CAPSTONE PROJECT

INTERIM PRESENTATION REPORT

Batch details	PGPDSE CHN Jan 2023 (Group - 6)
Team members	1. Kousalya R 2. Mohammed Abrar Z 3. Ananthasayanan 4. Sadheesh Kumar E 5. Hrishikesh K 6. Faheem Anwar
Domain of Project	Travel and Tourism Industry
Project Title	Instant Booking Eligibility Prediction
Group Number	Group 6
Team Leader	Kousalya R
Mentor Name	Ms. Vibha Santhanam

Date: 15.05.2023

Signature of the Mentor

(Ms. Vibha Santhanam)

TABLE OF CONTENTS

OVERVIEW

Travel and tourism industry is one of the world's great industrial sectors. It drives economic growth, creates jobs, improves social development and promotes peace. Hundreds of millions of people around the world are dependent on the sector for their employment.

A classification model can provide valuable insights for the traveling industry and help travel companies to improve their customer experience, marketing and promotional efforts, fraud prevention measures, inventory management, and safety and security measures.

We are Building a classification model that assists hosts in determining the instant bookability of their property on an online booking platform. The model will help hosts improve the performance, pricing, and amenities of their property to increase the likelihood of it being instantly bookable

BUSINESS PROBLEM STATEMENT

Business Problem Statement:

To build a classification model to assist hosts in identifying whether their property will be instantly bookable or not on an online booking platform, based on the details of the property name, amenities, location, services etc. The objective of the model is, it should be able to accurately classify properties as either instantly bookable or not, to help hosts for the better management of their availability and maximize their booking potential.

Business Problem Understanding:

The main challenge for hosts is to optimize their bookings and revenue potential, and instant booking eligibility plays a crucial role in this regard. However, the eligibility criteria can vary across booking platforms and can be complex, making it difficult for hosts to understand and improve their eligibility.

This will help hosts make informed decisions about areas to improve so that Host bookability status can be positive, and will also help Airbnb improve the user experience for guests by providing them with more options for instant bookable properties. Ultimately, this will help Airbnb increase its booking volume and improve its market position.

Business Objective:

The primary business objective of Airbnb is to provide a platform for individuals to rent out their properties to travelers seeking short-term accommodations. Airbnb aims to connect hosts and guests from around the world and to offer unique and personalized travel experiences. It mainly focuses on Growing the platform, Improving the user experience, generating revenue and Building trust and safety. Overall, the business objective of Airbnb is to continue to grow and innovate in the travel industry while promoting a sense of community and cultural exchange among its users.

DATA DESCRIPTION

As our data set contains over 33 variables, we are only including an extract of the data dictionary here. Please find the complete data dictionary at the following link:

https://www.kaggle.com/datasets/ulrikthygepedersen/airbnb-listings?select=Listings.csv

Serial no.	Variable Name	Variable description
1	`listing_id`	A unique identifier for each property listing.
2	`name`	The name or title of the property listing.
3	`host_id`	A unique identifier for the host who owns the property.
4	`host_since`	The date when the host joined the platform.
5	`host_location`	The location of the host.
6	`host_response_time`	The average time it takes for the host to respond to booking requests.
7	`host_response_rate`	The percentage of booking requests that the host responds to.
8	`host_acceptance_rate`	The percentage of booking requests that the host accepts.
9	`host_is_superhost`	A binary indicator of whether the host has been awarded "superhost" status.
10	`host_total_listings_count`	The total number of properties that the host has listed on the platform.
11	`host_has_profile_pic`	A binary indicator of whether the host has uploaded a profile picture.
12	`host_identity_verified`	A binary indicator of whether the host's identity has been verified by
13	`neighborhood`	The neighborhood or district where the property is located.
14	`district`	The borough or region where the property is located.
15	`city`	The city where the property is located.
16	`latitude`	The latitude of the property's location.
17	`longitude`	The longitude of the property's location.
18	`property_type`	The type of property (e.g. apartment, house, etc.).
19	`room_type`	The type of room that is being listed (e.g. entire home, private room, shared room,etc.).
20	`accommodates`	The maximum number of guests that the property can accommodate.
21	`bedrooms`	The number of bedrooms in the property.
22	`amenities`	A list of amenities that are included with the property.
23	`price`	The nightly price of the property.
24	`minimum_nights`	The minimum number of nights that a guest must book.

25	`maximum_nights`	The maximum number of nights that a guest can book.
26	`review_scores_rating`	The overall rating score given by guests who have stayed at the property.
27	`review_scores_accuracy`	The rating score given by guests for the accuracy of the property description.
28	`review_scores_cleanliness`	The rating score given by guests for the cleanliness of the property.
29	`review_scores_checkin`	The rating score given by guests for the check-in process.
30	`review_scores_communication`	The rating score given by guests for the host's communication.
31	`review_scores_location`	The rating score given by guests for the property's location.
32	`review_scores_value`	The rating score given by guests for the value of the property.
33	`instant_bookable`	A binary indicator of whether the property can be booked instantly without the host's approval. We assume it as a predictor. If the property has good features then it is instantly bookable. Else it is not instantly bookable.

Information:

The dataset has 279712 rows and 33 columns

```
1 df_airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 279712 entries, 0 to 279711
Data columns (total 33 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
 0   listing_id                    279712 non-null  int64
 1   name                          279539 non-null  object
 2   host_id                       279712 non-null  int64
 3   host_since                    279547 non-null  object
 4   host_location                 278872 non-null  object
 5   host_response_time            150930 non-null  object
 6   host_response_rate            150930 non-null  float64
 7   host_acceptance_rate          166625 non-null  float64
 8   host_is_superhost             279547 non-null  object
 9   host_total_listings_count     279547 non-null  float64
 10  host_has_profile_pic          279547 non-null  object
 11  host_identity_verified        279547 non-null  object
 12  neighbourhood                 279712 non-null  object
 13  district                      37012 non-null   object
 14  city                          279712 non-null  object
 15  latitude                      279712 non-null  float64
 16  longitude                     279712 non-null  float64
 17  property_type                 279712 non-null  object
 18  room_type                     279712 non-null  object
 19  accommodates                  279712 non-null  int64
 20  bedrooms                      250277 non-null  float64
 21  amenities                     279712 non-null  object
 22  price                         279712 non-null  int64
 23  minimum_nights                279712 non-null  int64
 24  maximum_nights                279712 non-null  int64
 25  review_scores_rating          188307 non-null  float64
 26  review_scores_accuracy        187999 non-null  float64
 27  review_scores_cleanliness     188047 non-null  float64
 28  review_scores_checkin         187941 non-null  float64
 29  review_scores_communication   188025 non-null  float64
 30  review_scores_location        187937 non-null  float64
 31  review_scores_value           187927 non-null  float64
 32  instant_bookable              279712 non-null  object
dtypes: float64(13), int64(6), object(14)
memory usage: 70.4+ MB
```

There are totally 14 categorical variables and 19 numerical variables in this dataset.

DATA CLEANING:

Null value Treatment:

	Count of Missing values	Percentage of Missing values
name	173	0.061849
host_since	165	0.058989
host_location	840	0.300309
host_response_time	128782	46.040928
host_response_rate	128782	46.040928
host_acceptance_rate	113087	40.429799
host_is_superhost	165	0.058989
host_total_listings_count	165	0.058989
host_has_profile_pic	165	0.058989
host_identity_verified	165	0.058989
district	242700	86.767818
bedrooms	29435	10.523324
review_scores_rating	91405	32.678255
review_scores_accuracy	91713	32.788368
review_scores_cleanliness	91665	32.771208
review_scores_checkin	91771	32.809104
review_scores_communication	91687	32.779073
review_scores_location	91775	32.810534
review_scores_value	91785	32.814109

Dropped Columns:

There are 70% of null values in the **"District"** column so it can be dropped.

The variables like **host_location, latitude, longitude** and city convey the same purpose of defining the location. So we can keep 'City' column and dropped others.

The columns property_type and room_type conveys the same meaning of describing the property. On moving forward with analysis we drop **property_type** variable.

For reviews there are variables like review_scores_accuracy, review_scores_cleanliness, review_scores_checkin, review_scores_communication, review_scores_location, review_scores_value and combining all these there is an overall review_rating variable. On moving forward we can keep only review_scores_rating and dropping other variables.

Imputing missing values:

Host_response_time:

For host_response_time variable values for a new customer will be considered as unknown and for null values of existing customer we could impute it with the same unknown. Hence we can impute all null values with unknown.

host_acceptance_rate:

For variable like host_acceptance_rate values for a new customer will be considered as zero and for null values of existing customer we could impute it with zero. We can impute null values with zero.

host_response_rate:

For host_response_rate values for a new customer will be considered as zero and for null values of existing customer we could impute it with zero. We can impute null values with zero.

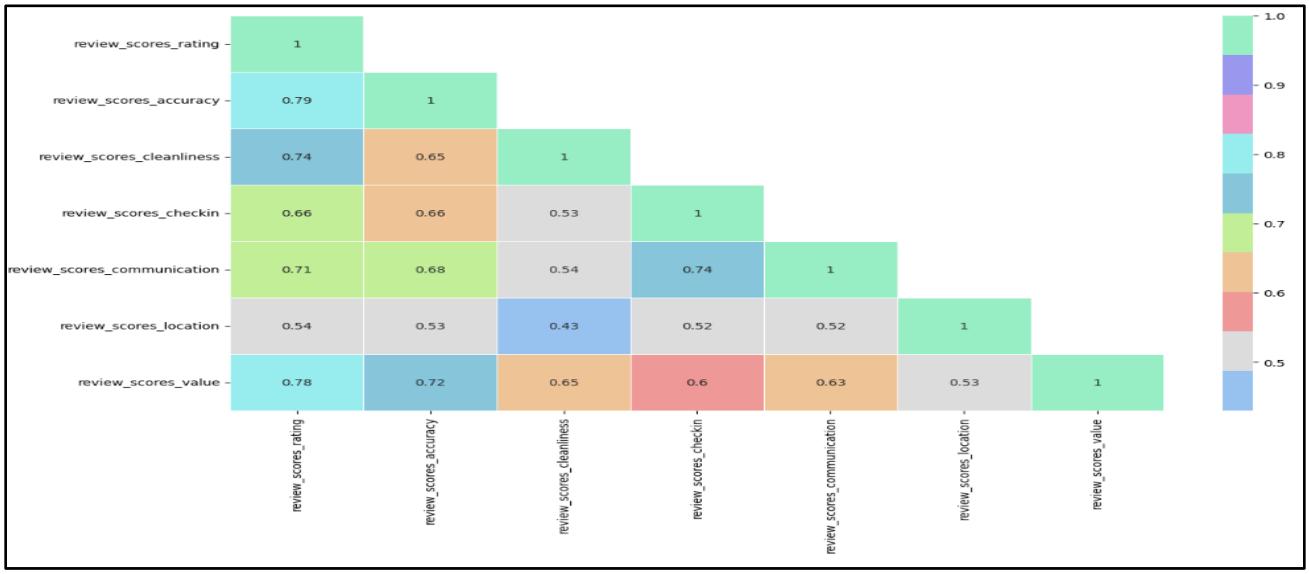
host_has_profile_pic,host_is_superhost,host_identity_verified:

For variables like host_has_profile_pic,host_is_superhost,host_identity_verified we can impute it with false as it will be in default false.

Bedrooms:

To impute in bedrooms variable. We can take do groupby of room_type and take median of bedrooms for each room_type and can impute with that median values.

Review Scores Rating:



Since there is a corelation between review score rating and other reviews column it can be imputed using linear model, we can impute review_scores_rating variable we can create a linear regression model which will predict the review scores with the help of variables like review_scores_accuracy,review_scores_cleanliness,review_scores_checkin, review_scores_communication, review_scores_location and review_scores_value. Review feature is imputed using linear regression model.

Descriptive Statistics:

Object:

	count	unique	top	freq
host_since	279547	4240	02-09-2019	710
host_response_time	279547	5	unknown	128617
host_is_superhost	279547	2	f	229294
host_has_profile_pic	279547	2	t	278631
host_identity_verified	279547	2	t	201191
neighbourhood	279547	660	I Centro Storico	14869
city	279547	10	Paris	64657
room_type	279547	4	Entire place	181886
amenities	279547	244872	["Long term stays allowed"]	1385
instant_bookable	279547	2	f	163995
Cluster	279547	4	West	75567

From above statistics we get to know:

- There are total 279547 rows and 10 unique classes in city. In that majority class with a count of 64657 is Paris.
- Variable neighborhood is having 660 unique values. Out of 660 unique districts majority of the property is located in I Centro Storico and their count is 14869
- Property of room_type Entire place is highest among the other 3 room_type classes with a count of 181886
- In the target variable we can see it is a binary class and there is no class imbalance. Majority of the property has instant bookability as false with a count of 163995

Number:

	count	mean	std	min	25%	50%	75%	max
host_response_rate	279547.0	0.467528	4.793091e-01	0.0	0.0	0.140000	1.000000	1.000000e+00
host_acceptance_rate	279547.0	0.493037	4.632412e-01	0.0	0.0	0.590000	1.000000	1.000000e+00
host_total_listings_count	279547.0	11.571296	4.666035e+01	1.0	1.0	1.000000	4.000000	6.270000e+02
accommodates	279547.0	3.288731	2.133470e+00	0.0	2.0	2.000000	4.000000	1.600000e+01
bedrooms	279547.0	1.461239	1.102258e+00	1.0	1.0	1.000000	2.000000	5.000000e+01
price	279547.0	608.884635	3.442785e+03	0.0	75.0	150.000000	474.000000	6.252160e+05
minimum_nights	279547.0	8.051998	3.152749e+01	1.0	1.0	2.000000	5.000000	9.999000e+03
maximum_nights	279547.0	27574.425889	7.285024e+06	1.0	45.0	1125.000000	1125.000000	2.147484e+09
review_scores_rating	279547.0	72.384389	3.982501e+01	0.0	72.0	93.648854	98.758926	1.000000e+02

From above statistics we get to know:

- Average of price of properties listed is $608
- Minimum nights allowed to stay is one night
- Maximum people allowed to stay is 16 members
- Size of bedrooms range from room type, minimum rooms available is 1 and maximum is 50

STATISTICAL ANALYSIS

The statistical test shows that all the variables are significant. Which means that the sample taken from the population is significant and meaningful.

	Columns	Pvalue	Remarks
0	host_response_rate	0.000000e+00	Reject H0
1	host_acceptance_rate	0.000000e+00	Reject H0
2	host_total_listings_count	0.000000e+00	Reject H0
3	latitude	1.218462e-01	Failed to reject H0
4	longitude	1.115112e-134	Reject H0
5	accommodates	1.290423e-06	Reject H0
6	bedrooms	4.834544e-18	Reject H0
7	price	5.595255e-13	Reject H0
8	minimum_nights	1.209669e-123	Reject H0
9	maximum_nights	6.246272e-01	Failed to reject H0
10	review_scores_rating	1.805145e-184	Reject H0
11	review_scores_accuracy	1.411651e-43	Reject H0
12	review_scores_cleanliness	3.888389e-21	Reject H0
13	review_scores_checkin	7.071339e-35	Reject H0
14	review_scores_communication	3.402854e-39	Reject H0
15	review_scores_location	1.199631e-22	Reject H0
16	review_scores_value	2.815450e-37	Reject H0
17	host_response_time	0.000000e+00	Reject H0
18	host_is_superhost	3.887066e-79	Reject H0
19	host_has_profile_pic	3.101958e-01	Failed to reject H0
20	host_identity_verified	1.208082e-03	Reject H0
21	neighbourhood	0.000000e+00	Reject H0
22	city	0.000000e+00	Reject H0
23	property_type	0.000000e+00	Reject H0
24	room_type	0.000000e+00	Reject H0
25	amenities	1.482095e-185	Reject H0
26	Region	0.000000e+00	Reject H0

Hypothesis :

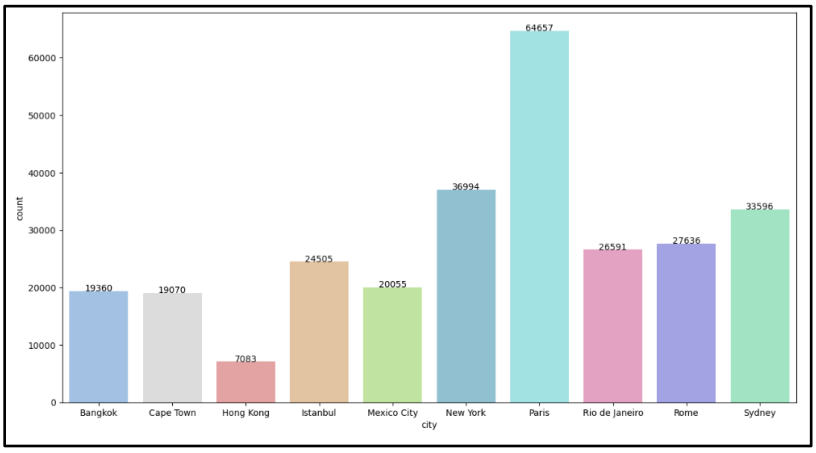
H0 (Null Hypothesis) : There is no significant relationship between the variables being tested.
Ha (Alternative Hypothesis) : There is a significant relationship between the variables being tested

Consider significance level as 0.05

EXPLORATORY DATA ANALYSIS

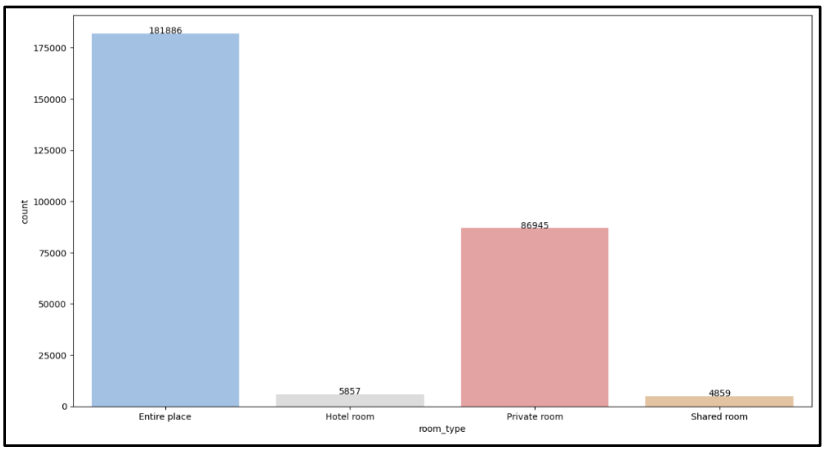
Univariate Analysis:

City:



From above countplot it is clearly evident that more properties are available in Paris of count 54657 followed by NewYork 36994 and Sydney 33596. Least property is in HongKong city with total properties of 7083. Also we can see that Bangkok and Cape Town almost has same number of properties available

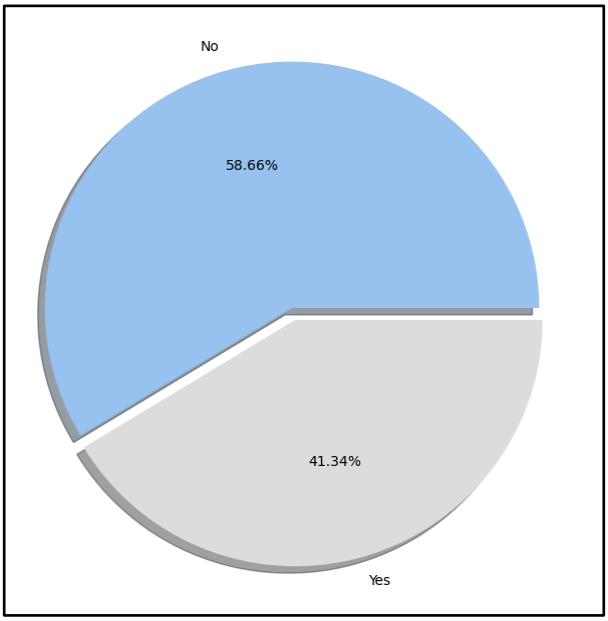
Room_type:



Room_type describes the type of room or space that is being offered for rent. The room_type variable can be used as a feature in data analysis or modeling to predict the likelihood of a booking, and to identify the factors that are most predictive of a guest's preference for a particular room type.

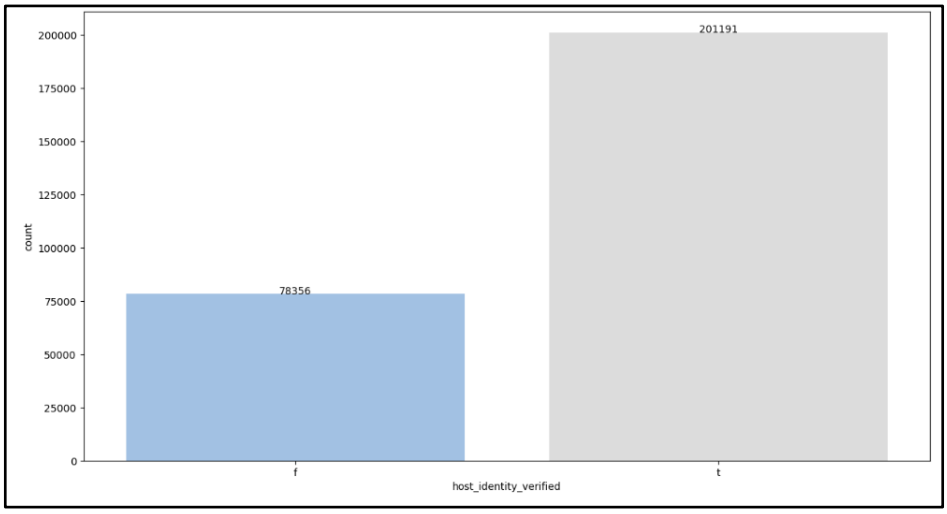
From above countplot we can infer that most of the properties available are Entire place with a count of 181886. Next to Entire place is Private room having 86945 properties. Comparatively Shared room and Hotel room share the same number of properties

Instant_bookable:



From above countplot and value_counts we can see that binary class is not in equal proportions but still it is not class imbalanced since the percentage of minority class is nearly 40%. Creating synthetic data using smote technique will not required.

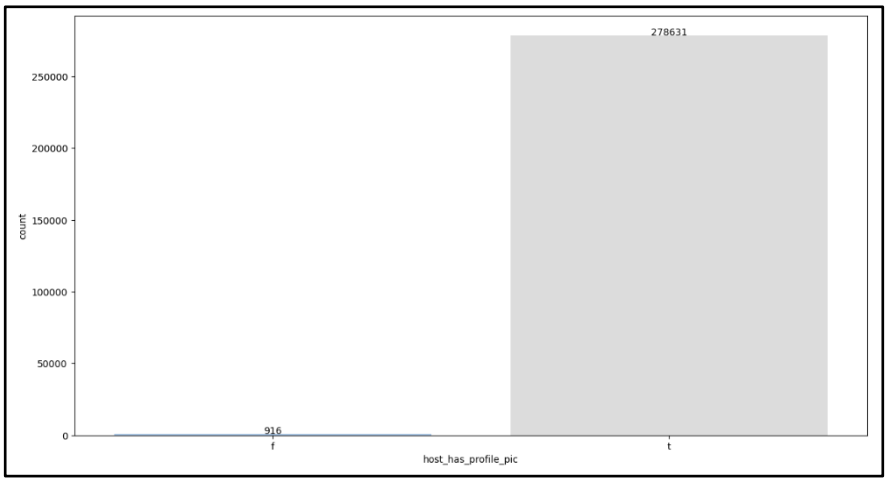
Host_identity_verified:



This variable shows whether the host's identity has been verified by Airbnb or not. This variable can be useful for guests to assess the trustworthiness and safety of a potential host. A host with a verified identity is generally perceived as more trustworthy and reliable, as it suggests that Airbnb

has confirmed their identity and has taken steps to ensure that they are who they claim to be. Majority of the properties posted we have host verified their idenity by Airbnb.

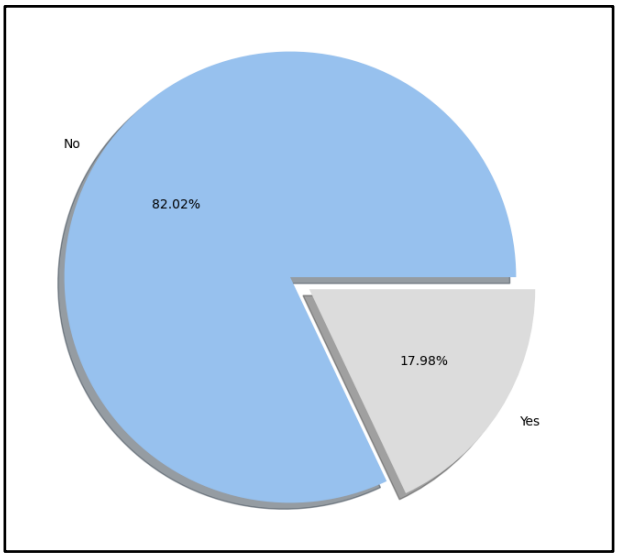
Host_profile_pic:



This variable shows whether the host has uploaded a profile picture or not. This variable can be useful for guests to assess the trustworthiness and professionalism of a potential host. A host with a profile picture is generally perceived as more trustworthy and reliable, as it suggests that they are invested in their Airbnb hosting and are more likely to be responsive to guests' needs.

From above countplot we get to know that majority of the host has updated their profile picture.

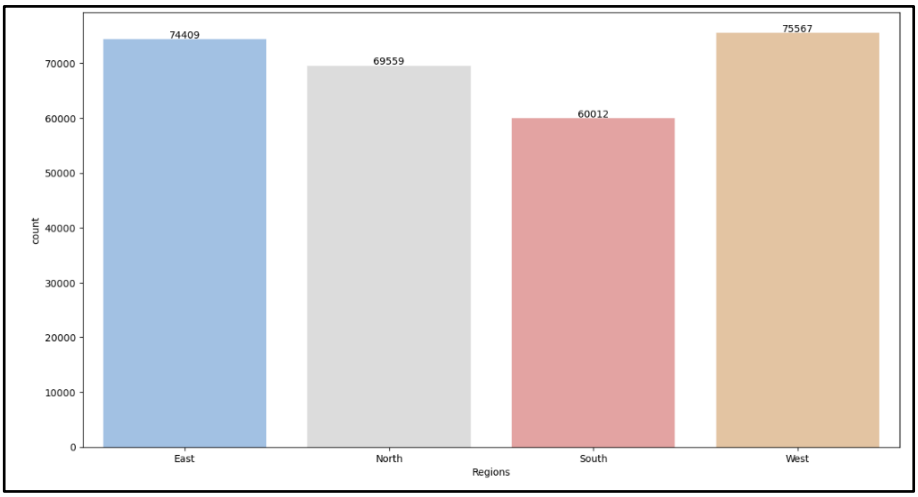
Host_is_superhost:



"Superhost" is a highly-rated and experienced host who provides exceptional hospitality to guests. The Superhost program is designed to recognize hosts who go above and beyond to provide great experiences for their guests and to help guests find the best possible hosts.

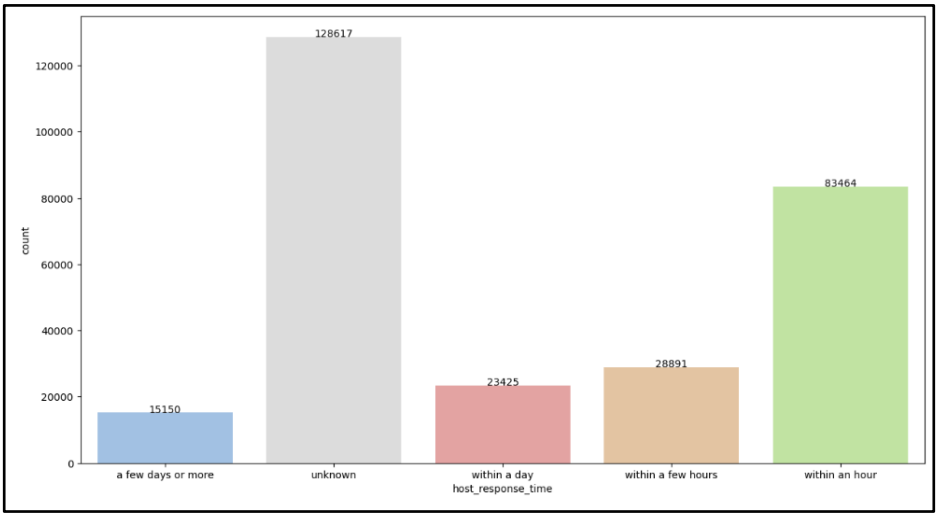
As we can see from above pie chart most of the host are not a superhost.

Regions:



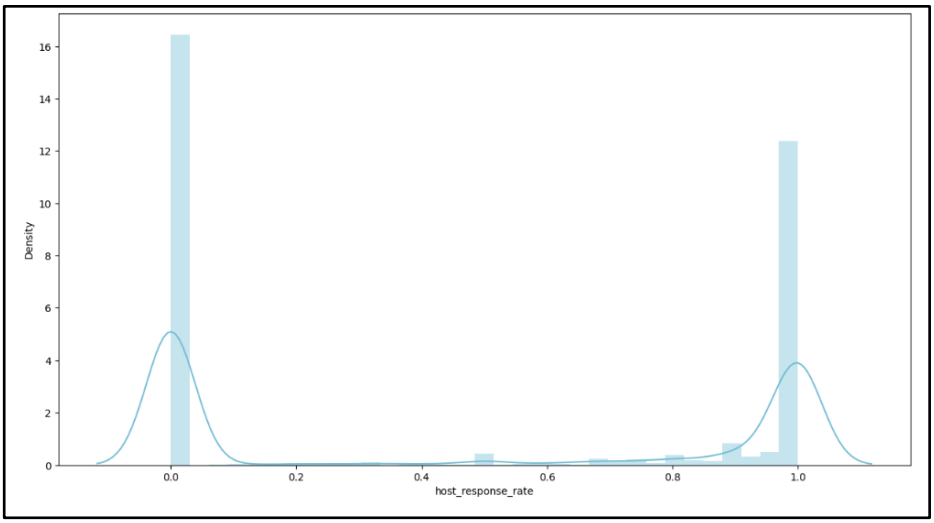
It shows whether the property lies in the northern, southern, Western or Eastern part of the city.

Host_response_time:



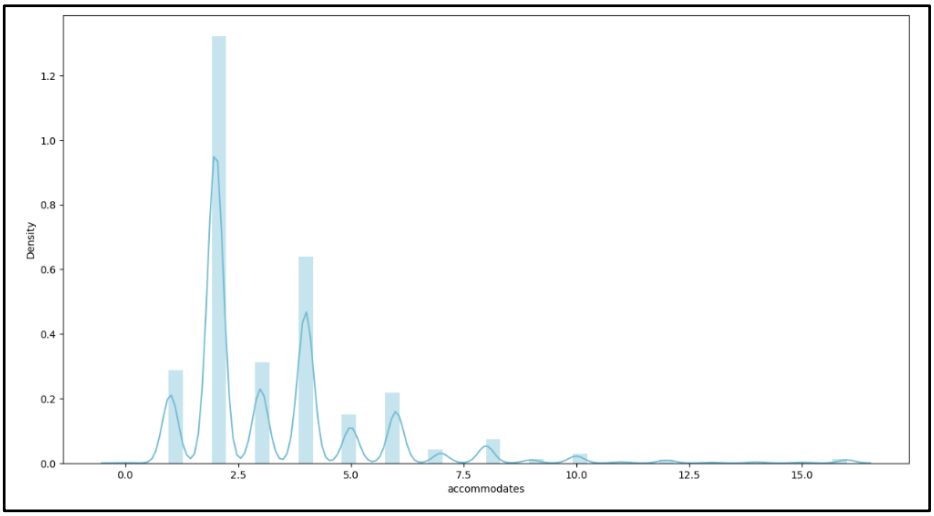
This variable indicates the percentage of enquiries or messages that a host responds to within a certain timeframe. And it is expressed in percentage, ranging from 0 to 100.

Host_response_rate:



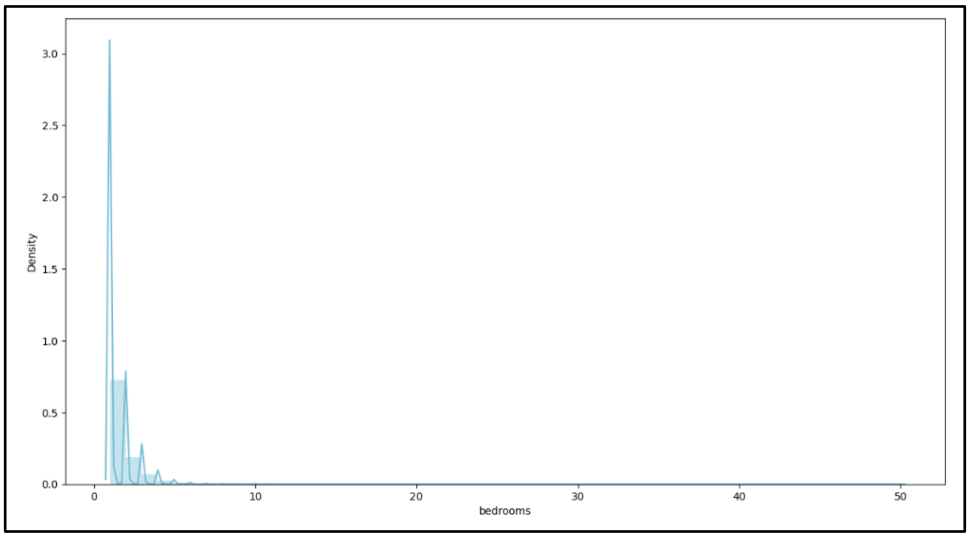
This variable indicates the percentage of booking requests that a host accepts. And it is expressed in percentage, ranging from 0 to 100.

Accomodates:



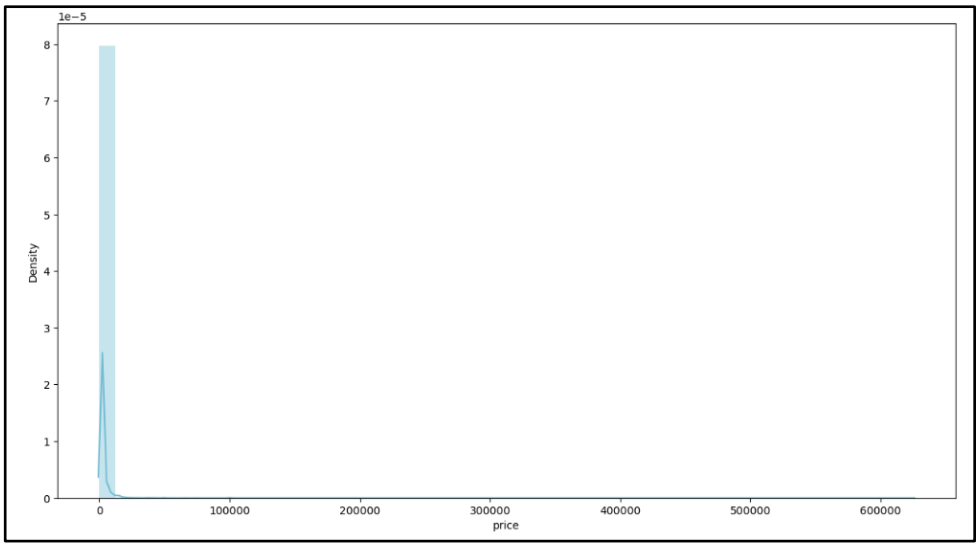
It indicates the maximum number of guests that a particular listing can accommodate.

Bedrooms:



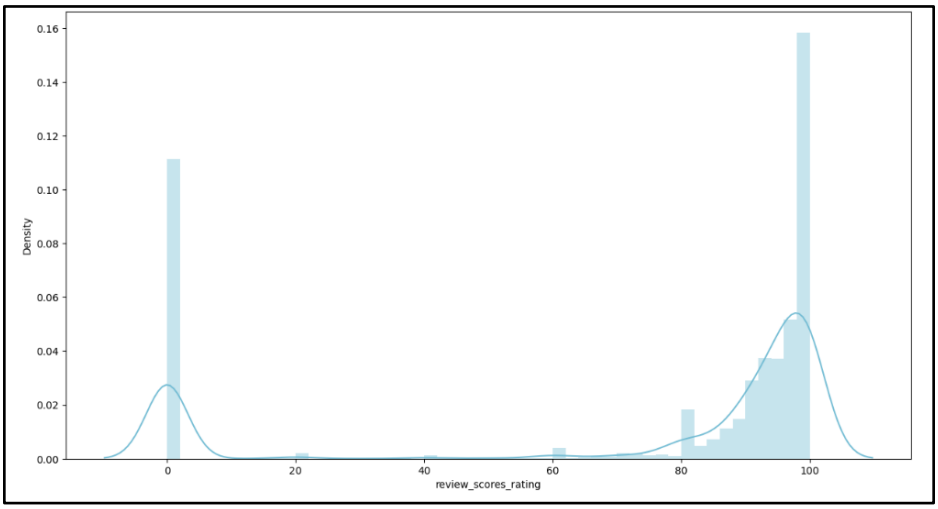
This variable indicates the number of bedrooms available in a particular listing.

Price:



It indicates the nightly price of a particular listing and it is often one of the primary factors that guests consider when choosing a listing to book.

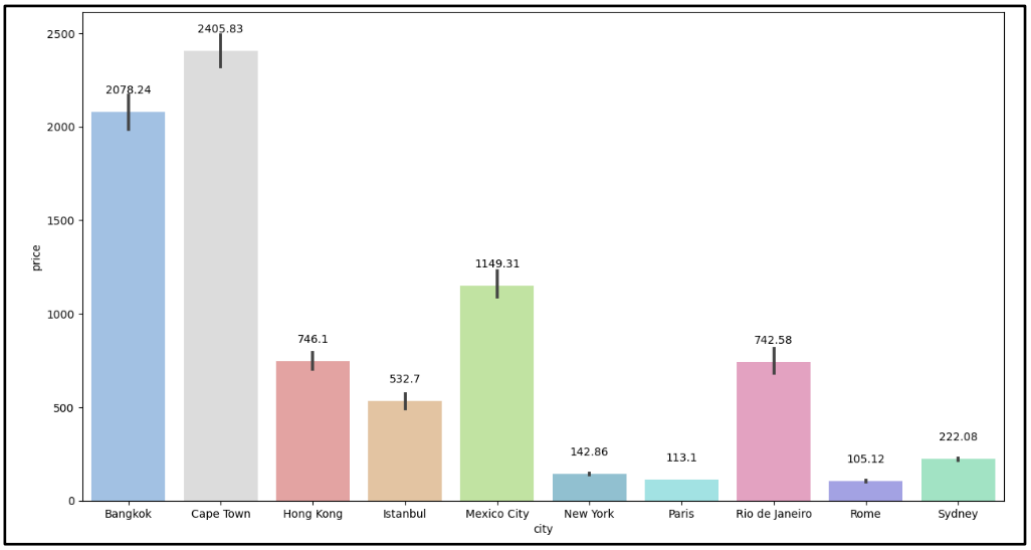
Review_scores_rating:



This variable indicates the overall satisfaction rating of guests who have stayed at a particular listing and it provides insight into the quality of the listing and the experiences of previous guests.

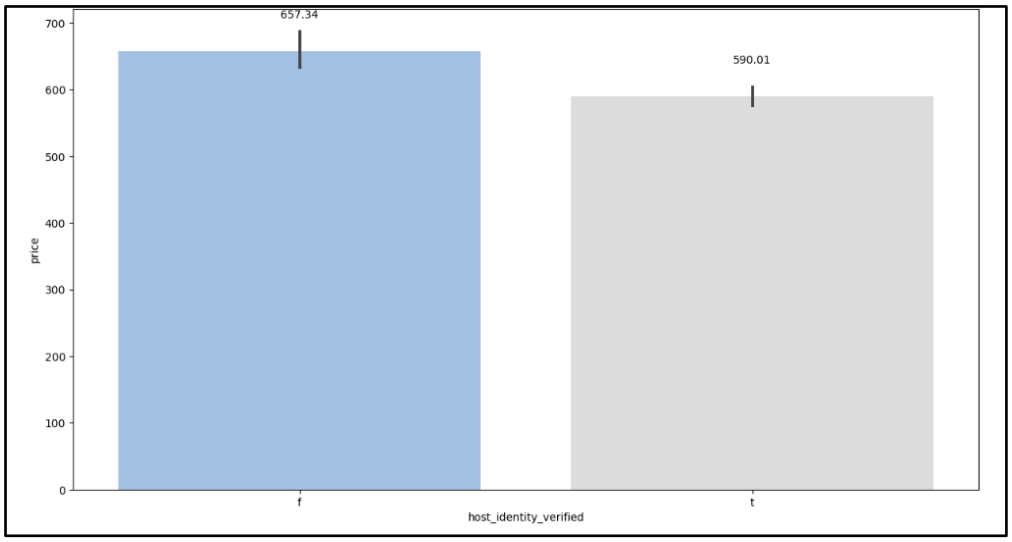
Bivariate Analysis:

City vs Price¶:



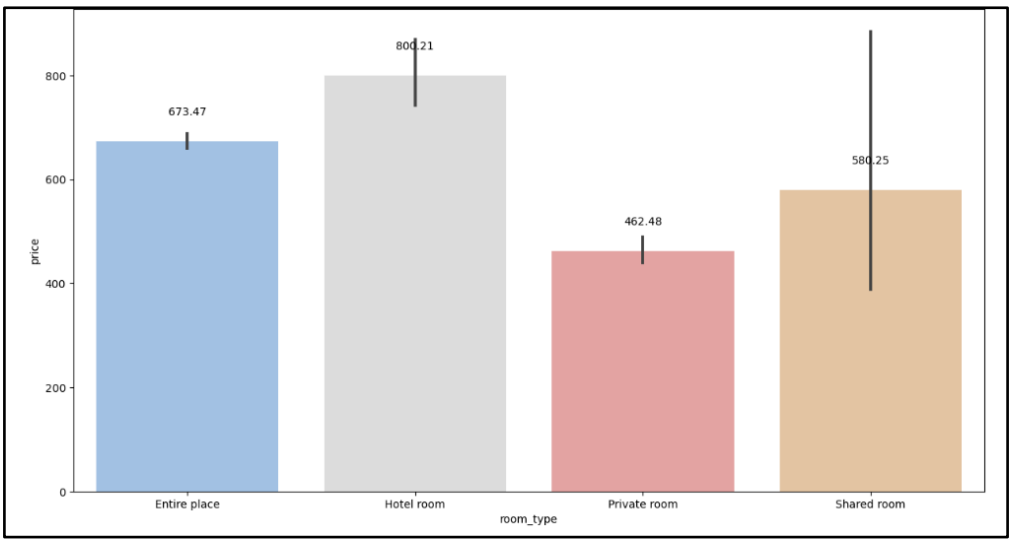
From the plot we can infer that Highest price is in Cape Town of 2405 per night followed by Bankgkok 2078 per night. Average price is in Mexico of 1149 and least price is in the cities of NewYork, Paris and Rome.

Host_identity_verified vs Price:



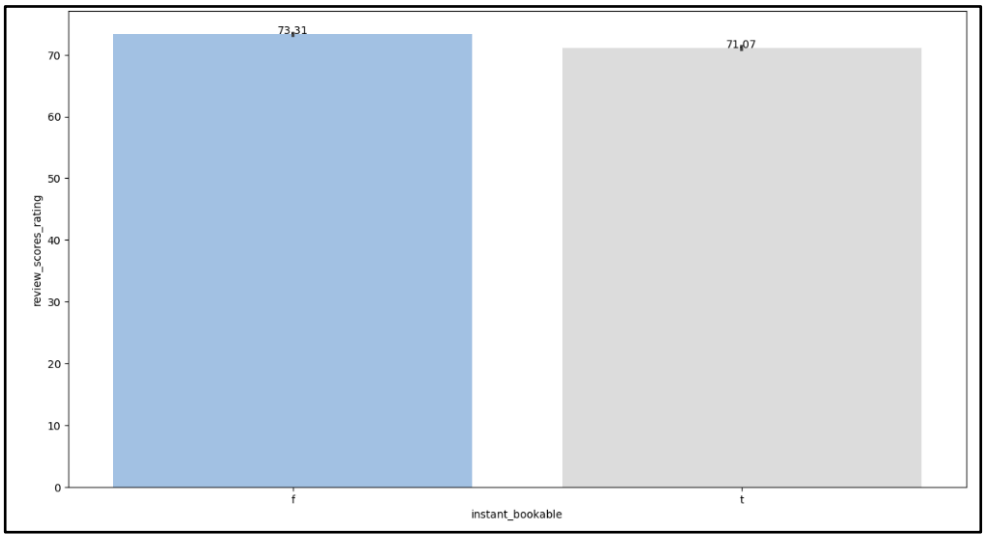
From above plot it is clearly evident that there is no much significal difference in price between having verified status or not. Price is almost same for both classes.

Room_type vs Price:



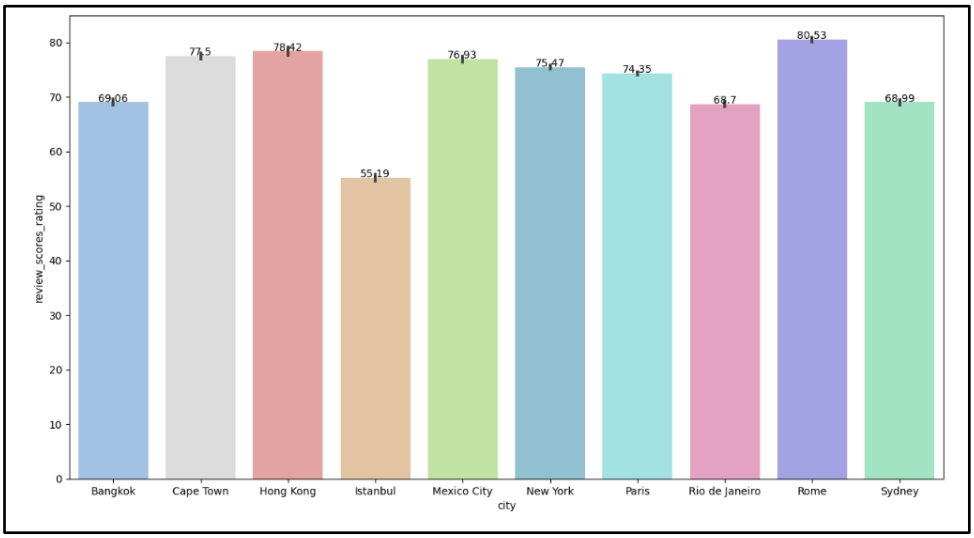
From above barplot we can see that price for Hotel room and Entire place is expensive compared to Private room and Shared room. The price of former is 800 & 673 and the latter is 462 & 580

Instant bookable vs Review score rating:



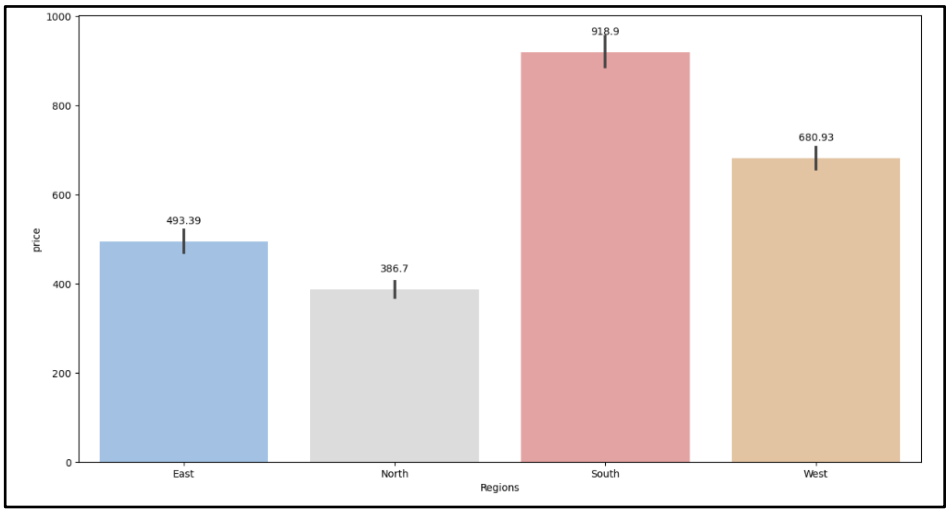
From above plot we get to know that there is not much significance difference in rating's irrespective of the instant bookable.

City vs Review score rating:



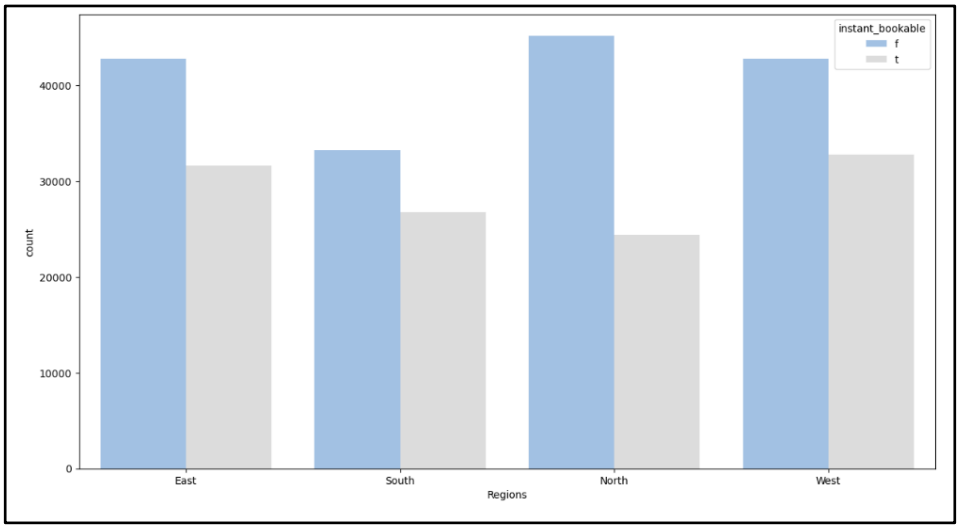
The above plot shows the distribution of rating's across different cities. We can see that there is no much diiference in the rating score among the cities.

Region vs price:



The above bar plot shows the distribution between the region and price. The mean of the price in the southside and westside is higher when compared to north and south.

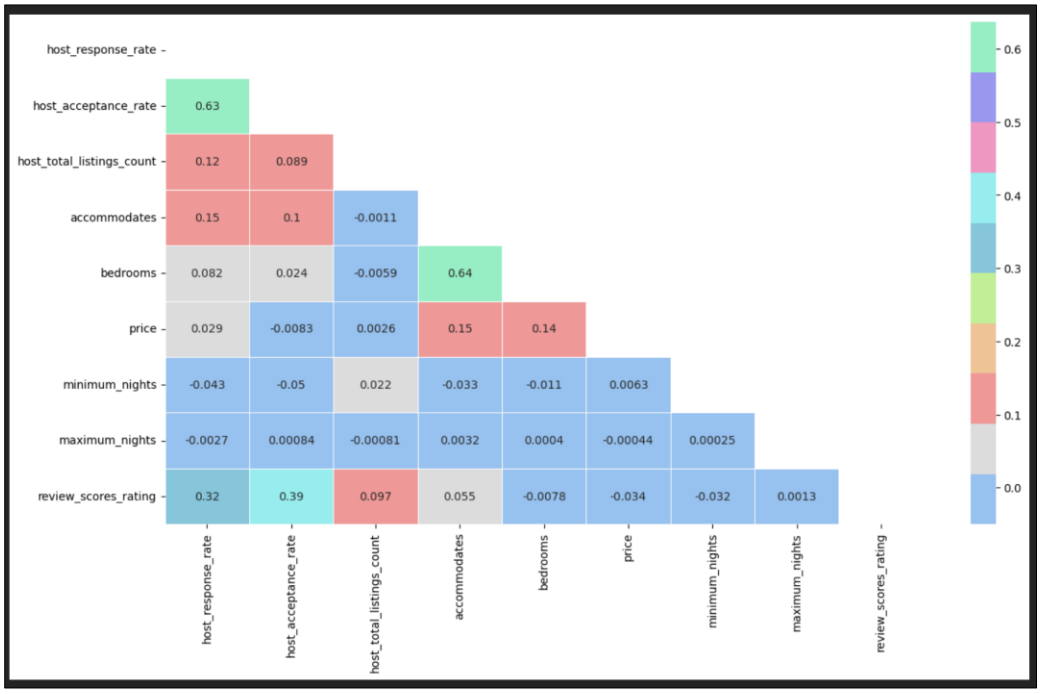
Region vs Instant_bookable:



The above bar plot shows the distribution between the region and instant_bookability.

Multivariate Analysis¶:

Checking correlation with heatmap:



Checking and treating of outliers:

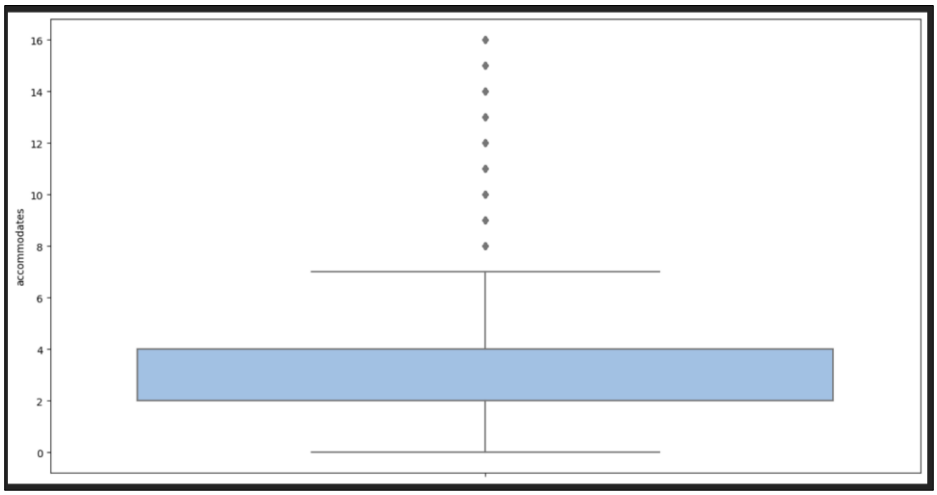
- ✓ **Outlier Treatment:**

- We have performed Power transform on all the numerical data and plotted a graph to check the skewness range before and after the transformation.
- For only features which showed better skewness reduction in data after power transform have been chosen for transformation techniques.
- For remaining features, we have planned to move further with the existing data itself.

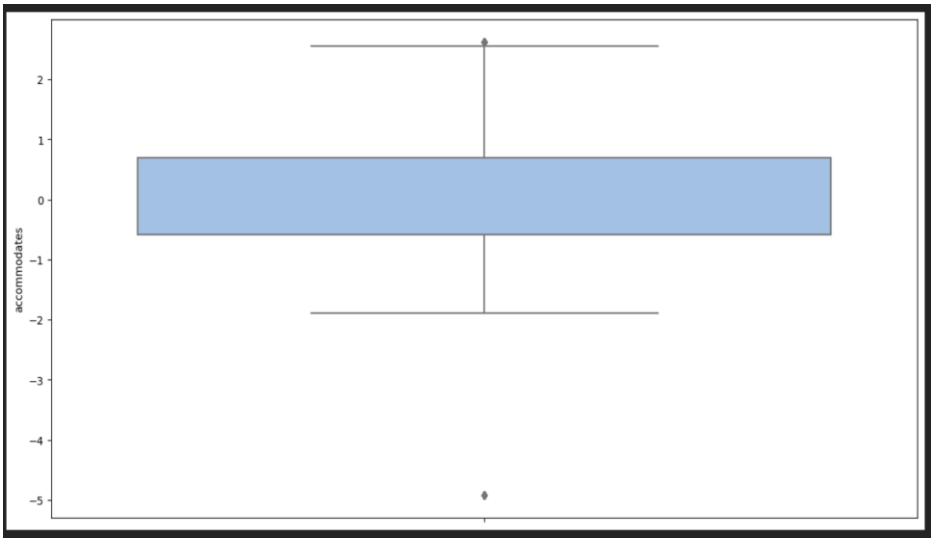
Accommodates:

From above plot it is clearly evident that there are outliers present. By doing IQR method we tend lose data. Hence we go forward by doing transformation technique.

Before Outlier Treatment:



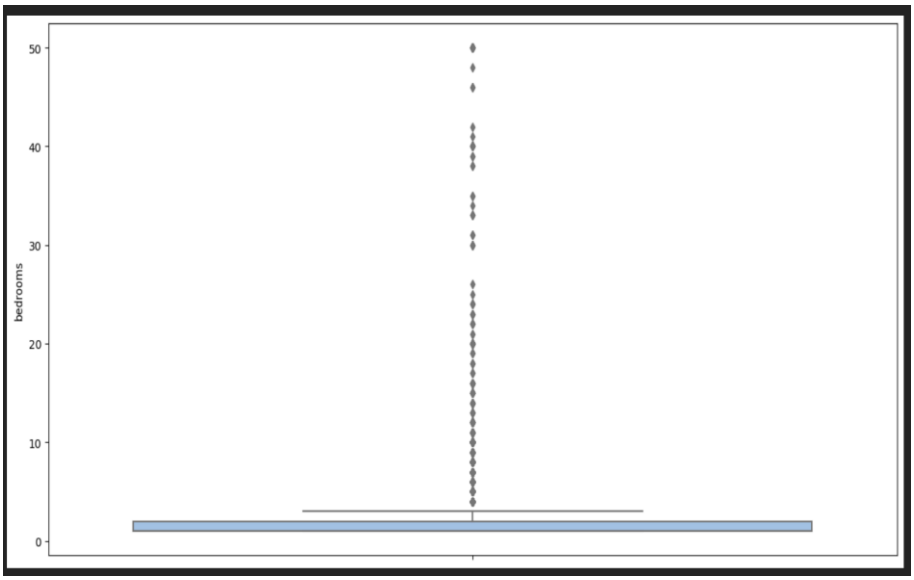
After Outlier Treatment:



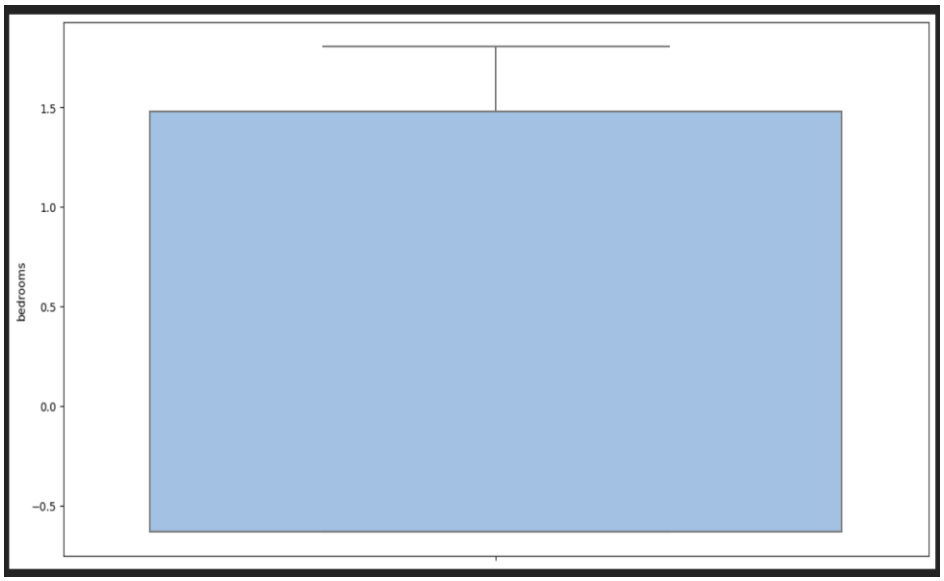
Bedrooms:

From above plot it is clearly evident that there are outliers present. By doing IQR method we tend lose data. Hence we go forward by doing transformation technique

Before Outlier Treatment:



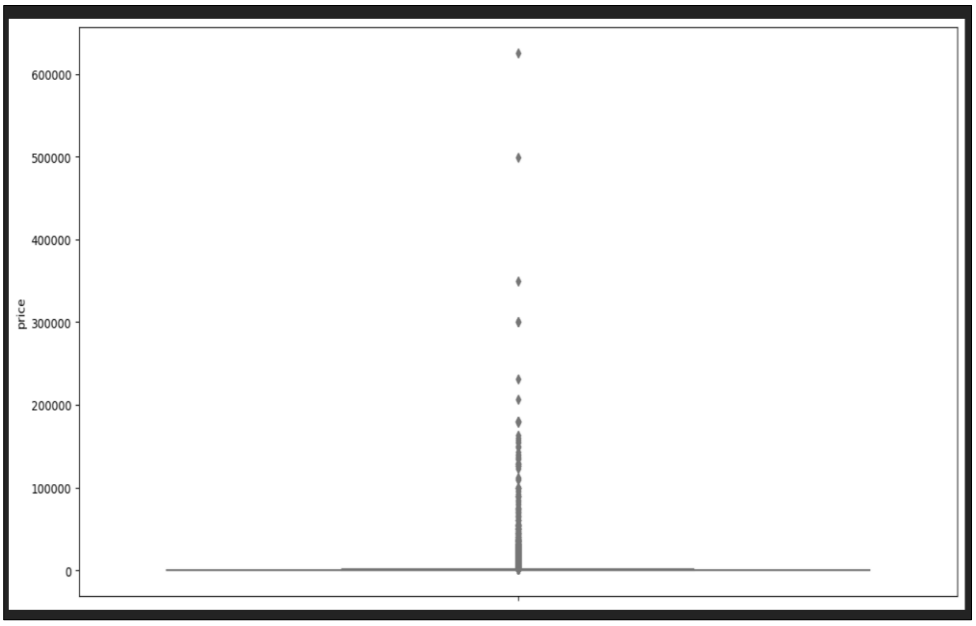
After Outlier Treatment:



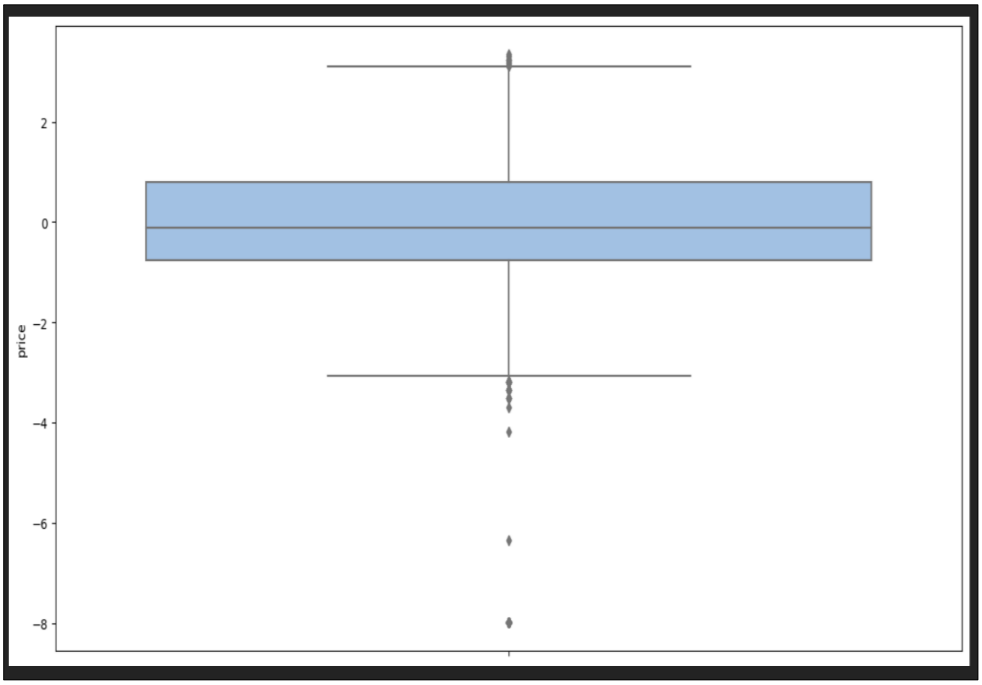
Price:

From above plot it is clearly evident that there are outliers present. By doing IQR method we tend lose data. Hence we go forward by doing transformation technique

Before Outlier Treatment:

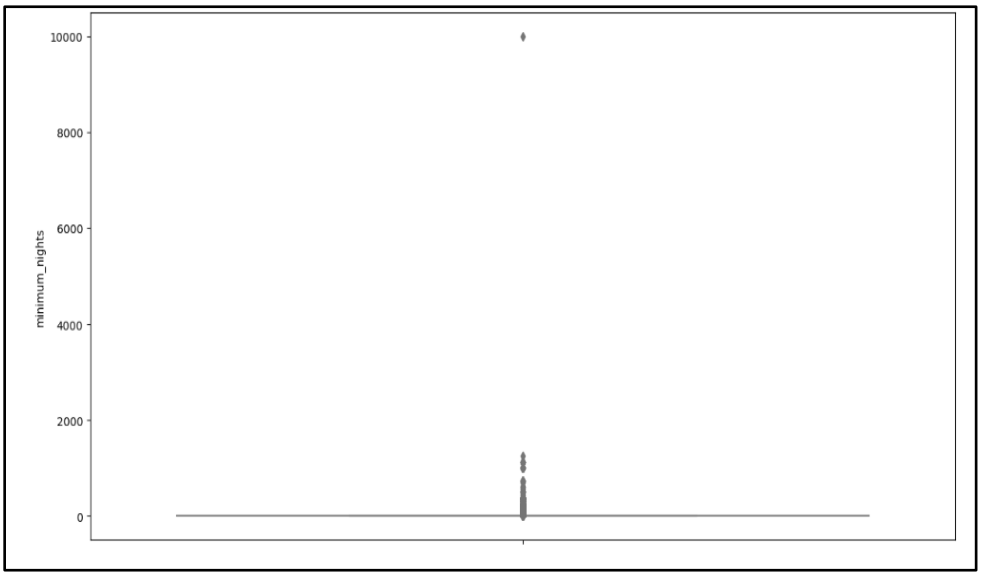


After Outlier Treatment:

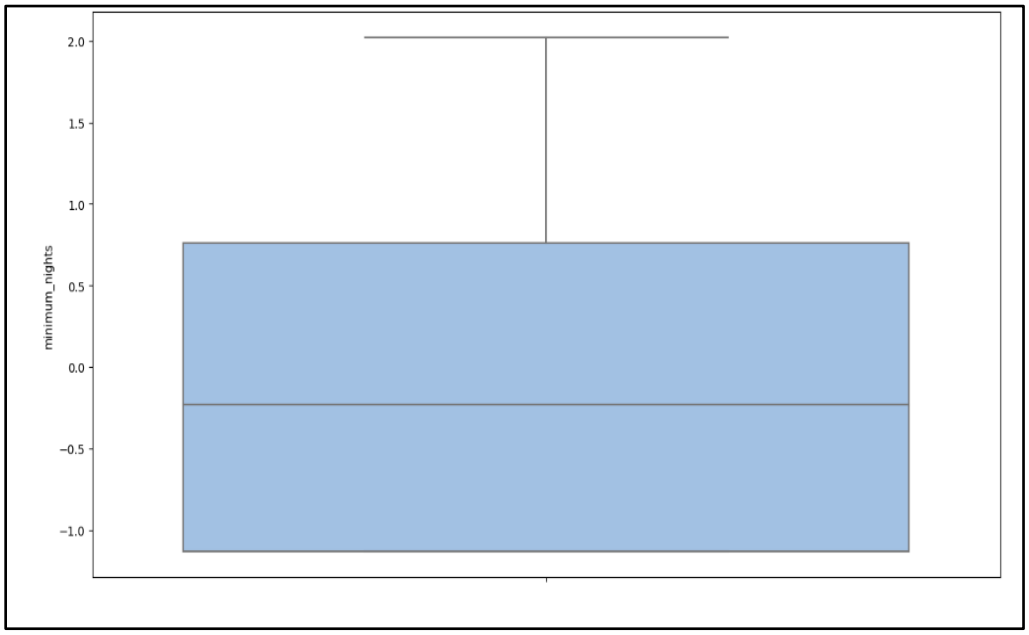


Minimum_nights:

Before Outlier Treatment:

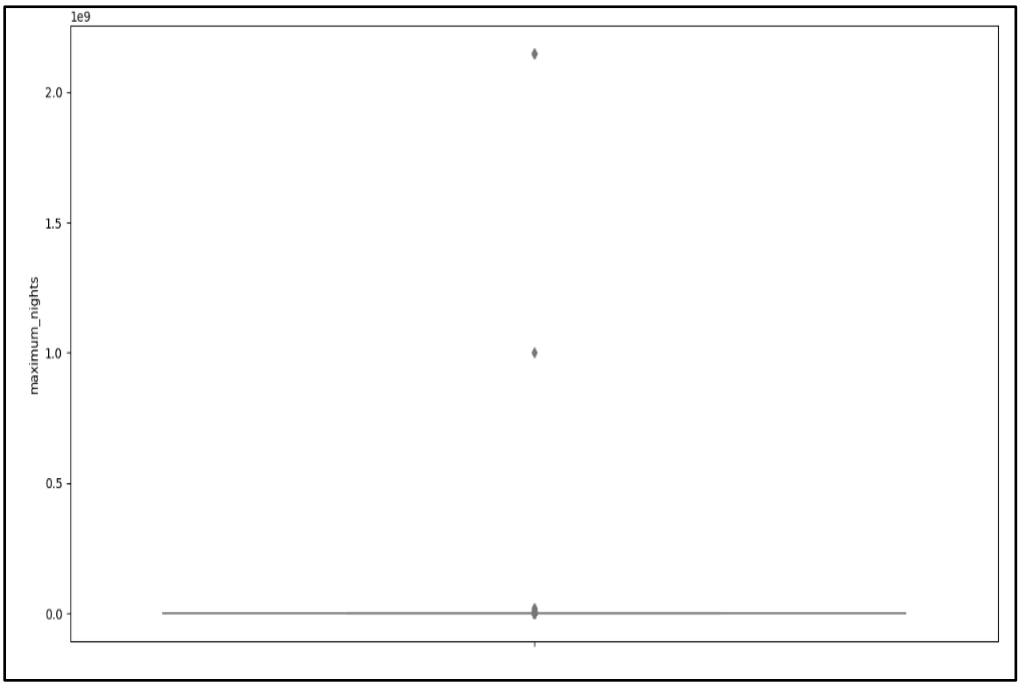


After Outlier Treatment:

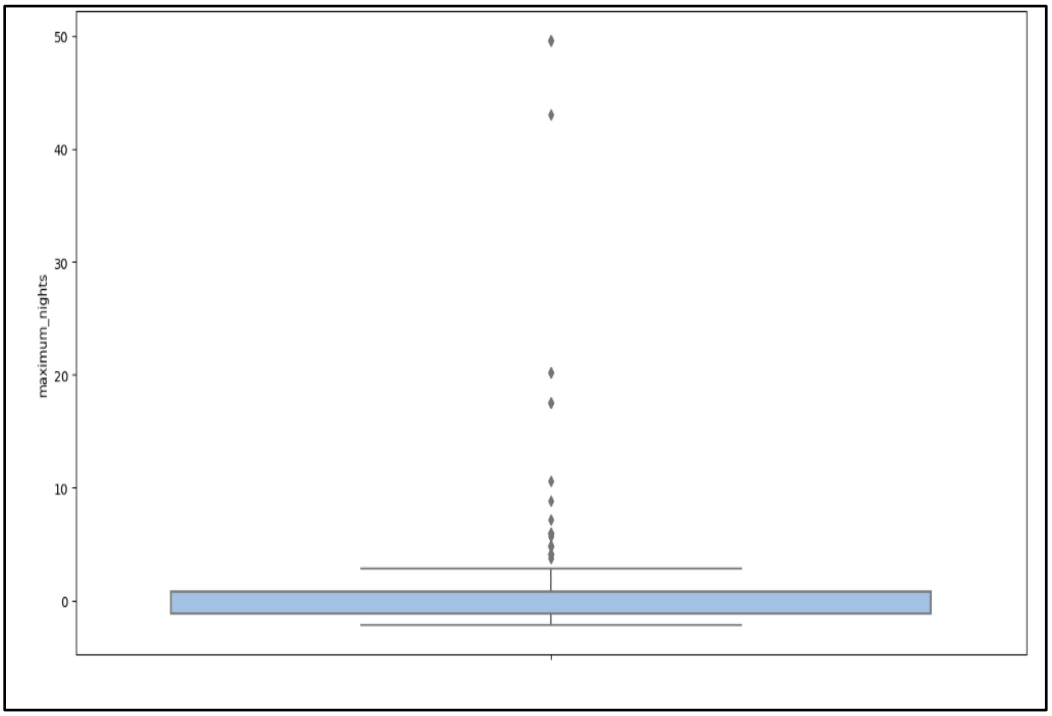


Maximum_nights:

Before Outlier Treatment:

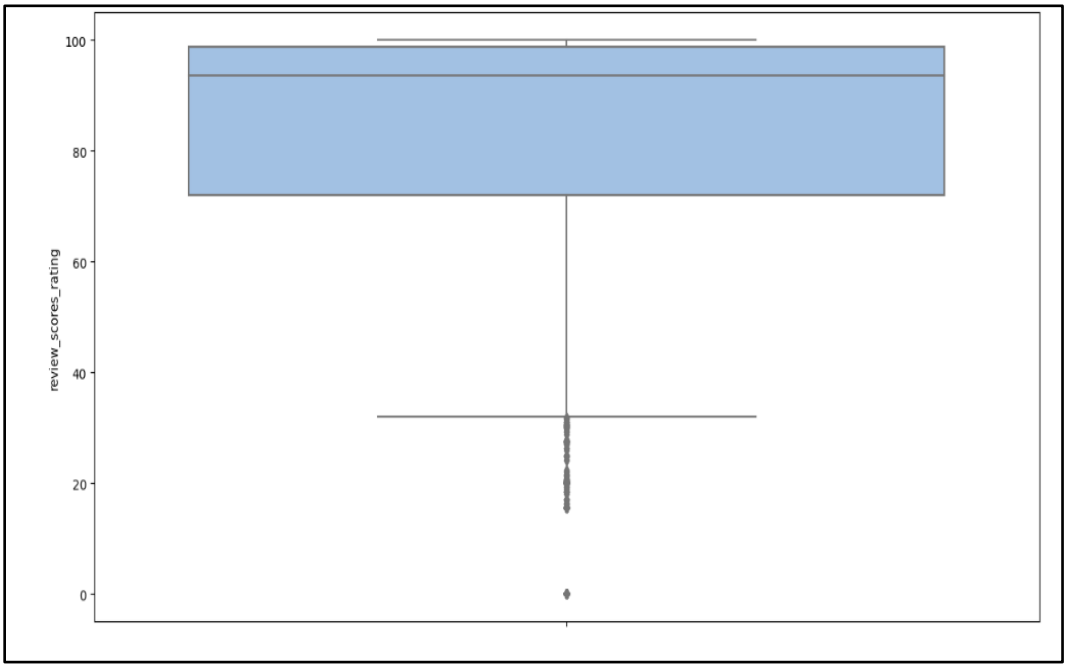


After Outlier Treatment:

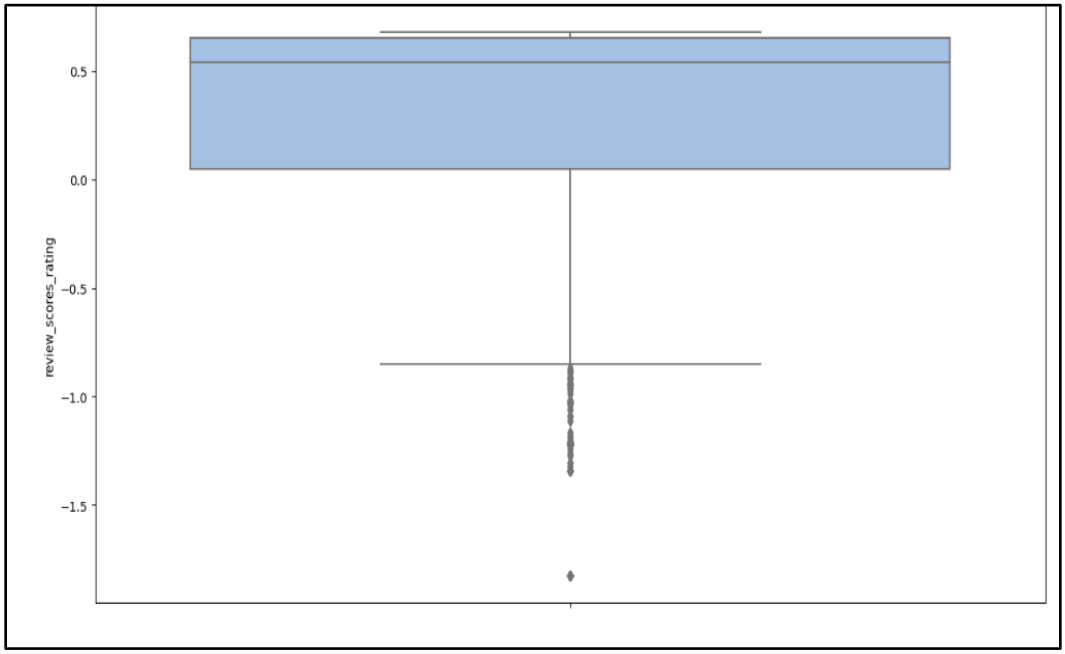


Review_scores_rating:

Before Outlier Treatment:



After Outlier Treatment:



ENCODING

Host_response_time:

Host response time variable is a ordinal categorical variable. It has a hierarchy between the subclasses. The host who responds within a few hours is more likely to get instant bookability than the host who responds a few days or more.

Host_is_superhost:

Host_is_superhost variable is a nominal categorical variable. There is no hierarchy between the subclasses. We can do dummy encoding.

Host_has_profile_pic:

Host_has_profile_pic variable is a nominal categorical variable. There is no hierarchy between the subclasses. We can do dummy encoding

Host_identity_verified:

Host_identity_verified variable is a nominal categorical variable. There is no hierarchy between the subclasses. We can do dummy encoding.

Instant_bookable:

Instant_bookable is a target variable. It is binary classification. We can do ordinal encoding.

Neighbourhood:

Neighbourhood variable is a nominal categorical variable. There is no hierarchy between the subclasses. But there is 509 subclasses. Hence we can do target encoding.

City:

City variable is a nominal categorical variable. There is no hierarchy between the subclasses. Since there is 10 subclasses we can do target encoding

Room type:

Room type variable is a ordinal categorical variable. It has a hierarchy between the subclasses.

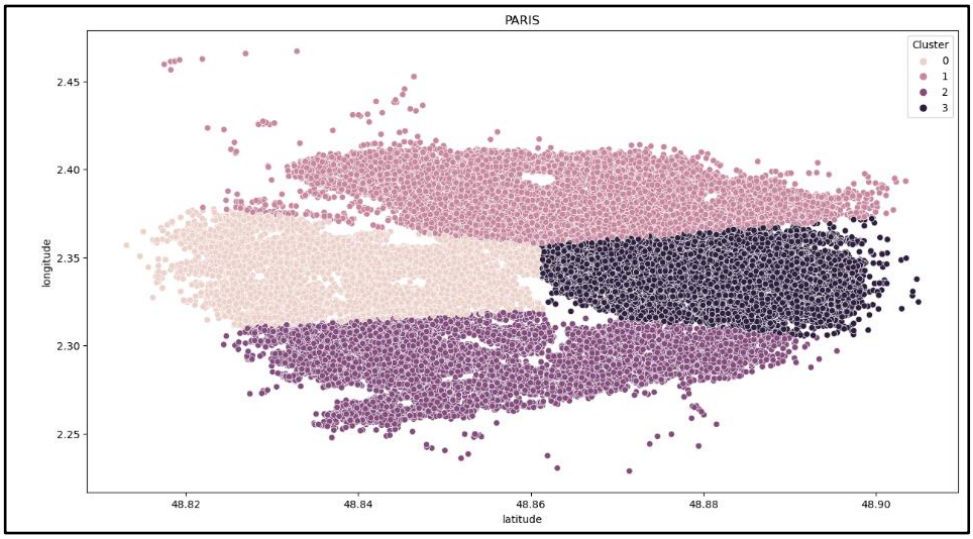
Regions:

Regions variable is a nominal categorical variable. There is no hierarchy between the subclasses. Since there is 4 subclasses we can do target encoding.

FEATURE ENGINEERING

We can infer few information on adding a new column named '**Region'** based on the latitude and longitude. On moving forward we decide to keep city variable and Region.

The Region variable is created by clustering the latitudes and longitudes of each and every city and they are categorized if that is in the Northern, Southern, Eastern or Western part of the city.



From the above scatterplot we infer that, the latitudes and longitudes of the city Paris is plotted. And they are divided into 4 clusters. Based on the latitude and longitude it is classified as 'North', 'South','West' or 'East'. With the help of latitude and longitude a new column 'Region' is inferred by feature engineering.

Train and Test Split:

TRAIN AND TEST SPLIT has been applied on the Dataset in which 70% data is a training set and 30% data is a testing test for the further analysis.

Scaling:

Scaling has been done to all numerical features by using Standard Scalar function.
Scaling done for train test first and with that mean and standard deviation , scaling is then done to test set for better model evaluation.

MODELLING:

- We have built a basic Logistic Regression model with all the features which we selected after VIF test.
- Since it is a Classification Problem, we performed model development with default Threshold value once and by calculating Optimum Threshold by using Youden's Index method once.

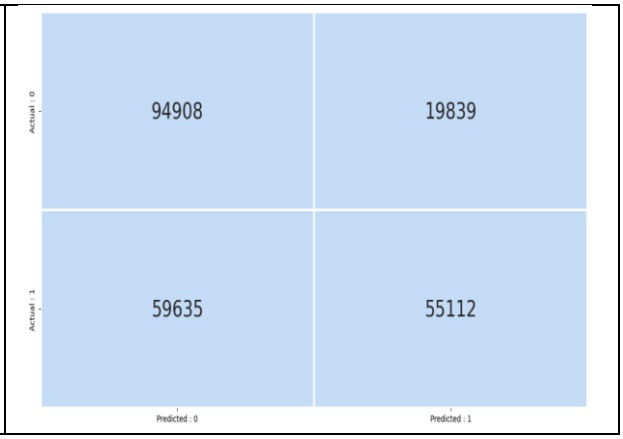
MODEL – 1: Base Model

Train report :

	precision	recall	f1-score	support
0	0.61	0.83	0.70	114747
1	0.74	0.48	0.58	114747
accuracy			0.65	229494
macro avg	0.67	0.65	0.64	229494
weighted avg	0.67	0.65	0.64	229494

Testreport :

	precision	recall	f1-score	support
0	0.61	0.83	0.70	49177
1	0.73	0.48	0.58	49177
accuracy			0.65	98354
macro avg	0.67	0.65	0.64	98354
weighted avg	0.67	0.65	0.64	98354



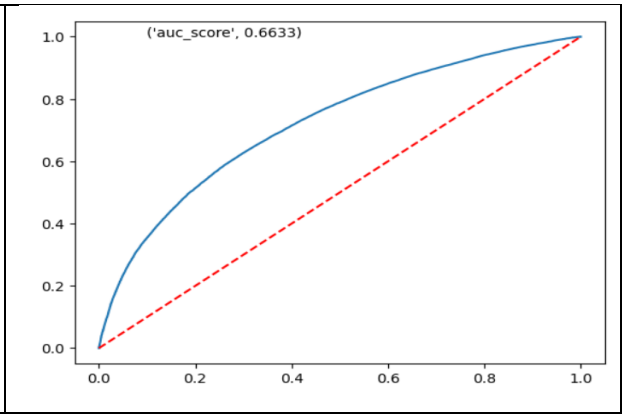
MODEL 2: Logistic Regression sklearn

Train Report :

	precision	recall	f1-score	support
0	0.66	0.69	0.67	114747
1	0.68	0.64	0.66	114747
accuracy			0.67	229494
macro avg	0.67	0.67	0.67	229494
weighted avg	0.67	0.67	0.67	229494

Test Report :

	precision	recall	f1-score	support
0	0.65	0.69	0.67	49177
1	0.67	0.63	0.65	49177
accuracy			0.66	98354
macro avg	0.66	0.66	0.66	98354
weighted avg	0.66	0.66	0.66	98354



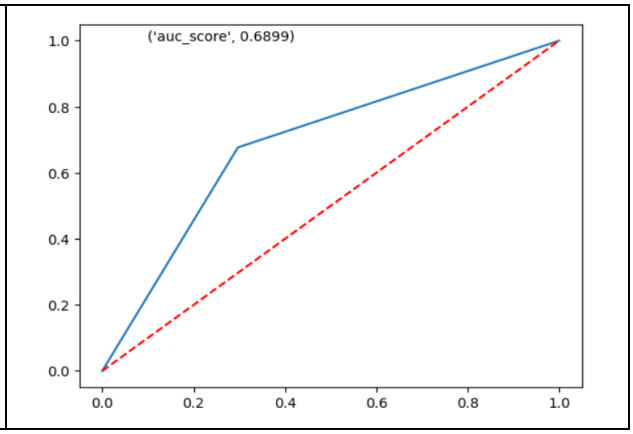
MODEL 3: Decision Tree without Tuning

Train Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	114747
1	1.00	1.00	1.00	114747
accuracy			1.00	229494
macro avg	1.00	1.00	1.00	229494
weighted avg	1.00	1.00	1.00	229494

Test Report :

	precision	recall	f1-score	support
0	0.69	0.70	0.69	49177
1	0.70	0.68	0.69	49177
accuracy			0.69	98354
macro avg	0.69	0.69	0.69	98354
weighted avg	0.69	0.69	0.69	98354



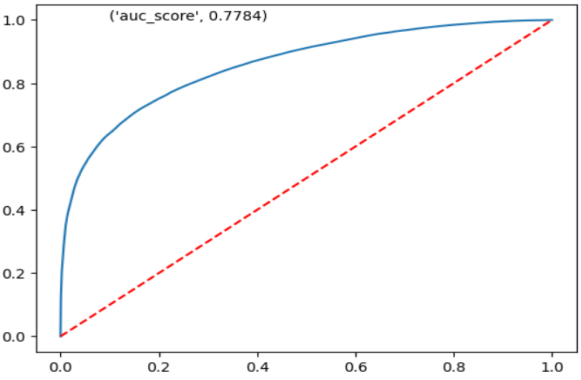
MODEL 4 Random Forest without Tuning:

Train Report :

	precision	recall	f1-score	support
0	1.00	1.00	1.00	114747
1	1.00	1.00	1.00	114747
accuracy			1.00	229494
macro avg	1.00	1.00	1.00	229494
weighted avg	1.00	1.00	1.00	229494

Test Report :

	precision	recall	f1-score	support
0	0.75	0.84	0.79	49177
1	0.82	0.72	0.76	49177
accuracy			0.78	98354
macro avg	0.78	0.78	0.78	98354
weighted avg	0.78	0.78	0.78	98354



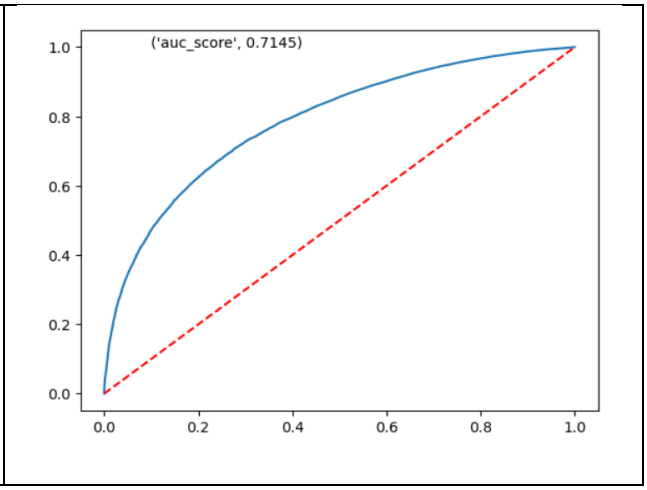
MODEL 5: Ada boost without Tuning

Train Report :

	precision	recall	f1-score	support
0	0.70	0.74	0.72	114747
1	0.72	0.69	0.71	114747
accuracy			0.71	229494
macro avg	0.71	0.71	0.71	229494
weighted avg	0.71	0.71	0.71	229494

Test Report :

	precision	recall	f1-score	support
0	0.71	0.74	0.72	49177
1	0.72	0.69	0.71	49177
accuracy			0.71	98354
macro avg	0.71	0.71	0.71	98354
weighted avg	0.71	0.71	0.71	98354



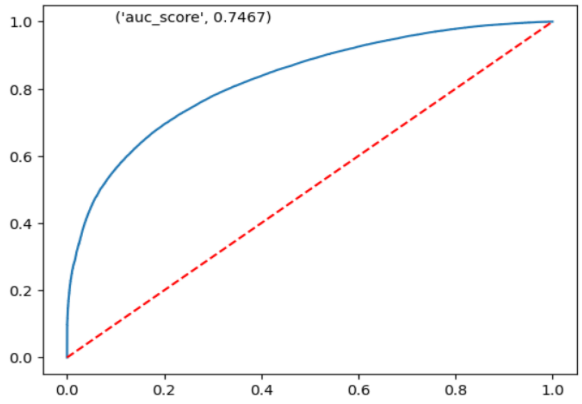
MODEL 6 Gradient Boosting without tuning:

Train Report :

	precision	recall	f1-score	support
0	0.73	0.78	0.76	114747
1	0.77	0.71	0.74	114747
accuracy			0.75	229494
macro avg	0.75	0.75	0.75	229494
weighted avg	0.75	0.75	0.75	229494

Test Report :

	precision	recall	f1-score	support
0	0.73	0.78	0.76	49177
1	0.77	0.71	0.74	49177
accuracy			0.75	98354
macro avg	0.75	0.75	0.75	98354
weighted avg	0.75	0.75	0.75	98354



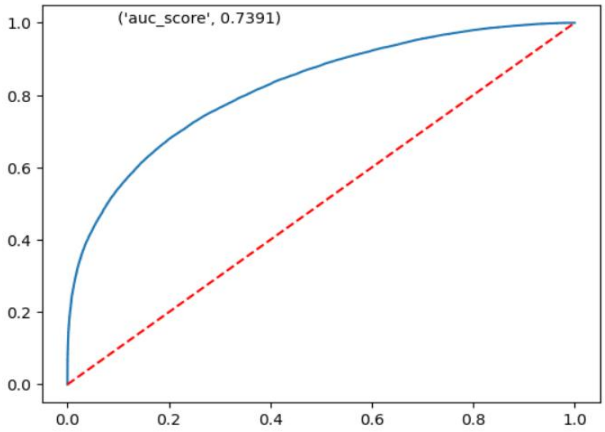
MODEL 7: Neural Network MLP Classifier

Train Report :

	precision	recall	f1-score	support
0	0.73	0.81	0.77	114747
1	0.79	0.70	0.74	114747
accuracy			0.75	229494
macro avg	0.76	0.75	0.75	229494
weighted avg	0.76	0.75	0.75	229494

Test Report :

	precision	recall	f1-score	support
0	0.71	0.80	0.75	49177
1	0.77	0.68	0.72	49177
accuracy			0.74	98354
macro avg	0.74	0.74	0.74	98354
weighted avg	0.74	0.74	0.74	98354



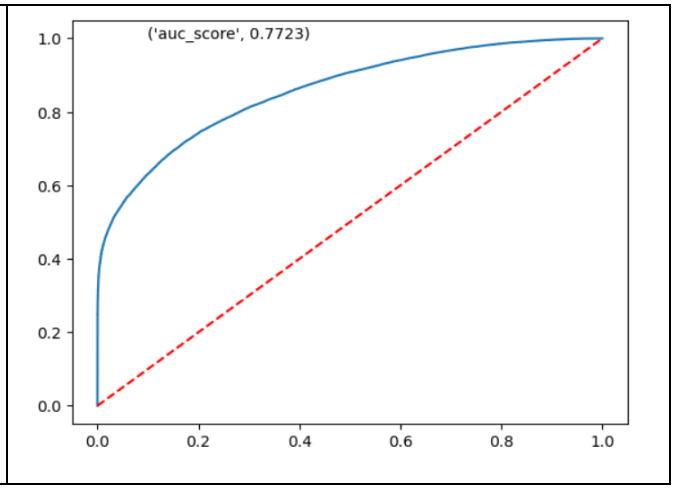
MODEL 8: Extreme Gradient Boosting Without Tuning:

Train Report :

	precision	recall	f1-score	support
0	0.76	0.85	0.80	114747
1	0.83	0.73	0.78	114747
accuracy			0.79	229494
macro avg	0.79	0.79	0.79	229494
weighted avg	0.79	0.79	0.79	229494

Test Report :

	precision	recall	f1-score	support
0	0.74	0.83	0.78	49177
1	0.81	0.71	0.76	49177
accuracy			0.77	98354
macro avg	0.78	0.77	0.77	98354
weighted avg	0.78	0.77	0.77	98354



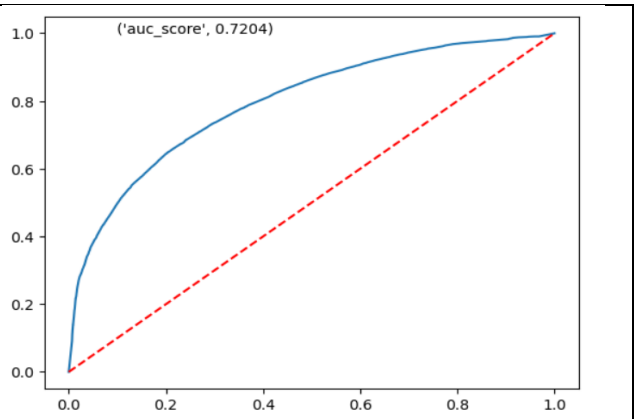
MODEL 9 Decision Tree Pruned:

Train Report :

	precision	recall	f1-score	support
0	0.73	0.79	0.76	114747
1	0.77	0.71	0.74	114747
accuracy			0.75	229494
macro avg	0.75	0.75	0.75	229494
weighted avg	0.75	0.75	0.75	229494

Test Report :

	precision	recall	f1-score	support
0	0.70	0.76	0.73	49177
1	0.74	0.68	0.71	49177
accuracy			0.72	98354
macro avg	0.72	0.72	0.72	98354
weighted avg	0.72	0.72	0.72	98354



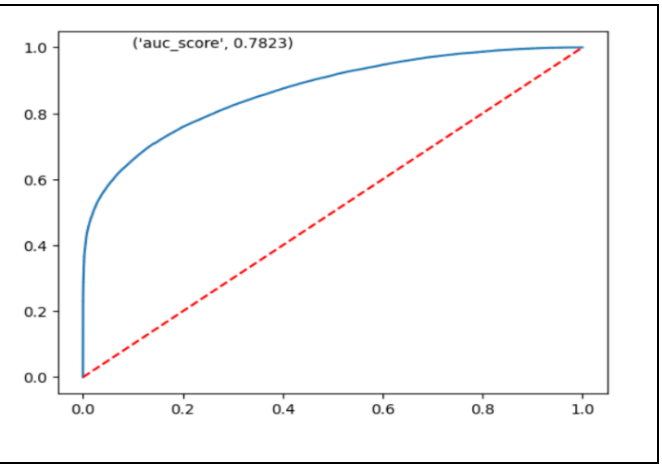
MODEL 10 Gradient Boosting Tuned

Train Report :

	precision	recall	f1-score	support
0	0.84	0.92	0.88	114747
1	0.91	0.82	0.86	114747
accuracy			0.87	229494
macro avg	0.87	0.87	0.87	229494
weighted avg	0.87	0.87	0.87	229494

Test Report :

	precision	recall	f1-score	support
0	0.75	0.84	0.79	49177
1	0.82	0.73	0.77	49177
accuracy			0.78	98354
macro avg	0.79	0.78	0.78	98354
weighted avg	0.79	0.78	0.78	98354



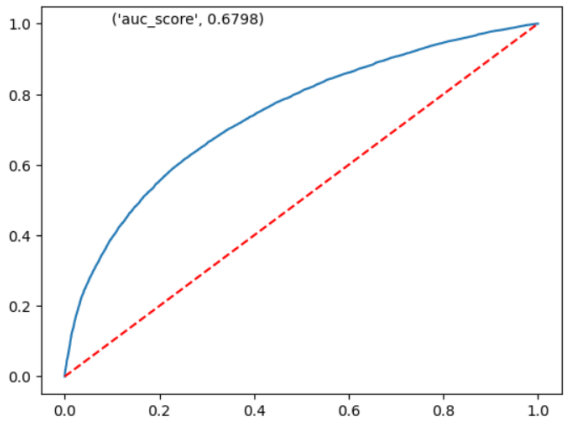
MODEL 11: AdaBoost Pruned:

Train Report :

	precision	recall	f1-score	support
0	0.67	0.71	0.69	114747
1	0.69	0.65	0.67	114747
accuracy			0.68	229494
macro avg	0.68	0.68	0.68	229494
weighted avg	0.68	0.68	0.68	229494

Test Report :

	precision	recall	f1-score	support
0	0.67	0.71	0.69	49177
1	0.69	0.65	0.67	49177
accuracy			0.68	98354
macro avg	0.68	0.68	0.68	98354
weighted avg	0.68	0.68	0.68	98354



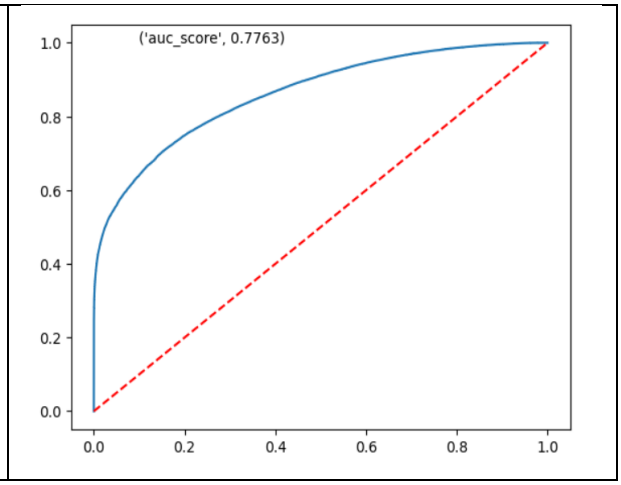
MODEL 12: Cat boost Pruned

Train Report :

	precision	recall	f1-score	support
0	0.77	0.86	0.81	114747
1	0.84	0.75	0.79	114747
accuracy			0.80	229494
macro avg	0.81	0.80	0.80	229494
weighted avg	0.81	0.80	0.80	229494

Test Report :

	precision	recall	f1-score	support
0	0.75	0.84	0.79	49177
1	0.81	0.72	0.76	49177
accuracy			0.78	98354
macro avg	0.78	0.78	0.78	98354
weighted avg	0.78	0.78	0.78	98354



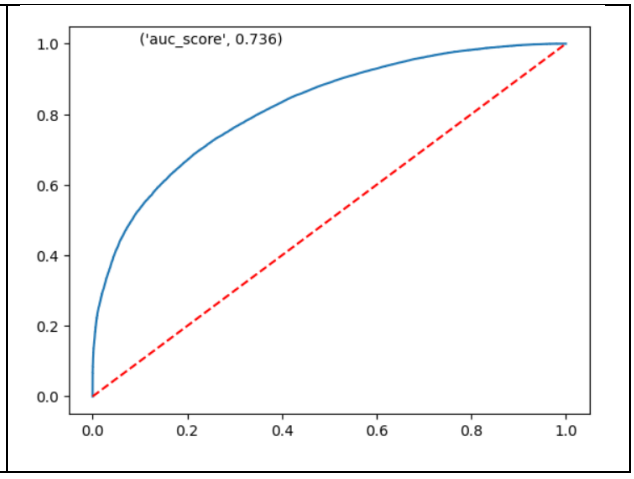
MODEL 13 : Random Forest Pruned

Train Report :

	precision	recall	f1-score	support
0	0.73	0.78	0.76	114747
1	0.77	0.71	0.74	114747
accuracy			0.75	229494
macro avg	0.75	0.75	0.75	229494
weighted avg	0.75	0.75	0.75	229494

Test Report :

	precision	recall	f1-score	support
0	0.72	0.77	0.74	49177
1	0.75	0.70	0.73	49177
accuracy			0.74	98354
macro avg	0.74	0.74	0.74	98354
weighted avg	0.74	0.74	0.74	98354



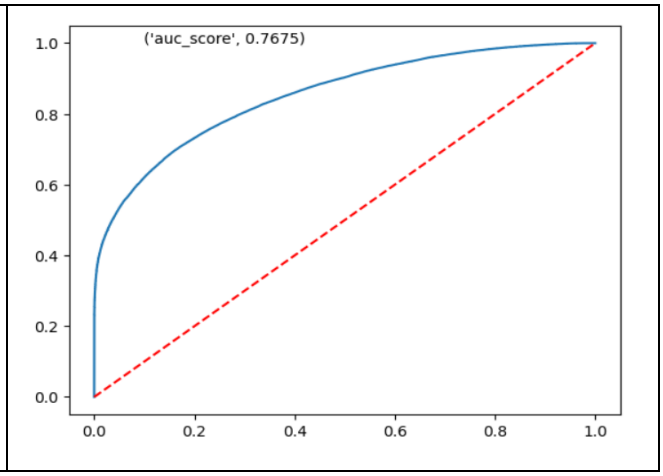
MODEL 14: XGB Tuned:

Train Report :

	precision	recall	f1-score	support
0	0.75	0.83	0.79	114747
1	0.81	0.72	0.76	114747
accuracy			0.77	229494
macro avg	0.78	0.77	0.77	229494
weighted avg	0.78	0.77	0.77	229494

Test Report :

	precision	recall	f1-score	support
0	0.74	0.82	0.78	49177
1	0.80	0.71	0.75	49177
accuracy			0.77	98354
macro avg	0.77	0.77	0.77	98354
weighted avg	0.77	0.77	0.77	98354



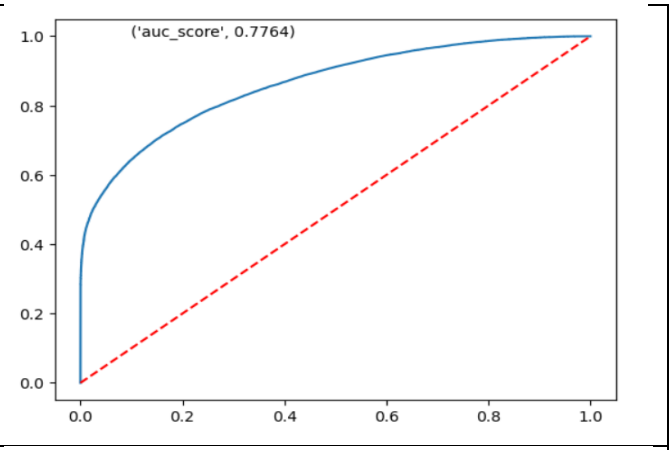
MODEL 15 Cat Boost Without Tuning (Best model)

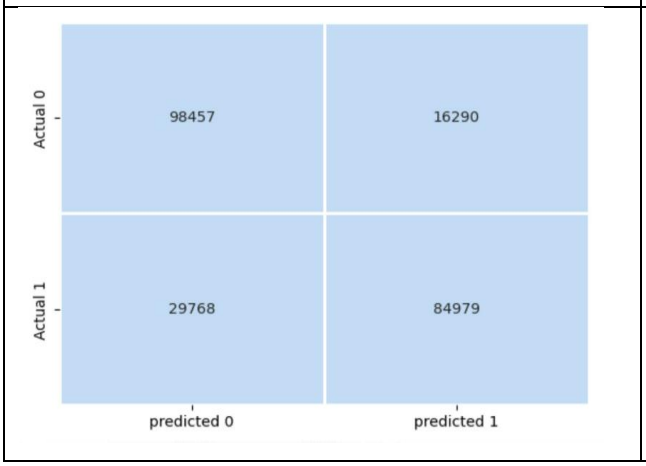
Train Report :

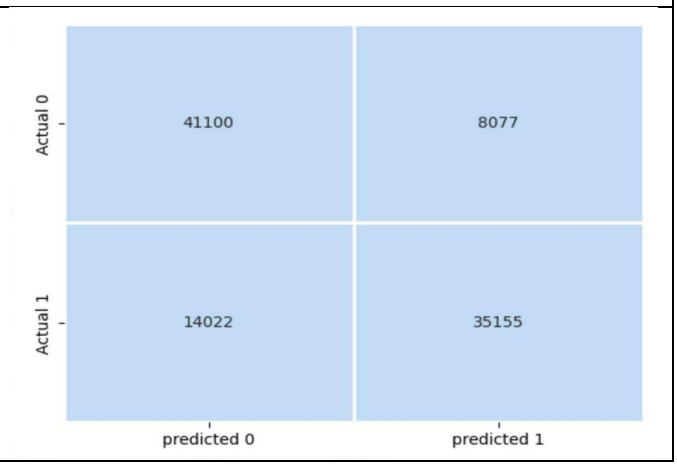
	precision	recall	f1-score	support
0	0.77	0.86	0.81	114747
1	0.84	0.74	0.79	114747
accuracy			0.80	229494
macro avg	0.80	0.80	0.80	229494
weighted avg	0.80	0.80	0.80	229494

Test Report :

	precision	recall	f1-score	support
0	0.75	0.84	0.79	49177
1	0.81	0.72	0.76	49177
accuracy			0.78	98354
macro avg	0.78	0.78	0.78	98354
weighted avg	0.78	0.78	0.78	98354







OVERALL MODEL PERFORMANCE:

	Model_Name	Train_Accuracy	Train_F1score	Test_Accuracy	Test_F1score	Precision_Score	Recall_Score	AUC_Score	Remarks
0	Logistic Regression sklearn	67.000000	0.660000	66.000000	0.650000	0.670000	0.630000	0.721500	Good Fit
1	DecisionTree Model without tuning	100.000000	1.000000	69.000000	0.690000	0.700000	0.680000	0.690000	Over Fit
2	Random Forest without tuning	100.000000	1.000000	78.000000	0.760000	0.820000	0.720000	0.859100	Over Fit
3	AdaBoost without tuning	71.000000	0.710000	71.000000	0.710000	0.720000	0.690000	0.787000	Good Fit
4	GradientBoosting without tuning	75.000000	0.740000	75.000000	0.740000	0.770000	0.710000	0.826500	Good Fit
5	Neural Network	75.000000	0.740000	74.000000	0.720000	0.770000	0.680000	0.820100	Good Fit
6	Xtreme Gradient Boosting without tuning	79.000000	0.780000	77.000000	0.760000	0.810000	0.710000	0.857800	Good Fit
7	Catboost	80.000000	0.790000	78.000000	0.760000	0.810000	0.720000	0.862500	Good Fit
8	Decision Tree tuned	75.000000	0.740000	72.000000	0.710000	0.740000	0.680000	0.796100	Good Fit
9	GradientBoosting Classifier tuned	87.000000	0.860000	78.000000	0.770000	0.820000	0.730000	0.867700	Over Fit
10	AdaBoost Classifier tuned	68.000000	0.670000	68.000000	0.670000	0.690000	0.650000	0.741700	Good Fit
11	Cat Boost Tuned	80.000000	0.790000	78.000000	0.760000	0.810000	0.720000	0.861600	Good Fit
12	Xtreme Gradient Boosting tuned	77.000000	0.760000	77.000000	0.750000	0.800000	0.710000	0.852600	Good Fit
13	Random forest Model tuned	75.000000	0.740000	74.000000	0.730000	0.750000	0.700000	0.819900	Good Fit

After assessing various models, it was observed that some models exhibited a significant drop in performance when applied to unseen data, indicating overfitting. However, there were models that consistently performed well on both training and unseen data. Notably, the Catboost model outperformed other models in terms of performance(78%). Hence, based on its superior performance and generalization ability, we can confidently consider the Catboost model as our final choice.

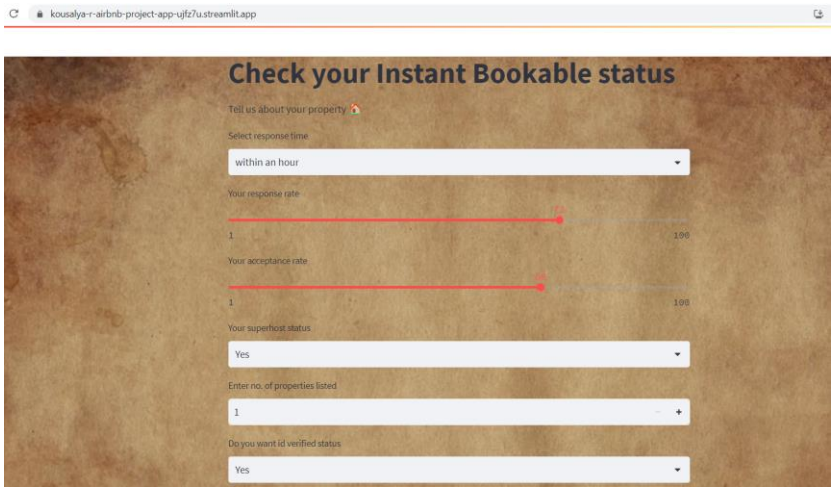
DEPLOYMENT:

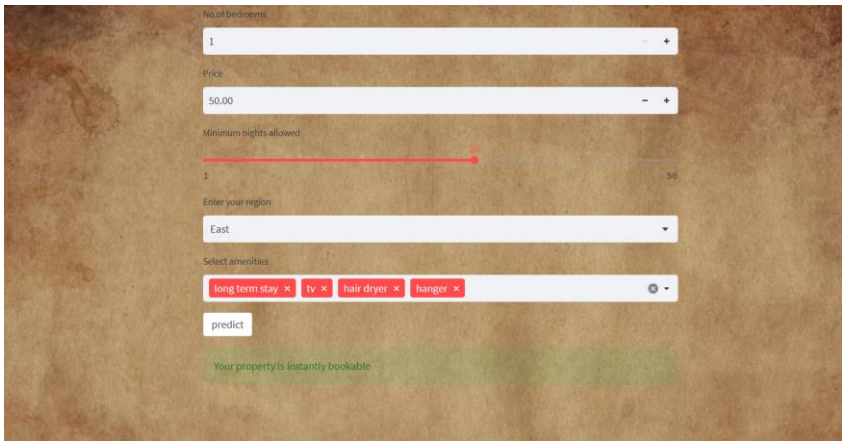
The predictive model is integrated with the Streamlit application, it accurately predicts the instant bookability of the Airbnb properties. By considering features such as amenities, host response time, location, minimum nights of stay, bedrooms available, superhost status, verified property status, and host acceptance rate etc.

This model provides valuable insights to guests and property owners to make booking decisions or business decisions. With this innovative solution, the Airbnb booking process becomes seamless, efficient, and tailored to individual preferences, ultimately enhancing the user experience and satisfaction. This also helps the property owners to predict if their property is instantly bookable or not. Accordingly, they can take the business decisions to increase their features.

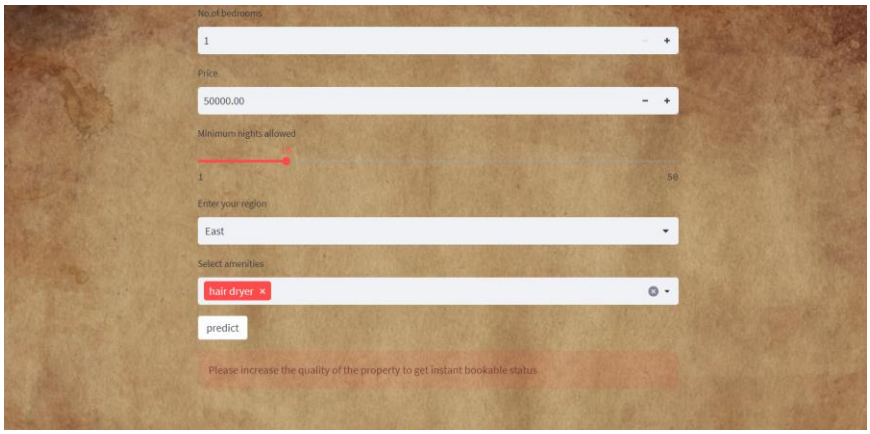
Deployment link: [airbnb_instant_bookability_predictor]

UI DESIGN:





If the property is instantly bookable. It is popped up to the user as shown above.



If the property is not instantly bookable. It is popped up to the user as shown above.
These property owners should increase the quality of their property to get the instant bookability status.

COMPARISON TO BENCHMARK:

To evaluate the effectiveness of our model, we establish a benchmark by comparing its performance against existing industry standards or alternative approaches.

We utilize various evaluation metrics such as accuracy, precision, recall, and F1-score to measure the model's predictive capability.

Additionally, we employ techniques like cross-validation and train-test splits to ensure robustness in our comparisons.

By benchmarking our model, we gain insights into its strengths and weaknesses relative to other methodologies.

IMPLICATIONS:

Our developed model has significant implications for both property owners and users in the Airbnb ecosystem. By accurately predicting instant bookability, property owners can gauge the attractiveness of their listings, optimize pricing strategies, and make necessary improvements to enhance bookability.

Users, on the other hand, benefit from a more streamlined booking process, enabling them to quickly identify high-quality properties that meet their specific preferences and requirements.

This model fosters trust and transparency, benefiting both sides of the Airbnb marketplace.

LIMITATIONS:

Despite the effectiveness of our model, it is important to acknowledge its limitations.

First, the predictions rely heavily on the dataset used for training, and any biases or inconsistencies present in the data may impact the model's performance.

Additionally, the model's accuracy is subject to external factors such as changes in user preferences, market dynamics, and evolving Airbnb policies.

Furthermore, the model may not account for contextual information that can influence bookability, such as local events, seasonality, or unique property characteristics. It is crucial to consider these limitations when interpreting and utilizing the model's predictions.

CLOSING REFLECTIONS:

What have you learned from the process? What would you do differently next time?

- In the process of working and delivering this capstone project, apart from deep knowledge about the subject, we have also learned how to work well in teams-brainstorming ideas together and further working on constructive criticism/feedback from our mentor.
- During this process, we have learnt to execute various models such as logistic regression, decision tree and ensemble techniques such as bagging, boosting and catboost, making us equipped with the different approaches followed by different models to reach the most precise metric value for our model.

Next time, we would like to choose a better dataset, which can provide us with more information on certain variables such as location of the banks.

REFERENCES:

https://www.rentalscaleup.com/airbnb-search-ranking-algorithm/

https://towardsdatascience.com/why-not-mse-as-a-loss-function-for-logistic-regression-589816b5e03c