

Booking Price Prediction

Table of Content

Introduction

1. Background

Method

1. Data Source
2. Model Used
3. Visualization Used

Result

Discussion

1. Model Selection
2. Model Interpretation
3. Model Check

Conclusion

1. Implication
2. Limitation
3. Future Direction

**Introduction**

1. Background

There are plenty of benefits from Airbnb Sharing Economy, as the environmental benefits. It has become a greener way to travel. While there are several advantages to live in Airbnb, it comes with several difficulties for the operators and Property Owners. Online Airbnb websites have hundreds of rental housing available. Determining the right price for one accommodation becomes difficult because of too much house pricing on the list. To solve the supply and demand problem for housing, they need to know an optimal pricing strategy. The pricing Recommender system could help hosts by suggesting a probable list of suggested room price from which they can select the optimized one. The pricing recommender system could make operators aware of similar rental housings which are available to provide for customers. For this report, I will present a detailed and systematical analysis of building a price recommender system.

**Method**

1. Data Source

This “listings” dataset consists of 45053 observations with 17 variables. Each row corresponds to a customer booking history record. To bring the data into a consistent format, several steps of data cleaning are taken, including dropping unnecessary columns, checking for

A screenshot of a cell phone

Description automatically generated

Table 2.1

checking for invalid data. Unnecessary columns including “scrape\_id”,” review\_scores\_location”,” review\_score\_communication” which are irrelevant to the main content of data analysis and those variables are removed from the “listing” dataset. Those time variables, including “first review”, “last review”, “calendar\_last\_scraped” are invalid data since structure, are read in date number only. We see that in Table 1.1 there are too many missing values on variable “square\_feet”, “monthly\_price” and “weekly\_price”. The rows with any of these missing values will, therefore, be removed. Finally, check columns names and make sure the name of each variable makes sense

A screenshot of a cell phone

Description automatically generated

Table 2.2

Some essential variables that datasets provide are (after dropping duplicate records and deal with missing values and outliers):

* Bedrooms: Positive integer documenting the number of bedrooms in each booking record
* Neighborhood: the neighborhood of the listing
* Accommodates: the number of guests that the rental can accommodate

Another dataset named “reviews\_final” contain customer reviews on Airbnb booking website.

1. Model Used

Models in this report include multiple regressions, logistic regression and multilevel regression. It is not likely that all observations coming from the same neighborhood group are similarly independent, and sometimes it might have skewed residuals. Multilevel Regressions are analyzed by different groups. Response variables measuring from each rental record can reasonably be assumed independent. Particular customers in specific neighborhood group tend to have a relatively high possibility of renting higher price of housing, so that know the average group price of renting makes it more likely that range of prediction price would somewhere near the average price in that particular group.

1. Visualization Used
2. univariate summary

Uni-variate exploration includes bar plot, pie chart for price and covariate (security deposit, guests\_included, cleaning fees. etc.) respectively

1. Bivariate summaries
2. Bivariate exploration includes an examination of numerical and graphical summaries of the relationship between price and covariate (security deposit, guests\_included, cleaning\_fees. etc.). Graphs include boxplot and scatter plot.

**Result**

For Exploratory Data Analysis, it includes several aspects, such as reviews text analysis, neighborhood analysis, group analysis, room type analysis, price analysis, mapping of restaurants, and property type analysis.

1. Reviews Text analysis

Customer reviews could offer tremendous insights into what customers like and dislike about the rental experience. Existing reviews always heavily influence the booking decision of new customers. Based on the variable concerning the customer reviews, the next step is to use the word cloud graph to identify the most frequently used words in the customer review.

While comparing words with different colors, we observe that top three most frequently used words in the customer review in table 2.1 are “stay”, “location”, “clean”, which gives us a reason to believe that most Airbnb houses’ comfortableness is essential. From these results, we can infer that location and cleanness might be crucial for customers to consider.

A screenshot of a cell phone

Description automatically generated

Table 3.1

1. Neighborhood Analysis

Now, looking at the graphs 2.2 below, we can see that the most significant number of booking is in the Hollywood area, which contains 697. Neighborhood analysis helps to understand the booking popularity in each neighborhood. Based on that, I would say that the larger the number of booking in that area is, the higher the demand for renting.

A screenshot of a cell phone

Description automatically generated

Table 3.2

3.Group Analysis

A screenshot of a cell phone

Description automatically generated

Table 3.3

Each color indicates the distribution of one variable. There are four variables in this analysis, including bedrooms, beds, accommodates and bathrooms. To identify the total characteristics of booking, I count the number of observations of each group and plot bar graphs. An unusual situation can notice with the number of bedrooms and beds on the chart above, where we see that some of the housing does not have bedrooms or beds. It can be inferred that the demand for booking on Airbnb is not always belonged to the usual style of living. There might be tents in the living rooms etc. From the results, we observe that the majority number of beds, bedrooms and accommodates center at one and two. With the finding on the graph above, it would be promising for the host to try housing rental with a unique theme, which is also expected.

4. Room Type Analysis

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A picture containing text

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

Table 3.4

From the boxplot above, entire home/apartment tend to have a more significant number of accommodates. The boxplot with cleaning fee shows higher cleaning fee is in a broader range. For the number of the security deposit, the shared room has more significant standard deviations from the center as compared to another room type. When looking at the boxplots on the number of beds and bedrooms, both private rooms have a short average compared to entire home/apt, and this may be because there are small varieties in private rooms.

A screenshot of a cell phone

Description automatically generated

Table 3.5

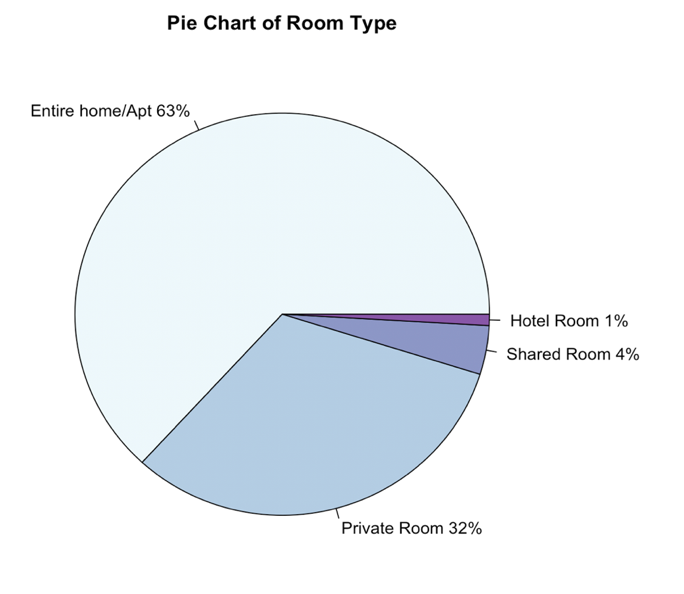


Table 3.6

I then visualized the number of booking record in each room type on the pie chart. Dark purple corresponds to the hotel room, which accounts for 1%, and light purple compares to a shared room with about four percent. The highest proportion of room type in the dataset for all bookings is about 63 percent, which belongs to the entire home/apt.

A screenshot of a cell phone

Description automatically generated

Table 3.7

The number of booking records based on room type is in the range between zero and five hundred as expected shows that most private room has priced in the range between zero and three hundred, with significant peaks in around seventy-five. Similarly, for the entire home/apt, there are lower price booking than the price which over three hundred. We can see a peak of around one hundred.

1. Price Analysis

The following plot shows the distribution of price lower than five hundred among all observations. As we know, the majority of housing would have a similar amount if they have the same location and size; this can also be inferred from the graph. The majority of the observations in the data frame are lower than two hundred and fifty.

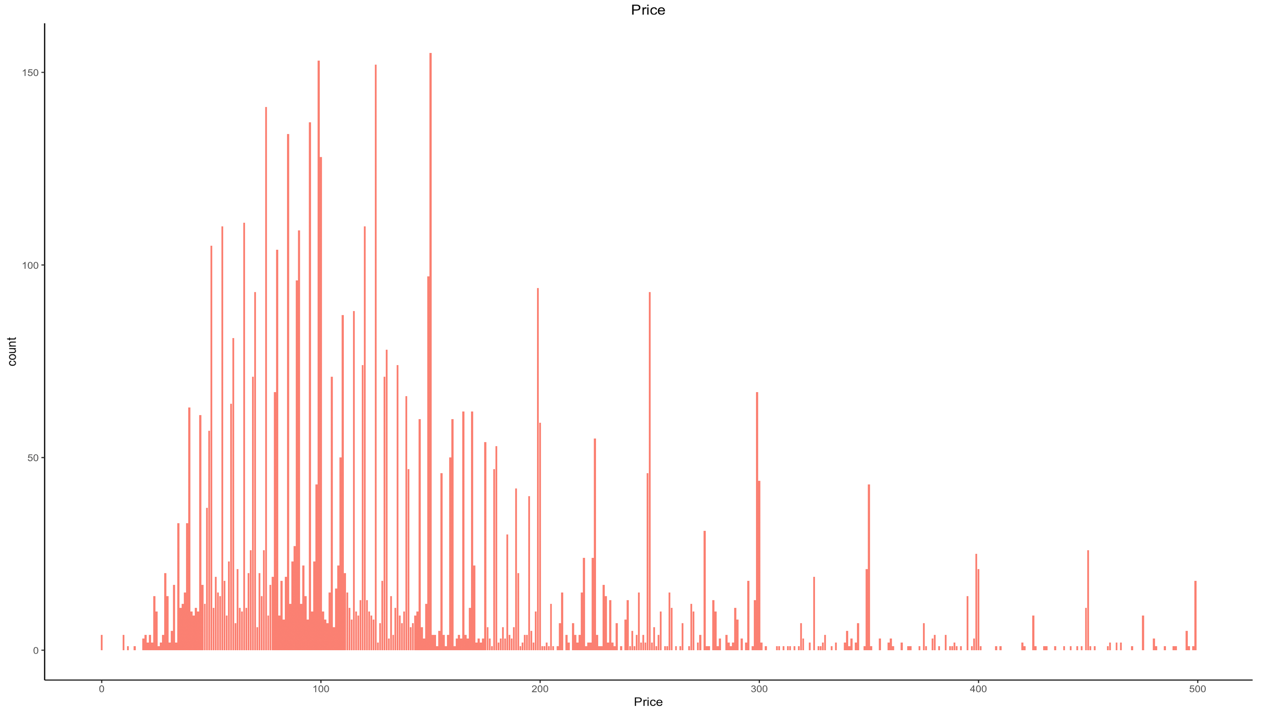


Table 3.8

6. Mapping of Rental Housing

The map below shows the location of Airbnb housing in Los Angela. There is a large number of certain house areas. This is likely to be correlated with surroundings. Since Hollywood, Mid-Wilshire and Venice are the most developed and popular area compared to other neighborhoods. Higher demand leads to a large amount of housing for rental.

A close up of a map

Description automatically generated

Table 3.9

7. Property Type Analysis

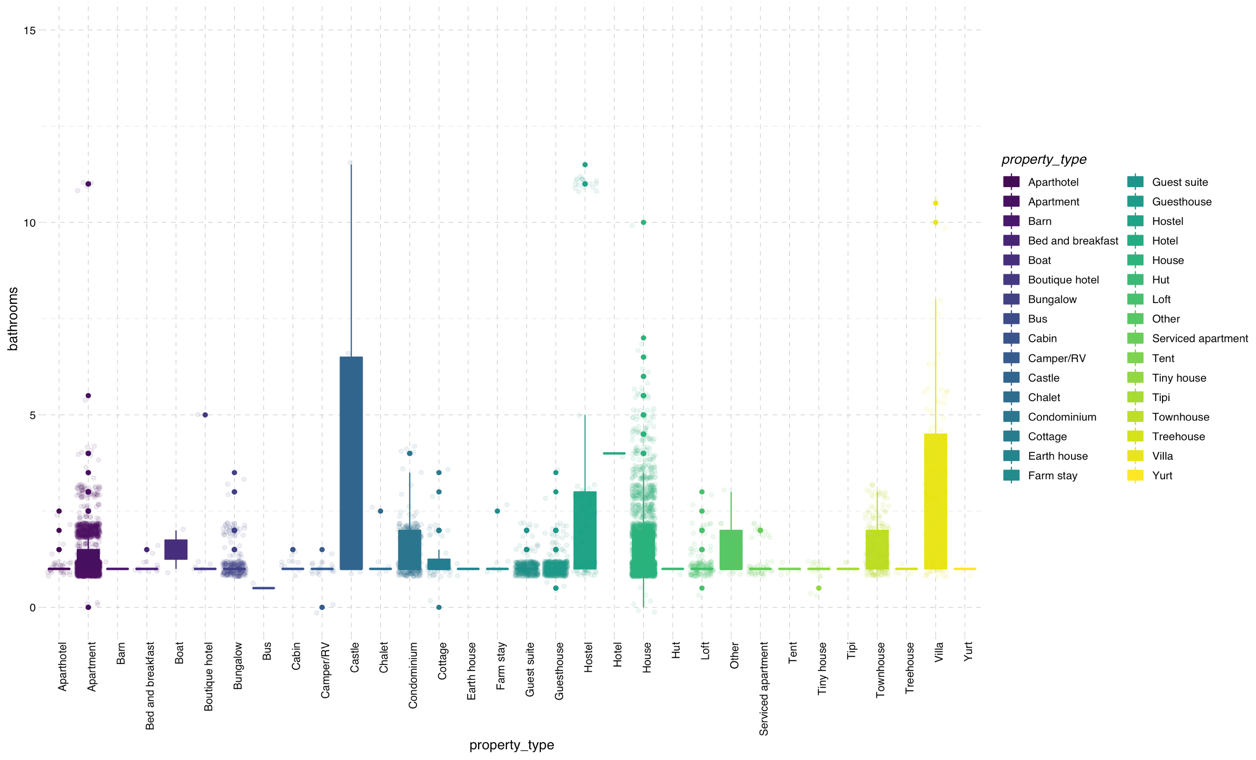


Table 3.10

Instead of categorizing booking records based on room type, I would like to investigate the number of bathrooms grouped by property type. As you can see, the number of bathrooms in the castle has a more significant standard deviation from the mean.

**Discussion**

The task of the machine learning algorithms is to build models that for a given rental booking what price it is.

1. Multiple Regressions

I examine the factors predicting the Airbnb rental price at Los Angela’s. Table below are variables that might be considered into price prediction.

A screenshot of a cell phone

Description automatically generated

Table 4.1

1. Model Selection

A picture containing text, newspaper

Description automatically generated

Table 4.2

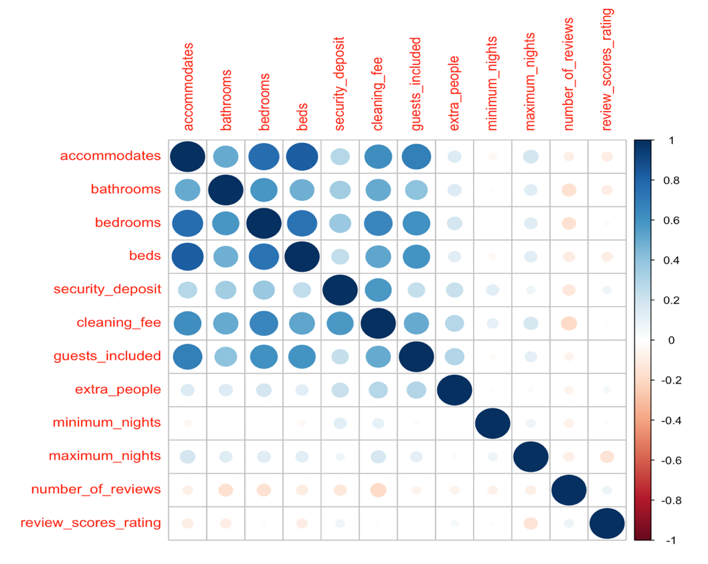


Table 4.3

A close up of a white wall

Description automatically generatedAfter making correlation graphs among variables, I had about half of variables that have large correlations. Accommodates and beds/bedrooms are correlated positively based on the graph above. Additionally, we can notice that accommodates are not strongly correlated with some other variables, such as “number of reviews”, “review scores rating” etc., Although there is no strong positive relationship among variables that I’ve mentioned above, they are not negatively correlated to each other.

Table 4.4

After making correlation graphs among variables, I had about half of the variables that have significant correlations. Accommodates and beds/bedrooms are correlated positively based on the diagram above. Additionally, we can notice that accommodates are not strongly correlated with some other variables, such as “number of reviews”, “review scores rating” etc.. Although there is no strong positive relationship among variables that I’ve mentioned above, they are not negatively correlated to each other.

1. Interpretation

The first attempt uses the following model to predict the prices:

A screenshot of text

Description automatically generated

Table 4.5

Coefficient of accommodates means if extra\_people value increases by 1 unit, we’d expect our price variable to increase by 0.76 unit while keeping all other coefficients as constant; coefficient of maximun\_nights means if maximun\_nights value increases by 1 unit, we’d expect our price variable to increase by 0.0294 unit while keeping all other coefficients as constant; coefficient of number\_of\_reviews means if number\_of\_reviews value increases by 1 unit, we’d expect our price variable to decrease by 0.026 unit while keeping all other coefficients as constant; coefficient of review\_scores\_rating means if review\_scores\_rating increases by 1 unit, we’d expect our price variable to increase by 1.75 unit while keeping all other coefficients as constant; coefficient of guests included means if guests included value increases by 1 unit, we’d expect our price variable to increase by 44.4unit while keeping all other coefficients as constant; intercept means if all six variables equal to 0, then the expected value of earn would be -108.8.

The adjusted R2 and AIC achieved by all multiple regression models are calculated, respectively. Comparing the results from each model, I finally choose the model with the lowest AIC or lowest adjusted R2 with no multicollinearity. Since multiple regression model is not the optimal algorithm for price prediction, the model result above is only one of the model (not the best one) that I’ve tried for all the algorithms.

A screenshot of a cell phone

Description automatically generated

Table 4.6

1. Model Checking

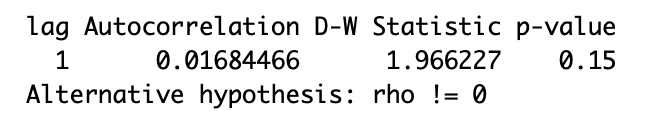


Table 4.8

To make sure that the model above can be used, I did a model check to verify that models didn’t violate any regression assumptions. Based on the result from Table 4.8, it is evident that DW statistics is closed to 2 and p-value is larger than 0.05. It is concluded that the assumption of independence of error is not able to be rejected.

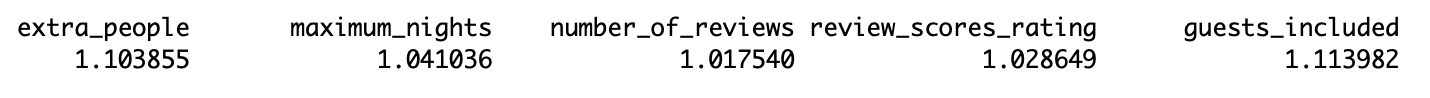


Table 4.9

Table 4.9 is used to check multicollinearity among each predictor. To make sure that variables that are used in the model are strongly correlated, I made a correlation matrix with all predictor variables and calculated variance inflation factor (VIF). Since all the number from each variable is not larger than 10, there is no multicollinearity.

A picture containing screenshot

Description automatically generated

Table 4.10

Based on Table 4.10 above, it is the plot between fitted values and residuals. Residual dot is randomly placed around the horizontal zero. Also, the linearity assumption is satisfied since the red line is straight and horizontal.

A screenshot of a social media post

Description automatically generated

Table 4.11

Since normality assumption is based on the residuals, QQ plot would appropriate to analyze. On table 4.11, aside from three data points that have large residuals, most of the observations lie along the 45-degree line.

A close up of a map

Description automatically generated

Table 4.12

Based on the scale-location plot above, the red line is almost flat and does not have a too apparent positive slope. And the data points are randomly spread out. After removing the observations 5519, 4753 and 3254, there would be more randomly spread on residuals.

A close up of a map

Description automatically generated

Table 4.13

It can be noticed from the above graphs that there is red Cook’s distance curved line contributing to influential data points. The situation presented above means that multiple regression might not be suitable enough for price prediction. Further changes on the multiple regression include removing outliers, make transformations on the variable. Perhaps I can achieve a higher result by making a transformation on particular variables.

1. Logistic Regressions
2. Model Selection

To make the problem more tractable, I start by grouping prices into a few categories: price large than 400 and price small than 400. I spilt the dataset into a training set and a test set. To rate the success of the model, I calculate the accuracy score.

1. Interpretation

A screenshot of a cell phone

Description automatically generated

A close up of text on a white background

Description automatically generated

The logistic regression model and interpretation for this project is as below:

logit(P)= -7.68+0.12\*accommodates*+0.46\*bathrooms*+0.5\*bedrooms*-0.07\*beds+0.01\*cleaning fee+0.0004\*security deposit*

* Intercept: With all other variables equal to 0 would have log odds of -7.67 to have price over 400.
* Accommodates: With the same level of all the rest variables, when accommodates level increases by 1, then the expected value of the price’s log odds would increase by 0.12 unit.
* Bathrooms: With the same level of all the rest variables, when bathrooms level increase by 1, the expected value of the price’s log odds would increase by 0.46 unit.
* Bedrooms: With the same level of all the rest variables, when bedrooms level increases by 1, then the expected value of the price’s log odds would increase by 0.5 unit.
* Beds: With the same level of all the rest variables, when beds level increases by 1, then the expected value of the price’s log odds would decrease by 0.08 unit.
* Cleaning fee: With the same level of all the rest variables, when cleaning fee level increases by 1, then the expected value of the price’s log odds would increase by 0.5 unit.
* Security deposit: With the same level of all the rest variables, when security deposit level increases by 1, then the expected value of the voter’s log odds of support for Bush would increase by 0.0004 unit.

1. Model Checking

As you can see, this logistic regression model does better than multiple regression. The model fits well based on marginal model plots.

A close up of a map

Description automatically generated

Table 4.14

Below is a plot of the binned residual between expected values and average residual. It shows that most of the points fall into the confidence bands.

A close up of a map

Description automatically generated

Table 4.15

A screenshot of a map

Description automatically generated

Table 4.16

I use the training set to feed into the logistic regression model. By graphing the ROC (Table 4.16) and getting confusion matrix based on the result of the logistic regression model, I obtain the accuracy of about 94.89. It allows us to have a better understanding of how the price would be predicted by setting the response variable into a binary outcome and running the logistic regression model.

1. Multi-level Regressions
2. Model Selection

A random forest is an algorithm that works to find the essential variables in building the multilevel regression. Referring to the table, we can see that variables “cleaning fee”, “guests included”, “security deposit”, “cancel policy”, “instant bookable” and “extra people” have a more significant decrease in MSE compared to other variables.

A screenshot of a cell phone

Description automatically generated

Table 4.17

1. Interpretation
2. Random intercept

A screenshot of text

Description automatically generated

Table 4.18

A screenshot of a cell phone

Description automatically generated

Table 4.19

Among fixed effects:

Coefficient of security\_deposit means if security\_deposit value increases by 1 unit, we’d expect our price variable to increase by 1.28e^-5 unit while keeping all other coefficients as constant; coefficient of cleaning\_fee means if cleaning\_fee value increases by 1 unit, we’d expect our price variable to increase by 7.97e^-4 unit while keeping all other coefficients as constant; coefficient of guests\_included means if guests\_included value increases by 1 unit, we’d expect our price variable to decrease by 1.65e-2 unit while keeping all other coefficients as constant; coefficient of instant\_bookablet means if instant\_bookablet increases by 1 unit, we’d expect our price variable to increase by 6.266e^-3 unit while keeping all other coefficients as constant

Among random effects:

The model with random intercept effects for the first neighborhood (Alhambra) is 1.458 as table shows above.

1. Random slope

A screenshot of a cell phone

Description automatically generated

Table 4.20

Among fixed effect:

Coefficient of security\_deposit for Alhambra means if security\_deposit value increases by 1 unit, we’d expect our price variable to increase by 1.36e^-5 unit while keeping all other coefficients as constant; coefficient of guests\_included means if guests\_included value increases by 1 unit, we’d expect our price variable to increase by 1.65e-2 unit while keeping all other coefficients as constant; intercept means if all variables equals to 0,then the expected value of earn would be 1.499

Among random effects:

The model with random slope effects on cleaning\_fee for the first neighborhood (Alhambra) is 4.8e^-4 as table shows above.

A screenshot of a cell phone

Description automatically generated

Table 4.21

By using the cross-validation method on multi-level models and compare the result of MSE on three models (random intercept, random slope, random intercept and slope), we found that the first model has the smallest result on MSE. It is concluded that the model with the random intercept grouped by neighborhood would be the optimized model among all other multi-level regression models.

1. Model Checking

The QQ plot of residuals/parameters and residual plots from Table 4.18-4.20 give us generalized information about normality and independence of residuals. Errors in both multilevel models are not related to each other. The computation of residuals relies on the assumption of independence. Similarly, there are equal variance if criterion at different levels of predictor, which make the parameter estimates optimal.

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generatedTable 4.22

Table 4.23

A screenshot of a cell phone

Description automatically generated

Table 4.24

**Conclusion**

1. Implication

With the end goal to make booking price prediction, this comprehensive exploratory data analysis and statistical modelling on their open-source dataset helped us to understand the underlying patterns and characteristics of different levels of predictors and how do predictors affect the price.

1. Limitation

When I worked on model check for multiple regression, there are some data points which have large residuals. I did not remove these points at the beginning since I thought it might not be the influential outliers. However, the model fit better after I adjusted the original model.

1. Future Direction

Building upon current data analysis, it would be interesting to work on the booking records from other cities in the US. Would it be appropriate to use logistic regression as well in different places, or is it only feasible in Los Angela’s? The inferences would be solidified by further analysis. Airbnb sharing economy should have much more tremendous positive influence not only on travelers but property owners.

Reference

https://bookdown.org/roback/bookdown-bysh/

Appendix:

Other EDA:

A picture containing indoor

Description automatically generated

R code:

knitr::opts\_chunk$set(echo = TRUE)

library(randomForest)

library(lmerTest)

library(car)

library(ggplot2)

library(gridExtra)

library(stats)

library(lmtest)

list=read.csv("listings.csv")

list<-list[,-c(2,3,4,9,16,17,18,19,20,21,23,28,30,31,33,34,41,43,44,47,48,70,71,72,73,74,75,77,78,79,80,81,82,84,85,86,88,89,90,91,92,93,94,95,96,98,102,103,104,105)]

list\_sum<-read.csv("listings\_sum.csv")

#review\_final<-read.csv("reviews\_final.csv")

# Select 5000 random rows

#review<-review\_final[sample(nrow(review\_final), 5000), ]

#write.csv(review,file="review.csv")

review<-read.csv("review.csv")

library(Amelia)

#Check any NA

missmap(list,col=c('yellow','black'),y.at=1,y.labels='',legend=TRUE)

#drop irrelevant columns

list<-list[,-c(39,41,42)]

#check proportition of NA in whole dataset

missmap(list,col=c('yellow','black'),y.at=1,y.labels='',legend=TRUE)

list$host\_since<-list\_sum$host\_since

list[list==""]<-NA

#drop missing values completely

list<-na.omit(list)

write.csv(list,file="list.csv")

#remove rows that have N/A in "host\_response\_time","host\_response\_rate""

library(dplyr)

list = filter(list, host\_response\_time != "N/A" & host\_response\_rate != "N/A")

#write.csv(list,file="airbnb.csv")

#Display the data dimensions

dim(list)

# Display the column names

colnames(list)

# Display the data structures

str(list)

#review/summaries text analysis

library(gridExtra)

library(grid)

par(mfrow=c(4,4))

library(plotly)

library(ggthemes)

g<-ggplot(data = list) +

geom\_bar(aes(x = bedrooms),fill="#D53E4F") +

xlab('Bedrooms') +

labs(title = "Numbers of bedrooms")+xlim(-1, 10)+

theme(plot.title = element\_text(hjust = 0.5,size=13),panel.grid.major =element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(),axis.line = element\_line(colour = "black"))

h<-ggplot(data = list) +

geom\_bar(aes(x = beds),fill="#DE77AE") +

xlab('Beds') +

labs(title = "Numbers of beds")+xlim(-1, 10)+

theme(plot.title = element\_text(hjust = 0.5,size=13),panel.grid.major =element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(),axis.line = element\_line(colour = "black"))

j<-ggplot(data = list) +

geom\_bar(aes(x = accommodates),fill="#3288BD") +

xlab('Accommodates') +

labs(title = "Numbers of accommodates")+xlim(0, 10)+

theme(plot.title = element\_text(hjust = 0.5,size=13),panel.grid.major =element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(),axis.line = element\_line(colour = "black"))

i<-ggplot(data = list) +

geom\_bar(aes(x = bathrooms),fill="#FB8072") +

xlab('Bathrooms') +

labs(title = "Numbers of bathrooms")+xlim(0, 7)+

theme(plot.title = element\_text(hjust = 0.5,size=13),panel.grid.major =element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(),axis.line = element\_line(colour = "black"))

grid.arrange(g, h, j,i ,ncol=2)

#price analysis

ggplot(data = list) +

geom\_bar(aes(x = price),fill="#FB8072") +

xlab('Price') +

labs(title = "Price")+xlim(-1, 500)+

theme(plot.title = element\_text(hjust = 0.5,size=13),panel.grid.major =element\_blank(), panel.grid.minor = element\_blank(),

panel.background = element\_blank(),axis.line = element\_line(colour = "black"))

#neighborhood analysis

library(dplyr)

library(kableExtra)

list1<-list%>%group\_by(list$neighbourhood) %>% summarise(number = n())%>%arrange(desc(number))

head(list1)

Latitude<-list[,28]

Latitude<-data.frame(Latitude)

Latitude$long<-list[,29]

Latitude\_sample<-Latitude[sample(nrow(Latitude), 100), ]

# get the location of housing

#library(leaflet)

#Latitude\_sample %>%

#leaflet() %>%

# addTiles() %>%

# addTiles() %>%

#addMarkers(popup="sites")

#property type analysis

list$property\_type = as.factor(list$property\_type)

ggplot(aes(x = property\_type, y = bathrooms,color=property\_type,fill=property\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,15))+

theme( axis.text.x = element\_text(angle=90, hjust=1, vjust=0.9))+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

#room type analysis

par(mfrow=c(4,4))

list$room\_typee = as.factor(list$room\_type)

ggplot(aes(x = room\_type, y = bathrooms,color=room\_type,fill=room\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,15))+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

xlab('Bathrooms') +

labs(title = "Numbers of bathrooms")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

ggplot(aes(x = room\_type, y = accommodates,color=room\_type,fill=room\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,15))+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

xlab('Bathrooms') +

labs(title = "Numbers of accommodates")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

ggplot(aes(x = room\_type, y = beds,color=room\_type,fill=room\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,15))+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

xlab('Beds') +

labs(title = "Numbers of beds")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

ggplot(aes(x = room\_type, y = bedrooms,color=room\_type,fill=room\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,15))+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

xlab('Bedrooms') +

labs(title = "Numbers of bedrooms")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

ggplot(aes(x = room\_type, y = security\_deposit,color=room\_type,fill=room\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,1500))+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

xlab('Security Deposit') +

labs(title = "Numbers of security\_deposit")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

ggplot(aes(x = room\_type, y = cleaning\_fee,color=room\_type,fill=room\_type), data = list) +

geom\_boxplot() +

geom\_jitter(alpha = 0.1)+

coord\_cartesian(ylim=c(0,200))+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

xlab('Cleaning Fee') +

labs(title = "Cleaning Fee")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

#number of booking records based on room type

m<-ggplot(aes(x = price,fill=room\_type,color=room\_type), data = list) +

geom\_histogram()+

facet\_wrap(~room\_type)+xlim(0, 500)+

theme( axis.text.x = element\_text(angle=35, hjust=1, vjust=0.9))+

labs(x = "Price", y = " Count") +

ggtitle("Number of Booking Records among each Room Type")+

scale\_fill\_viridis\_d(option = "viridis") +

scale\_color\_viridis\_d(option = "viridis") +

theme\_pander()

m

#room type numbers in each category

library(dplyr)

list\_roomtype<-list%>%group\_by(list$room\_type) %>% summarise(number = n())%>%arrange(desc(number))

library(knitr)

kable(list\_roomtype,format = "markdown")

library("RColorBrewer")

#pie chart

# Pie Chart with Percentages

slices <- c(28468, 14410, 1769, 406)

lbls <- c("Entire home/Apt", "Private Room", "Shared Room", "Hotel Room")

pct <- round(slices/sum(slices)\*100)

lbls <- paste(lbls, pct) # add percents to labels

lbls <- paste(lbls,"%",sep="") # ad % to labels

coul <- brewer.pal(5, "BuPu")

pie(slices,labels = lbls, col=coul,

main="Pie Chart of Room Type")

library(tidyverse)

library(tidytext)

library(knitr)

library(textdata)

library(magrittr)

summary<-data.frame(list$summary)

summary$list.summary<-as.character(summary$list.summary)

tidy\_word <- summary %>%

unnest\_tokens(word,list.summary)

#find the most frequently used words in summary

library(wordcloud)

library(magrittr)

tidy\_word %>%

anti\_join(stop\_words) %>%

count(word) %>%

with(wordcloud(word, n, max.words = 20,colors = brewer.pal(7, 'Dark2'), random.order = FALSE,rot.per=0.75))

review<-na.omit(review)

comment<-data.frame(review$comments)

comment$review.comments<-as.character(comment$review.comments)

tidy\_word\_com <- comment %>%

unnest\_tokens(word,review.comments)

tidy\_word\_com %>%

anti\_join(stop\_words) %>%

count(word) %>%

with(wordcloud(word, n, max.words = 100,colors = brewer.pal(7, 'Dark2'), random.order = FALSE,rot.per=0.35))

#get relevant columns in the dataset for regression

model\_data<-list[,c(33:36,39:45,47,48)]

head(model\_data)

#Find the corrleation among each variable

set.seed(200)

library(GGally)

ggpairs(model\_data,cardinality\_threshold = 100) +

theme(text = element\_text(size = 8)) +

theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1, size = 4))

#correlation plot

pairs(model\_data)

#Correlogram

library(corrgram)

library(PerformanceAnalytics)

corMat <- cor(model\_data, use = "complete")

round(corMat, 3)

library(corrplot)

corrplot(cor(model\_data), method = "circle")

#multiple regression

linear<-lm(price~extra\_people+maximum\_nights+number\_of\_reviews+review\_scores\_rating+guests\_included,data=model\_data)

summary(linear)

#calculate AIC

AIC(linear)

plot(residuals(linear))

#residual plot

hist(linear$residuals)

#calcualte VIF

vif(linear)

#drop some observations(might be outlier)

model\_data<-model\_data[-c(5519,6209),]

adj.linear<-lm(price~extra\_people+maximum\_nights+number\_of\_reviews+review\_scores\_rating+guests\_included,data=model\_data)

summary(adj.linear)

#model check( adjusted model)

plot(adj.linear)

#model check(old model)

plot(linear)

# F-statistic

summary(linear)$fstatistic

# confidence interval

confint(linear)

# visualize the confidence intervals

library(coefplot)

coefplot(linear, intercept = FALSE)

dwt(linear)

#make a scatter plot

ggplot(list)+aes(x=accommodates,y=price)+

geom\_point()+geom\_smooth(method="lm",se=FALSE)

model2<-lm(price~accommodates+bathrooms+bedrooms+beds+cleaning\_fee+security\_deposit,data=list)

summary(model2)

AIC(model2)

model1<-lm(price~accommodates+bathrooms+bedrooms+beds+cleaning\_fee+guests\_included,data=list)

summary(model1)

hist(model1$residuals)

qqnorm(model1$residuals)

qqline(model1$residuals)

library(coefplot)

coefplot(model1)

plot(fitted(model1),model1$residuals)

abline(0,0,col="red")

library(tidyverse)

library(gridExtra)

library(car)

#Checking the assumption of independence

dwt(model1)

# VIF

vif(model1)

# tolerance

1/vif(model1)

# mean VIF

mean(vif(model1))

plot(model1)

# F-statistic

summary(model1)$fstatistic

# confidence interval

confint(model1)

# visualize the confidence intervals

library(coefplot)

coefplot(model1, intercept = FALSE)

library(MASS)

#Shapiro-Wilk Normality Test

## Distribution of studentized residuals

student\_residuals <- studres(model1)

shapiro.test(sample(student\_residuals, size = 5000))

#p-value is less than 0.05, reject the null hypothesis that residuals are normally distributed.

#change price into binary outcome

library(magrittr)

library(tidyverse)

log\_list<-list

log\_list$price<-as.factor(ifelse(log\_list$price>400,1,0))

#get confusion matrix

(table(log\_list$price))

6759/(6759+421)

library(caTools)

#Splitting Training & Testing Data

# Randomly split data

set.seed(6888)

split = sample.split(log\_list$price, SplitRatio = 0.94)

# Create training and testing sets

priceTrain = subset(log\_list, split == TRUE)

priceTrain<-data.frame(priceTrain)

priceTest = subset(log\_list, split == FALSE)

priceTest<-data.frame(priceTest)

nrow(priceTrain)

nrow(priceTest)

#logistic regression

logistic.model1=glm(price~accommodates+bathrooms+bedrooms+beds+cleaning\_fee+security\_deposit,data=priceTrain , family=binomial )

summary(logistic.model1)

#model check

library(car)

marginalModelPlots(logistic.model1)

#binned residual plot

library(arm)

binnedplot(fitted(logistic.model1),residuals(logistic.model1,type="response"))

#use model to get prediction

predictTrain = predict(logistic.model1, type="response")

summary(predictTrain)

tapply(predictTrain, priceTrain$price, mean)

#Confusion matrix for threshold of 0.5

table(priceTrain$price, predictTrain > 0.5)

#sensitivity

82/(339+82)

#Iload ROCR package

library(ROCR)

ROCRpred = prediction(predictTrain, priceTrain$price)

# Performance function

ROCRperf = performance(ROCRpred, "tpr", "fpr")

# Add threshold labels

plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))

predictTest = predict(logistic.model1, type = "response", newdata = priceTest)

table(priceTest$price,predictTest >= 0.3)

# Accuracy

(400+9)/(409+22)

#multi-level regression

data<-read.csv("list.csv")

neigh<-data$neighbourhood

data<-data[,c(40:55)]

#data$neigh<-neigh

data$host\_since<-NULL

data$calendar\_updated<-NULL

data$cancellation\_policy<-as.factor(data$cancellation\_policy)

data$require\_guest\_phone\_verification<-as.factor(data$require\_guest\_phone\_verification)

data$require\_guest\_profile\_picture<-as.factor(data$require\_guest\_profile\_picture)

index<-sample(7180,7180/2,replace = F)

variable<-data[index,]

modelset<-data[-index,]

# function for leave-one-out cv

looCv<- function(model){

mean(residuals(model)^2/(1-hatvalues(model))^2)

}

# random forest

mod1<-randomForest(price~.,data = variable,importance=T,ntree=500)

mod2<-randomForest(price~.,data = variable,importance=T,ntree=1000)

varImpPlot(mod1)

varImpPlot(mod2)

# select top 6

finaldata<-data[,c(1,2,3,4,5)]

finaldata$neigh<-neigh

finaldata$cancelPolicy<-data$cancellation\_policy

finaldata$instant\_bookable<-data$instant\_bookable

#find relationship between price and each variable

p1<-ggplot(finaldata, aes(x=security\_deposit, y=price)) +

geom\_smooth()

p2<-ggplot(finaldata, aes(x=cleaning\_fee, y=price)) +

geom\_smooth()

p3<-ggplot(finaldata, aes(x=guests\_included, y=price)) +

geom\_smooth()

p4<-ggplot(finaldata, aes(x=extra\_people, y=price)) +

geom\_smooth()

grid.arrange(p1,p2,p3,p4,nrow=2)

# random intercept

finaldata<-na.omit(finaldata)

mod1<-lmer(price^0.1~security\_deposit+cleaning\_fee+guests\_included+(-extra\_people^2)+cancelPolicy+instant\_bookable+(1|neigh),data = finaldata)

# normality of residual

qqnorm(residuals(mod1))

qqline(residuals(mod1))

summary(mod1)

#head(coef(mod1))

library(arm)

display(mod1)

# normality of parameters

para<-data.frame(ranef(mod1))

qqnorm(para[,4])

qqline(para[,4])

# independentce of residual

plot(residuals(mod1))

# constant variable

plot(residuals(mod1))

# check multicollinearity

vif(mod1)

#calculate MSE after cross validation for model 1

mse1<-looCv(mod1)

mse1

# random slope

mod2<-lmer(price^0.1~security\_deposit+cleaning\_fee+guests\_included+cancelPolicy+instant\_bookable+(0+cleaning\_fee|neigh),data = finaldata)

# normality of residual

qqnorm(residuals(mod2))

qqline(residuals(mod2))

# normality of parameters

para<-data.frame(ranef(mod2))

qqnorm(para[,4])

qqline(para[,4])

# independentce of residual

plot(residuals(mod2))

# constant variance

plot(residuals(mod2))

# check multicollinearity

vif(mod2)

summary(mod2)

#coef(mod2)

display(mod2)

#calculate MSE after cross validation for model 2

mse2<-looCv(mod2)

mse2

# random intercept and slope

mod3<-lmer(price^0.1~security\_deposit+cleaning\_fee+guests\_included+extra\_people+cancelPolicy+instant\_bookable+(1+security\_deposit+cleaning\_fee|neigh),data = finaldata)

# normality of residual

qqnorm(residuals(mod3))

qqline(residuals(mod3))

# normality of parameters

para<-data.frame(ranef(mod3))

qqnorm(para[,4])

qqline(para[,4])

# independentce of residual

plot(residuals(mod3))

# constant variable

plot(residuals(mod3))

# check multicollinearity

vif(mod3)

#graph relationship between price and each variable grouped by different category of instant\_bookable

p1<-ggplot(finaldata, aes(x=security\_deposit, y=price,color=instant\_bookable)) +

geom\_smooth()

p2<-ggplot(finaldata, aes(x=cleaning\_fee, y=price,color=instant\_bookable)) +

geom\_smooth()

p3<-ggplot(finaldata, aes(x=guests\_included, y=price,color=instant\_bookable)) +

geom\_smooth()

p4<-ggplot(finaldata, aes(x=extra\_people, y=price,color=instant\_bookable)) +

geom\_smooth()

grid.arrange(p1,p2,p3,p4,nrow=2)

#calculate MSE after cross validation for model 3

mse3<-looCv(mod3)

mse3

compare <- cbind(mse1,mse2,mse3)%>%as.data.frame()

knitr::kable(compare)%>%kableExtra::kable\_styling(bootstrap\_options = c("striped", "hover"))

# random intercept

finaldata<-na.omit(finaldata)

mod4<-lmer(price^0.1~security\_deposit+cleaning\_fee+guests\_included+(-extra\_people^2)+cancelPolicy+(1|instant\_bookable),data = finaldata)

display(mod4)

# normality of residual

qqnorm(residuals(mod4))

qqline(residuals(mod4))

# normality of parameters

para<-data.frame(ranef(mod4))

qqnorm(para[,4])

qqline(para[,4])

# independentce of residual

plot(residuals(mod4))

# constant variable

plot(residuals(mod4))

# check multicollinearity

vif(mod4)

mse4<-looCv(mod4)

mse4

glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 10000000))

mod5<-lmer(price~security\_deposit+cleaning\_fee+guests\_included+cancelPolicy+(0+security\_deposit+cleaning\_fee+guests\_included|instant\_bookable),data = finaldata)

# normality of residual

qqnorm(residuals(mod5))

qqline(residuals(mod5))

# normality of parameters

para<-data.frame(ranef(mod5))

qqnorm(para[,4])

qqline(para[,4])

# independentce of residual

plot(residuals(mod5))

# constant variance

plot(residuals(mod5))

# check multicollinearity

vif(mod5)

mse5<-looCv(mod5)

mse5

# random intercept and slope

mod6<-lmer(price^0.1~security\_deposit+cleaning\_fee+guests\_included+cancelPolicy+(1+security\_deposit+cleaning\_fee+guests\_included|instant\_bookable),data = finaldata)

# normality of residual

qqnorm(residuals(mod6))

qqline(residuals(mod6))

# normality of parameters

para<-data.frame(ranef(mod6))

qqnorm(para[,4])

qqline(para[,4])

# independentce of residual

plot(residuals(mod6))

# constant variable

plot(residuals(mod6))

# check multicollinearity

vif(mod6)

mse6<-looCv(mod6)

mse6

compare\_new <- cbind(mse4,mse5,mse6)%>%as.data.frame()

knitr::kable(compare\_new)%>%kableExtra::kable\_styling(bootstrap\_options = c("striped", "hover"))