BUS212A Final Project

Yuzhou Liu, Leiyuxiang Wu, Yutian Lai

12/11/2018

Executive Summary

For the purpose of forecasting average occupancy and revenue for Airbnb listers, we apply 3 different models on adjusted Boston Airbnb listing data. As Airbnb lister's consultant, due to unsatisfying model performance, we cannot select a best model, but we gain some insights from the result. 1. Commercial Airbnb are not popular. 2. People are not satisfied with high cleaning fees 3. Reviews seem less important than what we expected.

1. Introduction

The goal of this project is to find factors contributing to popular Airbnb listings and provide hosts with reference on how to lift revenue by manipulating key contributing factors. We combine external data like text reviews, Boston attractions and crime rate by neighborhood with Boston Airbnb listing data, applying typical models such as RandomForest, KNN and Elastic Net Regression to find which of those variables best interpret the popularity of the given properties. Then we can use relationships between factors and properties popularity to help hosts boost revenue.

2. Data Description

The main dataset investigated here is Boston airbnb listing data from **Inside Airbnb** webiste. The airbnb lisitng data include features of about 6000 current airbnb listings around great boston area on 09/14/2018. The raw dataset contains daily price, cleaning fee, security deposite and other 27 variables. The target variable of our interests is available_90, which is the historical average 90-day availablility (# of daies). As mentioned before, one of our aims is to perdict monthly revenues for airbnb owners, and the available_90 data contains the average monthly occupation, which can convert to monthly revenue multiplying by daily price. One of the supplymenting datasets is reviews of Boston airbnb listing also from **Inside Airbnb** website. The reviews are posted by customers after leaving the places and usually including some comments and feelings about the hosts and facilities. The data contains totally 178,308 reviews and contains host id, reviewer ids and name, the review id (unique), review date and reviews in text, as shown in the samples below.

listing_id	id	date	reviewer_id	reviewer_name	comments
3781	37776825	2015-07-10	36059247	Greg	The apartment was as advertised and Frank was inc
3781	41842494	2015-08-09	10459388	Tai	It was a pleasure to stay at Frank's place. The plac
3781	45282151	2015-09-01	12264652	Damien	The apartment description is entirely faithful, and t

The features extracting from text reviews are the data actually contributing in our models, which will be discussed latter.

The second supplement dataset is "distance from main Boston attractions". In this dataset, we recorded 35 attractions and their longitudes and latititudes in the Greater Boston Area. The dataset is handmade from ourselves. First,we selected 35 popular attractions from aviewoncities.com and get their cordinates from GPS cordinates, then we calculate every single airbnb property's distance from these 35 attractions in Boston.

Crime data by neighborhoods describe crime rate, including violent crimes and property crimes, in neighborhoods of Boston. Crime rate is calculated annually per 100,000 residents, which means how many crimes happened during one year among 100,000 residents. Violent crimes contain assault, murder, rape and robbery, while property crimes contain burglary, theft and motor vehicle theft.

3. Data Preprocessing

Target Variable

For our target variable, **availability_90**, we converted it to average occupations in 90 (OCC) daies for revenue prediction by simply deducting from 90. Thus, as mentioned before, it can be easily converted to monthly revenue multiplying by daily price/3.

Input Varibales Selection and Transformation

First of all, we converted **host_id** to **same_host_lists**. Each airbnb listing has a host, while some hosts might have several listings under their management. While the orignial host id varibale is not contributing prediction power, the number of listings under same host may be useful in predicting revenue, since it shows the level of commercialization of that airbnb listing. From the experience of hotels, commercialization brings effeciency and boosts revenue. On the other hand, airbnb customers may prefer pensonalized relationship with hosts and commercialization hinders revenue growth, and we will see which effect dominates in Boston area.

```
#Convert host_id to same_host_lists
airbnb.df <- inner_join(airbnb.raw,count(airbnb.raw, host_id), "host_id")
airbnb.df$host_id <- NULL
names(airbnb.df)[30] <- "same_host_lists"

#Transforming target varibale availability_90
airbnb.df$OCC <- 90- airbnb.df$availability_90
airbnb.df$availability_90 <- NULL
```

Second we discard the following several variables from raw data. The information inside **host_name** and **Commercial** overlaps with **host_id** and its derived varibale **same_host_lists**. The rest deletion is mainly due to incomplete observations. There are over 80% NA records in **square_feet**, **weekly_price** and **monthly_price**.In addition, there are only 800 False in 6000 observations in **is_location_exact**, the variation is just not enough to get plausible prediction patterns.

```
#Discard host_name
airbnb.df$host_name <- NULL
#Discard is_location_exact
airbnb.df$is_location_exact <- NULL
#Discard square_feet
airbnb.df$square_feet <- NULL
#Discard weeekly_price
airbnb.df$weekly_price <- NULL
#Discard monthly_price
airbnb.df$monthly_price <- NULL
#Discard Commercial
airbnb.df$Commercial <- NULL
```

Third, we convert **host_since**, a date variable, to **host_duration** (in days), a numeric variable. Date variables are hardly compatible in most numerical based models, the conversion reserves the key information of

how long the airbnb has listed as an indicator for hosting experience, and makes the variable more compatible.

```
#Covert host_since to host_duration
airbnb.df$host_duration <- as.Date("2018/09/14") -
  as.Date(airbnb.df$host_since, format = "%m/%d/%Y")
airbnb.df$host_since <- NULL
airbnb.df$host_duration <- as.numeric(airbnb.df$host_duration)
```

Fourth, there are a bunch of variables indicating the neighbourhood or location of the listings. For the exact location information from **longtitude** and **latitude**, we converted it to distance from famous attractions, which will be revealed latter in External data part. For all other neighbourhood variables, they contains overlapping information on different scales, so we only keep **neighbourhood_cleansed** for our analysis. Further more, the neighbourhood information itself bring strong but vague prediction power. To be more specific, we modifed the **price** and **neighbourhood_cleansed** forming **neighbourhood_ave_price**, the average daily price in the same neighbourhood airbnbs. **neighbourhood_ave_price** mimic the simple neighbourhood price comparison process of airbnb customers, so we believe will have strong and clear prediction power for the listing revenue.

```
#Convert neighbourhood_cleansed to neighbourhood_ave_price
airbnb.df <- inner_join(airbnb.df,airbnb.df %>%
                          group_by(neighbourhood_cleansed) %>%
                          summarise(mean(price)), "neighbourhood_cleansed")
names(airbnb.df)[25] <- "neighbourhood_ave_price"
airbnb.df$neighbourhood_cleansed <- NULL

#Discard neighbourhood
airbnb.df$neighbourhood <- NULL
#Discard zipcode
airbnb.df$zipcode <- NULL
#Discard smart_location
airbnb.df$smart_location <- NULL
#Discard city
airbnb.df$city <- NULL
#Discard latitude
airbnb.df$latitude <- NULL
#Discard longitude
airbnb.df$longitude <- NULL
```

Fifth, since over 90 percent of **property_type** come from four main categories: Apartment, House, Condominium, and Serviced Apartment, we reudce the levels to five different levels and recode these types as "Apt/House/Condo/Service Apartment/Others".

```
# Recode property_type to reduce categorical levels
airbnb.df$property_type <- recode(airbnb.df$property_type,
    "Apartment"= "Apt", "House" = "House",
    "Condominium" = "Condo", "Serviced apartment" = "Serviced apartment", .default ="Other")
```

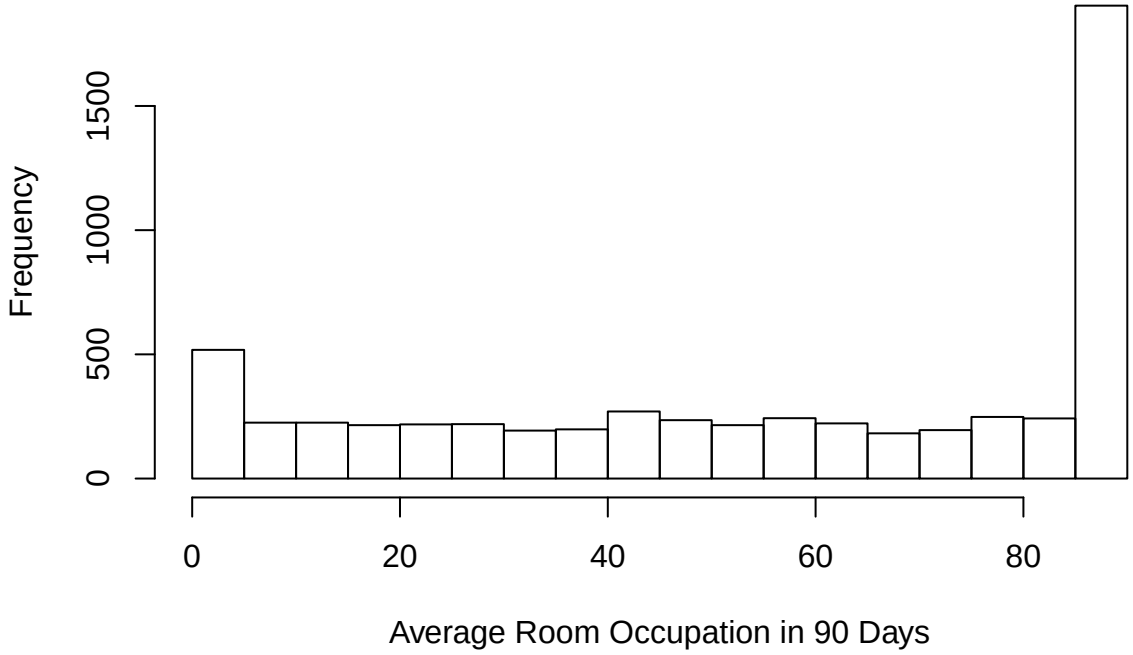
In this part, we focus on NA values and extrem (influential/outliers) values in our variables.First, in "host_response_time", NA values are classified as "a few days or more", since we assume that the NA response time is caused by no reply or no inquiry. Second, we assign NA values as 0 in "security_deposit" and "cleaning_fee", because we assume NA fees basically equalling not collecting them.

At last, we remove the influential observations from "maximum_nights","minimum_nights". When maximum nights restrictions is too large, it becomes unimportant in customers' consideration, thus losses its prediction power. Also When minimum nights restrictions is too large, it becomes special listings that only certain type customers consider it, thus losses its prediction power. In addition, 7 observations record 0 dayily price, which we believe is wrong data and pairwise delete them.

```r
#Convert NA value in host_response_time
airbnb.df$host_response_time[which(airbnb.df$host_response_time == "N/A")] <- "a few days or more"

#Deal with NA value in security_deposit
airbnb.df$security_deposit[is.na(airbnb.df$security_deposit)] <- 0

#Deal with NA value in cleaning_fee
airbnb.df$cleaning_fee[is.na(airbnb.df$cleaning_fee)] <- 0

#Remove influencial value in maximum_nights
airbnb.df <- airbnb.df %>%
  filter(maximum_nights < 2000)

#Remove influencial value in munimum_nights
airbnb.df <- airbnb.df %>%
  filter(minimum_nights < 181)

#Remove outlier in price (0 value)
airbnb.df <- airbnb.df %>%
  filter(price != 0)
```

Problems associated with the target varibale

From the simply visulization of our target variable, **OCC**, it is obvious that a large porportion of observations (about 26%) has full occupations in 90 days, which is unnormal situation in reality. In addition, the lack of variation in target variables will hurt the prediction power of our models.



Digging further into the problem, we find that most of the full occupation observations (about 60%) involving NA response rate as shown below. The NA response rate indicates the abnormal situation existence in this airbnbs. These listings involving less inquiry and responses but has high occupation rate, and the best guess we have is that these airbnbs may involve long-term rent arrangements. Thus, these observations with full occupation and NA response rate are not suitable for our revenue prediction model, as they are not normal

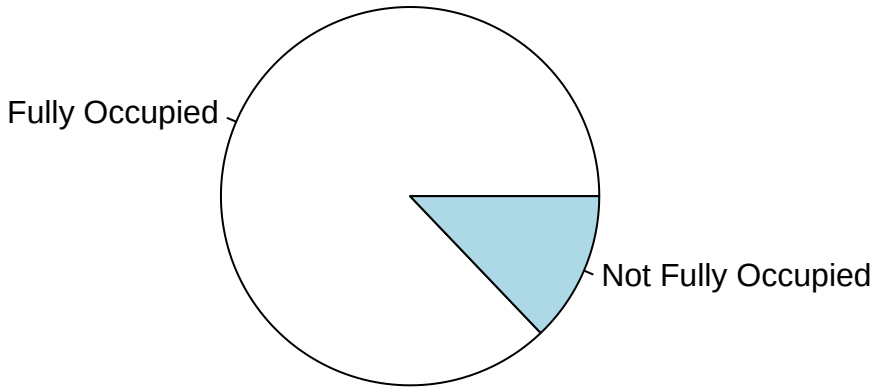
airbnb listing attracting customers on a broad basis.

```
#Covert chr value in host_response_rate
airbnb.df$host_response_rate <- as.numeric(sub("%", "", airbnb.df$host_response_rate))

#Check NA value in host_response_rate
OCC90 <- airbnb.df %>%
  filter(OCC == 90)

table(is.na(OCC90$host_response_rate))
```

```
##
## FALSE  TRUE
##   638   924
```

After we exclude all NA response rate observations, the listings with full occupation drop to about 10% of all airbnb listings, as shown in the below chart. The dataset now lines up with reality, because it is a normal phenomenon that about 10% airbnb listings are hot and usually fully occupied.

The Percentage of Fully Occupied Airbnb Listing after Adjustment



External Data

Considering that when tourists plan to travel somewhere, choose some places to live, they are likely to prefer airbnbs that are close to attractions, which can provide great convenience and save a lot transportation expenses, we believe the distance from attractions should be in the model as a variable if we intend to predict its occupancy and potential income in the future.

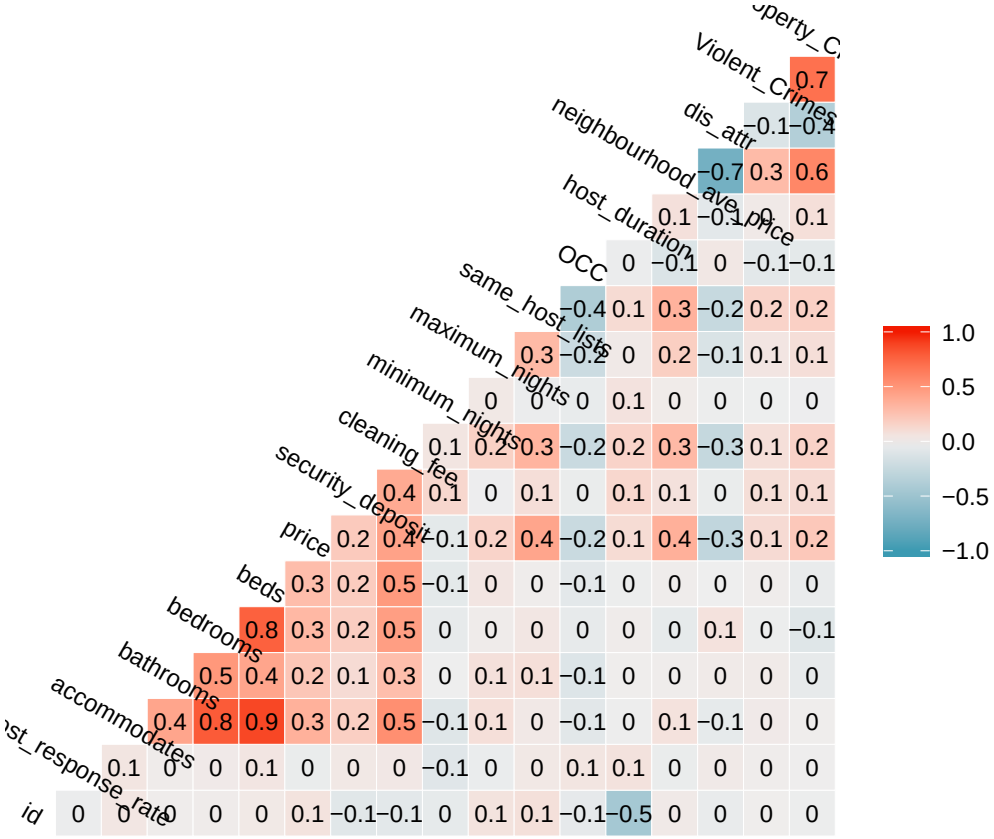
In consequence, we add one variale based on our calculation demonstrating each airbnb property's average distance to the closest 3 attractions in Boston. Besides transportation convenience, tourists also concern about safety around their living place. This is another hot spot issue we need to take into account, so we found crime rate sorted by neighborhood to add into our dataset.

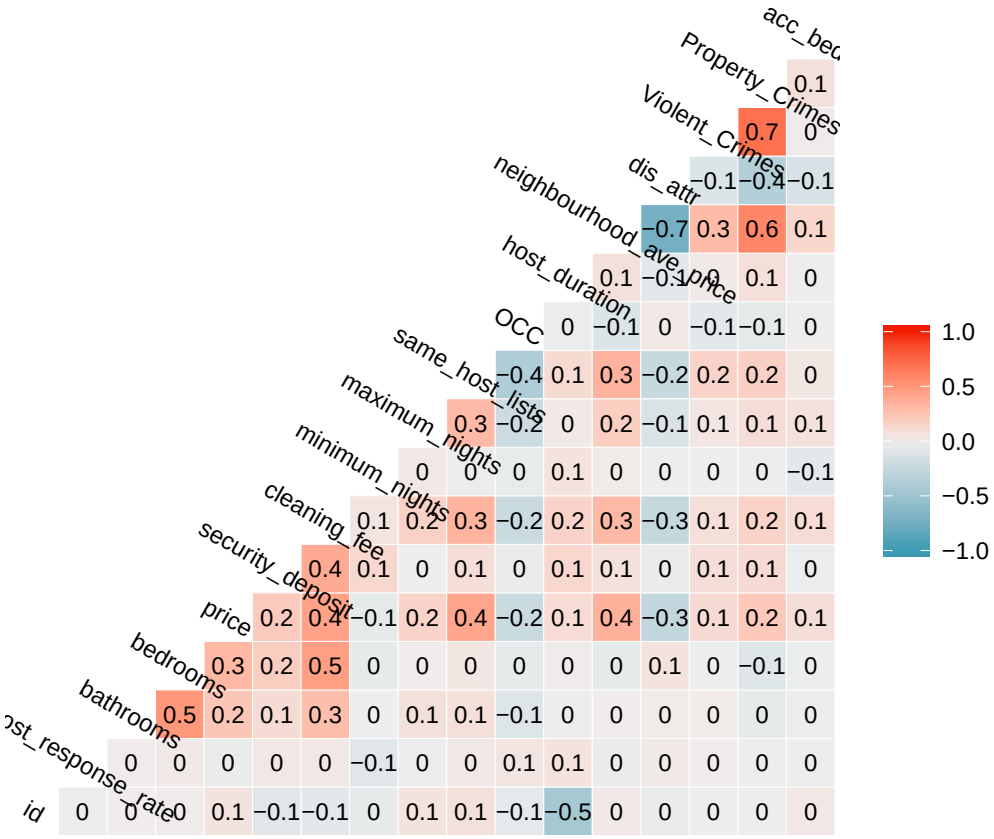
```
#Distance to top 3 close attractions
pt <- as.matrix(airbnb.raw %>% select(longitude,latitude))
dis<-matrix(rep(NA, 35*5986), ncol=35)
dis_attr<-rep(NA,5986)

for (i in 1:5986) {
dis[i,] <- spDistsN1(as.matrix(attractions.raw[,1:2]), pt[i,], longlat = TRUE)
```

```
dis_attr[i] <-mean(head(sort(dis[i,]),3))
}

dis_attr.df <- data.frame(id = airbnb.raw$id, dis_attr = dis_attr)

airbnb.df <- inner_join(airbnb.df, dis_attr.df, "id")
```

Check Input Variables

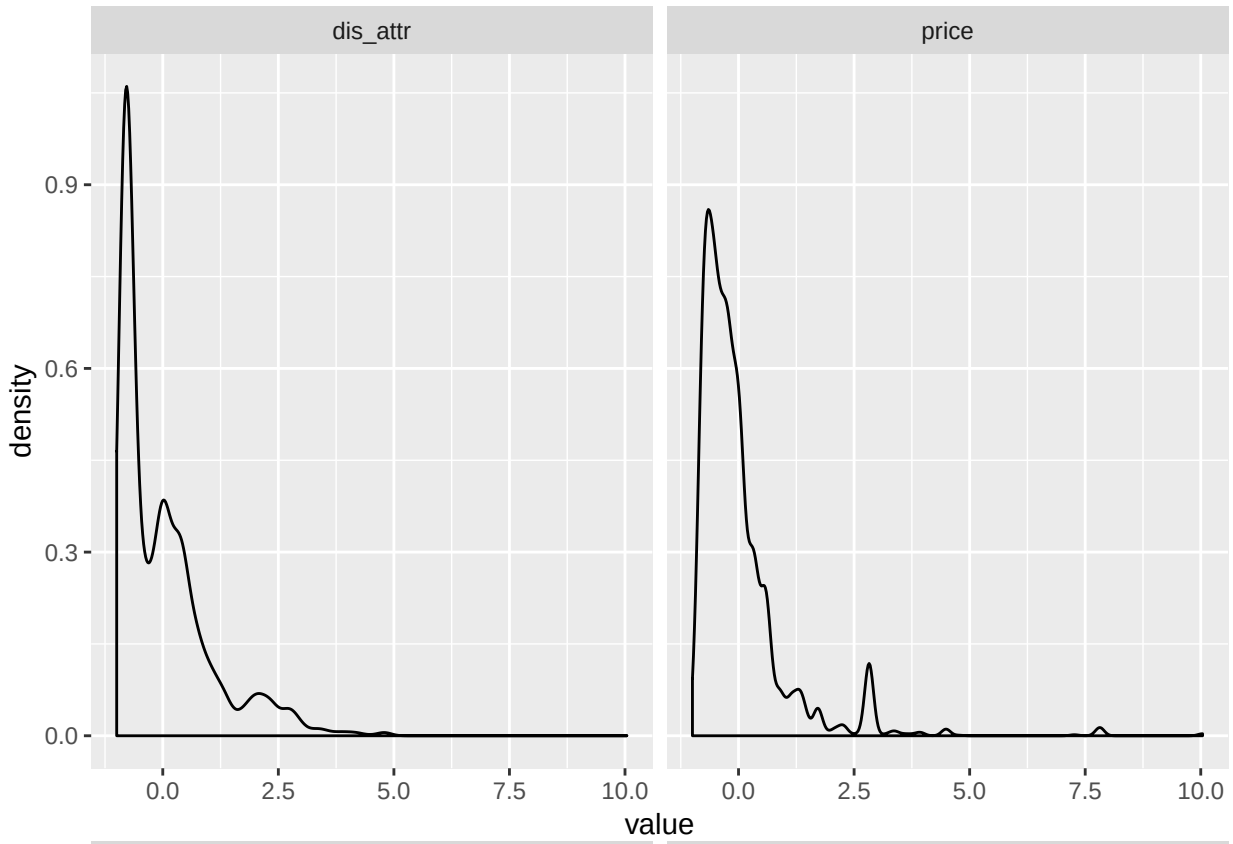
We first check correlation between normalized numeric inputs and found high correlation among number of beds, number of bathrooms and number of people the property can accommodate. This makes sense because these three inputs all increase as the property gets bigger. We decide to use accommodates per bed to replace inputs accommodates and beds.

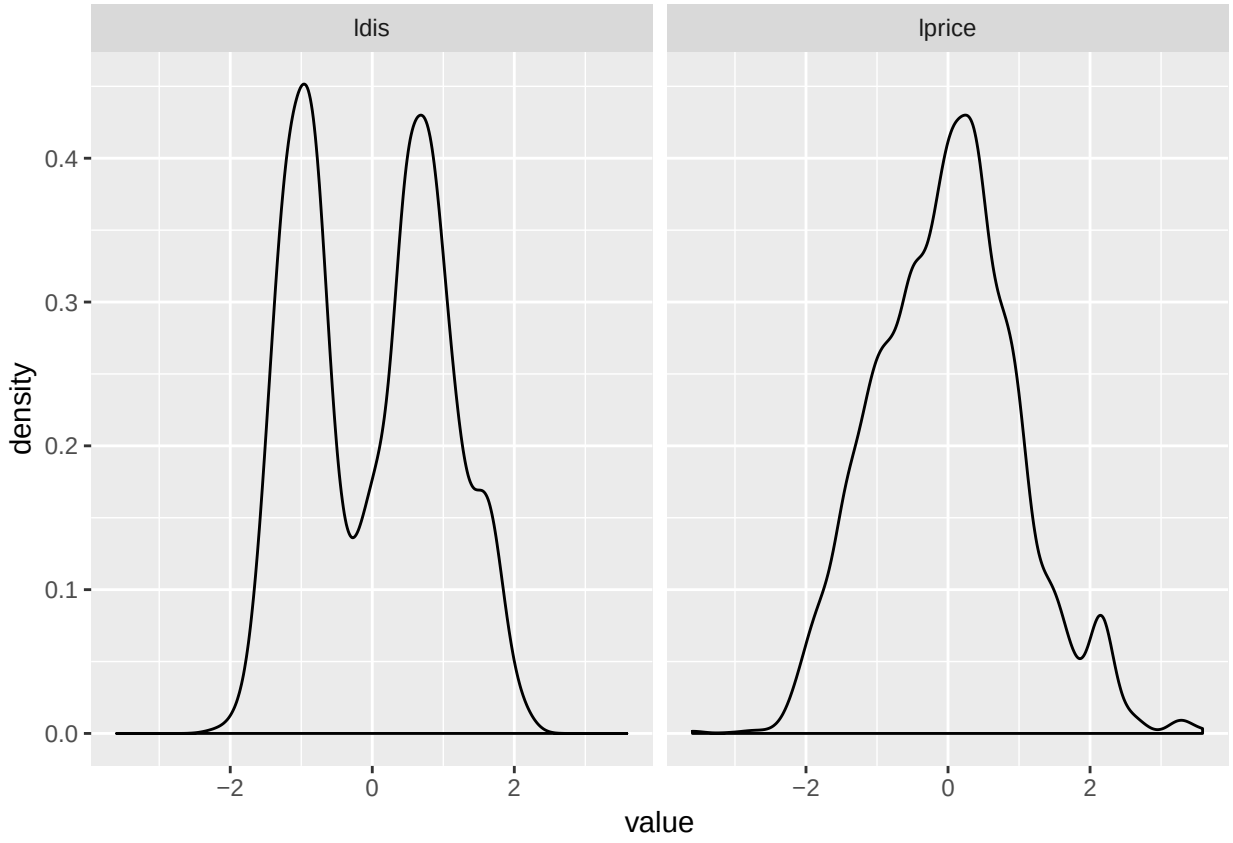




Checking correlation again, correlation is eliminated.

We are concerned about the output of our model influenced by skewness in some variables, so we check the distribution of all variables in the dataset. We found “price” and “dis_attr” are both quite skewed, thus decide to make some transformation to avoid negative impact on the output. Taking log is our decision. The result is also demonstrated in the graph below.


dis_attr
price
density
0.0
0.3
0.6
0.9
0.0
2.5
5.0
7.5
10.0
value


ldis
lprice
density
0.0
0.1
0.2
0.3
0.4
−2
0
2
value

Extrac features from review

To extract features from cutomer reviews for our models, we based on LDA topic model to extract hiden topics inside these reviews. Then, based on the porportion of the each topic (gamma in lda model) in reviews about each listings, we vectorlized the text variable, **Comments**, to 4 review features.

Because of the size of the reviews and time consuming processing, we processed the text mining part sepratedly, and only load the calculated data here. The detailed codes and results are in "lib/Text mining for Airbnb reviews.R".

```
#Load review feature data
load("../output/ReviewFeature.RData")
airbnb.df <- inner_join(airbnb.df, review.feature, "id")

#Final Data
airbnb.df$id <- NULL
```

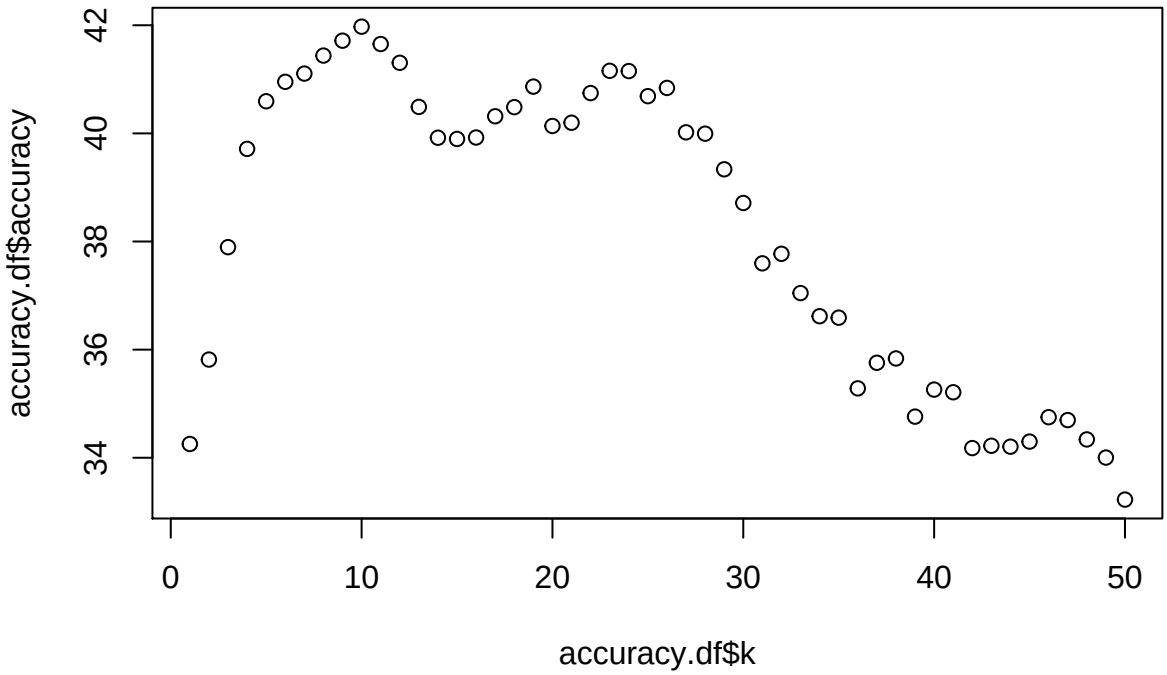
For the purpose of training our model better and evaluating model performance, we divide our dataset into training set(70%) and validataion set(30%). In addition, we transform categorical variables into dummy variables and scale these variables so as to satisfy some model assumptions.

```
#Data Partitioning
set.seed(1948)
train <- sample(1:nrow(airbnb.df), 0.7*nrow(airbnb.df))
airbnb.tra <- airbnb.df[train,]
airbnb.va <- airbnb.df[-train,]

#Covert categorical to dummy
airbnb.df.num <- dummy_columns(airbnb.df,
                           c("host_response_time","property_type", "room_type"),
                           remove_most_frequent_dummy = TRUE)
airbnb.df.num$host_response_time <- NULL
airbnb.df.num$property_type <- NULL
airbnb.df.num$room_type <- NULL

#Scale
airbnb.df.num <- cbind(as.data.frame(sapply(airbnb.df.num[,-(21:29)] %>% select(-OCC), scale)),
                       airbnb.df.num %>% select(21:29, OCC))
airbnb.tra.num <- airbnb.df.num[train,]
airbnb.va.num <- airbnb.df.num[-train,]
```

Model Selection

Model Knn

First, we apply k-nearest-neighbor model to have a glimpse on the data we finally got after all these processes, because knn is a nonparametric method that does not involve estimation of parameters in an assumed function form but draws information from similarities between the predictor values of the records in the dataset.



Model Regression

We want to predict the average occupations in 90 days, so in this part we use the regression method – "Elastic Net Regression". Compared to "Simple Linear Regression", "Ridge Rregression" and "LASSO", "Elastic Net" is the combination of Ridge and LASSO, and it collects the advantages of Ridge and LASSO but avoids their weakness. "Elastic Net" really has a great performance under multicollinearity and it has great model selection capability. Since there are nearly 30 variables, we need a model that obtains a high model selection capability.

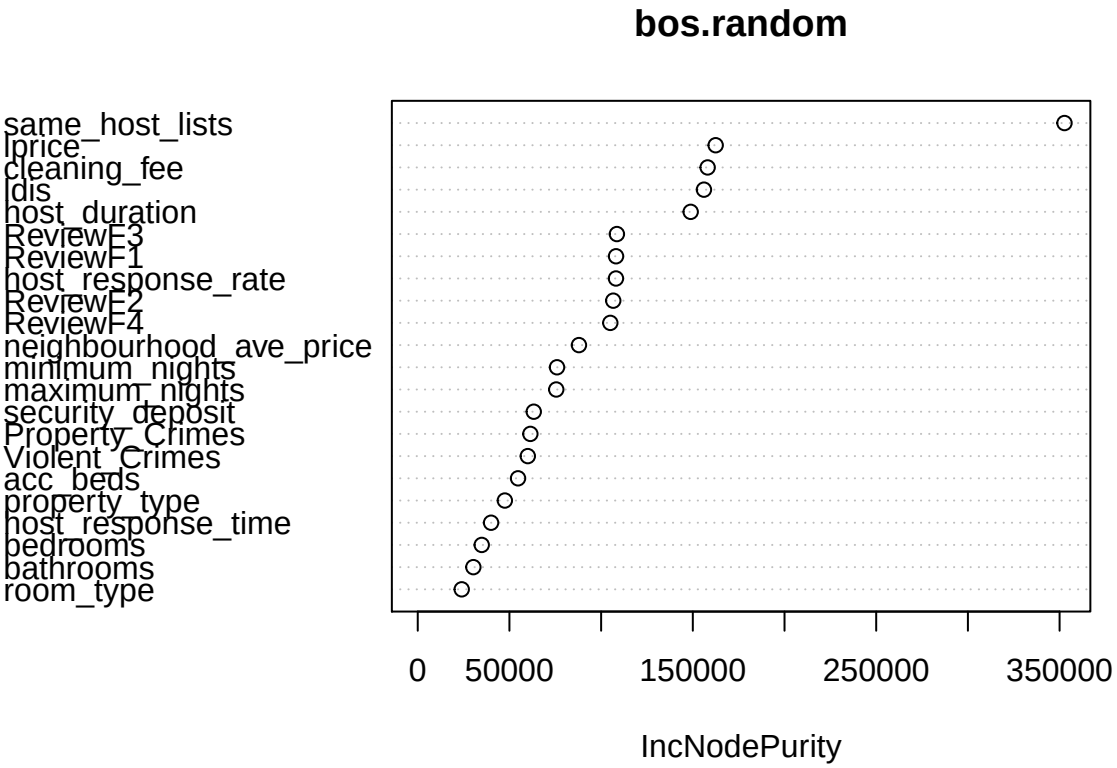
In "Elastic Net" model, we define "alpha" as the ratio of L1-penalty(LASSO) and set 10 cross validation folds to find the best alpha which produces the smallest MSE, and then we use this alpha to do the regression to get the coefficients from the results.

```
trainX <- as.matrix(airbnb.tra.num %>% select(-OCC))
trainY <- airbnb.tra.num$OCC

testX <- as.matrix(airbnb.va.num %>% select(-OCC))
testY <- airbnb.va.num$OCC

# ELASTIC NET WITH 0 < ALPHA < 1
a <- seq(0.1, 0.9, 0.05)
search <- foreach(i = a, .combine = rbind) %dopar% {
  cv <- cv.glmnet(trainX, trainY, family = "gaussian",
                  nfold = 10, type.measure = "mse",
                  paralle = TRUE, alpha = i)
  data.frame(cvm = cv$cvm[cv$lambda == cv$lambda.min],
             lambda.min = cv$lambda.min, alpha = i)
}
cv3 <- search[search$cvm == min(search$cvm), ]
md3 <- glmnet(trainX, trainY, family = "gaussian", lambda = cv3$lambda.min,
              alpha = cv3$alpha)
```

```
coefs <- as.data.frame(as.matrix(coef(md3)))
names(coefs) <- c("Coefficient")


RMSE.va[2] <- sqrt(mean((predict(md3, testX, type = "response")-testY)^2))
RMSE.tr[2] <- sqrt(mean((predict(md3, trainX, type = "response")-trainY)^2))
```

Model Random forest

In order to predict numerica outcomes, we use Random Forest model to discover patterns. Random Forest model outperforms simple regression tree by combining results from multiple trees to achieve stable model results. Balancing between interpretability of a single tree from regression tree and "variable importance" scores from Random Forest model, we choose the latter one because our goal is to give reference on improving revenue through key factors. We calculate RMSE in validation data and also try to find the best model based on OOB error. But the "best" model has even higher RMSE in validation sample because of overfitting.



As the important variables plot shows, the same_host_lists has highest score. The larger this variable, the more likely the property is for commercial purposes, which makes sense because business modes of commercial and non-commercial properties have different attraction to people. Other important inputs include log form of price, cleaning fees, log form of distance to attractions and operating time since started. Specific explanations are described in interpretation of model results. ###Model Selection

	RMSE.in.Training.set	RMSE.in.Testing.set
KNN	NA	34.25494
Regression	25.517291	26.00202
Random Forest	9.620905	22.80145

RMSE table is given above. All three of our models give undesirable prediction due to pretty high RMSE.

RMSE of Elastic Net Regression is larger than that of Random Forest but they are pretty close. Given the fact that Random Forest only gives important scores of each input but lack the accurate measurement of how availability in 90 days would change according to change in other inputs, Elastic Net Regression is better. Coefficients of Elastic Net Regression provide us with numeric and clear estimate about relationship of inputs and availability in 90 days, but we might not be able to give reliable suggestion on revenue improvement of hosts due to high RMSE.

Considering KNN model, it gives vague patterns among data and inaccurate predictions in validation sample given its high RMSE. In addition, KNN works best when k=1, implying model only uses 1 the nearst record to vote for projection. This causes high risks for misclassification so we drop KNN model due to high RMSE and just use it as our benchmark.

Results interpretation(insights)

As mentioned earily, all our 3 models gives undesirable prediction performance in validation set. The chosed model, Elastic Net gets an RMSE eauqls to 26 which means it is not very proper to use the model as revenue prediction tool on this dataset, but we can also get useful insights from the results.

	Coefficient
(Intercept)	54.6110304
host_response_rate	0.0000000
bathrooms	-0.3852380
bedrooms	1.3808613
security_deposit	0.2753479
cleaning_fee	-4.2533237
minimum_nights	0.0462308
maximum_nights	-2.7336319
same_host_lists	-9.9515996
host_duration	0.7928468
neighbourhood_ave_price	0.9471519
Violent_Crimes	0.0000000
Property_Crimes	0.0000000
acc_beds	0.0000000
lprice	-3.8126176
ldis	-1.4161755
ReviewF1	0.0000000
ReviewF2	-0.5957799
ReviewF3	0.0000000
ReviewF4	0.0000000
host_response_time_within a day	3.2778031
host_response_time_within a few hours	-1.3943250
host_response_time_a few days or more	0.0000000
property_type_Other	-1.2905986
property_type_Condo	2.7299216
property_type_Serviced apartment	-15.9032003
property_type_House	-2.7339644
room_type_Private room	-8.0881291
room_type_Shared room	-0.9549367

First, some variables such as “crime rates”, “accomodates and beds” and most features in “customer comments” are eliminated from this model, which means that these variables are unimportant or irrelavent for Boston

airbnb listing revenue.

Second, we find that **same_host_lists**, **cleaning_fee** and **lprice** affects the airbnb hosts' revenue negatively most. While the negative relationship between price and occupation is common sense, it is surprisingly that cleaning fee has a more negative impact on average occupation than price level. This may indicates airbnb customers are more repelled by high cleaning fees than high prices. Also the commercialized level indicating by **same_host_lists** negatively affects the listings' revenue, so the host relationship effect, as mentioned early, govern in Boston airbnb listings. In addition, we also find some other negative factors whose affections are less than the above outstanding factors, such as "distance to nearest boston attractions" and "maximum nights".

Third, we find some obvious positive factors such as **bedrooms**, **host_duration** and **neighbourhood_ave_price**. While the revenue increasing as the host duration and host experience increasing is common sense, the strong positive coefficient in # of bedrooms shows that airbnb customers prefer more private bedrooms in Boston area. Also high price neighbourhood is usually easier to generate high occupancy rate, together with relative high price, thus higher overall revenue.

Finally, there is a bunch of insights from the coefficients of dummy variables associated with property type, room type and response time. As for room type, the pirvate room with shared places is the less attractive type. As for property types, the serviced appartment shows huge disadvantages in airbnb revenue. Since most serviced appartment by definition are commercial owned, this findings verified and lines up with the negative revenue effect of commercialized airbnb found above. Also, "Condominium" type shows advantegies in overall revenue, which means that guests are more likely to choose the Condominium apartments.

Conclusion

Although the output of our models are not ideal and we are unable to offer a perfect model to predict average occupancy and revenue for Airbnb listers, our work still sheds light on what an Airbnb lister can do to improve the business value of their Airbnb listing property. Here are our advice:

1. Commercialization hurts potential revenue for Airbnb listers. We would recommend any one who wants to join Airbnb to be an individual lister, which might be more popular as people prefer to be familiar with a real-world warm host rather than hotels.
2. Cleaning fee seems to be awful. People feel uncomfortable with expensive cleaning fees, so we suggest listers include cleaning fee in the price. When setting price per night, listers can just consider bidding higher with the effect of cleaning fee, but set the cleaning fee to 0.
3. Reviews are not important as what we originally thought. Listers should pay more attention on how to improve their service rather than sitting there worrying about a few negative reviews.