****

**Competition Project Report**

Professor Courtney Paulson

BUDT758T: Data Mining and Predictive Analytics

By:

Jiaying Lu / Junyuan Ma / Yuran Wang / Yifan Dang

5/11/2018

**Executive Summary**

This report is prepared to use collected data on the booking rate to predict which listings have the great potential to be identified with high booking rate. This information is designed to help Airbnb increase its revenues by improving their overall booking rate.

We are trying to help Airbnb in two ways. One is figuring out the optimal prediction model with the highest accuracy that can be used to predict which Airbnb will be most likely booked by customers, the other is to evaluate the importance and determine the correlation between high booking rate and each variable through an interpretation model. Both information will lead to two purposes. The first one is to increase the exposure rate of potential high-booking-rate homes. Those highly popular homes will be on the hot searching listings on Airbnb’s official website. Additionally, Airbnb can recommend these appealing listings to customers by putting them in a front position of the customers’ searching pages which can increase the matching ratio between customers and hosts. The second purpose is to find out the listings with low booking rate and then make use of other information to give these hosts some suggestions to improve their listings’ booking rate.

In order to understand the relationship between factors and booking rate, we have to find a model which is good for interpretation. After understanding the relationships between factors and booking rate, we can provide feasible solutions to hosts on how to improve booking condition, and move forward to boost the overall booking rate in Airbnb.

Overall, we run several models which include logistic regression, Ridge, Lasso, classification tree, SVM, random forests and boosting. It turns out that the optimal prediction model is boosting model with the highest accuracy of 85.026% for the testing dataset. And we chose logistic regression as our interpretation model since it explains the information we need more detailed.

After analyzing this data through a logistic model, we found quite a few factors which are significant variables related to high booking rate. In general, we found that listings with the lower price, lower cleaning fee, lower minimum\_nights are more likely to have high booking rates. Listings in big cities such as Boston, New York City and Washington, DC are more highly booked than other cities on average. In order to increase booking rate, hosts may better become a superhost, reply to potential clients as much as they can and collect more verified information. Listings with amenities which bring guests more convenience and provide more comfort utilities may also help to increase the booking rate. For example, heater, air conditioning, and hot water make guests comfortable during their stay. Self-check and shampoo give travelers convenience.

Therefore, our recommendation is as follows:

* Use logistic regression to detect the relation between each variable and high\_booking\_rate. Give suggestions according to the relationship between high book rate and factors to hosts who do not have high booking rate.
* Use the boosting model to predict which listing would be a high booking rate home and rearrange the listing’s priority in search results.
* If two models have any inconsistent prediction, use the boosting model as the final decision. Meanwhile, check the variables in the logistic model to see if there are enough variables to explain the book rate and whether the variables are unbiased.

**Exploring Data and Feature Selection**

A total of 69 variables were included in the training dataset, some of which were highly correlated. For example, the content of variable “city” and “city\_name” are quite close except that “city” sometimes referred to a more specific district and more customized. There were also several variables that were describing locations. In these cases, we only chose variables that were more complete and more standardized in formats among similar variables.

The dataset also contained several text variables such as “amenities” and “transit” and “description”. Since customers were open to enter everything they like into these variables, it is tricky to handle these variables. We believed that when customers are searching for rooms, they must care about amenities. As “amenities” came in a relatively standardized format, we constructed a document-term matrix which included amenities with sparsity over 5%. Any observations with a non-blank “host\_verifications” were regarded as having a verification. The same method was also applied to “house\_rules”. We divided the variable “access” into three levels “Entire”, “Partial” and “Unknown”. If the text contains certain words, we assign it to “Entire” or “Partial”, otherwise it is “Unknown”. Text variables other than the above four were deleted.

For the “host\_response\_rate”, we filled with the mean of the existed numbers which is 0.96. And we inserted the blank of “host\_response\_time” with the mode “within an hour”.

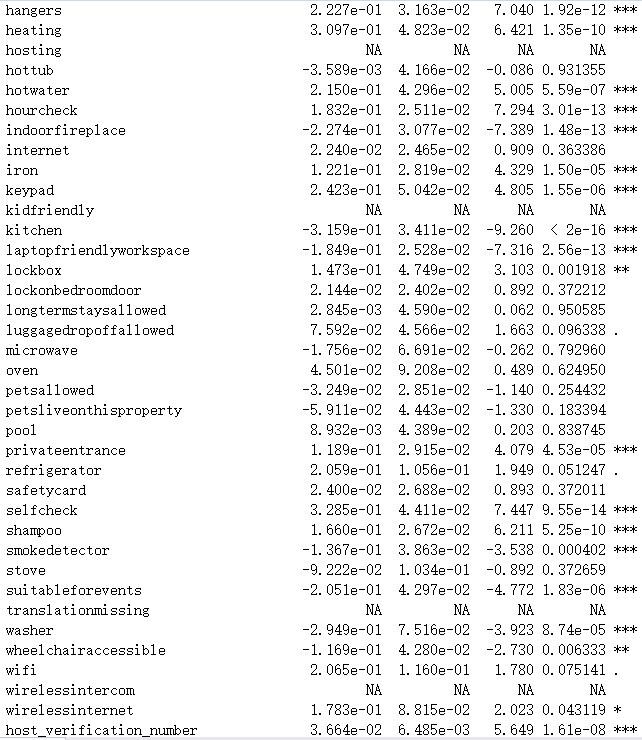
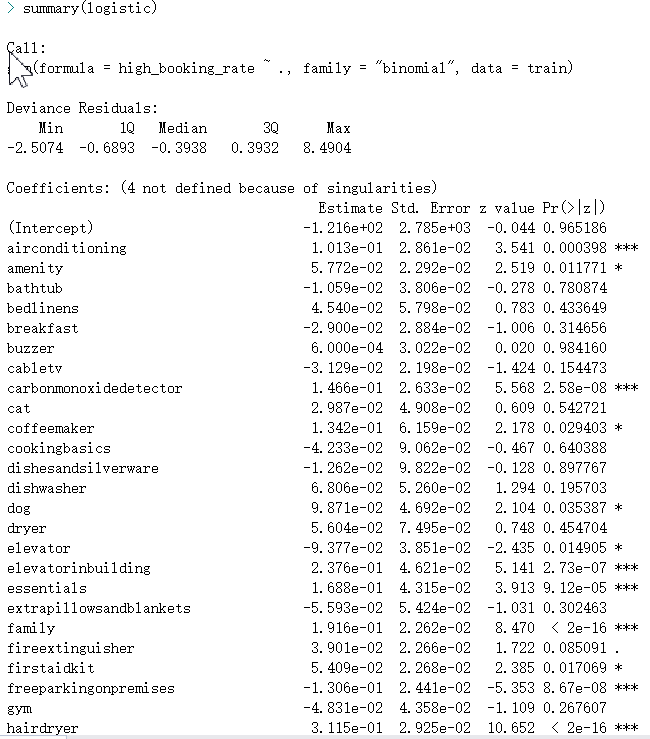
**Model Evaluation**

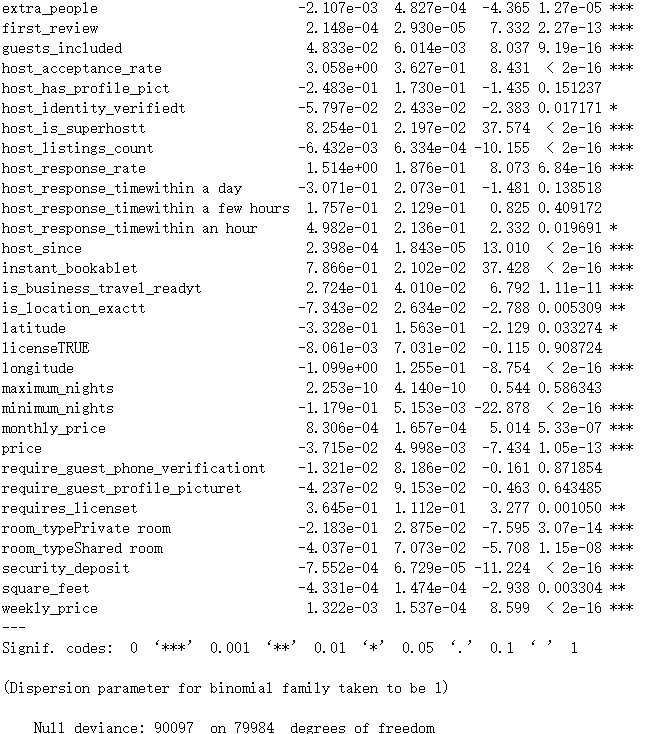
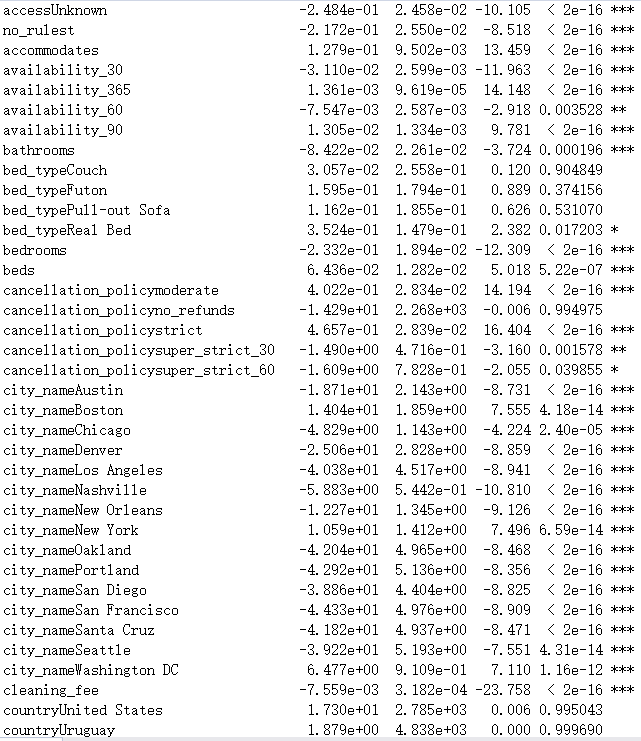
The goal is to choose the best classification method to predict whether or not an Airbnb listing will be a high booking rate (1 represents high\_booking\_rate) and to find out an interpretation model that could help us give suggestions to hosts. We combined the given train independent variables with given dependent variable. We set the seed to 12345 and randomly partitioned the given train data into 30% validation data and 70% remaining(training) data. The baseline we used was 75.11%, which is the percentage of the majority group (Not High Booking Rate) in the validation data. We used a cutoff of 0.5 for prediction classification and calculated the accuracy of all models on the validation data and compare them with the baseline. If the model's accuracy was higher than the baseline, it would a good model for our further model selection. Among those models with a higher accuracy than baseline, we chose the one with the highest accuracy on the validation data for interpretation purpose. Due to the factor that some models are hard to understand and interpret, we also chose one method with more detailed information, which is able to best interpret the relationships between features and booking rate. That is, we can get the exact coefficient not only for numerical variables but also for each level of categorical factors.

**Model Details**

* Logistic

Our first model is the logistic regression. Our initial attempt on logistic regression did not include any variable derived from text mining. The model reached an accuracy of 78.6% for the validation data. After we added the matrix of amenities, the accuracy of predicting the validation data increased to 79.6%. This confirmed our expectation that amenities is a major concern when people book their rooms. However as we added other variables derived from text mining, we didn’t see an increase in accuracy.

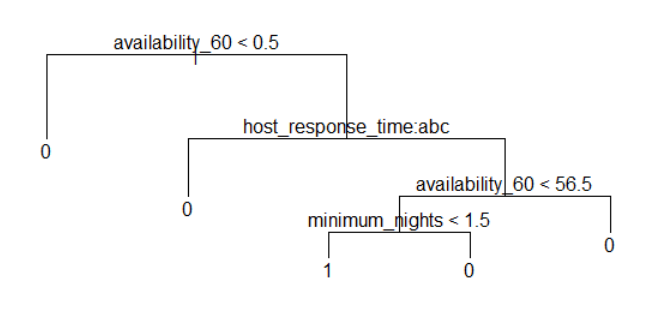




From the summary, we can see that customers more care about the convenience and comfort of a room. Room seekers tend to book rooms which have temperature control, elevator and can check in quickly. As we expected, variable “country” is not significant because almost all observations are in the US. Customers also don’t seem to care about hosts’ license and profile picture. Though they don’t quite care about the maximum nights, they do need to consider the minimum nights required to book a room. Logistic model also suggests that the most popular room\_type is the entire home. If a host becomes a super host and does more verifications on the website, their effort will more likely to get payback. All these information are critically for us to help with those low booking rate listings to improve.

* Classification Tree

We applied the classification tree as well. Our tree is shown as below:



Our prediction accuracy for decision tree is 76.97%. However, after we run the model, the tree only selects 3 variables, “availability\_60”, “host\_repsonse\_time”, “minimum\_nights” and eliminates all the other variables. According to the dendrogram, any Airbnb availability less than half day will predict the book rate as 0, any Airbnb availability more than half and has a certain amount of host\_reponse \_time will predict high booking rate as 0. For availability\_60 less than 56.5 and minimum nights less than 1.5 will predict high booking rate as 1. And other will be 0. And any availability larger than 56.5 will predict high booking rate as 0. One potential reason for the model eliminates a lot of variables is because most of the variables are highly correlated with each other, and decision tree recognized the highly involved interaction of different variables so that it deletes the correlated one but leave the most significant one. However, since this model eliminates most of the variables, it ends up with very low accuracy so that we decide to not use decision tree anyway.

* Ridge (additional model output is in appendix)

We also applied a ridge regression with training data if there is high correlation variables in our dataset. Opposed to LASSO regression, Ridge wouldn’t take away any variables from the model. The significant coefficients are those separate far from zero, so we chose those variables whose absolute coefficients’ value are above 0.3. Accordingly, we draw a conclusion that the significant variables were elevator building, heating, host\_acceptance\_rate, host\_is\_superhost, host\_response\_time, instant\_bookable, and requires\_license.

According to the output, we found that an increase in host\_acceptance\_rate would increase the probability of high booking rate. In addition, compared with the listings without elevator and heating, the listings have these amenities tend to have high book rate. And compared with the listings don’t require licenses, the listings require license has less probability of having high book rate. What’s more, the probability of having high book rate would also increase owing to a super host, a shorter response time, or instant bookable.

However, in order to run out the regression, we translated all variables into a numerical format which resulted in a difficulty in interpreting the categorical variables. What’s more, with this model, our team discovered an accuracy of only 78.65% on our validation data. Although this was higher than the validation baseline of 75.11%, it was much lower than boosting model we tested, so ultimately we do not recommend a Ridge model for this problem.

* LASSO (additional model output is in appendix)

LASSO performs both regularization and variable selection, which can improve the prediction accuracy and enhance interpretability of model. It is supposed to be a better model for interpretation purpose. We used the training data to train the LASSO model. We do not need a separate validation data split here since LASSO uses cross-validation for their model choices.

Any variable not be taken away can be considered significant in LASSO. According to the output, only breakfast, dryer, oven, maximum\_nights, minimum\_nights, and monthly\_price are excluded in this model. It indicates that these features are not useful when evaluating the importance and determining the correlation between high booking rate and each feature. Among the significant variables, country, host\_acceptance\_rate, host\_is\_superhost, host\_response\_rate, host\_response\_time, instant\_bookable, license, and reuqires\_license seem to have more influences on predicting a high booking rate listing, since their absolute values of the coefficient are higher than others.

It should be noted with this model, our team discovered an accuracy of only 78.75% on our validation data. Although this was higher than the validation baseline of 75.11%, the prediction accuracy of this model was lower than boosting model we tested. To run this model, we converted all the categorical variables into numerical. Therefore, this conversion affected our interpretation results on the final model. Thus, we do not recommend the LASSO model for both interpretation and prediction purpose.

* Random Forests (additional model output is in appendix)

Since the random forests can avoid overfitting and reduce variance compared to a classification tree, we also tried the random forests model for prediction purpose. We used the training data to train the model.

According to the output, by sorting variable importance from high to low, city\_name, minimun\_nighs, availability\_365, hosting\_listings\_count, and cleaning fee are the top 5 important variables for this model to predict whether an Airbnb listing will be a high booking rate. With this model, our team discovered an accuracy of only 83.66% on our validation data. Although this was higher than the validation baseline of 75.11%, it was slightly lower than the boosting model we tested. Additionally, since it is hard for this model to interpret the relationships between each variable and high booking rate, so ultimately we do not recommend a Random Forest model for this problem.

* Support Vector Machine (additional model output is in appendix)

We also tried Support Vector Machine since it works on both linear and non-linear classification that might improve the our prediction accuracy. We used the training data to train the Support Vector Machine model.

According to the output, based on the training data set, the SVM identified 26421 support vectors. By extracting variable weights from this SVM model, beds, host\_is\_superhost, cleaning\_fee, instant\_bookable, and maximum\_nights are the top 5 relatively important variables this model used to separate the data set since they have larger absolute vector weights. With this model, our team discovered an accuracy of only 81.16% on our validation data. Although this was higher than the validation baseline of 75.11%, it was lower than the boosting model we tested and as it is hard to interpret, so ultimately we do not recommend an SVM model for this problem.

* Boosting (additional model output is in appendix)

In order to increase the accuracy, we employed boosting with our training dataset. According to the output, the variables are listed with the relative influence. The longitude, availibity\_30, first\_review, latitude and minimum nights are the top five significant in predicting the high book rate listing. Conversely, the hosting, kid-friendly, translation missing, wireless intercom and country are uselessly indicated by this model.

With this model, our team discovered an accuracy of 85.08% on our validation data, which is the highest accuracy among all the model we used. Thus, we recommended a boosting model only for the purpose of prediction.

**Appendices**

* Group member roles

All group members run the traditional logistic regression for double check purpose.

Junyuan Ma: take charge of cleaning the whole dataset with different method. And responsible for running logistic and neural network models.

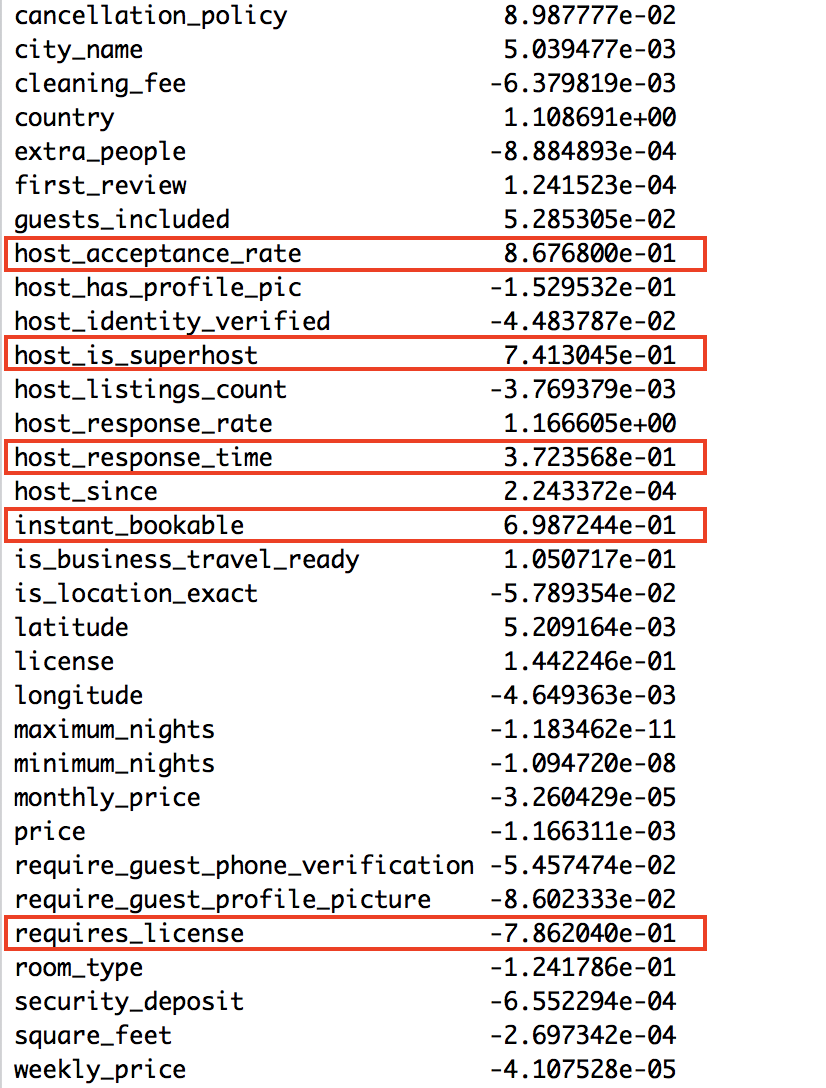
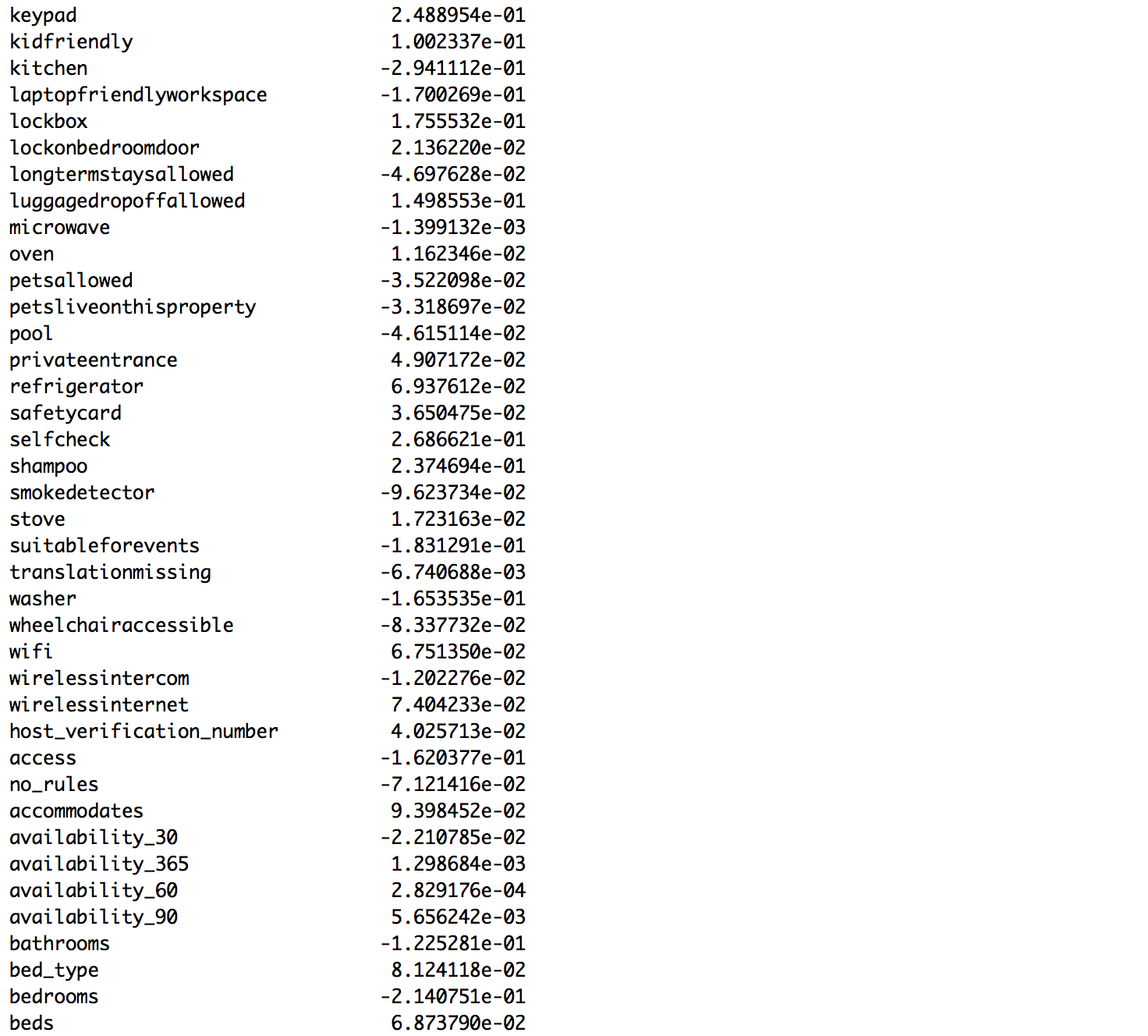
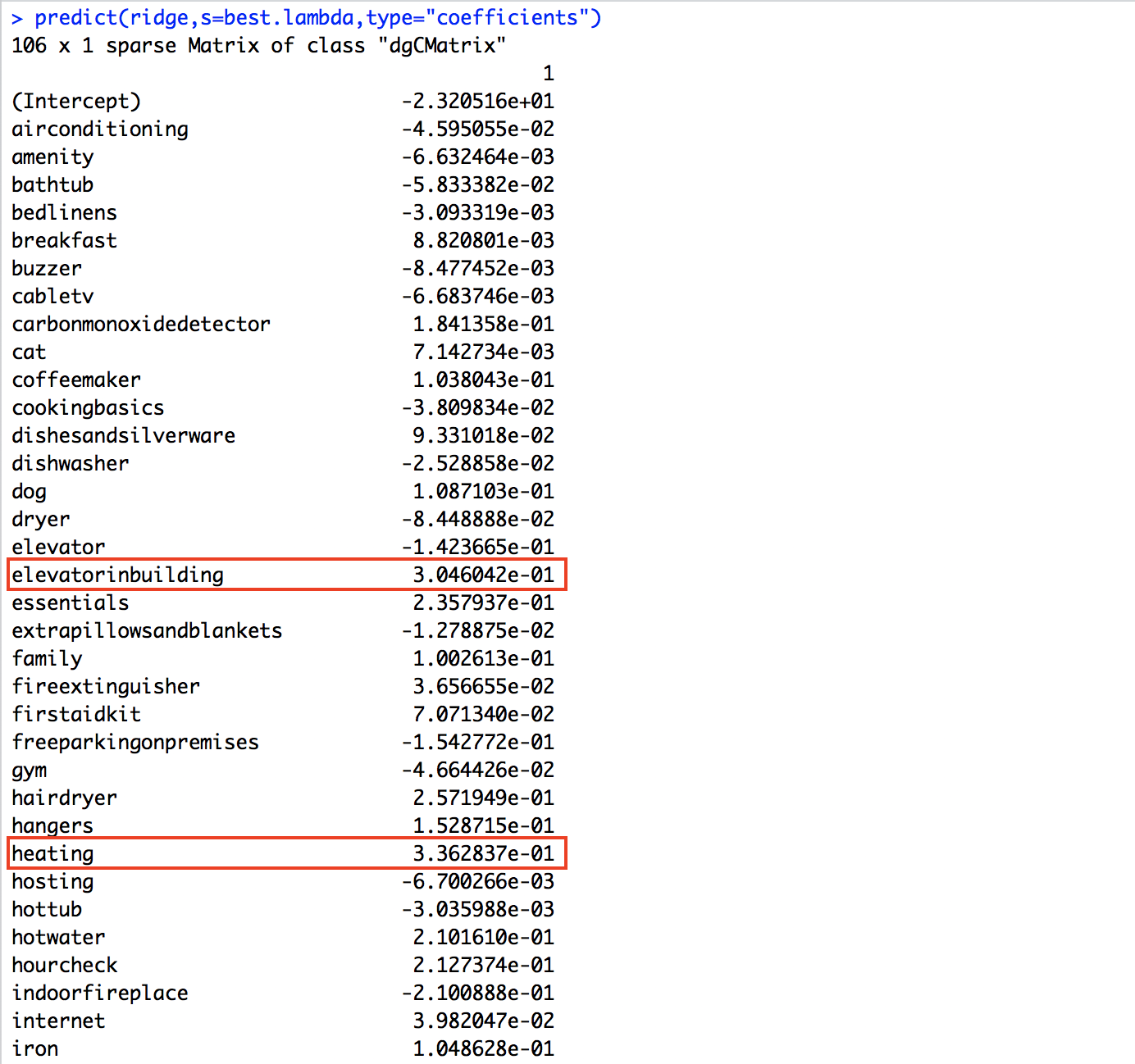
Jiaying Lu: take charge of cleaning the first 23 variables for the original version of dataset. And running the Random Forests, Support Vector Machine, and boosting models.

Yuran Wang: take charge of cleaning the middle 23 variables for the original version of dataset. And running the Ridge, LASSO, and boosting models.

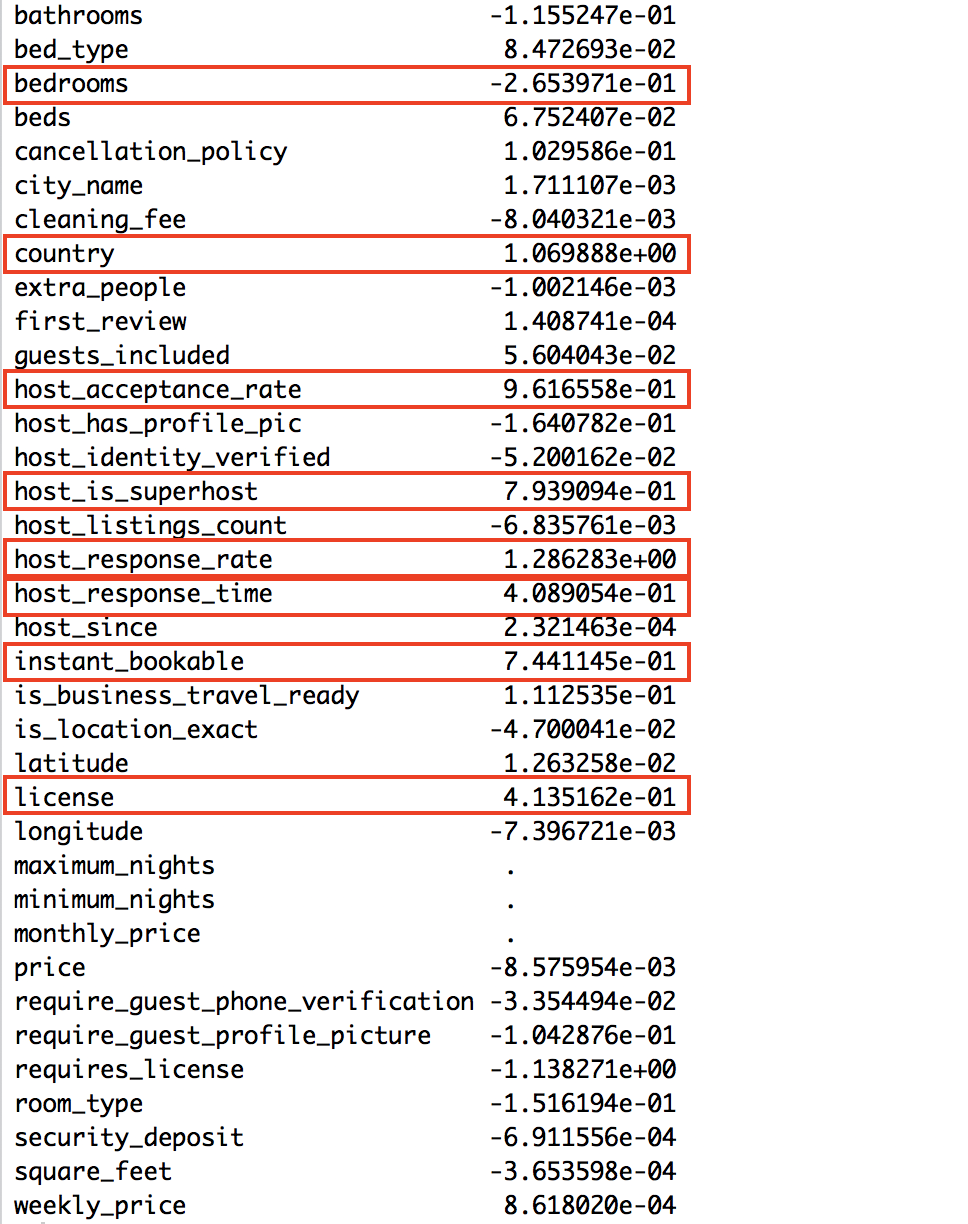
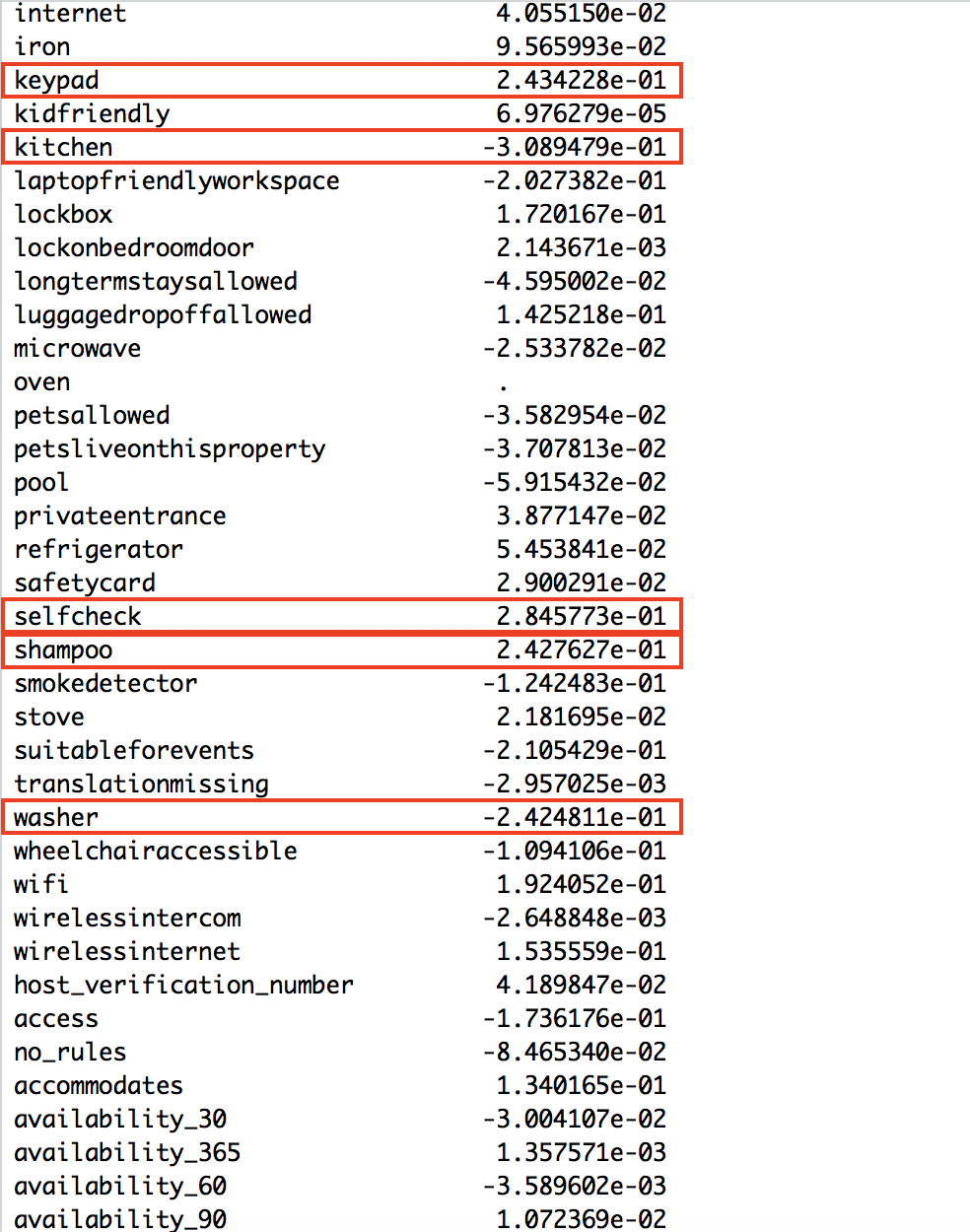
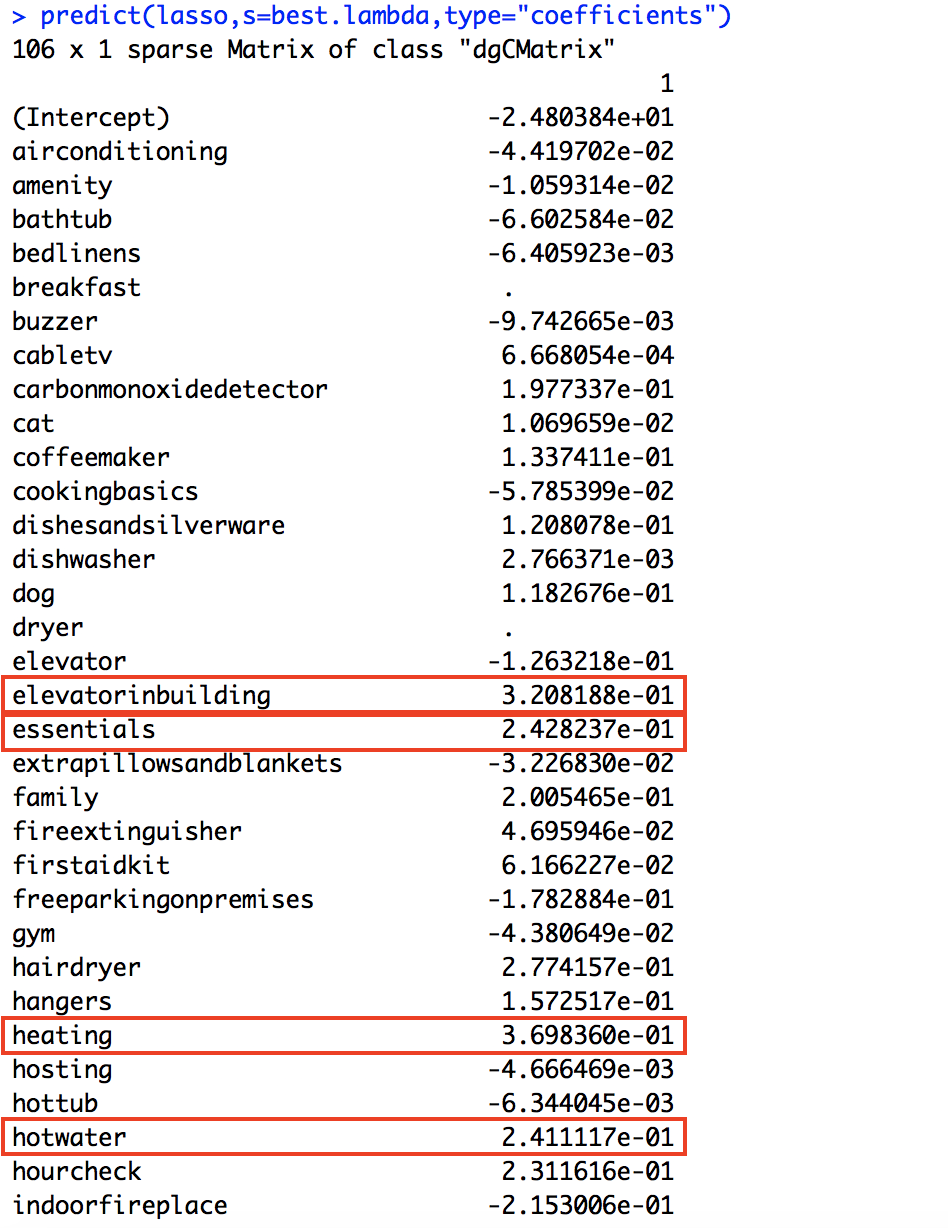
Yifan Dang: take charge of cleaning the last 23 variables for the original version of dataset. And running the Classification Tree, neural network, and boosting models.

* Outputs

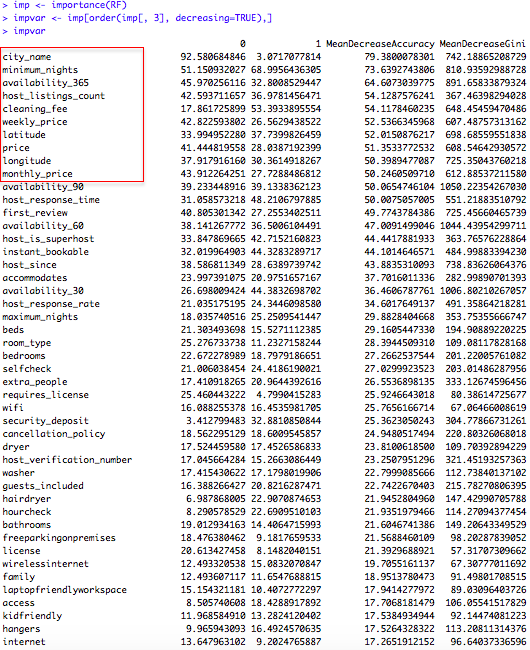
Ridge Details



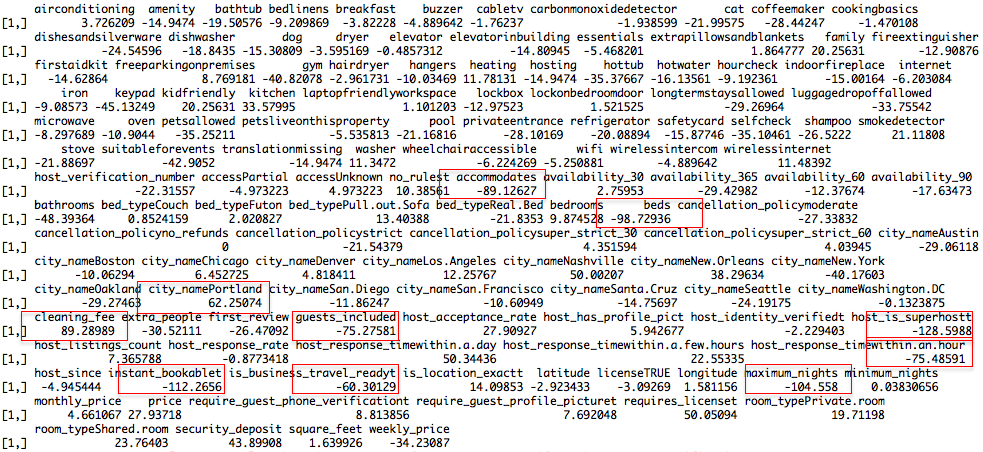
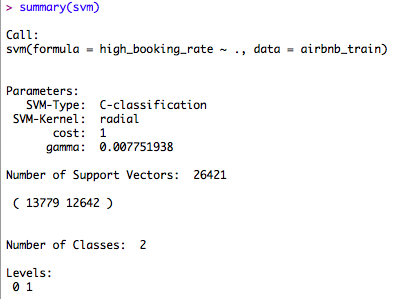
LASSO Details



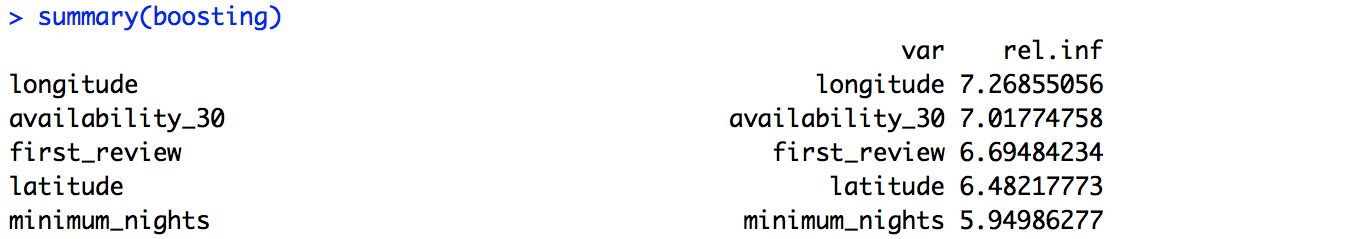
Random Forests Details (Partial Output)



SVM Details



Boosting Details (Partial Output)



* R Code (Partial)

######################### Logistic #######################

airbnb <-readRDS("~/Desktop/train.rds")

set.seed(12345)

validation\_instn <- sample(nrow(airbnb), 0.3 \* nrow(airbnb))

airbnb\_validation <- airbnb[validation\_instn, ]

airbnb\_train <- airbnb[-validation\_instn, ]

logistic <- glm(high\_booking\_rate ~ ., data = airbnb\_train, family = "binomial")

summary(logistic)

log\_pred\_valid <- predict(logistic, newdata=airbnb\_validation, type = "response")

log\_pre <- ifelse(log\_pred\_valid>0.5,1,0)

table(airbnb\_validation$high\_booking\_rate, log\_pre)

accuracy\_log\_valid = sum(ifelse(airbnb\_validation$high\_booking\_rate == log\_pre, 1, 0)) / nrow(airbnb\_validation)

######################### Ridge #######################

library(glmnet)

ridge <- glmnet(data.matrix(airbnb\_train[,c(1:105)]),airbnb\_train$high\_booking\_rate,

family="binomial",alpha=0)

ridge.cv <- cv.glmnet(data.matrix(airbnb\_train[,c(1:105)]),airbnb\_train$high\_booking\_rate,

family="binomial",alpha=0)

best.lambda <- ridge.cv$lambda.min

predict(ridge,s=best.lambda,type="coefficients")

ridge.pred <- predict(ridge, s=best.lambda,newx = data.matrix(airbnb\_validation[,c(1:105)]),

type="response")

class\_ridge <- ifelse(ridge.pred>0.5,1,0)

accuracy\_ridge <- sum(ifelse(airbnb\_validation$high\_booking\_rate==class\_ridge,1,0))/

nrow(airbnb\_validation)

######################### LASSO #######################

lasso <- glmnet(data.matrix(airbnb\_train[,c(1:105)]),airbnb\_train$high\_booking\_rate,family = "binomial",alpha = 1)

lasso.cv <- cv.glmnet(data.matrix(airbnb\_train[,c(1:105)]),airbnb\_train$high\_booking\_rate,

family="binomial",alpha=1)

best.lambda <- lasso.cv$lambda.min

predict(lasso,s=best.lambda,type="coefficients")

lasso.pred <- predict(lasso,s=best.lambda,newx=data.matrix(airbnb\_validation[,c(1:105)]),

type="response")

class\_lasso <- ifelse(lasso.pred>0.5,1,0)

accuracy\_lasso <- sum(ifelse(airbnb\_validation$high\_booking\_rate==class\_lasso,1,0))/

nrow(airbnb\_validation)

######################### Boosting #######################

airbnb <- data.matrix(airbnb)

airbnb <- data.frame(airbnb)

airbnb$high\_booking\_rate <- airbnb$high\_booking\_rate-1

airbnb$license <- as.factor(airbnb$license)

set.seed(12345)

validation\_instn <- sample(nrow(airbnb),0.3\*nrow(airbnb))

airbnb\_validation <- airbnb[validation\_instn,]

airbnb\_train <- airbnb[-validation\_instn,]

library(gbm)

boosting <- gbm(high\_booking\_rate~.,data=airbnb\_train,distribution = "bernoulli",n.tree=10000,

interaction.depth = 5,shrinkage = 0.05)

boosting\_pred <- predict(boosting,newdata=airbnb\_validation,type="response",n.tree=10000)

class\_boost <- ifelse(boosting\_pred>0.5,1,0)

accuracy\_boost <- sum(ifelse(airbnb\_validation$high\_booking\_rate==class\_boost,1,0))/nrow(airbnb\_validation)

summary(boosting)

##################### SVM ############################

airbnb <- readRDS("~/Desktop/train.rds")

test <- readRDS("~/Desktop/test1.rds")

airbnb <- airbnb[,-77]

test <- test[,-77]

common <- intersect(names(airbnb), names(test))

for (p in common) { if (class(airbnb[[p]]) == "factor") { levels(test[[p]]) <- levels(airbnb[[p]]) }}

set.seed(12345)

validation\_instn <- sample(nrow(airbnb), 0.3 \* nrow(airbnb))

airbnb\_validation <- airbnb[validation\_instn, ]

airbnb\_train <- airbnb[-validation\_instn, ]

library(e1071)

svm <- svm(high\_booking\_rate~. , data=airbnb\_train, scale = TRUE)

summary(svm)

w = t(svm$coefs) %\*% svm$SV

svm\_pred\_valid <- predict(svm, newdata=airbnb\_validation, type = "response")

table(airbnb\_validation$high\_booking\_rate, svm\_pred\_valid)

accuracy\_svm\_valid = sum(ifelse(airbnb\_validation$high\_booking\_rate == svm\_pred\_valid, 1, 0)) /nrow(airbnb\_validation)

##################### Random Forests ########################

airbnb <-readRDS("~/Desktop/train.rds")

test <-readRDS("~/Desktop/test1.rds")

common <- intersect(names(airbnb), names(test))

for (p in common) { if (class(airbnb[[p]]) == "factor") { levels(test[[p]]) <- levels(airbnb[[p]]) }}

set.seed(12345)

validation\_instn <- sample(nrow(airbnb), 0.3 \* nrow(airbnb))

airbnb\_validation <- airbnb[validation\_instn, ]

airbnb\_train <- airbnb[-validation\_instn, ]

library(randomForest)

RF <- randomForest (high\_booking\_rate~.,airbnb\_train,ntree=500,norm.votes=FALSE, do.trace=10, importance=TRUE)

summary(RF)

imp <- importance(RF)

impvar <- imp[order(imp[, 3], decreasing=TRUE),]

rf\_pred\_valid <- predict(RF, newdata=airbnb\_validation, type = "response")

table(airbnb\_validation$high\_booking\_rate, rf\_pred\_valid)

accuracy\_rf\_valid = sum(ifelse(airbnb\_validation$high\_booking\_rate == rf\_pred\_valid, 1, 0)) / nrow(airbnb\_validation)