**Dongquan Qiu**

**APAN 5200**

**Kaggle Project Report**

According to approach the Kaggle project, I was trying to make prediction prices based on the dataset about Airbnb rentals with given data. Within the dataset, there are 36839 observations under 91 variables, including character variables, numeric variables, logical variables, and dates as following: host lisitings\_count, accommodates, beds, price, security deposit, cleaning fee, number of reviews and reviews. We need to select and adjust needed data in order make the prediction with the model successfully.

First, I need to clean the dataset since there are some variables that may not be necessary to put into the prediction model. In addition, repetitive information including location and street of Airbnb rentals dataset can be simplified. The most complicated part of the data cleaning is about the missing values following the columns. Some variables are lack of information and I decided to rule them out first. Then, I included the key variables into my consideration for the following cleaning.

The following is listed as variables that I choice in this project:

host\_listings\_count, accommodates, bedrooms, beds, price, security\_deposit, guests\_included, cleaning\_fee, minimum\_nights, maximum\_nights, number\_of\_reviews, number\_of\_reviews\_ltm, extra\_people, review\_stores\_location, review\_scores\_value, review\_scores\_rating, review\_scores\_review\_scores\_accuracy, reviews\_per\_month, availability\_30, availability\_60, availability\_90, availability\_365, alculated\_host\_listings\_count, calculated\_host\_listings\_count\_etire\_homes, calculated\_host\_listing\_count\_private\_rooms,

calculated\_host\_listing\_cout\_shared\_rooms, neighbourhood\_group\_cleansed, room\_type, property\_type, cancellation\_policy.

After determining the variables and processing data cleaning, I tried to recognize the missing values “NA”first and then adjust them. At the beginning I decided to replace them with the mean for rest of the same variables but lately I changed some with 0. For example, for the variables of host\_listings\_count, beds, cleaning\_fee, reviews\_per\_month, I replaced the “NA” with means. For the security deposit, I changed the “NA” with number 0. They are adjusted in the analysisData1 dataset. Then I applied the same way into the scoringData and adjusted the missing values. If given more time for improvement, I could try to implement packages such as ‘caret’ to replace the missing values rather than simply replace with mean values.

Then, for the feature selection part, I chose to use forward selection to help determine necessary variables to put my prediction models. After the forward selection, the updated variables included host\_listings\_couht, accommodates, bedrooms, beds, price, security\_deposit, guests\_included, cleaning\_fee, minimum\_nights, maimum\_nights, number\_of\_reviews, number\_of\_reviews\_ltm, extra\_people, review\_stores\_location, review\_scores\_value, review\_scores\_rating, review\_scores\_review\_scores\_accuracy, reviews\_per\_month, availability\_30, availability\_60, availability\_90, availability\_365, calculated\_host\_listings\_count, calculated\_host\_listings\_count\_etire\_homes, calculated\_host\_listing\_count\_private\_rooms, calculated\_host\_listing\_cout\_shared\_rooms, neighbourhood\_group\_cleansed, room\_type, property\_type, cancellation\_policy. If given more time, I think I can try other selectios including subset selection to help me decide with better variables choices.

Finally, I applied random forest model with updated variables and factors to predict prices. I also include cvForest to help decide with the best option for mtry. I found out that each time when I input different numbers of ntree, the results would slight change. I have tried ntree = 100, 200, 400 in the model; however, the drawback of random forest is that it takes so much time running it and I could not easily input higher number of ntree like the boosting model. If given more time, I would like to try with boosting model since it takes less time to perform the results in the model.

----------------------------------------------------------------------------------

**R Code:**

library(randomForest)

library(lattice)

library(caret)

library(ggplot2)

library(tidyr)

library(leaps)

analysisData <- read\_csv("~/Desktop/APAN5200/analysisData.csv")

scoringData <- read\_csv("~/Desktop/APAN5200/scoringData.csv")

analysisData$calculated\_host\_listings\_count\_shared\_rooms

#data clean, remove all variables with words another irrelevant variables

analysisData1 = subset(analysisData, select = c(host\_listings\_count, accommodates, bedrooms, beds, price, security\_deposit,

guests\_included, cleaning\_fee, minimum\_nights,maximum\_nights, number\_of\_reviews,

number\_of\_reviews\_ltm, extra\_people, review\_scores\_location, review\_scores\_value,

review\_scores\_rating, review\_scores\_accuracy, reviews\_per\_month,

availability\_30, availability\_60, availability\_90, availability\_365,

calculated\_host\_listings\_count, calculated\_host\_listings\_count\_entire\_homes,

calculated\_host\_listings\_count\_private\_rooms, calculated\_host\_listings\_count\_shared\_rooms,

neighbourhood\_group\_cleansed, room\_type, property\_type, cancellation\_policy))

#check for missing values and adjust them

sum(is.na(analysisData1$host\_listings\_count))

sum(is.na(analysisData1$accommodates))

sum(is.na(analysisData1$bathrooms))

sum(is.na(analysisData1$bedrooms))

sum(is.na(analysisData1$beds))

sum(is.na(analysisData1$price))

sum(is.na(analysisData1$security\_deposit))

sum(is.na(analysisData1$cleaning\_fee))

sum(is.na(analysisData1$guests\_included))

sum(is.na(analysisData1$extra\_people))

sum(is.na(analysisData1$minimum\_nights))

sum(is.na(analysisData1$maximum\_nights))

sum(is.na(analysisData1$number\_of\_reviews))

sum(is.na(analysisData1$number\_of\_reviews\_ltm))

sum(is.na(analysisData1$review\_scores\_location))

sum(is.na(analysisData1$review\_scores\_value))

sum(is.na(analysisData1$review\_scores\_rating))

sum(is.na(analysisData1$review\_scores\_accuracy))

sum(is.na(analysisData1$availability\_30))

sum(is.na(analysisData1$availability\_60))

sum(is.na(analysisData1$availability\_90))

sum(is.na(analysisData1$availability\_365))

sum(is.na(analysisData1$calculated\_host\_listings\_count))

sum(is.na(analysisData1$calculated\_host\_listings\_count\_entire\_homes))

sum(is.na(analysisData1$calculated\_host\_listings\_count\_private\_rooms))

sum(is.na(analysisData1$calculated\_host\_listings\_count\_shared\_rooms))

sum(is.na(analysisData1$reviews\_per\_month))

#host\_listings\_count

analysisData1$host\_listings\_count[which(is.na(analysisData1$host\_listings\_count))] =

mean(analysisData1$host\_listings\_count,na.rm = TRUE)

scoringData$host\_listings\_count[which(is.na(scoringData$host\_listings\_count))] =

mean(scoringData$host\_listings\_count,na.rm = TRUE)

#beds

analysisData1$beds[which(is.na(analysisData1$beds))] =

mean(analysisData1$beds,na.rm = TRUE)

scoringData$beds[which(is.na(scoringData$beds))] =

mean(scoringData$beds,na.rm = TRUE)

#security

analysisData1$security\_deposit[which(is.na(analysisData1$security\_deposit))] = 0

scoringData$security\_deposit[which(is.na(scoringData$security\_deposit))] = 0

#cleaning\_fee

analysisData1$cleaning\_fee[which(is.na(analysisData1$cleaning\_fee))] =

mean(analysisData1$cleaning\_fee,na.rm = TRUE)

scoringData$cleaning\_fee[which(is.na(scoringData$cleaning\_fee))] =

mean(scoringData$cleaning\_fee,na.rm = TRUE)

#reviews\_per\_month

analysisData1$reviews\_per\_month[which(is.na(analysisData1$reviews\_per\_month))] =

mean(analysisData1$reviews\_per\_month,na.rm = TRUE)

scoringData$reviews\_per\_month[which(is.na(scoringData$reviews\_per\_month))] =

mean(scoringData$reviews\_per\_month,na.rm = TRUE)

#change factors variables

analysisData1$cancellation\_policy = as.factor(analysisData1$cancellation\_policy)

analysisData1$neighbourhood\_group\_cleansed = as.factor(analysisData1$neighbourhood\_group\_cleansed)

analysisData1$room\_type = as.factor(analysisData1$room\_type)

analysisData1$property\_type = as.factor(analysisData1$property\_type)

set.seed(100)

train <- sample(nrow(analysisData1), 0.8\*nrow(analysisData1), replace = FALSE) ##try small

trains<- analysisData1[train,]

tests <- analysisData1[-train,]

dim(tests)

#Feature Selection

#Forward Selection

start\_mod = lm(price~1,data=analysisData1)

empty\_mod = lm(price~1,data=analysisData1)

full\_mod = lm(price~.,data=analysisData1)

forwardStepwise = step(start\_mod,

scope=list(upper=full\_mod,lower=empty\_mod),

direction='forward')

#random forest

test\_model <- randomForest(price~ accommodates+ bedrooms+ price+

guests\_included+ minimum\_nights+maximum\_nights+number\_of\_reviews+

number\_of\_reviews\_ltm+extra\_people+review\_scores\_location+ review\_scores\_value+

review\_scores\_rating+review\_scores\_accuracy+

availability\_30+availability\_60+ availability\_90+ availability\_365+

calculated\_host\_listings\_count+ calculated\_host\_listings\_count\_entire\_homes+

calculated\_host\_listings\_count\_private\_rooms+ calculated\_host\_listings\_count\_shared\_rooms+

neighbourhood\_group\_cleansed+room\_type+property\_type+ cancellation\_policy,

ntree = 400, mtry = 10, data = trains, importance = TRUE)

predRForest = predict(test\_model,newdata=test)

rmse = sqrt(mean((predRForest-scoringData$price)^2))

trControl=trainControl(method="cv",number=10)

tuneGrid = expand.grid(mtry=1:10)

set.seed(200)

cvForest = train(price~host\_listings\_count+accommodates+bedrooms+extra\_people+availability\_30+availability\_60+availability\_90+availability\_365+

number\_of\_reviews+number\_of\_reviews\_ltm+review\_scores\_rating+review\_scores\_location+

review\_scores\_value+calculated\_host\_listings\_count\_entire\_homes+calculated\_host\_listings\_count\_private\_rooms+

calculated\_host\_listings\_count\_shared\_rooms+reviews\_per\_month+cleaning\_fee+neighbourhood\_group\_cleansed+

room\_type+cancellation\_policy+security\_deposit,data=trains, method="rf",ntree=400,trControl=trControl,tuneGrid=tuneGrid )

#submission for prediction

submissionFile = data.frame(id = scoringData$id, price = predRForest)

write.csv(submissionFile, 'sample\_submission.csv',row.names = F)