**PREDICTION OF THE COST PER NIGHT FOR THE HOTELS IN ONLINE TRAVEL AGENT PLATFORM**

**Post Graduate Program in Data Science Engineering**

Location: **Bangalore** Batch: **PGPDSE-FT Sept’21**

**Submitted by**

Bhuvanesh

Kumaran

Mohith

Nagesh

Rajat

Varun

**Mentored by**

Mr. Romil Gupta

**ACKNOWLEDGEMENT**

Any endeavour in a specific field requires the guidance and support of many people for successful completion. The sense of achievement on completing anything remains incomplete if the people who were instrumental in its execution are not properly acknowledged. We would like to take this opportunity to verbalize our deepest sense of indebtedness to our project mentor, Mr. Romil Gupta, who was a constant pillar of support and continually provided us with valuable insights to improve upon our project and make it a success. Further, we would like to thank our parents for encouraging us and providing us a platform wherein we got an opportunity to design our own project.

**DECLARATION**

We hereby declare, that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**INTRODUCTION:**

A hotel is a managed building or establishment, which provides guests with a place to stay overnight – on a short-term basis – in exchange for money. Due to the modern internet revolution, the reservation of a room in any hotel can be made via online by using the Online Travel Agents (OTA). These agents create search engine applications that acts as middle man between the hotel merchants and customers to process direct online reservations. An OTA acts as commission-free direct bookings agent for the customers; also provide the way to optimize the merchant’s sales strategy and maximize profit. These agents take an average commission fee from merchants between 10% to 25% after each successful stay of the customers. From the article „‟US Hotel gross booking by OTA with the comparison hotel applications /websites” released by „The wall Street Journal‟ , the country “United States of America” have started to make higher turnover from the F.Y(Financial Year)-2016. By the F.Y-2017, the gross booking by OTA agents made $34.8 billion (B) and whereas the hotel websites/apps made only $32.4B. The key advantages of having the hotel business in OTA platforms are Increased Exposure (Visibility), Drive Traffic, More Bookings, Right Guests, Better Rankings and Better Reviews.

**PROBLEM STATEMENT:**

**1.Business Problem Understanding:**

AirBnB is one such OTA platform acts as the middle man between hotels and customers to book a room; also provides lodging, homestays for vacations and tourism activities. The platform provides businesses an increased opportunity to reach out to a wider customer base and at the same time it empowers the customer by providing them with beforehand details of the type of the property, amenities, location of stay and plan out their budget for the trip accordingly. The major part of the OTA platform is to provide an accurate prediction of Cost Per Night to the customers.

**2.Business Objective:**

**Cost Per Night**: The total amount that end user has to spend for staying at a property for a single night is known as cost per night. There are various factors that determine the cost per night from property to property also depends upon features like quality of the property, its size, neighborhood, facilities/amenities and location

Travelers often budget their travel costs in order to ensure that they spend only the optimal amount and knowing cost per night beforehand would help them plan their travel efficiently. This helps the OTA platforms by providing a comparison benchmark between properties and also help in identifying the various features of a property that will determine the cost per night. This information will also help the host/owners of a property in identifying the short coming of their property and help improve their properties by matching facilities/amenities provided by their competitors.

**VARIABLE CATEGORIZATION WITH DESCRIPTION:**

The dataset consists of 74 variables. Out of these variables 73 are independent variables and 1 is a target variable. The variables are a mixture of both numerical and categorical type.

1. **Numerical:**

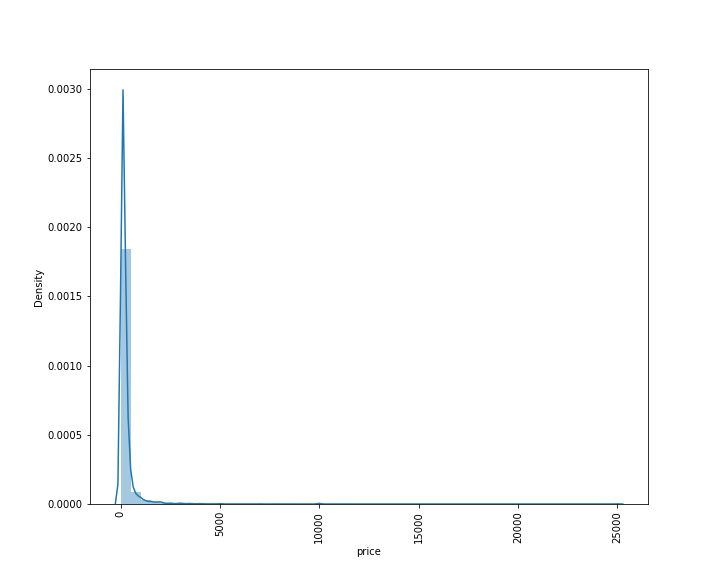
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Attributes** | **Dtype** | **Desicription** |
| 1 | id | int64 | Airbnb's unique identifier for the listing |
| 2 | scrape\_id | int64 | Inside Airbnb "Scrape" this was part of |
| 3 | host\_id | int64 | Airbnb's unique identifier for the host/user |
| 4 | host\_listings\_count | float64 | The number of listings the host has. |
| 5 | host\_total\_listings\_count | float64 | The number of listings the host has. |
| 6 | latitude | float64 | Uses the World Geodetic System (WGS84) projection for latitude and longitude. |
| 7 | longitude | float64 | Uses the World Geodetic System (WGS84) projection for latitude and longitude. |
| 8 | accommodates | int64 | The maximum capacity of the listing |
| 9 | bathrooms | float64 | The number of bathrooms in the listing |
| 10 | bedrooms | float64 | The number of bedrooms |
| 11 | beds | float64 | The number of bed(s) |
| 12 | minimum\_nights | int64 | minimum number of night stay for the listing. |
| 13 | maximum\_nights | int64 | maximum number of night stay for the listing. |
| 14 | minimum\_minimum\_nights | float64 | the smallest minimum\_night value from the calender (looking 365 nights in the future) |
| 15 | maximum\_minimum\_nights | float64 | the largest minimum\_night value from the calender (looking 365 nights in the future) |
| 16 | minimum\_maximum\_nights | float64 | the smallest maximum\_night value from the calender (looking 365 nights in the future) |
| 17 | maximum\_maximum\_nights | float64 | the largest maximum\_night value from the calender (looking 365 nights in the future) |
| 18 | minimum\_nights\_avg\_ntm | float64 | the average minimum\_night value from the calender (looking 365 nights in the future) |
| 19 | maximum\_nights\_avg\_ntm | float64 | the average maximum\_night value from the calender (looking 365 nights in the future) |
| 20 | calendar\_updated | float64 | Calendar |
| 21 | availability\_30 | int64 | avaliability\_30. The availability of the listing 30 days in the future as determined by the calendar. |
| 22 | availability\_60 | int64 | avaliability\_60. The availability of the listing 60 days in the future as determined by the calendar. |
| 23 | availability\_90 | int64 | avaliability\_90. The availability of the listing 90 days in the future as determined by the calendar. |
| 24 | availability\_365 | int64 | avaliability\_365. The availability of the listing 365 days in the future as determined by the calendar. |
| 25 | number\_of\_reviews | int64 | The number of reviews the listing has |
| 26 | number\_of\_reviews\_ltm | int64 | The number of reviews the listing has (in the last 12 months) |
| 27 | number\_of\_reviews\_l30d | int64 | The number of reviews the listing has (in the last 30 days) |
| 28 | review\_scores\_rating | float64 | The overall review for the stay |
| 29 | review\_scores\_accuracy | float64 | The accuracy review score for the stay |
| 30 | review\_scores\_cleanliness | float64 | The cleanliness review score for the stay |
| 31 | review\_scores\_checkin | float64 | The checkin review score for the stay |
| 32 | review\_scores\_communication | float64 | The communication review score for the stay |
| 33 | review\_scores\_location | float64 | The location review score for the stay |
| 34 | review\_scores\_value | float64 | The value review score for the stay |
| 35 | calculated\_host\_listings\_count | int64 | The number of listings the host has in the current scrape, in the city/region geography. |
| 36 | calculated\_host\_listings\_count\_entire\_homes | int64 | The number of Entire home/apt listings the host has in the current scrape, in the city/region geography |
| 37 | calculated\_host\_listings\_count\_private\_rooms | int64 | The number of Private room listings the host has in the current scrape, in the city/region geography |
| 38 | calculated\_host\_listings\_count\_shared\_rooms | int64 | The number of Shared room listings the host has in the current scrape, in the city/region geography |
| 39 | reviews\_per\_month | float64 | The number of reviews the listing has over the lifetime of the listing |

1. **Categorical:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Attributes** | **Dtype** | **Desicription** |
| 1 | listing\_url | object | Airbnb's unique url for the listing |
| 2 | last\_scraped | object | UTC. The date and time this listing was "scraped". |
| 3 | name | object | Name of the listing |
| 4 | description | object | Detailed description of the listing |
| 5 | neighborhood\_overview | object | Host's description of the neighbourhood |
| 6 | picture\_url | object | URL to the Airbnb hosted regular sized image for the listing |
| 7 | host\_url | object | The Airbnb page for the host |
| 8 | host\_name | object | Name of the host. Usually just the first name(s). |
| 9 | host\_since | object | The date the host/user was created. For hosts that are Airbnb guests this could be the date they registered as a guest. |
| 10 | host\_location | object | The host's self-reported location |
| 11 | host\_about | object | Description about the host |
| 12 | host\_response\_time | object | The Time the host take to respond to a booking |
| 13 | host\_response\_rate | object | The rate at which a host responds booking requests. |
| 14 | host\_acceptance\_rate | object | The rate at which a host accepts booking requests. |
| 15 | host\_is\_superhost | object | [t=true; f=false] |
| 16 | host\_thumbnail\_url | object | Url for host thumbnail |
| 17 | host\_picture\_url | object | Url for host pictures |
| 18 | host\_neighbourhood | object | The neighbourhood that the host is located |
| 19 | host\_verifications | object | Host verification |
| 20 | host\_has\_profile\_pic | object | Host profile picture |
| 21 | host\_identity\_verified | object | Host identity verified |
| 22 | neighbourhood | object | The neighbourhood as geocoded using the latitude and longitude |
| 23 | neighbourhood\_cleansed | object | The neighbourhood as geocoded using the latitude and longitude against neighborhoods as defined by open or public digital shapefiles. |
| 24 | neighbourhood\_group\_cleansed | object | The neighborhood group as geocoded using the latitude and longitude against neighborhoods as defined by open or public digital shapefiles. |
| 25 | property\_type | object | Self-selected property type. Hotels and Bed and Breakfasts are described as such by their hosts in this field |
| 26 | room\_type | object | All homes are grouped into the following three-room types:  Entire place Private room Shared room Entire place |
| 27 | bathrooms\_text | object | The number of bathrooms in the listing.  On the Airbnb web-site, the bathrooms field has evolved from a number to a textual description. For older scrapes, bathrooms are used. |
| 28 | amenities | object | The various consumable and facilities that come with the room |
| 29 | has\_availability | object | [t=true; f=false] |
| 30 | calendar\_last\_scraped | object | Calendar\_last\_scarped |
| 31 | first\_review | object | The date of the first/oldest review |
| 32 | last\_review | object | The date of the last/newest review |
| 33 | license | object | The licence/permit/registration number |
| 34 | instant\_bookable | object | [t=true; f=false]. Whether the guest can automatically book the listing without the host requiring to accept their booking request. An indicator of a commercial listing. |

1. **Target Variable**

The target variable of the above dataset is price. We have to predict the daily price for the stay.



We observe that the price values are highly right or positively skewed

**METHODOLOGY TO BE FOLLOWED**

CRISP-DM which stands for Cross Industry Standard Process for Data Mining is a methodology created to help shape data mining projects. It describes the different phases/tasks involved in the project and provides an overview of data mining life cycle.

**1. Business Understanding** - It focuses on determining the business requirements/objectives and understanding what outcome to achieve. Also determine the business units being affected. Convert this business problem into a data mining problem and carve out an initial plan.

* Determine the business objectives: Understand what is needed to be accomplished for the customer.
* Assess situation: Determine resources availability, project requirements, assess risks and contingencies, and conduct a cost-benefit analysis.
* Determine data mining goals: Convert business problem to a data mining problem and recognize the data mining problem type such as classification, regression or clustering, etc.
* Produce a project plan: Devise a step-to-step plan for executing the project.

**2. Data understanding -** This phase starts with collecting the data and then examining the data for its surface properties like data format, number of records, etc. The next step is to better understand the data by understanding each attribute and perform basic statistics on them. Understand the relationship between different attributes. Determine the quality of data by checking the missing values, outliers, duplicates, etc.

* + Collect initial data: Acquire the data and load it into the analysis tool to be used.
  + Describe data: Examine the data and document its surface properties like data format, number of records, or field identities. Understand the meaning of each attribute and attribute value in business terms. For each attribute, compute basic statistics so as to get a higher-level understanding.
  + Explore data: Find insights from the data. Query it, visualize it, and identify relationships among the data.
  + Verify data quality: Identify special values, missing attributes and null data. Determine how clean/dirty is the data.

**3. Data preparation -** This stage, which is often referred to as data wrangling, has the objective to develop the final data set for EDA and modelling. Covers all activities to construct the final dataset from the initial raw data. Some of the tasks include table, record and attribute selection as well as transformation and cleaning of data for modelling tools.

* + Select data: Determine which attributes/features will be used and document reasons for inclusion/exclusion.
  + Clean data: Correct, impute and remove the improper data.
  + Extract data: Derive new attributes from the existing ones
  + Integrate data: Create features by combining data from multiple sources.
  + Format data: Re-format data as necessary. For example, convert string values to numeric values so as to perform mathematical operations.

**4. Modelling -** In this stage we build and assess different models built using various techniques from the training dataset.

* Select modelling technique: Determine the algorithms to be used to model the data based on the business requirement.
* Generate test design: In order to build and test the model, we need to divide the dataset into training and testing data set. In this step we divide the data into train and test data set.
* Build model: Based on the modelling technique selected, build the model on the input data set.
* Assess model: Compare the results of different models based on confusion matrix. The outcome of this step frequently leads to model tuning iterations until the best model is found.

**5. Evaluation -** Evaluate the models and review the steps executed to construct the model to be certain it properly achieves the business objectives.

* Evaluate results: Understand the data mining results and check how impactful they are in achieving the data mining goal. Select appropriate model based on confusion matrix.
* Review process: Review the work accomplished and make sure that nothing was overlooked and all steps were properly executed. Summarize the findings and correct anything if needed.
* Determine next steps: Based on the previous three tasks, determine whether to proceed to deployment, iterate further, or initiate new projects.

**DATA PRE-PROCESSING:**

* Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.
* A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.
* The data consists of 33000 rows and 74 columns. Out of these we have 35 categorical columns and the rest as numerical.

**1. Variable Datatype:**

* We first check the data types of each of the columns of the data.

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Attributes** | **Dtype** |
| 1 | id | int64 |
| 2 | listing\_url | object |
| 3 | scrape\_id | int64 |
| 4 | last\_scraped | object |
| 5 | name | object |
| 6 | description | object |
| 7 | neighborhood\_overview | object |
| 8 | picture\_url | object |
| 9 | host\_id | int64 |
| 10 | host\_url | object |
| 11 | host\_name | object |
| 12 | host\_since | object |
| 13 | host\_location | object |
| 14 | host\_about | object |
| 15 | host\_response\_time | object |
| 16 | host\_response\_rate | object |
| 17 | host\_acceptance\_rate | object |
| 18 | host\_is\_superhost | object |
| 19 | host\_thumbnail\_url | object |
| 20 | host\_picture\_url | object |
| 21 | host\_neighbourhood | object |
| 22 | host\_listings\_count | float64 |
| 23 | host\_total\_listings\_count | float64 |
| 24 | host\_verifications | object |
| 25 | host\_has\_profile\_pic | object |
| 26 | host\_identity\_verified | object |
| 27 | neighbourhood | object |
| 28 | neighbourhood\_cleansed | object |
| 29 | neighbourhood\_group\_cleansed | object |
| 30 | latitude | float64 |
| 31 | longitude | float64 |
| 32 | property\_type | object |
| 33 | room\_type | object |
| 34 | accommodates | int64 |
| 35 | bathrooms | float64 |
| 36 | bathrooms\_text | object |
| 37 | bedrooms | float64 |
| 38 | beds | float64 |
| 39 | amenities | object |
| 40 | price | object |
| 41 | minimum\_nights | int64 |
| 42 | maximum\_nights | int64 |
| 43 | minimum\_minimum\_nights | float64 |
| 44 | maximum\_minimum\_nights | float64 |
| 45 | minimum\_maximum\_nights | float64 |
| 46 | maximum\_maximum\_nights | float64 |
| 47 | minimum\_nights\_avg\_ntm | float64 |
| 48 | maximum\_nights\_avg\_ntm | float64 |
| 49 | calendar\_updated | float64 |
| 50 | has\_availability | object |
| 51 | availability\_30 | int64 |
| 52 | availability\_60 | int64 |
| 53 | availability\_90 | int64 |
| 54 | availability\_365 | int64 |
| 55 | calendar\_last\_scraped | object |
| 56 | number\_of\_reviews | int64 |
| 57 | number\_of\_reviews\_ltm | int64 |
| 58 | number\_of\_reviews\_l30d | int64 |
| 59 | first\_review | object |
| 60 | last\_review | object |
| 61 | review\_scores\_rating | float64 |
| 62 | review\_scores\_accuracy | float64 |
| 63 | review\_scores\_cleanliness | float64 |
| 64 | review\_scores\_checkin | float64 |
| 65 | review\_scores\_communication | float64 |
| 66 | review\_scores\_location | float64 |
| 67 | review\_scores\_value | float64 |
| 68 | license | object |
| 69 | instant\_bookable | object |
| 70 | calculated\_host\_listings\_count | int64 |
| 71 | calculated\_host\_listings\_count\_entire\_homes | int64 |
| 72 | calculated\_host\_listings\_count\_private\_rooms | int64 |
| 73 | calculated\_host\_listings\_count\_shared\_rooms | int64 |
| 74 | reviews\_per\_month | float64 |

* From here we observe that host\_response\_rate, host\_acceptance\_rate and the target variable ‘price’ are numerical data but are stated as object. We need to convert them to numerical data by removing special characters like %, $ and comma.

**2. Removal of unwanted attribute:**

* + After checking the dataset in detail, we can see the several of the categorical have 100% unique variable or are URLs which will not provide any value to the predictions, hence we can drop these variables.
  + Similarly numerical variable like id, host\_id and scrape\_id will not provide any value to the predictions; hence we can drop these variables also.
  + Variables with 100% missing values such as calendar updated and bathrooms can be dropped.

**3. Missing Value Treatment:**

The next step of data pre-processing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Attributes** | **Null Value Percentage** | **Dtype** |
| 1 | neighborhood | 39.30 | object |
| 2 | Host\_response\_time | 32.42 | object |
| 3 | host\_response\_rate(in %) | 32.42 | float64 |
| 4 | host\_acceptance\_rate(in %) | 31.22 | float64 |
| 5 | review\_scores\_value | 25.16 | float64 |
| 6 | review\_scores\_location | 25.15 | float64 |
| 7 | review\_scores\_checkin | 25.14 | float64 |
| 8 | review\_scores\_communication | 25.11 | float64 |
| 9 | review\_scores\_accuracy | 25.11 | float64 |
| 10 | review\_scores\_cleanliness | 25.11 | float64 |
| 11 | reviews\_per\_month | 24.09 | float64 |
| 12 | review\_scores\_rating | 24.09 | float64 |
| 13 | host\_neighbourhood | 19.90 | object |
| 14 | bedrooms | 11.46 | float64 |
| 15 | beds | 6.19 | float64 |
| 16 | bathrooms\_text | 0.19 | object |
| 17 | host\_total\_listings\_count | 0.08 | float64 |
| 18 | host\_listings\_count | 0.08 | float64 |
| 19 | Host\_is\_superhost | 0.08 | object |
| 20 | minimum\_maximum\_nights | 0.01 | float64 |
| 21 | minimum\_nights\_avg\_ntm | 0.01 | float64 |
| 22 | minimum\_nights\_avg\_ntm | 0.01 | float64 |
| 23 | maximum\_maximum\_nights | 0.01 | float64 |
| 24 | maximum\_minimum\_nights | 0.01 | float64 |
| 25 | minimum\_minimum\_nights | 0.01 | float64 |

* We can observe that 25 attributes have missing values in them.
* For initial assessment we can impute the entire numerical variable with median and the categorical variables with mode.

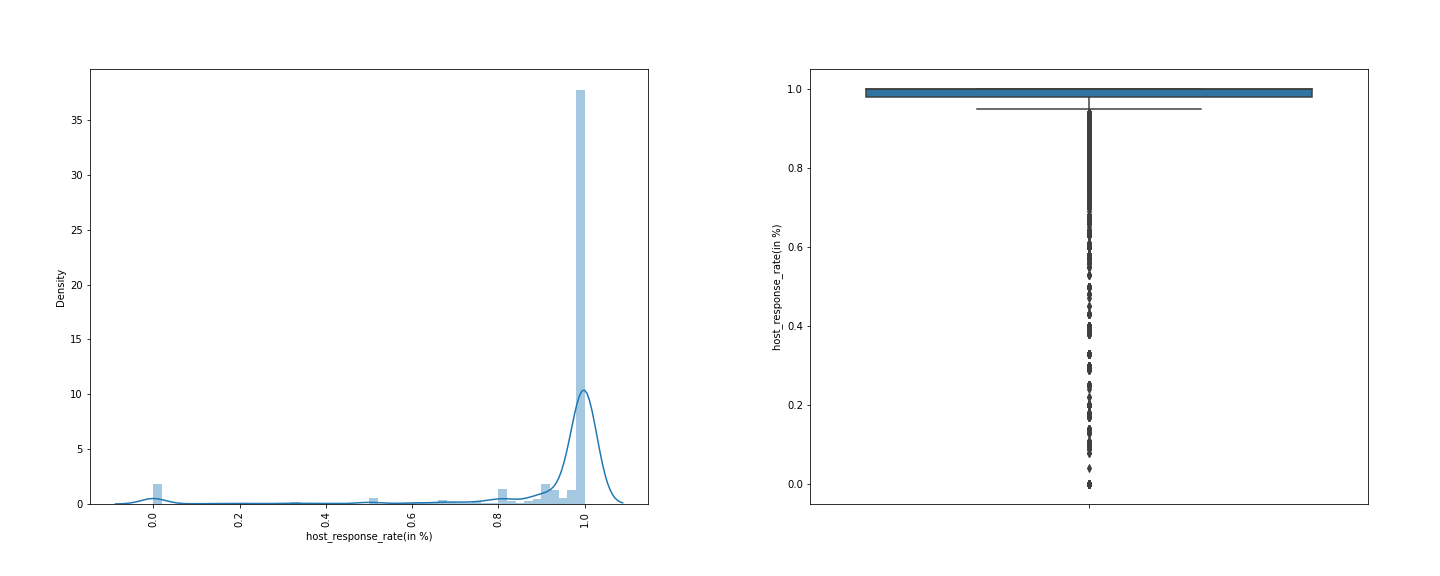
**CHECK FOR OUTLIERS:**

* Data has outliers present in each of the numerical columns. For making the base model, we do not perform any outlier treatment and retain all the rows present in the data.

**UNIVARIATE ANALYSIS:**

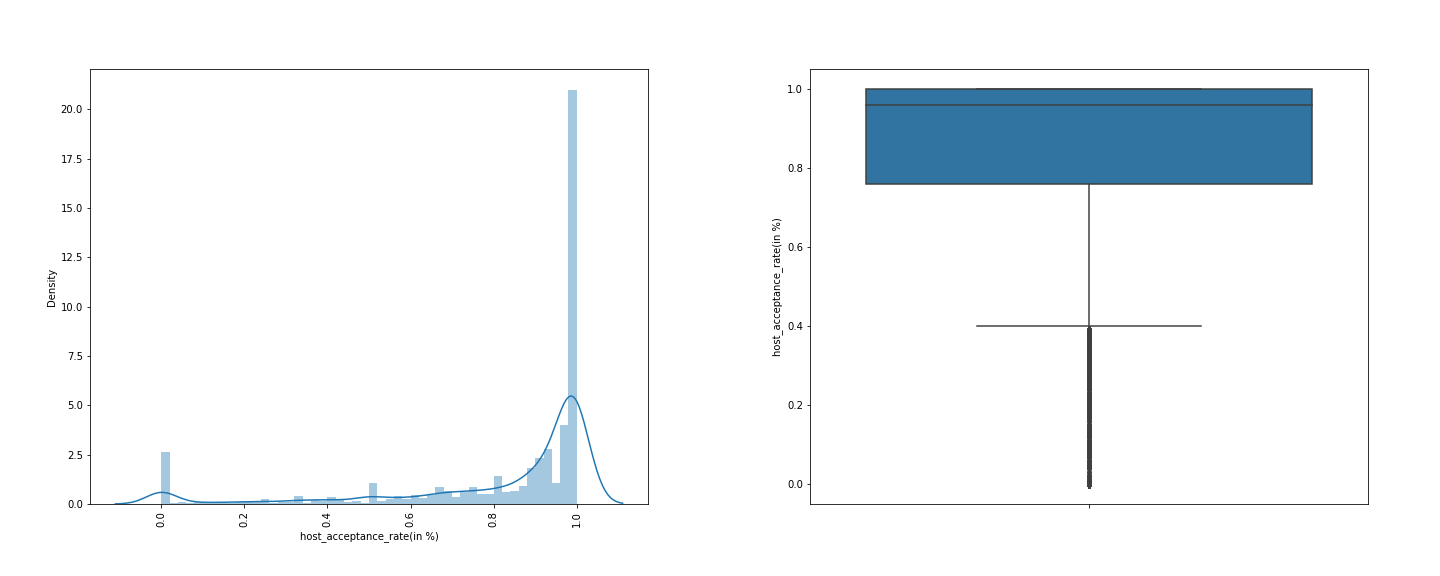
**1.1. Numerical:**

**a. Host\_response\_rate (in %)**



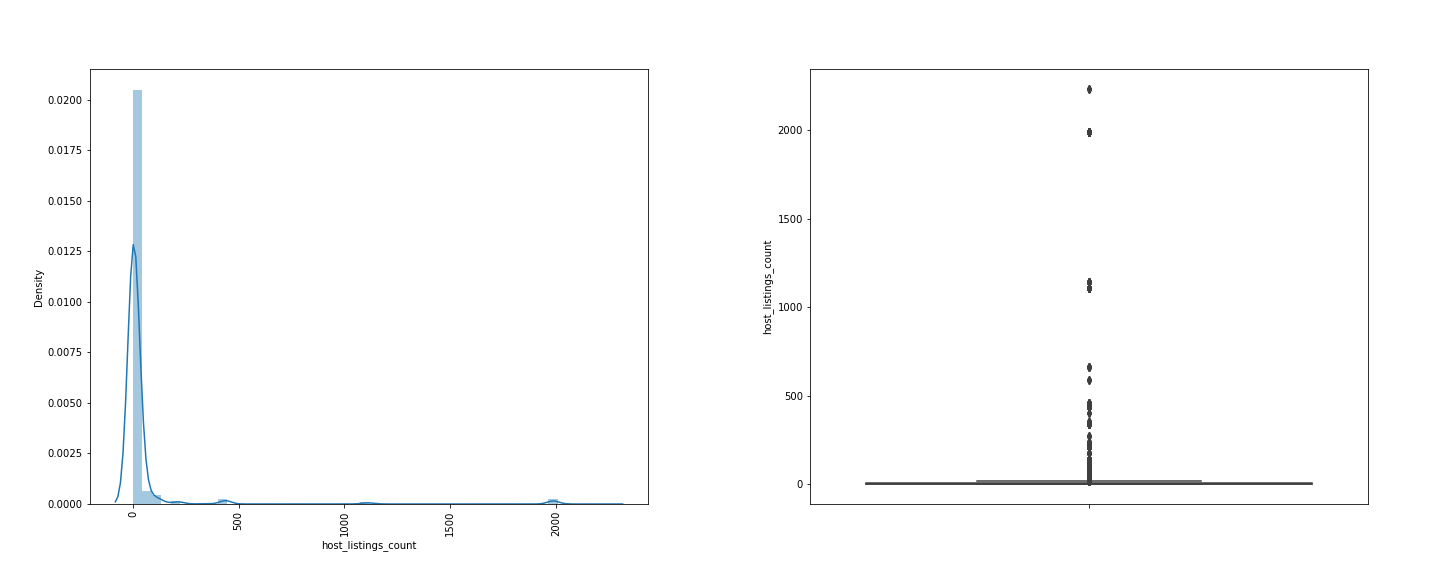
* Host response time is left skewed.
* It is leptokurtic, and has wide tail
* IQR lies at 1.0
* Outliers are present.

**b. Host\_acceptance\_rate(in%)**



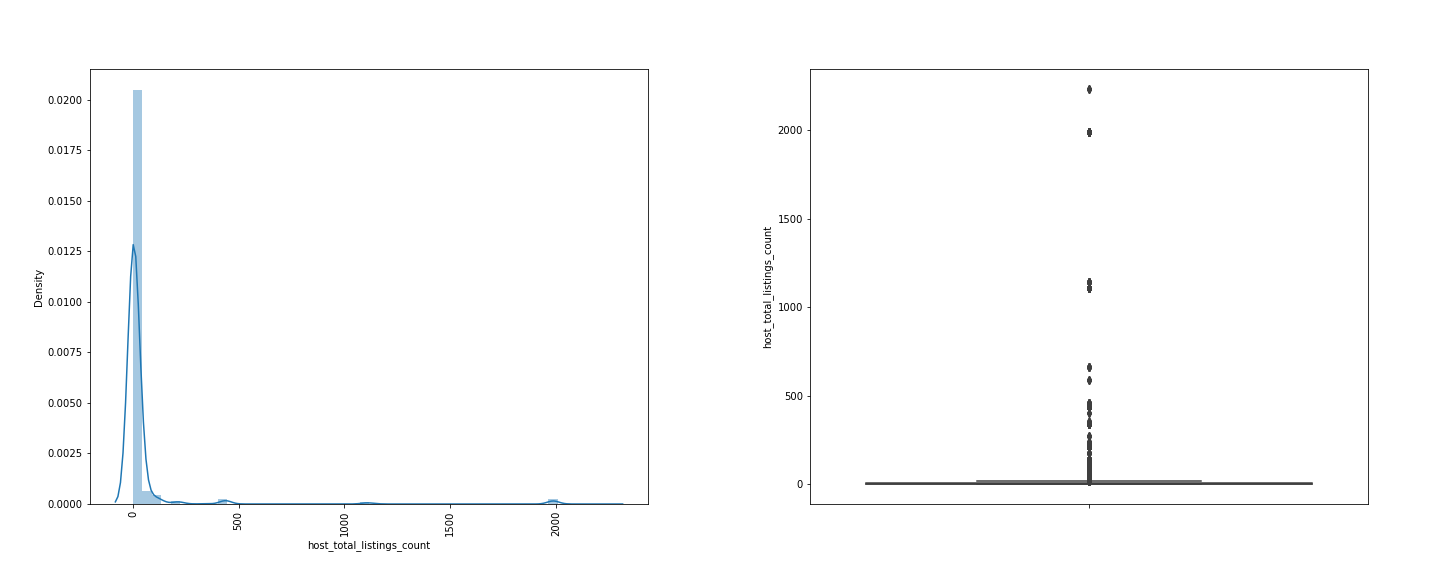
* Host acceptance rate time is left skewed.
* It is leptokurtic, and has wide tail
* IQR lies from 1.0 to 0.8.
* Outliers are present

**c. Host\_listings\_count**



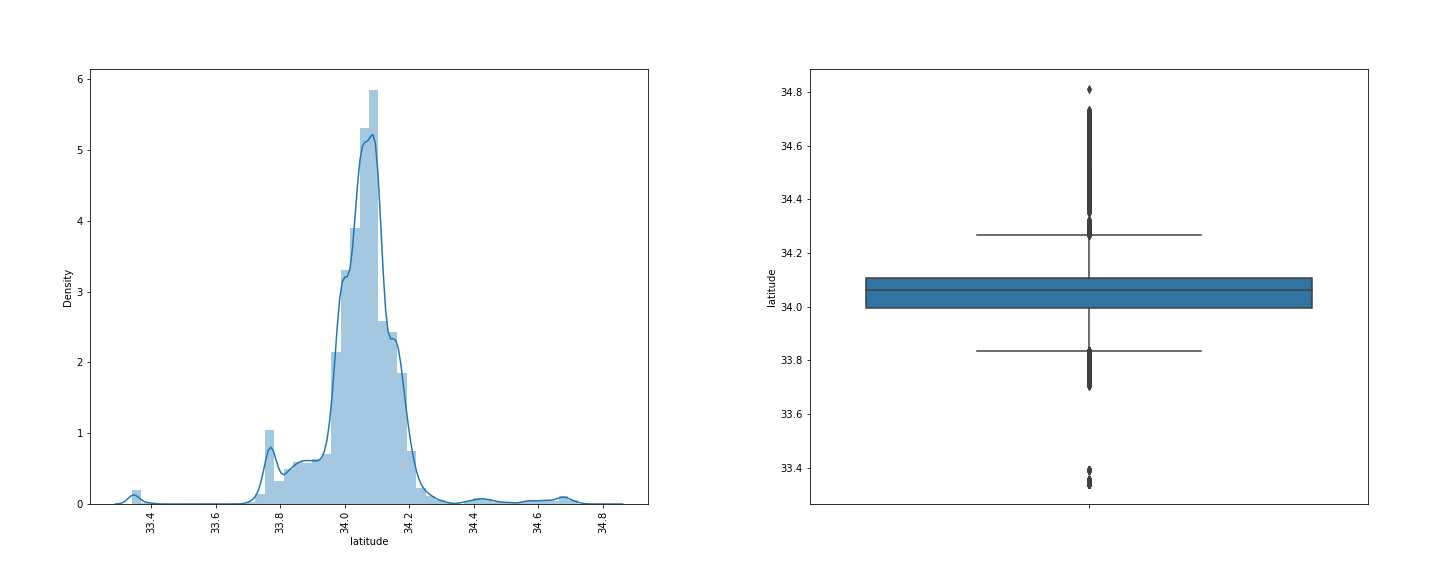
* Host listings count is right skewed.
* It is leptokurtic, and has wide tail
* IQR lies at 0.
* Outliers are present

**d. Host\_total\_listings\_count**



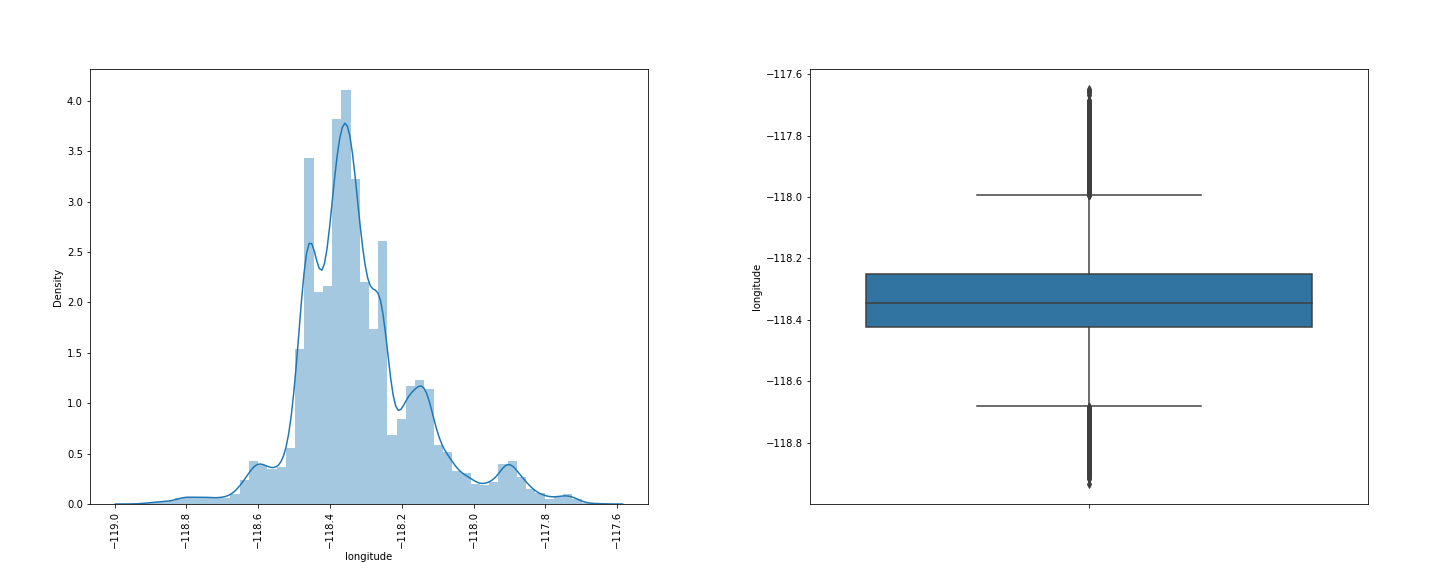
* Host total listings count is right skewed.
* it is leptokurtic, and has wide tail
* IQR lies at 0.
* Outliers are present.

1. **Latitude**



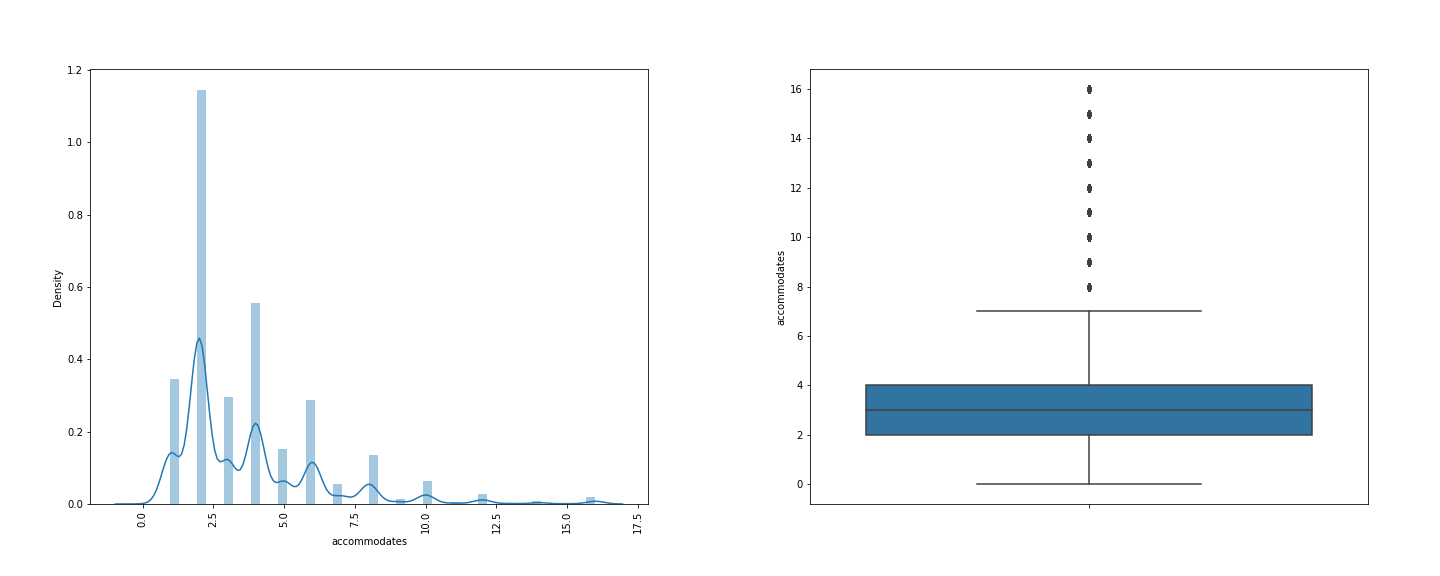
* Latitude has normal distribution.
* It is leptokurtic, and has wide tail
* IQR lies from 33.8 to 34.3.
* Outliers are present

1. **Longtitude**



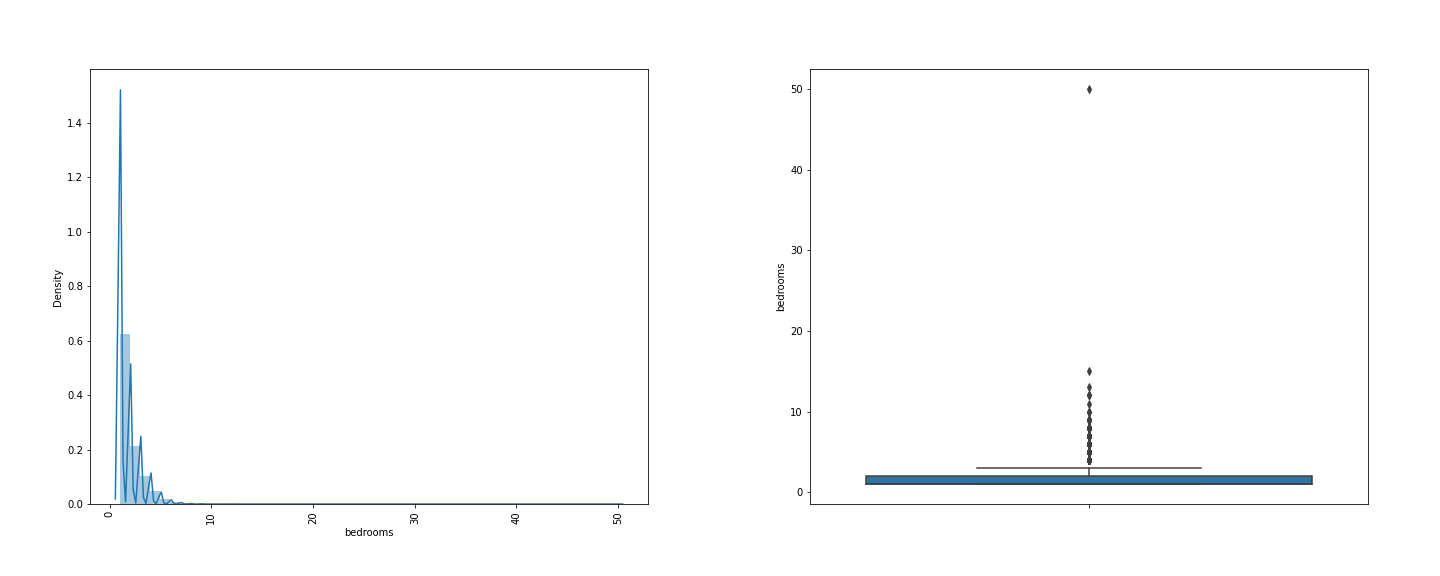
* Longitude has right skew.
* It is leptokurtic, and has wide tail
* IQR lies from -118.6 to 118.
* Outliers are present.

1. **Accommodates**

****

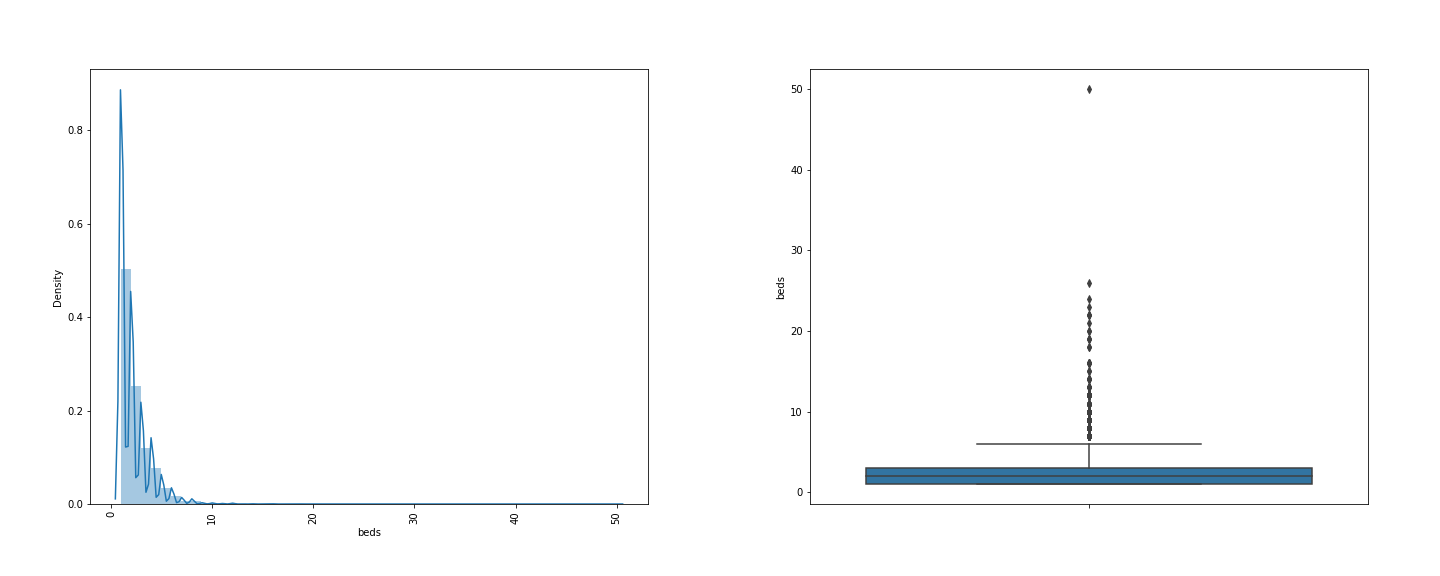
* Accommodates has right skew.
* It is leptokurtic, and has wide tail
* IQR lies from 0 to 7.
* Outliers are present.

1. **Bedrooms**



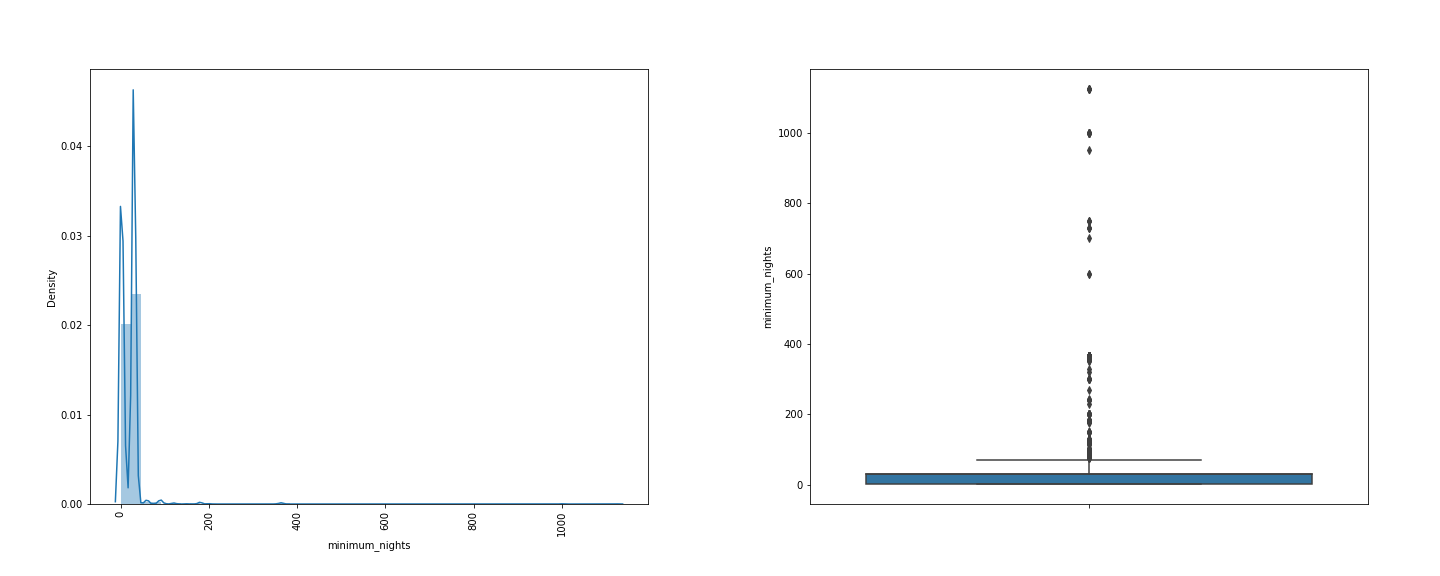
* Bedrooms has right skew.
* It is leptokurtic, and has wide tail
* IQR lies from 0 to 3.
* Outliers are present.

1. **Beds**



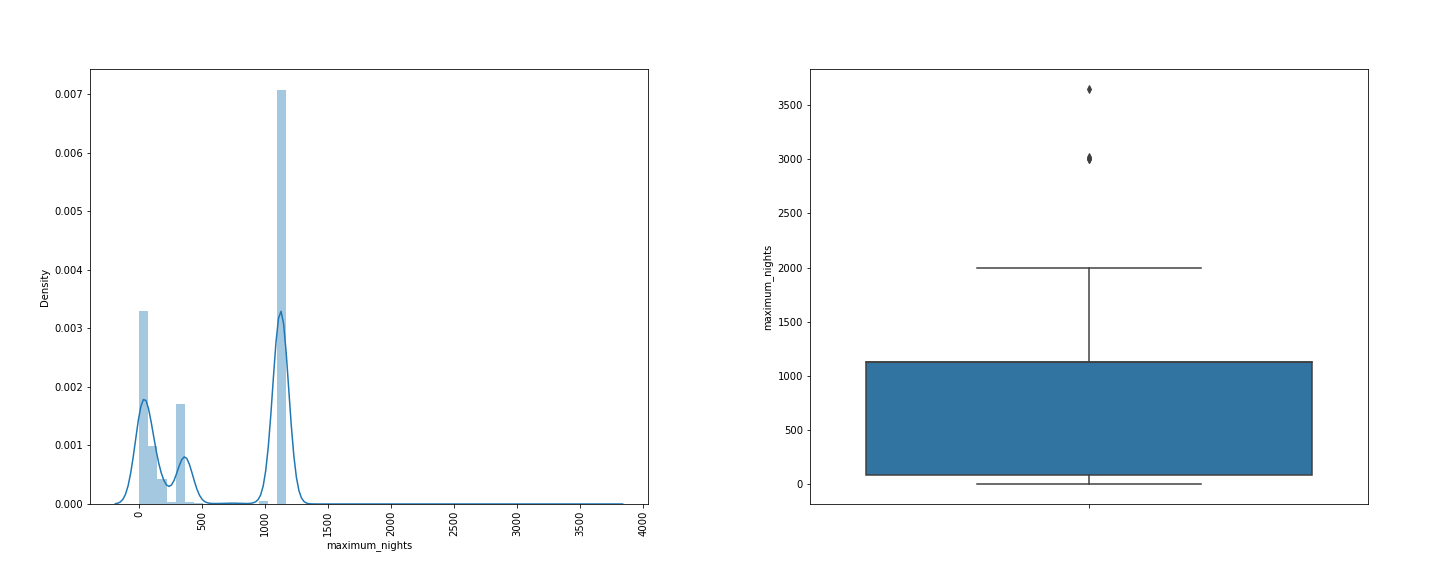
* Beds has right skew.
* It is leptokurtic, and has wide tail
* IQR lies from 0 to 7.
* Outliers are present.

1. **Minimum\_nights**



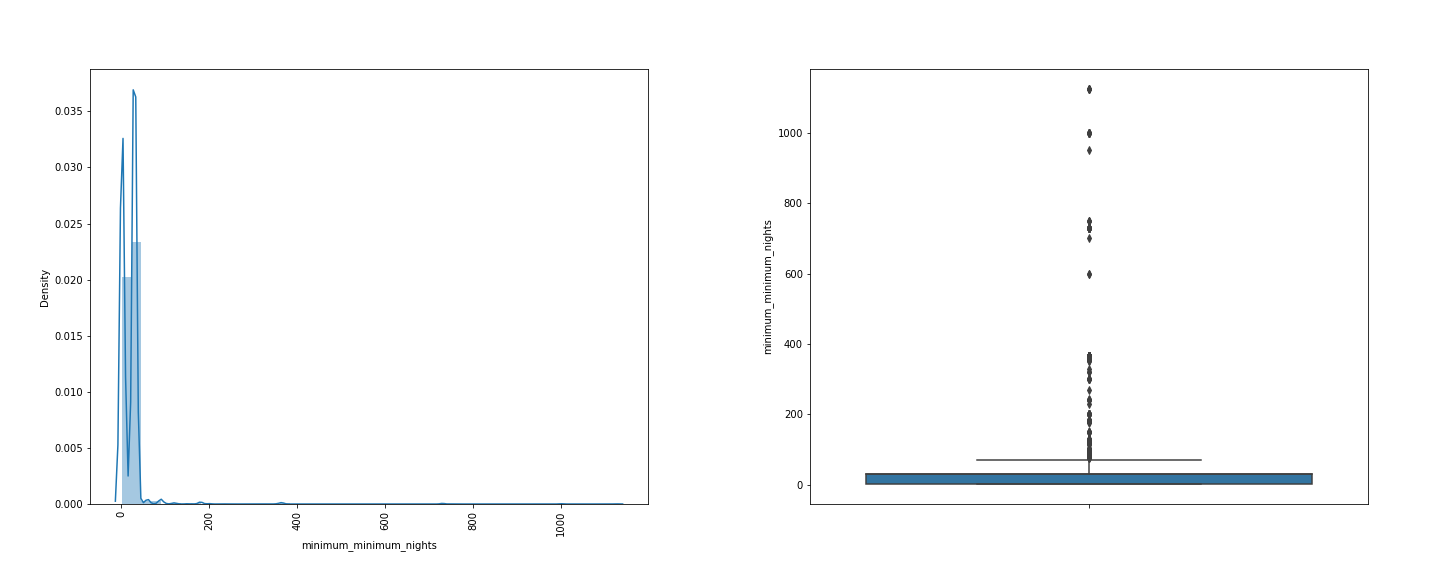
* Beds has right skew.
* It is leptokurtic, and has wide tail
* IQR lies from 0 to 100.
* Outliers are present.

1. **Maximum\_nights**



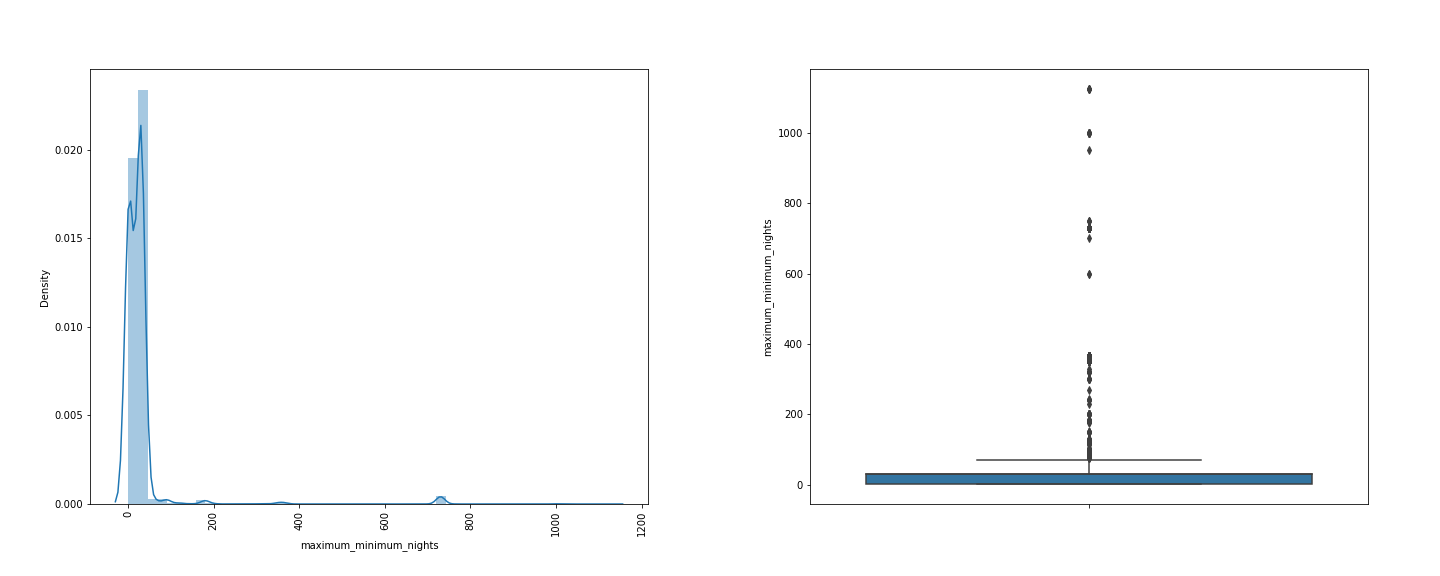
* Maximum nights is normal distributed.
* It is platykurtic, and has wide tail
* IQR lies from 0 to 1100.
* Only few outliers are present.

1. **Minimum\_minimum\_nights**



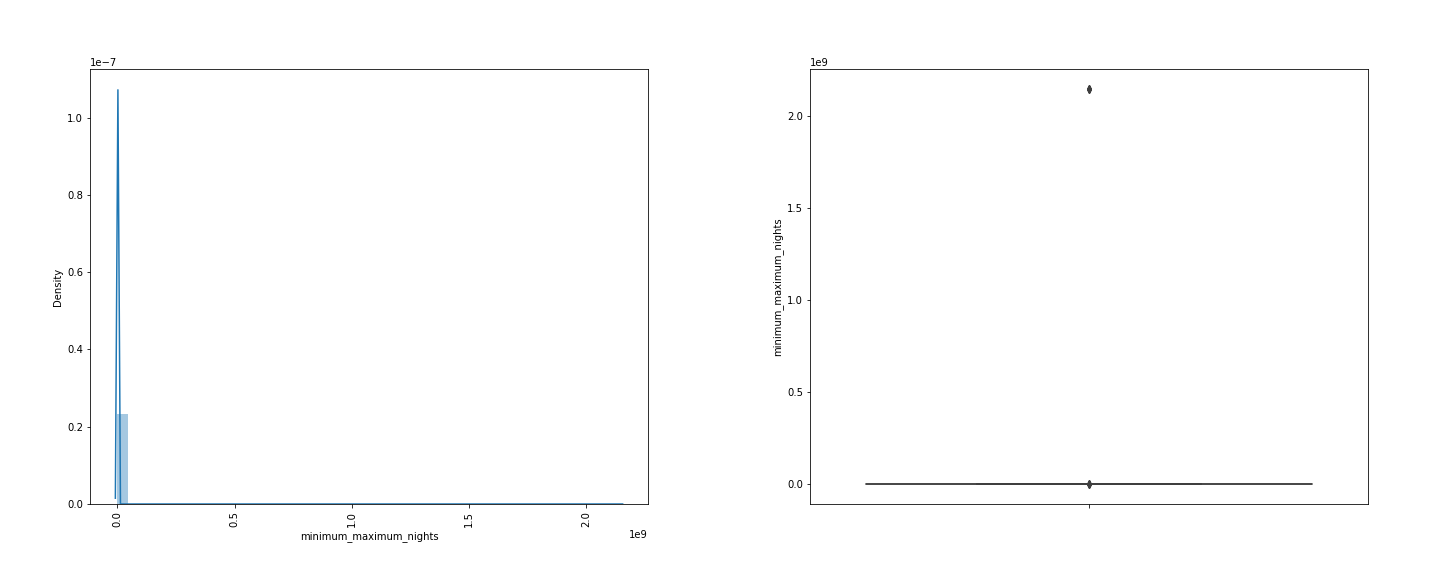
* Minimum minimum nights is right skewed.
* It is leptokurtic, and has narrow tailed.
* IQR lies from 0 to 100.
* Outliers are present.

1. **Maximum\_minimum\_nights**



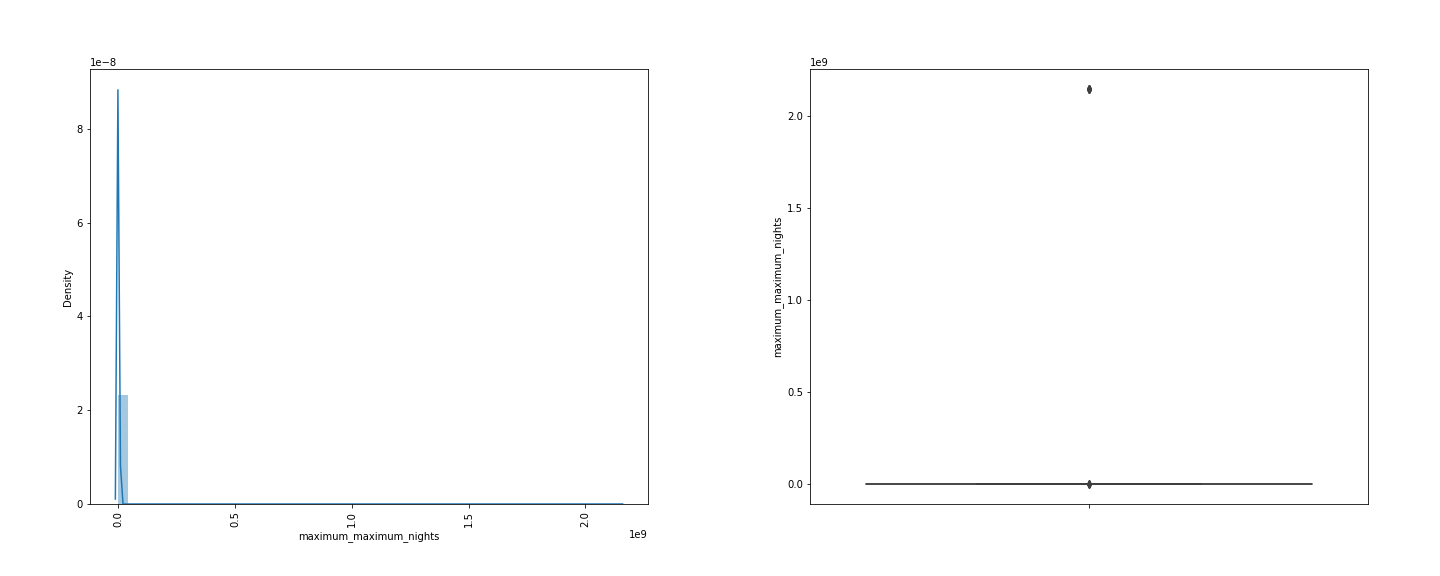
* Maximum minimum nights is right skewed.
* It is leptokurtic, and has wide tail
* IQR lies from 0 to 100.
* Outliers are present.

1. **Minimum\_maximum\_nights**



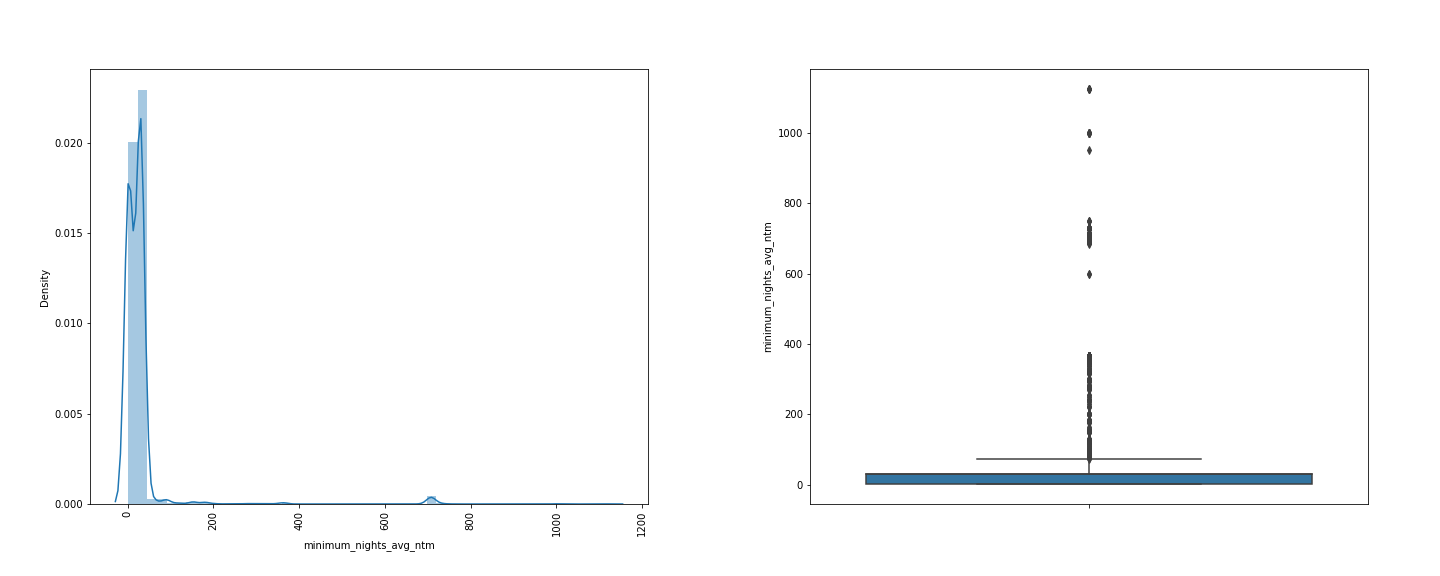
* Minimum maximum nights is right skewed.
* It is leptokurtic, and has narrow tailed.
* IQR lies at 0.
* Only few outliers are present.

1. **Maximum\_maximum\_nights**

****

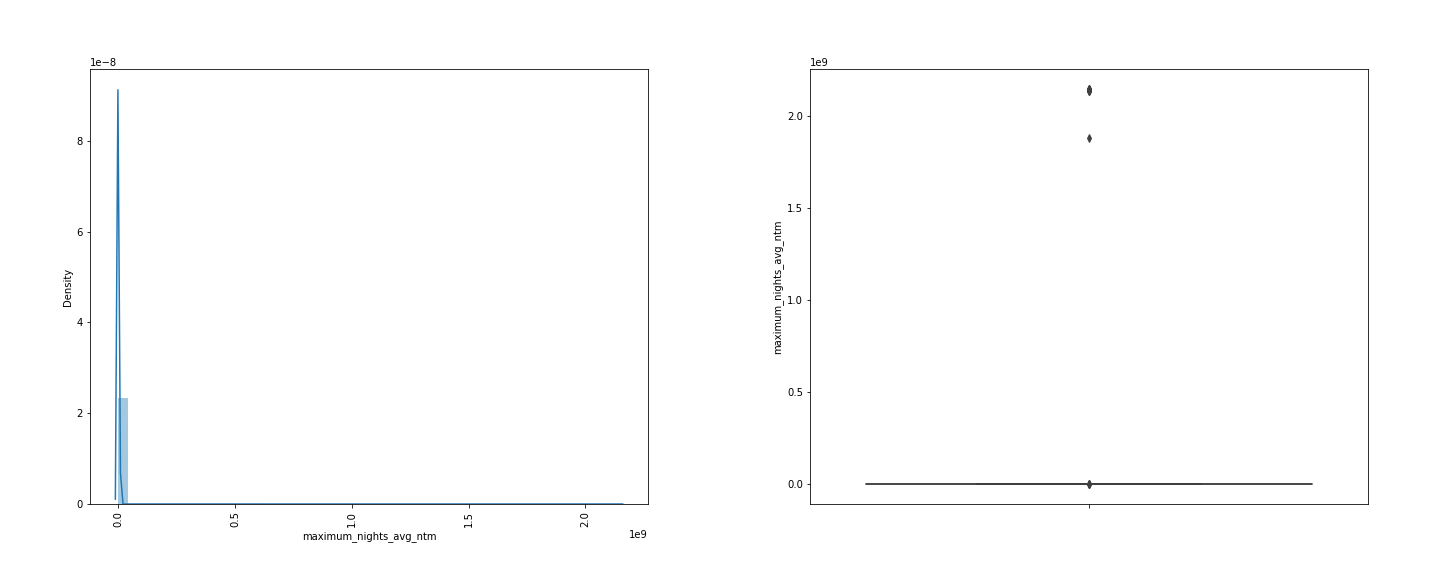
* Maximum maximum nights is right skewed.
* It is leptokurtic, and narrow tailed.
* IQR lies at 0.
* Only few outliers are present.

1. **Minimum\_nights\_avg\_ntm**



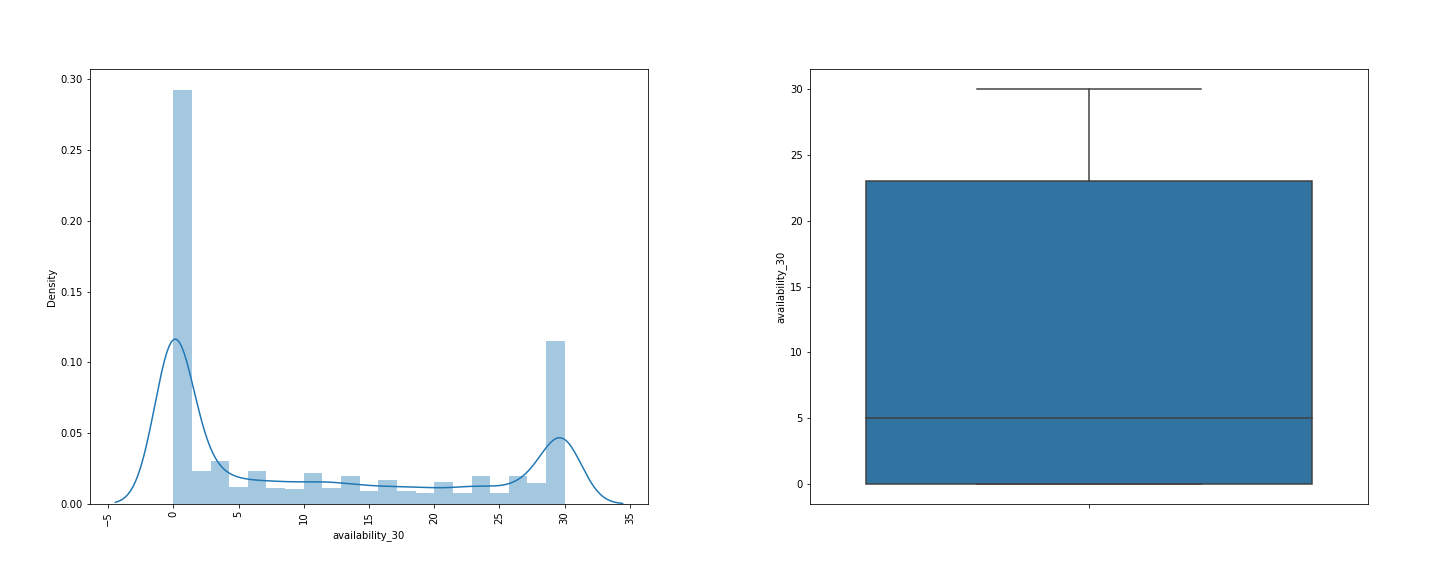
* Minimum nights avg ntm is right skewed.
* It is leptokurtic, and wide tailed.
* IQR lies from 0 to 100.
* Outliers are present.

1. **Maximum\_nights\_avg\_ntm**



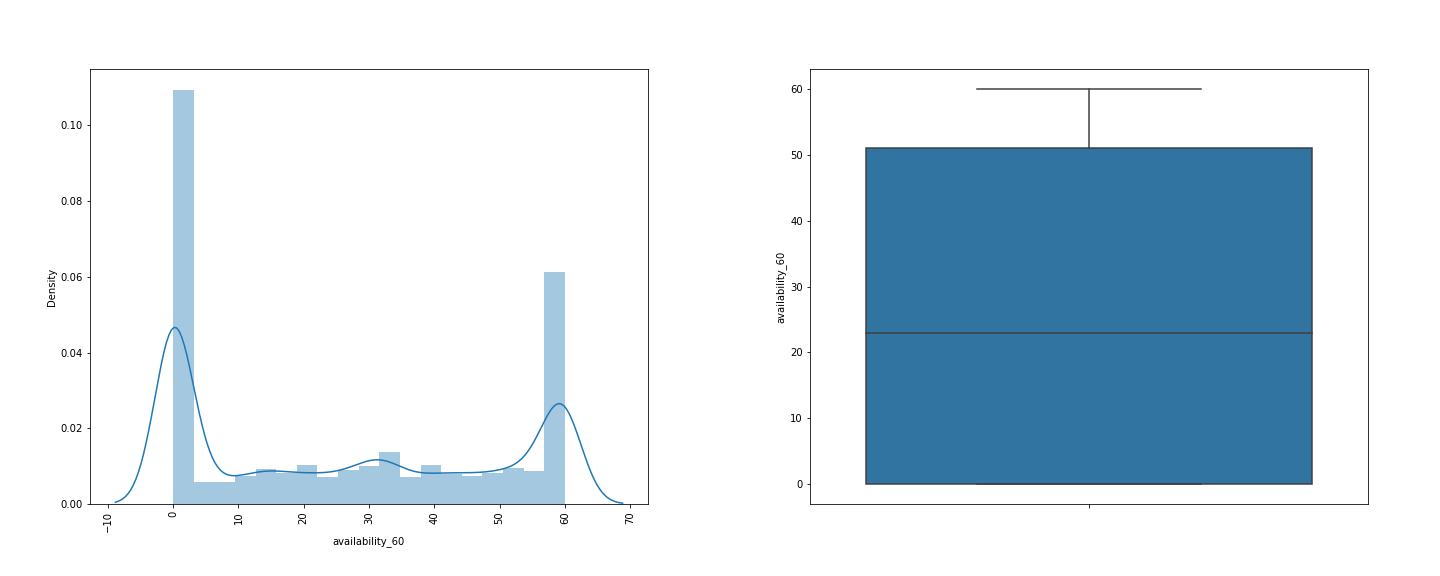
* Maximum nights avg ntm is right skewed.
* It is leptokurtic, and narrow tailed.
* IQR lies from 0.
* Only few outliers are present.

1. **Availability\_30**



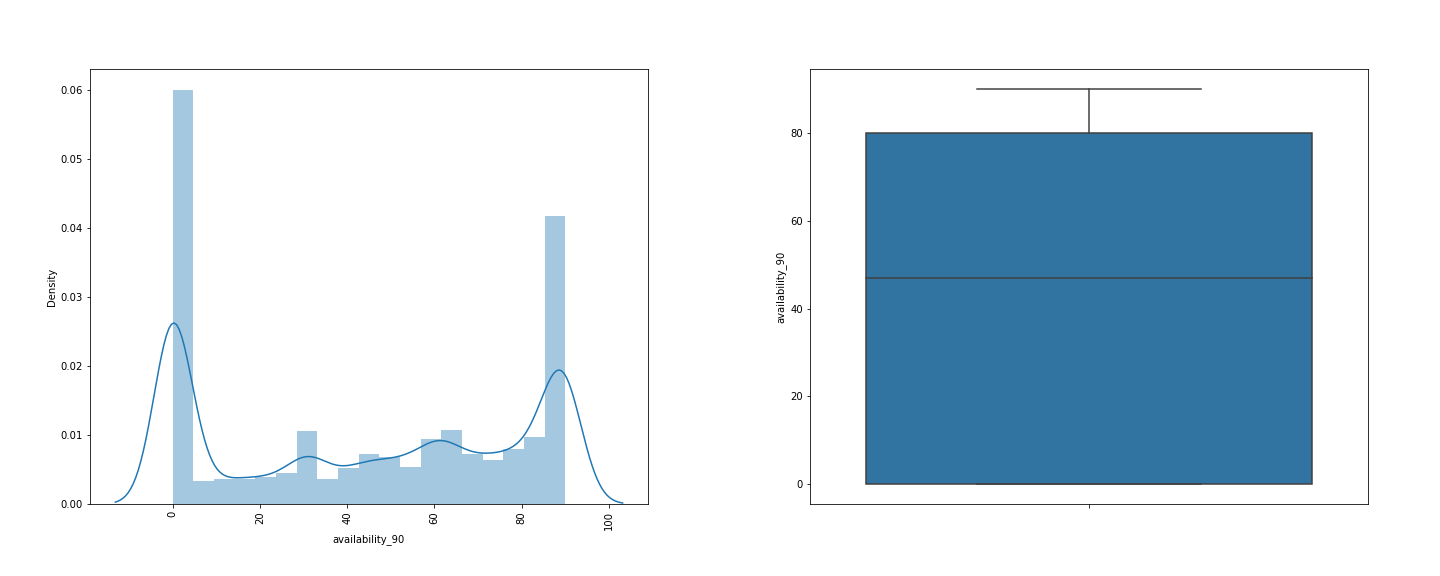
* Availability\_30 is right skewed.
* It is Platykurtic, and wide tailed.
* IQR lies from 0 to 24.
* No outliers are present.

1. **Availability\_60**



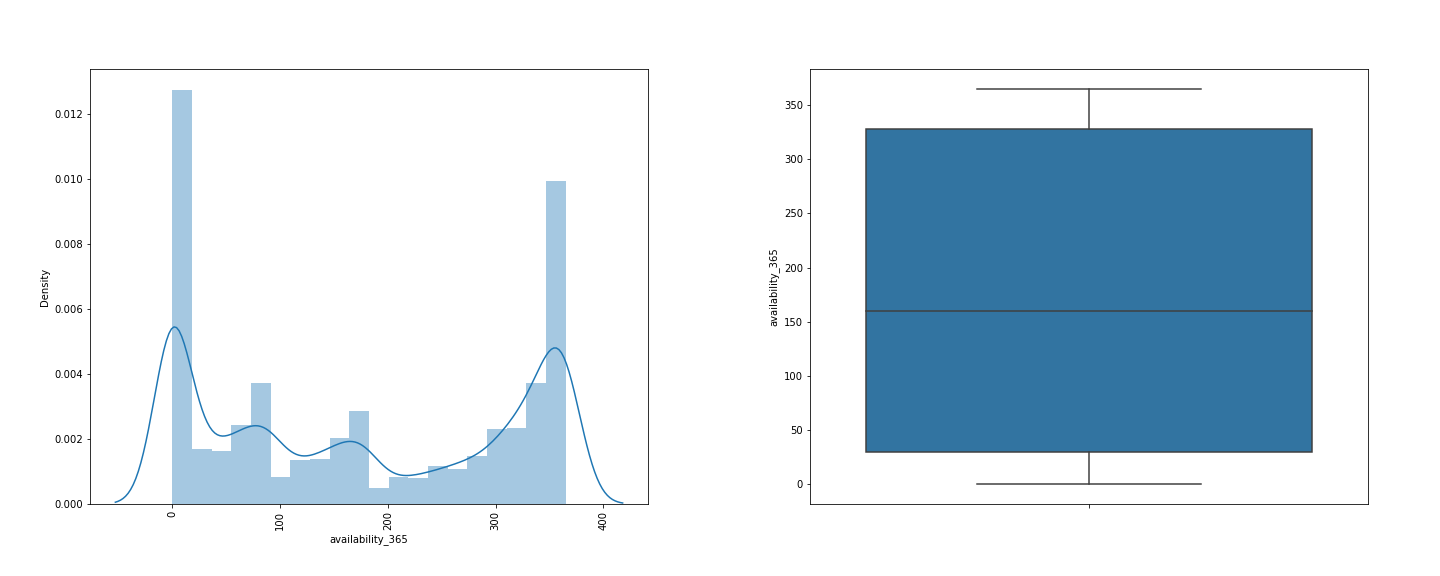
* Availability\_60 is normally distributed.
* It is platykurtic, and wide tailed.
* IQR lies from 0 to 50.
* No outliers are present.

1. **Availability\_90**



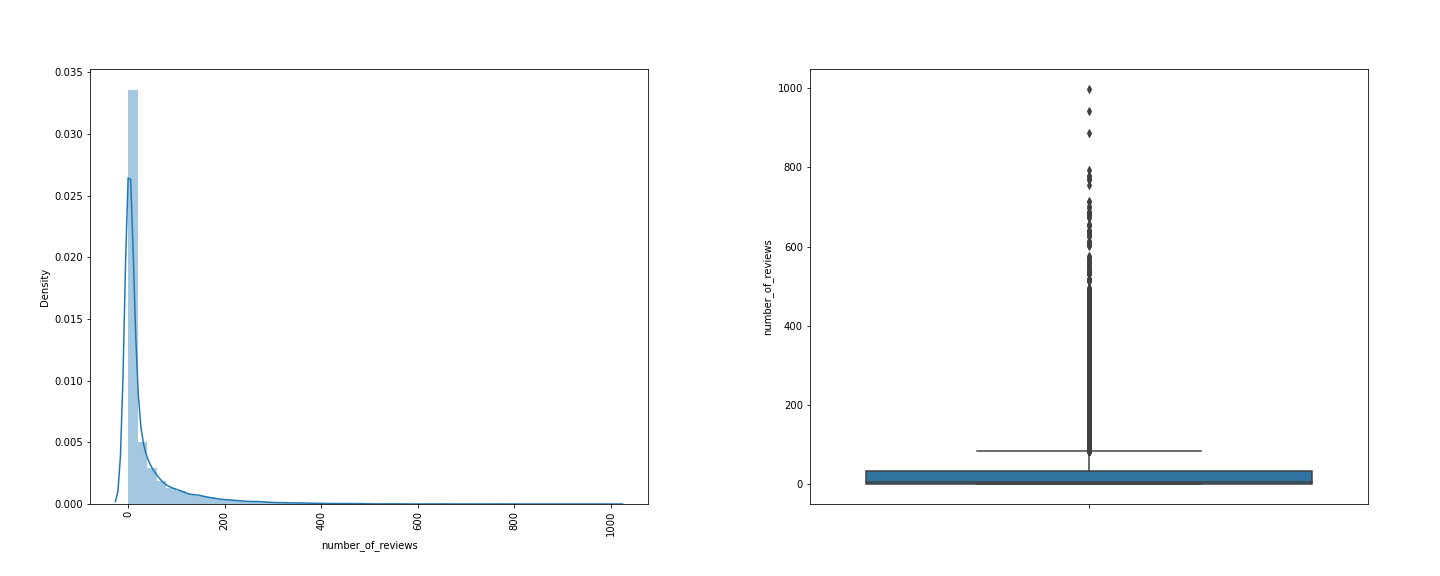
* Availability\_60 is normally distributed.
* It is platykurtic, and wide tailed.
* IQR lies from 0 to 80.
* No outliers are present.

1. **Availability\_365**



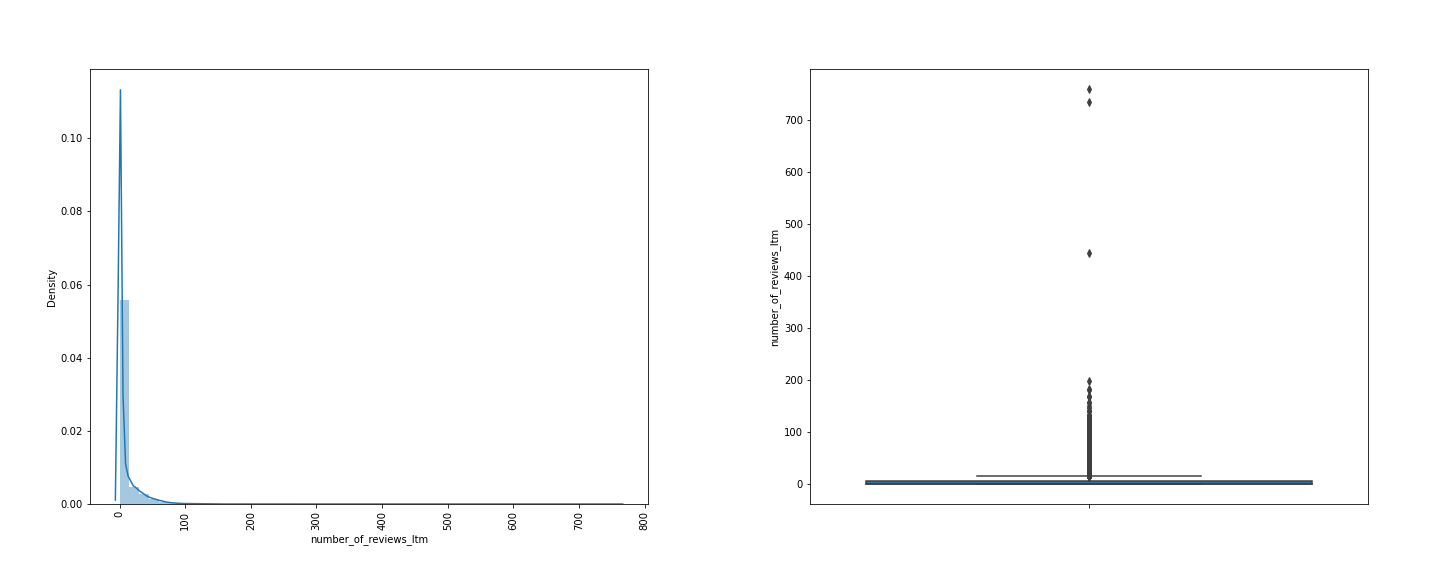
* Availability\_365 is normally distributed.
* It is platykurtic, and wide tailed.
* IQR lies from 0 to 350.
* No outliers are present.

1. **Number\_of\_reviews**

****

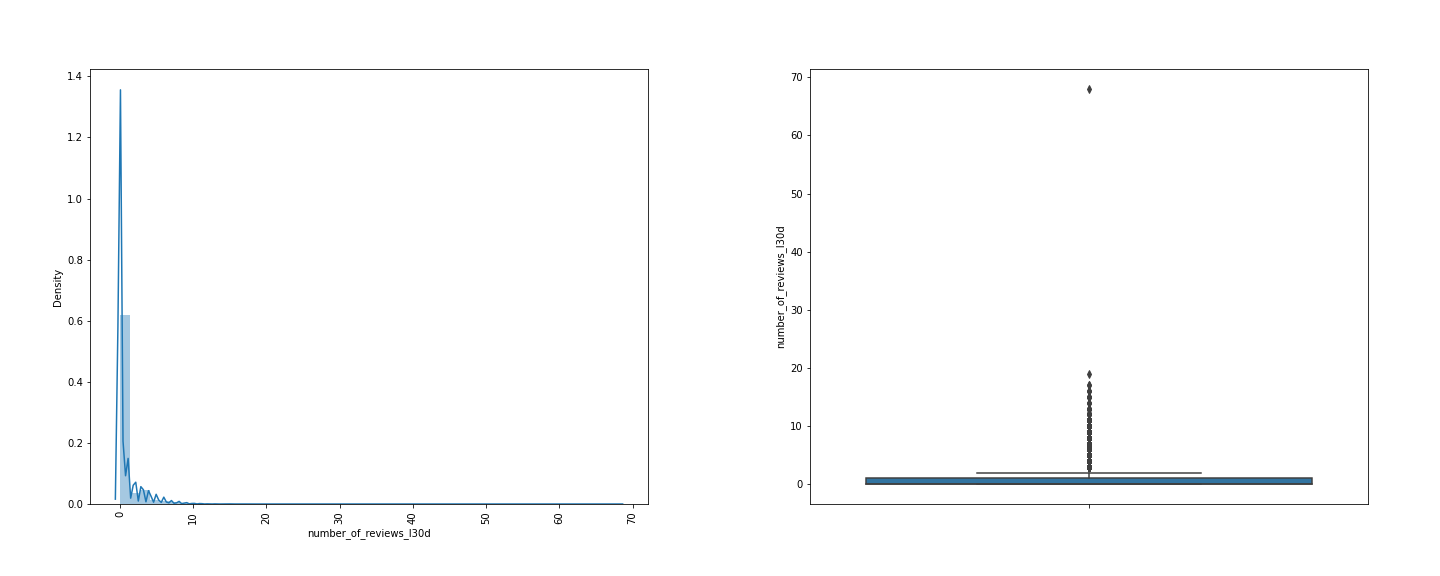
* Number of reviews is rightly skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 0 to 100.
* Outliers are present.

1. **Number\_of\_reviews\_itm**

****

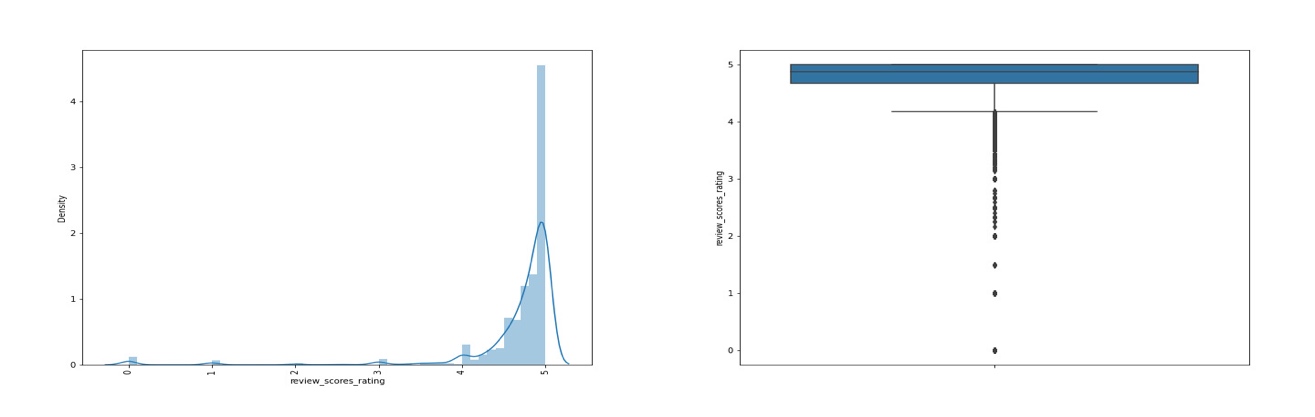
* Number of reviews itm is rightly skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies from 0 to 10.
* Outliers are present.

1. **Number\_of\_reviews\_I30d**

****

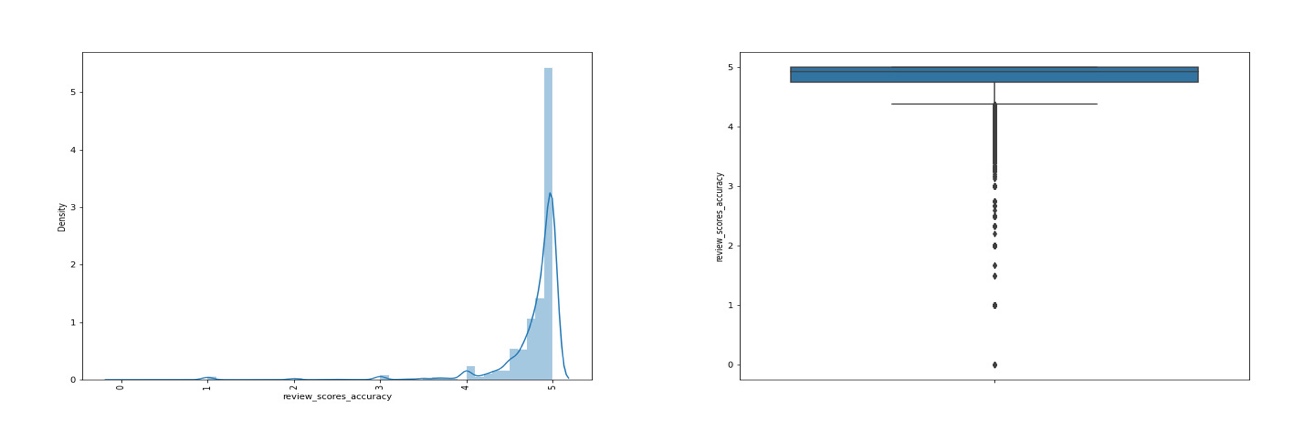
* Number of reviews itm is rightly skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies from 0 to 10.
* Outliers are present.

1. **Review\_scores\_rating**



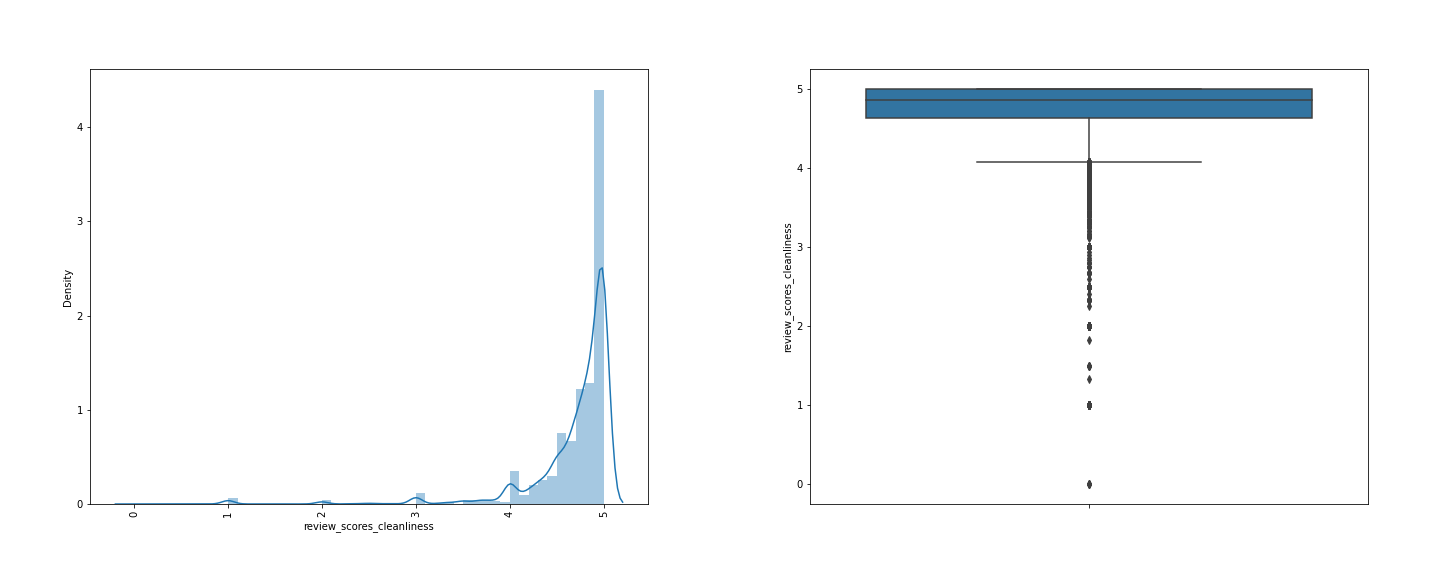
* Review scores rating is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Review\_scores\_accuracy**



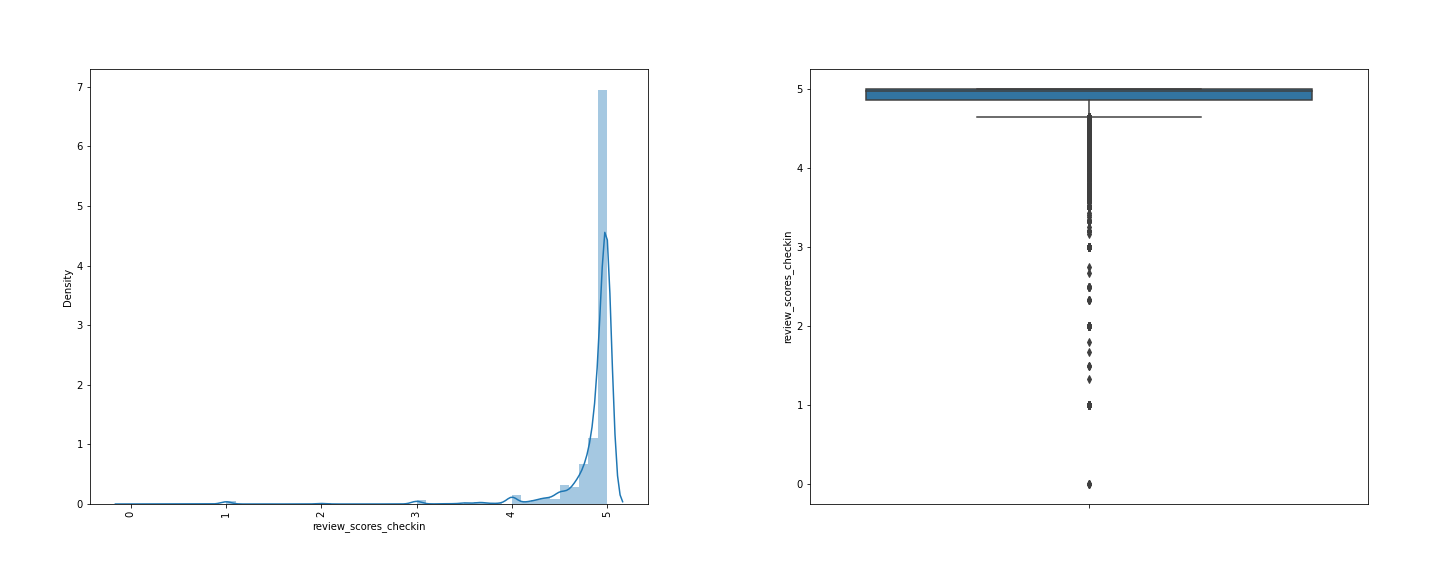
* Review scores accuracy is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Review\_scores\_cleanliness**



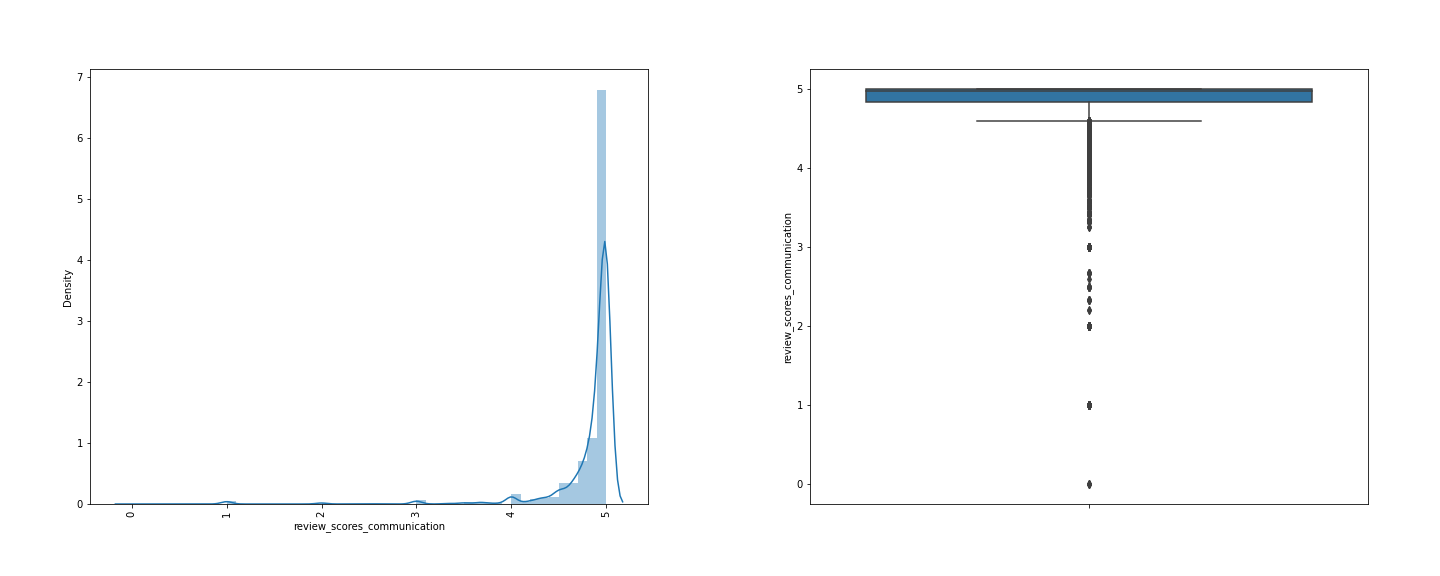
* Review scores cleanliness is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Review\_scores\_checkin**



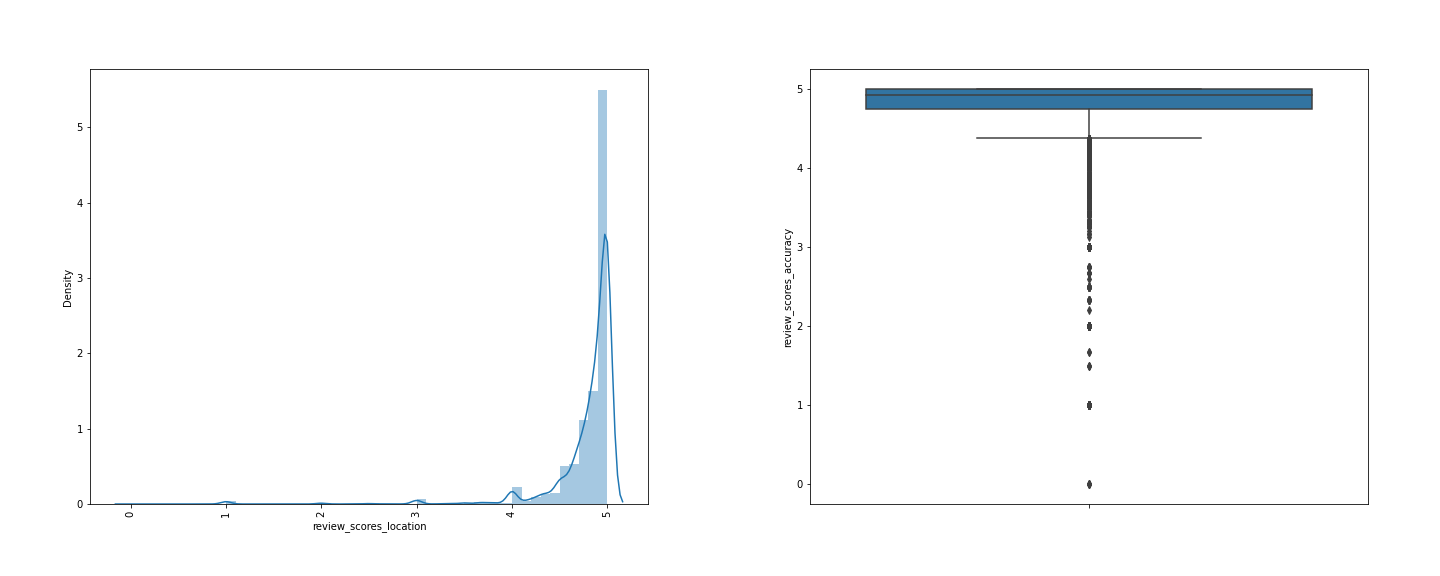
* Review scores checkin is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Review\_scores\_communication**



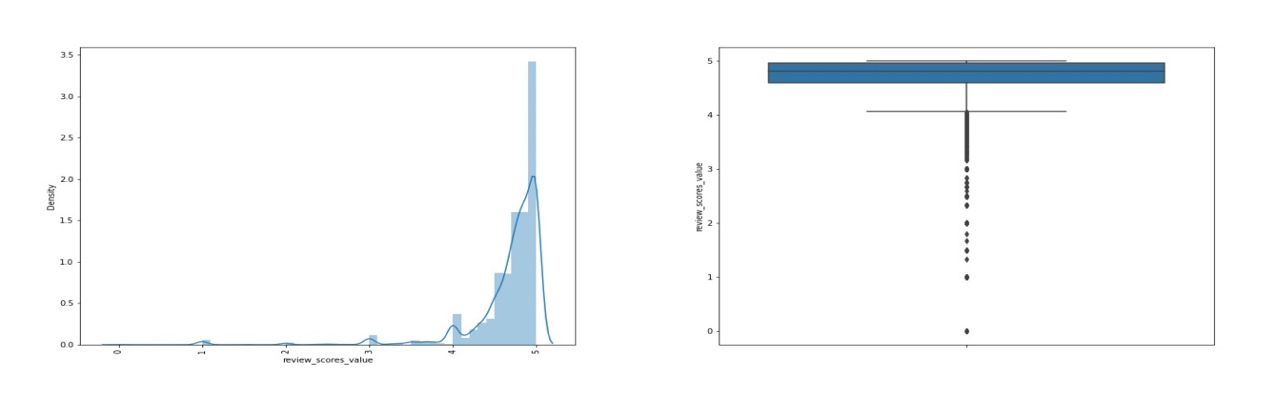
* Review scores communication is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Review\_scores\_location**



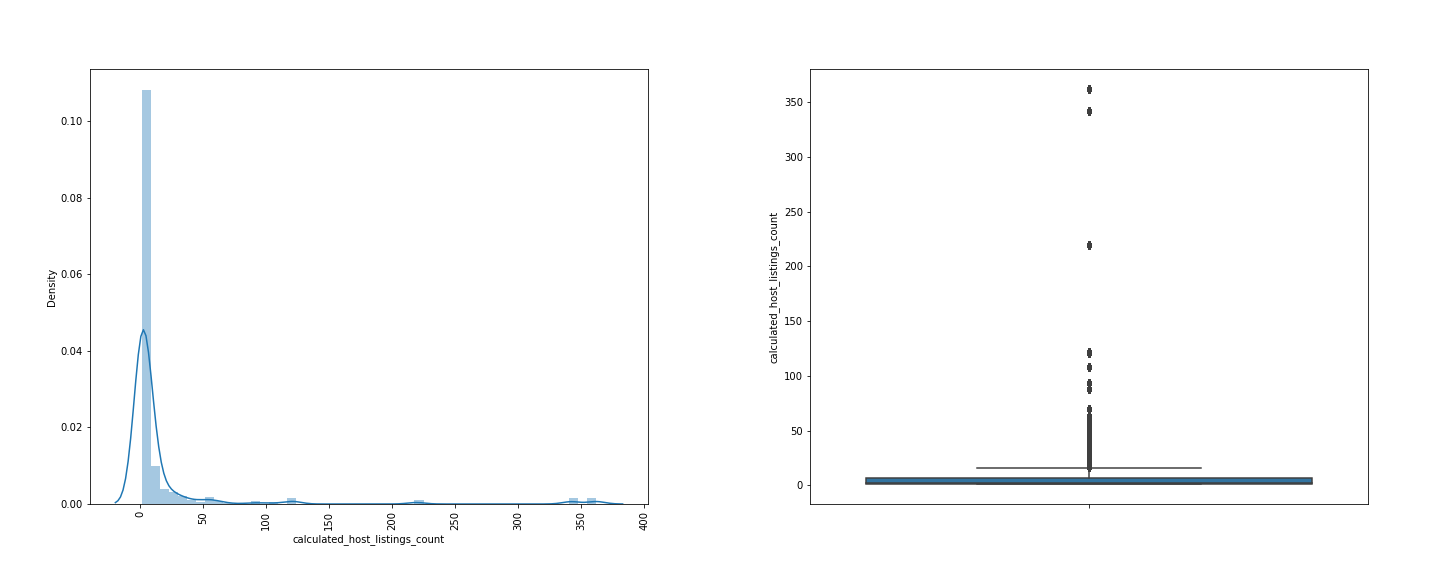
* Review scores location is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Review\_scores\_value**



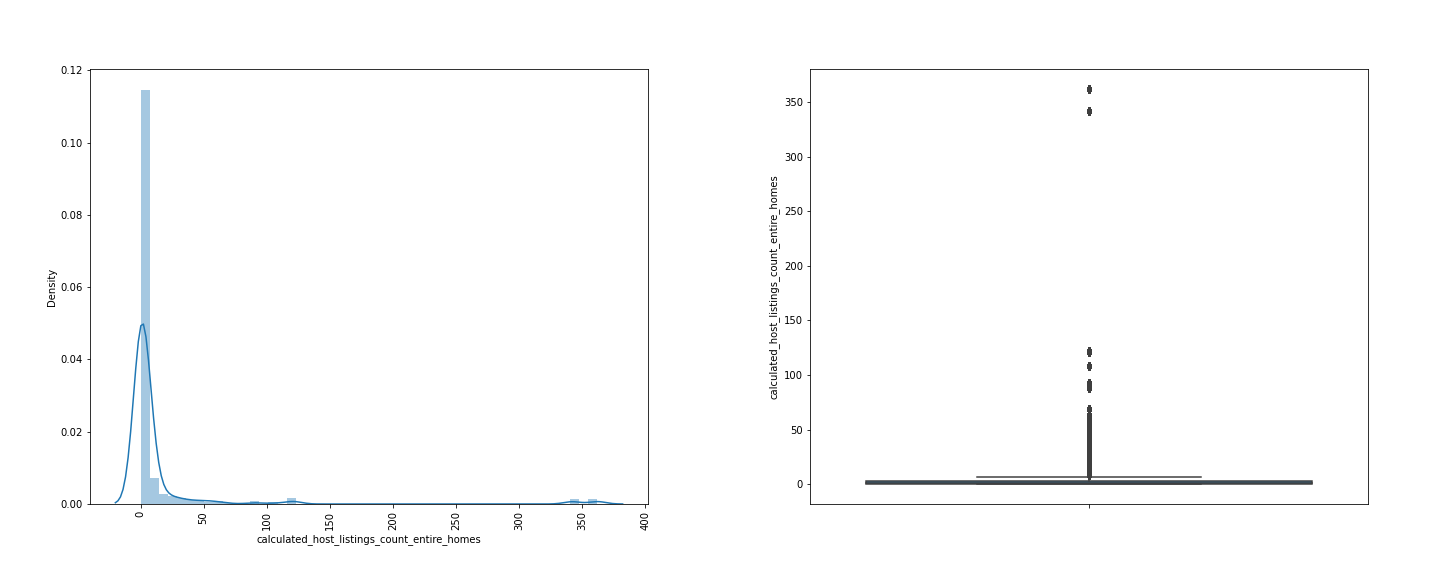
* Review scores value is left skewed.
* It is Leptokurtic, and wide tailed.
* IQR lies from 5 to 4.
* Outliers are present.

1. **Calculated\_host\_listings\_count**



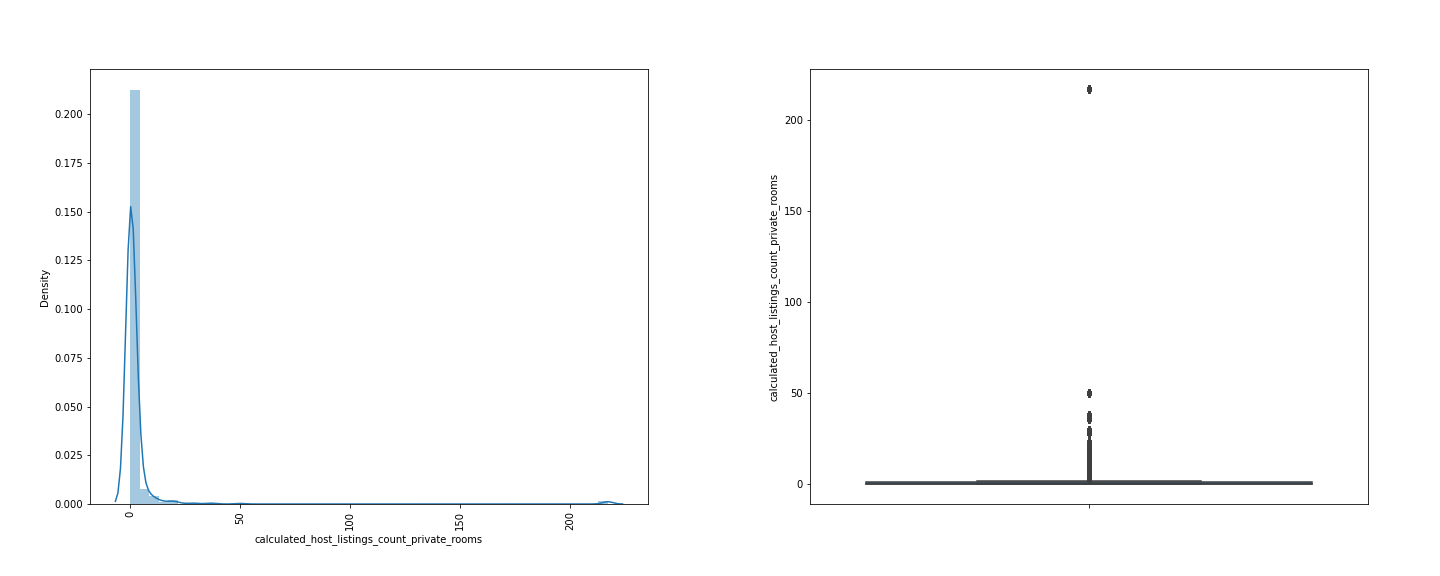
* Calculated host listings count is right skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies from 0 to 20.
* Outliers are present.

1. **Calculated\_host\_listings\_count\_entire\_homes**



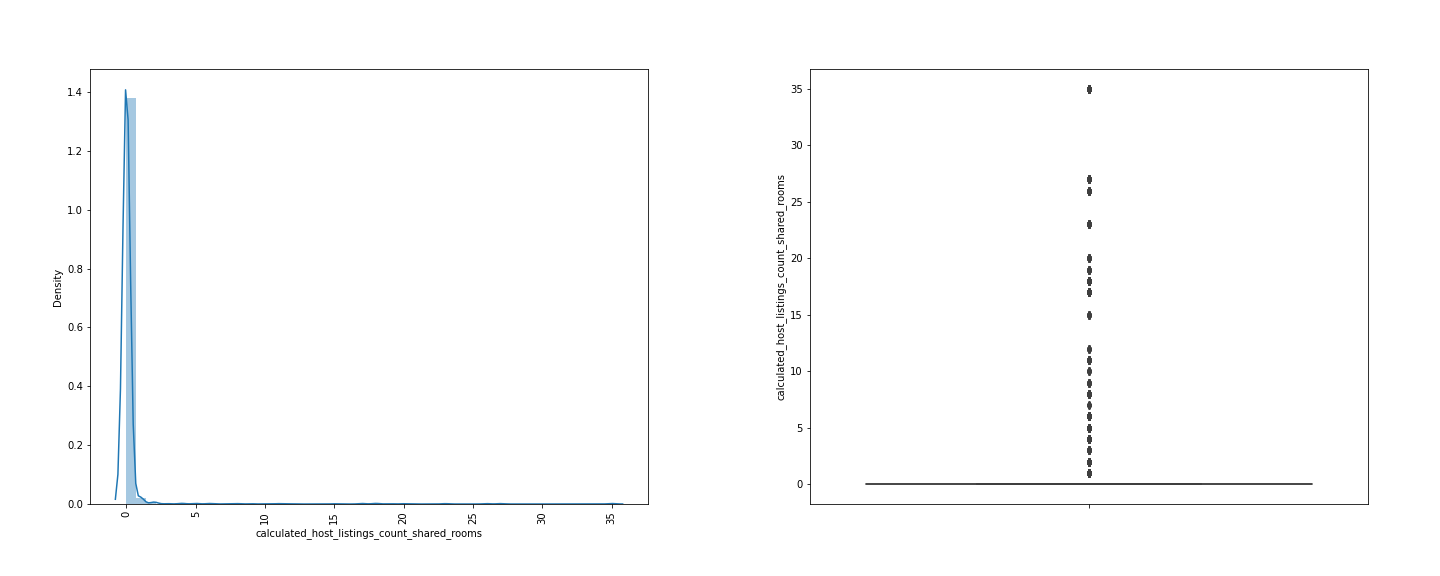
* Calculated host listings count entire homes is right skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies from 0 to 10.
* Outliers are present.

1. **Calculated\_host\_listings\_count\_private\_rooms**



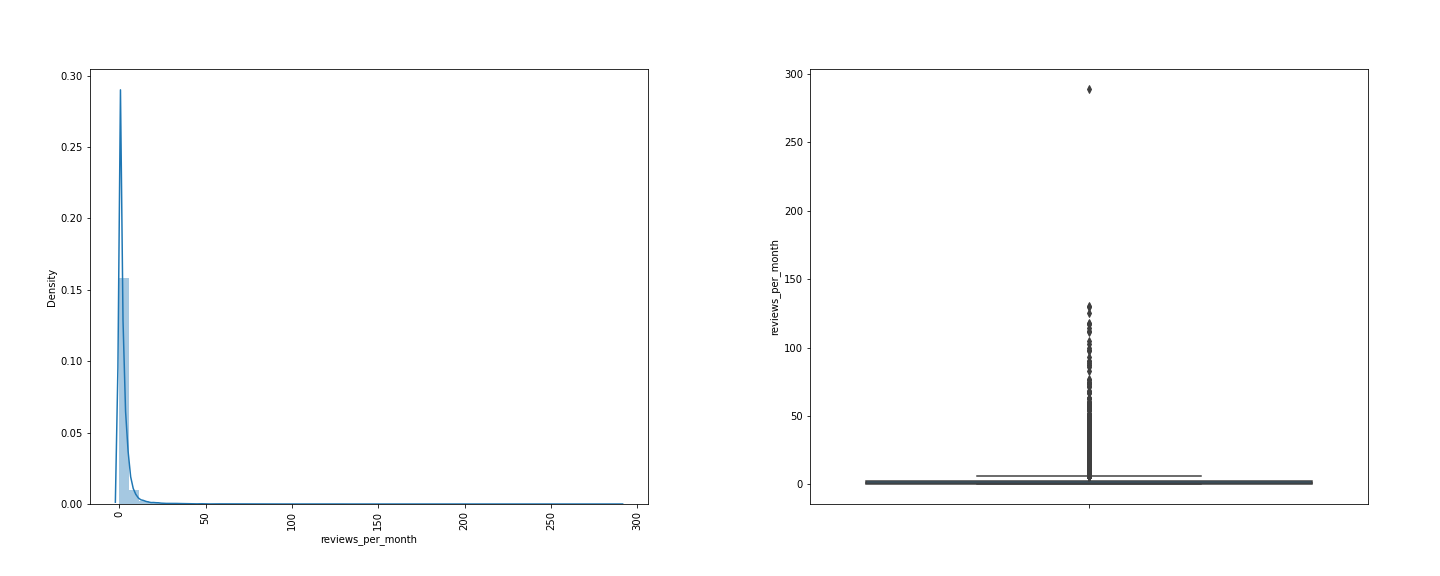
* Calculated host listings count entire homes is right skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies at 0 .
* Outliers are present.

1. **Calculated\_host\_listings\_count\_shared\_rooms**



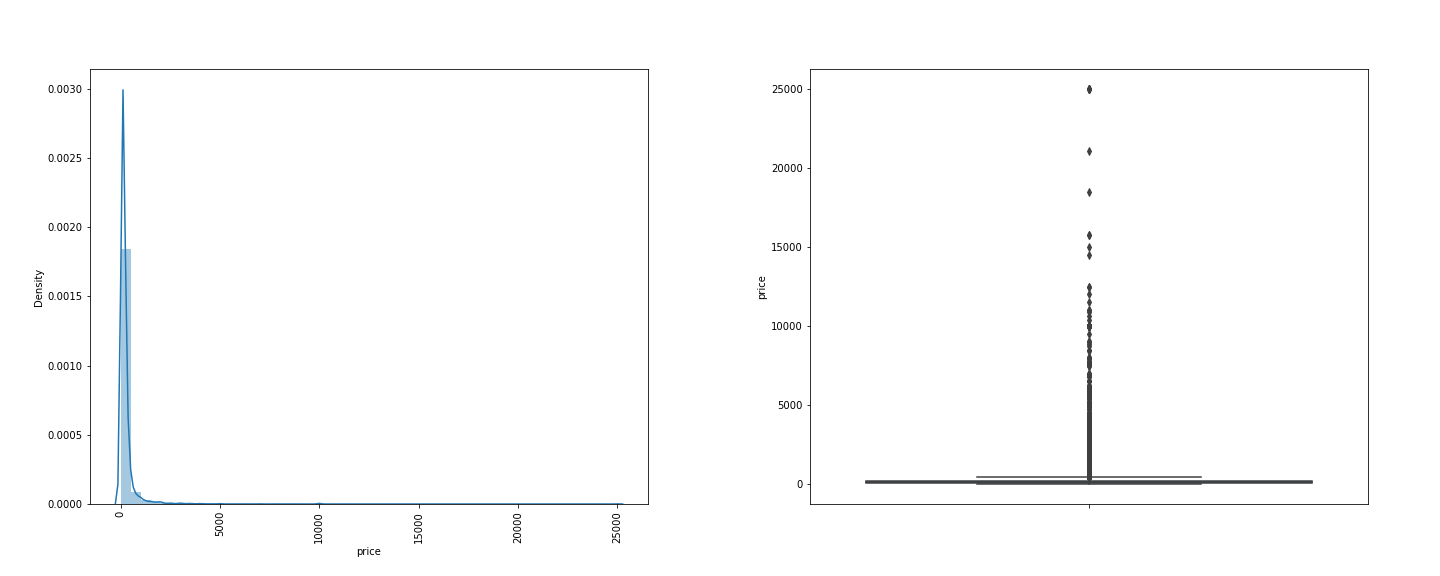
* Calculated host listings count shared rooms is right skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies at 0 .
* Outliers are present.

1. **Reviews\_per\_month**



* Reviews per month is right skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies from 0 to 10 .
* Outliers are present.

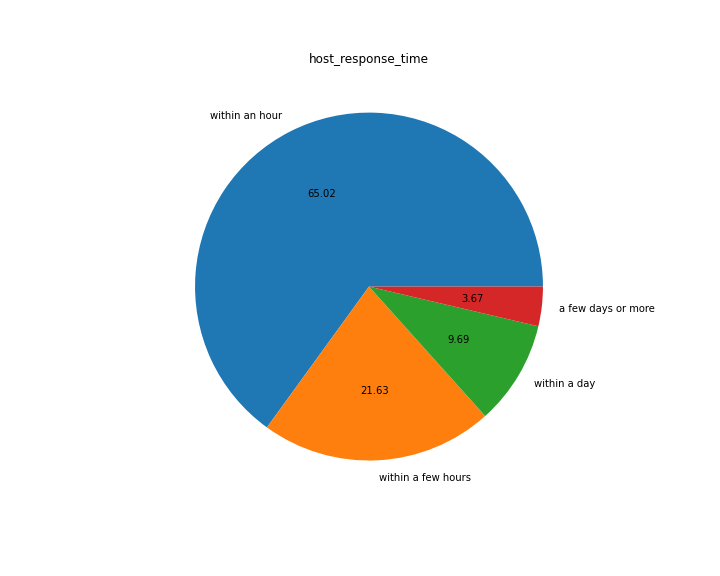
1. **Price**

****

* Price is right skewed.
* It is Leptokurtic, and narrow tailed.
* IQR lies at 0 .
* Outliers are present.

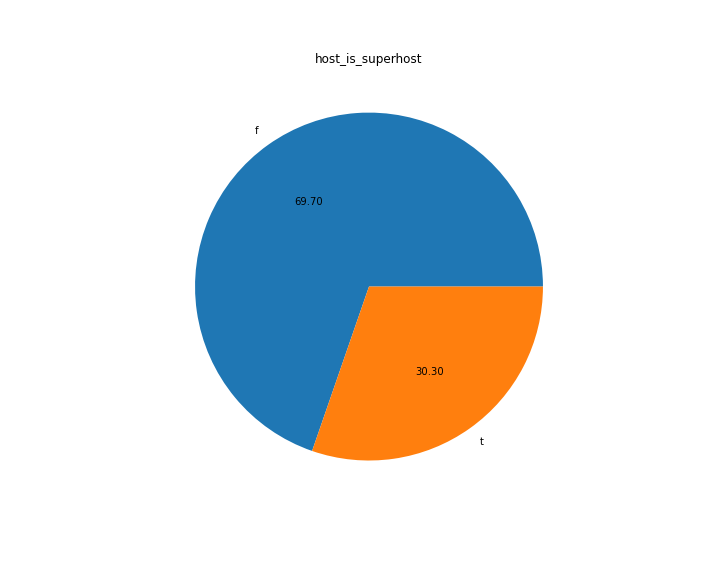
**1.2 Categorical:**

1. **Host\_response\_time**



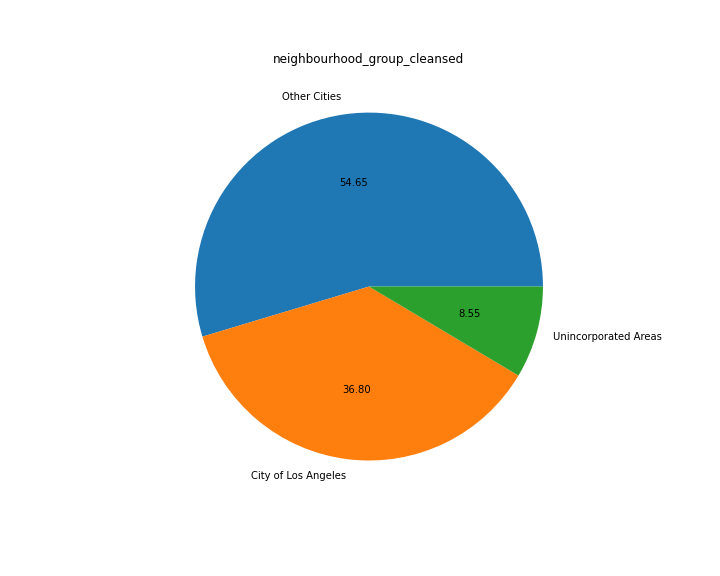
**Inference:** 65.02% of the host response time were handled within an hour followed 21.63% are handled within a few hours from the time of call.

1. **Host\_is\_superhost**

****

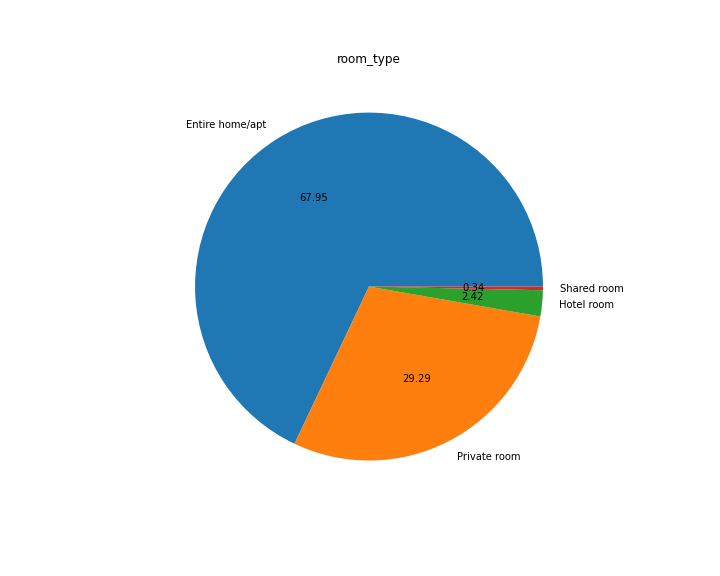
**Inference:** Among the host of the properties,69.70% is super host and 30.30% are not super host.

1. **Neighbourhood\_group\_cleanse:**

****

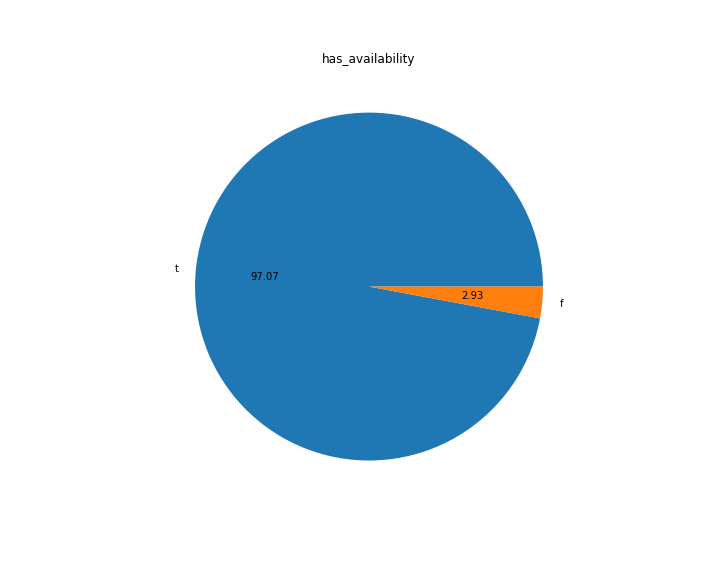
**Inference:** 54.65% of other cities have high percentage of neighborhood group cleansed whereas the city of Los Angeles has second highest percentage of 36.80% and 8.55% of Unincorporated Areas.

1. **Room\_type**



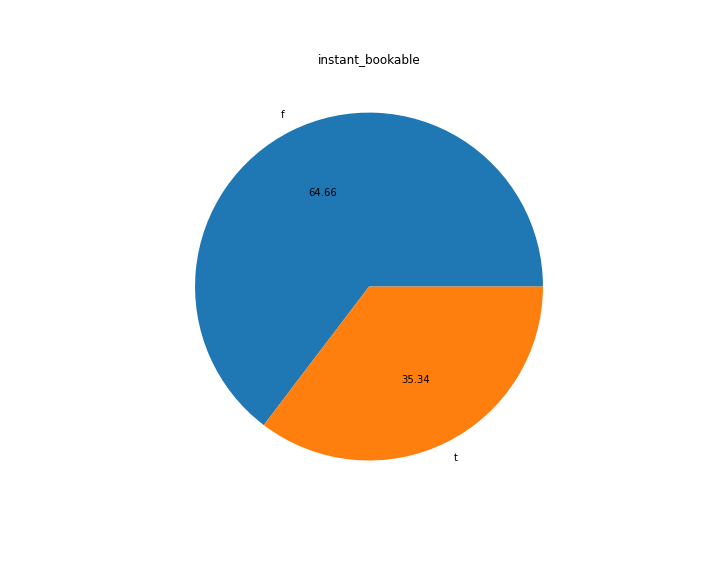
**Inference:** 67.95% of room type are belong to Entire home/apt, 29.29 is private room, 2.42 is hotel room and 0.34 is shared room.

1. **Has\_availability**

****

**Inference:** When the customer-tired booking via Airbnb had the chance of booking availability of 97.07% and 2.93 of non-availability chances.

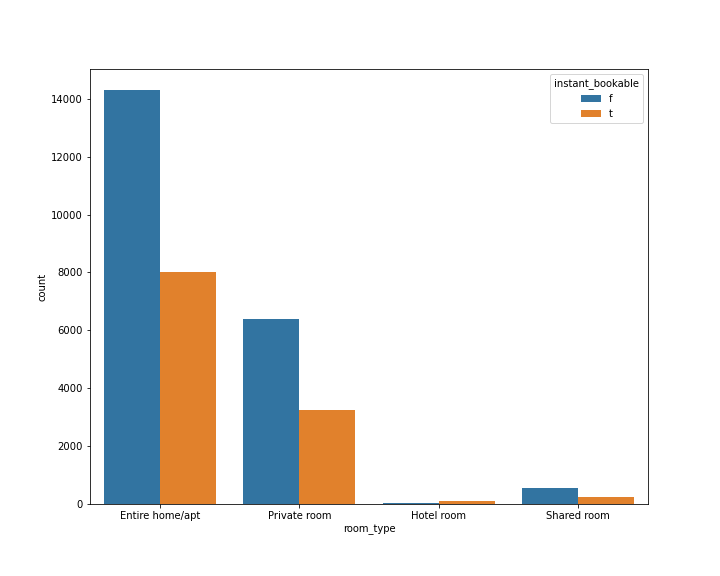
1. **Instant\_bookable**

****

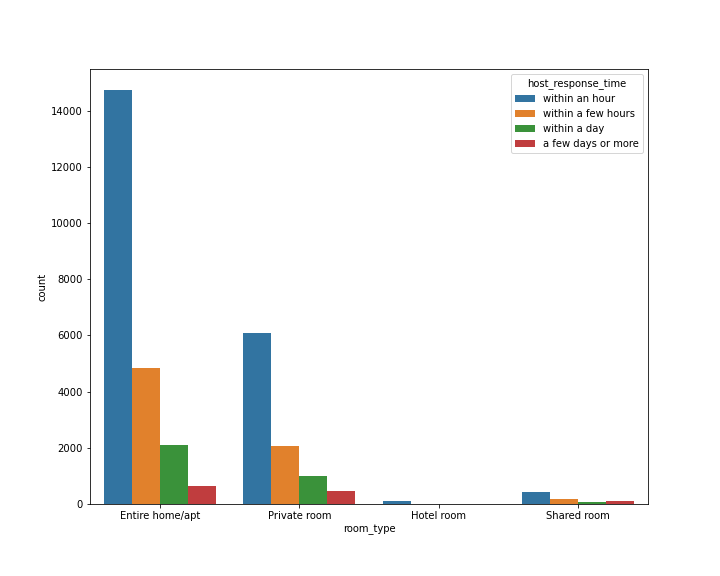
**Inference:** Customers were able to book the room at an instance basis of 64.66% and remaining 35.34% were not able to book at instance basis.

**2. BIVARIATE ANALYSIS:**

**2. a. Room\_type vs instant\_bookable:**

**Inference:** From the above bar plot, we could interpret that the category of Entire/Apt and Private Rooms are booked highly at an instant basis compare to hotel rooms and shared rooms.

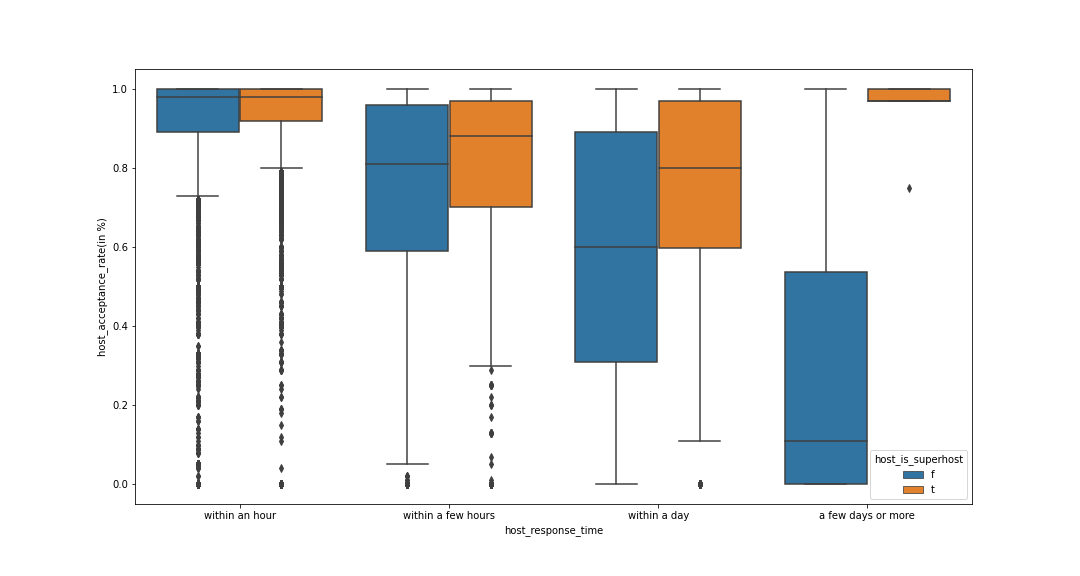
**2. b. Room\_type vs Host\_respose\_time:**



**Inference:** From the above bar plot, we interpret that the high calls were made for entire home/apt and private rooms categories compare to hotel room and shared room. Most of these calls were handled within an hour and few hours from the time of calls.

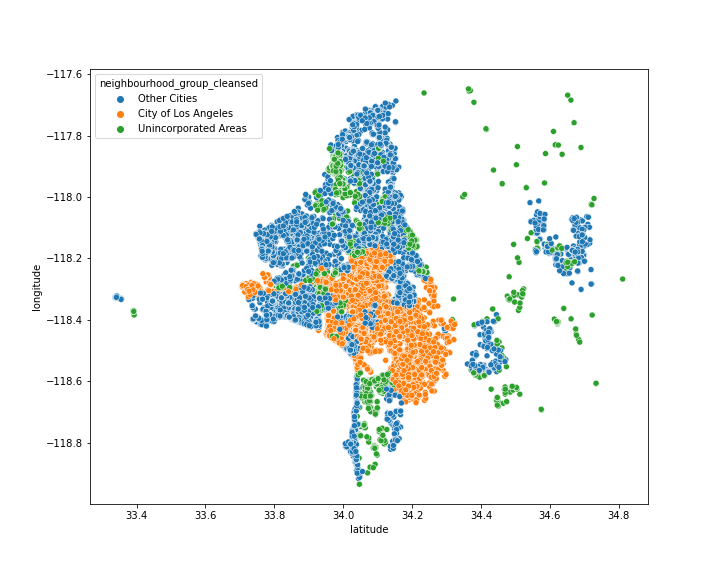
**3. MULTIVARIATE ANALYSIS:**

**3.a. Host\_acceptance\_rate vs Host\_response\_time vs Host\_is\_superhost:**



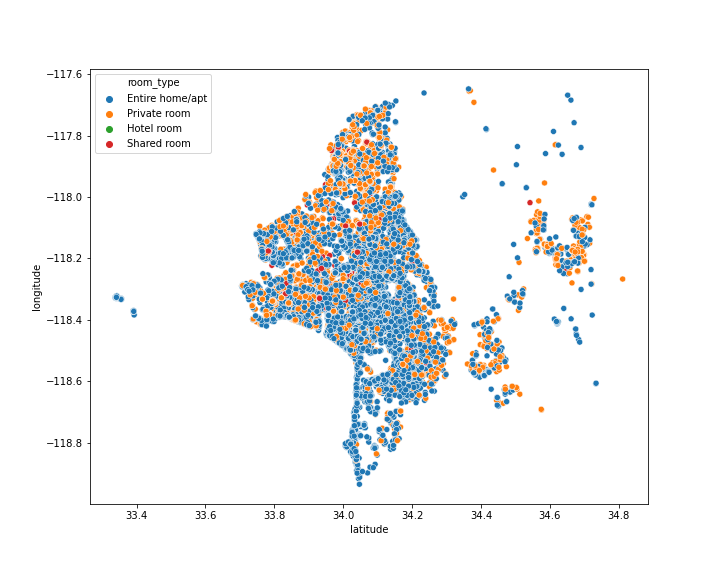
**Inference:** From the above graph, we could interpret that both Super host and non-Super host mostly took a day to accept the bookings and sometimes the super host also took few days to accept bookings. Whereas the acceptance rate within an hour and few hours are very less compare to with a day and few days categories.

**3.b. Latitude vs Longitude vs Neighbourhood\_group\_cleansed:**



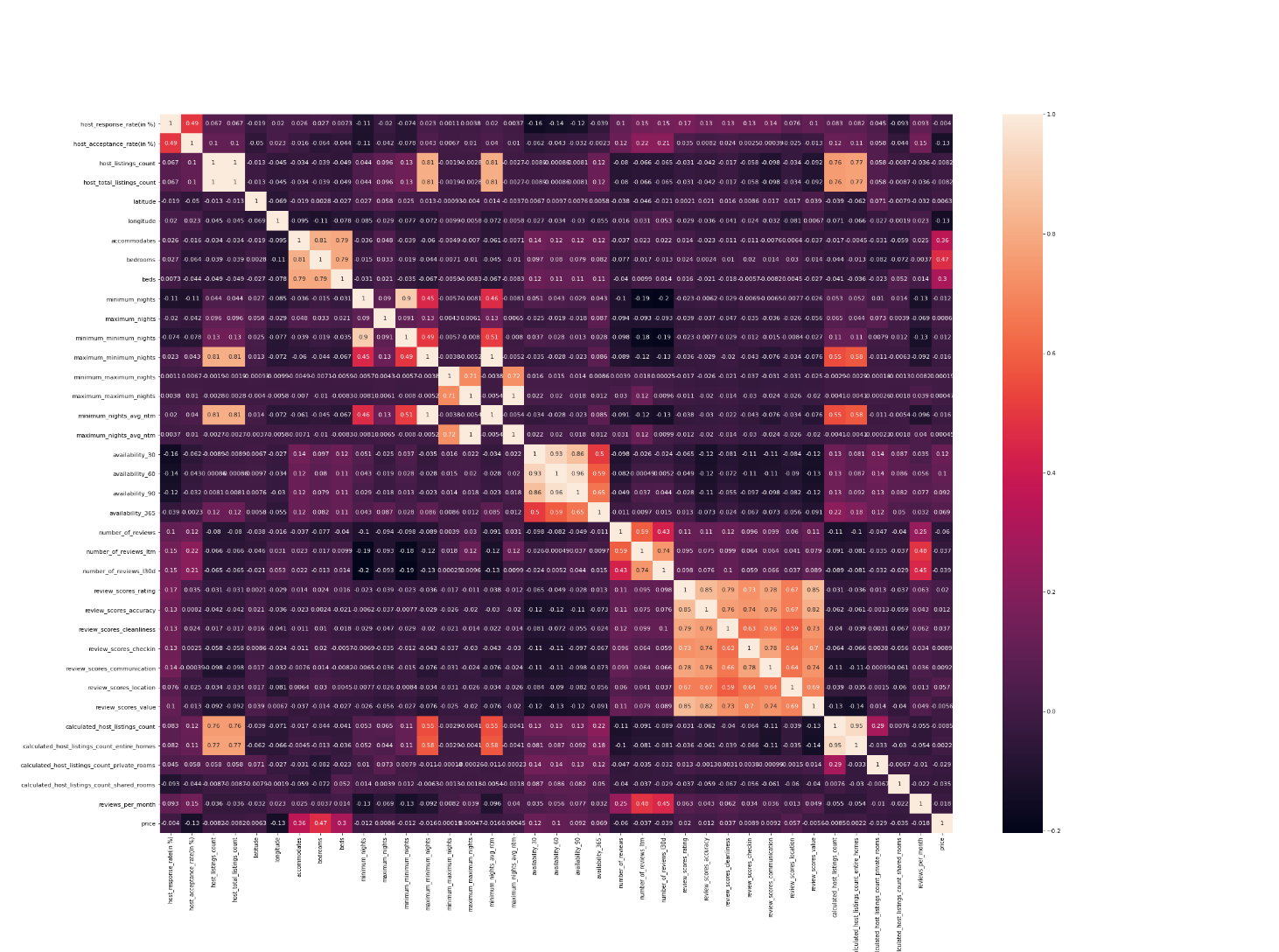
**Inference:** From the above graph, we could interpret that the high cleansed neighborhood group are from other cities followed by city of Los Angeles and finally the Unincorporated Areas.

**3.c. Latitude vs Longitude vs Room\_type:**



**Inference:** From the above graph, we could interpret that most of the host are hosting majorly the entire home/apt followed private room and very negligible amount of hotel and shared room.

**CORRELATION MATRIX**



**Inference:**

**1.Highly co-related variables**

host\_total\_listings\_count vs host\_listings\_count, Beds vs Bedroom, Beds vs Accommodates, Bedroom vs Accommodates, maximum\_minimum nights vs host\_listings\_count, minimum\_nights\_avg\_ntm vs host\_listings\_count, minimum\_nights\_avg\_ntm vs host\_total\_listings\_count, maximum\_minimum nights vs host\_total\_listings\_count, minimum minimum nights vs minimum nights, calculated\_host\_listings\_count vs calculated\_host\_listings\_count\_entire\_homes.

**2. Moderate co-related Variables:**

Host response rate vs Host acceptance rate, Maximum Minimum nights vs minimum nights, Maximum minimum nights vs Minimum Minimum Nights, calculated\_host\_listings\_count vs Maximum minimum nights.

**STATISTISTICAL TEST:**

**a. NUMERICAL VARIABLE VS NUMERICAL VARIABLE**

**1: Independent Variable = host\_response\_rate : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between host\_response\_rate and Price.

HA: There is a relationship between host\_response\_rate and Price.

Inference : Accept null hypothesis, the variable - host\_response\_rate is not a significant feature to predict the target column – Price

**2: Independent Variable = host\_acceptance\_rate : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between host\_acceptance\_rate and Price.

HA: There is a relationship between host\_acceptance\_rate and Price.

Inference : Accept null hypothesis, the variable - host\_acceptance\_rate is not a significant feature to predict the target column – Price

**3: Independent Variable = host\_total\_listings\_count : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between host\_total\_listings\_count and Price.

HA: There is a relationship between host\_total\_listings\_count and Price.

Inference : Accepting to reject null hypothesis, the variable - host\_total\_listings\_count is a signficant feature to predict the target column – Price

**4: Independent Variable = latitude : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between latitude and Price.

HA: There is a relationship between latitude and Price.

Inference : Accepting to reject null hypothesis, the variable - latitude is a significant feature to predict the target column – Price

**5: Independent Variable = longitude : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between longitude and Price.

HA: There is a relationship between longitude and Price.

Inference : Accepting to reject null hypothesis, the variable - longitude is a significant feature to predict the target column - Price

**6: Independent Variable = accommodates : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between accommodates and Price.

HA: There is a relationship between accommodates and Price.

Inference : Accepting to reject null hypothesis, the variable - accommodates is a significant feature to predict the target column - Price

**7: Independent Variable = bathrooms\_text : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between bathrooms\_text and Price.

HA: There is a relationship between bathrooms\_text and Price.

Inference : Accepting to reject null hypothesis, the variable - bathrooms\_text is a significant feature to predict the target column - Price

**8: Independent Variable = bedrooms : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between bedrooms and Price.

HA: There is a relationship between bedrooms and Price.

Inference : Accepting to reject null hypothesis, the variable - bedrooms is a significant feature to predict the target column - Price

**9: Independent Variable = beds : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between beds and Price.

HA: There is a relationship between beds and Price.

Inference : Accepting to reject null hypothesis, the variable - beds is a signficant feature to predict the target column - Price

**10: Independent Variable = amenities : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between amenities and Price.

HA: There is a relationship between amenities and Price

Inference : Accepting to reject null hypothesis, the variable - amenities is a signficant feature to predict the target column - Price

**11 Independent Variable = minimum\_nights : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between minimum nights and Price.

HA: There is a relationship between minimum\_nights and Price.

Inference : Accepting to reject null hypothesis, the variable - minimum\_nights is a signficant feature to predict the target column - Price

**12: Independent Variable = maximum\_nights : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between maximum\_nights and Price.

HA: There is a relationship between maximum\_nights and Price.

Inference : Accepting to reject null hypothesis, the variable - maximum\_nights is a signficant feature to predict the target column - Price

**13: Independent Variable = minimum\_minimum\_nights : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between minimum\_minimum\_nights and Price.

HA: There is a relationship between minimum\_minimum\_nights and Price.

Inference : Accepting to reject null hypothesis, the variable - minimum\_minimum\_nights is a signficant feature to predict the target column - Price

**14: Independent Variable = maximum\_minimum\_nights : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between maximum\_minimum\_nights and Price.

HA: There is a relationship between maximum\_minimum\_nights and Price.

Inference : Accepting to reject null hypothesis, the variable - maximum\_minimum\_nights is a signficant feature to predict the target column - Price

**15: Independent Variable = minimum\_maximum\_nights : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between minimum\_maximum\_nights and Price.

HA: There is a relationship between minimum\_maximum\_nights and Price.

Inference : Accepting to reject null hypothesis, the variable - minimum\_maximum\_nights is a signficant feature to predict the target column – Price

**16: Independent Variable = maximum\_maximum\_nights : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between maximum\_maximum\_nights and Price.

HA: There is a relationship between maximum\_maximum\_nights and Price.

Inference : Accept null hypothesis, the variable - maximum\_maximum\_nights is not a significant feature to predict the target column - Price

**17: Independent Variable = minimum\_nights\_avg\_ntm : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between minimum\_nights\_avg\_ntm and Price.

HA: There is a relationship between minimum\_nights\_avg\_ntm and Price.

Inference : Accepting to reject null hypothesis, the variable - minimum\_nights\_avg\_ntm is a signficant feature to predict the target column – Price

**18: Independent Variable = minimum\_nights\_avg\_ntm : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between minimum\_nights\_avg\_ntm and Price.

HA: There is a relationship between minimum\_nights\_avg\_ntm and Price.

Inference : Accepting to reject null hypothesis, the variable - minimum\_nights\_avg\_ntm is a signficant feature to predict the target column - Price

**19: Independent Variable = availability\_30 : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between availability\_30 and Price.

HA: There is a relationship between availability\_30 and Price.

Inference : Accepting to reject null hypothesis, the variable - availability\_30 is a signficant feature to predict the target column - Price

**20: Independent Variable = availability\_60 : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between availability\_60 and Price.

HA: There is a relationship between availability\_60 and Price.

Inference : Accepting to reject null hypothesis, the variable - availability\_60 is a signficant feature to predict the target column - Price

**21: Independent Variable = availability\_90 : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between availability\_90 and Price.

HA: There is a relationship between availability\_90 and Price.

Inference : Accepting to reject null hypothesis, the variable - availability\_90 is a signficant feature to predict the target column – Price

**22: Independent Variable = availability\_365 : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between availability\_365 and Price.

HA: There is a relationship between availability\_365 and Price.

Inference : Accepting to reject null hypothesis, the variable - availability\_365 is a signficant feature to predict the target column – Price

**23: Independent Variable = number\_of\_reviews : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between number\_of\_reviews and Price.

HA: There is a relationship between number\_of\_reviews and Price.

Inference : Accepting to reject null hypothesis, the variable - number\_of\_reviews is a signficant feature to predict the target column - Price

**24: Independent Variable = number\_of\_reviews\_ltm : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between number\_of\_reviews\_ltm and Price.

HA: There is a relationship between number\_of\_reviews\_ltm and Price.

Inference : Accepting to reject null hypothesis, the variable - number\_of\_reviews\_ltm is a signficant feature to predict the target column - Price

**25: Independent Variable = number\_of\_reviews\_l30d : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between number\_of\_reviews\_l30d and Price.

HA: There is a relationship between number\_of\_reviews\_l30d and Price.

Inference : Accepting to reject null hypothesis, the variable - number\_of\_reviews\_l30d is a signficant feature to predict the target column - Price

**26: Independent Variable = review\_scores\_rating : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_rating and Price.

HA: There is a relationship between review\_scores\_rating and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_rating is a signficant feature to predict the target column - Price

**27: Independent Variable = review\_scores\_accuracy : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_accuracy and Price.

HA: There is a relationship between review\_scores\_accuracy and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_accuracy is a signficant feature to predict the target column – Price

**28: Independent Variable = review\_scores\_cleanliness : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_cleanliness and Price.

HA: There is a relationship between review\_scores\_cleanliness and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_cleanliness is a signficant feature to predict the target column - Price

**29: Independent Variable = review\_scores\_checkin : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_checkin and Price.

HA: There is a relationship between review\_scores\_checkin and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_checkin is a signficant feature to predict the target column – Price

**30: Independent Variable = review\_scores\_communication : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_communication and Price.

HA: There is a relationship between review\_scores\_communication and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_communication is a signficant feature to predict the target column - Price

**31: Independent Variable = review\_scores\_location : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_location and Price.

HA: There is a relationship between review\_scores\_location and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_location is a

significant feature to predict the target column - Price

**32: Independent Variable = review\_scores\_value : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between review\_scores\_value and Price.

HA: There is a relationship between review\_scores\_value and Price.

Inference : Accepting to reject null hypothesis, the variable - review\_scores\_value is a

significant feature to predict the target column - Price

**33: Independent Variable = calculated\_host\_listings\_count : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between calculated\_host\_listings\_count and Price.

HA: There is a relationship between calculated\_host\_listings\_count and Price.

Inference : Accept null hypothesis, the variable - calculated\_host\_listings\_count is not a significant feature to predict the target column – Price

**34: Independent Variable = calculated\_host\_listings\_count\_entire\_homes : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between calculated\_host\_listings\_count\_entire\_homes and Price.

HA: There is a relationship between calculated\_host\_listings\_count\_entire\_homes and Price.

Inference : Accepting to reject null hypothesis, the variable - calculated\_host\_listings\_count\_entire\_homes is a signficant feature to predict the target column - Price

**35: Independent Variable = calculated\_host\_listings\_count\_private\_rooms : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between calculated\_host\_listings\_count\_private\_rooms and Price.

HA: There is a relationship between calculated\_host\_listings\_count\_private\_rooms and Price.

Inference : Accepting to reject null hypothesis, the variable - calculated\_host\_listings\_count\_private\_rooms is a signficant feature to predict the target column - Price

**36: Independent Variable = calculated\_host\_listings\_count\_shared\_rooms : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

**Hypothesis for SpearMan Correlation test**

H0: There is no relationship between calculated\_host\_listings\_count\_shared\_rooms and Price.

HA: There is a relationship between calculated\_host\_listings\_count\_shared\_rooms and Price.

Inference : Accepting to reject null hypothesis, the variable - calculated\_host\_listings\_count\_shared\_rooms is a signficant feature to predict the target column - Price

**37: Independent Variable = reviews\_per\_month : Target Variable = Price**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The data is not normally distributed, hence using non parametric test - SpearMan Correlation.

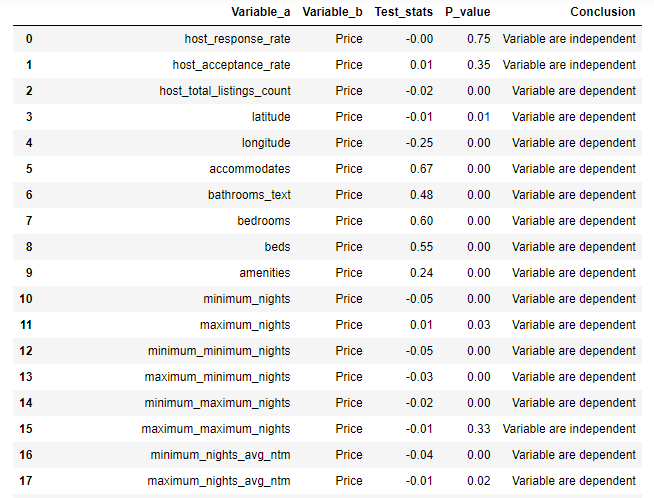
**Hypothesis for SpearMan Correlation test**

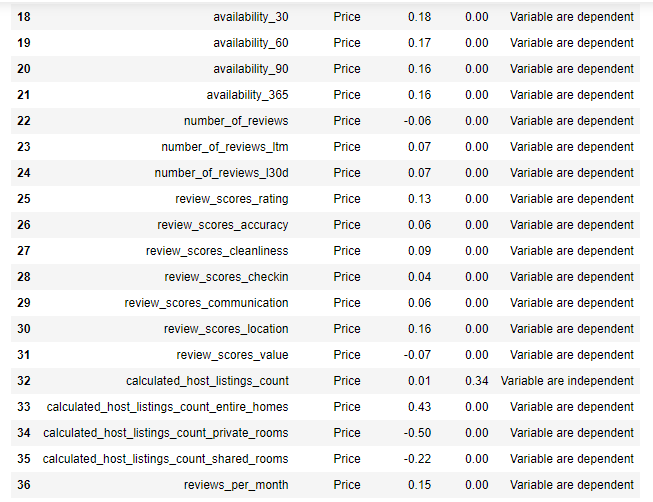
H0: There is no relationship between reviews\_per\_month and Price.

HA: There is a relationship between reviews\_per\_month and Price.

Inference : Accepting to reject null hypothesis, the variable - reviews\_per\_month is a signficant feature to predict the target column – Price

**Summary of Correlation test:**





**b. CATEGORICAL VS NUMERICAL:**

**Hypothesis for Normality**

H0: The data is normally distributed

HA: The data is not normally distributed

Inference : The price data is not normally distributed, hence using non-parametric test like Mann-Whitney and Kruskal-Wallis test for checking dependency.

**1: Independent Variable = host\_responce\_time : Target Variable = Price**

**Hypothesis for Kruskal-Wallis test**

H0: The host\_response\_time and price are independent.

HA: The host\_response\_time and price are dependent.

Inference : Accepting to reject null hypothesis, the variable - host\_response\_time is a signficant feature to predict the target column – Price

**2: Independent Variable = host\_is\_superhost : Target Variable = Price**

**Hypothesis for Mann-Whitney test**

H0: The host\_ is\_superhost and price are independent.

HA: The host\_ is\_superhost and price are dependent.

Inference : Accepting to reject null hypothesis, the variable - host\_ is\_superhost is a signficant feature to predict the target column – Price

**3: Independent Variable = neighbourhood\_group\_cleansed : Target Variable = Price**

**Hypothesis for Kruskal-Wallis test**

H0: The neighbourhood\_group\_cleansed and price are independent.

HA: The neighbourhood\_group\_cleansed and price are dependent.

Inference : Accepting to reject null hypothesis, the variable - neighbourhood\_group\_cleansed is a significant feature to predict the target column – Price

**4: Independent Variable = room\_type : Target Variable = Price**

**Hypothesis for Kruskal-Wallis test**

H0: The room\_type and price are independent.

HA: The room\_type and price are dependent.

Inference : Accepting to reject null hypothesis, the variable - room\_type is a signficant feature to predict the target column – Price

**5: Independent Variable = has\_availability : Target Variable = Price**

**Hypothesis for Mann-Whitney test**

H0: The has\_availability and price are independent.

HA: The has\_availability and price are dependent.

Inference : Accepting to reject null hypothesis, the variable - has\_availability is a significant feature to predict the target column – Price

**6: Independent Variable = instant\_bookable : Target Variable = Price**

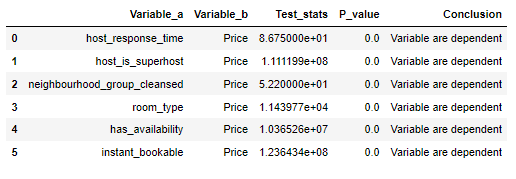
**Hypothesis for Mann-Whitney test**

H0: The instant\_bookable and price are independent.

HA: The instant\_bookable and price are dependent.

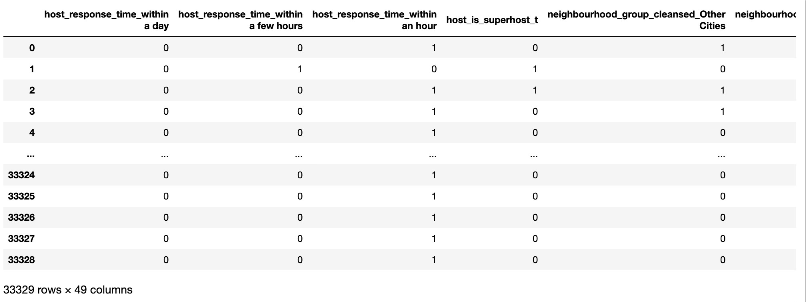
Inference : Accepting to reject null hypothesis, the variable - instant\_bookable is a signficant feature to predict the target column – Price

**Summary for Categorical Statistical analysis:**



**Encoding:**

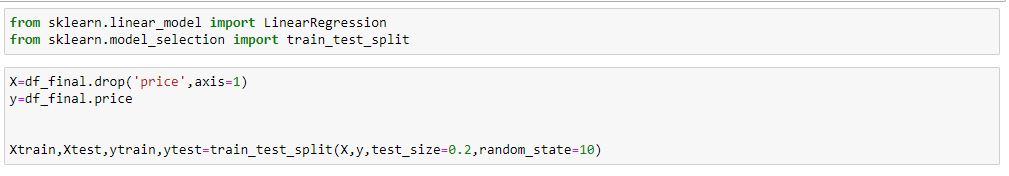
We have 38 numeric features and 6 categorical features in the table. As these 6 categorical variables do not have any ordinal data type and it is nominal in nature. We are concluding to use the one hot encoding imputation with criteria of dropping of first column from each variable. Post the imputation, we have 49 features among these 11 features are from categorical imputation.

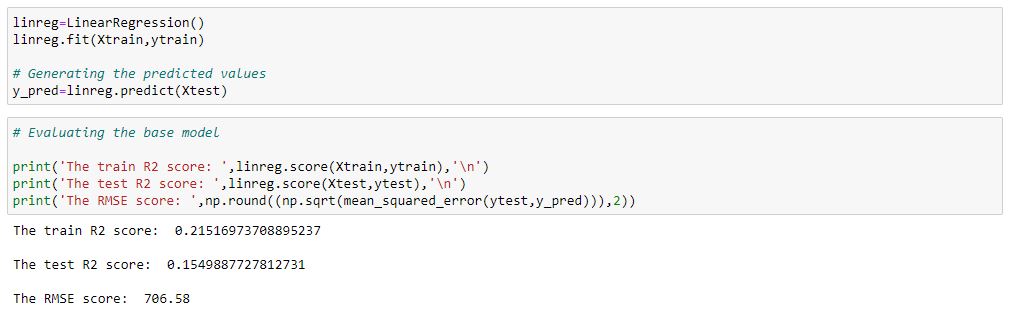


**BASE MODEL**

**Linear Regression:**

We have selected Linear Regression as our base model. The purpose of selecting this model is very useful to understand about each features contribution from the data to predict the target variable and also that it is suitable to check whether the data is satisfying all the assumptions of linear regression.

****

****

**Inference:** By the only activities of imputing null values, encoding the categorical variables. With the base model linear regression, we were able to achieve only the train R\_square of 0.21 and test R\_square of 0.15. Thus, states that created base model is underfit model which has high bias error.

**ASSUMPTIONS OF LINEAR REGRESSION:**

**1.Mutlicolinearity**

Multicollinearity is the occurrence of high intercorrelations among two or more independent variables in multiple regression model. When two or more independent variables are representing the same information, thus leads to inappropriate information in prediction of the target variable. From the OLS model, we can check whether that the presence of multicollinearity from the parameter called ‘Condition Number’.

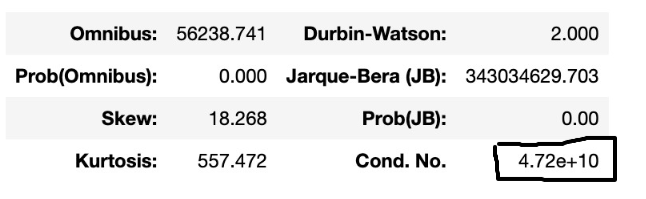
In real time scenario, we cannot expect all the time that model supposed to have 100% multicollinearity free features with the condition number less than 100. For an ideal situation, we can have the condition number between 100 to 1000 and still no treatment is required. If the condition no is greater than 1000, then we can conclude that there is strong relationship between the independent features and it is to be treated.

Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables.

* 1. When the VIF is <= 1, we can conclude that no multicollinearity between the independent features.
  2. VIF between >1 to <=5, we can conclude that moderate multicollinearity between the independent features.
  3. VIF >5, we can conclude that strong multicollinearity presents between the independent features.

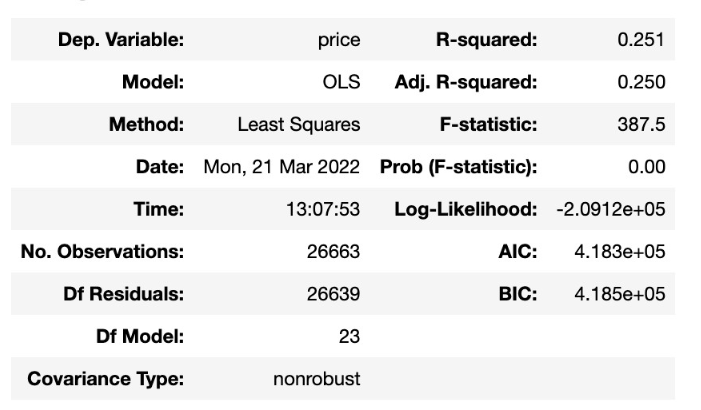
**Multi collinearity model check:**

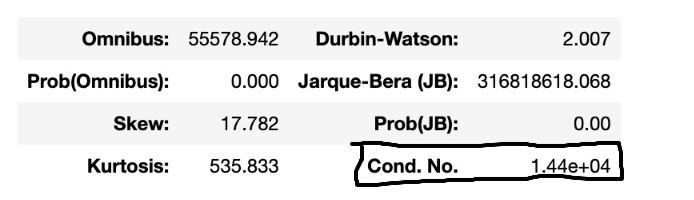
From the base model, we have got the condition no of 4.72e+10 and it is greater than condition no of 1000. Thus, states clearly that the independent features from the model have very strong relationship among them.



We have dropped the individual independent features one by one and reiterated the VIF values for them. We have only 23 features that that has VIF values less than or equal to 5 from 49 features.

Performed another regression model with these 23 features where VIF values are less than 5. However, we could see that the accuracy score of the base model has been improved from 0.21% to 0.25% but the condition No has not reduced to less than 1000 and it still ranges around 1.44e+04.





**2.Linearity:**

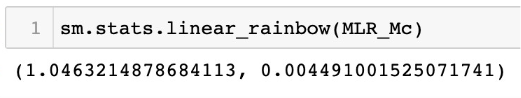
The linearity check is one of the main assumptions that talks about independent variables supposed to have linear relationship with target variable in order to predict the target variable for the better accurate prediction.

We can check the linearity assumption by using the rainbow test.

**Rainbow test assumption of hypothesis:**

* H0: It is good linear model
* HA: It is not good linear model

**Test Statistics and P\_value:**



**Conclusion:**

From the test statistics, we have got the pvalue as 0.004 and it is lesser than alpha value of 0.05. Hence, we are agreeing to reject the null hypothesis and concluding that the independent variables are not having linear relationship with the target variable.

**3.Autocorrelation**

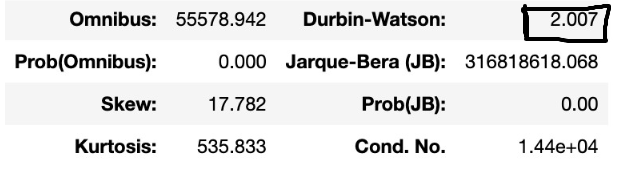
Independence of observation should exist (i.e.,) when the residuals are correlated within themselves then the assumption is violated.

The following assumption can be evaluated by using ‘Durbin-Watson-Test’.

The value from the test can be denoted as ‘d’.

* + 1. 0 < d < 2 - Positive auto-correlation,
    2. d = 2 - No auto correlation,
    3. 2 < d < 4 - Negative correlation.

**Auto-correlation check**



**Conclusion:**

From the OLS model, we have derived the score of Durbin-Waston as 2.007. Hence, we can conclude that there is no auto-correlation between the residuals.

**4.** **Homoscedastic**

Homoscedastic refers to a condition in which the variance of the residual, or error term, in a regression model is constant. If the variance of the error term is homoscedastic, the model was well-defined. If there is too much variance, the model may not be defined well and then it is called as heteroscedastic.

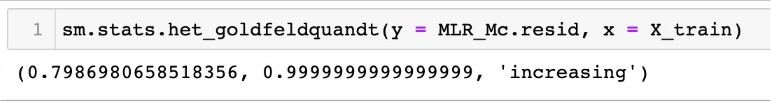
We can check the homoscedastic assumption by using the goldfeld test.

**Homoscedastic check:**

**Hypothesis\_for\_goldfeld\_test:**

* H0: Heteroscedasticity is not present
* HA: Heteroscedasticity is present.

**Test Statistics and P\_value:**

****

**Conclusion:**

From the test statistics, we have got the pvalue as 0.99 and it is greater than alpha value of 0.05. Hence, we are agreeing to accept the null hypothesis and concluding that the residuals have same variance.

**5. Normality of the residue:**

Normality is the assumption that the underlying residuals or error terms are normally distributed, or approximately.

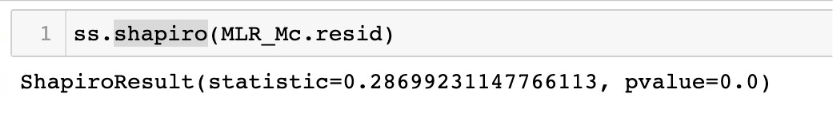
**Normality of the residue check:**

We can check the normality of the residue assumption by using the shapiro test.

**Hypothesis for Normality**

H0: Residue is normal  
HA: Residue is not normal

**Test Statistics and P\_value:**



**Conclusion:**

From the test statistics, we have got the pvalue as 0.00 and it is lesser than alpha value of 0.05. Hence, we are agreeing to reject the null hypothesis and concluding that the residuals of the model do not form normal distribution.

**ASSUMPTIONS INFERENCE:**

From the above statistical approach, we could observe that non-violated and violated assumptions are enlisted below.

**Non-Violated Assumption:**

1. Homoscedasticity,
2. Autocorrelation.

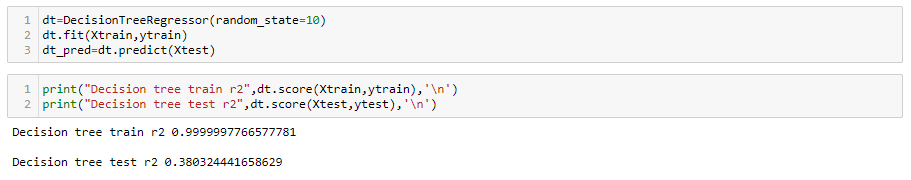
**Violated Assumption:**

1. Multicollinearity,
2. Linearity,
3. Normality check of the residuals.

As, three assumption checks are violating and fitting the above data in the linear model is strongly not recommendable. Hence, we can go-ahead and start using the non-linear models for the prediction.

**Non-Linear Model:**

**1. Decision Tree:**



While building a Decision Tree Model using sklearn we have come across the above R2 value of the Decision Tree Model.

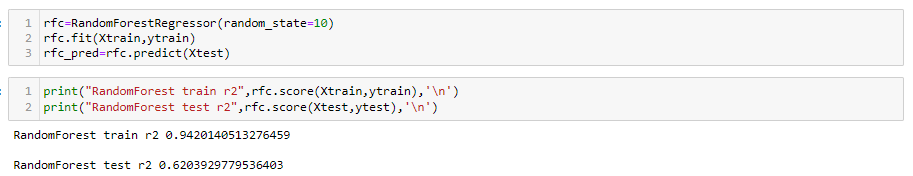
Inferences for Decision Tree Regressor Model:

• The train R2 and test R2 shows that the model is overfitting

• The Root\_mean\_square\_error of the model came around 614.32.

By doing the R2 score analysis for train and test of the decision tree model. The R2 of train is coming around to be 99.99 percent and the R2 of test is around 38.03 percent. So it states that the model is over fitting. We can overcome the problem of over fitting by doing decision tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**2. Random Forest:**



While building a Random forest using sklearn we have come across the above R2 value of the Random forest Model.

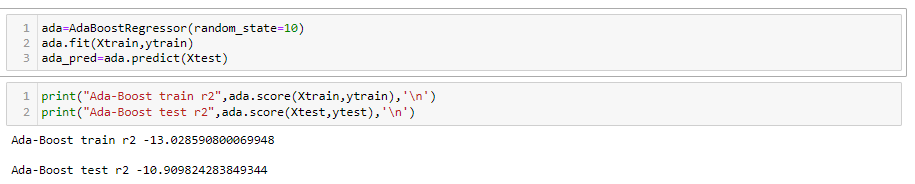
Inferences for Random forest Regressor Model:

• The train R2 and test R2 shows that the model is overfitting

• The Root\_mean\_square\_error of the model came around 480.82.

By doing the R2 score analysis for train and test of the decision tree model. The R2 of train is coming around to be 94.20 percent and the R2 of test is around 62.03 percent. So it states that the model is over fitting. We can overcome the problem of over fitting by doing decision tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**3. Ada-Boosting Regressor:**



While building an Ada-Boosting using sklearn we have come across the above R2 value of the Ada-Boosting Model.

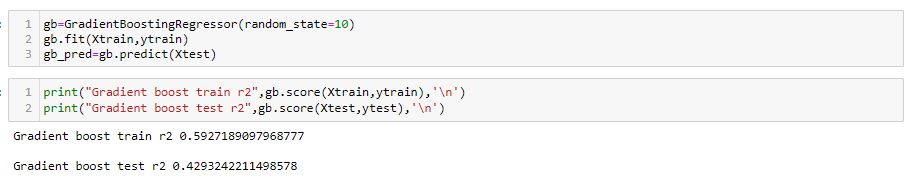
Inferences for Ada-Boosting Regressor Model:

• The train R2 and test R2 shows that the model is underfitting

• The Root\_mean\_square\_error of the model came around 2693.18.

By doing the R2 score analysis for train and test of the decision tree model. The R2 of train is coming around to be -13.02 percent and the R2 of test is around -10.90 percent. So it states that the model is under fitting. We can overcome the problem of over fitting by doing decision tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**4. Gradient Boosting Regressor:**



While building a Gradient Boosting using sklearn we have come across the above R2 value of the Gradient Boosting Model.

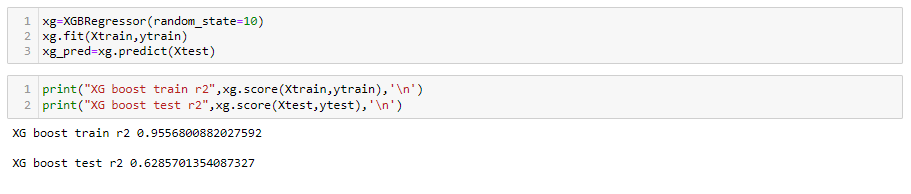
Inferences for Gradient Boosting Regressor Model:

• The train R2 and test R2 shows that the model is underfitting

• The Root\_mean\_square\_error of the model came around 589.53.

By doing the R2 score analysis for train and test of the decision tree model. The R2 of train is coming around to be 59.27 percent and the R2 of test is around 42.93 percent. So it states that the model is under fitting. We can overcome the problem of over fitting by doing decision tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**5. XG Boosting Regressor:**



While building a XG Boosting using xgboost library we have come across the above R2 value of the XG Boosting Model.

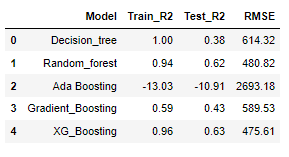
Inferences for XG Boosting Regressor Model:

• The train R2 and test R2 shows that the model is overfitting

• The Root\_mean\_square\_error of the model came around 475.61.

By doing the R2 score analysis for train and test of the decision tree model. The R2 of train is coming around to be 95.56 percent and the R2 of test is around 62.85 percent. So it states that the model is under fitting. We can overcome the problem of over fitting by doing decision tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

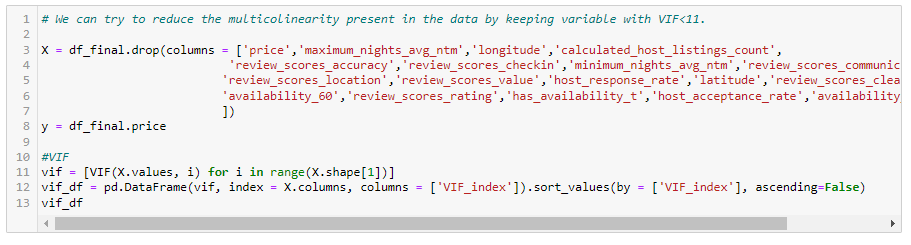
**Inference from the non-linear model:**

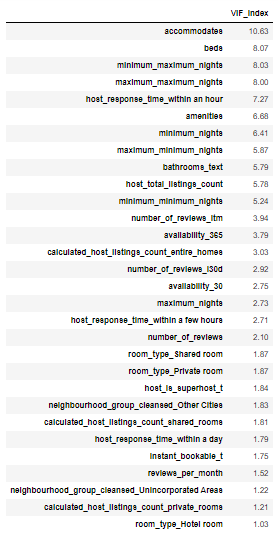


From the five non-linear model performed we can come to the conclusion that all the model are under performing and is influenced by various features of the data. We can improve the perform of these non-linear model by doing the following:

* Treating the multi-collinearity in the data.
* Treating the outliers in the data.

**Treating the multi- collinearity:**



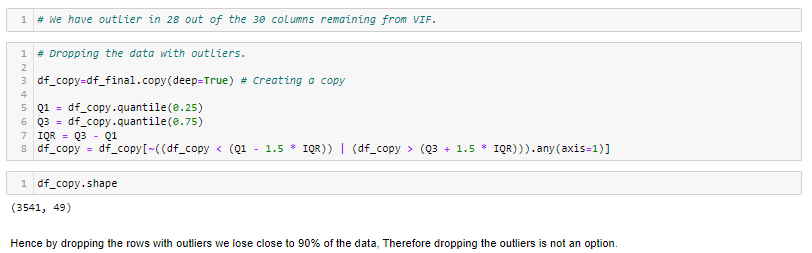


We can reduce the multi-collinearity using the VIF, removing the variable with VIF threshold less than 11.

We are left with 30 attributes as shown by the above table.

**Treating the outliers in the model:**

* **Dropping the outlier:**

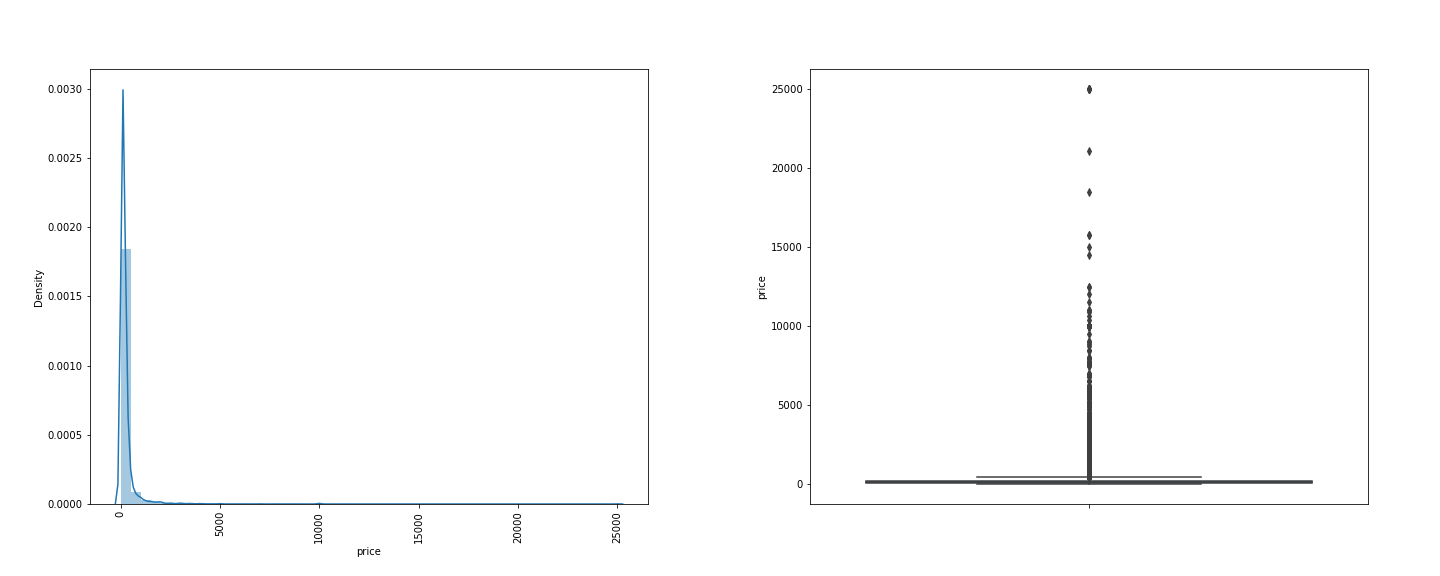


**Inference:** As we can see by dropping the rows with outliers we are left with only 3541 rows out of the 33000 rows, which is approx. 90% of the data. Hence we have to find a better way to treat the outliers.

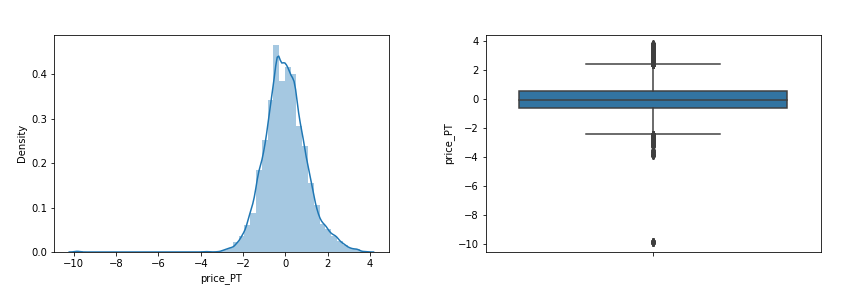
* **Transforming the target:**

Also we can have outliers is 28 out of the 30 variables left hence transforming all the variable is not a good practice. Therefore we can try and transform only the target variable and run the non-linear model again.

Price distribution before transformation:

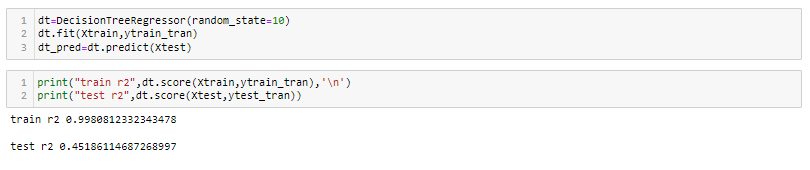
****



****

**Non-Linear Model after treating multi-collinearity and outliers:**

**1. Decision Tree:**



While building a Decision Tree Model using sklearn we have come across the above R2 value of the Decision Tree Model.

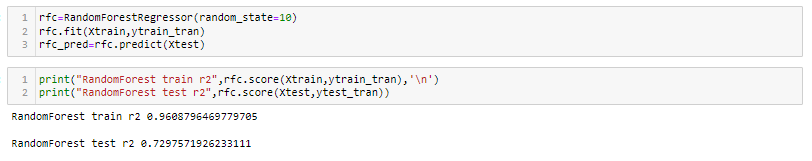
Inferences for Decision Tree Regressor Model:

• The train R2 and test R2 shows that the model is overfitting

• The Root\_mean\_square\_error of the model came around 0.73.

By doing the R2 score analysis for train and test of the decision tree model. The R2 of train is coming around to be 99.80 percent and the R2 of test is around 45.18 percent. So it states that the model is over fitting. We can overcome the problem of over fitting by doing decision tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**2. Random Forest:**



While building a Random forest using sklearn we have come across the above R2 value of the Random forest Model.

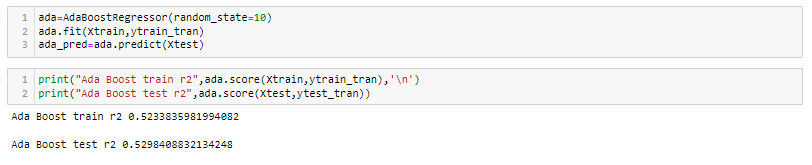
Inferences for Random forest Regressor Model:

• The train R2 and test R2 shows that the model is overfitting

• The Root\_mean\_square\_error of the model came around 0.51.

By doing the R2 score analysis for train and test of the Random forest model. The R2 of train is coming around to be 96.08 percent and the R2 of test is around 72.97 percent. So it states that the model is over fitting. We can overcome the problem of over fitting by doing random forest tree pruning or hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**3. Ada-Boosting Regressor:**



While building an Ada-Boosting using sklearn we have come across the above R2 value of the Ada-Boosting Model.

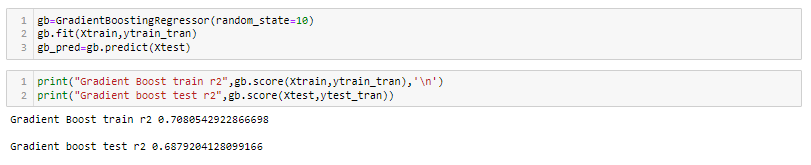
Inferences for Ada-Boosting Regressor Model:

• The train R2 and test R2 shows that the model is underfitting

• The Root\_mean\_square\_error of the model came around 0.68.

By doing the R2 score analysis for train and test of the Ada-Boosting model. The R2 of train is coming around to be 52.33 percent and the R2 of test is around 52.98 percent. So it states that the model is under fitting. We can overcome the problem of under fitting by hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**4. Gradient Boosting Regressor:**



While building a Gradient Boosting using sklearn we have come across the above R2 value of the Gradient Boosting Model.

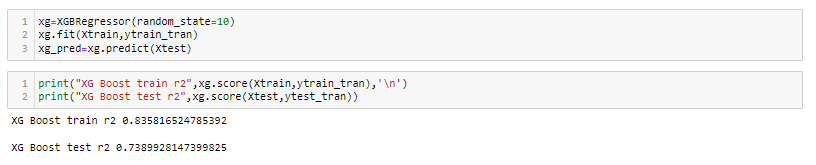
Inferences for Gradient Boosting Regressor Model:

• The train R2 and test R2 shows that the model is slightly underfitting.

• The Root\_mean\_square\_error of the model came around 0.55.

By doing the R2 score analysis for train and test of the Gradient Boosting model. The R2 of train is coming around to be 70.80 percent and the R2 of test is around 68.79 percent. So it states that the model is slightly under fitting. We can overcome the problem of under fitting by hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**5. XG Boosting Regressor:**



While building a XG Boosting using xgboost library we have come across the above R2 value of the XG Boosting Model.

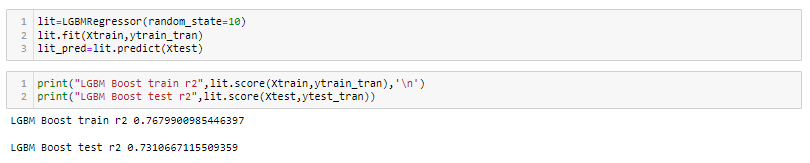
Inferences for XG Boosting Regressor Model:

• The train R2 and test R2 shows that the model is performing well.

• The Root\_mean\_square\_error of the model came around 0.51.

By doing the R2 score analysis for train and test of the XG Boosting model. The R2 of train is coming around to be 83.58 percent and the R2 of test is around 73.89 percent. So we can see that the model performance is good and we can improve the model by performing hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**6. Light Gradient Boosting Regressor:**



While building a Light Gradient Boosting using lgmboost we have come across the above R2 value of the Light Gradient Boosting Model.

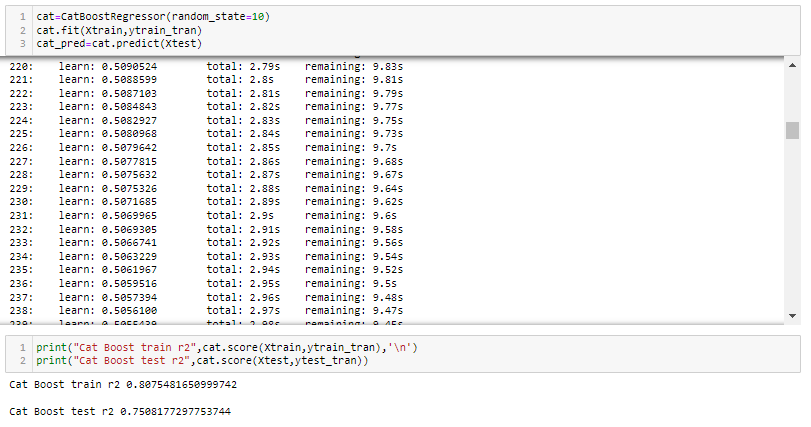
Inferences for Light Gradient Boosting Regressor Model:

• The train R2 and test R2 shows that the model is performing well.

• The Root\_mean\_square\_error of the model came around 0.51.

By doing the R2 score analysis for train and test of the Light Gradient Boosting model. The R2 of train is coming around to be 76.79 percent and the R2 of test is around 73.10 percent. So we can see that the model performance is good and we can improve the model by performing hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**7. CAT Boosting Regressor:**



While building a CAT Boosting using catboost library we have come across the above R2 value of the CAT Boosting Model.

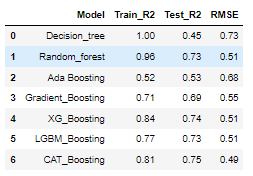
Inferences for CAT Boosting Regressor Model:

• The train R2 and test R2 shows that the model is performing well.

• The Root\_mean\_square\_error of the model came around 0.49.

By doing the R2 score analysis for train and test of the XG Boosting model. The R2 of train is coming around to be 80.75 percent and the R2 of test is around 75.08 percent. So we can see that the model performance is good and we can improve the model by performing hyper parameter tuning with a given set of parameters using Grid Search CV that will help us getting the best parameters for our model building.

**Inference from the non-linear model after treating outlier and multi-colinearity:**



From the above evaluation table we can see that the performance for all the non-linear model after treating the multi-colinearity and the transforming the target has significantly improved. Among the six non-linear model we can see the Random forest, XG Boost and the CAT Boost model perform the best.

These base model performance can be further improved by fine tuning the hyper parameters of these algorithms using the GridSearchCV.

**Hyperparameter Tunning:**

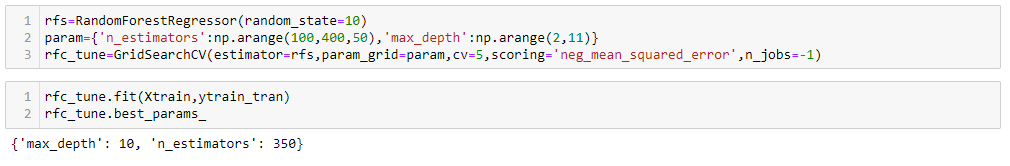
Random forest, XG Boost and the CAT Boost model have been chosen and a set of parameters has been considered for fine tuning.

Using these set of parameters we have used Grid Search Cross Validation technique for

Hyper parameter tuning the models where the Cross Validation method considered is 5-

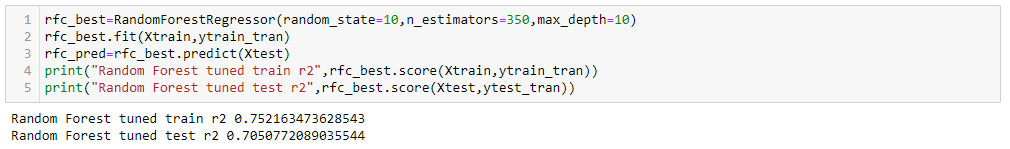
Fold Cross Validation.

1. **Random Forest regressor:**



Random Forest Regressor : {'n\_estimator': 350, 'max\_depth': 10}

**Refitting the Tunned Based models to compare the performances**



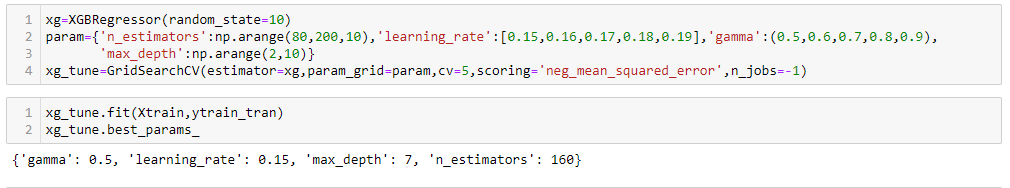
After refitting the tunned base model of Random forest regressor we carried out the R2 score analysis for the tunned Random Forest model.

**Inference:**

* The R2 for the train data came around 75.21 percent
* The R2 for the train data came around 70.5 percent
* The Root\_mean\_square for the model : 0.54
* The Mean\_absolute\_error for the model: 0.39
* The Mean\_absolute\_percentage\_error for the model: 143.94

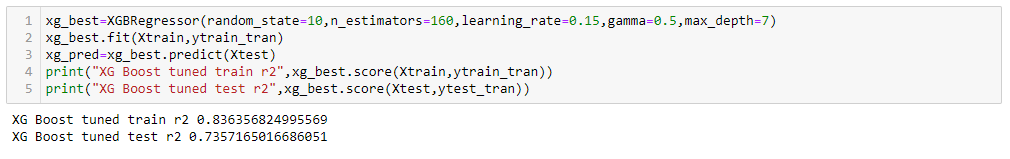
From the above R2 score analysis we can see that the overfitting present the based model was mitigated. The overall performs of the model looks good.

1. **XG Boost Regressor:**



XG Boost Regressor : {'n\_estimator': 160, 'max\_depth': 7, ‘learning\_rate’: 0.15,

‘gamma’:0.5}



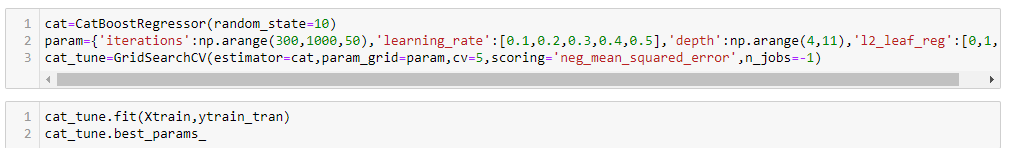
After refitting the tunned base model of XG Boost regressor we carried out the R2 score analysis for the tunned Random Forest model.

**Inference:**

* The R2 for the train data came around 83.63 percent
* The R2 for the train data came around 73.57 percent
* The Root\_mean\_square for the model : 0.51
* The Mean\_absolute\_error for the model: 0.36
* The Mean\_absolute\_percentage\_error for the model: 138.22

From the above R2 score analysis we can see that the overall performance wrt base model has not improved any further. The overall performs of the tuned XG Boost regeressor model looks good.

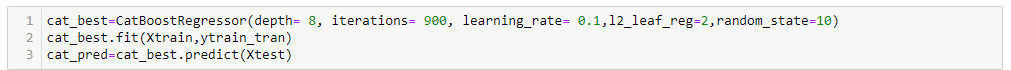
1. **CAT Boost Regeressor:**

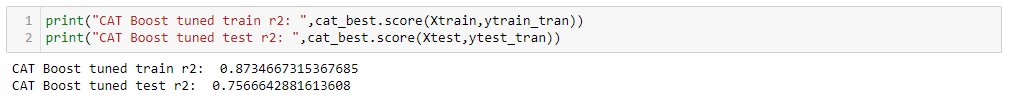




CAT Boost Regressor : {‘iterations’: 900, 'depth': 8, ‘learning\_rate’: 0.1,

‘l2\_leaf\_reg’: 2}





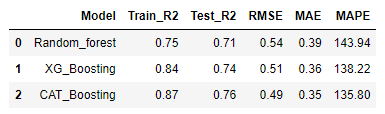
After refitting the tunned base model of CAT Boost regressor we carried out the R2 score analysis for the tunned Random Forest model.

**Inference:**

* The R2 for the train data came around 87.34 percent
* The R2 for the train data came around 75.66 percent
* The Root\_mean\_square for the model : 0.49
* The Mean\_absolute\_error for the model: 0.35
* The Mean\_absolute\_percentage\_error for the model: 135.80

From the above R2 score analysis we can see that the overall performance wrt base model has improved considerably. The overall performs of the tuned CAT Boost regeressor model looks great.

**Evaluation of the tuned model:**



From the above evaluation matrix, we can see that the tuned CAT Boost regressor is the best performing model on all the evaluation parameter.

**Feature Selection:**

Feature Selection is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature Selection is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set. The process of feature selection is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature Selection can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine’s efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

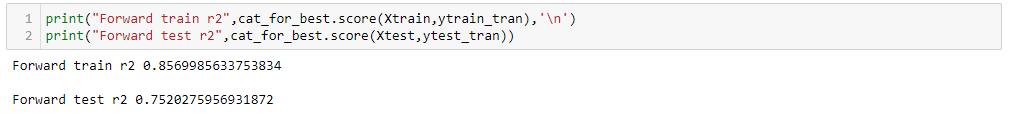
**Using Forward Feature Selection:**

Since the analysis shows that the CAT boost regressor has the best overall performance from all the previously deployed base and tuned models, it will be considered for deriving the final set of features affecting the ML algorithm using the forward selection model.

The Forward selection process was carried out and the best 26 feature that have significant impact on the target was identified.

**['host\_response\_time\_within a day','host\_response\_time\_within an hour', 'host\_is\_superhost\_t', 'neighbourhood\_group\_cleansed\_Other Cities', 'neighbourhood\_group\_cleansed\_Unincorporated Areas', 'room\_type\_Hotel room', 'room\_type\_Private room', 'room\_type\_Shared room', 'instant\_bookable\_t', 'host\_total\_listings\_count', 'accommodates', 'bathrooms\_text', 'beds', 'minimum\_nights', 'maximum\_nights', 'maximum\_minimum\_nights', 'minimum\_maximum\_nights', 'maximum\_maximum\_nights', 'availability\_30', 'availability\_365', 'number\_of\_reviews\_ltm', 'number\_of\_reviews\_l30d', 'calculated\_host\_listings\_count\_entire\_homes', 'calculated\_host\_listings\_count\_private\_rooms', 'calculated\_host\_listings\_count\_shared\_rooms', 'reviews\_per\_month']**

**Refitting the tuned CAT Boost regressor model with the 26 significant variable:**



After refitting the model we checked the R2 for the model and the train R2 came around: 85.69 percent and the test R2 came around 75.20 percent.

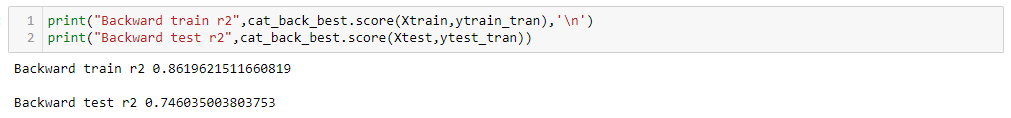
**Using Backward Feature Selection:**

Since the analysis shows that the CAT boost regressor has the best overall performance from all the previously deployed base and tuned models, it will be considered for deriving the final set of features affecting the ML algorithm using the backward selection model.

The Backward selection process was carried out and the best 23 feature that have significant impact on the target was identified.

**['host\_response\_time\_within a day', 'host\_response\_time\_within an hour', 'neighbourhood\_group\_cleansed\_Other Cities', 'neighbourhood\_group\_cleansed\_Unincorporated Areas', 'room\_type\_Hotel room', 'room\_type\_Private room', 'room\_type\_Shared room', 'instant\_bookable\_t', 'accommodates', 'bathrooms\_text', 'beds', 'amenities', 'maximum\_nights', 'maximum\_minimum\_nights', 'maximum\_maximum\_nights', 'availability\_30', 'availability\_365', 'number\_of\_reviews\_ltm', 'number\_of\_reviews\_l30d', 'calculated\_host\_listings\_count\_entire\_homes', 'calculated\_host\_listings\_count\_private\_rooms', 'calculated\_host\_listings\_count\_shared\_rooms', 'reviews\_per\_month']**

**Refitting the tuned CAT Boost regressor model with the 23 significant variable:**



After refitting the model, we checked the R2 for the model and the train R2 came around: 86.19 percent and the test R2 came around 74.60 percent.

The R2 score is better for the model fitted with features obtained from backward selection, also with lesser number variables. Hence, we can conclude that the backward feature selection method produced the better model.

**Final Prediction Model (CostPrediction\_CATBoostModel):**

***Source Code:***

*def CostPrediction\_CATBoostModel(fin\_model):*

*X = cost\_df [['host\_response\_time\_within a day', 'host\_response\_time\_within an hour', 'neighbourhood\_group\_cleansed\_Other Cities', 'neighbourhood\_group\_cleansed\_Unincorporated Areas', 'room\_type\_Hotel room', 'room\_type\_Private room', 'room\_type\_Shared room', 'instant\_bookable\_t', 'accommodates', 'bathrooms\_text', 'beds', 'amenities', 'maximum\_nights', 'maximum\_minimum\_nights', 'maximum\_maximum\_nights', 'availability\_30', 'availability\_365', 'number\_of\_reviews\_ltm', 'number\_of\_reviews\_l30d', 'calculated\_host\_listings\_count\_entire\_homes', 'calculated\_host\_listings\_count\_private\_rooms', 'calculated\_host\_listings\_count\_shared\_rooms', 'reviews\_per\_month']]*

*Y=cost\_df[‘price\_PT’]*

*X\_train, X\_test, Y\_train, Y\_test = train\_test\_split( X, Y, test\_size=0.2, random\_state=10)*

*Res\_model = fin\_model.fit(X\_train,Y\_train)*

*Y\_pred = Res\_model.predict(X\_test)*

*print(“Overall R2 for the CATBoost Model for train data : ”,*

*np.round(Res\_model.score(X\_train,Y\_train),2))*

*print(“Overall R2 for the CATBoost Model for test data : ”,*

*np.round(Res\_model.score(X\_test,Y\_test),2))*

*print(“Root mean squared error for the CATBoost Model: ”,*

*np.round(mean\_squared\_error(Y\_test,Y\_pred,squared=False),2))*

*print(“Mean absolute error for the CATBoost Model: ”,*

*np.round(mean\_absolute\_error(Y\_test,Y\_pred),2))*

*print(“Mean absolute percentage error for the CATBoost Model: ”,*

*np.round(mape(Y\_test,Y\_pred),2))*

*print('-------------------------------------------------------------------------------','\n')*

*df\_feature = pd.DataFrame( {'Features':Xtrain.columns,'Coefficient':fin\_model.feature\_importances\_})*

*df\_feature.sort\_values('Coefficient',ascending=False,inplace=True)*

*print(“The Significant variable and the coefficient are: ”,df\_feature)*

*plt.figure(figsize=(15,8))*

*sns.barplot(data=df\_feature,x='Features',y='Coefficient')*

*plt.title('Coefficient of Variable')*

*plt.xticks(rotation=90)*

*plt.show()*

*k=KFold(n\_splits=5)*

*score=cross\_val\_score(estimator=fin\_model,X=X,y=Y,cv=k,scoring='r2',n\_jobs=-1)*

*print("Mean Score : ",np.mean(scores))*

*print("Bias error : ",(1-np.mean(scores))\*100)*

*print("Variance error : ",(np.std(scores)/np.mean(scores))\*100,'\n')*

*print('-------------------------------------------------------------------------------','\n')*

*fin\_model=* *CatBoostRegressor(depth= 8, iterations= 900, learning\_rate= 0.1, l2\_leaf\_reg=2, random\_state=10)*

*CostPrediction\_CATBoostModel(fin\_model)*

**Output:**

Overall R2 for the CATBoost Model for train data : 0.86

Overall R2 for the CATBoost Model for test data : 0.75

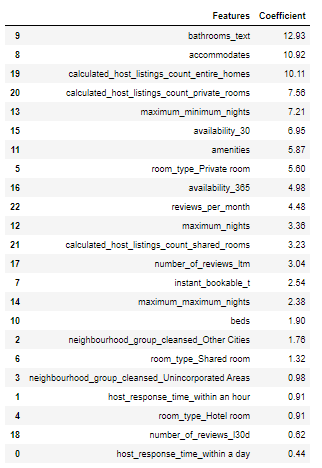
Root mean squared error for the CATBoost Model: 0.50

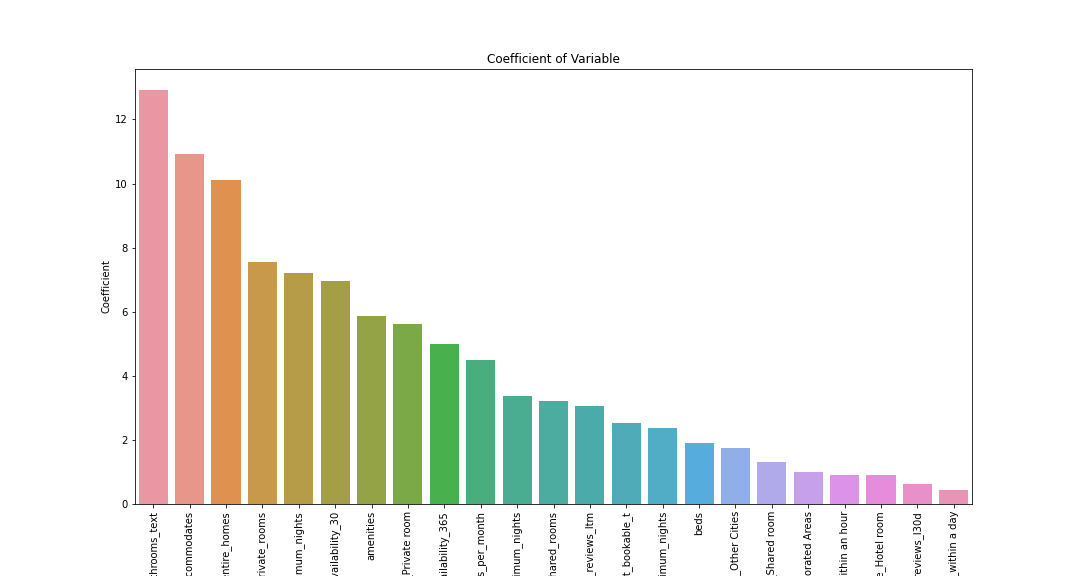
Mean absolute error for the CATBoost Model: 0.36

Mean absolute percentage error for the CATBoost Model: 139.78

---------------------------------------------------------------------------------------------------------------

The Significant variable and the coefficient are:



****

Mean Score: 0.686833741595677

Bias error: 31.316625840432298

Variance error: 7.103604294714113

**INFERENCES AND RECOMMENDATIONS:**

Inference – Some of the important features are bathrooms\_text (no\_of\_bathrooms), accomdates,calculated\_host\_listings\_count\_entire\_homes,calculated\_host\_listings\_count\_private\_homes, maximum\_minimum\_nights, availability\_30, amenities,room\_type\_private, reviews\_per\_month, beds and etc.

1. The no of accomdates helps to predict and also increase the cost of the property for a night.
2. The no of bathrooms tends to predict and increase the cost of the property for a night.
3. The amenities tends to predict and the increase of the property for a night.
4. Types of rooms – entire home has high price for the property for a night.
5. Types of rooms – private rooms has lesser price than compare to entire homes of the property for a night.
6. The maximum and minimum night’s feature helps to predict the price of the property for a night.
7. Previous number of reviews helps to predict the price of the property for a night.
8. Instant\_Bookable feature helps to predict the price of the property for a night.
9. The more cleaner neighbourhood tends to increase the price of the property for a night.

**BUSINESS INFERENCE:**

* Based on the type of rooms – entire home/type, its amenities, no of beds and no of bathrooms and no of accomdates are the top features that helps to predict the price of the property for a night.
* Despite of the property size and its amenities, if the property is located at the cleaner neighbourhood that tends to increase the price of the property for a night.
* Also the features like instant bookable and no of reviews helps to predict the price of the property.

**LIMITATIONS, CHALLENGES AND SCOPE**

**Limitations of Data**

The dataset belongs to properties from the location of Los Angeles, USA. The model will be more robust if the data would have belonged from different regions of the USA.

**Challenges**

1. High cardinality results in huge training effort in model tuning due to increase in model complexity (i.e. more number of features)
2. We also faced challenges on robust model tuning on all the models. Due to computational limitations, we are limited to very limited hyper parameters tuning technique using Grid Search.

**Scope**

1. We can approach the same concepts, using PCA as more than 60% of features having multicollinearity. Explanation of the each feature contribution’s is not feasible.
2. Can perform much more hyper parameter tuning for the CatBoost model. Due to lower processing power of our laptops, we couldn’t approach that.
3. Exploring Google collab as an option for model training and tuning with faster lead time.
4. Exploring some robust data sampling technique as part of choosing smaller sample (a true representation of population data) from the population data.