Kaggle Report

# import datasets

train= read.csv("~/Desktop/data/analysisData.csv")  
test= read.csv("~/Desktop/data/scoringData.csv")

# observe the summary of train dataset and check the number of levels of all factors in train to find out unnecessary variables. Some of them are text description of the house, and some of them only contain NAs or only one level.

summary(train)

sapply(train[,sapply(train, is.factor)], nlevels)

**#delete those unnecessary variables**

library(dplyr)

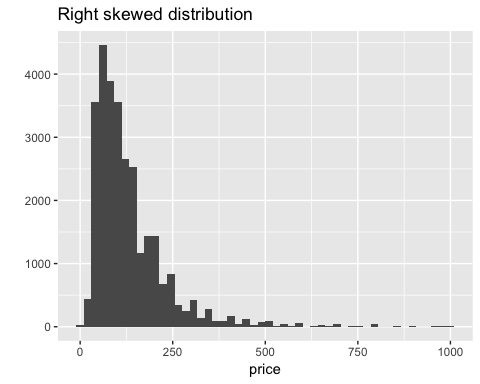
train<-select(train, -listing\_url,-scrape\_id,-last\_scraped,-name, -summary,-space,-description,-experiences\_offered,-neighborhood\_overview,-notes,-transit,-access,-interaction,-house\_rules,-thumbnail\_url, -medium\_url,-picture\_url,-xl\_picture\_url,-host\_id,-host\_url,-host\_name,-host\_since,-host\_about,-host\_thumbnail\_url,-host\_picture\_url,-country\_code,-country,-amenities,  
 -license,-jurisdiction\_names, -host\_acceptance\_rate, -requires\_license, -has\_availability)  
   
 test<-select(test, -listing\_url,-scrape\_id,-last\_scraped,-name, -summary,-space,-description,-experiences\_offered,-neighborhood\_overview,-notes,-transit,-access,-interaction,-house\_rules,-thumbnail\_url, -medium\_url,-picture\_url,-xl\_picture\_url,-host\_id,-host\_url,-host\_name,-host\_since,-host\_about,-host\_thumbnail\_url,-host\_picture\_url,-country\_code,-country,-amenities,  
 -license,-jurisdiction\_names, -host\_acceptance\_rate, -requires\_license, -has\_availability)

**check the str of test data and find zipcode is int in test, should be transferred to factor.And then combine train and test.**

test$zipcode<-as.factor(test$zipcode)  
test$price = NA  
all = rbind(train,test)

**exploring important variables. First, figure out the distribution of price.**

library(ggplot2)  
qplot(price, data = all[!is.na(all$price),], bins = 50, main = "Right skewed distribution")



**#check the missing value.zipcode:93, beds:19, square\_feet:35977,weekly\_price: 31644, monthly\_price:36134, security\_deposit: 14715, cleaning\_fee: 7104, reviews\_per\_month: 1.Find square\_feet, weekly\_price monthly\_price have 90% of the data as NA.**

colSums(is.na(all))

**#impute the missing data separately. NAs in zipcode and security\_deposit are replaced by the most frequently occured level. #Replace NAs in reviews\_per\_month, cleaning\_fee, weekly\_price, monthly\_price and square\_feet with median #Replace NAs in beds with 0**

sort(table(all$zipcode),decreasing = T)[1:3]

##   
## 11211 11221 11206   
## 1820 1400 1174

all$zipcode[is.na(all$zipcode)]="11211"  
sort(table(all$security\_deposit),decreasing = T)[1:3]

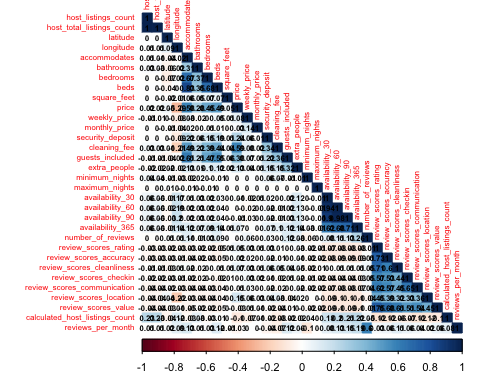
##   
## 0 100 200   
## 6196 3564 2582

all$security\_deposit[is.na(all$security\_deposit)]=0  
all$reviews\_per\_month[is.na(all$reviews\_per\_month)]<-median(all$reviews\_per\_month,na.rm=T)  
all$cleaning\_fee[is.na(all$cleaning\_fee)]<-median(all$cleaning\_fee,na.rm=T)  
all$weekly\_price[is.na(all$weekly\_price)]<-median(all$weekly\_price,na.rm=T)  
all$monthly\_price[is.na(all$monthly\_price)]<-median(all$monthly\_price,na.rm=T)  
all$square\_feet[is.na(all$square\_feet)]<-median(all$square\_feet,na.rm=T)  
all$beds[is.na(all$beds)]<-0

**#explore correlation of variable with price**

library(corrplot)

numericVars <- which(sapply(all, is.numeric))  
all\_numVar <- all[, numericVars]  
cor\_numVar <- cor(all\_numVar, use="pairwise.complete.obs")  
corrplot(cor(all\_numVar[!is.na(all$price),-1]),method = 'color',type = 'lower',addCoef.col = 'black',diag = T, tl.cex = .5,cl.cex = .7, number.cex=.5)



**Delete highly correlated variables**

all<-select(all,-availability\_90,-availability\_60,-review\_scores\_value,-review\_scores\_communication,-review\_scores\_checkin,-review\_scores\_accuracy)

#divide all data to train and test

train<-all[!is.na(all$price),]  
 test<-all[is.na(all$price),]

**#I tried random forest here and found some of the levels are larger than 53 which prevent random forest from working. forest<-randomForest(price~.-id,data=train,ntree=1000)**

**#check the number of levels of factors, and delete the variables with more than 53 levels.**

sapply(all[,sapply(all, is.factor)], nlevels)

all<-select(all,-host\_location,-host\_response\_rate,-host\_neighbourhood,-host\_verifications,-street,-neighbourhood,-neighbourhood\_cleansed,-city,-zipcode,-smart\_location,-calendar\_updated,-first\_review,-last\_review)

**#first move out id, and do feature selection.**

all\_noid<-select(all,-id)  
start\_mod = lm(price~1,data = all\_noid)  
empty\_mod = lm(price~1,data = all\_noid)  
full\_mod = lm(price~.,data = all\_noid)  
hybridStepwise = step(start\_mod,scope = list(upper=full\_mod,lower = empty\_mod),direction = "both")

**#run random forest to find variable importance**

library(randomForest)

forest = randomForest(price~cleaning\_fee + accommodates + neighbourhood\_group\_cleansed +

room\_type + bathrooms + longitude + bedrooms + property\_type +

availability\_30 + review\_scores\_location + reviews\_per\_month +

latitude + minimum\_nights + host\_response\_time + availability\_365 +

review\_scores\_cleanliness + beds + host\_is\_superhost + guests\_included +

weekly\_price + cancellation\_policy + is\_business\_travel\_ready +

extra\_people + number\_of\_reviews + host\_identity\_verified +

calculated\_host\_listings\_count + square\_feet + host\_listings\_count +

review\_scores\_rating + host\_has\_profile\_pic

,data=train,ntree = 1000)

```

**#find the importance of variables and delete the less important variables**

varImpPlot(forest)

importance(forest)

**#linear regression (one of the submissions)**

model = lm(price~cleaning\_fee + accommodates + neighbourhood\_group\_cleansed + room\_type + bathrooms + longitude + bedrooms + property\_type +availability\_30 + review\_scores\_location + reviews\_per\_month + latitude + minimum\_nights + host\_response\_time + availability\_365 +review\_scores\_cleanliness + beds + host\_is\_superhost + guests\_included +weekly\_price + cancellation\_policy + is\_business\_travel\_ready +extra\_people + number\_of\_reviews + host\_identity\_verified + calculated\_host\_listings\_count + host\_listings\_count + review\_scores\_rating,data=train)

**#property\_type has new level, replace the new levels with the existing ones**

test1<-all[is.na(all$price),]

levels(test1$property\_type)[levels(test1$property\_type)=="Cottage"] <- "Apartment"

levels(test1$property\_type)[levels(test1$property\_type)=="Hut"] <- "Apartment"

**#random forest (The one with lowest RMSE)**

```{r}

library(randomForest)

forest = randomForest(price~cleaning\_fee + accommodates + neighbourhood\_group\_cleansed +

room\_type + bathrooms + longitude + bedrooms + property\_type +

availability\_30 + review\_scores\_location + reviews\_per\_month +

latitude + minimum\_nights + host\_response\_time + availability\_365 +

review\_scores\_cleanliness + beds + host\_is\_superhost + guests\_included +

weekly\_price + cancellation\_policy + is\_business\_travel\_ready +

extra\_people + number\_of\_reviews + host\_identity\_verified +

calculated\_host\_listings\_count + host\_listings\_count +

review\_scores\_rating,data=train,ntree = 1000)