Airbnb Price Prediction
Using
Machine Learning Algorithms

Predicting the pricing of the rental property on Airbnb was a very interesting and challenging task. Using the acquired data science knowledge, I performed all the steps of a data science project life cycle, including Data Acquisition, Data Cleansing, Data Manipulation, Feature Engineering, Modeling, Evaluation, Optimization, and Testing. This project aims to develop a reliable price prediction model using data processing and machine learning techniques to help the owners of Airbnb and customers by providing the price evaluation using the Kaggle Airbnb dataset. The dataset includes various helpful attributes and features, which include id, name, host_id, host_name, neighbourhood_group, neighborhood, latitude, longitude, room_type, minimum_nights, number_of_reviews, last_review, reviews_per_month, floor, noise(dB), and price.

Here is the description of each variable in the dataset.

1. id -> Airbnb Listing Id
2. name -> Airbnb Listing Name
3. host_id -> Host Name who is hosting the Airbnb Listing
4. Host_name -> Name of the host
5. neighbourhood_group -> Name of Borough in New York
6. neighbourhood -> Name of Neighbourhood in New York
7. latitude -> Latitude of Airbnb Listing
8. longitude -> Longitude of Airbnb Listing
9. room_type -> Type of Room
10. minimum_nights -> Minimum number of nights to stay for that particular Airbnb Listing
11. number_of_reviews -> Number of Reviews of that particular Airbnb Listing
12. last_review -> Date of the last review of that particular listing
13. reviews_per_month -> Average Reviews each month
14. floor → The floor of the airbnb listing
15. noise (dB) → Average sound levels recorded in dB
16. price -> Price of that particular Airbnb Listing

References: Kaggle Airbnb Dataset

Dataset

The public Airbnb dataset for New York City was used as the main data source for this study. The dataset included 39119 entries, each with 16 features. I tried to gain a better understanding of the dataset using the head(), info(), and describe() methods and gathered information about the count, max, min, and frequency of each variable.

Data Cleaning

For the initial prepossessing, I performed data quality analysis and data cleaning techniques, including removing missing values, filling in missing values using the imputation technique, removing duplicates, removing outliers, normalization, and converting categorical variables.
Steps:

1) Imported the data analysis libraries like NumPy, pandas, re, string, and collections
2) Renaming columns: I renamed the columns by giving each column a unique and informative name to make it clearer and informative.
3) Unuseful variable: One column named ‘NaN’ consisted of all the NaN values; therefore, I dropped that specific column
4) Normalization:
    a) **id:** there were some id having '$$$', therefore replaced it using str.replace() method
    b) **host_name:** removed punctuations in 'host_name'
    c) **neighbourhood_group**: looked at the unique values in the 'neighbourhood_group' column and found five main boroughs of New York City. Replaced all duplicates in the column for better consistency. Now, I have five main boroughs - (Manhattan, Brooklyn, Queens, Staten Island, Bronx) in the neighbourhood_group column

d) **neighbourhood:** Removed all punctuations, whitespaces except before a capital letter, duplicated words, and the unuseful words that represented its neighbourhood_group. I tried removing repetitive words as we already have a separate column for neighbourhood_group

e) **latitude:** Removed all alphabetic characters from the column and removed all radian sign '°' from the column for better readability and consistency

f) **longitude:** Removed all alphabetic characters from the column and removed all radian sign '°' from the column for better readability and consistency

g) **minimum_nights:** Removed all alphabetic characters in the column so that the column has only numeric values representing the minimum number of nights people stayed at the particular Airbnb

h) **number_of_reviews:** Remove all alphabetic characters in column

i) **last_review: (date column) - Feature engineering:** created new columns for year, month, and day

j) **reviews_per_month:** Remove all alphabetic characters in the column as it represents the number of reviews for the particular Airbnb

k) **floor:** Removed the word 'floor' from the column and converted first, second, third... to 1,2,3.

l) **noise(dB)**: In the column 'noise(dB)', dropped all non-numerical values

m) **price:** In the column 'price,' removed '$' so that the column has only numerical values

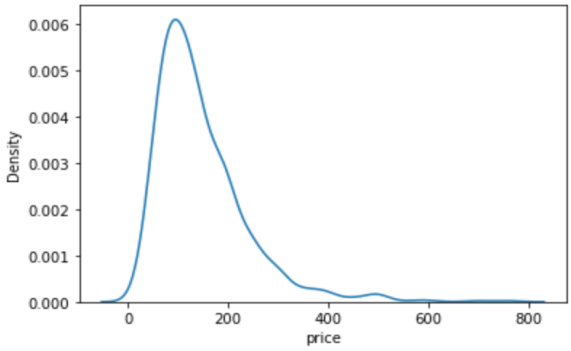
After performing the data cleaning processes, I performed imputation for all the columns having missing values. For the columns: number_of_reviews, reviews_per_month, floor, year, month, and day, I filled in the missing values by taking the mean of all the values.

Exploratory Data Analysis

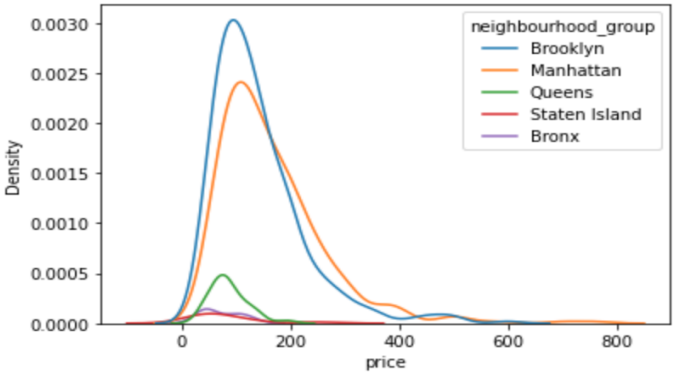
Data Visualizations

Data visualization is one of the best ways to gain a better understanding of the data by finding trends, patterns, and relationships between different attributes in the dataset. It helps us to find the important features in the data.

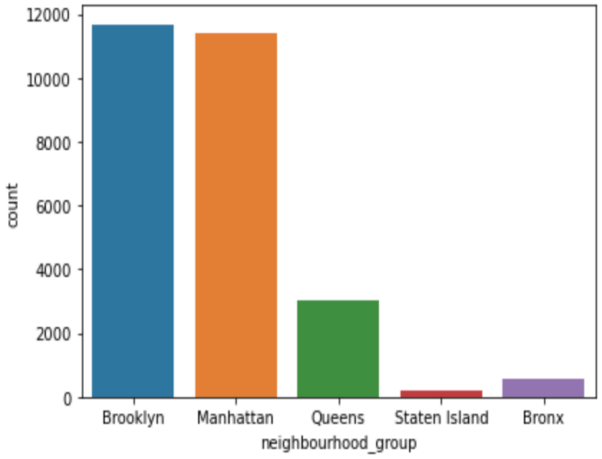
Kdeplot: This plot shows that Airbnb prices mostly range from 0 to 800.



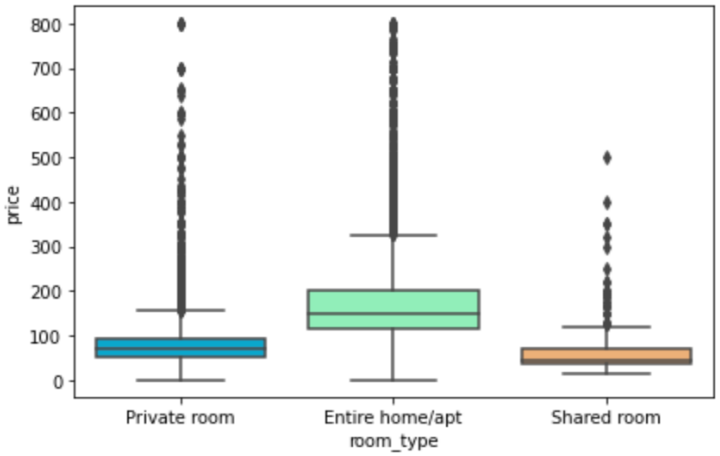
This kdeplot and boxplot represent the Airbnb price range for each neighbourhood_group which shows that Airbnb in Brooklyn and Manhattan are the most expensive ones, which seems to be true as Manhattan and Brooklyn have the most tourist attractions.



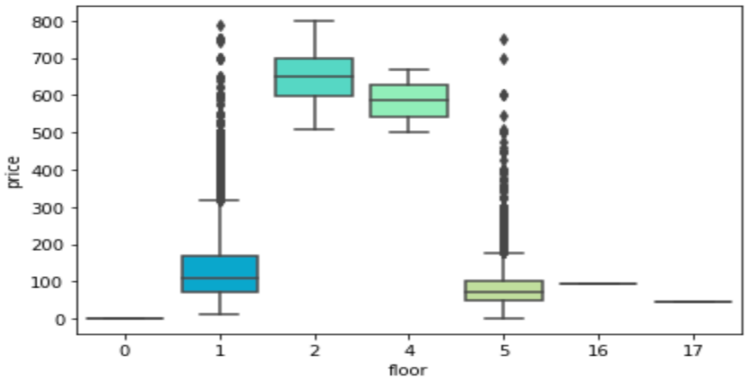
Countplot: it shows the frequency of the neighbourhood_group column.
Brooklyn and Manhattan have the highest frequency, which means that this dataset has more Airbnbs located in Brooklyn and Manhattan than Airbnbs located in the other boroughs.

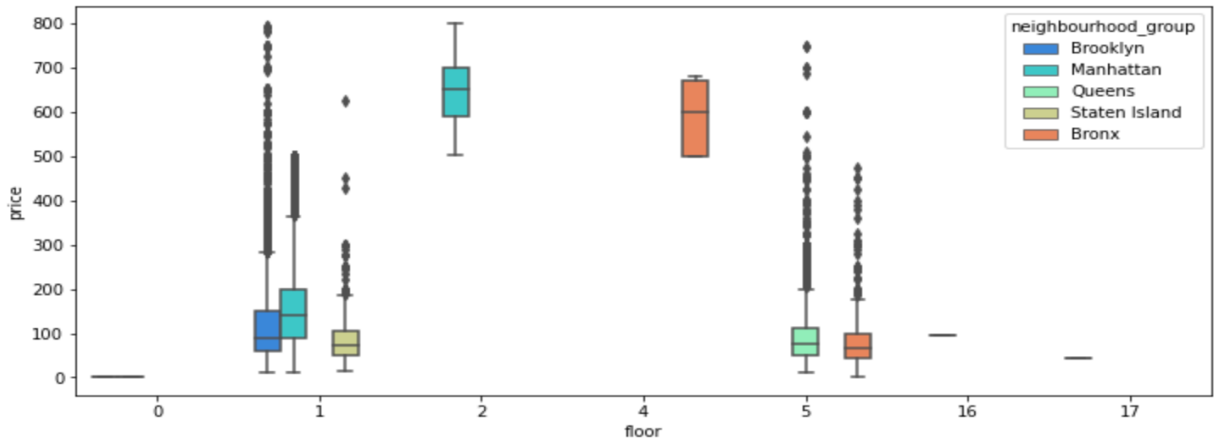


Boxplot: The boxplot represents the Airbnb prices for the different room_type distributions. Here, prices of private rooms and shared rooms are lower than prices of entire homes/apt

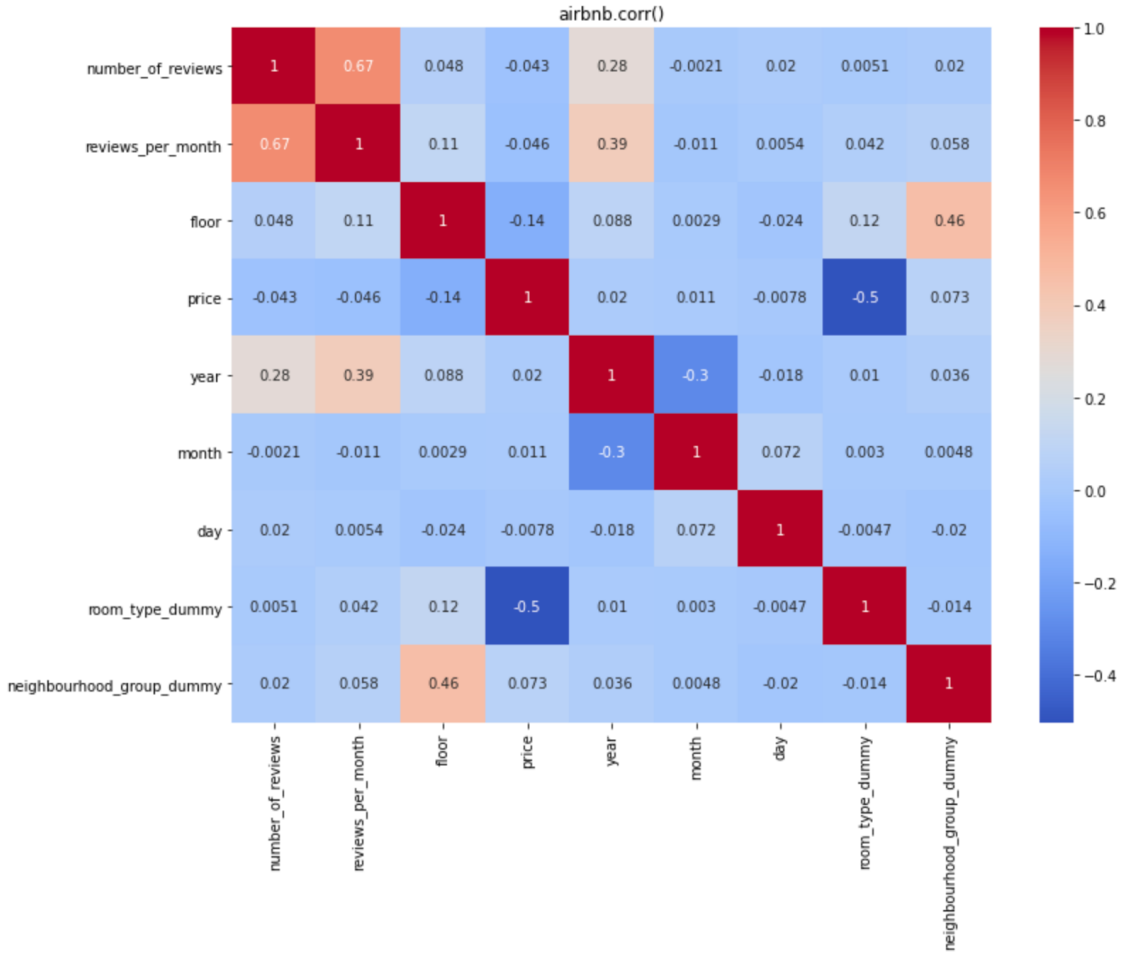


Boxplot: This boxplot represents the Airbnb prices for the various floors. If we look at the frequency, we can see the first, second, and third floors have the highest occupancy. Therefore, if we compare the first, second, and third floors, we can see that the prices of the second and third floors are higher than the first floor. One of the reasons behind people choosing a higher floor could be the view. We can see that there is a major difference between the prices of the first floor and the second floor in Manhattan, which is a significant tourist place.





Correlation Plot



Top 10 variables that were highly correlated

We can see that neighbourhood_group, latitude, and minimum_nighhts are among the most important features for predicting Airbnb prices as they are highly correlated with the price column.

	Correlation
neighbourhood_group_dummy	0.051727
latitude	0.045591
month	0.016284
minimum_nights	0.014971
year	0.001443
day	-0.007216
number_of_reviews	-0.033385
reviews_per_month	-0.060241
floor	-0.145633
noise(dB)	-0.199206
longitude	-0.263200
room_type_dummy	-0.514363

Hypothesis Testing

Null Hypothesis: There is no difference between the Airbnb prices (Brooklyn or Manhattan) and other neighbourhood_groups

Alternate Hypothesis: Airbnb prices in Brooklyn or Manhattan are greater than prices in other neighbourhood_groups

1) I calculated the Z score by finding mean, raw score, and the standard deviation
2) I also calculated the p-value from Z score
3) I got the p-value 0 which is less than 0.05. Therefore, we reject the null hypothesis

```
prices_brook_man mean value: 123.34680125987505
prices_all mean value: 0.14674446222956575
prices_brook_man std value: 110.31116894670267
prices_all std value: 0.3538509926995668
0.0
reject null hypothesis
```

Null Hypothesis: There is no difference in Airbnb prices for private room type and entire home/apt

Alternate Hypothesis: The Airbnb prices for private room type are lower than prices for entire home/apt

1) I calculated the Z score by finding mean, raw score and the standard deviation
2) I also calculated the p-value from Z score
3) I got the p-value 0 which is less than 0.05. Therefore, we reject the null hypothesis

```
home_price mean value: 97.17790571590851
room_price mean value: 39.73994423503898
home_price std value: 122.49462523909588
room_price std value: 57.79727995996652
0.0
reject null hypothesis
```

Data Preprocessing

Outlier detection: I tried detecting outliers using the IQR method and removed all points that lie outside the range defined by the quartiles +/- 1.5 * IQR. I performed outlier detection for the variables below:

'noise(dB)','minimum_nights','longitude','latitude','number_of_reviews','reviews_per_month', 'price', 'floor'

Feature Extraction: Using the last_review (date) column, I added the year, month, and day in the dataset

Correlation: I found the correlation between the variable ‘price’ and all the other numerical variables

Converting categorical variables: I converted two categorical variables (room_type and neighbourhood_group) into dummy variables to run the algorithm on it. Using label encoder, converted categorical values in neighbourhood column to numerical values.

Standardizing variables: Using StandardScaler() method, I standardized all the variables except the targe variable.

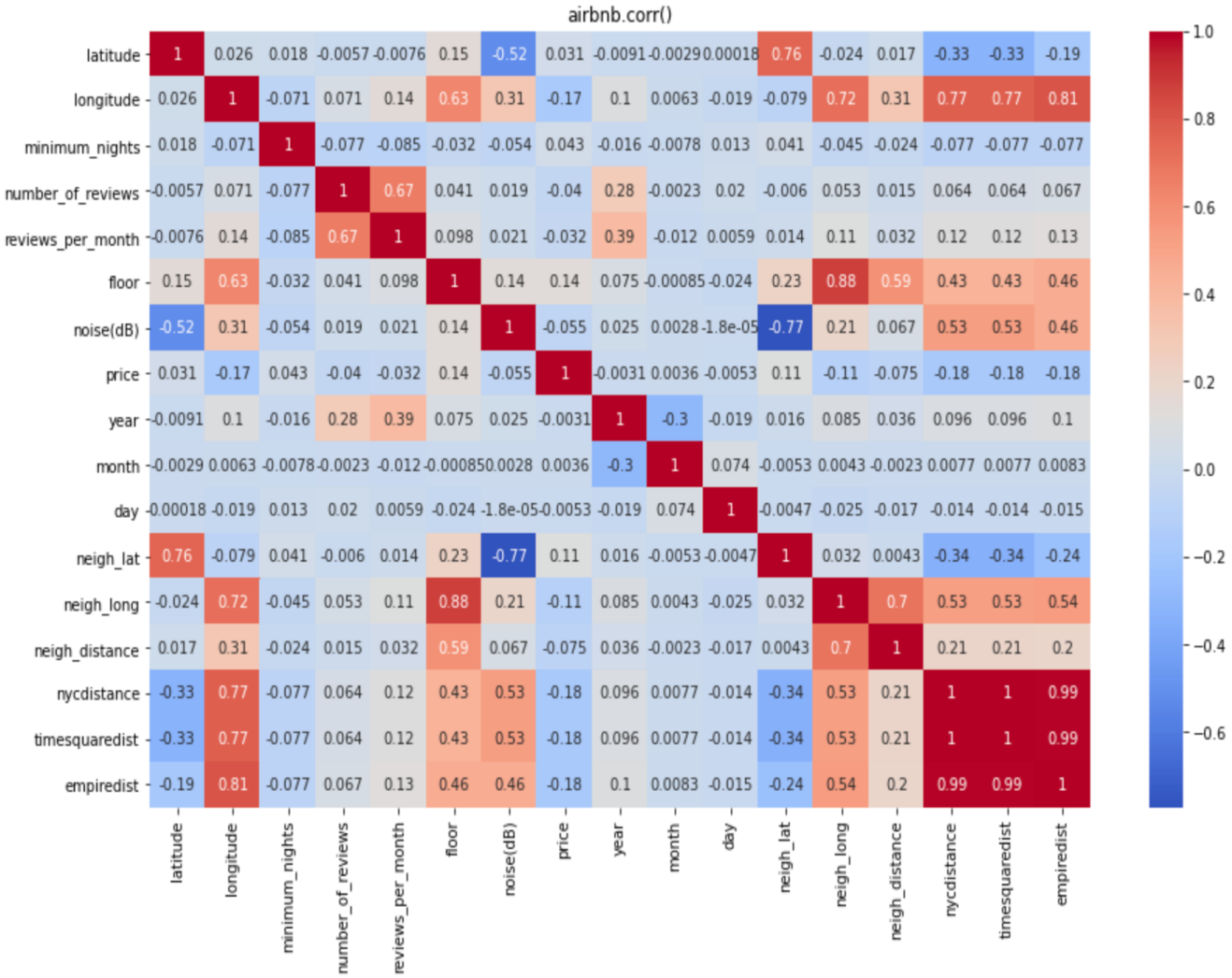
Feature Engineering

Using the domain knowledge of the data, I found the features that will be helpful for predictive modeling.

1) I found the latitude and longitude of each neighborhood function and tried finding the proximity of interest between Airbnb and their respective neighborhood.
2) I found the proximity of interest between Airbnb and New York City, the Empire State building, Central Park, and Time Square using their latitude and longitudes. I chose

these places as it is likely for the people who rent an Airbnb in New York to visit these places.

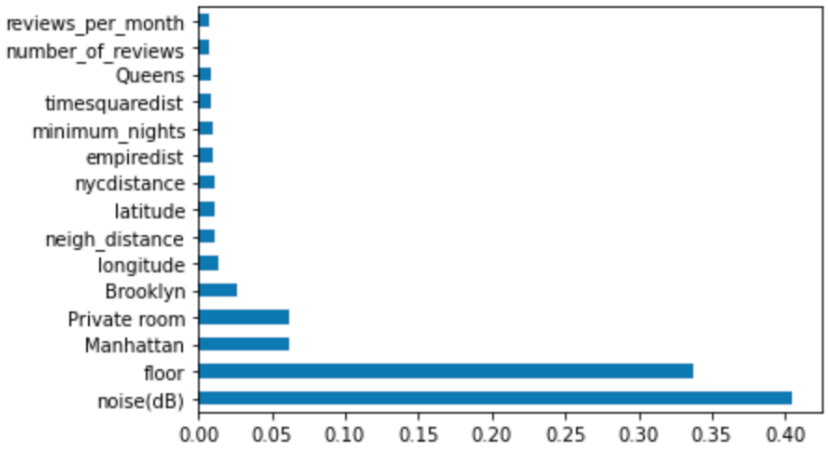
Correlation with the new variables



We can see that there is a 0.99 correlation between “nycdistance” and “empiredist”, therefore, there is no need to have two variables that have more or less the same correlation with the price. Therefore, I dropped “nycdistance” in the data.

Feature Selection

Using the ExtraTreesRegressor, I found the variables that are important for the prediction of price. I considered adding or dropping some variables in the predictive modeling by looking at this feature importance plot.



Feature Importance

```
importance = lm_model.coef_
# summarize feature importance
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
plt.bar([x for x in range(len(importance))], importance)
plt.show()
```

```
Feature: 0, Score: 0.06021
Feature: 1, Score: -117.15553
Feature: 2, Score: 28.83156
Feature: 3, Score: 0.10530
Feature: 4, Score: -0.02293
Feature: 5, Score: 0.45011
Feature: 6, Score: 19.12401
Feature: 7, Score: 305.33617
Feature: 8, Score: -0.00735
Feature: 9, Score: 0.03677
Feature: 10, Score: 0.03677
Feature: 11, Score: -0.11666
Feature: 12, Score: 2909.72772
Feature: 13, Score: 6910.55369
Feature: 14, Score: 4870.49224
Feature: 15, Score: -75.02680
Feature: 16, Score: -85.82565
```

Predictive Modeling

Machine Learnng Algorithms

Calculated mean squared error using both train test split method and K fold cross validation technique

K-fold cross validation

```
cv = KFold(n_splits=10, random_state=1, shuffle=True)
scores = cross_val_score(lm, X, y, cv=cv,scoring='neg_mean_squared_error')
print("Cross-validated scores:", scores)
```

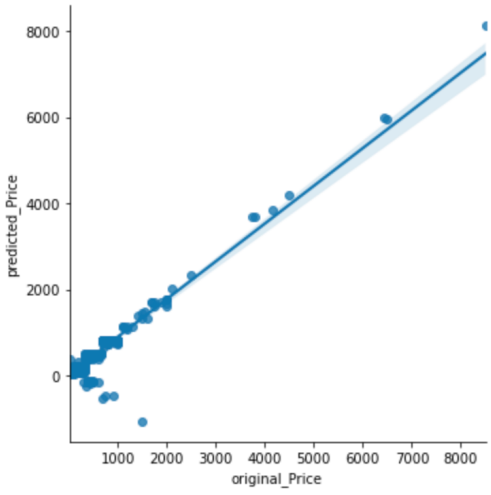
```
Cross-validated scores: [ -3719.63138873  -3900.14510018  -7978.1479953  -15964.13432345
  -9046.34819068 -16225.87504395  -9477.59872391  -4025.49835482
  -5561.72638534  -4192.666951  ]
```

```
mean(((scores)))
```

```
-8009.177245735688
```

Linear regression:

Mean squared error : 8009.177245735688




```
cv = KFold(n_splits=10, random_state=1, shuffle=True)
scores = cross_val_score(lm, X, y, cv=cv,scoring='neg_mean_squared_error')
print("Cross-validated scores:", scores)
```

```
Cross-validated scores: [ -3719.63138873  -3900.14510018  -7978.1479953  -15964.13432345
  -9046.34819068 -16225.87504395  -9477.59872391  -4025.49835482
  -5561.72638534  -4192.666951  ]
```

```
print('Cross validation MAE:', mean(scores))
print('Cross validation RMSE:', sqrt(mean(absolute(scores))))
```

```
Cross validation MAE: -8009.177245735688
Cross validation RMSE: 89.49400675875277
```

Random Forest: 7820.6917079220975

```
# Performed hyperparameter tuning, and got the best parameters for random forest regressor
rfc = RandomForestRegressor(n_estimators=100,
                            criterion='squared_error',
                            max_features=6,
                            n_jobs=-1,
                            random_state=1)
```

```
cv = KFold(n_splits=10, random_state=1, shuffle=True)
scores = cross_val_score(rfc, X, y, cv=cv,scoring='neg_mean_squared_error')
print("Cross-validated scores:", scores)
```

```
Cross-validated scores: [ -4344.97469945  -9909.72690226  -9844.98147366 -13194.01908794
  -4802.1695604  -14881.13169307  -6175.67259358  -4721.76895938
  -4707.16315136  -5625.30895811]
```

```
print('Cross validation MAE:', mean(scores))
print('Cross validation RMSE:', sqrt(mean(absolute(scores))))
```

```
Cross validation MAE: -7820.6917079220975
Cross validation RMSE: 88.43467480531659
```

Winning Model: Random Forest

Predictive Modeling on new dataset

Data Preprocessing: Performed all the steps of feature engineering on the new dataset

Predicting the Airbnb prices using two machine learning algorithms :

Random Forest

Mean squared error: 3083.0917367477277

```
cv = KFold(n_splits=10, random_state=1, shuffle=True)
scores = cross_val_score(rfc, X, y, cv=cv,scoring='neg_mean_squared_error')
print("Cross-validated scores:", scores)
```

```
Cross-validated scores: [-2959.66750564 -2788.20147756 -3003.7469706  -3136.19401316
 -3033.44858563 -3443.28554867 -3009.21991784 -3225.08728868
 -3279.87116157 -2952.19489814]
```

```
mean(scores)
```

```
-3083.0917367477277
```

```
print('Cross validation MAE:', mean(scores))
print('Cross validation RMSE:', sqrt(mean(absolute(scores))))
```

```
Cross validation MAE: -3083.0917367477277
Cross validation RMSE: 55.525595329971274
```

DecisionTreeRegressor:

I performed hyperparameter tuning using GridSearch which helped me to select the parameters in the decision tree model.

Mean Squared Error: 2698.251877794187

```
cv = KFold(n_splits=10, random_state=1, shuffle=True)
scores = cross_val_score(decision_tree, X, y, cv=cv,scoring='neg_mean_squared_error')
print("Cross-validated scores:", scores)
```

```
Cross-validated scores: [-2671.11159115 -2493.75498541 -2637.62162874 -2766.13029658
 -2561.11248788 -2962.63255082 -2617.28423978 -2749.22255043
 -2891.6106883  -2632.03775885]
```

```
mean(scores)
```

```
-2698.251877794187
```

Best Model: DecisionTreeRegressor - 88% accuracy

I chose Decison Tree model as my best model because it gave me the lowest mean squared error. Also, I also performed feature engineering by adding proximity to interest between airbnb and tourist attraction which improved the accuracy of the model

I performed predictive modeling on the test data and got mse: 2641.96856

Conclusion:

Feature Engineering seemed to be playing one of the important role for improving the accuracy of the model. I learnt new techniques including outlier detection, feature selection, hyperparamtertuning which had an impact on the accuracy and improved the model performance. I wish, I had performed more feature engineering by including zipcode of the neighbourhood or finding proximity to interest between airbnb and train station as airbnb prices near train station are much higher compared to the other places.