MGSC 310 Final

Justin Lewinski

2023-05-16

```
library(readr)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.2 ──
## ✔ ggplot2 3.4.1     ✔ dplyr   1.1.0
## ✔ tibble  3.1.8     ✔ stringr 1.5.0
## ✔ tidyr   1.3.0     ✔ forcats 1.0.0
## ✔ purrr   1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(plotROC)
library(ggcorrplot)
library(ISLR)
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the eve
nt.
## Use the argument `event_level = "second"` to alter this as needed.
##
## Attaching package: 'yardstick'
##
## The following objects are masked from 'package:caret':
##
##     precision, recall, sensitivity, specificity
##
## The following object is masked from 'package:readr':
```

```
##
##     spec
```

```
library(tidyverse)
library(rsample)
library(glmnet)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-7
```

```
library(glmnetUtils)
```

```
##
## Attaching package: 'glmnetUtils'
##
## The following objects are masked from 'package:glmnet':
##
##     cv.glmnet, glmnet
```

```
library(forcats)
library(randomForestExplainer)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggplot2)
library(randomForest)
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

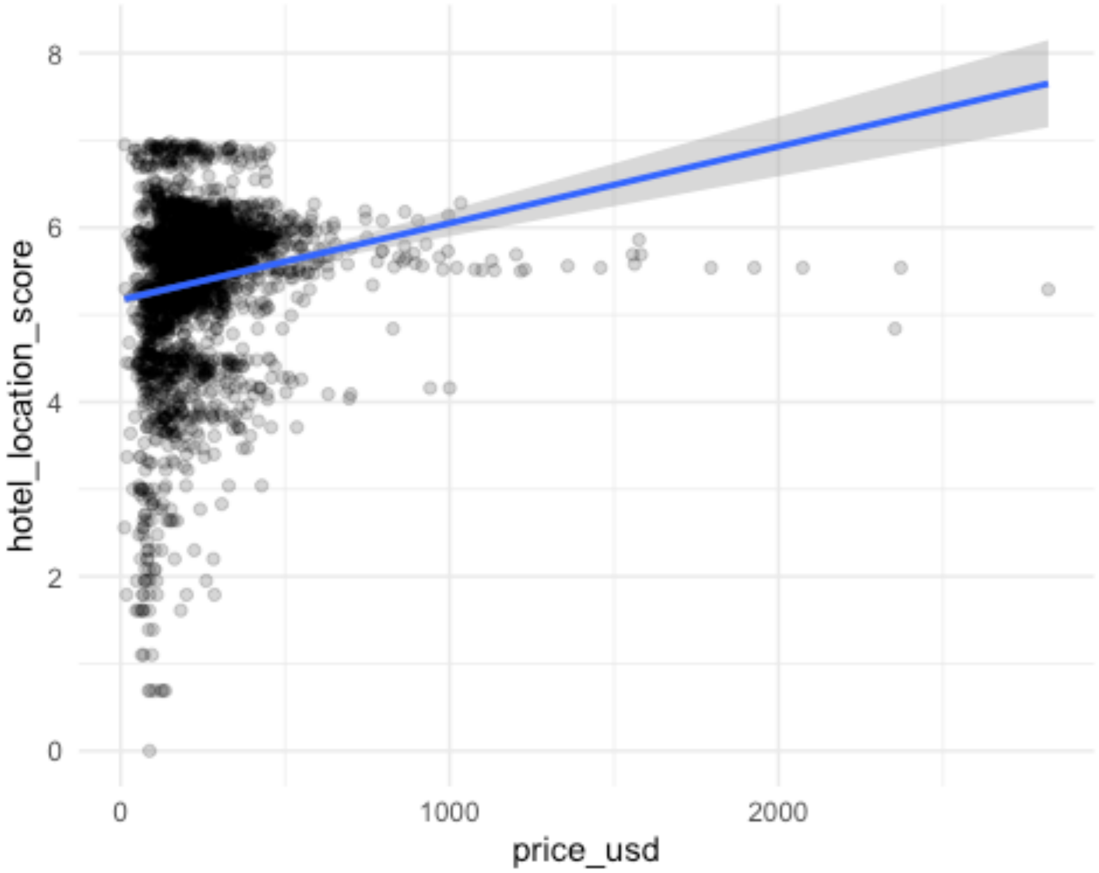
2

```
data <- read_csv('datasets/booking-1.csv')

## Rows: 2632 Columns: 24
## ── Column specification ──────────────────────────────────────────────────
──────
## Delimiter: ","
## dbl  (23): srch_id, site_id, visitor_country_id, hotel_country_id, hotel_i
d,...
## dttm  (1): date_time
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

data %>% summary()

##     srch_id         date_time                          site_id     
##  Min.   :    93   Min.   :2012-11-06 05:18:44.00   Min.   : 1.00  
##  1st Qu.:214708   1st Qu.:2013-02-10 17:09:01.00   1st Qu.: 5.00  
##  Median :218049   Median :2013-04-26 03:07:51.00   Median :12.00  
##  Mean   :259787   Mean   :2013-04-04 15:55:42.73   Mean   :11.96  
##  3rd Qu.:237124   3rd Qu.:2013-05-24 21:51:53.00   3rd Qu.:16.00  
##  Max.   :665537   Max.   :2013-06-30 19:02:59.00   Max.   :34.00  
##  visitor_country_id hotel_country_id    hotel_id        hotel_stars   
##  Min.   :  2.0      Min.   : 31.0    Min.   :   223   Min.   :0.000  
##  1st Qu.: 59.0      1st Qu.: 99.0    1st Qu.: 31304   1st Qu.:3.000  
##  Median :129.0      Median :129.0    Median : 63445   Median :4.000  
##  Mean   :141.3      Mean   :137.8    Mean   : 66169   Mean   :3.776  
##  3rd Qu.:219.0      3rd Qu.:219.0    3rd Qu.:101677   3rd Qu.:4.000  
##  Max.   :228.0      Max.   :219.0    Max.   :140694   Max.   :5.000  
##  hotel_review_score  hotel_chain     hotel_location_score
##  Min.   :0.000      Min.   :0.0000   Min.   :0.000       
##  1st Qu.:4.000      1st Qu.:0.0000   1st Qu.:5.110       
##  Median :4.000      Median :1.0000   Median :5.660       
##  Mean   :4.022      Mean   :0.5129   Mean   :5.387       
##  3rd Qu.:4.500      3rd Qu.:1.0000   3rd Qu.:5.900       
##  Max.   :5.000      Max.   :1.0000   Max.   :6.980       
##  hotel_historical_price search_ranking    price_usd        promotion     
##  Min.   :0.000          Min.   : 1.00   Min.   :  12.0   Min.   :0.0000  
##  1st Qu.:4.940          1st Qu.: 6.00   1st Qu.: 140.7   1st Qu.:0.0000  
##  Median :5.450          Median :13.00   Median : 216.0   Median :0.0000  
##  Mean   :4.598          Mean   :14.73   Mean   : 246.1   Mean   :0.3891  
##  3rd Qu.:5.790          3rd Qu.:24.00   3rd Qu.: 301.1   3rd Qu.:1.0000  
##  Max.   :6.210          Max.   :36.00   Max.   :2820.0   Max.   :1.0000  
##  length_of_stay   booking_window    adults_count    children_count  
##  Min.   : 1.000   Min.   :  0.00   Min.   :1.000   Min.   :0.0000  
##  1st Qu.: 1.000   1st Qu.:  5.00   1st Qu.:1.000   1st Qu.:0.0000  
##  Median : 3.000   Median : 20.00   Median :2.000   Median :0.0000  
##  Mean   : 2.803   Mean   : 27.99   Mean   :1.874   Mean   :0.3131  
##  3rd Qu.: 4.000   3rd Qu.: 46.00   3rd Qu.:2.000   3rd Qu.:1.0000  
##  Max.   :13.000   Max.   :173.00   Max.   :7.000   Max.   :3.0000  
##    room_count    saturday_night    random_sort       comp_rate      
##  Min.   :1.000   Min.   :0.0000   Min.   :0.0000   Min.   :-1.0000  
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.0000  
##  Median :1.000   Median :0.0000   Median :0.0000   Median : 1.0000  
##  Mean   :1.267   Mean   :0.4734   Mean   :0.2272   Mean   : 0.5258  
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 1.0000  
##  Max.   :7.000   Max.   :1.0000   Max.   :1.0000   Max.   : 1.0000  
##     comp_inv          booking      
##  Min.   :-1.000   Min.   :0.0000  
##  1st Qu.: 0.000   1st Qu.:0.0000  
##  Median : 0.000   Median :0.0000  
##  Mean   : 0.019   Mean   :0.3625  
##  3rd Qu.: 0.000   3rd Qu.:1.0000  
##  Max.   : 1.000   Max.   :1.0000
```

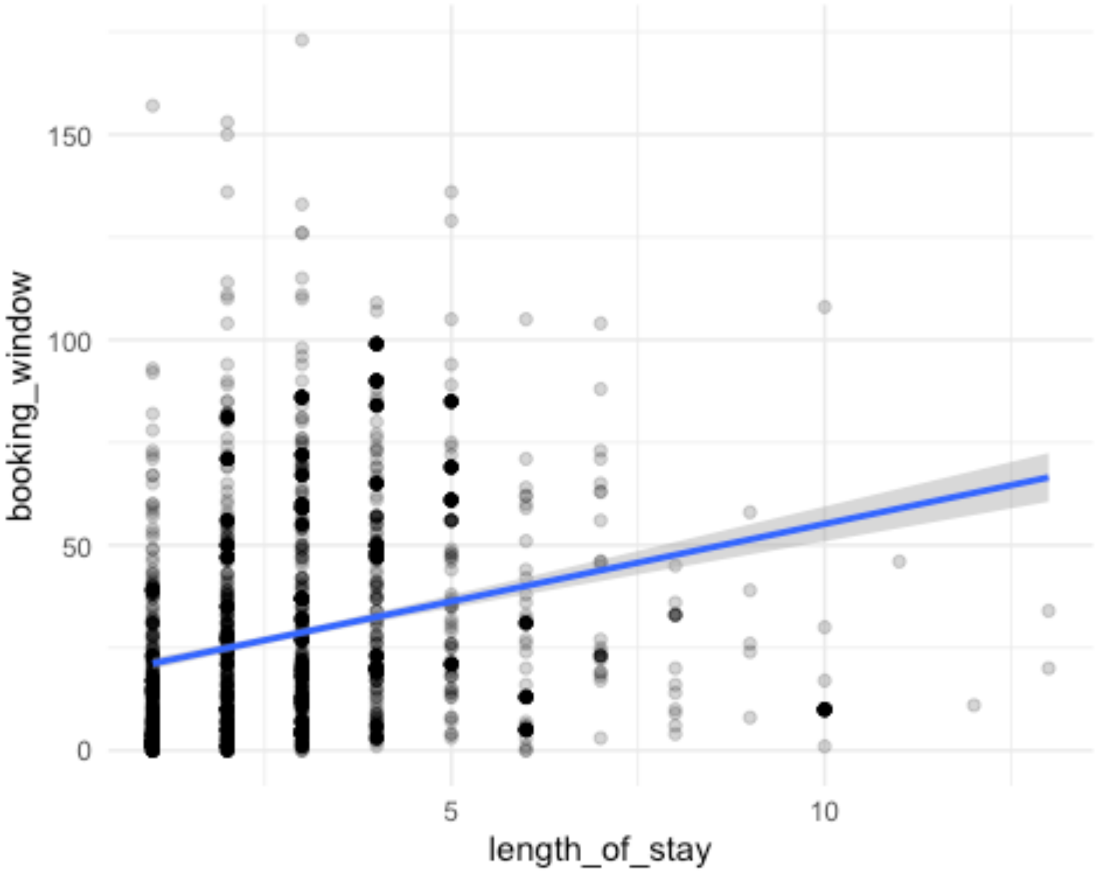
```
data_clean <-
  data %>%
  mutate(reviewscore2 = hotel_review_score**2,
         hotel_stars2 = hotel_stars**2,
         price2 = price_usd**2,
         booking_window_2 = booking_window**2)
```

```
ggplot(data_clean, aes(x = price_usd, y = hotel_location_score)) + geom_point
(alpha = 0.2) + geom_smooth(method = lm) + theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(data_clean, aes(x = length_of_stay, y = booking_window)) + geom_point(
alpha = 0.2) + geom_smooth(method = lm) + theme_minimal()

## `geom_smooth()` using formula = 'y ~ x'
```



3

```
set.seed(2370404)
initial_split_hotel <- initial_split(data_clean, prop = .75)
hotel_train <- training(initial_split_hotel)
hotel_test <- testing(initial_split_hotel)

hotel_train %>% glimpse

## Rows: 1,974
## Columns: 28
## $ srch_id                   <dbl> 312781, 218049, 215038, 237161, 322621, 864
24, …
## $ date_time                 <dttm> 2013-05-13 20:38:38, 2013-01-31 15:29:28,
2013…
## $ site_id                   <dbl> 15, 24, 5, 7, 24, 15, 16, 24, 15, 14, 5, 32
, 15…
## $ visitor_country_id        <dbl> 129, 216, 219, 219, 216, 55, 31, 99, 55, 10
0, 2…
## $ hotel_country_id          <dbl> 129, 56, 219, 219, 181, 132, 60, 99, 99, 99
, 99…
## $ hotel_id                  <dbl> 62765, 79465, 89359, 68487, 110250, 32831,
1357…
```

```
## $ hotel_stars            <dbl> 5, 4, 2, 4, 4, 4, 4, 3, 3, 3, 3, 4, 3, 4, 3
, 4,…
## $ hotel_review_score     <dbl> 4.5, 4.0, 2.5, 4.0, 4.0, 4.0, 4.0, 4.0, 3.5
, 4.…
## $ hotel_chain            <dbl> 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0
, 0,…
## $ hotel_location_score   <dbl> 4.66, 4.82, 5.79, 5.61, 4.33, 5.54, 4.47, 5
.10,…
## $ hotel_historical_price <dbl> 5.46, 5.26, 5.48, 5.73, 0.00, 5.49, 4.77, 5
.28,…
## $ search_ranking         <dbl> 8, 9, 22, 2, 1, 2, 2, 20, 15, 18, 9, 8, 16,
8, …
## $ price_usd              <dbl> 183.57, 121.26, 335.00, 286.88, 106.64, 134
.06,…
## $ promotion              <dbl> 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0,…
## $ length_of_stay         <dbl> 3, 6, 3, 7, 3, 1, 2, 1, 5, 3, 1, 10, 1, 1,
1, 2…
## $ booking_window         <dbl> 4, 5, 21, 23, 20, 32, 6, 0, 69, 76, 3, 108,
23,…
## $ adults_count           <dbl> 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 4, 3, 3, 2, 4
, 2,…
## $ children_count         <dbl> 1, 0, 3, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0
, 1,…
## $ room_count             <dbl> 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 2
, 2,…
## $ saturday_night         <dbl> 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1
, 0,…
## $ random_sort            <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1
, 0,…
## $ comp_rate              <dbl> 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1,…
## $ comp_inv               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0,…
## $ booking                <dbl> 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0
, 1,…
## $ reviewscore2           <dbl> 20.25, 16.00, 6.25, 16.00, 16.00, 16.00, 16
.00,…
## $ hotel_stars2           <dbl> 25, 16, 4, 16, 16, 16, 16, 9, 9, 9, 9, 16,
9, 1…
## $ price2                 <dbl> 33697.945, 14703.988, 112225.000, 82300.134
, 11…
## $ booking_window_2       <dbl> 16, 25, 441, 529, 400, 1024, 36, 0, 4761, 5
776,…
```

```
logit <- glm(booking ~ hotel_stars + hotel_review_score + hotel_chain +
                       hotel_location_score + hotel_historical_price + search
_ranking
                       + price_usd + promotion + length_of_stay + booking_win
dow +
                         adults_count + children_count + room_count + saturda
y_night
                       + comp_rate + comp_inv, data = hotel_train, family = b
inomial)
logit %>% summary()

## 
## Call:
## glm(formula = booking ~ hotel_stars + hotel_review_score + hotel_chain + 
##     hotel_location_score + hotel_historical_price + search_ranking + 
##     price_usd + promotion + length_of_stay + booking_window + 
##     adults_count + children_count + room_count + saturday_night + 
##     comp_rate + comp_inv, family = binomial, data = hotel_train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.9258  -0.8535  -0.4898   0.9502   2.7506  
## 
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)    
## (Intercept)             1.2750113  0.4747510   2.686 0.007239 ** 
## hotel_stars             0.1433822  0.0654857   2.190 0.028559 *  
## hotel_review_score      0.0418149  0.0756897   0.552 0.580639    
## hotel_chain             0.0729350  0.1129255   0.646 0.518365    
## hotel_location_score    0.1205084  0.0570104   2.114 0.034533 *  
## hotel_historical_price -0.1177641  0.0255086  -4.617 3.90e-06 ***
## search_ranking         -0.0799179  0.0056578 -14.125  < 2e-16 ***
## price_usd              -0.0052525  0.0005691  -9.230  < 2e-16 ***
## promotion               0.0586445  0.1143186   0.513 0.607957    
## length_of_stay         -0.0498322  0.0345273  -1.443 0.148944    
## booking_window          0.0008761  0.0021523   0.407 0.683965    
## adults_count            0.0974074  0.0683817   1.424 0.154311    
## children_count          0.4020869  0.1068958   3.761 0.000169 ***
## room_count             -0.5048533  0.1202061  -4.200 2.67e-05 ***
## saturday_night          0.0141528  0.1131327   0.125 0.900445    
## comp_rate              -0.0018489  0.0855885  -0.022 0.982765    
## comp_inv               -0.5303007  0.3260470  -1.626 0.103853    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2592.5  on 1973  degrees of freedom
## Residual deviance: 2130.7  on 1957  degrees of freedom
## AIC: 2164.7
## 
## Number of Fisher Scoring iterations: 5

exp(logit$coefficients)
```

```
##            (Intercept)            hotel_stars     hotel_review_score 
##              3.5787418              1.1541709              1.0427015 
##            hotel_chain   hotel_location_score hotel_historical_price 
##              1.0756606              1.1280703              0.8889057 
##         search_ranking              price_usd              promotion 
##              0.9231921              0.9947612              1.0603982 
##         length_of_stay         booking_window           adults_count 
##              0.9513890              1.0008765              1.1023094 
##         children_count             room_count         saturday_night 
##              1.4949413              0.6035941              1.0142534 
##              comp_rate               comp_inv 
##              0.9981528              0.5884280
```

```
scores_train <- predict(logit, typ = "response")
scores_test <- predict(logit, type = "response", newdata = hotel_test)
```

```
predicted_train<- ifelse(scores_train>0.5,"1","0")
head(predicted_train)
```

```
##   1   2   3   4   5   6 
## "1" "1" "0" "0" "1" "1"
```

```
predicted_test <-  ifelse(scores_test>0.5,"1","0")
head(predicted_test)
```

```
##   1   2   3   4   5   6 
## "0" "0" "0" "0" "0" "0"
```

```
results_train <- data.frame(
  true = factor(hotel_train$booking),
  predicted = factor(predicted_train),
  score = scores_train)
sum(results_train$predicted == 1)
```

```
## [1] 604
```

```
sum(results_train$predicted == 0)
```

```
## [1] 1370
```

```
results_test <- data.frame(
  true = factor(hotel_test$booking),
  predicted = factor(predicted_test),
  score = scores_test)
results_test %>% glimpse()
```

```
## Rows: 658
## Columns: 3
## $ true      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ predicted <fct> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
0, …
```

```
## $ score     <dbl> 0.3066496687, 0.2589455048, 0.3094074926, 0.0007639443,
0.02…
```

```
cm_trainlogit <- conf_mat(results_train,
                 truth = true,
                 estimate = predicted)

cm_testlogit <- conf_mat(results_test,
                 truth = true,
                 estimate = predicted)

print(cm_trainlogit)
```

```
##            Truth
## Prediction    0    1
##           0 1022  348
##           1  230  374
```

```
print(cm_testlogit)
```

```
##            Truth
## Prediction   0   1
##           0 353  98
##           1  73 134
```

```
TN_trainlogit = 1022
TP_trainlogit = 374
FN_trainlogit = 348
FP_trainlogit = 230



TN_testlogit = 353
TP_testlogit = 134
FN_testlogit = 98
FP_testlogit = 73

print('train_scores')
```

```
## [1] "train_scores"
```

```
acc_trainlogit = (TN_trainlogit +TP_trainlogit ) / (TN_trainlogit + TP_trainl
ogit + FN_trainlogit + FP_trainlogit)
print(acc_trainlogit)
```

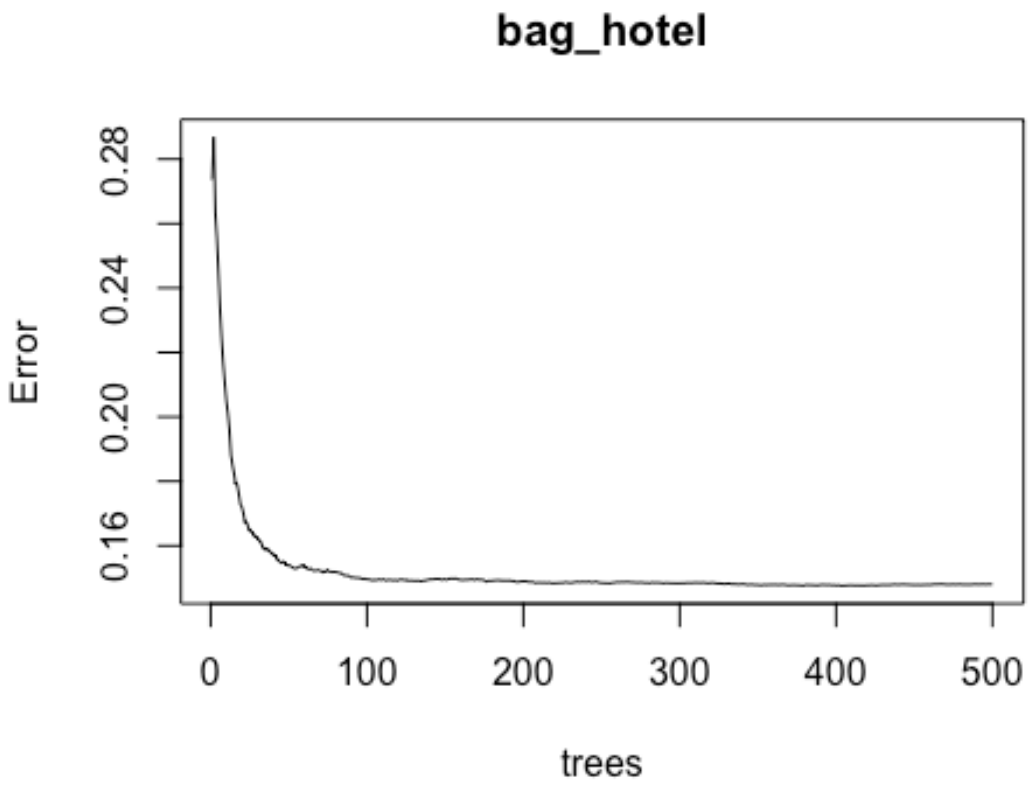
```
## [1] 0.7071935
```

```
sen_trainlogit = TP_trainlogit/(TP_trainlogit +FN_trainlogit)
print(sen_trainlogit)
```

```
## [1] 0.5180055
```

```
spe_trainlogit = TN_trainlogit/(TN_trainlogit + FP_trainlogit)

print(spe_trainlogit)

## [1] 0.8162939

print('test scores')

## [1] "test scores"

acc_testlogit = (TN_testlogit +TP_testlogit ) / (TN_testlogit + TP_testlogit
+ FN_testlogit + FP_testlogit)
print(acc_testlogit)

## [1] 0.7401216

sen_testlogit = TP_testlogit/(TP_testlogit +FN_testlogit)
print(sen_testlogit)

## [1] 0.5775862

spe_testlogit = TN_testlogit/(TN_testlogit + FP_testlogit)

print(spe_testlogit)

## [1] 0.8286385

bag_hotel <- randomForest(booking ~ hotel_stars + hotel_review_score + hotel_
chain +
                       hotel_location_score + hotel_historical_price + search
_ranking
                       + price_usd + promotion + length_of_stay + booking_win
dow +
                         adults_count + children_count + room_count + saturda
y_night
                       + comp_rate + comp_inv, data = hotel_train)

## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?

print(bag_hotel)

##
## Call:
##  randomForest(formula = booking ~ hotel_stars + hotel_review_score +
hotel_chain + hotel_location_score + hotel_historical_price +      search_ran
king + price_usd + promotion + length_of_stay +      booking_window + adults_
count + children_count + room_count +      saturday_night + comp_rate + comp_
inv, data = hotel_train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
```

```
##           Mean of squared residuals: 0.1480358
##                     % Var explained: 36.19

plot(bag_hotel)
```



```
scores_trainbag <- predict(bag_hotel, typ = "response")
scores_testbag <- predict(bag_hotel, type = "response", newdata = hotel_test)

predicted_trainbag<- ifelse(scores_trainbag>0.5,"1","0")
head(predicted_trainbag)

##   1   2   3   4   5   6 
## "0" "0" "0" "0" "1" "1"

predicted_testbag <-  ifelse(scores_testbag>0.5,"1","0")
head(predicted_testbag)

##   1   2   3   4   5   6 
## "0" "0" "0" "0" "0" "0"

results_trainbag <- data.frame(
  true = factor(hotel_train$booking),
  predicted = factor(predicted_trainbag),
  score = scores_trainbag)
sum(results_trainbag$predicted == 1)

## [1] 594

sum(results_trainbag$predicted == 0)

## [1] 1380

results_testbag <- data.frame(
  true = factor(hotel_test$booking),
  predicted = factor(predicted_testbag),
  score = scores_testbag)
results_testbag %>% glimpse()

## Rows: 658
## Columns: 3
## $ true      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ predicted <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, …
## $ score     <dbl> 0.23620000, 0.20953333, 0.25796667, 0.07130000, 0.177800
00, …

cm_trainbag <- conf_mat(results_trainbag,
                  truth = true,
                  estimate = predicted)

cm_testbag <- conf_mat(results_testbag,
                  truth = true,
                  estimate = predicted)
print(cm_trainbag)

##           Truth
## Prediction    0    1
##          0 1103  277
##          1  149  445

print(cm_testbag)

##           Truth
## Prediction   0   1
##          0 379  80
##          1  47 152

TN_trainbag = 1108
TP_trainbag = 443
FN_trainbag = 279
FP_trainbag = 144

TN_testbag = 379
TP_testbag = 151
```

```
FN_testbag = 81
FP_testbag = 47

print('train scores forest')

## [1] "train scores forest"

acc_trainbag = (TN_trainbag +TP_trainbag ) / (TN_trainbag + TP_trainbag + FN_
trainbag + FP_trainbag)
print(acc_trainbag)

## [1] 0.7857143

sen_trainbag = TP_trainbag/(TP_trainbag +FN_trainbag)
print(sen_trainbag)

## [1] 0.6135734

spe_trainbag = TN_trainbag/(TN_trainbag + FP_trainbag)

print(spe_trainbag)

## [1] 0.884984

print('test scores forest')

## [1] "test scores forest"

acc_testbag = (TN_testbag +TP_testbag ) / (TN_testbag + TP_testbag + FN_testb
ag + FP_testbag)
print(acc_testbag)

## [1] 0.8054711

sen_testbag = TP_testbag/(TP_testbag +FN_testbag)
print(sen_testbag)

## [1] 0.6508621

spe_testbag = TN_testbag/(TN_testbag + FP_testbag)

print(spe_testbag)

## [1] 0.8896714

print('Logit Model Test')

## [1] "Logit Model Test"

print(cm_testlogit)

##             Truth
## Prediction   0   1
##           0 353  98
##           1  73 134

print('Bagging Model Test')

## [1] "Bagging Model Test"

print(cm_testbag)

##             Truth
## Prediction   0   1
##           0 379  80
##           1  47 152

print('TP % Increase From logit to bag = ')

## [1] "TP % Increase From logit to bag = "

print(1 - (TP_testlogit/TP_testbag))

## [1] 0.1125828

print('TN % Increase From logit to bag = ')

## [1] "TN % Increase From logit to bag = "

print(1 - (TN_testlogit/TN_testbag))

## [1] 0.06860158

print('FP % decrease From logit to bag = ')

## [1] "FP % decrease From logit to bag = "

print(1 - (FP_testbag/FP_testlogit))

## [1] 0.3561644

print('FN % decrease From logit to bag = ')

## [1] "FN % decrease From logit to bag = "

print(1 - (FN_testbag/FN_testlogit))

## [1] 0.1734694
```

5

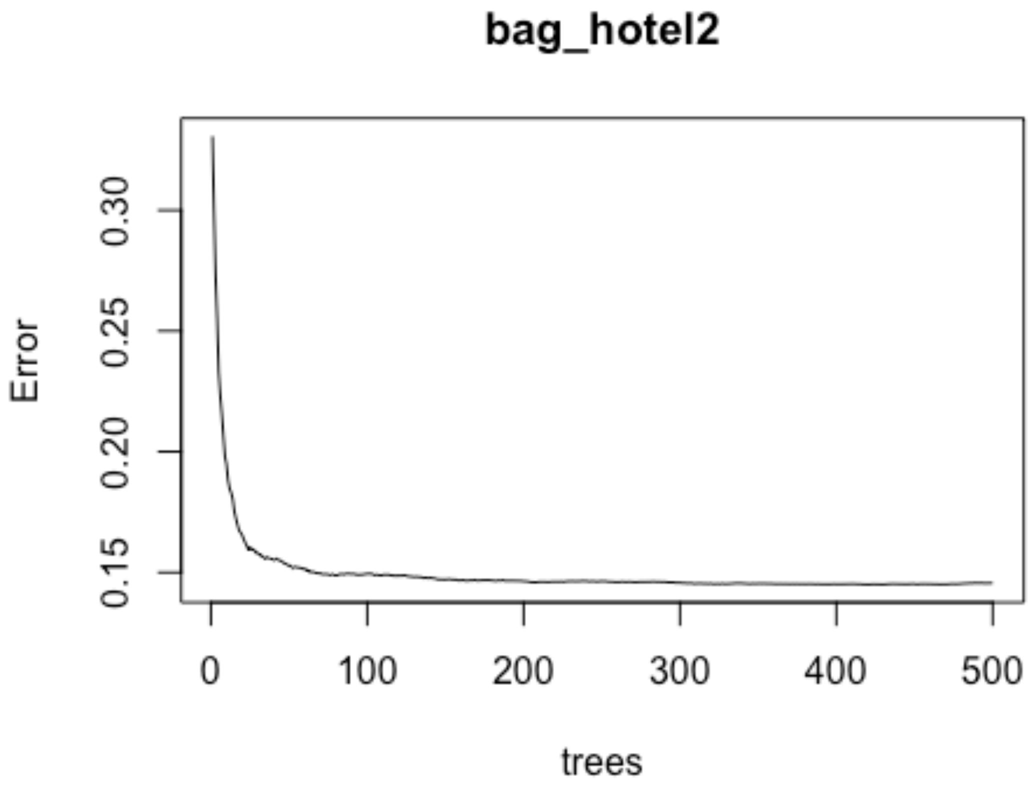
```
bag_hotel2 <- randomForest(booking ~ hotel_stars + hotel_review_score + hotel
_chain +
                       hotel_location_score + hotel_historical_price + search
_ranking
                       + price_usd + promotion + length_of_stay + booking_win
dow +
                         adults_count + children_count + room_count + saturda
y_night
                       + comp_rate + comp_inv + reviewscore2 + hotel_stars2 +
price2 + booking_window_2, data = hotel_train)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
print(bag_hotel2)
```

```
## 
## Call:
##  randomForest(formula = booking ~ hotel_stars + hotel_review_score +
hotel_chain + hotel_location_score + hotel_historical_price +      search_ran
king + price_usd + promotion + length_of_stay +      booking_window + adults_
count + children_count + room_count +      saturday_night + comp_rate + comp_
inv + reviewscore2 + hotel_stars2 +      price2 + booking_window_2, data = ho
tel_train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
## 
##           Mean of squared residuals: 0.1455956
##                     % Var explained: 37.24
```

```
plot(bag_hotel2)
```



```
scores_trainbag2 <- predict(bag_hotel2, typ = "response")
scores_testbag2 <- predict(bag_hotel2, type = "response", newdata = hotel_tes
t)

predicted_trainbag2<- ifelse(scores_trainbag>0.5,"1","0")
head(predicted_trainbag2)

##   1   2   3   4   5   6
## "0" "0" "0" "0" "1" "1"

predicted_testbag2 <-  ifelse(scores_testbag>0.5,"1","0")
head(predicted_testbag2)

##   1   2   3   4   5   6
## "0" "0" "0" "0" "0" "0"

results_trainbag2 <- data.frame(
  true = factor(hotel_train$booking),
  predicted = factor(predicted_trainbag2),
  score = scores_trainbag2)
results_testbag2 <- data.frame(
  true = factor(hotel_test$booking),
  predicted = factor(predicted_testbag2),
  score = scores_testbag2)
```

```
results_testbag %>% glimpse()
```

```
## Rows: 658
## Columns: 3
## $ true      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ predicted <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, …
## $ score     <dbl> 0.23620000, 0.20953333, 0.25796667, 0.07130000, 0.177800
00, …
```

```
sum(results_trainbag2$predicted == 1)
```

```
## [1] 594
```

```
sum(results_trainbag2$predicted == 0)
```

```
## [1] 1380
```

```
cm_trainbag2 <- conf_mat(results_trainbag2,
                truth = true,
                estimate = predicted)

cm_testbag2 <- conf_mat(results_testbag2,
                truth = true,
                estimate = predicted)
print(cm_trainbag2)
```

```
##             Truth
## Prediction     0    1
##          0 1103  277
##          1  149  445
```

```
print(cm_testbag2)
```

```
##             Truth
## Prediction   0   1
##          0 379  80
##          1  47 152
```

```
TN_trainbag2 = 1116
TP_trainbag2 = 445
FN_trainbag2 = 277
FP_trainbag2 = 136

TN_testbag2 = 379
TP_testbag2 = 160
FN_testbag2 = 72
FP_testbag2 = 47

print('test scores forest improved model')
```

```
## [1] "test scores forest improved model"

acc_testbag2 = (TN_testbag2 +TP_testbag2 ) / (TN_testbag2 + TP_testbag2 + FN_
testbag2 + FP_testbag2)
print(acc_testbag2)

## [1] 0.8191489

sen_testbag2 = TP_testbag2/(TP_testbag2 +FN_testbag)
print(sen_testbag2)

## [1] 0.6639004

spe_testbag2 = TN_testbag2/(TN_testbag2 + FP_testbag2)

print(spe_testbag2)

## [1] 0.8896714
```