**BAN 620 Course Project**

**Airbnb Market Analysis on Listing Prices**

**Team Members:**

**Puja Gundu : VW8112**

**Varshitha Pereddy: AE3909**

**Surya Taruni Bulusu : RR5880**

**Date: 05/12/2024**

Table of Contents

[Project Summary 3](#_Toc166447641)

[Introduction 3](#_Toc166447642)

[Main Chapter 4](#_Toc166447643)

[1. Develop Understanding 4](#_Toc166447644)

[2. Obtain Data for Analysis 4](#_Toc166447645)

[3-4. Explore, clean and preprocess data 4](#_Toc166447646)

[5. Determine the Data Mining Task 9](#_Toc166447647)

[6. Partition Data 10](#_Toc166447648)

[7. Techniques 10](#_Toc166447649)

[8. Algorithm and Measures 10](#_Toc166447650)

[Logistic Regression 10](#_Toc166447651)

[Decision tree 14](#_Toc166447652)

[9. Interpret the Results 15](#_Toc166447653)

[10. Deploy the best technique 22](#_Toc166447654)

[Logistic Regression 22](#_Toc166447655)

[Decision Tree 23](#_Toc166447656)

[Conclusion 23](#_Toc166447657)

[Logistic Regression 23](#_Toc166447658)

[Classification Tree 23](#_Toc166447659)

[Summary 24](#_Toc166447660)

[Bibliography 24](#_Toc166447661)

# Project Summary

The project, "Airbnb Market Analysis on Listing Prices," delves into the complexities of the Airbnb marketplace, focusing on predicting listing prices in London. Leveraging Kaggle data, the analysis aims to understand the factors influencing Airbnb listing price ranges and develop predictive models for price range prediction.

The project begins with obtaining and preprocessing the dataset, including data cleaning, handling categorical variables, and transforming the price variable into categorical data. To predict listing prices, the team adopts a novel approach of transforming numeric prices into categorical data, facilitating classification-based prediction. Logistic regression and decision trees are chosen as predictive models due to their suitability for classification tasks.

The data is partitioned into training and validation set. Initially, we started with logistic regression model and used backward elimination to remove less relevant predictors in the data sets. We also employed the decision tree model to achieve the best model by comparing the accuracy and misclassification. We have taken a step more to build random forest and boosted the tree model to improve decision tree accuracy.

# Introduction

This project explores the world of Airbnb Market analysis listings in London, offering a granular look at pricing strategies. We've compiled a comprehensive dataset encompassing various factors that influence Airbnb prices, including:

Listing details: Room type (private, shared, entire place, etc.), number of bedrooms, and distance from the city center. Host characteristics: Superhot status (a designation for experienced and highly-rated hosts). Guest experience: Overall guest satisfaction ratings for the host's listings.

By analyzing this data through spatial econometric methods, we aim to identify the key determinants that shape Airbnb pricing in London. This project sheds light on how social dynamics and geographical factors interact with pricing range strategies, ultimately influencing profitability for Airbnb hosts in the city.

This exploration will provide valuable insights for both potential Airbnb guests seeking the best value and hosts aiming to optimize their pricing strategies for maximum success.

# Main Chapter

## Develop Understanding

Our project is geared towards supporting individuals entering the Airbnb hosting market in London. We utilize area-specific data and Airbnb criteria to achieve precise price predictions. This predictive model serves as a valuable resource for aspiring hosts, providing them with insights into anticipated prices influenced by variables such as room type and attraction indexes.

Our objective goes beyond mere prediction; it's an ongoing mission. As we gather real-time data from various neighborhoods in London, we anticipate fine-tuning our model continually to improve accuracy. Ultimately, our aim is to equip new Airbnb hosts with actionable insights, enabling them to make informed decisions for a prosperous business endeavor.

## Obtain Data for Analysis

The AirbnbLondon.csv file is based on data retrieved from the https://www.kaggle.com/datasets/thedevastator/airbnb-prices-in-european-cities.

After reading the data set we presented the number of rows and columns. The number of rows and columns in the data set are (5379,19)

A screenshot of a phone

Description automatically generated

## 3-4. Explore, clean and preprocess data

For the AirbnbLondon data file we can see there are 5379 rows and 19 columns.

**A screenshot of a computer code

Description automatically generated**

There are few categorical columns which needed to convert the dummy variables.

**A screenshot of a computer code

Description automatically generated**

Here we can see the 5 observations of the data file like how representing**.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Sometimes sampling may yield too few “interesting” cases to effectively train a model

Oversampling a popular approach to oversample the rare cases, to obtain a more balanced data set.

Here, we removed near to 1% data for an better efficient model.

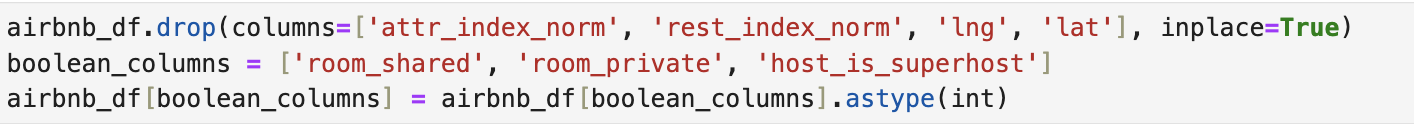
After removing the outliers we can see the rows and columns reduced to 5269 rows.

**A screenshot of a computer

Description automatically generated**

We dropped few columns in the data set which are attr\_index\_norm, rest\_index\_norm, lng, lat. As. attr\_index\_norm, rest\_index\_norm are already present as attr\_index, rest\_index. Lng and lat might not be very useful as we have the distance as one of the column.

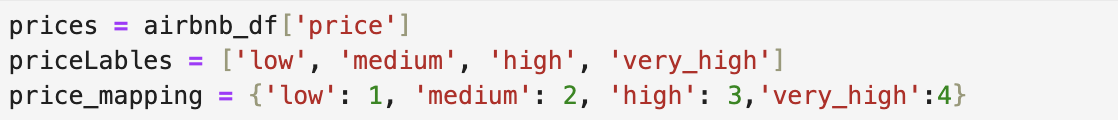
Here, we can see the Boolean columns are ‘room\_shared’, ’room\_private’, ’host\_is\_superhost.So, now we are converting these Boolean columns to integer data types.

****

The "room\_type" column is currently in object data type. We are converting this categorical data type into dummy variables. Any two-word titles will be converted by replacing the space with an underscore () character. Additionally, for categorica variables, we'll prefix them with an underscore ().

**A screenshot of a computer program

Description automatically generated**

We are transforming the "Price" column into four classes: low, medium, high, and very high. As, we want to predict the ranges of price, which is the use case of the project Therefore, we've converted these numerical values into binary variables. Specifically, we've categorized prices as follows: low as 1, medium as 2, high as 3, and very high as 4.****

To convert numerical data to categorical, we have selected two options one using mean(cut) and other using median(qcut).

Below are two histogram depicting descriptive statistics. The ‘qcut’ option divides the data into equal bins, but as the variable is price ‘cut’ options suits our use case. So, we select option 1.

**A screen shot of a graph

Description automatically generated**

**A screen shot of a graph

Description automatically generated**

Here, we can see the five observations of all the columns after dropping the variables. This is showing that the variables which are in object /Boolean converted to categorical and integer variables.

A screenshot of a computer

Description automatically generated

After, removing the unnecessary columns we represented the missing columns in the data set. So, we can see that there is no missing columns in the data set.

**A screenshot of a computer program

Description automatically generated**

Here, we can see the descriptive statistics of the data set.

**A screenshot of a data

Description automatically generated**

**A screenshot of a data

Description automatically generated**

## 5. Determine the Data Mining Task

• The aim of the task is to find the category in which the price of the property belongs to. There are four potential outcome for price ‘low’, ‘medium’, ‘high’ and ‘very\_high’.

• Hence it a classification problem.

• The predictor variables are listed below:

['room\_shared','room\_private','person\_capacity','host\_is\_superhost','multi','biz','cleanliness\_rating','guest\_satisfaction\_overall','bedrooms’,‘dist','metro\_dist','attr\_index','rest\_index','room\_type\_Entire\_home/apt','room\_type\_Private\_room','room\_type\_Shared\_room']

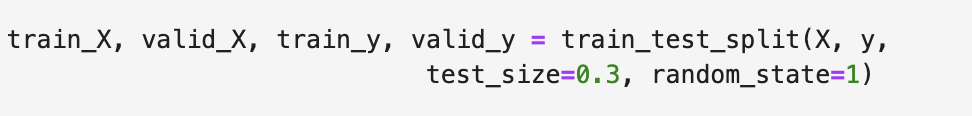
## 6. Partition Data

To avoid an overfitting situation, we use partitions to develop our data by using

train\_test\_split with test-size at 30%. While we will split the data into ‘Training’ and

‘Validation’, the Training partition contains 70% of the data to develop the model along

with Validation contains 30% of data to evaluate on new data performance.



So, after converting the training and validation partition. The training partition the data set is with values (3688,16)

For, the validation partitions the data set is with values (1581,16).

## 7. Techniques

Since the outcome variables are classifications rather than numerical values, we've opted for logistic regression and classification tree methods. To facilitate this, we converted the numerical data into categorical data.

## 8. Algorithm and Measures

### Logistic Regression

As there are 4 categories the logistic regression model results in 4 probabilities P(0), P(1), P(2), and P(3).

P(0)- Low, P(1) – Medium, P(2) – High, and P(3) – Very high

Below image represents the intercepts of each class.

**A black and white text with numbers

Description automatically generated**

**Logistic Regression (low):**

Log(odds) = 5.225 + 1.702 room\_shared + 1.990 room\_private -2.540 person\_capacity-3.060 host\_is\_superhost+2.300 multi -7.050 biz -9.700 cleanliness\_rating -6 guest\_satisfaction\_overall -1.646 bedrooms +2.010 dist +8.480 metro\_dist -3attr\_index -0 rest\_index +1.533 room\_type\_Entire\_home/apt +1.990 room\_type\_Private\_room +1.702 room\_type\_Shared\_room

**Logistic Regression (medium):**

Log(odds) = 0.928 -7.910 room\_shared +5.260 room\_private +1.060 person\_capacity -2.010 host\_is\_superhost +1.070 multi -1.410 biz +5.200 cleanliness\_rating -7 guest\_satisfaction\_overall -4.310 bedrooms +5 dist +5.89 metro\_dist -13attr\_index +0 rest\_index +1.193 room\_type\_Entire\_home/apt +5.26 room\_type\_Private\_room -7.910 room\_type\_Private\_room

**Logistic Regression (high):**

Log(Odds) = -3.798+ -7.400 room\_shared -1.563 room\_private +1.290 person\_capacity -2.620 host\_is\_superhost -1.430 multi +5.650 biz +2.020 cleanliness\_rating +1.100 guest\_satisfaction\_overall +7.550 bedrooms - 4.100 dist +9.300 metro\_dist +1 attr\_index +0 rest\_index -1.496 room\_type\_Entire\_home/apt -1.563 room\_type\_Private\_room -7.400 room\_type\_Private\_room

**Logistic Regression (very\_high):**

Log(Odds) = -2.355 +-1.710 room\_shared -9.540 room\_private +1.900 person\_capacity +7.690 host\_is\_superhost +1.200 multi +2.820 biz -1.560 cleanliness\_rating +1 guest\_satisfaction\_overall +1.323 bedrooms -1.650 dist -1.529 metro\_dist +3 attr\_index -0 rest\_index -1.230 room\_type\_Entire\_home/apt -9.540563 room\_type\_Private\_room -1.710 room\_type\_Private\_room

Below image represent for sample predictions of validation data set.

**A screenshot of a computer screen

Description automatically generated**

To enhance the model's performance, we employed the backward elimination algorithm to assess variable reduction. The analysis identified three variables for removal, aiming to achieve an improved outcome. Additionally, the algorithm identified the best-performing variables, further refining the model for optimal results.

A white background with black text

Description automatically generated

**Logistic Regression using backward elimination algorithm (low):**

Log(Odds) : 4.606 + 1.064room\_shared +2.975 room\_private-2.630 person\_capacity-3.500 host\_is\_superhost +6.500 multi-6.610 biz-7 guest\_satisfaction\_overall-1.555 bedrooms+3.550 dist-2 attr\_index-0 rest\_index+5.680 room\_type\_Entire\_home/apt+1.064 room\_type\_Shared\_room

**Logistic Regression using backward elimination algorithm (medium):**

Log(Odds): 0.745-5 room\_shared +5.260 room\_private +9.200 person\_capacity -1.850 host\_is\_superhost +1.810 multi -6.300 biz +1 guest\_satisfaction\_overall -3.920 bedrooms +9 dist -0 attr\_index +7.190 room\_type\_Entire\_home/apt -5 room\_type\_Shared\_room

**Logistic Regression using backward elimination algorithm (high):**

Log(odds): -3.34 -4.470 room\_shared -2.301 room\_private +1.110 person\_capacity -1.970 host\_is\_superhost -1.870 multi +5.850 biz +2 guest\_satisfaction\_overall +7.340 bedrooms -6.70 dist +0 attr\_index +0-5.920 room\_type\_Entire\_home/apt -4.470 room\_type\_Shared\_room

**Logistic Regression using backward elimination algorithm (very\_high):**

Log(Odds) : -2.011-1.160 room\_shared -1.200 room\_private +6 person\_capacity +7.320 host\_is\_superhost -5.900 multi +1.390 biz +-1.300 guest\_satisfaction\_overall +1.213 bedrooms -3.780 dist +2 attr\_index -0-6.940 room\_type\_Entire\_home/apt -1.160 room\_type\_Shared\_room

A number with numbers on it

Description automatically generated with medium confidence

Here, we can see the first 10 classification records for the validation data set using a backward elimination algorithm.

A screenshot of a computer screen

Description automatically generated

### Decision tree

As a second approach we choose decision tree, as we can solve classification problem using decision tree model.

Here we did the same partitioning for the data set with same random value to have same data trained and validated.

Training Partition values will be (3688,16)

Validation Partition values will be (1581,16)

Subsequently, we conducted a grid search, adjusting parameters such as max\_depth, min\_impurity\_decrease, and min\_samples\_split. This refinement process led to an improved decision tree score of 0.85. Notably, the optimized parameters were determined as follows: max\_depth = 5, min\_impurity\_decrease = 0.001, and min\_samples\_split = 5.

A screenshot of a computer program

Description automatically generated

**A diagram of a house

Description automatically generated with medium confidence**

To improve accuracy of the decision tree we have used random forest and boosted tree approaches. The accuracy of these approaches are compared in next section.

## 9. Interpret the Results

**Accuracy Measures for Logistic Regression:**

Here, we can see the accuracy of training partition and the validation partition:

Training Partition Accuracy: ( 2779+339+18+1)/3681 = 0.8506 = 85.06%

Misclassification rate for Training Partition is: (111+2+1+264+10+22+101+5+13+11)/3681 = 0.1494 = 14.94%

Validation Partition Accuracy: (1186+183+3+0)/1581 = 0.8678 = 86.78%

Misclassification rate for Validation Partition is: (39+2+1+102+5+11+34+4+8+3)/1581 = 0.1322 = 13.22%

**A screenshot of a computer

Description automatically generated**

**Gain and Lift charts for Logistic Regression:**

The lift chart for the ‘low’ price shows the ratio of the proportion of classifications as 0 (‘low’) using the model vs. proportion of the ‘low’ prices taken randomly for different percentiles in the validation partition. For the top 40% of the data most probable to be 0, the logistic model provides 1.3 times higher chance of 1 than the proportion of 1’s taken randomly.

**A graph and chart with numbers

Description automatically generated**

The lift chart for the ‘medium’ price shows the ratio of the proportion of classifications as 1 (‘medium’) using the model vs. proportion of the ‘medium’ prices taken randomly for different percentiles in the validation partition. For the top 10% of the data most probable to be 1, the logistic model provides 3.5 times higher chance of 1 than the proportion of 1’s taken randomly.

**A graph and a diagram

Description automatically generated with medium confidence**

The lift chart for the ‘high’ price shows the ratio of the proportion of classifications as 2 (‘high’) using the model vs. proportion of the ‘high’ prices taken randomly for different percentiles in the validation partition. For the top 10% of the data most probable to be 0, the logistic model provides 6.9 times higher chance of 2 than the proportion of 2’s taken randomly.

**A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated**

The lift chart for the ‘very\_high’ price shows the ratio of the proportion of classifications as 2 (‘very\_high’) using the model vs. proportion of the ‘very\_high’ prices taken randomly for different percentiles in the validation partition. For the top 10% of the data most probable to be 3, the logistic model provides 8.0 times higher chance of 3 than the proportion of 3’s taken randomly.

**A graph of a line and a line

Description automatically generated with medium confidence**

**Accuracy Measures for Logistic Regression using Backward elimination algorithm:**

Here, we can see the accuracy of the training partition and the validation partition:

Training Partition Accuracy: (2788+339+16+1)/3681 = 0.8525 = 85.25%

Misclassification rate for Training Partition is: (111+2+264+10+19+107+5+13+11)/3681 = 0.1459 = 14.59%

Validation Partition Accuracy: (1182+174+2+0)/1581 = 0.8590 = 85.91%

Misclassification rate for Validation Partition is: (44+1+1+112+4+10+36+4+6+5)/1581 = 0.1410 = 14.10%

**A screenshot of a computer

Description automatically generated**

Comparing the accuracy performance measures of logistic regression and logistic regression using the backward elimination algorithm reveals only a slight difference. The validation partition accuracy is 0.8678 for logistic regression and 0.8590 for logistic regression with backward elimination. Despite the marginal variance, logistic regression emerges as the preferred model due to its slightly higher accuracy. Although logistic regression with backward elimination may theoretically offer advantages, its longer runtime due to iterations renders logistic regression the more practical choice. Hence, we consider logistic regression the better model for our purposes.

**Accuracy Measures for Decision Tree:**

Here, we can see the accuracy of training partition and the validation partition:

Training Partition Accuracy: (29+2761+369+0)/3681 = 0.8566 = 85.66%

Misclassification rate for Training Partition is (21+94+140+14+230+16+7+7)/3681 = 0.1434 = 14.34%

Validation Partition Accuracy: (6+1163+183+3)/1581 = 0.8552 = 85.52%

Misclassification rate for Validation Partition is: (5+37+65+10+97+8+4+3)/1581 = 0.1448= 14.48%

**A screenshot of a computer

Description automatically generated**

**Accuracy Measures for Random Forest:**

Here, we can see the accuracy of the training partition and the validation partition:

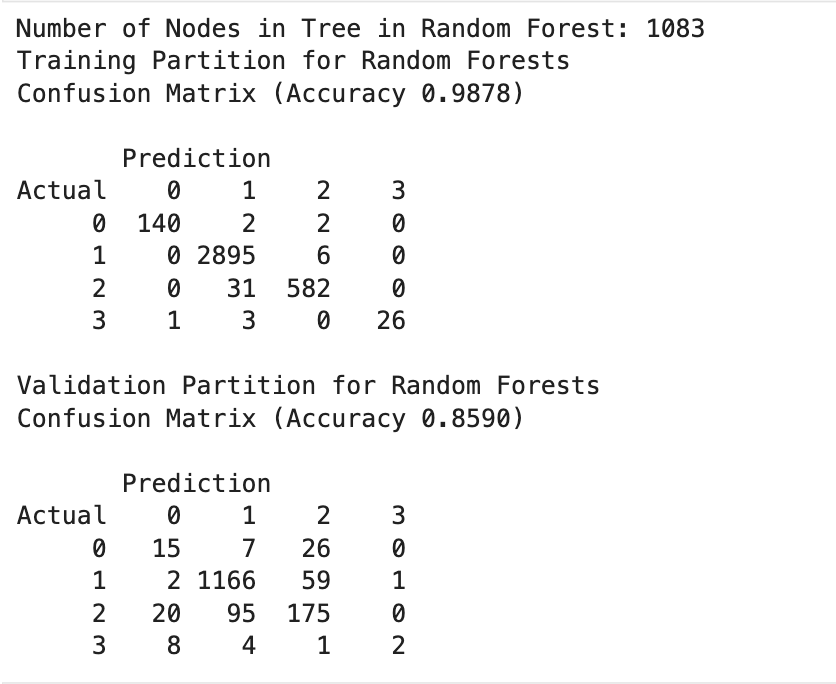
Training Partition Accuracy: (140+2895+582+26)/3681 = 0.9878 = 98.78%

Misclassification rate for Training Partition is (2+2+6+31+1+3)/3681 = 0.0122 = 1.22%

Validation Partition Accuracy: (15+1166+175+2)/1581 = 0.8590 = 85.90%

The misclassification rate for Validation Partition is: (7+26+2+59+20+95+8+4+1)/1581 = 0.141= 14.1%.

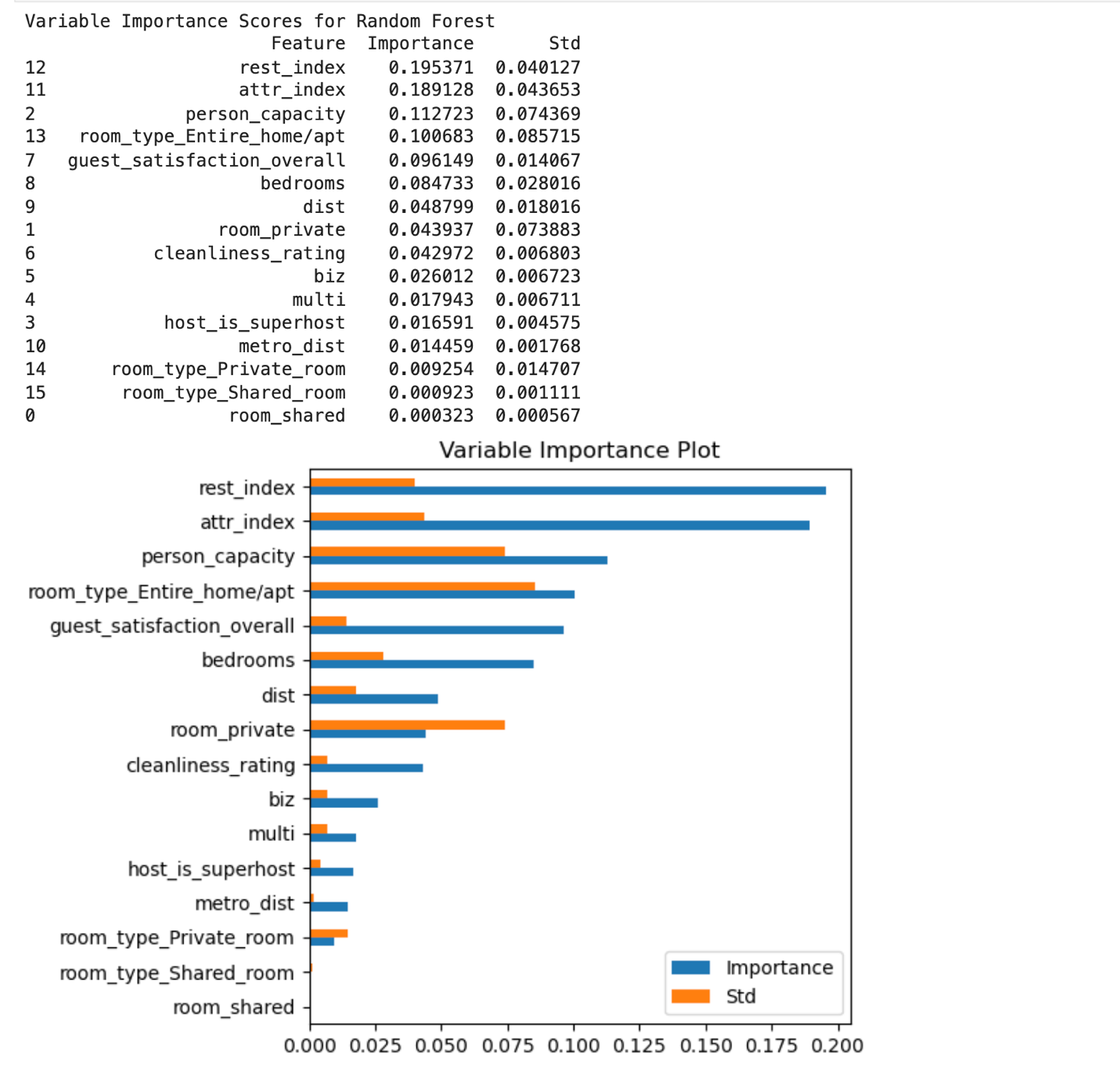
So, here we can see more than 10% difference in the validation and partition data which overfits the data. So, we cannot make any predictions with this model for this data set.

****

Below graph displays the variable importance plot for a decision tree model. It shows the relative importance of different features or variables in predicting the target variable or making decisions within the tree model.

The bars represent the importance score (blue) and standard deviation (orange) for each feature. Features at the top, like "rest\_index" and "attr\_index", have higher importance scores, indicating they are more influential in the model's predictions or decisions. Features towards the bottom, like "room\_shared" and "room\_type\_Shared\_room", have lower importance and contribute less to the model's output.

This type of plot helps identify the most critical features driving the decision tree's performance and can aid in feature selection or interpretation of the model's behavior.

****

**Accuracy Measures for Boosted Tree:**

Here, we can see the accuracy of the training partition and the validation partition

Training Partition Accuracy: (78+2804+419+25)/3681 = 0.9018 = 90.18%

Misclassification rate for Training Partition is (19+47+97+3+191+3+2)/3681 = 0.0982= 9.82%

Validation Partition Accuracy: (10+1171+199+2)/1581 = 0.8672 = 86.72%

The misclassification rate for Validation Partition is: (9+28+1+3+53+1+21+81+6+4+3)/1581 = 0.1328= 13.28%

**A screenshot of a computer

Description automatically generated**

With validation accuracy measures of 0.8552 for the decision tree and 0.8672 for the Boosted tree, the latter exhibits superior performance. Thus, we can conclude that the Boosted tree model is likely to yield better results compared to the decision tree.

## 10. Deploy the best technique

### Logistic Regression

According to the new records, the first record, the price would be low. For the second record, the price would be low. For the third record, the price would be medium.

**A screenshot of a computer program

Description automatically generated**

### Decision Tree

According to new records, the first record would be medium price. The second record would be medium price. The third record would be medium price.

**A screenshot of a computer program

Description automatically generated**

# Conclusion

## Logistic Regression

The logistic regression model achieves an impressive 86% accuracy when using the full set of predictors to predict price outcomes. However, employing the backward elimination algorithm led to an accuracy of 85%, the highest achieved in our analysis after eliminating three predictors. Despite this, logistic regression emerges as the preferred model due to its overall performance and efficiency. The longer runtime associated with the backward elimination algorithm for this dataset further supports this decision. Therefore, the logistic regression model with all variables retained is chosen as the superior option for prediction in current data.

## Classification Tree

In our exploration of classification trees, we experimented with three types: decision trees, random forest trees, and boosted trees. Following grid search, the decision tree yielded a validation partition accuracy of 85%, a result matched by the random forest tree. However, overfitting was observed in the random forest model, rendering it unsuitable for predictions in this dataset. On the other hand, the boosted tree achieved a validation partition accuracy of 86%, making it the preferred model among the classification trees.

## Summary

The logistic regression model and classification trees both performed well in predicting the prices. In general, the logistic model is used to predict the category if there are two classes. If there are more than 2 classes, it is suggested to review other approaches.

Each has their advantages and disadvantages**.** The logistic regression model furnishes a detailed array of probability statistics, encompassing probabilities for each class (p(0), p(1), p(2), p(3)), thereby facilitating a nuanced comprehension of predicted outcomes. Conversely, the classification tree offers a visually appealing depiction of variable importance, enabling rapid identification of key predictors influencing the model's decisions.

The final recommendation is to choose either model based on personal preference for price prediction. However, neither model is advised for predicting over and under, as their accuracy remains almost identical across all scenarios.

# Bibliography

1. Kaggle - Dataset – “Airbnb Prices in European Cities”

https://www.kaggle.com/datasets/thedevastator/airbnb-prices-in-european-cities

1. Class slides and python scripts.