ALY6020

Pedictive Analysis



Final Project

**Venkat Preran Dakshinadi**

1. **Introduction:**
2. **Analysis:**

The dataset that we are taking into consideration for our final project is AB\_NYC\_2019.

This dataset is about the listing activity and metrics in NYC, NY for 2019. Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world.

This data file includes all needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Columns** | **Column Description** | | id | listing ID | | name | name of the listing | | host\_id | host ID | | host\_name | name of the host | | neighbourhood\_group | location | | neighbourhood | area | | latitude | latitude coordinates | | longitude | longitude coordinates | | room\_type | listing space type | | price | price in dollars | | minimum\_nights | amount of nights minimum | | number\_of\_reviews | number of reviews | | last\_review | latest review | | reviews\_per\_month | number of reviews per month | | calculated\_host\_listings\_count | amount of listing per host | | availability\_365 | number of days when listing is available for booking | |

1. **Importing required libraries and dataset into python:**

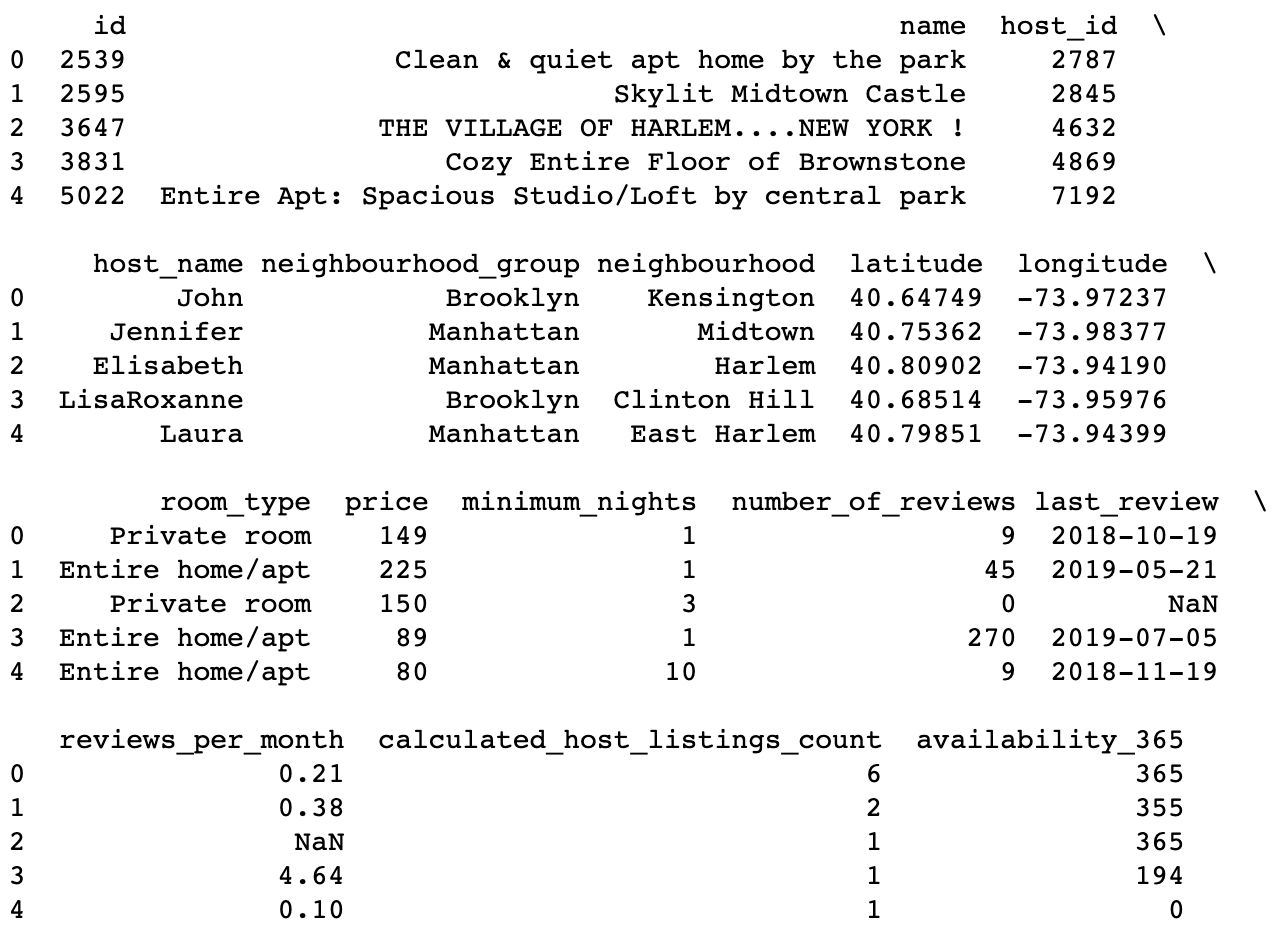
We need to import the following libraries into our system:

1. Numpy
2. Pandas
3. Matplotlib.pyplot
4. Seaborn
5. Sklearn

We will store the excel file into a data frame object. Let’s call it ds. We can validate the data is right or not by printing first few rows of the dataset.



Output:

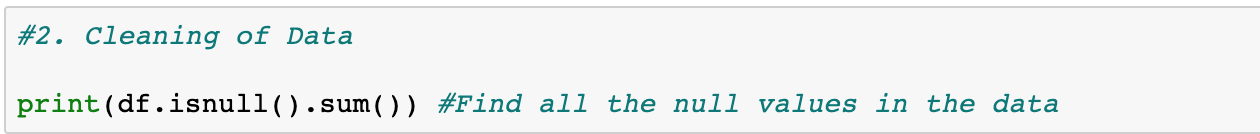


Once we get the data into our system we can move forward and perform the required analysis.

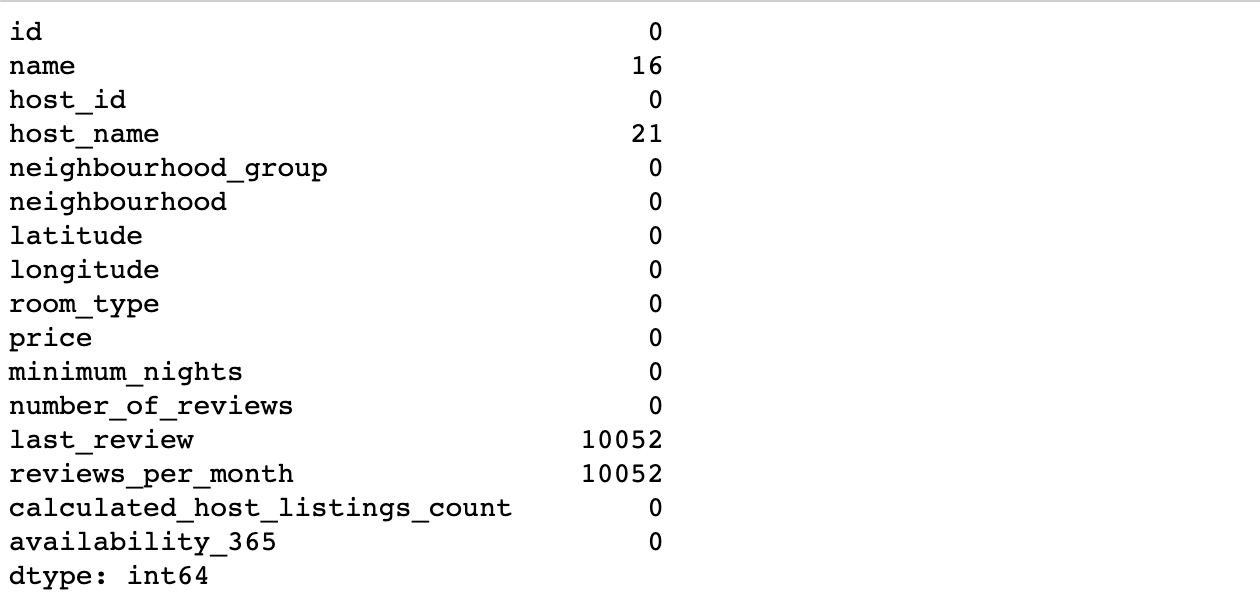
1. **Data cleansing:**

In this part, we will clean our dataset so that we can improve our data quality and in doing so increase the performance.

Let’s first seethe total number of NULL values that we have in our dataset.



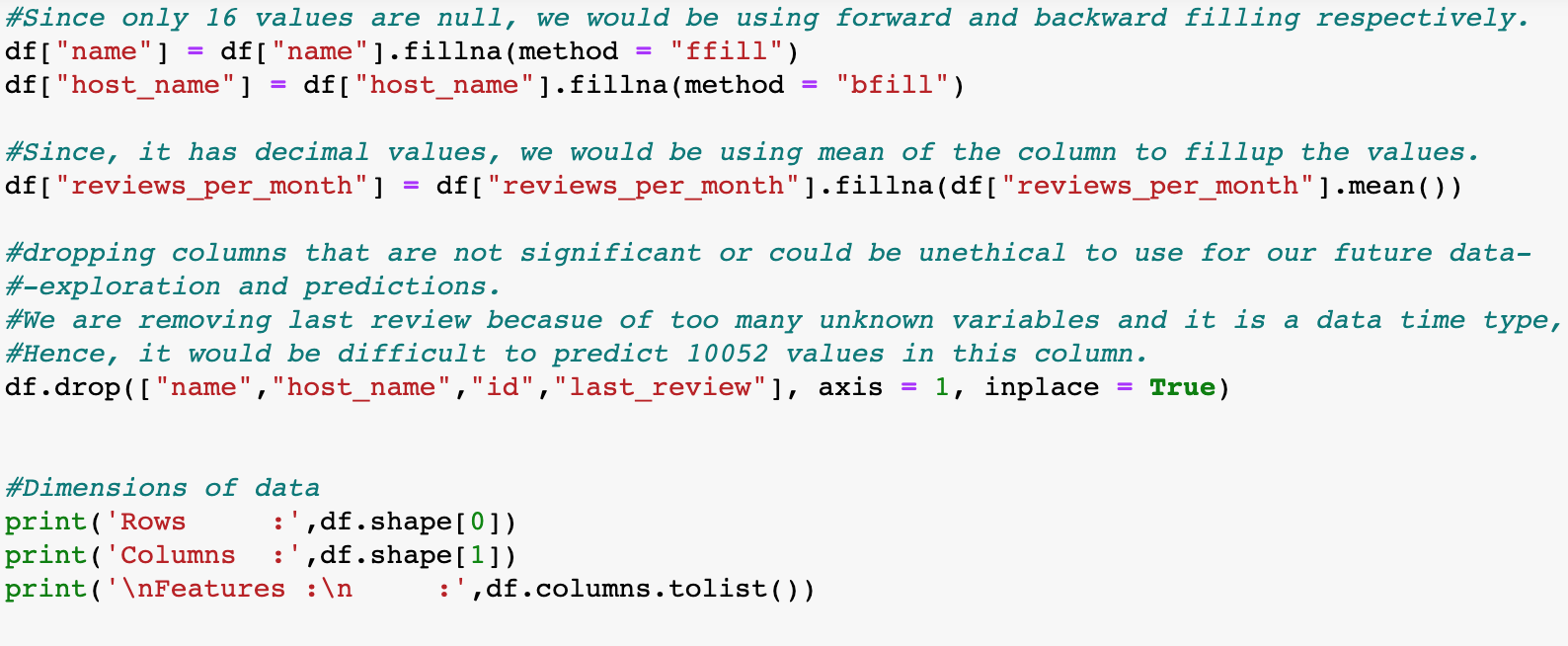
Output:



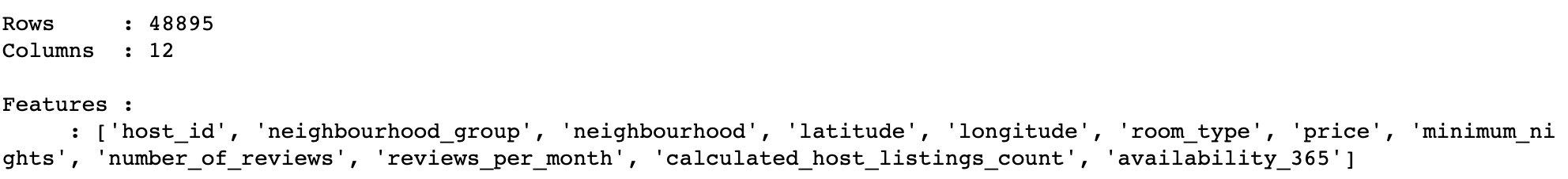
As only 16 values from name and 21 values from host name are there, we will use forward and backward filling on the dataset to fill those.

In case of column reviews per month, we will use mean of the column to fill up the values.

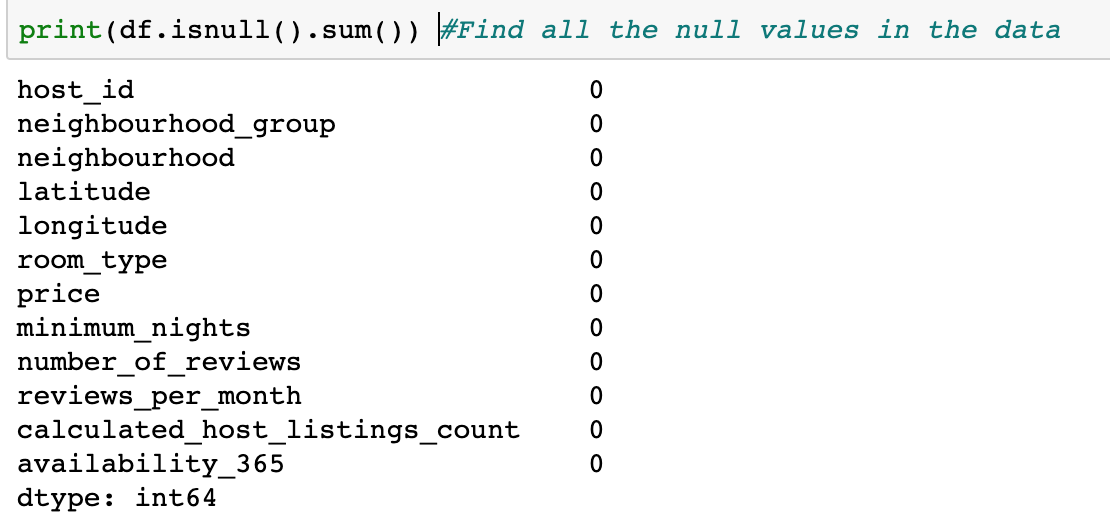
We are removing, last review as there are too many unknowns and it is a date time type. Hence it would be difficult to predict all those 10052 values.



**Output:**



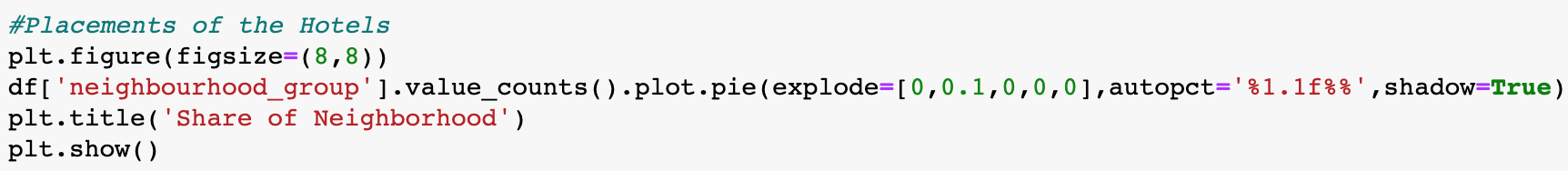
After cleaning our dataset, we can run the ISNULL () method to confirm that there is no NULL value that’s present. And there is no feature with name LAST\_REVIEW as we have removed it.



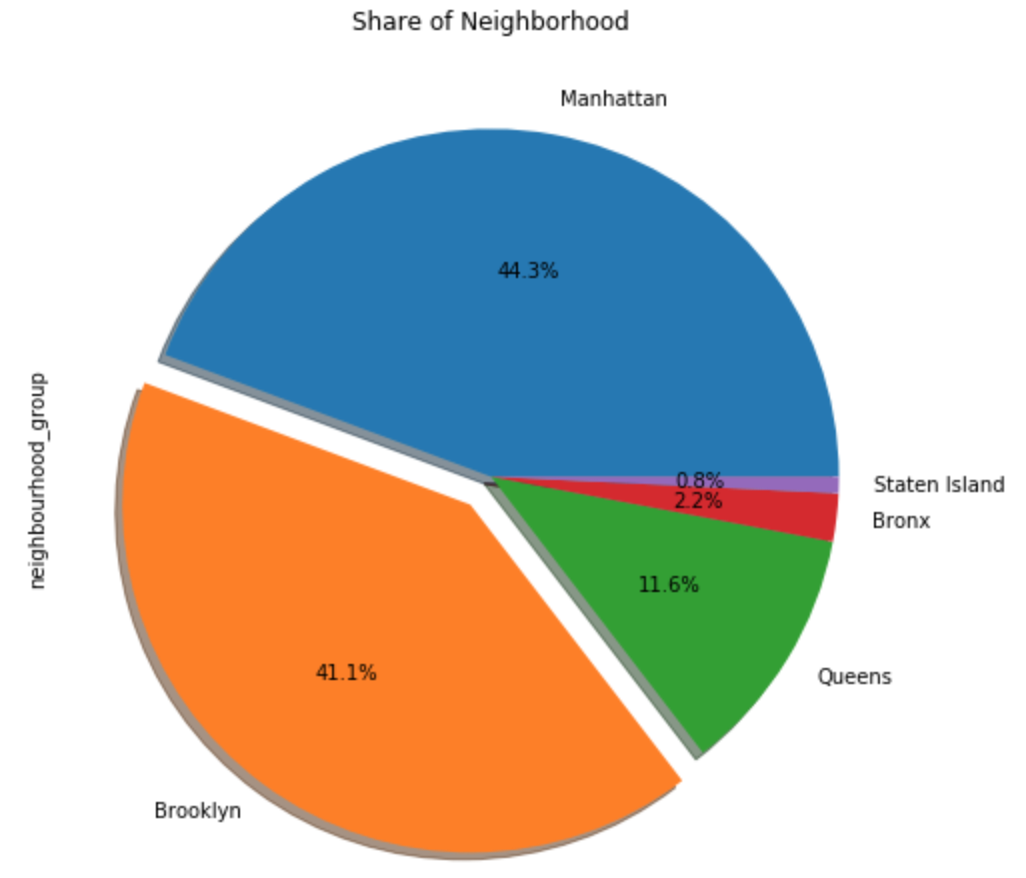
1. **Data visualization:**

After cleaning our dataset, we will now move onto the data visualization part.

1. **Let’s observe the placement of hotels as per the neighborhood group.**



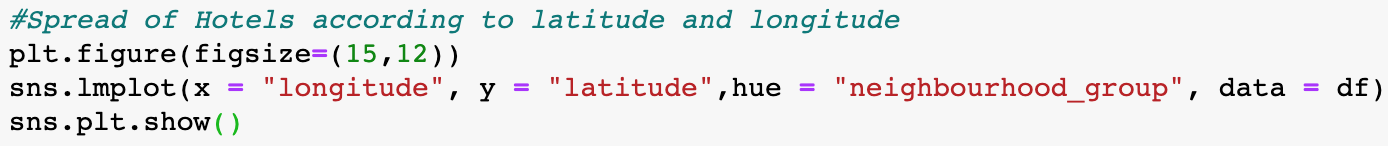
**Output:**



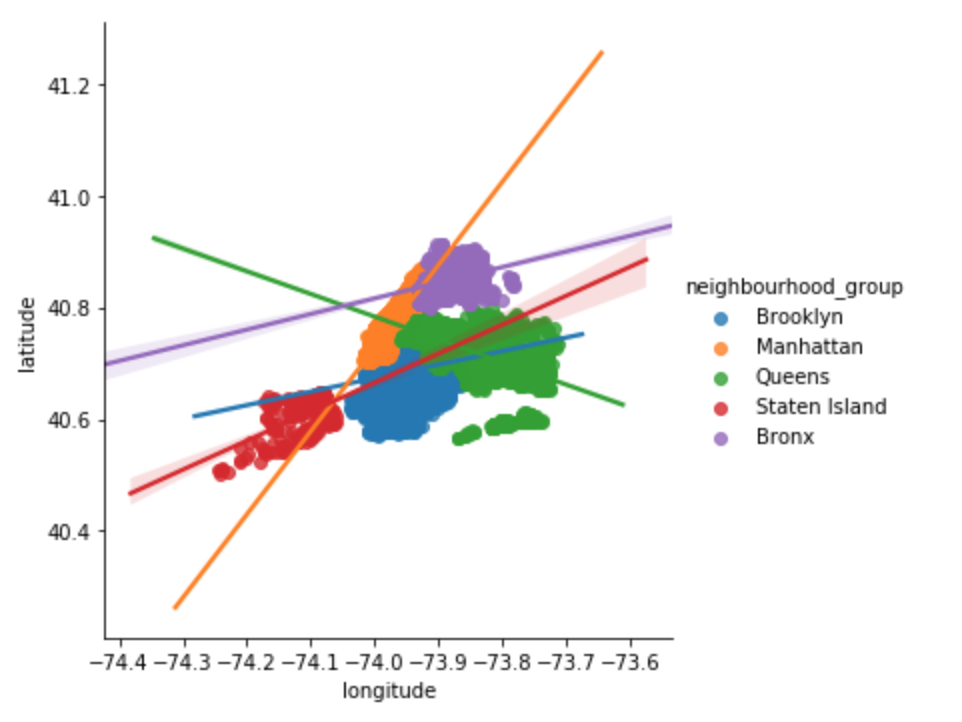
New York city has % county level administrative boroughs namely Manhattan, Brooklyn, queens, Bronx and Staten island. From the above plot, we can conclude that the top boroughs of New York are Manhattan, Brooklyn and Queens. while the least belongs to Bronx and Staten island.

1. **Now let’s observe the spread of hotel according to longitude and latitude:**

We are using simple linear regression to observe the latitude vs longitude graph as per the neighborhood group.



**Output:**



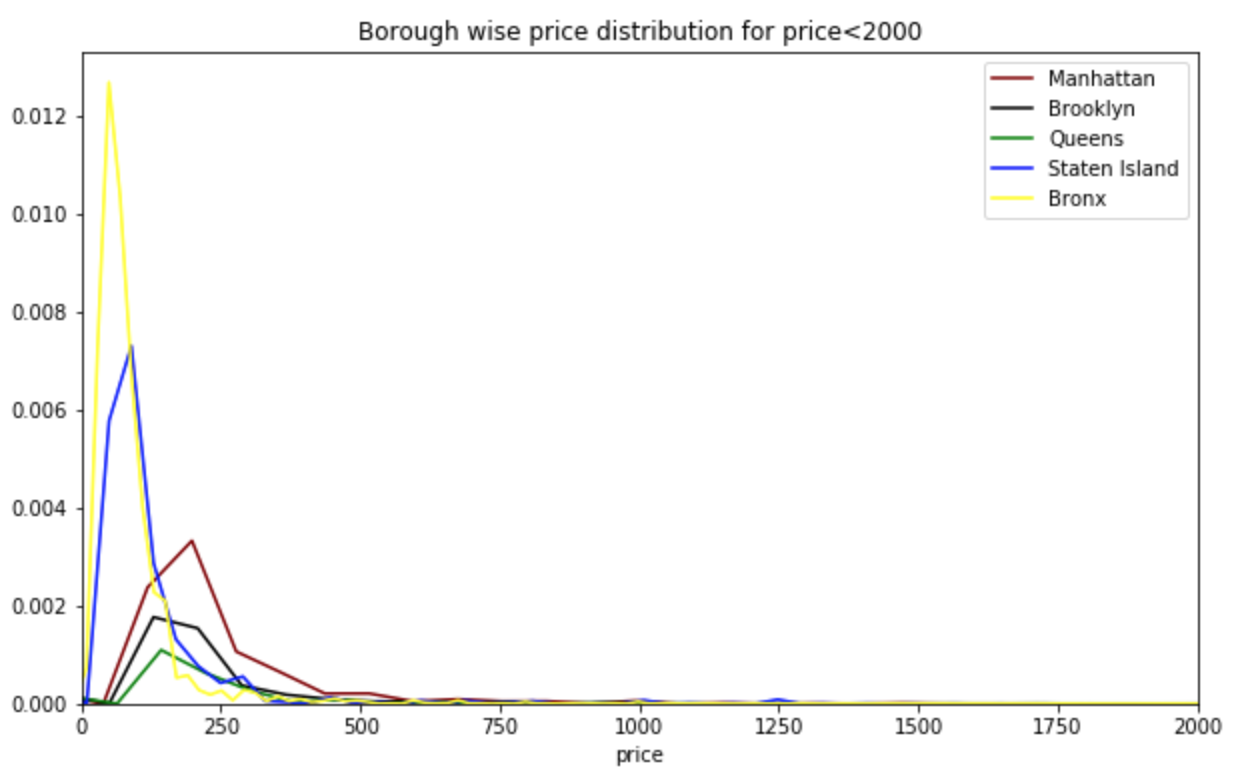
This is an interesting plot as we can see how the data is distributed overall, we can get the sense of how the different boroughs of New York city are placed and the area they occupy within the New York city.

1. **Variations of prices of hotels with area:**

In order to see the price variation as per the location, we can take the column NEIGHBORHOOD.GROUP and see the PRICE column for each group variations.



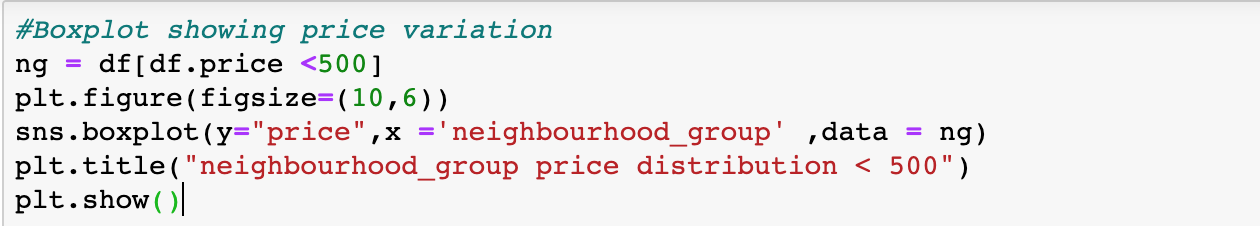
**Output:**



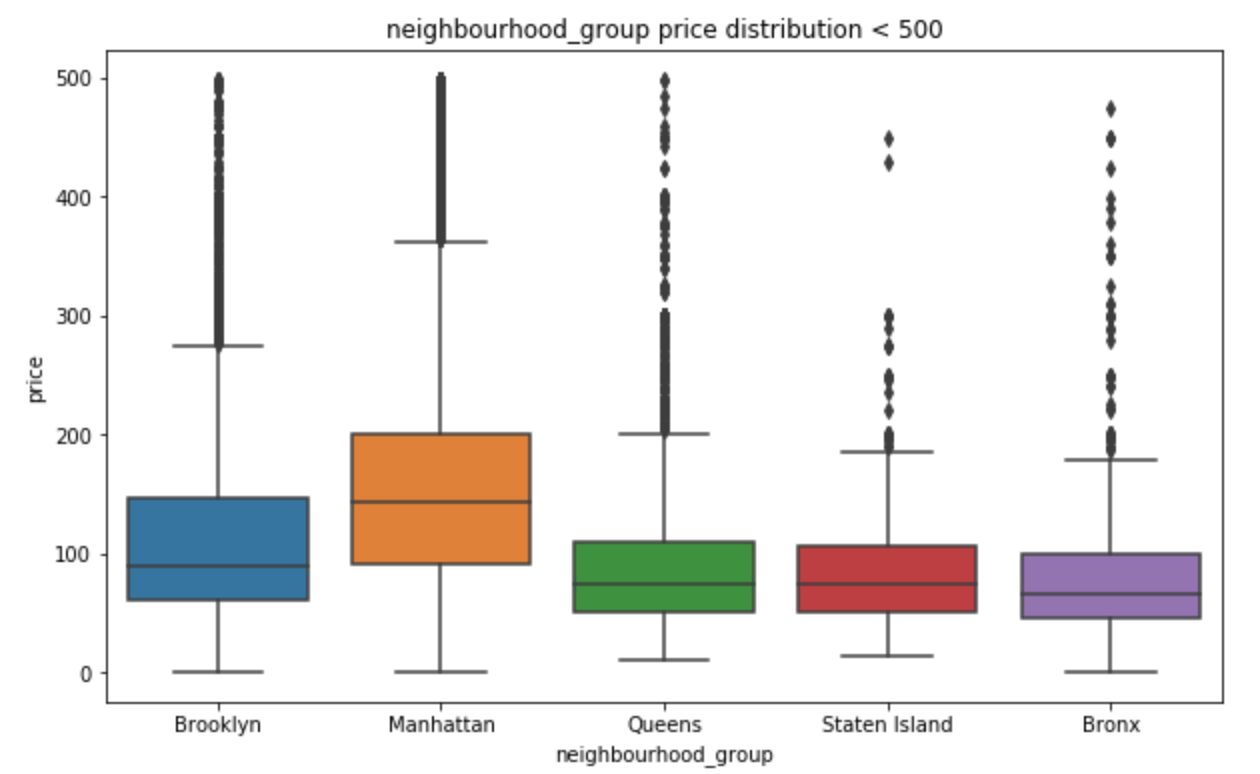
From the plot we can say that Bronx and Staten island are having lot of price variation from 10 to 200 (basically a lot of cheap places to stay). Whereas Manhattan and Brooklyn and Queens are somewhat costlier than the Bronx and Staten Island.

1. **We can also see the boxplot for variation in price:**

When we plot boxplot of the PRICE Vs NEIGHBOURHOOD\_GROUP. The price range that we are taking into consideration for this is from 0 to 500. Because from above graph we can see that most of the prices lie between this price range.



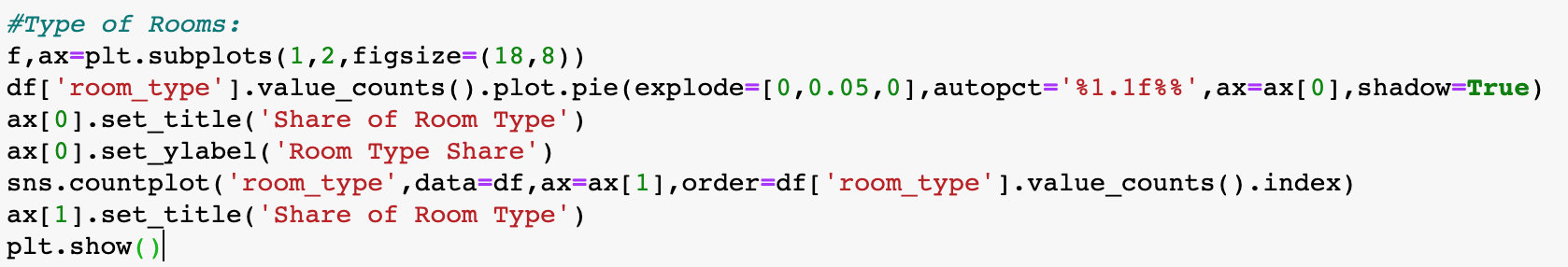
**Output:**



The boxplot tells about that the median, and the ranges of the price. As you can see the median for Queens, Staten island and Bronx are almost the same and the prices for Manhattan and Brooklyn are higher.

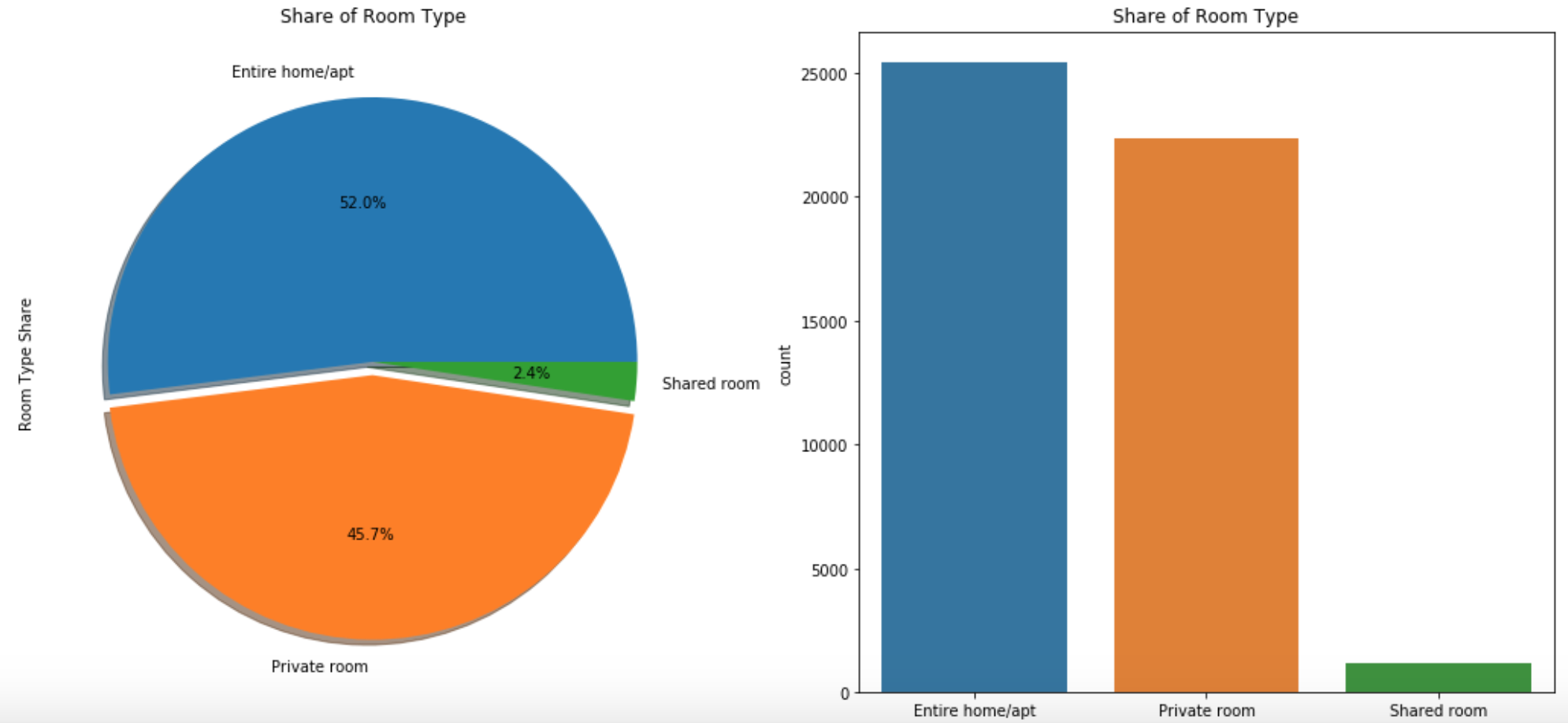
1. **Type of rooms:**

In the ROOM\_TYPE column, there are three different type of rooms – entire home/apartment, private room and shared room. We can check and compare the number for each type with the help of histogram or pie chart as below:



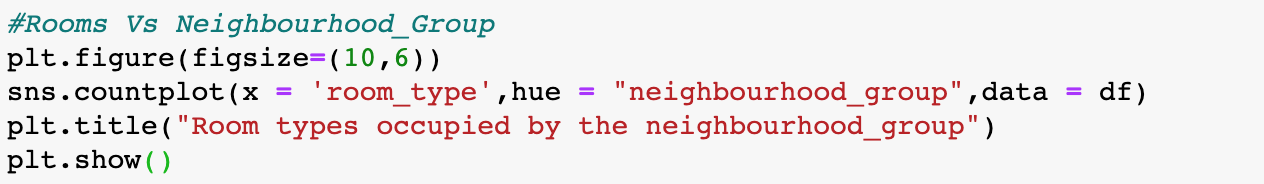
**Output:**

In the output, you can see that more than half of our listings belong to Entire home or apartment category, for which the count is around 25000. Then we have private room which is also around 45% (i.e. around 22,500). Rest of the rooms belong to shared rooms.

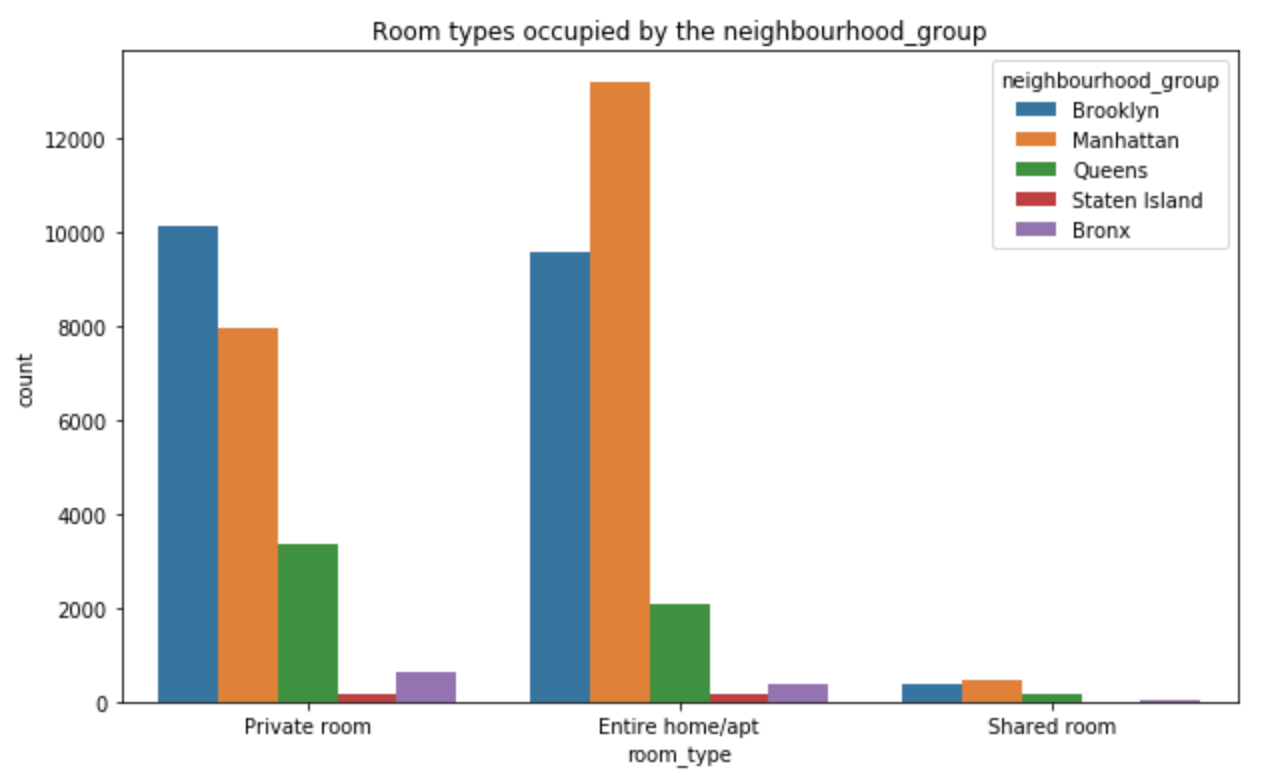


1. **ROOMS vs NEIGHBORHOOD\_GROUP:**

We can also see the distribution of different types of rooms with respect to each borough. Through this graph, we can get the idea of what to expect where. For ex. We can expect high number of private rooms in Brooklyn area. Whereas Manhattan has large number of entire home or the apartments.

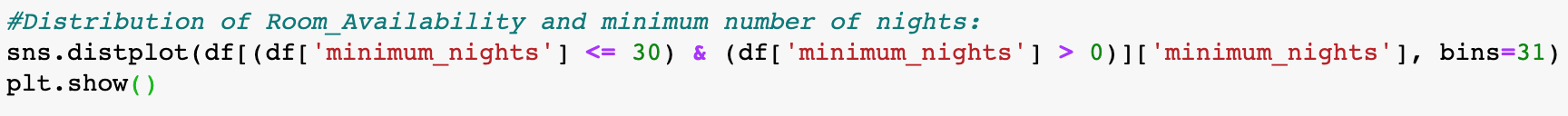


**Output:**



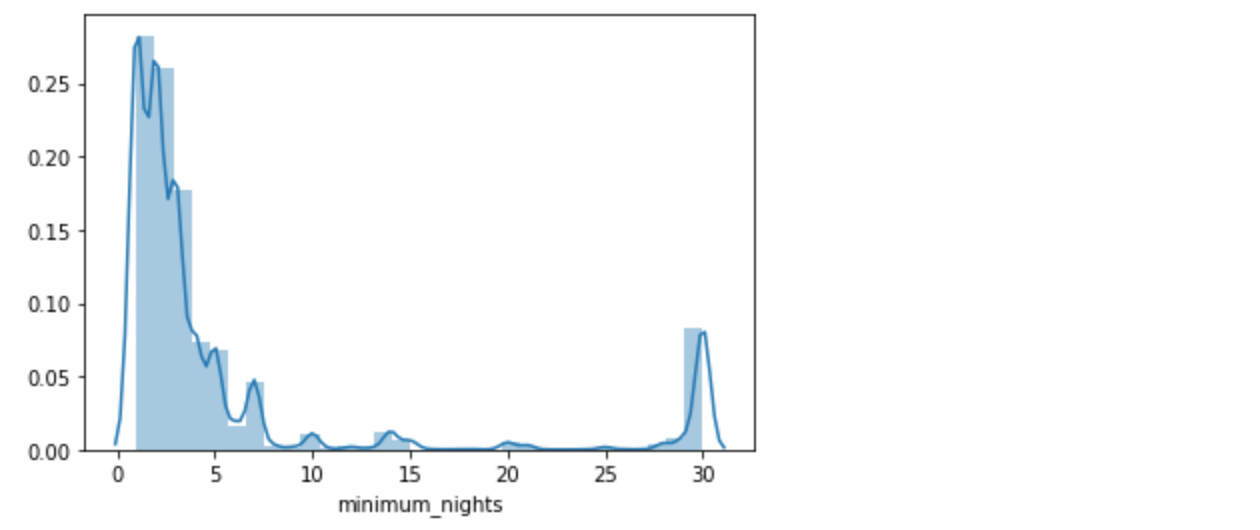
1. **Distribution of ROOM\_AVAILIBILITY and minimum number if nights:**

We can also have a plot that can help us understand what’s the minimum number of durations that one can get. We can plot the MINIMUM\_NIGHTS using DISTPLOT (). This function plots the histogram with a line over it.



**Output:**

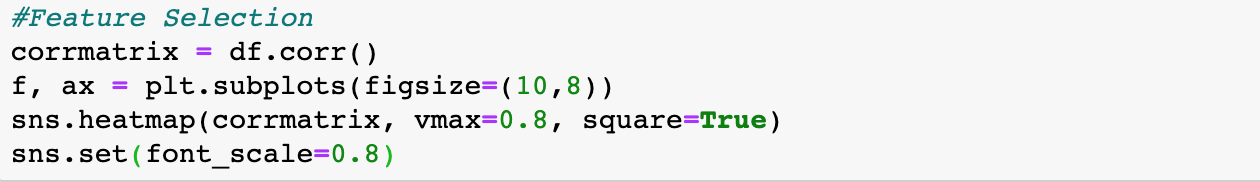
Below graph tells us that most of the listings offer between 1 to 5 days of minimum booking. There are also some of the listings which offer minimum of 30 days of stay which also help us understand that some of the listings are looking for people to stay on a monthly basis. Here we have considered a limit as 30, just to showcase the trend.



1. **Feature selection:**

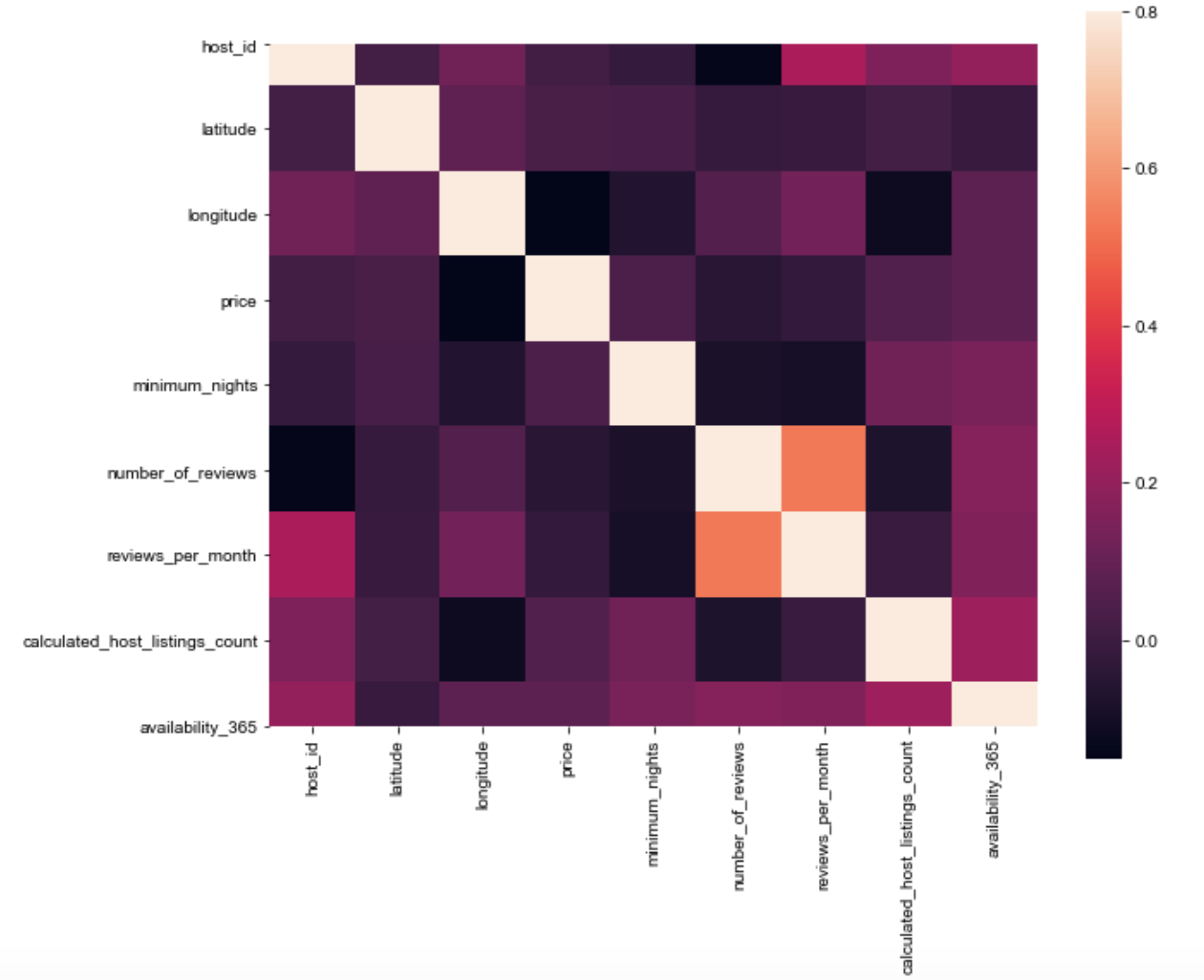
As there are lot of input parameters in our datasets. Hence, we need to perform feature selection to reduce the number of input parameters. this will not only reduce the computational time and model building cost but also it will help in improving our predictive model.

In order to understand what features to select, we can take help of correlation plot. Features that’s show high correlation are linearly dependent and hence those features have the same type of effect on the variable which is dependent. Thus, we can drop one feature from the two.

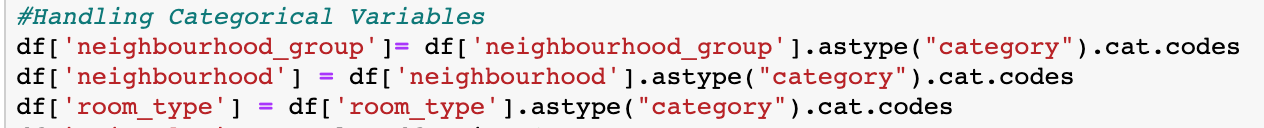


**Output:**

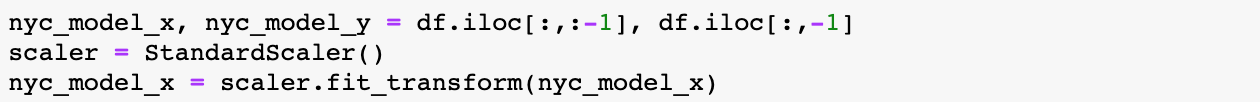
In the correlation plot, we can observe that there is high correlation between number of reviews and reviews per month.

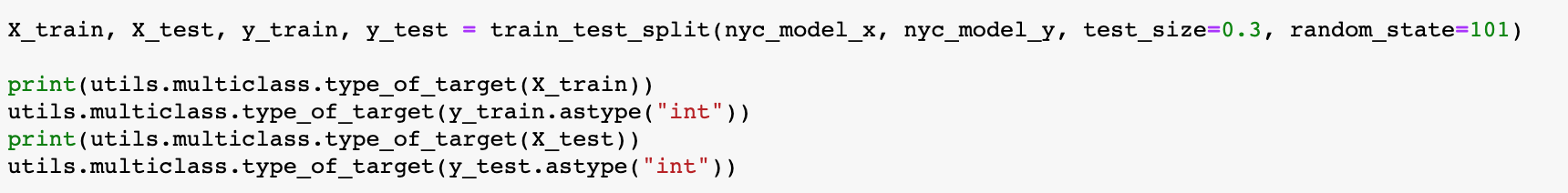


To handle the categorical variable, we need to firstly, categorize them by converting them into numbers and to do this we can use astype(‘category’)cat.codes.





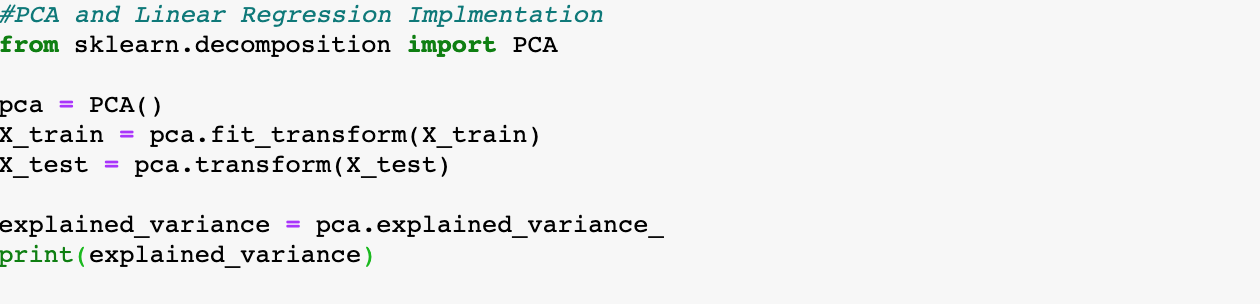




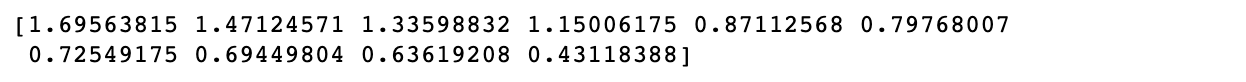
1. **PCA and linear regression implementation:**

PCA is done in order to reduce the present dimensionality in the data. It can also be used to describe the variation present in the data. To use PCA, we need to first import it from SKLEARN.DECOMPOSITION library.

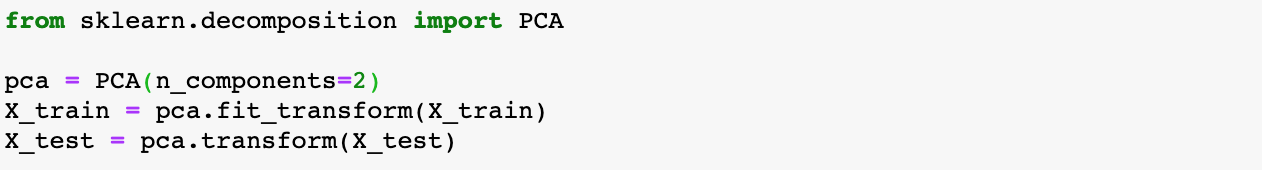
To use PCA, we need to standardize our data. We will normalize our data within a particular range with the help of transform and FIT\_TRANSFORM function. We will do this on our training data first and then transform or standardize our testing data. We can check the discrepancy between the model and our test data with the help of EXPLAINED\_VARIANCE method.



**Output:**

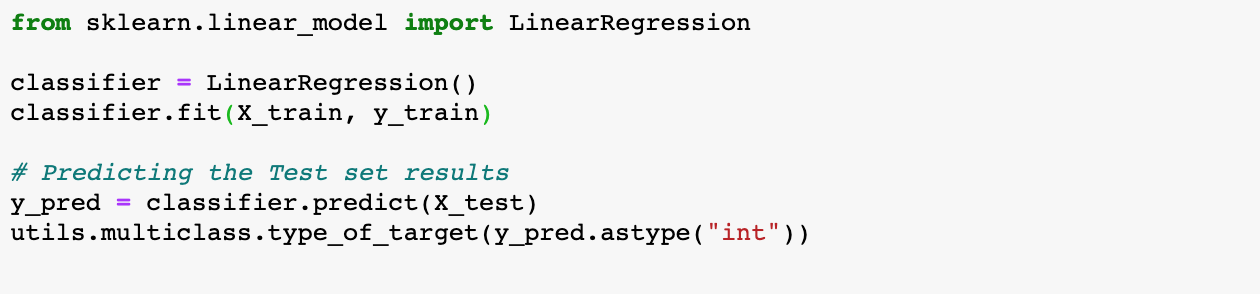


We got this as the output. Thus, we can say that there are two principal components which explains the entire variance of the model. Then we fit and transform our data by the instance of PCA which was created for two principal components.

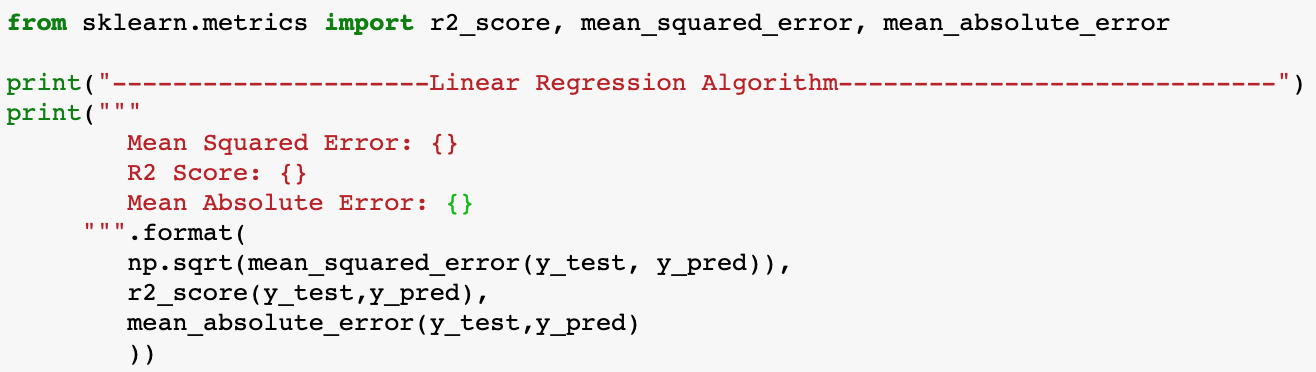


In order to perform linear regression on our dataset, we need to first create a model based on our train data. Then we will test the model with the help of test data. We are creating test and train groups by using train\_test\_split () function. This function will be imported from sklearn.model\_selection library.

Now we create the instance of linear regression and we will try to see how the designed model handles the test data. We want to fit our Y\_TRAIN to our X\_TRAIN.

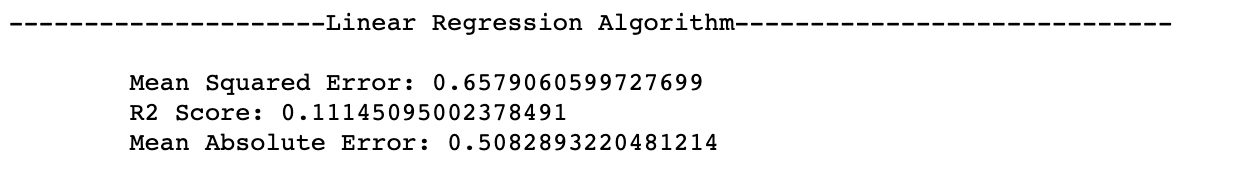


Now we will check the mean square error between our predicted values and actual test value. In order t do so we have written the following code:



**Output:**

As you can see the R2 value is 0.111 and mean squared error is 0.65 this means the difference between the actual and the predicted value is 0.65 which is quite high. Let’s see how the ridge will perform for the same.



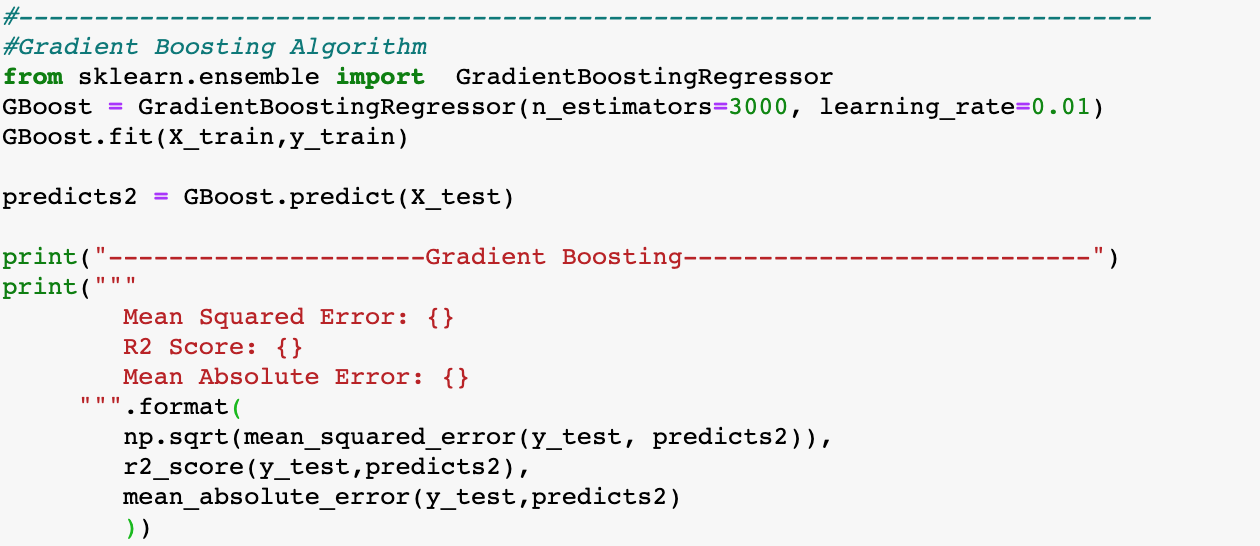
We will test this function with our X\_TRAIN and Y\_TRAIN to see for what value of alpha we get the best model. After running our function, we see that the for alpha 0.01 we get the best parameter value.

K- fold cross validation, to obtain the best model we will perform a resampling method, in this we will divide our datasets in different sets which are called as folds. We will keep changing our test dataset to see which model helps in getting the least possible mean square value.

1. **Gradient boosting algorithm:**

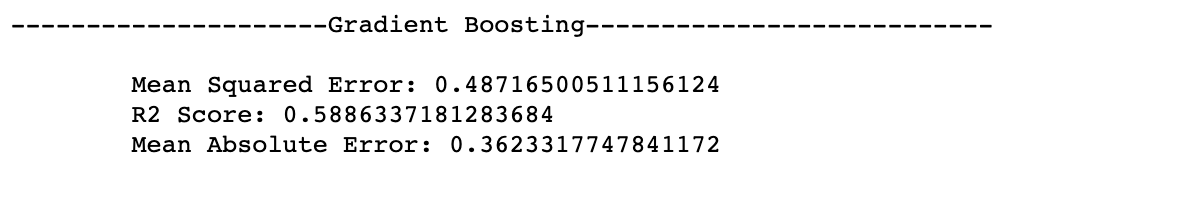
Gradient boosting basically helps the weak learners in the model to learn better. For example, let’s consider Tree 1 of the Decision Tree, it classifies some of the observation easily, this becomes our strong learners, our Tree 1 classifies some of observation incorrectly. Our task is to help these weak learners. Gradient Boosting algorithm computes the loss function, further gradient, and attempts to optimize only certain coefficients of cost function, hence the model is trained better.

We have used GRADIENTBOOSTINGREGRESSOR () method to implement regression using gradient boosting. This function has been imported through SKLEARN.ENSEMBLE. we will fit our model and then see the prediction that our model provides and compare it with the above methods.



**Output:**

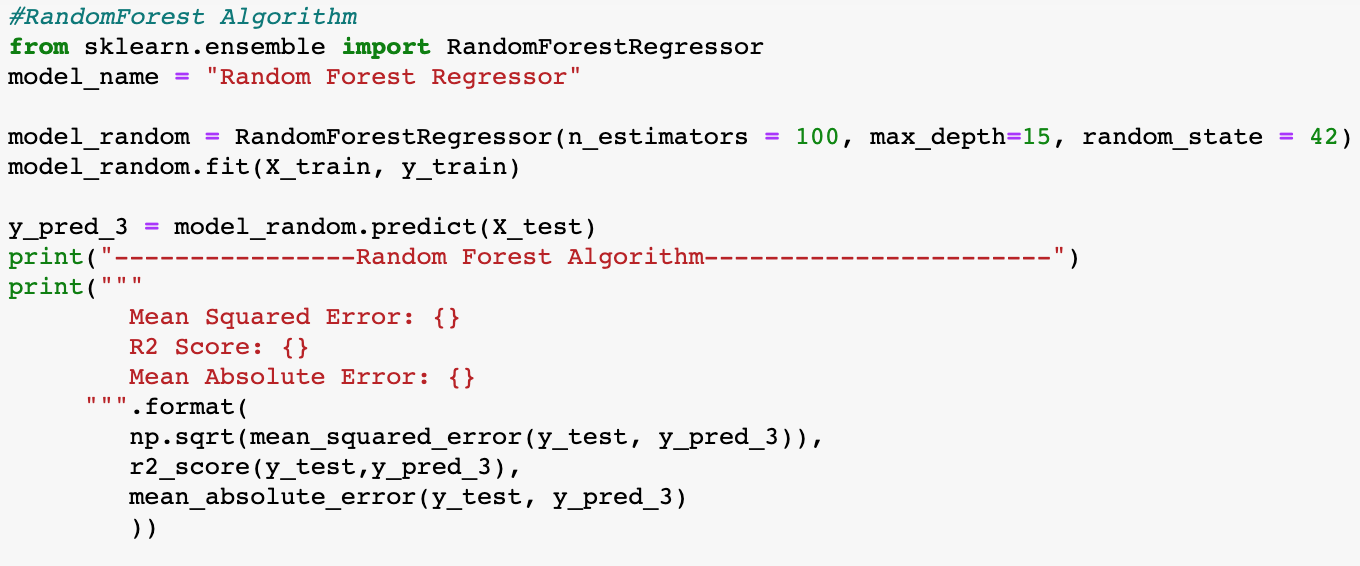
As you can see that we have r2 nearly 60%. And mean square error is coming as 0.49. this help us understand that the model is able to predict in a better way than the lasso and linear regression.



1. **Random forest algorithm:**

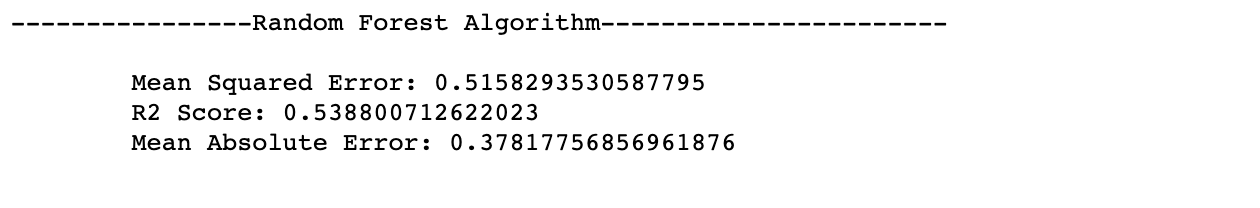
Random forest is a collective learning method for classification, regression and other tasks that operate by making a multitude of decision trees at training time and then we get outputting the classes that is classification or regression of each tree. This technique helps in correcting decision tree habit of overfitting to the available training data set.

To perform this method, we will import randomforestregressor() from the sklearn.ensemble. we will fit our training dataset and use the predict function to see how well our data predict compared to the actual values.

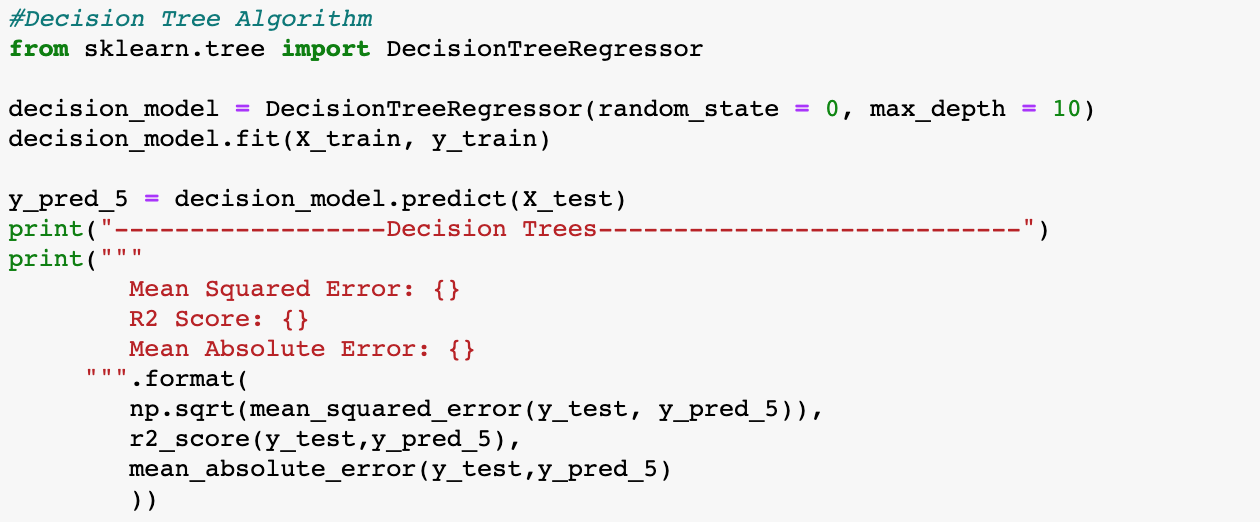


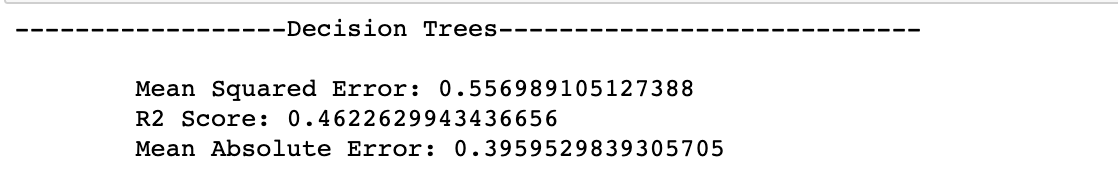
**Output:**

R2 square value that we get through this is 0.53. That means our developed model has 54% correlation and the mean square value is coming as 0.52.



1. **Decision tree implementation:**





1. **Conclusion:**

In the conclusion, I would say that I got to learn about how the solver works with the constraint to find the minimum or maximum optimal solution for a linear equation.

I also got to learn about the terminologies like shadow price, allowable increase and decrease.

1. **References:**

* Blackboard. Module 5: Optimization (I) – Linear Programming. Instructor’s perspective. (2020). Retrieved 7 February 2020, from https://northeastern.blackboard.com/webapps/blackboard/content/listContent.jsp?course\_id=\_2605185\_1&content\_id=\_22127

### Feature selection — Correlation and P-value

#### Feature selection — Correlation and P-value. (2019). Retrieved 14 February 2020, from https://towardsdatascience.com/feature-selection-correlation-and-p-value-da8921bfb3cf