My Kaggle Project Report

Shiyu Hu

November 29, 2021

Predicting Rent in New York

This project described how I predict the price for a rental using 90 variables on the property, host, and past reviews by analyzing a listing of over 40,000 Airbnb rentals in New York City

1.Initial exploration

I read the csv file in the R studio first. By using str function to analysisData, I found there were three main types of data: number, character, and time. For convenience in using models later, I converted character and time into number and factor. By view function, I had a direct view for the rental table for all 90 variables. I found square_feet, monthly_ price, license, and jurisdiction_names had large missing data, so I directly removed them from my dataset. The weekly_price has large missing data, but I think it is an important factor. I remained it in my dataset. Based on my business sense, I removed column host_name. The street, country, country_code market, smart_location and zipcode are similar feature, I only choose city into my dataset.

2.Data Cleaning

a. character to number

For the column with rental descriptive sentences, like column 2 to 10, I decided to use the number of words as the new feature. If the data set is blank, counting as 0.

```
analysisData$summary<-
ifelse(analysisData$summary=='',0,sapply(strsplit(analysisData$summary,' '),length)+1)
```

For character like "70%", I used gsub to delete percentage sign and then used as.numeric.

b. time to number

The original type of host_since, first_review, and last_review were character. Using as.Date and as.difftime, I got the time span from that day to now as my new feature, and counted it as unit month. If there was NA, changing the data with the mean of length.

```
analysisData$host_since<-as.Date(analysisData$host_since)
analysisData$host_since<-as.difftime(as.Date('2021-11-15')-
as.Date(analysisData$host_since))
analysisData$host_since<-as.numeric(analysisData$host_since)/30
```

c. character to factor
   After a rough glimpse at the analysisData table, I used table function to make sure if a character feature has limited levels. If it is true, using as.factor. If there was NA level, changing the data to the majority class.

```
table(analysisData$instant_bookable)
        f     t
    24884 16446
```

d. checking the numerical column
   Using function summary() to each numerical column, I checked if there was outlier (max or min) or NA entry. If there was outlier, deleting the specific data row. If there was NA, replenishing with mean value.

e. City column
   This was a messy but important variable. For the same city name, there were several types of formats. Table function clearly arranged all types of formats. I firstly converted several Chinese character into English, then unified all characters to lower case. I chose first three characters as the feature class. For those with blank in the front, I individually fixed them into the class.

```
analysisData$city<-tolower(analysisData$city)
analysisData$city<-substr(analysisData$city,1,3)
analysisData$city[which(analysisData$city ==' br')] = 'bro'
```

f. scoring data
   Doing the same cleaning steps for scoring data.

3. Models and Feature Selection

a. Feature Selection
   I only used lasso regression to do feature selection with alpha 1 and mean square error. It selected 29 significant variables. In fact, in the model training process, I found the model without feature selection could get a better prediction. So, I did not use the feature selection outcome as my final submission dataset.

b. Linear Regression with lasso regression
   For the prediction outcome, I fixed the negative price to the mean price. And I got the MSE = **76.26543**. When I used linear regression model to predict the price, I got the error message when there was new levels appearing, like '75 months ago' in calendar_updated

column. I manually changed '75 months ago' to '73 months ago' since it was not influential to price.

```
model=lm(price~.,data=analysisData)
```

c. Default Tree with lassao regression
For the prediction outcome, I fixed the negative price to the mean price. And I got the MSE =**82.2098**. In the process of running tree, I found it did not accept the feature with levels above 52. I recleaned the data for city and calendar_updated and packaged the similar data to degrade levels.

```
dftree = rpart(.,data=analysisData)
```

d. Random forest with lassao regression
For the random forest, I got a huge decrease in mean square error **68.33930**. I would compare the summary of analysis price and predict price to roughly check the accuracy of my prediction.

```
install.packages("randomForest")
library(randomForest)
forest=randomForest(price~.,data=analysisData,ntree = 1000,mtry=5)
```

e. Boosting model with lassao regression
Since I found that the tree model accepted the NA value, I added weekly_price into the model. And I got the MSE =**67.04256** by gbm function with 9532 for n.trees, 5 for interaction.depth, 0.005 for shrinkage and 5 for n.minobsinnode.

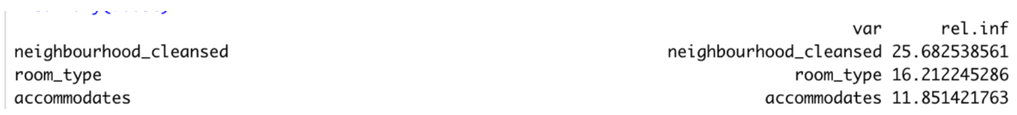
f. xgboosting model with lassao regression
I got the MSE =**105** for this model. It run faster than boosting model, but I did not choose xgboosting as my final submission.

g. Boosting model without feature selection (final submission)
Public score is **66.60866** and private score is **60.39846** with parameters n.trees = 9532, interaction.depth = 5, shrinkage = 0.005, n.minobsinnode = 5

4.Disscussion

I tried to tune the boosting model with cross_validation Boost. Even I set small vector for tuneGrid, it failed to get the outcome when I run it overnight. If I have more time and faster server, I can tune my model with more accrue parameter for gbm model. What I have done so far is adjusting the parameter slightly to run the model and compare the mean square error for test data prediction to find the best parameter.

For my final model, neighbourhood_cleased, room_type, and accommodates are top three variables relevant to the rental price.



```
                                                                     var      rel.inf
neighbourhood_cleansed                                neighbourhood_cleansed 25.682538561
room_type                                                          room_type 16.212245286
accommodates                                                    accommodates 11.851421763
```