Hotel Hospitality Analysis

2023-07-03

R Markdown

This data set contains booking information for a city hotel and a resort hotel, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things.

```
#Reading the dataset
#libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

```
## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: lattice
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v tibble  3.2.1     v purrr   1.0.1
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0

## Warning: package 'tibble' was built under R version 4.2.3

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
```

```
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.2.3
```

```
##
## Attaching package: 'xgboost'
##
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 4.2.3
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 4.2.3
```

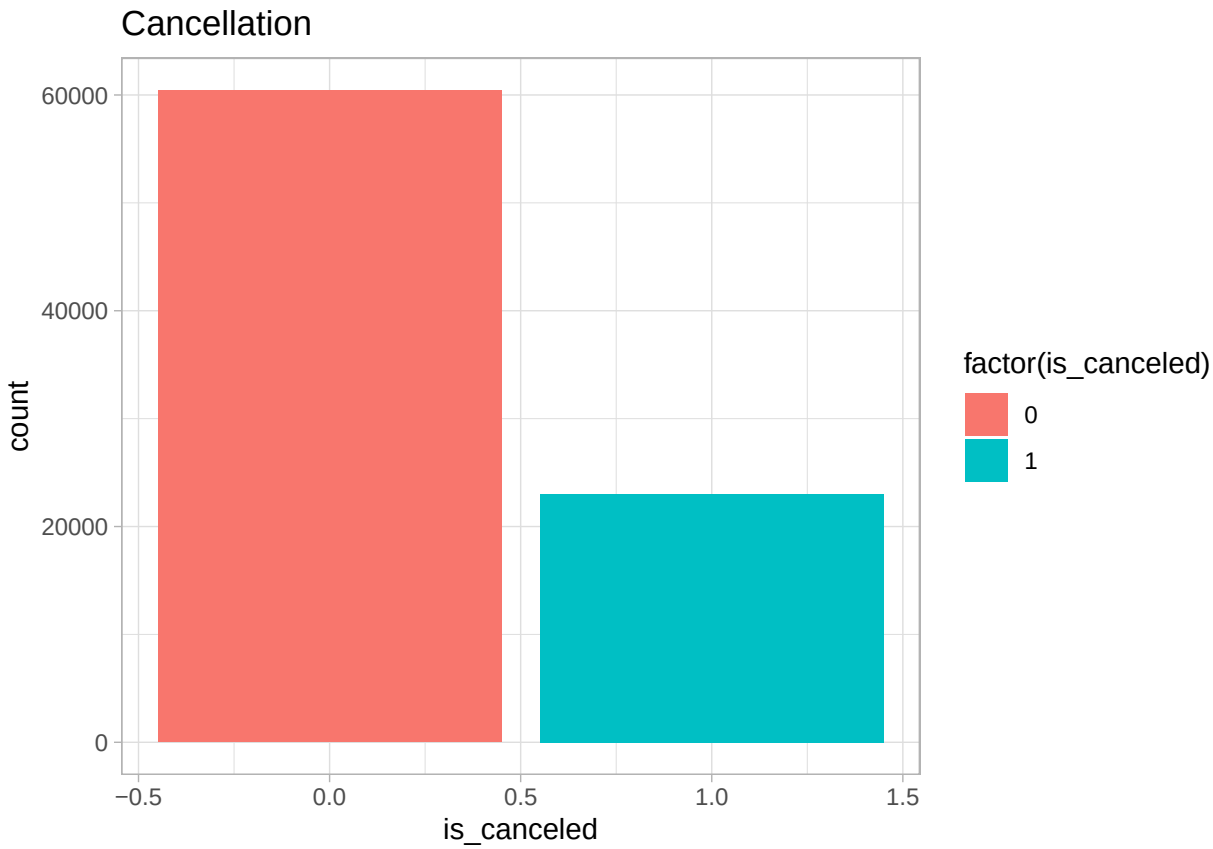
```
##
## Attaching package: 'foreach'
##
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
##
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 4.2.3
```

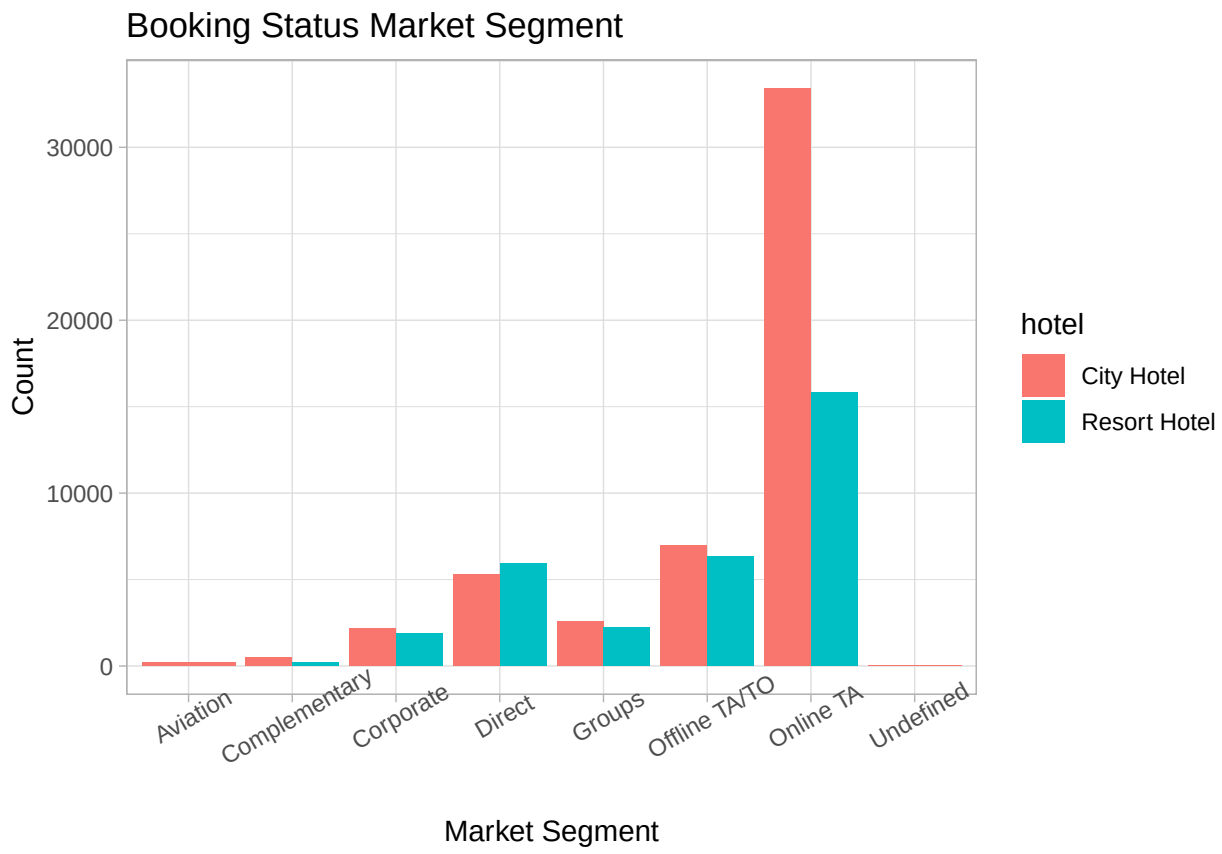
```
## Loading required package: parallel
```

```
hos_data <- read.csv("csv/output.csv", header=T)
```

```
ggplot(hos_data, aes(is_canceled, fill = factor(is_canceled))) +
  geom_bar()+
  labs(title = "Cancellation")+ theme_light()
```

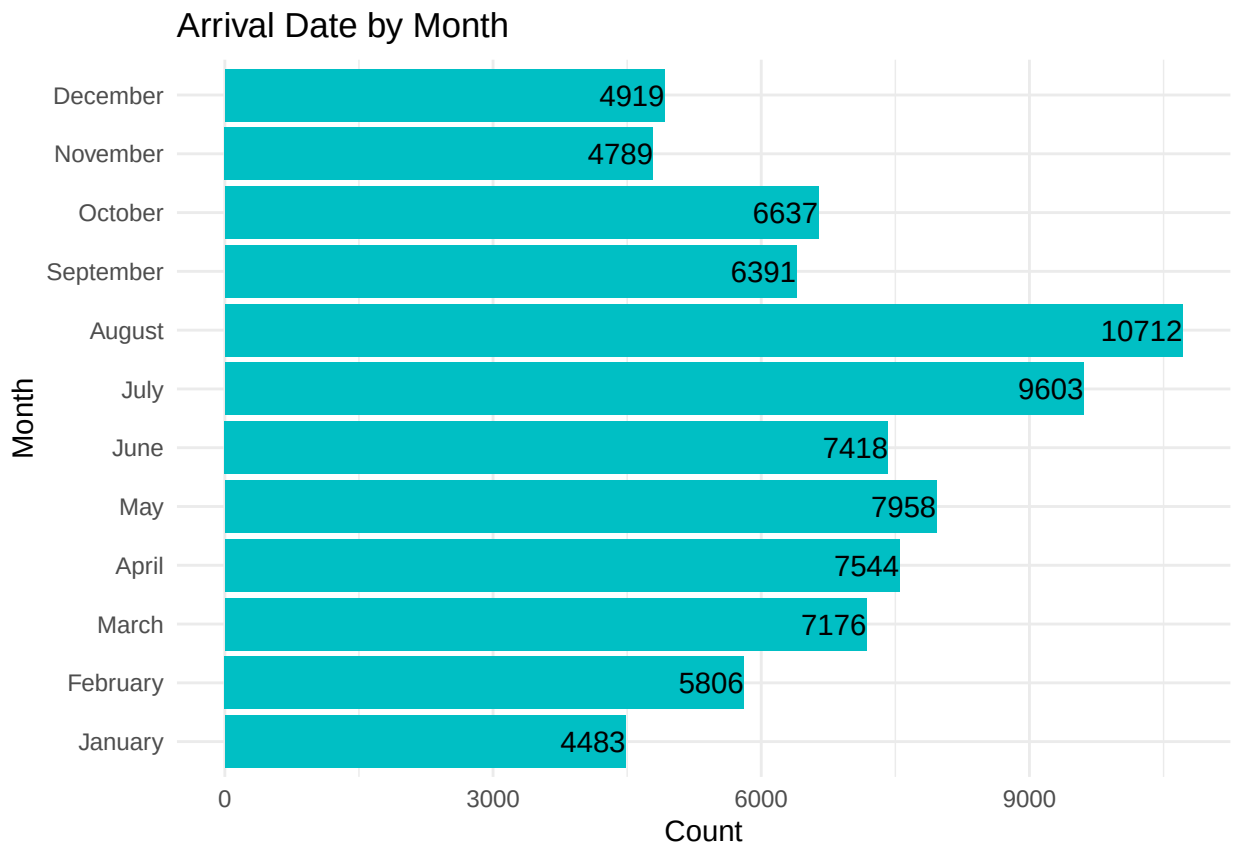



```
#Bar plot of Market Segmant and Hotel Type
ggplot(hos_data, aes(market_segment, fill = hotel)) +
  geom_bar(position = position_dodge()) +
  labs(title = "Booking Status Market Segment",
       x = "Market Segment",
       y = "Count")+ theme_light()+
  theme(axis.text.x = element_text(angle = 30))
```




```
#Visualization
hos_data$arrival_date_month <-
  factor(hos_data$arrival_date_month, levels = month.name)


ggplot(data = hos_data, aes(x = arrival_date_month)) +
  geom_bar(fill =  "#00BFC4") +
  geom_text(stat = "count", aes(label = ..count..), hjust = 1) +
  coord_flip() + labs(title = "Arrival Date by Month",
                      x = "Month",
                      y = "Count") +
  theme_minimal()
```

```
## Warning: The dot-dot notation (‘..count..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(count)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
#create dummy variables expect for the response
dummies_model <- dummyVars(is_canceled~ ., data = hos_data)
#if the response is a factor may get a warning that you can ignore


#provide only predictors that are now converted to dummy variables
hos_predictors_dummy<- data.frame(predict(dummies_model, newdata = hos_data))

#recombine predictors including dummy variables with response
hos_data <- cbind(is_canceled=hos_data$is_canceled, hos_predictors_dummy)

#change reponse to a factor
hos_data$is_canceled<-as.factor(hos_data$is_canceled)


# rename resonse
hos_data$is_canceled<-fct_recode(hos_data$is_canceled, is_canceled = "1",notCanceled = "0")

# relevel response
hos_data$is_canceled<- relevel(hos_data$is_canceled, ref = "is_canceled")

#make sure levels are correct
levels(hos_data$is_canceled)
```

```
## [1] "is_canceled" "notCanceled"
```

```r
#Partition the data into test and train data
set.seed(99)
index <-createDataPartition(hos_data$is_canceled, p = .8,list = FALSE)
hos_train <- hos_data[index,]

hos_test <- hos_data[-index,]
```

```r
#Running the xgboost model


#total number of cores on your computer
num_cores<-detectCores(logical=FALSE)

#Snce the model takes a lot of time to run, parallel processing can speed up the process
cl <- makePSOCKcluster(num_cores-1)
registerDoParallel(cl)


set.seed(8)
model_gbm <- train(is_canceled ~ .,
                   data = hos_train,
                   method = "xgbTree",
                   trControl =trainControl(method = "cv",
                                           number = 2,
                                           classProbs = TRUE,
                                           summaryFunction = twoClassSummary),
                   # provide a grid of parameters
                   tuneGrid = expand.grid(
                     nrounds = c(50,200),
                     eta = c(0.025, 0.05),
                     max_depth = c(2, 3),
                     gamma = 0,
                     colsample_bytree = 1,
                     min_child_weight = 1,
                     subsample = 1),
                     na.action=na.exclude,
                     metric="ROC"
                   )
```

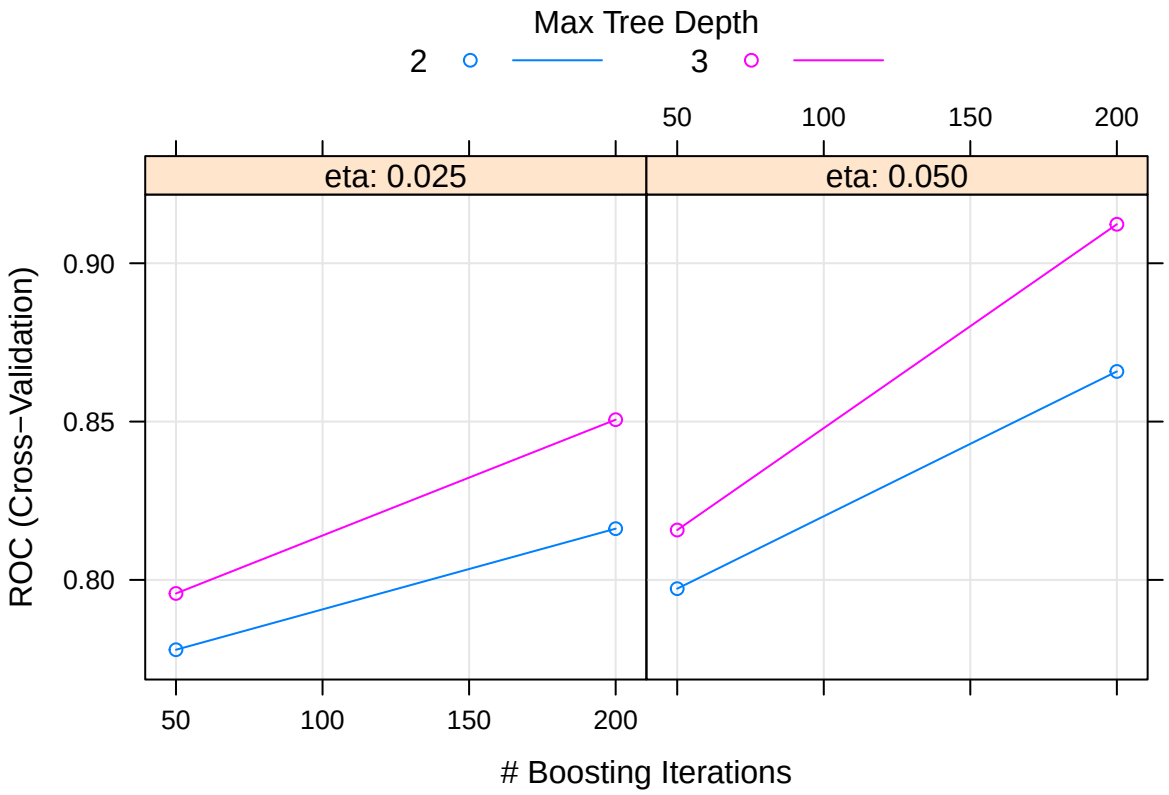
```
## [12:09:55] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:09:55] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:07] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:07] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:14] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:14] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:26] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:26] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:32] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:32] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:42] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:42] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:10:49] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
```

```
## [12:11:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
## [12:11:00] WARNING: src/c_api/c_api.cc:935: `ntree_limit` is deprecated, use `iteration_range` instea
```

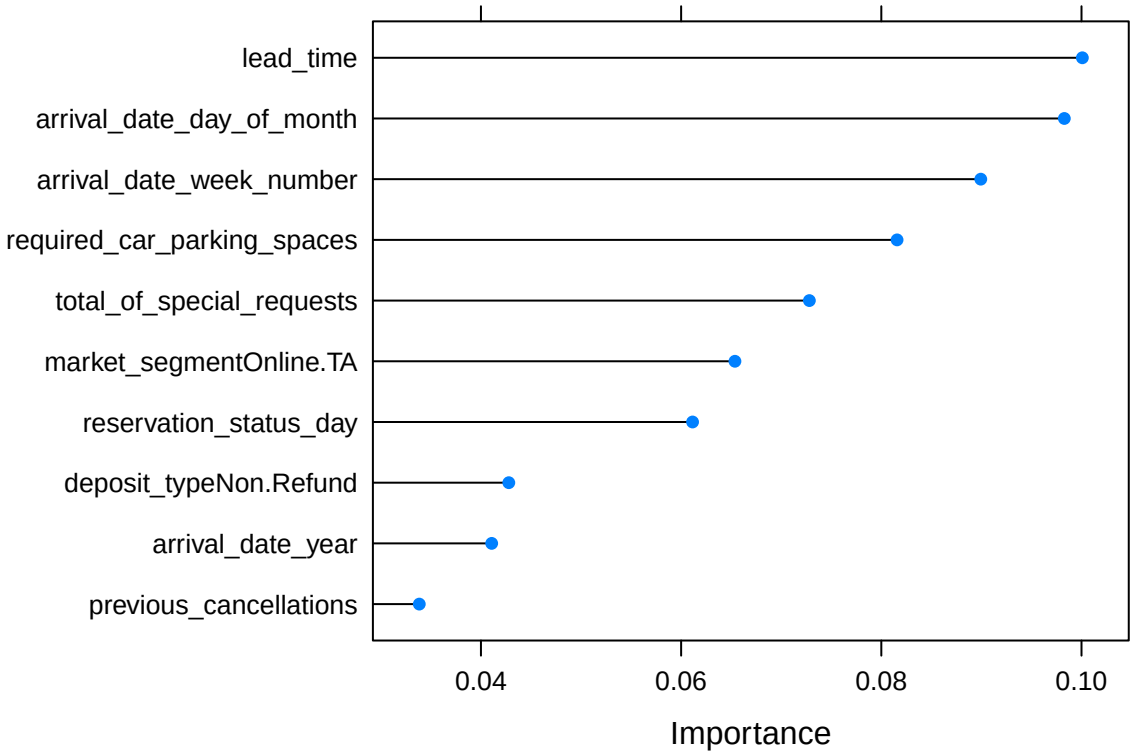
```
#stop parallel processing
stopCluster(cl)

registerDoSEQ()
 model_gbm
```

```
## eXtreme Gradient Boosting
##
## 66749 samples
##    92 predictor
##     2 classes: 'is_canceled', 'notCanceled'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 33375, 33374
## Resampling results across tuning parameters:
##
##   eta    max_depth  nrounds  ROC        Sens        Spec
##   0.025  2           50      0.7779462  0.00668294  1.0000000
##   0.025  2          200      0.8161772  0.35354651  0.9550513
##   0.025  3           50      0.7957209  0.41113800  0.9282848
##   0.025  3          200      0.8505984  0.45770180  0.9494870
##   0.050  2           50      0.7972270  0.05346390  0.9998552
##   0.050  2          200      0.8658329  0.48628104  0.9639459
##   0.050  3           50      0.8157474  0.40103327  0.9414612
##   0.050  3          200      0.9123096  0.61662639  0.9780117
##
## Tuning parameter 'gamma' was held constant at a value of 0
## Tuning
##
## Tuning parameter 'min_child_weight' was held constant at a value of 1
##
## Tuning parameter 'subsample' was held constant at a value of 1
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were nrounds = 200, max_depth = 3, eta
##  = 0.05, gamma = 0, colsample_bytree = 1, min_child_weight = 1 and subsample
##  = 1.
```

```
#plotting the model
plot(model_gbm)
```



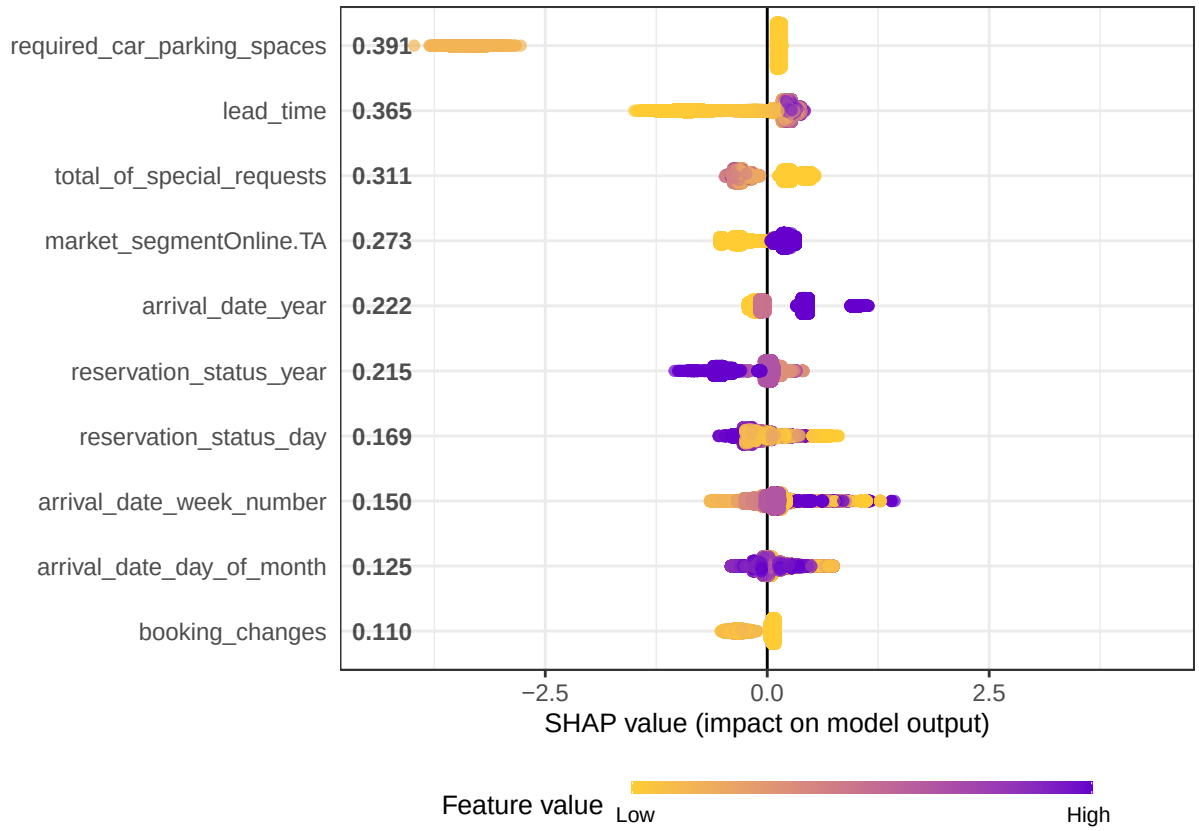

```
#plotting the model
plot(varImp(model_gbm, scale=FALSE), top = 10)
```




```
#plotting the shap model
library(SHAPforxgboost)
```

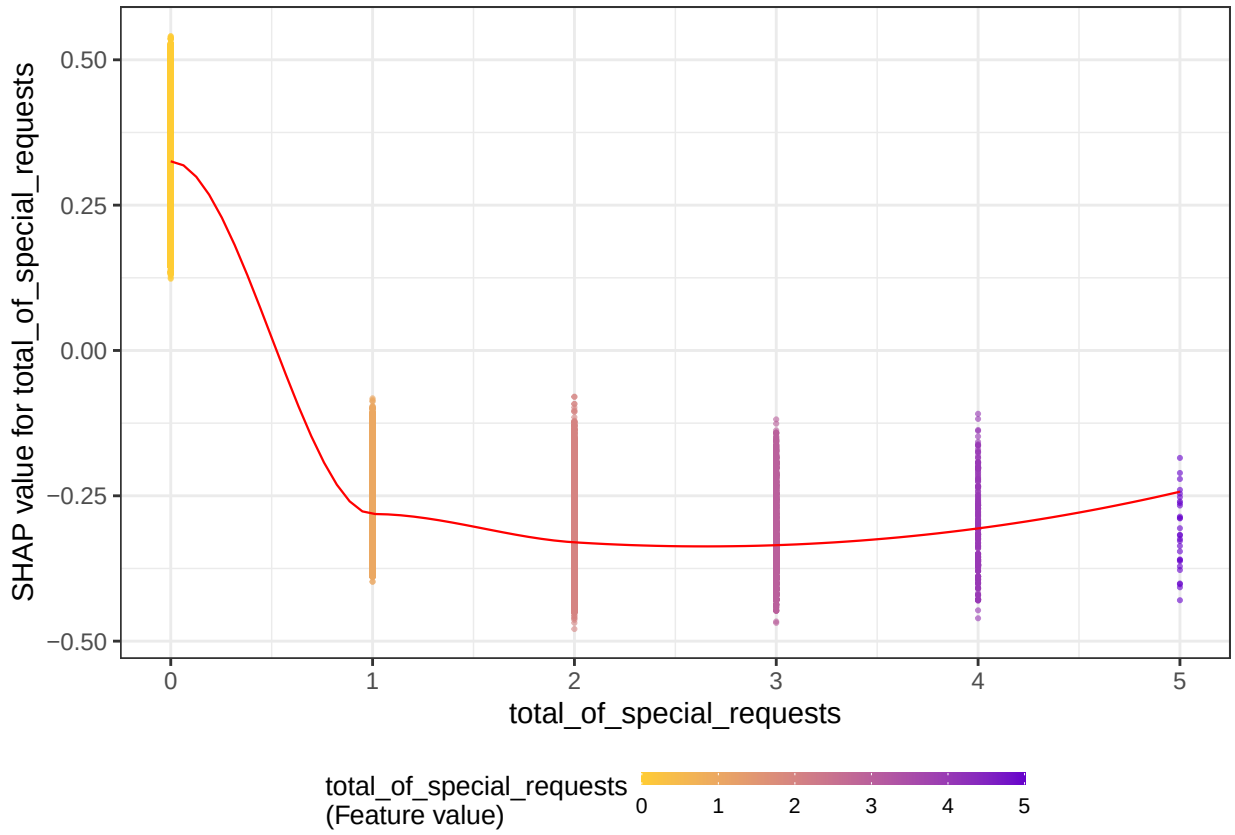
```
## Warning: package 'SHAPforxgboost' was built under R version 4.2.3
```

```
data<- subset(hos_train, select = -c(is_canceled))
Xdata <- as.matrix(data)
shap <- shap.prep(model_gbm$finalModel, X_train = Xdata, top_n = 10)

# SHAP importance plot
shap.plot.summary(shap)
```



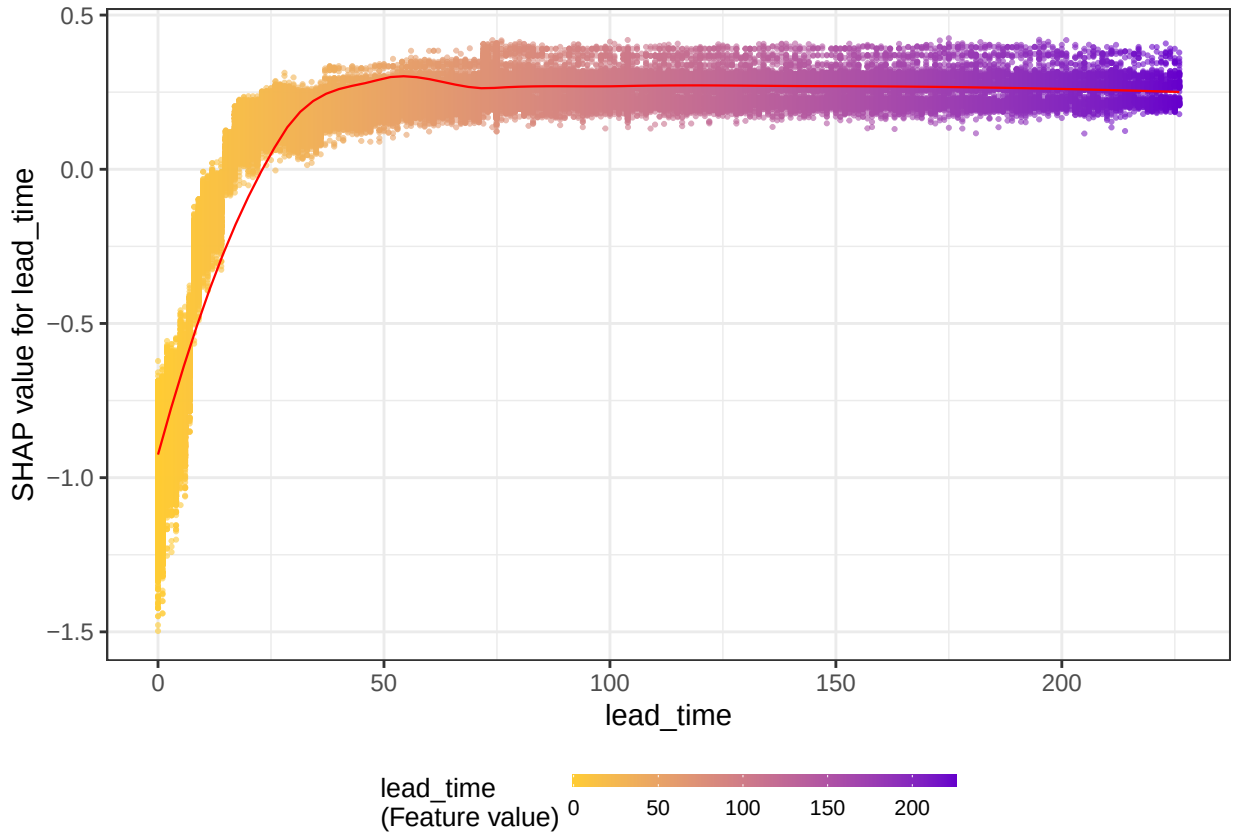

```
shap.plot.dependence(shap, x = "total_of_special_requests",
                     color_feature = "total_of_special_requests")
```

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : pseudoinverse used at -0.025

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : neighborhood radius 1.025

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : reciprocal condition number 5.1271e-26

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
## : There are other near singularities as well. 1
```




```
shap.plot.dependence(shap, x = "lead_time",
                     color_feature = "lead_time")
```

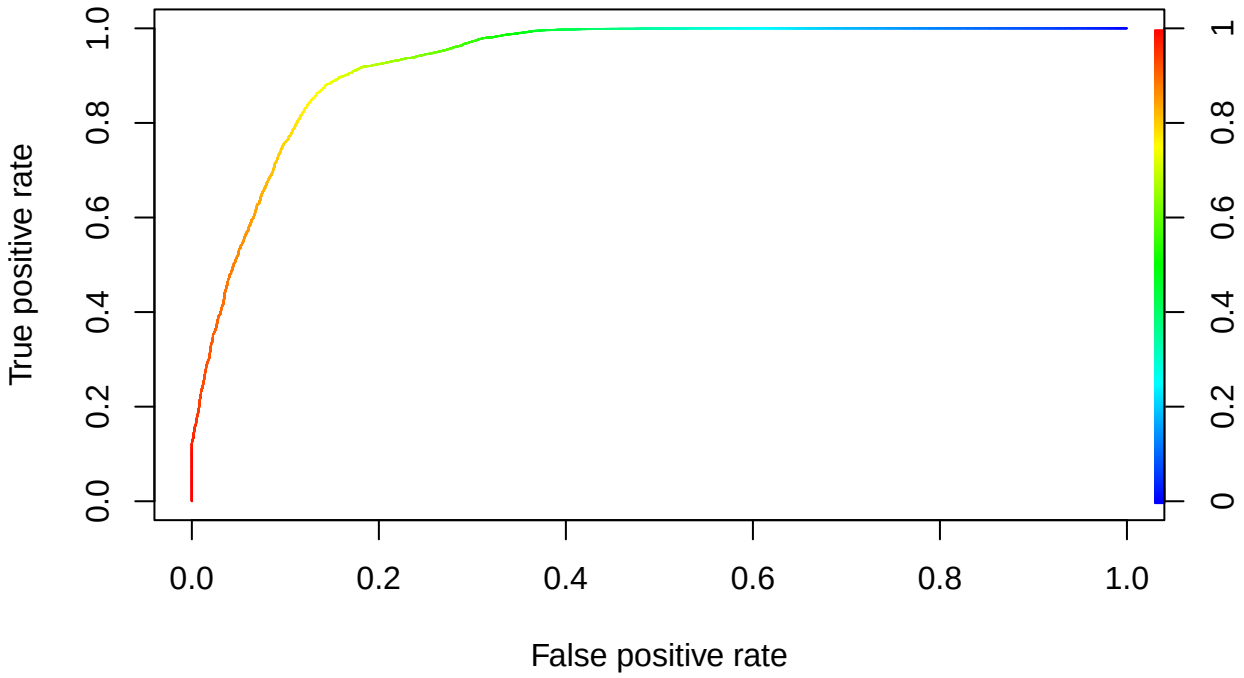
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.2.3
```

```
hos_prob<- predict(model_gbm, hos_test, type="prob")
pred = prediction(hos_prob[,2], hos_test$is_canceled,label.ordering =c("is_canceled","notCanceled"))
perf = performance(pred, "tpr", "fpr")

plot(perf, colorize=TRUE)
```




```
#Printing the AUC
slot(performance(pred, "auc"), "y.values")[[1]]
```

```
## [1] 0.93021
```