought for travelers who want to visit the most iconic city places.
* Relating the price/popularity variables suggest that people who travel and use Airbnb tend to prefer the posts which are cheaper
* There are two types of user posting rooms: Professionals, which are outliers, each one holding a high number of rooms; and Amateurs, who usually have only a few. Although amateurs can be making money as a business to, their volume is clearly inferior to the professional ones.
* The professional posts are located in the city centre.
* The way rooms are announced is different between professionals and amateurs. The first use more objective terms to describe the room whereas the second use more subjective.
* Having a room "near to" things affect to popularity (maybe it's a good idea to include this words in the title of the room).

This data analytics is very much mimicked on a higher level on Airbnb Data/Machine Learning team for better business decisions, control over the platform, marketing initiatives, implementation of new features and much more.

**REFERENCES**

1. <https://www.airbnb.com/help/article/2503/what-is-airbnb-and-how-does-it-work>
2. <https://automating-gis-processes.github.io/site/notebooks/L3/nearest-neighbor-faster.html>
3. <https://data.ny.gov/Transportation/NYC-Transit-Subway-Entrance-And-Exit-Data/i9wp-a4ja/data>
4. <http://go.euromonitor.com/rs/805-KOK-719/images/wpTop100Cities19.pdf?mkt_tok=eyJpIjoiTVRsaU5Ua3lObVUwT1RVdyIsInQiOiJ2RWRQREV6S1wvN0U5N0R3aXI2MjZrRmhsRFpRc1RSNFZBUDljXC9wcFBVWmZcL2thaVRSYm5pOHJERVVxcXpTWVNHbUd1SjhHa0NqQ0g4bGk1a2JreDIzME5UbFFrSFpTblhoczVDcm91OXRSZmh4ZVUyc1dyaVlUeFppWHZNT1U5SiJ9>
5. <https://www.kaggle.com/erikbruin/airbnb-the-amsterdam-story-with-interactive-maps>
6. <https://public.opendatasoft.com/explore/dataset/airbnb-listings/table/?disjunctive.host_verifications&disjunctive.amenities&disjunctive.features&refine.city=New+York>
7. <https://towardsdatascience.com/linear-regression-models-4a3d14b8d368>
8. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html>