# Boosted Tree Analysis

*March 11, 2019*

```r
library(tidyverse)
library(xgboost)
library(xgboostExplainer)
```

```r
getwd()
source('../02Analysis/dataLoad.R', chdir = T)
```

```r
#remove Hotel ID
Hotel_pred_df <- Hotel_pred_df[,-c(1)]
test <- test[,-c(1)]
train <- train[,-c(1)]

#Remove Hotel Country
Hotel_pred_df <- Hotel_pred_df[,-c(2)]
test <- test[,-c(2)]
train <- train[,-c(2)]


#factor other_cust
Hotel_pred_df$other_cust <- as.factor(Hotel_pred_df$other_cus)
test$other_cust <- as.factor(test$other_cus)
train$other_cust <- as.factor(train$other_cus)

#factor common score
Hotel_pred_df$CommonScore <- as.factor(Hotel_pred_df$CommonScore)
test$CommonScore <- as.factor(test$CommonScore)
train$CommonScore <- as.factor(train$CommonScore)

#factor tourists
Hotel_pred_df$tourists <- as.factor(Hotel_pred_df$tourists)
test$tourists <- as.factor(test$tourists)
train$tourists <- as.factor(train$tourists)

str(train)
```

```r
reviewer_region <- model.matrix( ~ Reviewer_sub_region - 1, data= train )
city <- model.matrix( ~ City - 1, data = train)
weather_summary <- model.matrix( ~ weather_summary - 1, data = train)
access_type <- model.matrix( ~ access_type - 1, data = train)
accessibility <- model.matrix( ~ accessibility - 1, data = train)
free <- model.matrix( ~ free - 1, data = train)
other_room_types <- model.matrix( ~ other_room_types - 1, data = train)
room_type <- model.matrix( ~ room_type - 1, data = train)
time_of_stay <- model.matrix( ~ time_of_stay - 1, data = train)
trip_type <- model.matrix( ~ trip_type - 1, data = train)
view_type <- model.matrix( ~ view_type - 1, data = train)
```

```r
access_type_cust <- model.matrix( ~ access_type_cust - 1, data = train)
accessibility_cust <- model.matrix( ~ accessibility_cust - 1, data = train)
free_cust <- model.matrix( ~ free_cust - 1, data = train)
other_cust <- model.matrix( ~ other_cust - 1, data = train)
other_room_types_cust <- model.matrix( ~ other_room_types_cust - 1, data = train)
room_type_cust <- model.matrix( ~ room_type_cust - 1, data = train)
time_of_stay_cust <- model.matrix( ~ time_of_stay_cust - 1, data = train)
trip_type_cust <- model.matrix( ~ trip_type_cust - 1, data = train)
view_type_cust <- model.matrix( ~ view_type_cust - 1, data = train)
CommonScore <- model.matrix( ~ CommonScore - 1, data = train)
tourists <- model.matrix( ~ tourists - 1, data = train)

score <- as.matrix(train$Reviewer_Score)

xgbtrain <- as.matrix(train[,c(2,5,6,7,10,11)])
xgbtrain <- cbind(xgbtrain, reviewer_region, city, weather_summary, access_type, access_type_cust, acce

str(xgbtrain)
```

```r
# reviewxgb <- xgboost(xgbtrain, score, max.depth = 3, eta = 1, nthread = 5, nrounds = 2000, objective
#
# importance_matrix <- xgb.importance(model = reviewxgb)
#
# # print(importance_matrix)
# xgb.plot.importance(importance_matrix = importance_matrix)
```

```r
reviewer_region2 <- model.matrix( ~ Reviewer_sub_region - 1, data= test )
city2 <- model.matrix( ~ City - 1, data = test)
weather_summary2 <- model.matrix( ~ weather_summary - 1, data = test)
access_type2 <- model.matrix( ~ access_type - 1, data = test)
accessibility2 <- model.matrix( ~ accessibility - 1, data = test)
free2 <- model.matrix( ~ free - 1, data = test)
other_room_types2 <- model.matrix( ~ other_room_types - 1, data = test)
room_type2 <- model.matrix( ~ room_type - 1, data = test)
time_of_stay2 <- model.matrix( ~ time_of_stay - 1, data = test)
trip_type2 <- model.matrix( ~ trip_type - 1, data = test)
view_type2 <- model.matrix( ~ view_type - 1, data = test)
access_type_cust2 <- model.matrix( ~ access_type_cust - 1, data = test)
accessibility_cust2 <- model.matrix( ~ accessibility_cust - 1, data = test)
free_cust2 <- model.matrix( ~ free_cust - 1, data = test)
other_cust2 <- model.matrix( ~ other_cust - 1, data = test)
other_room_types_cust2 <- model.matrix( ~ other_room_types_cust - 1, data = test)
room_type_cust2 <- model.matrix( ~ room_type_cust - 1, data = test)
time_of_stay_cust2 <- model.matrix( ~ time_of_stay_cust - 1, data = test)
trip_type_cust2 <- model.matrix( ~ trip_type_cust - 1, data = test)
view_type_cust2 <- model.matrix( ~ view_type_cust - 1, data = test)
CommonScore2 <- model.matrix( ~ CommonScore - 1, data = test)
tourists2 <- model.matrix( ~ tourists - 1, data = test)

score2 <- as.matrix(test$Reviewer_Score)

xgbtest <- as.matrix(test[,c(2,5,6,7,10,11)])
xgbtest <- cbind(xgbtest, reviewer_region2, city2, weather_summary2, access_type2, access_type_cust2, a
str(xgbtest)
```

```
# pred_score <- predict(reviewxgb, xgbtest)
#
# SSE <- sum((pred_score - score2)^2)
#
# SST <- sum((score2 - mean(score2))^2)
#
# 1 - (SSE / SST)
```

```
train_mean = mean(score)
test_mean = mean(score2)
xgbdatatrain <- xgb.DMatrix(xgbtrain, label = score)
xgbdatatest <- xgb.DMatrix(xgbtest, label = score2)
watchlist <- list(train = xgbdatatrain, test = xgbdatatest)
params <- list(max_depth = 3, eta = 1, nthread = 5)
reviewxgb_best <- xgb.train(params, xgbdatatrain, nrounds = 2000, early_stopping_rounds = 50,
                            watchlist = watchlist, print_every_n = 20, objective = "reg:linear",
                            base_score = train_mean)

# reviewxgb_cv <- xgb.cv()
```

```
error <- reviewxgb_compare7$evaluation_log

ggplot(error, aes(iter, test_rmse)) +
  geom_line()

pred_score2 <- predict(reviewxgb_best, xgbdatatest)


SSE2 <- sum((pred_score2 - score2)^2)

SST2 <- sum((score2 - mean(score2))^2)

1 - (SSE2 / SST2)

importance_matrix2 <- xgb.importance(model = reviewxgb_best)[1:20,]

print(importance_matrix2)
xgb.plot.importance(importance_matrix = importance_matrix2)


# eval <- as.tibble(reviewxgb_best$evaluation_log)
#
# top_n(eval, -5, test_rmse)
#
# reviewxgb_best$best_score
```

```
explainer = buildExplainer(reviewxgb_best, xgbdatatrain, type="regression", base_score = train_mean, tr

pred.breakdown = explainPredictions(reviewxgb_best, explainer, xgbdatatest)
```

```
cat('Breakdown Complete','\n')
weights = rowSums(pred.breakdown)
pred.xgb = 1/(1+exp(-weights))
```

```
cat(max(pred_score2-pred.xgb),'\n')

# idx_to_get = as.integer(802)
# pred.breakdown[idx_to_get,]
# showWaterfall(reviewxgb_best, explainer, xgbdatatest, data.matrix(test[,-1]) ,idx_to_get, type = "reg
```

```
plot(test[,"pct_positive"], pred.breakdown[,pct_positive], cex=0.4, pch=16, xlab = "Percent Positive", y

plot(test[,"distance"], pred.breakdown[,distance], cex=0.4, pch=16, xlab = "Distance", ylab = "Distance

plot(test[,"log_review_word_count"], pred.breakdown[,log_review_word_count], cex=0.4, pch=16, xlab = "Re

plot(test[,"TempLow"], pred.breakdown[,TempLow], cex=0.4, pch=16, xlab = "Low Temperature", ylab = "Low

plot(test[,"TempHigh"], pred.breakdown[,TempHigh], cex=0.4, pch=16, xlab = "High Temperature", ylab = "
```

## Section 3.4.1

```
# xgbtrain_noreview <- as.matrix(train[,c(2,6,10,11)])
# xgbtrain_noreview <- cbind(xgbtrain_noreview, reviewer_region, city, weather_summary, access_type, ac
#
# xgbtest_noreview <- as.matrix(test[,c(2,6,10,11)])
# xgbtest_noreview <- cbind(xgbtest_noreview, reviewer_region2, city2, weather_summary2, access_type2,
#
# xgbdatatrain_noreview <- xgb.DMatrix(xgbtrain_noreview, label = score)
# xgbdatatest_noreview <- xgb.DMatrix(xgbtest_noreview, label = score2)
# watchlist <- list(train = xgbdatatrain_noreview, test = xgbdatatest_noreview)

params <- list(max_depth = 3, eta = .9, nthread = 5)
reviewxgb_compare1 <- xgb.train(params, xgbdatatrain, nrounds = 2000, early_stopping_rounds = 50,
                                watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

params <- list(max_depth = 3, eta = .7, nthread = 5)
reviewxgb_compare2 <- xgb.train(params, xgbdatatrain, nrounds = 2000, early_stopping_rounds = 50,
                                watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

params <- list(max_depth = 3, eta = .5, nthread = 5)
reviewxgb_compare3 <- xgb.train(params, xgbdatatrain, nrounds = 2000, early_stopping_rounds = 50,
                                watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

params <- list(max_depth = 3, eta = .3, nthread = 5)
reviewxgb_compare4 <- xgb.train(params, xgbdatatrain, nrounds = 2000, early_stopping_rounds = 50,
                                watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

params <- list(max_depth = 3, eta = .1, nthread = 5)
reviewxgb_compare5 <- xgb.train(params, xgbdatatrain, nrounds = 2000, early_stopping_rounds = 50,
                                watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

# reviewxgb_cv <- xgb.cv()
```

```
params <- list(max_depth = 7, eta = .1, nthread = 8)
reviewxgb_compare6 <- xgb.train(params, xgbdatatrain, nrounds = 5000, early_stopping_rounds = 50,
                                watchlist = watchlist, print_every_n = 20, objective = "reg:linear")
```

```
params <- list(max_depth = 6, eta = .1, nthread = 8)
reviewxgb_compare7 <- xgb.train(params, xgbdatatrain, nrounds = 5000, early_stopping_rounds = 50,
                            watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

params <- list(max_depth = 5, eta = .1, nthread = 8)
reviewxgb_compare8 <- xgb.train(params, xgbdatatrain, nrounds = 5000, early_stopping_rounds = 50,
                            watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

# params <- list(max_depth = 3, eta = .1, nthread = 8)
# reviewxgb_compare9 <- xgb.train(params, xgbdatatrain, nrounds = 5000, early_stopping_rounds = 50,
#                             watchlist = watchlist, print_every_n = 20, objective = "reg:linear")
#
# params <- list(max_depth = 2, eta = .1, nthread = 8)
# reviewxgb_compare10 <- xgb.train(params, xgbdatatrain, nrounds = 5000, early_stopping_rounds = 50,
#                             watchlist = watchlist, print_every_n = 20, objective = "reg:linear")
```

## Section 3.4.2

```
params <- list(max_depth = 6, eta = .1, nthread = 8)
reviewxgb_final <- xgb.train(params, xgbdatatrain, nrounds = 5000, early_stopping_rounds = 50,
                            watchlist = watchlist, print_every_n = 20, objective = "reg:linear")

error <- reviewxgb_final$evaluation_log

ggplot(error[25:770,], aes(iter, test_rmse)) +
  geom_line() +
  ylab("Test Set RMSE") +
  xlab("Number of Trees")
```

```
## WARNING: THIS TAKES A LONG TIME TO RUN
pred_score_final <- predict(reviewxgb_final, xgbdatatest)


SSE_Final <- sum((pred_score_final - score2)^2)

SST <- sum((score2 - mean(score2))^2)

1 - (SSE3 / SST)

importance_matrix3 <- xgb.importance(model = reviewxgb_final)[1:20,]

ggplot(importance_matrix3[1:15,], aes(Feature, Gain)) +
  geom_bar(stat = "identity") +
  scale_x_discrete(limits = importance_matrix3$Feature[15:1]) +
  coord_flip()

print(importance_matrix3)
xgb.plot.importance(importance_matrix = importance_matrix3)

## TAKES A LONG TIME
explainer = buildExplainer(reviewxgb_final, xgbdatatrain, type="regression", base_score = train_mean, t

pred.breakdown = explainPredictions(reviewxgb_final, explainer, xgbdatatest)
```

```
plot(test[,"pct_positive"], pred.breakdown[,pct_positive], cex=0.4, pch=16, xlab = "Review positivity",

plot(test[,"distance"], pred.breakdown[,distance], cex=0.4, pch=16, xlab = "Distance from city center",

plot(test[,"log_review_word_count"], pred.breakdown[,log_review_word_count], cex=0.4, pch=16, xlab = "Lo

plot(test[,"TempLow"], pred.breakdown[,TempLow], cex=0.4, pch=16, xlab = "Low temperature", ylab = "Low

plot(test[,"TempHigh"], pred.breakdown[,TempHigh], cex=0.4, pch=16, xlab = "High temperature", ylab = "
```