**Airbnb House Price Prediction**

Group 4

Trieu Vo & Alok Shrivastava

College of Professional Studies, Northeastern University

ALY 6020: Predictive Analytics

November 16th, 2021

**Table of Contents**

[**I.** **Project Task Status** 2](#_Toc90824833)

[**II.** **Executive Summary** 3](#_Toc90824834)

[**1.** **Objectives of the project** 3](#_Toc90824835)

[**2.** **Main results** 3](#_Toc90824836)

[**III.** **Introduction** 4](#_Toc90824837)

[**1.** **Problem** 4](#_Toc90824838)

[**2.** **Dataset** 4](#_Toc90824839)

[**3.** **Techniques** 6](#_Toc90824840)

[**IV.** **Linear Regression Model** 6](#_Toc90824841)

[**1.** **Implementation Approach** 6](#_Toc90824842)

[**2.** **Data Analysis and Results** 10](#_Toc90824843)

[**3.** **Discussion** 15](#_Toc90824844)

[**V.** **Random Forest Model** 15](#_Toc90824845)

[**1.** **Implementation Approach** 15](#_Toc90824846)

[**2.** **Data Analysis and Results** 15](#_Toc90824847)

[**3.** **Discussion** 20](#_Toc90824848)

[**VI.** **Conclusion** 20](#_Toc90824849)

[**VII.** **References** 21](#_Toc90824850)

* 1. **Project Task Status**

Task Completed:

* Data Collection
* Data Separation
* Data Import
* Data Cleaning
* Data Manipulation
* Data Visualization
* Data Reduction
* Models Building: Linear Regression & Random Forest
* Models Evaluation
* Presentation
  1. **Executive Summary**

1. **Objectives of the project**

As more people choose Airbnb as their primary accommodation provider, Airbnb has effectively challenged the traditional hotel business. Since its establishment in 2008, Airbnb has experienced tremendous development, with the number of rentals listed on its website increasing at an exponential rate each year.

We use the Inside Airbnb dataset to identify significant trends in customer interest and predict the price of rentals in the US and Europe using Machine Learning models. This problem is similar to a classical use case of machine learning: house price prediction. We can then answer business questions:

1. How to predict the price for each listing?
2. Can we find out how many feature affected the price?
3. **Main results**

The first model is a Linear Regression model. When we test the model with the test set, the models of Trieu and Alok get similar R^2 Scores, around 0.23. / 1.00.

The second model is a Random Forest model. When we test the model with the test set, the model of Trieu gets an R^2 Score of 0.51 / 1.00, and the model of Alok gets an R^2 Score of 0.35 / 1.00.

4 features *longitude, latitude, accommodates and bedrooms* affect the price most.

* 1. **Introduction**

1. **Problem**

As more people choose Airbnb as their primary accommodation provider, Airbnb has effectively challenged the traditional hotel business. Since its establishment in 2008, Airbnb has experienced tremendous development, with the number of rentals listed on its website increasing at an exponential rate each year. We'll utilize the Airbnb dataset to predict the price and of various destinations. We can then provide solutions to business difficulties.

1. **Dataset**

The Inside Airbnb dataset contains approximately 1 million listings from various countries across the world. It also has 84 distinct listing characteristics.

We split the dataset into 2 subsets: the US listings and the Europe listings. Trieu is responsible for the Europe listings, with over 200,000 observations and 27 features. Alok is responsible for the US listings, with over 200,000 observations and 27 features.

There are 27 features in this dataset and their description are as follows:

*accommodates*: The maximum capacity of the listing

*amenities*: The amenities the homestay provides

*availability\_365*: The availability of the listing is 365 days in the future as determined by the calendar.

*bedrooms*: The number of bedrooms.

*beds*: The number of bed(s)

*calculated\_host\_listings\_count*: The number of listings the host has in the current scrape, in the city/region geography.

*host\_has\_profile\_pic*: boolean [t=true; f=false]

*host\_id*: Airbnb's unique identifier for the host/user

*host\_identity\_verified*: boolean [t=true; f=false]

*host\_name*: Name of the host. Usually just the first name(s).

*host\_response\_rate*: The percentage of the host will response

*host\_response\_time*: Waiting time for the host to respond

*id*: Airbnb's unique identifier for the listing

*instant\_bookable*: [t=true; f=false]. Whether the guest can automatically book the listing without the host requiring to accept their booking request. An indicator of a commercial listing.

*last\_review*: The date of the last/newest review

*latitude*: Uses the World Geodetic System (WGS84) projection for latitude and longitude.

*longitude*: Uses the World Geodetic System (WGS84) projection for latitude and longitude.

*maximum\_nights*: the maximum number of nights stay for the listing (calendar rules may be different).

*minimum\_nights*: the minimum number of nights stay for the listing (calendar rules may be different).

*name*: Name of the listing.

*neighborhood*: The neighborhood of the listing

*number\_of\_reviews*: The number of reviews the listing has

*property\_type*: Self-selected property type. Hotels and Bed and Breakfasts are described as such by their hosts in this field

*reviews\_per\_month*: The number of reviews the listing has over the lifetime of the listing

*review\_scores\_rating*: average rating

*room\_type*: [ Entire home/apt | Private room | Shared room | Hotel ]

*price*: the price of the listing (target variable)

1. **Techniques**

We go through different data science steps to preprocess the data, such as Data Collection, Data Cleaning, Data Manipulation, Data Visualization, and Data Reduction.

We build models to predict the price of listings. They are Linear Regression and Random Forest.

* 1. **Linear Regression Model**

1. **Implementation Approach**

**TRIEU VO:**

**Data Collection & Separation:** I split the original data into separate subsets for each project member. I get the listing links from the source of Inside Airbnb using Link Grabber Chrome extension, then I use Regular Expression and Sublime Text to get the US listing links and the Europe listing links, then download and separate them into 2 datasets. Each dataset contains over 200,000 listings. I get the Europe listing dataset and give Alok the US listing dataset.

**Data Import:** I import all necessary libraries during writing the code, such as janitor, dplyr, skimr, tidyr, zoo, tidyverse, dataPreparation, effects, sampling, caret, car, MASS, stringr, fastDummies, randomForest, class, C50, rpart. Then I read the CSV file from the directory and view the data, checking the shape, first 5 observations, and names of columns.

**Data Cleaning:** I find the number of missing values and the percentage of missing values each feature have, then replace those missing values with previous values using a function called na.locf()

**Data Manipulation:** I put all the categorical columns into a data frame and evaluate each column. I evaluate and replace hidden missing values with previous values, then I convert categorical features to numeric features using different techniques:

*amenities*: I counted the number of amenities of each observation and replaced this feature by its count

*host\_has\_profile\_pic, host\_identity\_verified, instant\_bookable*: I found out that these features have missing values as empty space, so I replaced those empty spaces with NA value and then filled these missing values by previous value. Then because these features have 2 values ‘t’, ‘f’ so I converted them into binary values ‘0’, ‘1’.

*host\_response\_rate*: I found out that this feature has missing values as empty space, so I replaced those empty spaces with NA values, and then filled these missing values with the previous value. Then I removed the ‘%’ sign and converted them into real numbers from 0 to 1.

*host\_response\_time*: I found out that this feature has missing values as empty space, so I replaced those empty spaces with NA values, and then filled these missing values with the previous value. It has 4 classes: a few days or more, within a day, within a few hours, within an hour. So I used a function called unclass() to convert them into 1, 2, 3, 4 based on their priorities.

*price*: I removed the ‘%’ sign, filled missing values by previous value, and converted them into real numbers.

*room\_type*: this feature has 4 classes (entire\_home\_apt, hotel\_room, private\_room, shared\_room), so I used a function called dummy\_cols() to convert them into dummy variables. In the end, I had 3 more new columns, and I removed the original feature ‘room\_type.

**Data Reduction:** Among numeric features, I drop unused features such as *host\_has\_profile\_pic* (all observations have this feature value = 1), *host\_id, id, host\_name, name…*

**ALOK SHRIVASTAVA:**

**Data Import:** In this Trieu provided me with AIRBNB listing data of the USA, where the number of rows is 210617 and number of columns are 27, out of which some of the variables like hostname, host ID, customer ID, etc. was not useful so I removed those columns with c() function in r studio. After data import, I have installed a lot of packages for the proper functioning of the function in the RMD code. Some of the important packages are gsubfun, janitor, skimr, dataPreparation, sampling, MASS, fastDummies, c50, rpart, etc.

**Data Cleaning:** This procedure involved several steps which were carried out during the cleaning of the dataset so that it will look beautiful and can be used for machine learning and predictive analytics. The steps are as follows:

* Cleaning column names: I used this function to clean the names of the variables and make it useful for the skim() function so that we can identify the missing values in the dataset.
* Finding missing values: I used this function to find the number of missing values present, the blank rows in the dataset, summary of the dataset in terms of the mean for numeric variables.
* Conversion of blank spaces with N/A: This step involves the conversion of all the blank spaces with N/A in the dataset. For this I used “listing\_final [listing\_final == " "] <-NA” to convert all the blank spaces to NA.
* Converting amenities from character to numeric datatype: This step was to convert the amenities to numbers that were mentioned in characters and were separated by commas. For this, I used the strsplit() function to remove the commas and convert the amenities into numbers.
* Text, letter

  Description automatically generated
* Removal of special character: The next step involves the removal of the special character “$” and converting the price variable from character to numeric variable. For this I used gsub() function and as.numeric() function.
* Replacement of N/A with previous values: This step involves the removal of N/A from all the variables which contain N/A and replacing them with the previous values. For this I used na.locf() function. The columns involved in this process were: bedrooms, beds, review\_scores\_rating, reviews\_per\_month, host\_identity\_verified, host\_has\_profile\_pic, and price.
* Replacement of characters with binary integers: In this step, I removed “t” and “f” from variables and converted them to binary values i.e., 0 and 1. In this, I used the as.integer() function to convert them to integers. This step also involved the conversion of these variables into factors using as.factor() function. The variables involved in this process are host\_identity\_verified, host\_has\_profile\_pic, and instant\_bookable.
* Creating dummy variables: Conversion of room\_type to dummy variables. In this step I converted the columns to factors first and then used dummy\_cols() function to create 4 new variables for “room\_type” column. The new four variables are: room\_type\_hotel\_room, room\_type\_private\_room, room\_type\_entire\_home\_apt and room\_type\_shared\_room.

1. **Data Analysis and Results**

**TRIEU VO:**

**Create a training set and a test set**: I use a function called sample() to create a training set and test set with a ratio of 7:3.

**Build a Linear Regression model:** I build a Negative Binomial GLM Model with the function called glm.nb() with the target variable being price and the other 19 predictor variables.

Call: glm.nb(formula = price ~ accommodates + amenities + availability\_365 +

bedrooms + beds + calculated\_host\_listings\_count + host\_identity\_verified +

host\_response\_rate + host\_response\_time + instant\_bookable +

latitude + longitude + maximum\_nights + minimum\_nights +

number\_of\_reviews + review\_scores\_rating + reviews\_per\_month +

room\_type\_entire\_home\_apt + room\_type\_private\_room, data = training\_set,

init.theta = 3.262220605, link = log)

Coefficients:

(Intercept) accommodates

1.694e+00 9.655e-02

amenities availability\_365

1.239e-03 4.754e-04

bedrooms beds

1.503e-01 -8.363e-03

calculated\_host\_listings\_count host\_identity\_verified

3.230e-03 2.630e-02

host\_response\_rate host\_response\_time

-8.169e-02 -6.973e-03

instant\_bookable latitude

7.534e-02 3.671e-02

longitude maximum\_nights

1.861e-03 4.308e-05

minimum\_nights number\_of\_reviews

-4.213e-04 -2.298e-03

review\_scores\_rating reviews\_per\_month

9.295e-02 1.513e-02

room\_type\_entire\_home\_apt room\_type\_private\_room

2.085e-01 NA

Degrees of Freedom: 252026 Total (i.e. Null); 252008 Residual

Null Deviance: 371000

Residual Deviance: 262600 AIC: 2674000

**Predict y\_train\_predict:** I use the model to predict the training set. I calculate measures such as MAE, MSE, RMSE using the caret library. The results are *MSE: 5787.244, MAE: 47.18271, RMSE: 76.07394*. I also created an RSQUARE function to calculate R^2 Score. R-Squared is a measure in regression problems. It is limited to the range from 0 to 1. If the value of R-Squared is 1, then the model fits the data perfectly. So it's similar to the measure of accuracy in classification problems. On the training set, *R^2 is 0.22*, which is pretty low. So the model didn't fit the training data very well. I also plot the first 150 predictions for the training set, so that I can get a good understanding of the model.

Graphical user interface, chart, histogram

Description automatically generated

**Predict y\_test\_predict:** I use the model to predict the test set. I calculate MAE, MSE, RMSE, R^2. Those results are *MSE: 5682.418, MAE: 47.06479, RMSE: 75.38182, R^2: 0.2292016.* On the test set, R^2 is 0.23, which is pretty low. I also plot 150 predictions for the test set.

Graphical user interface, chart

Description automatically generated

**Evaluate the result:** In conclusion, the Linear Regression model is not a good model to predict the price of listings in this dataset, because it gives a low R^2 score (0.22) on the test set.

**ALOK SHRIVASTAVA:**

**Creating Test and Train Dataset:** In this step, I used set.seed() function to create a random order and separate the dataset in test and train dataset using sample() and nrow() function. The dim() is used to observe the dimensions of the dataset. I separated the dataset as 70: 30 which came out to be 157962 rows and 20 columns for the training dataset and 52655 rows and 20 columns for the test dataset.

Graphical user interface, text, application

Description automatically generated

**Model Building:** General Linear Model (GLM). I tried to create a general linear model using some of the features from the USA listing dataset. The features are amenities, beds, bedrooms, accommodates, room\_type\_private\_room, reviews\_per\_month, and review\_scores\_rating. For creating the GLM model I downloaded the MASS package and installed the MASS library. I used this model because I wanted to understand the relation between the features and the target variable. For model building, I used the glm.nb() function to create a negative binomial model for the. For the model, I used factors that I thought are more related, so I used the above 7 variables. I used the summary() function to get the summary of the output from the negative binomial linear model. plot() function is used to plot the graph of a linear model.

For predicting the values, I used the predict() function to predict train values and to get the output for MSE, MAE, RMSE, and R^2 values. The observer values for the predictors are as follows:

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The R^2 value is obtained from the RSQUARE() and cat() function and the observed R^2 value is 0.25422. To find the accuracy in percentage we can multiply the R^2 value by 100 and will get the result. After multiplying we got 25.42% accuracy for the model. From this, we can say that the variables like amenities, beds, bedrooms, accommodates, room\_type\_private\_room, reviews\_per\_month, and review\_scores\_rating have higher relation to the price factor rather than using all the variables available. However, the accuracy is not satisfactory, and this model cannot be used for the prediction.

1. **Discussion**

Trieu and Alok's two Linear Regression models have similar R^2 Scores, around 0.23. This score is quite low. We conclude that this is not a good model for predicting listing prices in this dataset.

* 1. **Random Forest Model**

1. **Implementation Approach**

We take the same preprocessing steps as the Implementation Approach of the first model.

1. **Data Analysis and Results**

**TRIEU VO:**

**Create a training set and a test set**: I use a function called sample() to create a training set and test set with a ratio of 7:3.

**Build a Random Forest model:** I build a Random Forest Model with a library called randomForest and a function called randomForest. The target variable is price and there are 19 predictor variables. At first, I try to generate 500 trees in the Random Forest. But my laptop runs out of memory and gets a blue screen error. Even though my laptop has 16GB of RAM it still can't make many trees. So I try 100, 50, 10, 5 trees in the Random Forest to avoid memory overflow. And it takes me 30 minutes to generate 5 trees to train the model.

Call:

randomForest(formula = price ~ accommodates + amenities + availability\_365 + bedrooms + beds + calculated\_host\_listings\_count + host\_identity\_verified + host\_response\_rate + host\_response\_time + instant\_bookable + latitude + longitude + maximum\_nights + minimum\_nights + number\_of\_reviews + review\_scores\_rating + reviews\_per\_month + room\_type\_entire\_home\_apt + room\_type\_private\_room, data = training\_set, ntree = 5, importance = TRUE, na.action = na.omit)

Type of random forest: regression

Number of trees: 5

No. of variables tried at each split: 6

Mean of squared residuals: 5140.131

% Var explained: 30.51

This is the error plot of the Random Forest Model. We can see that the error decreases during training.

Chart, scatter chart

Description automatically generated

**Predict y\_train\_predict:** I use the model to predict the training set. I calculate measures such as MAE, MSE, RMSE using the caret library. The results are *MSE: 1058.071, MAE: 18.48099, RMSE: 32.528*. On the training set, *R^2 is 0.8619223*, which is high. So the model fits the training data very well. I also plot the first 150 predictions for the training set, so that I can get a good understanding of the model.

Graphical user interface, chart

Description automatically generated

**Predict y\_test\_predict:** I use the model to predict the test set. I calculate MAE, MSE, RMSE, R^2. Those results are *MSE: 3800.05, MAE: 38.25589, RMSE: 61.64455, R^2: 0.5026505.* On the test set, R^2 is 0.51, which is acceptable. I also plot 150 predictions for the test set.

Chart

Description automatically generated

I also generate the feature importance of the Random Forest model. From this graph, we can see that longitude, latitude, accomodates, and bedrooms are the most important features that will affect the price the most. From this, we can advise homeowners on these features to help them improve their listings.

Chart, bar chart

Description automatically generated

**Evaluate the result:** In conclusion, the Random Forest model is a good model to predict the price of listings in this dataset, because it gives an acceptable R^2 score (0.51) on the test set. This model also points out certain features which are important to predict the house price.

**ALOK SHRIVASTAVA**

**Model Building:** I tried to create a random forest model using the same features from the USA listing dataset as I used in the GLM model. The features are amenities, beds, bedrooms, accommodates, room\_type\_private\_room, reviews\_per\_month, and review\_scores\_rating. For creating the Random Forest model, I downloaded C50, rpart, and randomForest package and installed all required libraries. I used this model because I wanted to understand how accurate a random forest model could be for predicting the price. For model building, I used the randomForest() function to create the model. I used the summary() function to get the summary of the output from the model. plot() function is used to plot the graph of a linear model.

For predicting the values, I used the predict() function to predict train values and to get the output for MSE, MAE, RMSE, and R^2 values. The observer values for the predictors are as follows:

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

The R^2 value is obtained from the RSQUARE() and cat() function and the observed R^2 value is .3464. To find the accuracy in percentage we can multiply the R^2 value by 100 and will get the result. After multiplying we got 34.64% accuracy for the model, which is greater than the accuracy observed from the above model i.e., 24.51%.

1. **Discussion**

Trieu's Random Forest model gives acceptable results on the test set (R^2 = 0.51), while Alok's Random Forest model gives lower results on the test set (R^2 = 0.35). This may come from differences in how we preprocess the dataset and how we choose features to train those models. When we compare these results to the results of Linear Regression, they are completely higher. From this, we can see that the Random Forest model predicts house prices more accurately. Additionally, the Random Forest model can provide a graph of feature importance to help us understand which features influence prices the most. It has many advantages over linear regression models.

* 1. **Conclusion**

In this project, we use the Airbnb dataset with 200,000 listings each member and 27 features to explore.

We go through different data science steps to preprocess the data, such as Data Collection, Data Cleaning, Data Manipulation, Data Visualization, and Data Reduction.

We build 2 models to predict the price of listings. They are Linear Regression and Random Forest. The Linear Regression models give us similar results of the R^2 Scores (0.24). The Random Forest models give the highest R^2 score on the test set (Trieu: 0.51 vs. Alok: 0.35). The values of MAE and RMSE are also the smallest for the random forest. Thus we conclude that the random forest model is the best fit for the price prediction of Airbnb properties. 4 features longitude, latitude, accommodates and bedrooms affect the price most.

* 1. **References**

Airbnb, I. (2021, September 30). *Get the data - inside Airbnb. adding data to the debate.* Get the Data. Retrieved November 12, 2021, from <http://insideairbnb.com/get-the-data.html>.

Bento, C. (2020, August 1). Create and customize boxplots with Python’s Matplotlib to get lots of insights from your data. Medium. <https://towardsdatascience.com/create-and-customize-boxplots-with-pythons-matplotlib-to-get-lots-of-insights-from-your-data-d561c9883643>

Decision tree for regression in R programming. (2020, July 28). GeeksforGeeks. <https://www.geeksforgeeks.org/decision-tree-for-regression-in-r-programming/>

Finnstats. (2021, April 13). Random forest in R. R-bloggers. <https://www.r-bloggers.com/2021/04/random-forest-in-r/>

How to find coefficient of determination (R-squared) in R. (2020, November 13). Statology. <https://www.statology.org/r-squared-in-r/>

KNN algorithm | KNN in R | KNN algorithm example. (2020, June 26). Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/>

Machine learning Cho du Lieu Dang bang. (n.d.). Machine Learning cơ bản. <https://machinelearningcoban.com/tabml_book/ch_model/decision_tree.html>

Machine learning Cho du Lieu Dang bang. (n.d.). Machine Learning cơ bản. <https://machinelearningcoban.com/tabml_book/ch_model/random_forest.html>

Płoński, P. (2020, June 30). How many trees in the random forest? MLJAR. <https://mljar.com/blog/how-many-trees-in-random-forest/>

R random forest tutorial with example. (2021, October 7). Guru99. <https://www.guru99.com/r-random-forest-tutorial.html>

R squared in R - How to calculate R2 in R? - JournalDev. (2021, January 5). JournalDev. <https://www.journaldev.com/47888/r-squared-in-r-programming>

Random forest approach for regression in R programming. (2020, July 10). GeeksforGeeks. [https://www.geeksforgeeks.org/random-forest-approach-for-regression-in-r-programming/#](https://www.geeksforgeeks.org/random-forest-approach-for-regression-in-r-programming/)

Random forest approach for regression in R programming. (2020, July 10). GeeksforGeeks. <https://www.geeksforgeeks.org/random-forest-approach-for-regression-in-r-programming/?ref=gcse>

Random forests · UC business analytics R programming guide. (n.d.). UC Business Analytics R Programming Guide ·. <https://uc-r.github.io/random_forests>

Sirohi, K. (2018, December 30). K-nearest neighbors algorithm with examples in R (Simply explained knn). Medium. <https://towardsdatascience.com/k-nearest-neighbors-algorithm-with-examples-in-r-simply-explained-knn-1f2c88da405c>

Sumathipala, S. (2021, August 12). How to determine the number of trees to be generated in random forest algorithm? ResearchGate. <https://www.researchgate.net/post/How_to_determine_the_number_of_trees_to_be_generated_in_Random_Forest_algorithm>

Warning message: Glm.fit: algorithm did not converge. (2019, July 2). Data Science, Analytics and Big Data discussions. <https://discuss.analyticsvidhya.com/t/warning-message-glm-fit-algorithm-did-not-converge/5299/3>