Airbnb Dataset:

The dataset belongs to Airbnb New York[1], it has a total summary information of its host listing. To predict the prices of the host listing I have used the linear regression and gradient boosting algorithms. And to classify data on bases on room type I have used decision tree algorithm.

This dataset contains total of 15 columns and 48895 rows.

Columns	Column Description	Data Types
Id	Listing Id	Integer 64
Name	Name of the listing	Object
host_Id	Host Id	Integer 64
Host_name	Name of the host	Object
neighbourhood_group	Location	Object
neighbourhood	Area	Object
Latitude	Latitude coordinates	Float 64
Longitude	Longitude coordinates	Float 64
room_type	Listing space type	Object
price	Price in dollars	Integer 64
Minimum_nights	Amount of nights minimum	Integer 64
Number_of_reviews	Number of reviews	Integer 64
Last_review	Latest review	Object
reviews_per_month	Number of reviews per month	Float 64
Availability_365	Number of days when listing is available for booking	Integer 64

Fig: 1

Setup of target variable:

Here we have two variables, one is the price variable of the listings and second one is room type of the listing. By using the appropriate algorithms both the prediction of prices and classification of room types are done.

Data Preprocessing and Cleaning:

Here the target is the count of number of suicides happening for every 100 thousand population.

First, we check the count of number of nulls in our dataset.

```
name                              16
host_id                            0
host_name                         21
neighbourhood_group                0
neighbourhood                      0
latitude                           0
longitude                          0
room_type                          0
minimum_nights                     0
number_of_reviews                  0
last_review                    10052
reviews_per_month              10052
calculated_host_listings_count     0
availability_365                   0
price                              0
dtype: int64
```

Fig: 4

Here the last_reviw and review_per_month has 10052 null values. So, I have replaced all the null data for last_review with a data and replaced with zero at all null values in review_per_month column. And I have dropped the name and host_column as they are unique variables which aren't useful for prediction and classification.

---

[1] [1] "New York City Airbnb Open Data | Kaggle." https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data (accessed May 04, 2020).

```
host_id                           0
neighbourhood_group               0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
minimum_nights                    0
number_of_reviews                 0
last_review                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
price                             0
dtype: int64
```

Fig: 5

So, the above table is cleared data after removing all the null values.

Insights from the data:

Few insights have been taken out from the dataset by using relevant visualizations from pandas and seaborn.

The top 10 & least 10 NYC neighbourhood

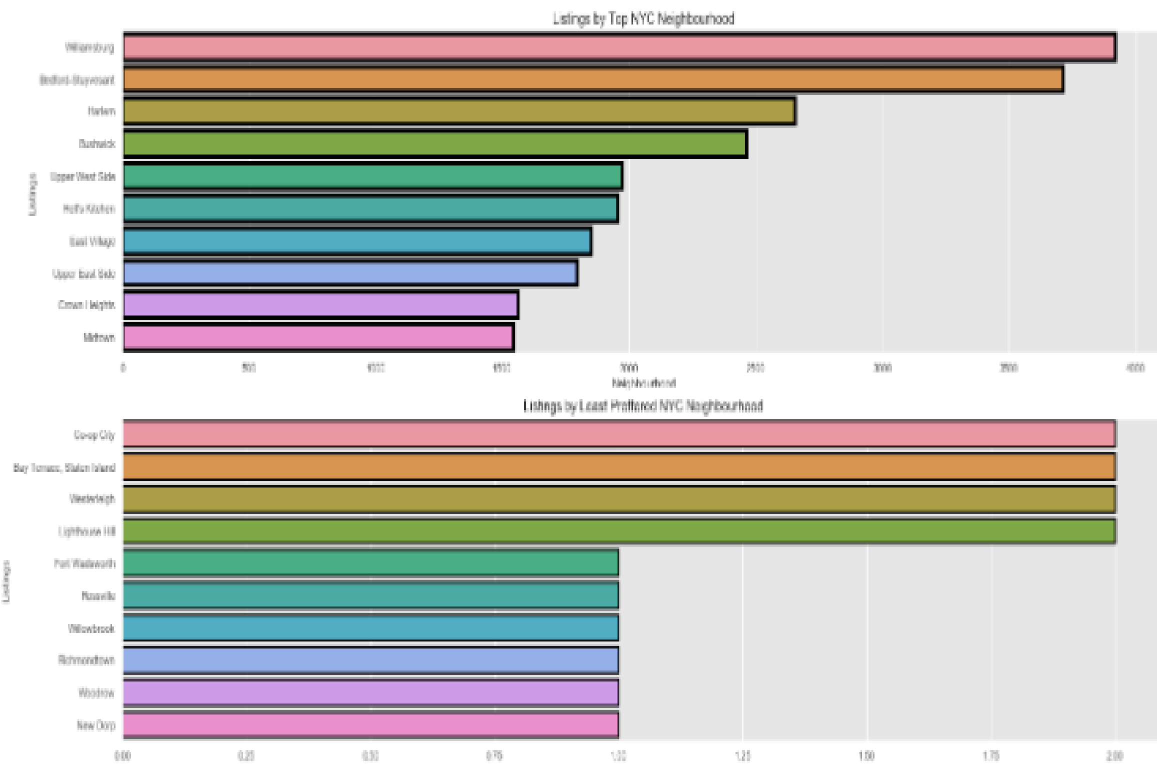


Fig: 10

Heatmap of room availability for 365 days

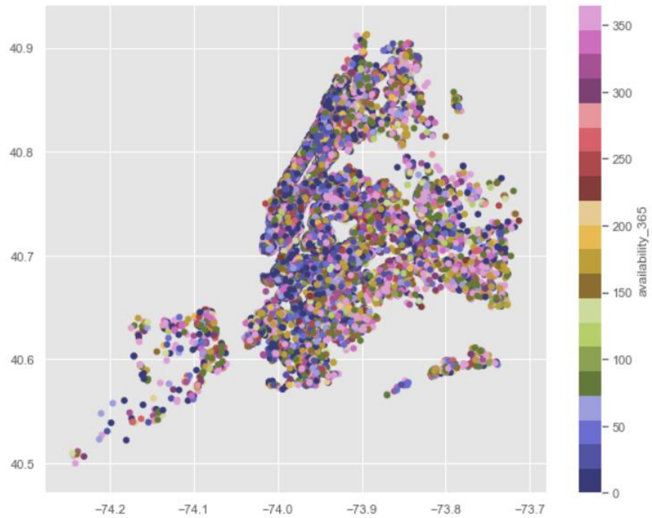


Fig: 11

Plot between the room types and neighbourhood groups.

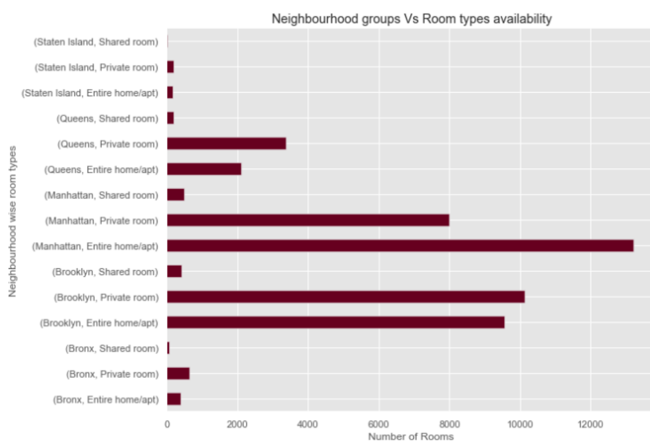


Fig: 12

Here Manhattan & Brooklyn have more number of private room and entire home.

\

All 5 brough's room types within a price range:

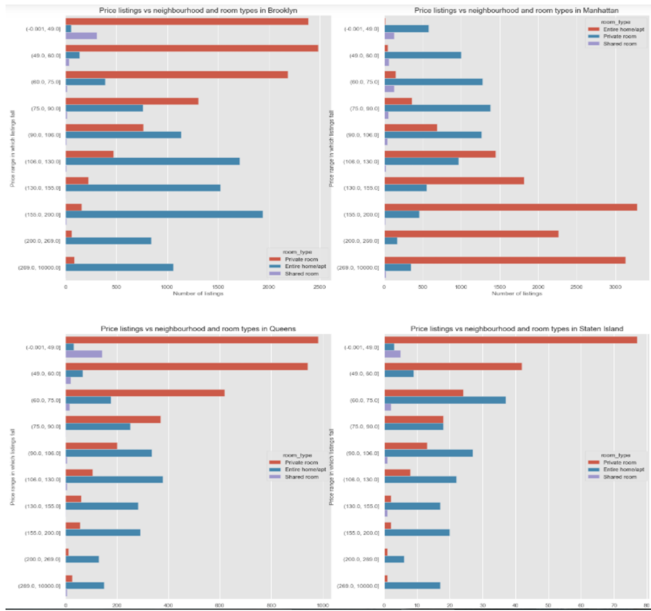
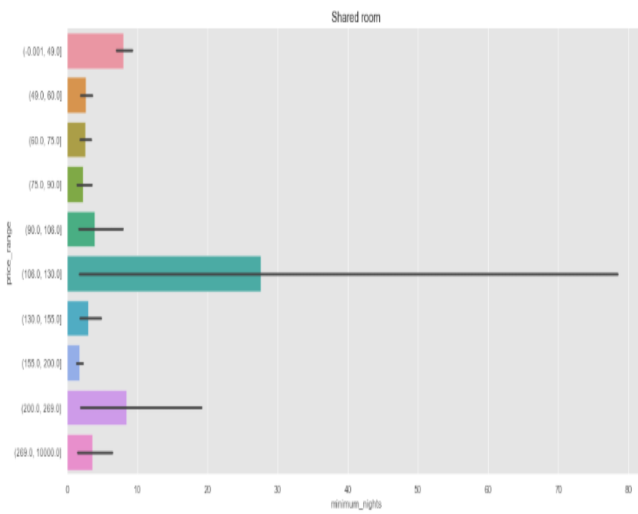


Fig: 13

From above graph, most of the neighbourhood properties comes under range of 500$. Private rooms of Manhattan has an average price of 116.78$ and individual apartment being 249.23$.

Plot of Price_Range and Minimum nights against the room type:



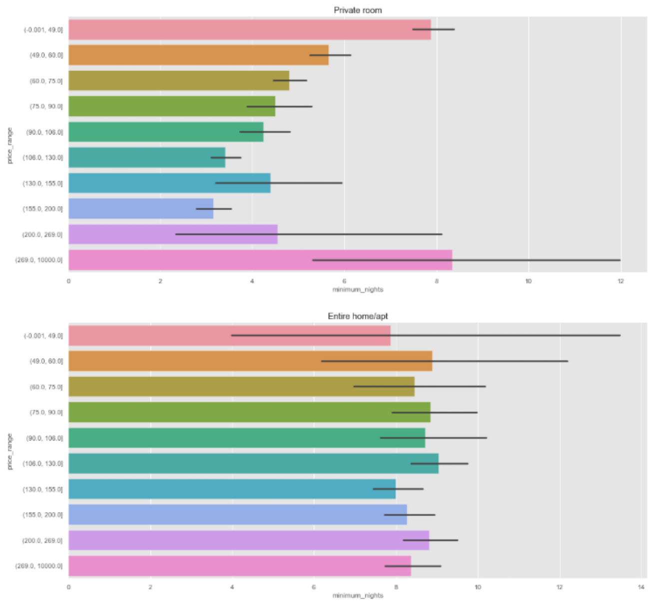


Fig: 14

Correlation Plot:

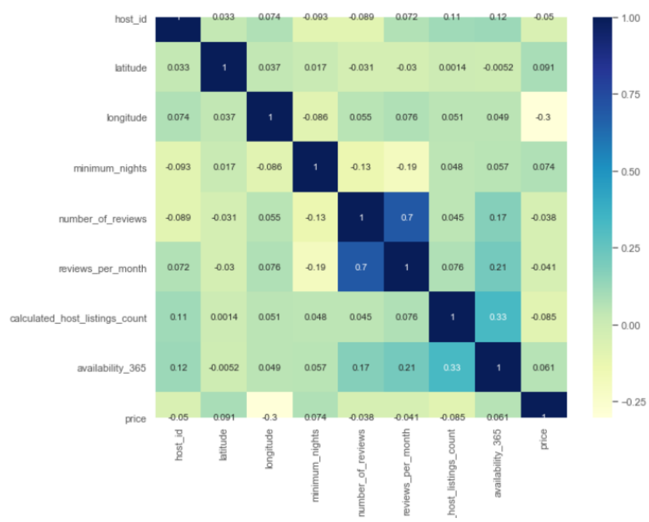


Fig: 15

The review_per_month has higher correlation value of 0.7 SO, we need to remove this column for the prediction.

After doing all the data preprocessing and cleaning I have ended up with 6 influential variables on our target variable.

Applying the data mining algorithm:

Linear Regression: For predicting the prices in Airbnb, I have applied the linear regression algorithm. Here I gave the 6 normalized variables as independent variables and dependent variable is Price.

Coefficients[a]

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.
1	(Constant)	130.568	34.983		3.732	.000
	minimum_nights	-.091	.051	-.008	-1.779	.075
	number_of_reviews	-.276	.024	-.051	-11.576	.000
	availability_365	.183	.008	.100	21.998	.000
	room_type_Privateroom	-98.433	2.191	-.204	-44.934	.000
	room_type_Sharedroom	-128.680	6.934	-.082	-18.558	.000

Fig: 28

From above table we can write the linear equation as follow:

$$Price = -0.91(\min nights) - 0.276\ (no\ of\ reviews) + 0.183(availability\ 365)$$
$$- 98.43\ (room\ type\ private) - 128.6(room\ tye\ shared) + 130.56$$

RMSE: 248.43
R2 score train: 0.08
R2 score test: 0.06

Mean Squared Error: 248.43140058190681
R2 Score: 5.574534134346409
Mean Absolute Error: 77.45655936680035

R square value is 5.57 means 55.7% of data points are near to regression line. The RMSE is 77.4% which means the variance between the target and input variable is about 77%.

Gradient Boosting: I have used this algorithm to predict the price of Airbnb properties once again for the better accuracy and to know which are influential features.

Following are the results of gradient boosting:

Mean Squared Error: 238.19134895287095
R2 Score: 11.12386231007183
Mean Absolute Error: 74.78079019978345

By using the gradient boosting our price prediction accuracy has been improved to 11.2 and the RMSE variance between the target and input variables has been improved to 74%.

Below are the deviance graph for train and test data:

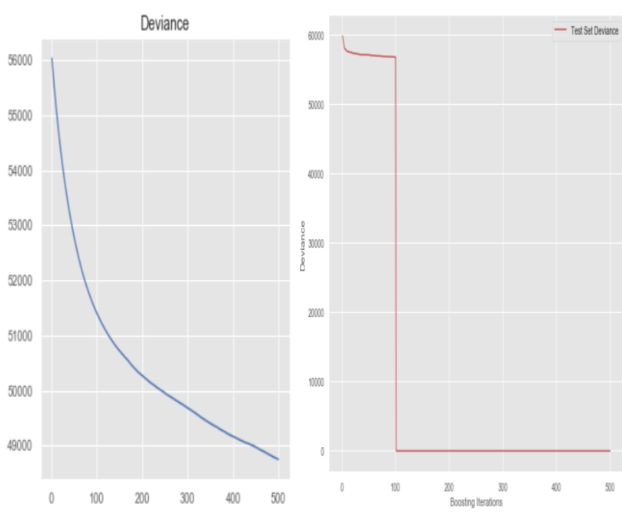


Fig: 29

Here we can understand that boosting algorithm worked well for the training dataset but there is under fitting for test data.

Important features from gradient boosting:

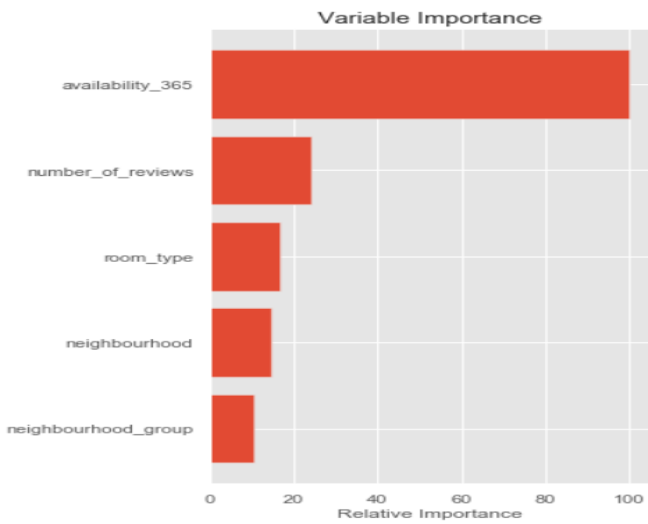


Fig: 30

Decision Tree:

I have used this decision tree algorithm to classify all my categorical and continuous variables, to know which are my top three influential variables in the tree.

The results for decision tree as follow:

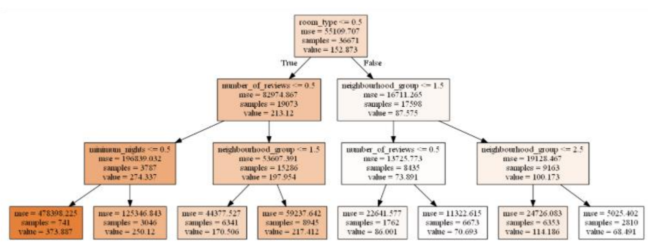


Fig: 31

From the above tree is the finalized pruned decision tree, here the room type has the highest information gain value so it's at the leaf node and on each node we have the sample size till the dataset.

```
Mean Squared Error: 243.46177259382262
R2 Score: 7.1472513058093545
Mean Absolute Error: 79.9806219562876
```

This decision tree has R square value of 71% which means its able to classify 71% of its variables properly. Both R square and MSE are vary good which mean our decision tree is a good fitting model.

1) Evaluation:

Here we can compare all three models and there performances in the accuracy, mean square error and root mean square error.

Algorithm	R Square	MSE	RMSE
Linear Regression	5.574	248.43	77.456
Gradient Boosting	11.12	238.19	74.78
Decision Tree	7.14	243.46	79.98