



UNDERSTANDING THE GUEST PREFERENCES TO STAY AHEAD OF THE GAME

CAPSTONE PROJECT- BA723

Hosain Ahmed

ID: 301209637

Executive Summary

Airbnb was thriving thanks to the emergence of a global middle class, which did more for the growth of tourism globally than did global economic prosperity. (Glusac, 2020) Positive indicators, however, swiftly disappear because to the Covid-19 epidemic, which decreased sales by 30% and increased losses by \$3.9 billion US in 2020. About 25% of Airbnb's personnel was let go to get through the crisis. It also raised \$2 billion in a combination of equity and debt, and reduced activities that were not essential to its core business model. (Riley, 2022)

However, the situation has recovered as to the new work from home trend proved to be beneficial for Airbnb. Furthermore, the economy is started to rebuild but the recent global political unrest has hurt that effort badly and almost every economic indicator is predicting an upcoming recession. Eventually, if that prediction becomes a reality, then Airbnb can expect more hosts will join stream and guests will prioritize the basic needs. Therefore, Airbnb should have a model which can predict the features that the Airbnb guests value most while booking a property. With this model Airbnb will be able to understand the most valued needs of its guests and can prepare its offerings accordingly to boost its revenues even during recessions. For my analysis, I have considered the days a property was available for booking as my target variable and used machine learning models to find which features of the property is more affective in determining the availability of bookings. To build the desired model that can predict the guest preferred features, I have applied four machine learning models including Random Forest Regressor, Multiple Linear Regression, K Neighbor Classifier and Neural Network and picked the best model based on the Root-mean-square error (RMSE). At the end of my analysis, Multiple Linear Regression to be the best model. Even though Multiple Linear Regression has the lowest model accuracy score among all the models used in this project, but it has the lowest RMSE score which mean that this model can predicting the features preferred by the Airbnb guests with lesser error than all the other models. Moreover, this model also figured few more insights such as preferred location, room type and duration of stay of the guests and recommendations are provided to Airbnb accordingly. However, considering the sensitivity of the model and changing nature of customer preferences, Airbnb have to monitor and review the results of this model on quarter basis.

Table of Contents

Introduction.....	4-5
1. Background.....	4
2. Problem Statement.....	5
3. Objectives & Measurement.....	5
4. Metrics of Measurement.....	5
5. Assumptions and Limitations.....	5
Data Sources.....	5-6
6. Data Set Introduction.....	5
7. Exclusions.....	6
8. Data Dictionary.....	6
Exploratory Data Analysis.....	7-10
9. Data Exploration Techniques.....	7
10. Major Findings of Data Exploration.....	8
11. Data Cleaning.....	9
12. Summary Report.....	10
Data Preparation and Feature Engineering.....	11-13
13. Data Partition.....	11
Secondary Data Analysis and Data Visualization.....	11
14. Data Visualization.....	11
15. Correlations.....	12
Tools and Models used.....	13-14
Analysis and Findings.....	15-26
16. Random Forest Regressor.....	15
17. K Nearest Neighbor.....	19
18. Neural Network.....	20
19. Multiple Linear Regression.....	22
Overall Insights of the Findings.....	27

Recommendations.....	27
Conclusions.....	28
Validation and Governance.....	29-32
References.....	33-34
Appendix.....	35-37
Heatmap.....	35
SNS Pair plot.....	36
Stat Model Regression Results.....	37

Introduction

1. Background

Airbnb, a San Francisco based American company has come quite a long way since it was founded in 2008. Airbnb was flourishing with the help of the rise of a global middle class, which contributed to the expansion of tourism worldwide even more than global economic growth. (Glusac, 2020) However, positive signs fade away quickly for Airbnb, like all businesses that involve human interaction as the company was hit hard by the Covid-19 pandemic. The pandemic, which was the ultimate over tourism disrupter, reduced the revenue generation of Airbnb to a significant point. To get through the crisis:

- Airbnb laid off about 25% of its workforce — about 1,900 of its 7,500 employees — and raised \$2 billion in a combination of equity and debt to make up for the poor revenue earnings. The equity portion of Airbnb dropped nearly half of what the company was worth in 2017.
- Moreover, Airbnb decided to cut down activities that are not directly related the core of its business model, such as transportation and Airbnb Studios, and scaled back its investments in hotels and luxury properties (Riley, 2022)

However, situation has started to change in 2022 and Airbnb is planning to prepare itself for the industry redemption. Therefore, I am planning to build a model which will analyze a dataset consisting of information regarding various features of Airbnb properties to find the features most responsible in determining the property demands. With the help of this model and this analysis, Airbnb will be able to prepare a complete marketing strategy including feature offerings that are most valued by its customers and hopefully the company will also be able to regain its revenue flow to pre-pandemic level or even better. Otherwise, the company will repeat its previous mistakes which may result in more customer disputes and lost market share to its competitors. However, the dataset which I am using for the analysis consist of customer data from New York, which will be suitable for the analysis as New York city a prime tourist destination also the New York is the perfect example of cultural melting pot.

2. Problem Statement

Building a machine learning model to predict the basic features that Airbnb guests value during booking a property.

3. Objectives & Measurement

Objective was to split the features based on location, price and other amenities against the demand for the property to find out following insights:

- To what extent the price of a property affects the availability.
- To what extent the location of a property affects the availability
- To what extent other features (i.e., length of travel, room type or other factors) of a property affect the availability.

4. Metrics of Measurement:

Four machine learning techniques were used in this analysis including Random Forest Regressor, Multiple Linear Regression, K Neighbor Classifier and Neural Network and picked the best model based on the Root-mean-square error (RMSE).

5. Assumptions and Limitations

5.1 Assumptions: I have chosen a dataset of Airbnb guests of New York City from year 2019. I have following assumptions for choosing the mentioned dataset.

- New York is the perfect example of cultural melting pot- The city is the home of people all around the world. Moreover, it is one of the most visited cities by the tourists all over the world.
- Few unique tourism practices are observed in New York i.e. birth tourism and immigration tourism which are not only very rare but also can offer great business opportunities to the company if addressed properly.
- Lastly, I have chosen a time frame before pandemic to concentrate on only to basic and standard features of the guests.

5.2 Limitations:

- I found two major limitations in the mentioned dataset. First, there were only number of reviews given by the guests but there was no way to measure the experience of the guests. Another, limitation was that there was not much detail information on property amenities.

Data Sources

6. Data Set Introduction

The “Airbnb” dataset was retrieved and downloaded from Kaggle. The dataset contains 48,895 entries with 16 columns including id, name, host_id, host_name, neighbourhood_group, neighbourhood, latitude, longitude, room_type, price, minimum_nights, number_of_reviews, last_review, reviews_per_month, calculated_host_listings_count and availability_365.

Kaggle Link: <https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data>

7. Exclusions

I excluded five variables from the mentioned dataset for following reasons:

- Considered as **irrelevant** data:
 - Id (listing ID), name (name of the listing), host_id (host ID), host_name (name of the host): All these four variables contain ID related data and does not have any effect on guests booking decisions.
- Considered as **redundant** data
 - reviews_per_month: This variable contains the average of total number of reviews given by guests in number of reviews columns. Therefore, I found it redundant.

8. Data Dictionary

	Name	Attribute	Description	Value
1	id	int64	listing ID	Numeric
2	name	object	name of the listing	Property listing Titles
3	host_id	int64	host ID	Numeric
4	host_name	object	name of the host	Host Names
5	neighbourhood_group	object	location	Brooklyn, Manhattan, Bronx, Queens, Staten Island
6	neighbourhood	object	area	Specific áreas under the neighborhood groups
7	latitude	float64	latitude coordinates	Numeric
8	longitude	float64	longitude coordinates	Numeric

9	room_type	object	listing space type	Entire apartment, Shared room, Private room
10	price	int64	price in dollars	Numeric
11	minimum_nights	int64	amount of nights minimum	Numeric
12	number_of_reviews	int64	number of reviews	Numeric
13	last_review	object	latest review	Date format
14	reviews_per_month	float64	number of reviews per month	Numeric
15	calculated_host_listings_count	int64	amount of listing per host	Numeric
16	availability_365	int64	number of days when listing is available for booking	Numeric

Exploratory Data Analysis

9. Data Exploration Techniques

For data exploration I have used Dataprep package in python. Using that mentioned package I have explored the following trends in the dataset.

- Frequency Distribution of variables using the bar charts.
- Interaction between two variables using the histogram.
- Correlation among individual variables using Pearson, Spearman and KendallTau.

Python Code for Exploratory Data Analysis using Data Prep:

```
##Installing Dataprep package
!pip install -U dataprep

[3] ##Importing the Airbnb Dataset
import pandas as pd
airbnb = pd.read_csv('AB_NYC_2019.csv')

[4] ##Irrelevant Variables('id','name','host_id','host_name') and Redundant Variable ('reviews_per_month') is dropped
airbnb.drop(['id','name','host_id','host_name','reviews_per_month'],axis=1,inplace = True)
```

10. Major Findings of Data Exploration:

I have found the following findings regarding the data after the exploratory analysis:

- Missing Values: last_review has missing values.
- Skewed: 'longitude', 'price', 'minimum_nights', 'number_of_reviews', 'calculated_host_listings_count', 'availability_365' these variables are skewed.
- High Cardinality: last_review and neighborhood both have high cardinality as by nature both these variables contain extreme unique entries.
- Low correlation among individual variables: The maximum positive correlation was found 0.41 between 'calculated_host_listings_count' and 'availability_365', and maximum negative correlation was found -0.44 between 'longitude' and 'price'.

Overview

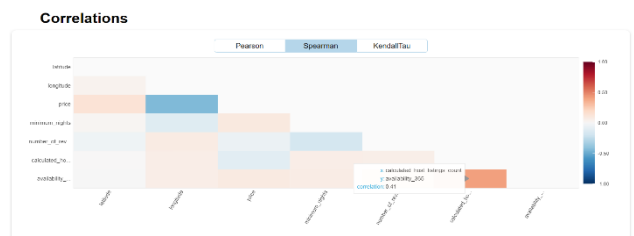
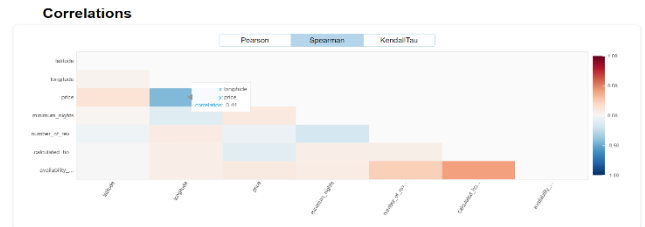
Dataset Statistics		Dataset Insights	
Number of Variables	11	last_review has 10052 (20.56%) missing values	Missing
Number of Rows	48895	longitude is skewed	Skewed
Missing Cells	10052	price is skewed	Skewed
Missing Cells (%)	1.9%	minimum_nights is skewed	Skewed
Duplicate Rows	0	number_of_reviews is skewed	Skewed
Duplicate Rows (%)	0.0%	calculated_host_listings_count is skewed	Skewed
Total Size in Memory	14.9 MB	availability_365 is skewed	Skewed
Average Row Size in Memory	320.0 B	neighbourhood has a high cardinality: 221 distinct values	High Cardinality
Variable Types	Categorical: 4 Numerical: 7	last_review has a high cardinality: 1764 distinct values	High Cardinality
		longitude has 48895 (100.0%) negatives	Negatives

```
[ ] ##Plotting the correlations among all the individual variables
from dataprep.eda import plot_correlation
plot_correlation(airbnb)
```

Stats	Pearson	Spearman	KendallTau
Highest Positive Correlation	0.226	0.407	0.331
Highest Negative Correlation	-0.15	-0.438	-0.302
Lowest Correlation	0.011	0.004	0.001
Mean Correlation	0.025	0.02	0.017

```
[ ] ##Finding the Missing Values
from dataprep.eda import plot_missing
plot_missing(airbnb)
```

Stats	Bar Chart	Spectrum	Heat Map	Dendrogram
Missing Statistics				
Missing Cells				10052
Missing Cells (%)				1.9%
Missing Columns				1
Missing Rows				10052
Avg Missing Cells per Column				913.82
Avg Missing Cells per Row				0.21



Dataset Insights



11. Data Cleaning:

I have used various data cleaning techniques in different stages of my analysis. Details of all the stages are explained as follows:

- Step-1: Removing variables irrelevant to the project

I have excluded Id, name, host_id, host_name and reviews_per_month these five variables as they were not relevant to the scope of this project.

```
[ ] ##Irrelevant Variables('id','name','host_id','host_name') and Redundant Variable ('reviews_per_month') is dropped
airbnb.drop(['id','name','host_id','host_name','reviews_per_month'],axis=1,inplace = True)
```

- Step-2: Removing variables irrelevant to the model building

I have excluded variables such as latitude and longitude which I only used to plotting the locations of the dataset into a map during my secondary data analysis.

```
[ ] ##Removing variables not useful for modeling##
airbnb.drop(['latitude','longitude'],axis=1,inplace = True)
```

- Step-3: Data Processing for Model building
 - **Converting Numeric Variables to categorical variable:** According to the summary statistics, the first 25% of values for the variable minimum nights are 1, the median is 3, and the 75% percentile is 5. So, one night to five nights or more would be a suitable range for categorization. I have also converted calculated_host_listings_count as categorical and named as 'calculated_host_listings_count_group'.
 - **Creating Dummy Variables:** Dummy variables were created for all the categorical variables ('neighbourhood_group', 'room_type', 'calculated_host_listings_count_group' and 'minimum_nights_group') to convert the categories into binary variable to ensure a separate categorical variable takes on a specific value.

```
[ ] ##Converting numeric variables into categorical variables##
airbnb['calculated_host_listings_count_group'] = 'Others'
airbnb['calculated_host_listings_count_group'][airbnb['calculated_host_listings_count'] == 1] = 'one listing'
airbnb['calculated_host_listings_count_group'][airbnb['calculated_host_listings_count'] == 2] = 'two listings'
airbnb['calculated_host_listings_count_group'][airbnb['calculated_host_listings_count'] > 2] = 'more than two listings'

airbnb['minimum_nights_group'] = 'Others'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 1] = 'one night'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 2] = 'two nights'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 3] = 'three nights'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 4] = 'four nights'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] > 4] = 'five nights or more'
```

- Step-4: Data Transformation

Data transformation is typically used to remedy skewness and outlier problems. These six variables were skewed in the dataset: longitude, price, minimum nights, number of reviews, computed host listing count, and availability 365. However, if they were all transformed, the outcomes of multiple regression would reach infinity. Therefore, I just used square root transformation for the transformation of "price" and "availability 365".

12. Summary Report

Complete Exploratory Data Analysis report link:

file:///C:/Users/abirh/OneDrive/Desktop/eda_report_hosain_ahmed_Dataprep.html#correlations

Data Preparation and Feature Engineering

13. Data Partition:

Data splitting is a common practice in predictive modelling for training and evaluating model performance. The training set is used to train the model, while the validation set is used to monitor and tune the model in order to improve its performance by optimizing the chosen model. For this analysis I will divide the data into two parts for this project: training data (70%) and validation data (30%).

Secondary Data Analysis and Data Visualization

14. Data Visualization

The dataset contains the information regarding various neighborhood. However, why guests one neighborhood over the other cannot be interpreted directly. Therefore, I have plotted the latitude and longitude information in New York City Map to find out the underlying reasons of preference for different location by the guests.

Plotting 'latitude' & 'longitude' to show the most demanding areas in the map

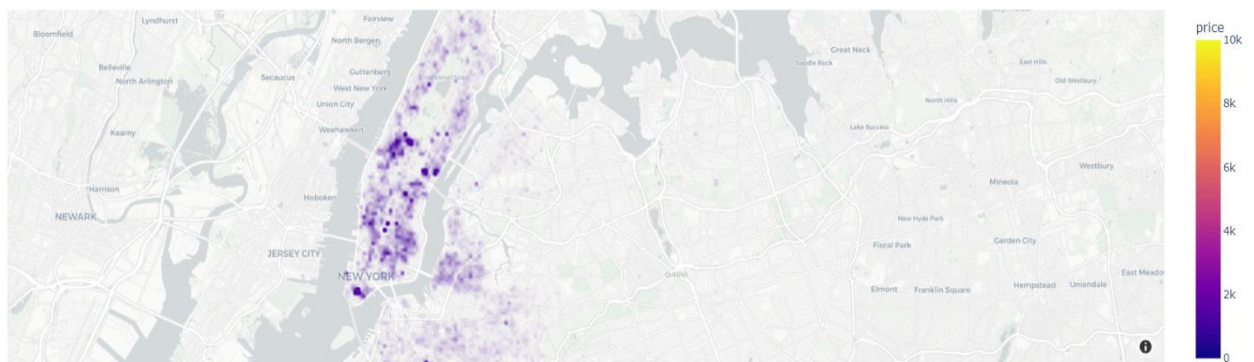
```
[ ] import plotly.express as px

lat = np.mean(airbnb['latitude'])
lon = np.mean(airbnb['longitude'])

fig = px.density_mapbox(airbnb, lat='latitude', lon='longitude', z='price', radius=2,
                        center=dict(lat = lat, lon = lon), zoom=10,
                        mapbox_style="carto-positron")

fig.show()
```

[]



Note: The above plot showing that most property offerings are located in Manhattan, on the south side of the Central Park, and around Williamsburg Bridge in Brooklyn.

15. Correlations:

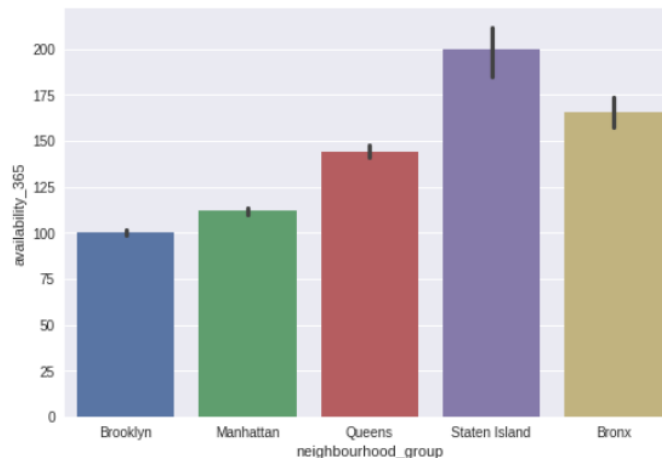
Before building the models, I have tested few co-relation among individual variables with the target variable to understand the affect of major individual variables with my target variable. This will give me an idea regarding what I may expect from my models. However, I have converted the minimum_nights numeric variable into categorical variables and named it as 'minimum_nights_group'.

```
airbnb['minimum_nights_group'] = 'Others'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 1] = 'one night'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 2] = 'two nights'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 3] = 'three nights'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] == 4] = 'four nights'
airbnb['minimum_nights_group'][airbnb['minimum_nights'] > 4] = 'five nights or more'
```

- Location with property demand: Brooklyn and Manhattan are the most preferred locations.

```
[ ] ##Finding demand variability in terms of neighbourhood_group##
sns.barplot(x="neighbourhood_group", y="availability_365", data=airbnb)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

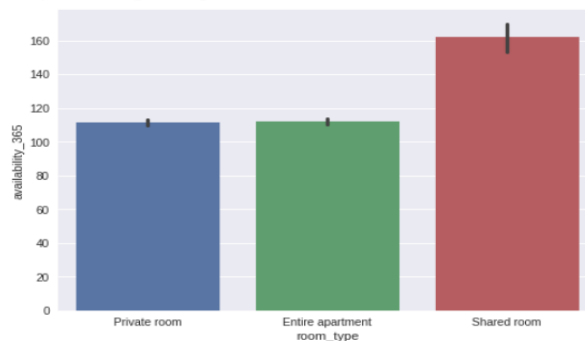
WARNING:matplotlib.legend:No handles with labels found to put in legend.
<matplotlib.legend.Legend at 0x7f971ad4fd10>



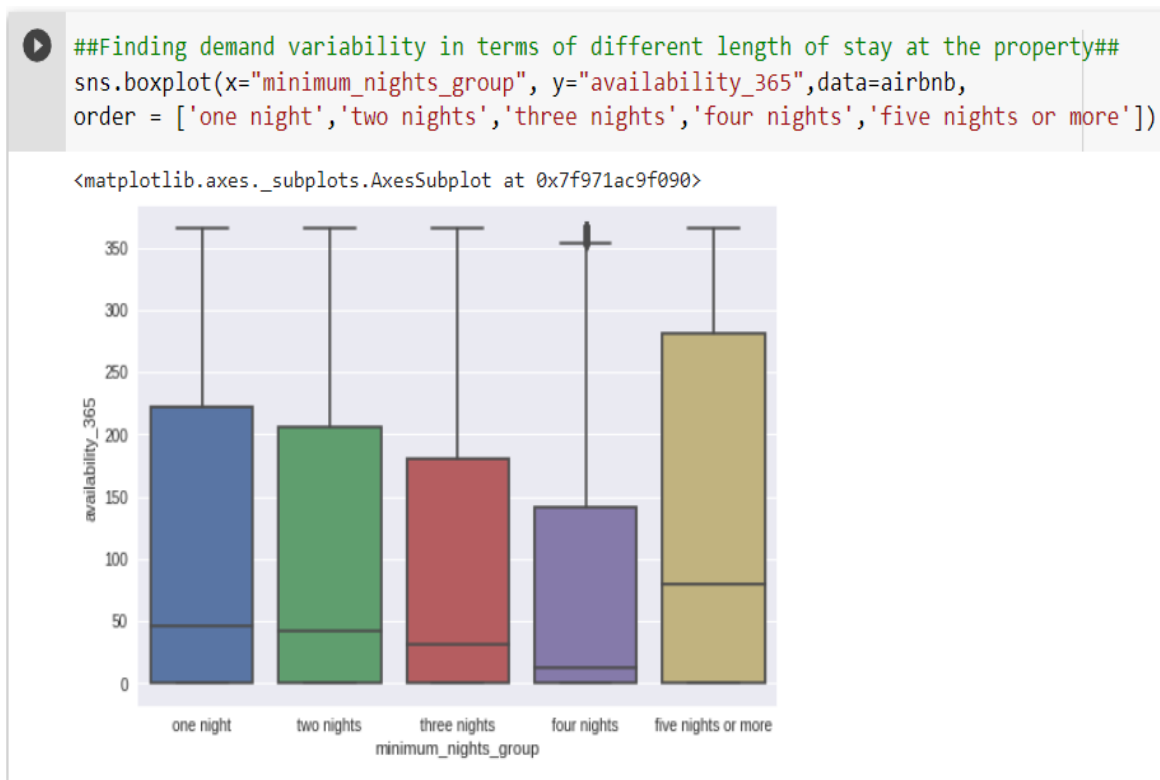
- Room Type with property demand: Entire Apartments are on demand as people value privacy.

```
##Finding demand variability in terms of room_type##
sns.barplot(x="room_type", y="availability_365", data=airbnb)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

WARNING:matplotlib.legend:No handles with labels found to put in legend.
<matplotlib.legend.Legend at 0x7f971add3850>



- Length of stay with property demand: People tends to stay longer than 3 days.



Tools and Models used

Python using Google Colab: Colaboratory, sometimes known as "Colab," is a Google Research product. Colab is particularly well suited to machine learning, data analysis, and teaching. It enables anybody to create and execute arbitrary Python code through the browser. Technically speaking, Colab is a hosted Jupyter notebook service that offers free access to computer resources, including GPUs, and requires no setup to use.

K Nearest Neighbor: K-Nearest Neighbors, or KNN for short, is utilized in a variety of organizations. KNN is a lazily learning method that is non-parametric. A approach is said to be non-parametric if it makes no assumptions about the underlying data. In other words, regardless of the characteristic the numerical values denote, it chooses depending on the selection's closeness to other data points. A lazy learning algorithm has little or no training phase, according to this definition. Therefore, as soon as fresh data points appear, we may rapidly classify them. N, S. D. (2020, August 7)

Multiple Linear Regression: Multiple linear regression is used to forecast the result of a variable based on the values of two or more other variables. It is a development of linear regression and is occasionally just referred to as multiple regression. The factors we use to predict the value of the dependent variable are known as independent or explanatory variables, whilst the variable we seek to forecast is known as the dependent variable. (Hayes, 2022)

Neural Network: Without any task-specific rules, these systems learn to do tasks by being exposed to a variety of datasets and examples. Based on biological neural network. (Ashtari, 2022)

- Neural network using impute with 100 iteration and up to 10 hidden units.

Random Forest Regressor: A random forest is a meta estimator that employs averaging to increase predicted accuracy and reduce overfitting after fitting many classification decision trees to different dataset subsamples. scikit. (n.d.)

- Regressor score is used to measure the model accuracy.
- Feature Importance feature of the model is used to find out the most significant factors.

Model Comparison Matrix: Root Mean Square Error (RMSE) measure is used to determine the model's predictive accuracy and to compare the models.

Analysis and Findings:

16. Random Forest Regressor:

Step-1: Removing irrelevant variables

Here I removed latitude and longitude variables as neighborhood and neighborhood_groups provide the similar information.

Step-2: Importing necessary packages

```
[ ] ##Removing variables not useful for modeling##
airbnb.drop(['latitude','longitude'],axis=1,inplace = True)

[ ] from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn import metrics
from wordcloud import WordCloud

import statsmodels.api as sm
import warnings

warnings.filterwarnings('ignore')
plt.style.use('seaborn')

!pip install dmbs
from dmbs import classificationSummary
```

Step-3: Data Preparation

Created dummy variables for all the categorical variables to ensure each individual category is included in the model. After creating the dummies, I have added those to the dataset and removed the original variables to avoid duplication.

```
[ ] ##Creating dummies for all the categorical variables##
dummies =['neighbourhood_group','room_type','calculated_host_listings_count_group','minimum_nights_group']
airbnb_dummies =pd.get_dummies(airbnb[dummies],prefix=dummies)

[ ] ##Adding dummies with the original dataset##
airbnb_model = pd.concat([airbnb,airbnb_dummies],axis=1)

[ ] ##Removing duplicate variables##
airbnb_model.drop(['last_review','neighbourhood','neighbourhood_group','room_type',
                  'minimum_nights_group','calculated_host_listings_count_group'],axis=1,inplace = True)
airbnb_model.head()
```


Step-4: Fitting the Random Forest Regressor Model Analyze the scores

```
▶ ##Fitting the Random Forest Regressor##  
rf_regressor = RandomForestRegressor(n_estimators=100,random_state=0)  
rf_regressor.fit(train_x,train_y)
```

```
RandomForestRegressor(random_state=0)
```

```
[17] ##Regressor score for training data#  
rf_regressor.score(train_x,train_y)
```

```
0.8514027638928328
```

```
[18] ##Regressor score validation data#  
rf_regressor.score(valid_x,valid_y)
```

```
0.22218657726040325
```

```
[19] ##RMSE score for training data#  
trainpred_y = rf_regressor.predict(train_x)  
print('Root Mean Squared Error(train):', np.sqrt(metrics.mean_squared_error(train_y, trainpred_y)))
```

```
Root Mean Squared Error(train): 50.427914003848905
```

```
[20] ##RMSE score for validation data#  
validpred_y = rf_regressor.predict(valid_x)  
print('Root Mean Squared Error(valid):', np.sqrt(metrics.mean_squared_error(valid_y, validpred_y)))
```

```
Root Mean Squared Error(valid): 116.37703341609594
```

Findings:

Model Name	Training Accuracy	Validation Accuracy	RMSE Score (Validation)
Random Forest Regressor	85.14%	22.21%	116.377

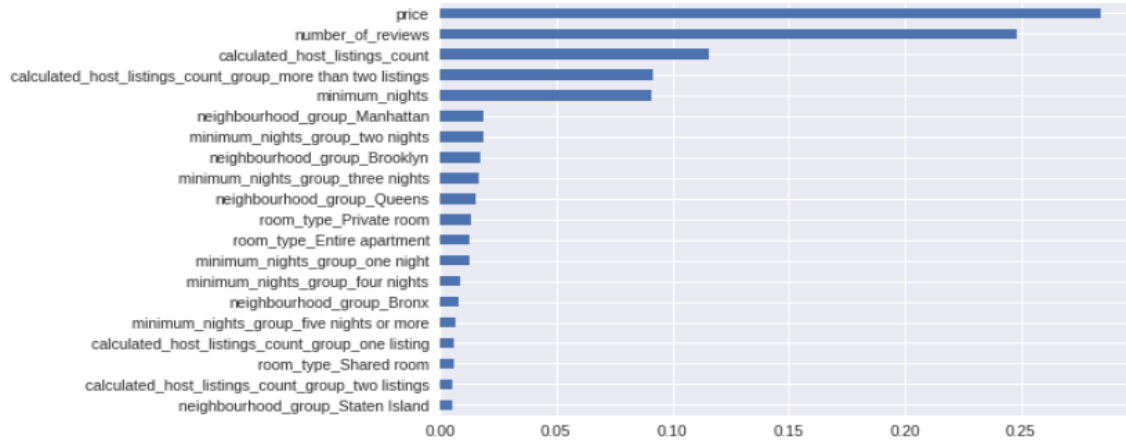
Analysis:

The Model has a high accuracy for the training data, however the model performed very poor in test data which indicates that this model is overfitting and need simplification. I could use Principal Component Analysis (PCA) to simplify the model and fix overfitting, but it would also reduce the ability to explain the results. Therefore, I did not apply PCA in this case.

Finding the best feature: price and number_of_reviews are the two best features of this model.

```
[21] ##Finding the most important feature for property demand##
feature_importance = pd.Series(rf_regressor.feature_importances_,index=x.columns)
feature_importance = feature_importance.sort_values()
feature_importance.plot(kind='barh')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f84f674a910>



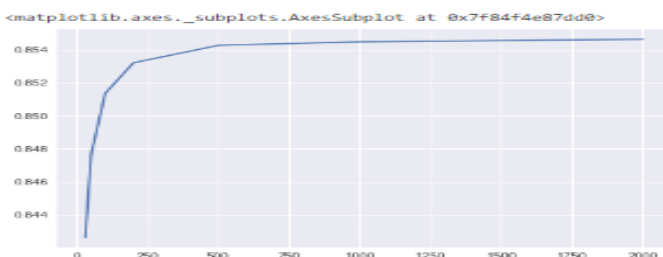
Finding the Optimal Random Forest Model: I have used trial and error to determine the optimal number of estimator and feature parameter to find the best random forest model.

```
[22] ##Finding the optimal esimator##
results_rf = []
n_estimator_options = [30,50,100,200,500,1000,2000]

for trees in n_estimator_options:
    model = RandomForestRegressor(trees,oob_score=True,n_jobs=-1,random_state=1)
    model.fit(train_x,train_y)
    print(trees," trees")
    score = model.score(train_x,train_y)
    print(score)
    results_rf.append(score)
    print("")

pd.Series(results_rf,n_estimator_options).plot()
# use 500 trees as after 500 estimator the score flattens
```

```
30 trees
0.8426152781362072
50 trees
0.8476224290202315
100 trees
0.8513524763994966
200 trees
0.8532070884691187
500 trees
0.8542663330097657
1000 trees
0.8544778523509824
2000 trees
0.8546351312598314
```



```

[23] ## finding max number of features parameter tuning##
results_rf = []
max_features_options = ['auto',None,'sqrt','log2',0.9,0.2]

for max_features in max_features_options:
    model = RandomForestRegressor(n_estimators=500,oob_score=True,n_jobs=-1,
                                random_state=42,max_features=max_features)
    model.fit(train_x,train_y)
    print(max_features," option")
    score = model.score(train_x,train_y)
    print(score)
    results_rf.append(score)
    print("")

pd.Series(results_rf,max_features_options).plot(kind='barh')

# use auto option

auto option
0.8542236373108447

None option
0.8542236373108447

sqrt option
0.8535584121038942

log2 option
0.8535584121038942

0.9 option
0.8543955388807647

0.2 option
0.8535584121038942

<matplotlib.axes._subplots.AxesSubplot at 0x7f84f66a3a90>

```



max_features_option	Score
0.2	0.8542236373108447
0.9	0.8543955388807647
log2	0.8535584121038942
sqrt	0.8535584121038942
None	0.8542236373108447
auto	0.8535584121038942

Score of the Best Regressor Model: This model has a score of 83.02% for the test data, which means this model can predict the guest preferred features with accuracy of 83.02%. However, since the original model was found overfitting therefore, I am considering this result.

```

[16] ##Finding the best regressor model##
rf_regressor = RandomForestRegressor(n_estimators=500,oob_score=True,n_jobs=-1,
                                    random_state=42,max_features='auto')

rf_regressor.fit(train_x,train_y)
rf_regressor.score(train_x,train_y)

0.8542236373108447

```

```

▶ rf_regressor.fit(valid_x,valid_y)
rf_regressor.score(valid_x,valid_y)

0.8302387582563964

```

17. K Nearest Neighbor:

I used the same dataset as Random Forest Regressor for the K Neighbors Classifier.

Step-1: Importing necessary packages

▼ KNeighborsClassifier

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    from sklearn import preprocessing
    from sklearn.neighbors import KNeighborsRegressor
    from sklearn.preprocessing import StandardScaler
    from sklearn.linear_model import LassoCV, LassoLarsCV
    from sklearn.model_selection import cross_val_score
```

Step-2: Fitting the K Neighbors Classifier Model Analyze the scores

```
[ ] neigh = KNeighborsClassifier(n_neighbors=3)

    #standardize data
    from sklearn.preprocessing import StandardScaler
    scaler = StandardScaler()
    scaler.fit(train_x)

    x_train = scaler.transform(train_x)
    x_test = scaler.transform(valid_x)

[ ] # Predicted class
    y_pred=neigh.predict(x_test)

[ ] # Calculate the accuracy of the model for test data
    print(neigh.score(x_test, valid_y))

    0.30785637070149297

[ ] print('Root Mean Squared Error (valid):', np.sqrt(metrics.mean_squared_error(valid_y, y_pred)))

    Root Mean Squared Error (valid): 144.03491368072815

[ ] y_pred2=neigh.predict(x_train)

[ ] # Calculate the accuracy of the model for train data
    print(neigh.score(x_train, train_y))

    0.5001363512407963

[ ] print('Root Mean Squared Error (train):', np.sqrt(metrics.mean_squared_error(train_y, y_pred2)))

    Root Mean Squared Error (train): 126.24680508800108
```

Findings & Analysis:

Model Name	Training Accuracy	Validation Accuracy	RMSE Score (Validation)
K Neighbors Classifier	50.01%	30.78%	144.0349

- Model Accuracy: The model was 30.78% accurate in predicting the variables under consideration which was better than the Random Forest Regressor and the model was not overfitting.
- However, the Root Mean Squared Error (RMSE) for this model is 144.0349 which worse than Random Forest Regressor model.

18. Neural Network:

I used the same dataset for the Neural Network as last two model used.

Step-1: Importing necessary packages & Fitting the Model

Neural Network

```
[25] from sklearn.neural_network import MLPClassifier

[26] ##Fitting the NN Model##
airbnb_NN=MLPClassifier(hidden_layer_sizes=2, activation='logistic', solver='lbfgs', random_state=1)
airbnb_NN.fit(train_x, train_y)
```

MLPClassifier(activation='logistic', hidden_layer_sizes=2, random_state=1, solver='lbfgs')

Step-2: Finding the Confusion Matrix, Accuracy Score

```
[ ] confusion_matrix(train_y, airbnb_NN.predict(train_x))

array([[ 5302,    0,    0, ...,    0,    0,    0],
       [  129,    0,    0, ...,    0,    0,    0],
       [  102,    0,    0, ...,    0,    0,    0],
       ...,
       [   74,    0,    0, ...,    0,    0,    0],
       [  137,    0,    0, ...,    0,    0,    0],
       [  409,    0,    0, ...,    0,    0,    0]])
```

```
[ ] confusion_matrix(valid_y, airbnb_NN.predict(valid_x))

array([[12231,    0,    0, ...,    0,    0,    0],
       [  279,    0,    0, ...,    0,    0,    0],
       [  168,    0,    0, ...,    0,    0,    0],
       ...,
       [  165,    0,    0, ...,    0,    0,    0],
       [   354,    0,    0, ...,    0,    0,    0],
       [   886,    0,    0, ...,    0,    0,    0]])
```

```
[ ] y_pred=airbnb_NN.predict(train_x)
print(airbnb_NN.score(train_x, train_y))

0.3614671393509681
```

```
[ ] y_pred=airbnb_NN.predict(valid_x)
print(airbnb_NN.score(valid_x, valid_y))

0.3573494609518801
```

Step-3: Finding the RMSE Score and Analyze the scores

```
[ ] ##RMSE score of the NN Model for validation data##  
pred_y = airbnb_NN.predict(valid_x)  
print('Root Mean Squared Error(valid):', np.sqrt(metrics.mean_squared_error(valid_y, pred_y)))
```

```
Root Mean Squared Error(valid): 174.09280845941106
```

```
[ ] ##RMSE score of the NN Model for training data##  
pred_y = airbnb_NN.predict(train_x)  
print('Root Mean Squared Error(train):', np.sqrt(metrics.mean_squared_error(train_y, pred_y)))
```

```
Root Mean Squared Error(train): 171.54083293189117
```

Findings:

Model Name	Training Accuracy	Validation Accuracy	RMSE Score (Validation)
Neural Network	36.14%	35.73%	174.0928

Analysis:

- Model Accuracy: The model was 35.73% accurate in predicting the variables and the model was not overfitting but both the K Neighbors Classifier and Random Forest Regressor have better accuracy scores.
- However, the Root Mean Squared Error (RMSE) for this model is 174.0928 which means this model gives the most more while predicting the guests features than Random Forest Regressor and K Neighbors Classifier model.

Finding the optimal Neural Network Model and its score:

```
[ ] from sklearn.model_selection import cross_val_score, GridSearchCV
```

```
[ ] param_grid = {'hidden_layer_sizes':[(1),(2),(3),(4),(5),(6),(7),(8),(10)]}
```

```
▶ gridsearch=GridSearchCV(MLPClassifier(activation='logistic', solver='lbfgs',  
                                     random_state=1, max_iter=500),  
                           param_grid=param_grid, cv=5, n_jobs=-1)  
gridsearch.fit(train_x, train_y)
```

```
[ ] gridsearch.best_score_
```

```
0.3648078385929243
```

```
[ ] gridsearch.best_params_
```

```
{'hidden_layer_sizes': 10}
```

19. Multiple Linear Regression:

Step-1: Import Necessary Packages and Transform Variables

Multiple Linear Regression

```
[ ] from sklearn.linear_model import LinearRegression
```

```
[ ] ##Transforming the variables using square root transform for removing outliers and skewness##  
data_trf1= airbnb_model['price'].transform([np.sqrt])  
airbnb_trf1=data_trf1.rename(columns={'sqrt':'sqrt_price'})  
print(airbnb_trf1)  
  
data_trf2= airbnb_model['availability_365'].transform([np.sqrt])  
airbnb_trf2=data_trf2.rename(columns={'sqrt':'sqrt_availability_365'})  
print(airbnb_trf2)
```

```
[ ] airbnb_reg_data = pd.concat([airbnb_model,airbnb_trf1,airbnb_trf2],axis=1)
```

```
[ ] airbnb_reg_data.drop(['price', 'availability_365'],axis=1,inplace = True)
```

```
[ ] airbnb_reg_data.head()
```

Step-2: Fitting the Multiple Linear Regression Model using Statmodel

```
[ ] # split train and validation dataset  
x = airbnb_reg_data.drop(['sqrt_availability_365'], axis=1)  
y = airbnb_reg_data['sqrt_availability_365'].astype(float)  
  
train_x, valid_x, train_y, valid_y = train_test_split(x,y , test_size=0.7, random_state=1)
```

```
[ ] ##Fitting the regression model##  
linear_model_sm = sm.OLS(train_y,sm.tools.add_constant(train_x).astype(float))  
results_sm = linear_model_sm.fit()  
print(results_sm.summary())
```

Step-3: Finding the Accuracy and RMSE Score using SK Learn and Analyze the scores

```
##Regression using SK learn##
linear_model_sk = LinearRegression()
linear_model_sk.fit(train_x, train_y)
linear_model_sk.score(valid_x, valid_y)

0.26084108787478877

[ ] validpred_y = linear_model_sk.predict(valid_x)
df = pd.DataFrame({'Actual': valid_y, 'Predicted': validpred_y})
print(linear_model_sk.score(valid_x, valid_y))

0.26084108787478877

[ ] ##RMSE score of the NN Model for validation data##
print('Root Mean Squared Error(valid):', np.sqrt(metrics.mean_squared_error(valid_y, validpred_y)))

Root Mean Squared Error(valid): 6.281613121953267

[ ] trainpred_y = linear_model_sk.predict(train_x)
df = pd.DataFrame({'Actual': train_y, 'Predicted': trainpred_y})
print(linear_model_sk.score(train_x, train_y))

0.26651606442789355

[ ] ##RMSE score of the NN Model for training data##
print('Root Mean Squared Error(train):', np.sqrt(metrics.mean_squared_error(train_y, trainpred_y)))

Root Mean Squared Error(train): 6.2265086263307134
```

Findings:

Model Name	Training Accuracy	Validation Accuracy	RMSE Score (Validation)
Multiple Linear Regression	26.08%	26.65%	6.28

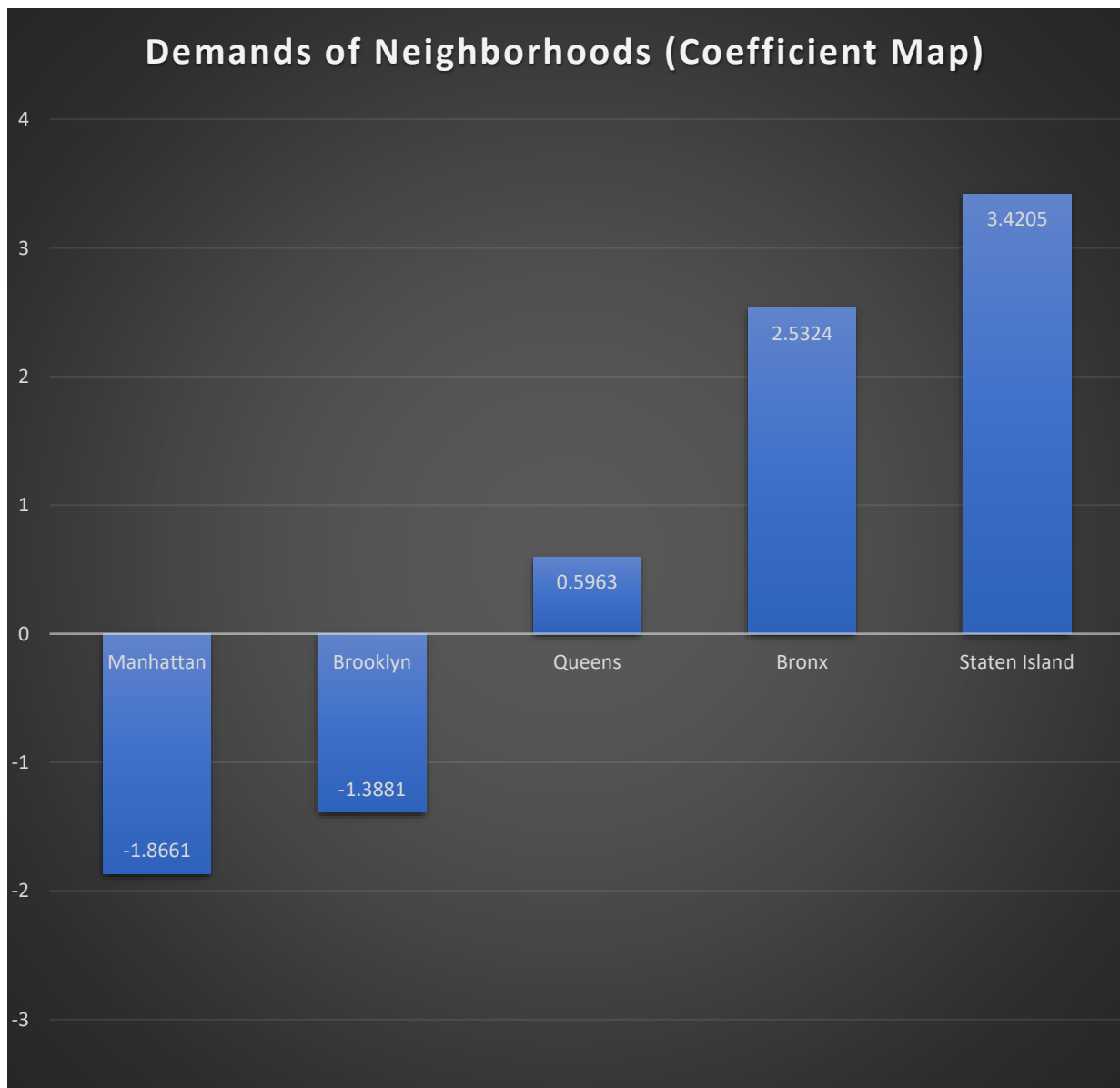
Analysis:

The model has 26.65% accuracy score and it's not overfitting. Even though Multiple Linear Regression accuracy is the lowest among all the model used this project, but it has the lowest RMSE score which mean that this model can predicting the features preferred by the Airbnb guests with lesser error than all the other models. Therefore, in my Analysis I have found Multiple Linear Regression to be the best model. variables under consideration which was better than the Random Forest Regressor but not better than K Neighbors Classifier and the model was not overfitting.

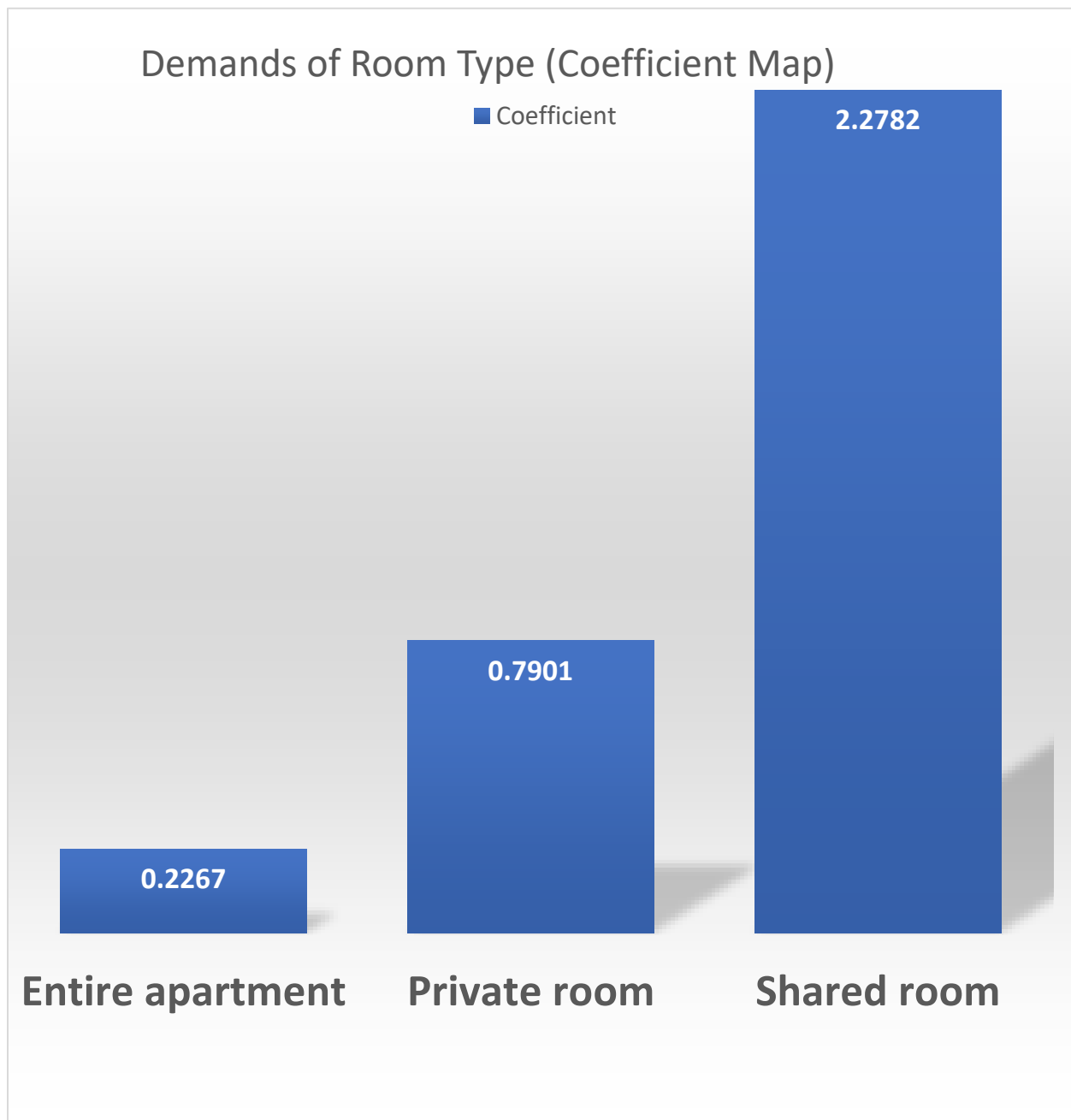
Finding the Best Features of the Model:

The R square of the model (which is 0.267) is not very high but surprisingly almost all the variables are found significant according to $p \text{ value} \leq 0.05$. However, after analysis the coefficients of the variables, I have noticed three major trends in terms of location, room type and price. coefficients measure the strength of the relationship between two variables. A correlation between variables indicates that as one variable changes in value, the other variable tends to change in a specific direction. (Frost, 2022)

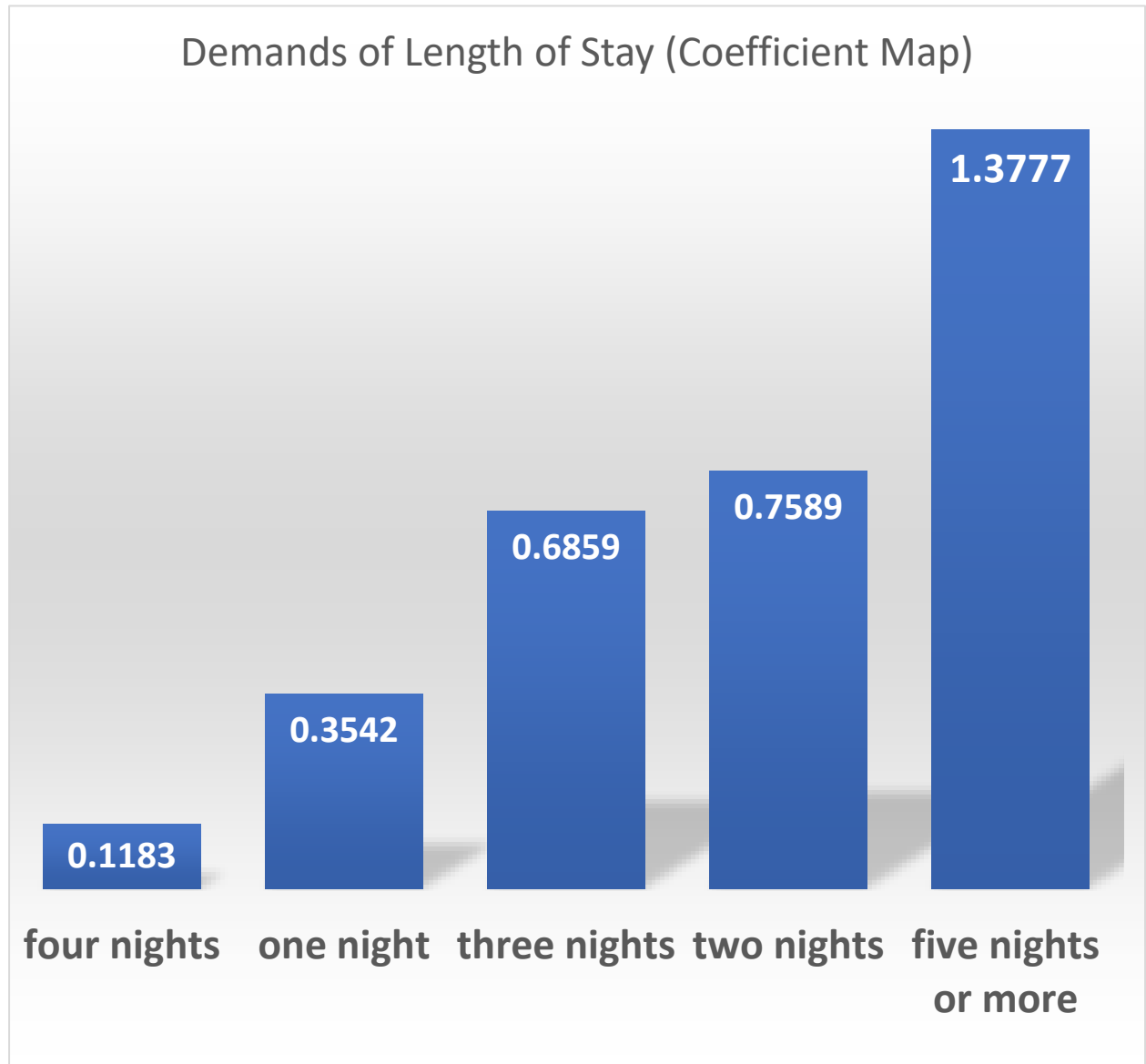
Trend-1: Airbnb Guests prefers Manhattan and Brooklyn followed by Queens however does not likes to stay in Bronx and Staten Island.



Trend-2: Airbnb Guests value their privacy as they prefer to book entire apartments and private rooms but tends to avoid shared rooms.



Trend-3: Airbnb Guests tends to stay longer than 3 nights as per the graph shown below and at a lower cost possible indicated by the positive relation between price and availability of the rooms.



Overall Insights of the Findings:

- Insights that we can the model tells us:
- People likes to stay I around downtown area where transport and other civil facilities are convenient.
- People prefers to stay for long time at cheaper rates.
- People value their privacy.

Recommendations:

- Queens area has demand but its lower than expectations.
- Probable Reasons:
- Under the table bookings for longer stays for more than 30days. Usually practiced in situations like birth tourism.
- Solution: Company must offer extra benefits to the hosts for longer bookings for more than 30days.
- Considering the popularity of Manhattan and Brooklyn Airbnb should acquire more offerings close to subways stations and superstores.
- Should shrink property offering in area like Bronx and Staten Island.
- Offers discounts to encourage longer stays for more than 7days.

Conclusions:

- Further Analysis: Airbnb can use text mining of the guests reviews to find out more on privacy and feature preference issues which was beyond the scope of this dataset.
- Word of Caution: Since the guest preferences are ever changing therefore Airbnb should monitor the results of this model in every 3 months and consider rebuilding the model if the RMSE score drifts for more than 25% from the original results.

Validation and Governance

Variable Level Monitoring:



latitude

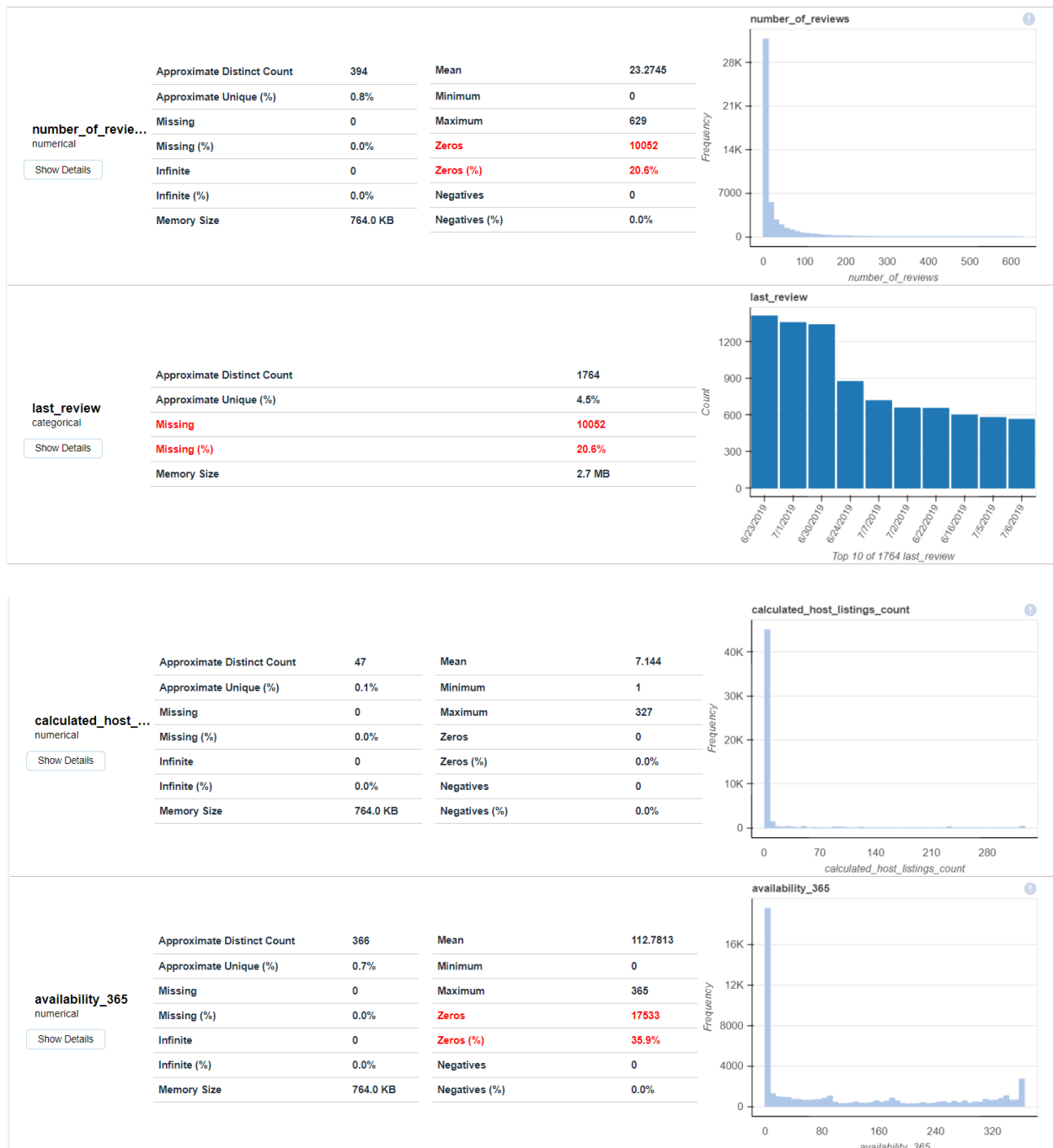
numerical

Show Details

Approximate Distinct Count	19048	Mean	40.7289
Approximate Unique (%)	39.0%	Minimum	40.4998
Missing	0	Maximum	40.9131
Missing (%)	0.0%	Zeros	0
Infinite	0	Zeros (%)	0.0%
Infinite (%)	0.0%	Negatives	0
Memory Size	764.0 KB	Negatives (%)	0.0%

latitude





calculated_host_listings_count

numerical

Show Details

Approximate Distinct Count	47	Mean	7.144
Approximate Unique (%)	0.1%	Minimum	1
Missing	0	Maximum	327
Missing (%)	0.0%	Zeros	0
Infinite	0	Zeros (%)	0.0%
Infinite (%)	0.0%	Negatives	0
Memory Size	764.0 KB	Negatives (%)	0.0%

availability_365

numerical

Show Details

Approximate Distinct Count	366	Mean	112.7813
Approximate Unique (%)	0.7%	Minimum	0
Missing	0	Maximum	365
Missing (%)	0.0%	Zeros	17533
Infinite	0	Zeros (%)	35.9%
Infinite (%)	0.0%	Negatives	0
Memory Size	764.0 KB	Negatives (%)	0.0%

Issues found in variables:

- Missing Values: last_review has missing values.
 - Handling of Missing Values: Number of missing values in last_review was equal to the number of Zeros in 'number_of_reviews' which proves that these missing values are

genuine. However, since I did not find the variable `last_review` relevant for regression analysis as it was not significant in previous models therefore no imputation was done.

- Skewed: `'longitude'`, `'price'`, `'minimum_nights'`, `'number_of_reviews'`, `'calculated_host_listings_count'`, `'availability_365'` these variables are skewed.
 - Handling of Skewness:
 - `latitude` and `'longitude'` were only used for data visualization and skewness in these variables were inevitable.
 - `'minimum_nights'`, `'number_of_reviews'`, `'calculated_host_listings_count'` were converted into categories by using dummy variables.
 - `'price'` and `'availability_365'` were transformed using squared root transformation.
- High Cardinality: `last_review` and `neighborhood` both have high cardinality as by nature both these variables contain extreme unique entries.
- Low correlation among individual variables: The maximum positive correlation was found 0.41 between `'calculated_host_listings_count'` and `'availability_365'`, and maximum negative correlation was found -0.44 between `'longitude'` and `'price'`.


Variable Drift Monitoring: Apart from Multiple Linear Regression, all the models showed high fluctuation in RMSE values between train and test (validation) dataset. This trend indicates that these models will drift significantly over time.

Model Health & Stability: Considering the high RMSE scores and high fluctuation between train and test dataset indicates that apart from Multiple Linear Regression Model has stability and can predict the variables in a consistent and reliable manner.

Tolerance for Drift of Each Variable: The industry where Airbnb operates is consistently changing and the industry environment has changed quite a lot since 2019 (Year of the Dataset) even if we don't consider the pandemic. Therefore, tolerance for the drift for this model should not be more than six months. However, if the company experience significant change in the model compared to industry average the model should be rebuild.

Risk Tiering: Since the model deals with the variables that are involved with individual customers preferred feature for property booking and these features does not include safety or any regulatory issues therefore, I believe the risk tiering will be minimal.

References

- 2019 Airbnb NYC availability prediction . (n.d.). Retrieved August 19, 2022, from <https://leo-you.github.io/Airbnb-Availability-Prediction/>
- Ashtari, H. (2022, August 3). *What is a neural network? definition, working, types, and applications in 2022*. Spiceworks. Retrieved August 19, 2022, from <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-a-neural-network/>
- Dgomonov. (2019, August 12). *New York City Airbnb Open Data*. Kaggle. Retrieved August 19, 2022, from <https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data>
- Form 10-K Airbnb. Airbnb. (n.d.). Retrieved August 19, 2022, from <https://investors.airbnb.com/financials/sec-filings/sec-filings-details/default.aspx?FilingId=15605629>
- Frost, J. (2022, July 22). *Interpreting correlation coefficients*. Statistics By Jim. Retrieved August 19, 2022, from <https://statisticsbyjim.com/basics/correlations/>
- Gcdatkin. (2020, October 30).  *NYC Airbnb Availability Prediction*. Kaggle. Retrieved August 19, 2022, from <https://www.kaggle.com/code/gcdatkin/nyc-airbnb-availability-prediction/notebook>
- Glusac, E. (2020, September 24). *The Future of Airbnb*. The New York Times. Retrieved August 11, 2022, from <https://www.nytimes.com/2020/09/24/travel/airbnb-pandemic.html>
- Hayes, A. (2022, July 8). *Multiple linear regression (MLR) definition*. Investopedia. Retrieved August 19, 2022, from <https://www.investopedia.com/terms/m/mlr.asp>
- N, S. D. (2020, August 7). *K-nearest neighbors algorithm*. Medium. Retrieved August 19, 2022, from <https://medium.com/analytics-vidhya/k-nearest-neighbors-algorithm-7952234c69a4>
- RileyCNBC. (2022, February 3). *Airbnb survived Covid, but the crisis mode in "Sharing" economy stays*. CNBC. Retrieved August 11, 2022, from <https://www.cnbc.com/2022/02/03/airbnb-survived-covid-but-the-crisis-mode-in-sharing-economy-stays.html>

Sklearn.random forest regressor. scikit. (n.d.). Retrieved August 19, 2022, from <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

Statista. (n.d.). *Topic: Airbnb*. Statista. Retrieved August 19, 2022, from <https://www.statista.com/topics/2273/airbnb/>

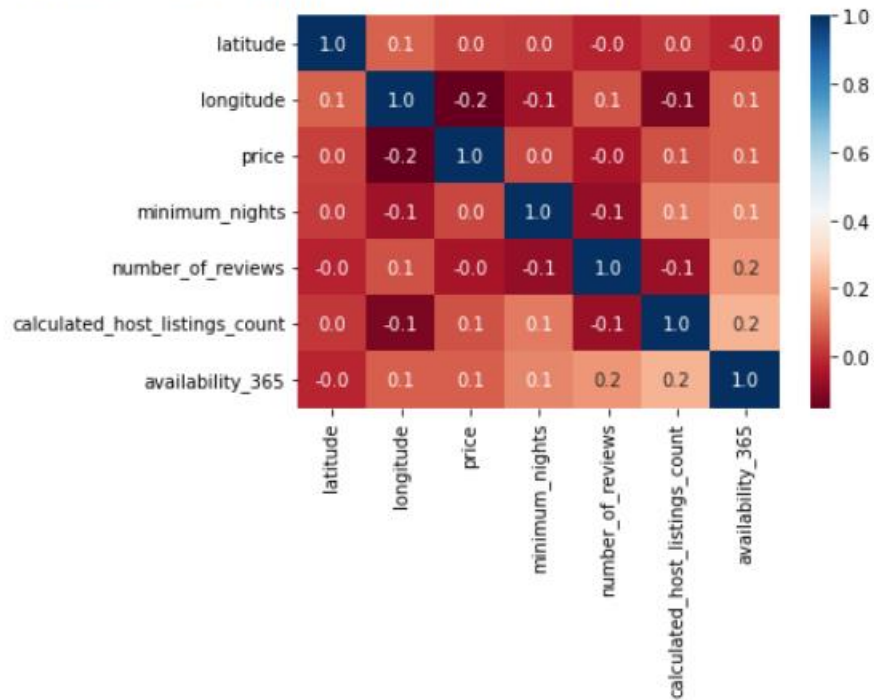
Wikimedia Foundation. (2022, August 18). *Airbnb*. Wikipedia. Retrieved August 19, 2022, from <https://en.wikipedia.org/wiki/Airbnb>

Appendix

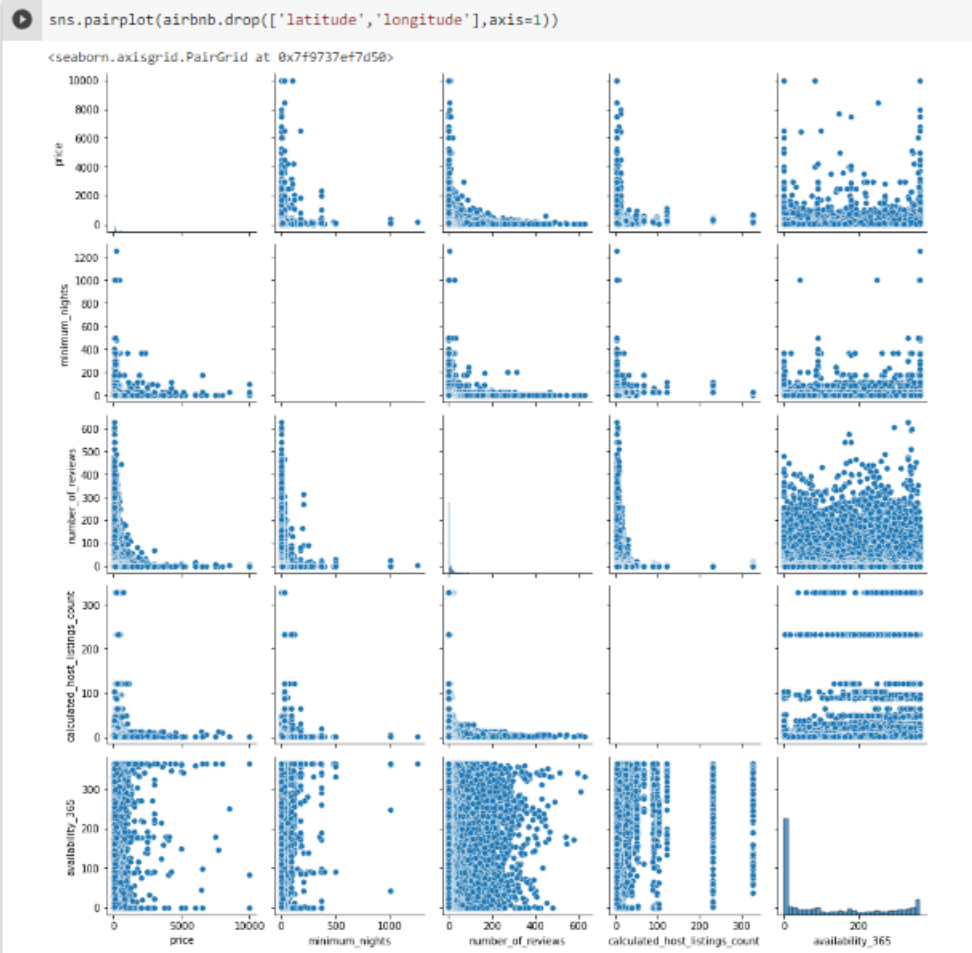
Heatmap:

```
[ ] ## using Heatmap to show the co-relation among individual variables##  
corr = airbnb.corr()  
sns.heatmap(corr, annot=True,fmt='.1f', cmap='RdBu')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f972dc92f10>



SNS Pairplot



Stat Model Regression Results

```
[ ]
=====
                        OLS Regression Results
=====
Dep. Variable:      sqrt_availability_365    R-squared:                0.267
Model:              OLS                    Adj. R-squared:           0.266
Method:              Least Squares          F-statistic:             332.7
Date:                Mon, 15 Aug 2022        Prob (F-statistic):       0.00
Time:                21:43:06               Log-Likelihood:          -47638.
No. Observations:    14668                 AIC:                    9.531e+04
Df Residuals:        14651                 BIC:                    9.544e+04
Df Model:            16
Covariance Type:     nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	3.2950	0.106	31.126	0.000	3.088	3.503
minimum_nights	0.0379	0.003	12.307	0.000	0.032	0.044
number_of_reviews	0.0384	0.001	31.008	0.000	0.036	0.041
calculated_host_listings_count	0.0180	0.002	10.479	0.000	0.015	0.021
neighbourhood_group_Bronx	2.5324	0.302	8.382	0.000	1.940	3.125
neighbourhood_group_Brooklyn	-1.3881	0.144	-9.663	0.000	-1.670	-1.106
neighbourhood_group_Manhattan	-1.8661	0.149	-12.547	0.000	-2.158	-1.575
neighbourhood_group_Queens	0.5963	0.176	3.387	0.001	0.251	0.941
neighbourhood_group_Staten Island	3.4205	0.478	7.151	0.000	2.483	4.358
room_type_Entire apartment	0.2267	0.135	1.681	0.093	-0.038	0.491
room_type_Private room	0.7901	0.114	6.939	0.000	0.567	1.013
room_type_Shared room	2.2782	0.248	9.181	0.000	1.792	2.765
calculated_host_listings_count_group_more than two listings	4.4128	0.107	41.417	0.000	4.204	4.622
calculated_host_listings_count_group_one listing	-1.7909	0.081	-22.191	0.000	-1.949	-1.633
calculated_host_listings_count_group_two listings	0.6731	0.114	5.918	0.000	0.450	0.896
minimum_nights_group_five nights or more	1.3777	0.110	12.515	0.000	1.162	1.593
minimum_nights_group_four nights	0.1183	0.169	0.702	0.482	-0.212	0.449
minimum_nights_group_one night	0.3542	0.103	3.432	0.001	0.152	0.557
minimum_nights_group_three nights	0.6859	0.117	5.886	0.000	0.457	0.914
minimum_nights_group_two nights	0.7589	0.103	7.398	0.000	0.558	0.960
sqrt_price	0.2845	0.013	21.971	0.000	0.259	0.310

```
=====
Omnibus:            599.318    Durbin-Watson:           2.000
Prob(Omnibus):      0.000     Jarque-Bera (JB):        449.524
Skew:                0.335     Prob(JB):                2.44e-98
Kurtosis:            2.465     Cond. No.:               3.78e+17
=====

```

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The smallest eigenvalue is 2.44e-28. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.