PROJECT REPORT

ON

Airbnb Price Prediction

Mini Project II

Department of Computer Science & Application

Institute of Engineering & Technology

SUBMITTED TO: -

Mr. Amir Khan Sir
(Technical Trainer)

SUBMITTED BY: -

Yash Kumar Gupta (201500820)
Vijay Kaushal (201500783)

DECLARATION

I would like to express my special thanks of gratitude to my project guide **Mr. Amir Khan Sir** who gave me the golden opportunity to do this wonderful project on the topic **AirBnb Price Prediction**, which also helped me in doing a lot of research and I came to know about so many new things I am thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Team Members:-

- Yash Kumar Gupta (201500820)
- Vijay Kaushal (201500783)

CERTIFICATE

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

Project Supervisor

Mr. Amir Khan Sir

Senior Trainer

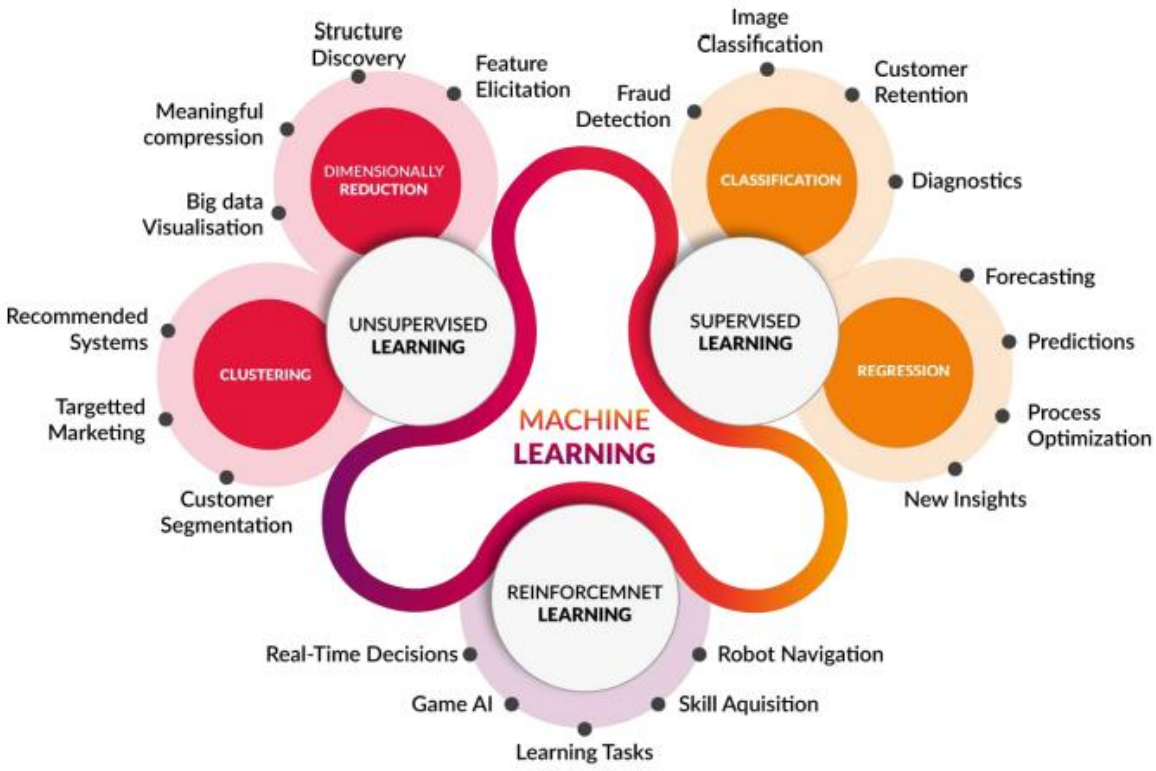
Date: 28rd Apr 2023

Table of Content

INTRODUCTION

In this project, we worked on datasets to provide a predictive price of an airbnb service provider. **For those who do not know it yet** for some **reason,** Airbnb is **an Internet** marketplace for short-term **rentals of homes** and **apartments. For example,** you **can** rent out **(advertise)** your home for **his** week or rent out **an** empty **bedroom while you are away.** One **of the challenges** Airbnb hosts face is determining the **best rental price per night.** In many **locations,** renters (hosts) **will see curated** listings **that** can **be filtered** by price, number of bedrooms, room type, and **other criteria.** Airbnb is a **marketplace, so** the amount a host can charge is ultimately tied to **the** market **price.**

We used different algorithms to get a accuracy in our price prediction. Pricing a rental property on Airbnb is a daunting task for owners as the location has customers. Customers, on the other hand, should evaluate the offered price with a minimum of knowledge of the property's optimal value. This white paper aims to develop reliable price prediction models using machine learning, deep learning and natural language processing techniques. Rental characteristics, owner characteristics, and customer ratings include linear regression to tree-based models, support vector regression (SVR), k-means clustering (KMC), and neural networks (NN) includes predictors and different methods. Used to build predictive models.



Machine Learning Factor

METHODOLOGY-

The problem statement is to predict the price of Airbnb house based on multiple factors like locality, reviews per month, availability, room type etc .

We implement different machine learning algorithms like ridge, lasso, decision tree etc. to predict the price.

We created a flask web app which takes input as neighbourhood group, room type availability and show predicted result on screen.

We performed exploratory data analysis to analyse (univariate and bivariate) the data set and summarize the data

SYSTEM REQUIREMENTS

Software Requirement-

To build application –

- 64-bit Windows 8/10/11
- Anaconda Jupyter

To Run application –

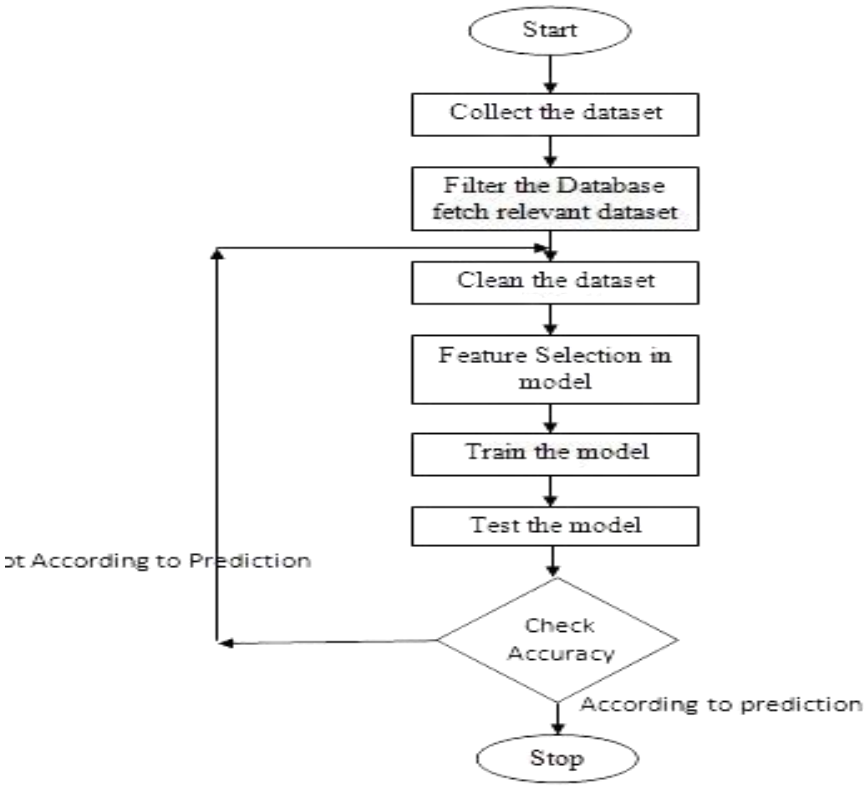
- Laptop or a system with min i3 processor and window 7 or higher versions

Hardware Requirement –

- x86_64 CPU architecture; 10th generation Intel Core or newer
- 4 GB RAM or more
- 8 GB of available disk space minimum

IMPLEMENTATION

Data Flow Diagram



Some of the more important features this project will look into are the following:

- accommodates: the number of guests the rental can accommodate
- bedrooms: number of bedrooms included in the rental
- bathrooms: number of bathrooms included in the rental
- beds: number of beds included in the rental
- price: nightly price for the rental
- minimum_nights: minimum number of nights a guest can stay for the rental

- maximum_nights: maximum number of nights a guest can stay for the rental
- number_of_reviews: number of reviews that previous guests have left.

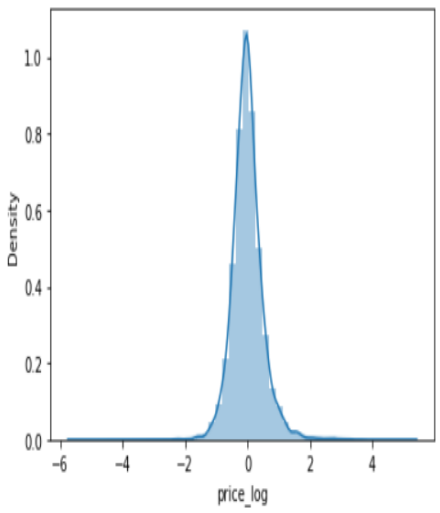
Code Snaps –

In [142]:
```
from sklearn.tree import DecisionTreeRegressor
DTree=DecisionTreeRegressor(min_samples_leaf=.0001)
DTree.fit(X_train,Y_train)
score_DTree = DTree.score(X_test, Y_test)
Y_DTree = DTree.predict(X_test)
print('R-squared score (training): {:.3f}'.format(DTree.score(X_train, Y_train)))
print('R-squared score (test): {:.3f}'.format(DTree.score(X_test, Y_test)))
sns.distplot(Y_test-Y_DTree)
```

```
R-squared score (training): 0.716
R-squared score (test): 0.500
```

```
C:\Users\Yash\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[142]: <AxesSubplot:xlabel='price_log', ylabel='Density'>



Importing Library

```
In [1]: import numpy as np
        import pandas as pd

        import matplotlib.pyplot as plt
        import matplotlib.image as mpimg
        %matplotlib inline
        import seaborn as sns
        from wordcloud import WordCloud

        from scipy.stats import norm
        from scipy import stats

        from sklearn.model_selection import cross_val_score
        from sklearn.metrics import r2_score
        from sklearn.model_selection import GridSearchCV,RandomizedSearchCV

        from sklearn.linear_model import LinearRegression
        from sklearn.preprocessing import PolynomialFeatures
        from sklearn.linear_model import Ridge
        from sklearn.linear_model import Lasso
        from sklearn.tree import DecisionTreeRegressor
```

Loading Dataset

```
In [2]: data=pd.read_csv('AB_NYC_2019.csv',encoding= 'unicode_escape')
```

Show Dataset

```
In [3]: data.head()
        #data.tail()
```

Out[3]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80

Size of Dataset

```
In [4]: data.shape
```

```
Out[4]: (48895, 16)
```

Examine the dataset

```
In [5]: data.dtypes
```

```
Out[5]: id                                int64
        name                             object
        host_id                           int64
        host_name                        object
        neighbourhood_group              object
        neighbourhood                    object
        latitude                        float64
        longitude                       float64
        room_type                        object
        price                             int64
        minimum_nights                    int64
        number_of_reviews                 int64
        last_review                      object
        reviews_per_month               float64
        calculated_host_listings_count    int64
        availability_365                  int64
        dtype: object
```

```
In [6]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              48895 non-null  int64
 1   name                            48879 non-null  object
 2   host_id                         48895 non-null  int64
 3   host_name                       48874 non-null  object
 4   neighbourhood_group             48895 non-null  object
 5   neighbourhood                   48895 non-null  object
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object
 9   price                           48895 non-null  int64
 10  minimum_nights                  48895 non-null  int64
 11  number_of_reviews               48895 non-null  int64
 12  last_review                     38843 non-null  object
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64
 15  availability_365                48895 non-null  int64
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

```
In [7]: data.describe()
```

Out[7]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	revie
count	4.889500e+04	4.889500e+04	48895.000000	48895.000000	48895.000000	48895.000000	48895.000000	
mean	1.901714e+07	6.762001e+07	40.728949	-73.952170	152.720687	7.029962	23.274466	
std	1.098311e+07	7.861097e+07	0.054530	0.046157	240.154170	20.510550	44.550582	
min	2.539000e+03	2.438000e+03	40.499790	-74.244420	0.000000	1.000000	0.000000	
25%	9.471945e+06	7.822033e+06	40.690100	-73.983070	69.000000	1.000000	1.000000	
50%	1.967728e+07	3.079382e+07	40.723070	-73.955680	106.000000	3.000000	5.000000	
75%	2.915218e+07	1.074344e+08	40.763115	-73.936275	175.000000	5.000000	24.000000	
max	3.648724e+07	2.743213e+08	40.913060	-73.712990	10000.000000	1250.000000	629.000000	

Categorical Features

```
In [8]: categorical_columns=[column for column in data.columns if data[column].dtypes=='O']
        categorical_columns
```

```
Out[8]: ['name',
         'host_name',
         'neighbourhood_group',
         'neighbourhood',
         'room_type',
         'last_review']
```

```
In [9]: data[categorical_columns].head()
```

Out[9]:

	name	host_name	neighbourhood_group	neighbourhood	room_type	last_review
0	Clean & quiet apt home by the park	John	Brooklyn	Kensington	Private room	10/19/2018
1	Skylit Midtown Castle	Jennifer	Manhattan	Midtown	Entire home/apt	5/21/2019
2	THE VILLAGE OF HARLEM....NEW YORK !	Elisabeth	Manhattan	Harlem	Private room	NaN
3	Cozy Entire Floor of Brownstone	LisaRoxanne	Brooklyn	Clinton Hill	Entire home/apt	7/5/2019
4	Entire Apt: Spacious Studio/Loft by central park	Laura	Manhattan	East Harlem	Entire home/apt	11/19/2018

Numeric Feature

```
In [10]: numeric_columns=[column for column in data.columns if column not in categorical_columns]
```

```
In [11]: data[numeric_columns].head()
```

Out[11]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listing
0	2539	2787	40.64749	-73.97237	149	1	9	0.21	
1	2595	2845	40.75362	-73.98377	225	1	45	0.38	
2	3647	4632	40.80902	-73.94190	150	3	0	NaN	
3	3831	4869	40.68514	-73.95976	89	1	270	4.64	
4	5022	7192	40.79851	-73.94399	80	10	9	0.10	

Discrete Features

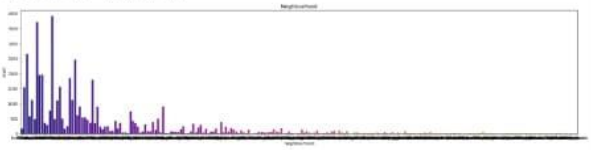
```
In [12]: discrete_features=[feature for feature in numeric_columns if len(data[feature].unique())<50]
```

Neighbourhood

```
In [32]: sns.countplot(data['neighbourhood'], palette="plasma")
         fig = plt.gcf()
         fig.set_size_inches(25,6)
         plt.title('Neighbourhood')
```

```
C:\Users\priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following
variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

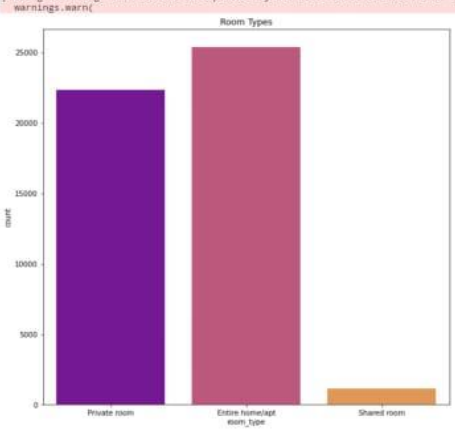
```
Out[32]: Text(0.5, 1.0, 'Neighbourhood')
```



Room Type

```
In [33]: sns.countplot(data['room_type'], palette="plasma")
         fig = plt.gcf()
         fig.set_size_inches(10,10)
         plt.title('Room Types')
         plt.show()
```

```
C:\Users\priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following
variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

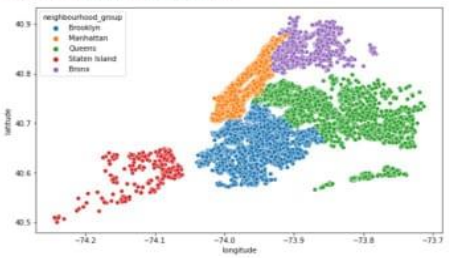


Map of Neighbourhood Group

```
In [34]: plt.figure(figsize=(10,6))
         sns.scatterplot(data.longitude,data.latitude,hue=data.neighbourhood_group)
         plt.ioff()
```

```
C:\Users\priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following
variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[34]: <contextlib.ExitStack at 0x2147b3e3eb0>
```

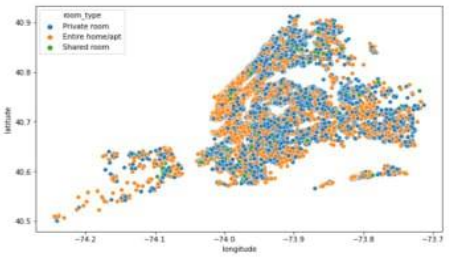


Map of Neighbourhood

```
In [35]: plt.figure(figsize=(10,6))
         sns.scatterplot(data.longitude,data.latitude,hue=data.room_type)
         plt.ioff()
```

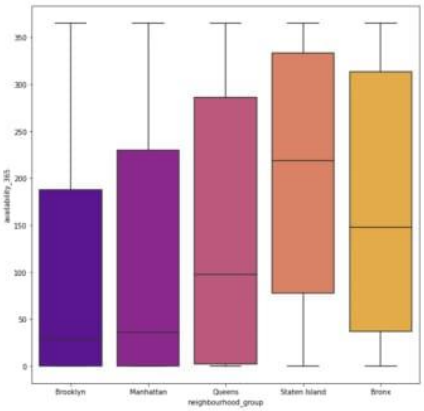
```
C:\Users\priya\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following
variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
Out[35]: <contextlib.ExitStack at 0x2147f2e4be0>
```

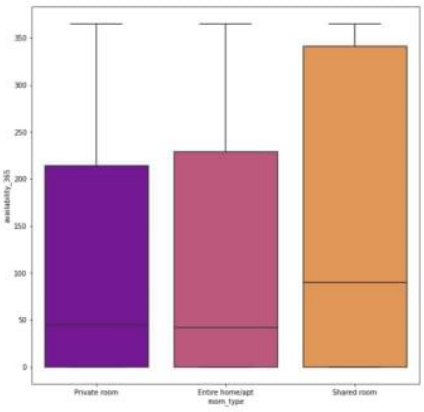


Relation between neighbourgroup and Availability of Room

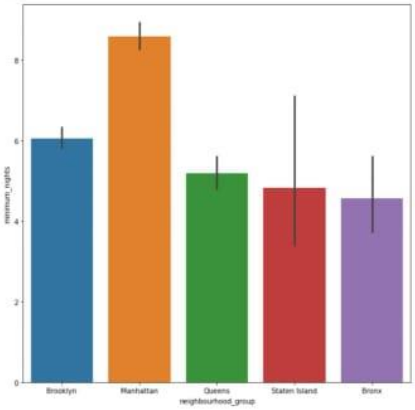
In [42]:
```
plt.figure(figsize=(10,10))
ax = sns.boxplot(data=data, x='neighbourhood_group',y='availability_365',palette='plasma')
```



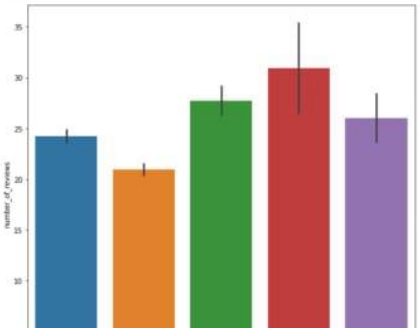
In [43]:
```
plt.figure(figsize=(10,10))
ax = sns.boxplot(data=data, x='room_type',y='availability_365',palette='plasma')
```



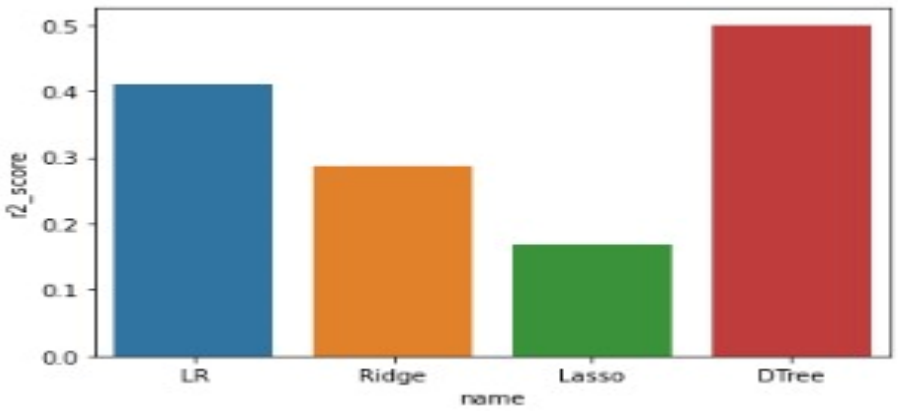
In [44]:
```
plt.figure(figsize=(10,10))
ax = sns.barplot(data=data, x='neighbourhood_group',y='minimum_nights')
plt.show()
```



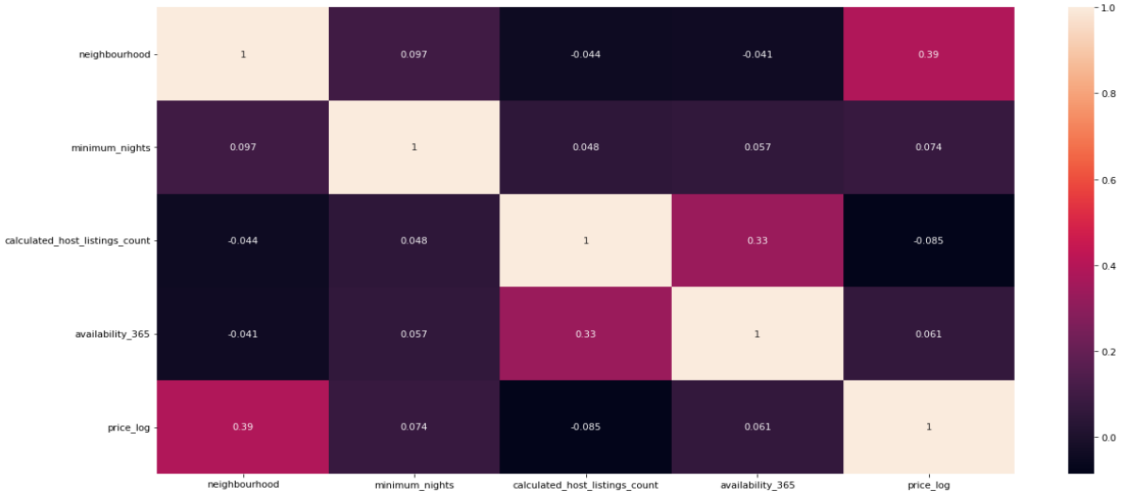
In [45]:
```
plt.figure(figsize=(10,10))
ax = sns.barplot(data=data, x='neighbourhood_group',y='number_of_reviews')
plt.show()
```



OUTPUT



Accuracy by different-different model



Confusion Matrix Calculation

CONCLUSION AND FUTURE WORK

Conclusion:

1. Through this research we are able to understand the 2019 data and the price distribution of the rental properties.
2. Avg. Price of Airbnb house in Manhattan is 31% which is maximum among

all the neighborhood group. Bronex is the cheapest as its contribution in pi chart is 14%. Most of the persons prefers Entire Room/ Apartment and very less people prefer Shared Room to stay.

3. Overall the Decision Tree gives the best result and it suits for the prediction of price of AirBnb (Air bed and breakfast).

Future work: -

For future works, we expect the price prediction model to be improved using a larger dataset with balanced customer reviews since the mean squared error in the Neural Network model is still decreasing at the end of the training phase. Additionally, public Airbnb datasets contain more positive reviews than negative reviews. A well-balanced dataset should be helpful to build a more accurate price prediction model. Other than customer reviews, historical prices might be another key factor to evaluate the price. Customers expect a lower price if the price is constantly decreasing.

Github link: -

https://github.com/Yash-Kumar-Gupta-0845/Mini-Project-2