Flight Satisfaction Prediction

John Wilson

6/22/2020

Executive Summary

Flying continues to be the preferred method of travel for most but competition is on the rise amongst airliines. It is convenient for consumers to be able to predict their satisfaction with their flight choice and for airlines to know where they have opportunies for improvement in getting market share. This analysis objective is to take some variables that are present in a dataset to predict customer satisfaction with their flight without having every customer fill out a questionaire. The dataset was originally downloaded from kaggle which provided ~130,000 reviews of 23 variables to determine whether the customer was satisfied or neutral/dissatisfied in 2 files (test and train). 7 models were fitted to a training set of data then checked for accuracy on a test set of data before finally the best model being applied to the validation set. The QDA model was the best fit from the training phase and when applied to the validation set resulted in an accuracy of 0.749.

Data Preparation

When downloading the data for this analysis it came in two files(test and train) from the kaggle website. The test data was saved for the validation of the final model and the train set was labeled as practice. Before proceeding with further dissection of the practice dataset into test and train sets, the structure was analyzed to see class and variables available.

```
library(tidyverse)

## -- Attaching packages ---------------------------------------------------
---------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------
---------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(dslabs)
library(broom)
library(caret)

## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

library(purrr)
library(knitr)
library(tinytex)
library(rpart)
library(data.table)

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose

options(digits = 3)


# Flight Satisfaction dataset:
# https://www.kaggle.com/teejmahal20/airline-passenger-satisfaction/data

dl <- tempfile()
download.file("https://github.com/john-h-wilson09/Flight-
Satisfaction/archive/master.zip", dl)

validation <- read.csv(unzip(dl,"Flight-Satisfaction-master/test.csv"))
practice <- read.csv(unzip(dl,"Flight-Satisfaction-master/train.csv"))
rm(dl)

str(practice)

## 'data.frame':    103904 obs. of  25 variables:
##  $ X                                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ id                               : int  70172 5047 110028 24026 119299
111157 82113 96462 79485 65725 ...
##  $ Gender                           : Factor w/ 2 levels "Female","Male":
2 2 1 1 2 1 2 1 1 2 ...
##  $ Customer.Type                    : Factor w/ 2 levels "disloyal
Customer",..: 2 1 2 2 2 2 2 2 2 1 ...
##  $ Age                              : int  13 25 26 25 61 26 47 52 41 20
...
##  $ Type.of.Travel                   : Factor w/ 2 levels "Business
```

```
travel",..: 2 1 1 1 1 2 2 1 1 1 ...
##  $ Class                            : Factor w/ 3 levels
"Business","Eco",..: 3 1 1 1 1 2 2 1 1 2 ...
##  $ Flight.Distance                  : int  460 235 1142 562 214 1180 1276
2035 853 1061 ...
##  $ Inflight.wifi.service            : int  3 3 2 2 3 3 2 4 1 3 ...
##  $ Departure.Arrival.time.convenient: int  4 2 2 5 3 4 4 3 2 3 ...
##  $ Ease.of.Online.booking           : int  3 3 2 5 3 2 2 4 2 3 ...
##  $ Gate.location                    : int  1 3 2 5 3 1 3 4 2 4 ...
##  $ Food.and.drink                   : int  5 1 5 2 4 1 2 5 4 2 ...
##  $ Online.boarding                  : int  3 3 5 2 5 2 2 5 3 3 ...
##  $ Seat.comfort                     : int  5 1 5 2 5 1 2 5 3 3 ...
##  $ Inflight.entertainment           : int  5 1 5 2 3 1 2 5 1 2 ...
##  $ On.board.service                 : int  4 1 4 2 3 3 3 5 1 2 ...
##  $ Leg.room.service                 : int  3 5 3 5 4 4 3 5 2 3 ...
##  $ Baggage.handling                 : int  4 3 4 3 4 4 4 5 1 4 ...
##  $ Checkin.service                  : int  4 1 4 1 3 4 3 4 4 4 ...
##  $ Inflight.service                 : int  5 4 4 4 3 4 5 5 1 3 ...
##  $ Cleanliness                      : int  5 1 5 2 3 1 2 4 2 2 ...
##  $ Departure.Delay.in.Minutes       : int  25 1 0 11 0 0 9 4 0 0 ...
##  $ Arrival.Delay.in.Minutes         : num  18 6 0 9 0 0 23 0 0 0 ...
##  $ satisfaction                     : Factor w/ 2 levels "neutral or
dissatisfied",..: 1 1 2 1 2 1 1 2 1 1 ...
```

It was decided to remove customer inputs as they would basically tell the customer’s satisfaction and just pick the objective variables so an airline or customer could predict their satisfaction prior to even taking the flight.

```
# Selecting the columns to keep for modeling
validation <- validation %>% dplyr::select(satisfaction, Gender, Age, Class,
Arrival.Delay.in.Minutes,
                                    Flight.Distance)
practice <- practice %>% dplyr::select(satisfaction, Gender, Age, Class,
Arrival.Delay.in.Minutes,
                                           Flight.Distance)
```

As the classes are noted above, most regression modeling methods are looking for numeric and factor class types so the classes were modified at this point to prepare for analysis. The int classes were turned into num and the Arrival Delay variable was changed from num into factors of levels late/ontime. The altration of Arrival Delay to just ontime or late was for simplification in calculations and often customers just want to know what percentage can they expect to be on time and not so much how many minutes are the flights late. This change in Arrival Delay left some NA values which were then filtered out.

```
# Changing arrival status to factor of levels late or ontime
validation$Arrival.Delay.in.Minutes <-
as.factor(ifelse(validation$Arrival.Delay.in.Minutes>0, "Late","OnTime"))
validation <- validation %>%
filter(!(is.na(validation$Arrival.Delay.in.Minutes))) #remove NA values
practice$Arrival.Delay.in.Minutes <-
```

```
as.factor(ifelse(practice$Arrival.Delay.in.Minutes>0, "Late","OnTime"))
practice <- practice %>% filter(!(is.na(practice$Arrival.Delay.in.Minutes)))
#remove NA values

# Change class to num from int
num <- c(3,6)
practice[,num] <- apply(practice[,num],2,as.numeric)
validation[,num] <- apply(validation[,num],2,as.numeric)
```

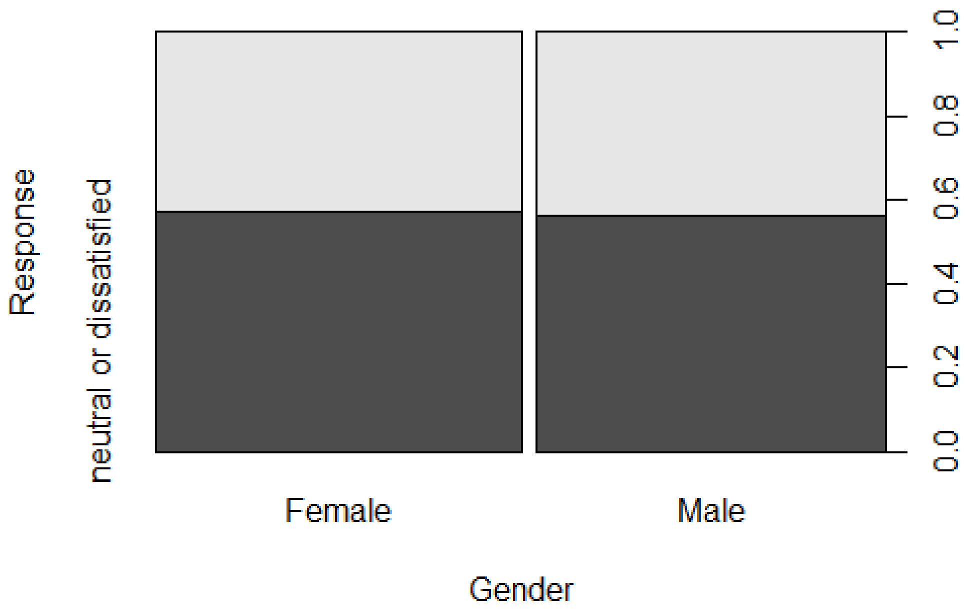
The practice set was then dissected into actual train and test sets for modeling purposes.

```
# Form test and train set within the practice dataset
set.seed(1, sample.kind = "Rounding")
test_index <-
createDataPartition(practice$satisfaction,times=1,p=0.5,list=FALSE)
train_set <- practice[-test_index,]
test_set <- practice[test_index,]
```

Technical Analysis

The first model performed was the LDA method due to its speed of calculation. From those results it is noted that ticket class was most important and the gender variable plays no role in the model (reflected in the varImp function and the bar chart).

```
# Regressions
lda_fit <- train(satisfaction ~ ., method = "lda", data=train_set)
lda_preds <- predict(lda_fit, test_set)
lda_acc <- mean(lda_preds==test_set$satisfaction)

varImp(lda_fit) #shows gender plays no influence in satisfaction so gender
will be removed to increase calc speed

## ROC curve variable importance
##
##                              Importance
## Class                             100.0
## Flight.Distance                    60.2
## Age                                32.8
## Arrival.Delay.in.Minutes           15.7
## Gender                              0.0

plot(train_set$Gender,train_set$satisfaction,xlab="Gender",ylab="Response")
```

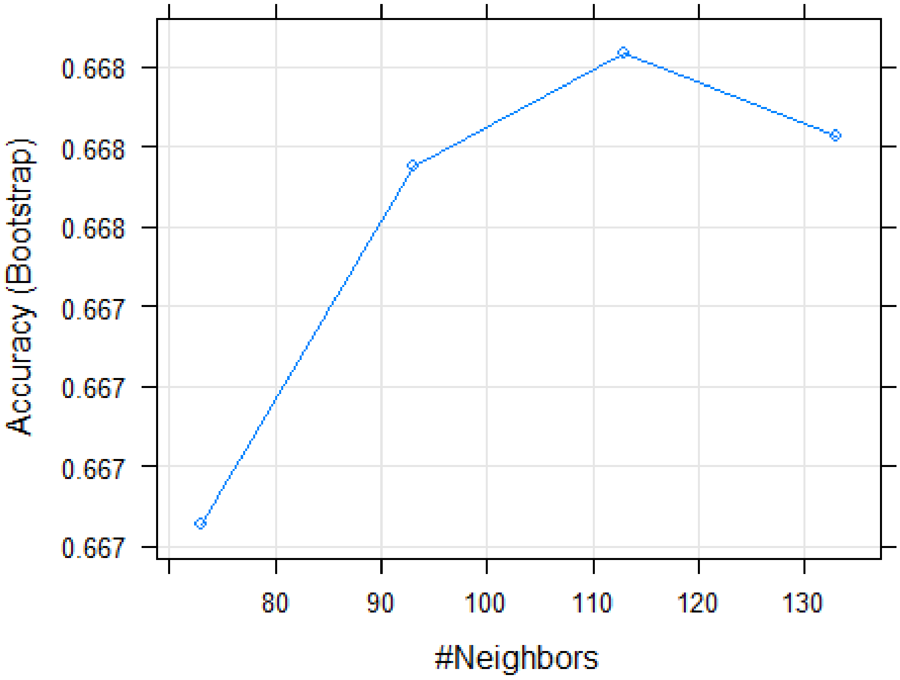


The gender column was removed from the train dataset and analysis was continued. A function was developed to run models for all the other methods that receive both the numeric and factor inputs in one step by mapping in the methods. Loess model was ran separately on only the numeric variables to create predictions.

```
train_set <- train_set[-2] #removed gender variable

# Regressions
models = c("glm","qda","rpart")
model_acc <- map(models, function(mod){
  fit <- train(satisfaction ~ ., method = mod, data=train_set)
  preds <- predict(fit,test_set)
  mean(preds==test_set$satisfaction)
  })

loe_fit <- train(satisfaction ~ Flight.Distance+Age, method = "gamLoess",
data=train_set)

## Loading required package: gam

## Loading required package: splines

## Loading required package: foreach

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

## Loaded gam 1.16.1

loe_preds <- predict(loe_fit,test_set)
loe_acc <- mean(loe_preds==test_set$satisfaction)
```

The knn method took the longest on computation but was ran separately to find the optimum k value. The range was determined by taking the sqrt(N) as this is a large dataset, which resulted in a target k=113. Then a few data points were selected around it to confirm the selection as seen in the k value vs accuracy plot. More points could be run but would require more computing power and time.

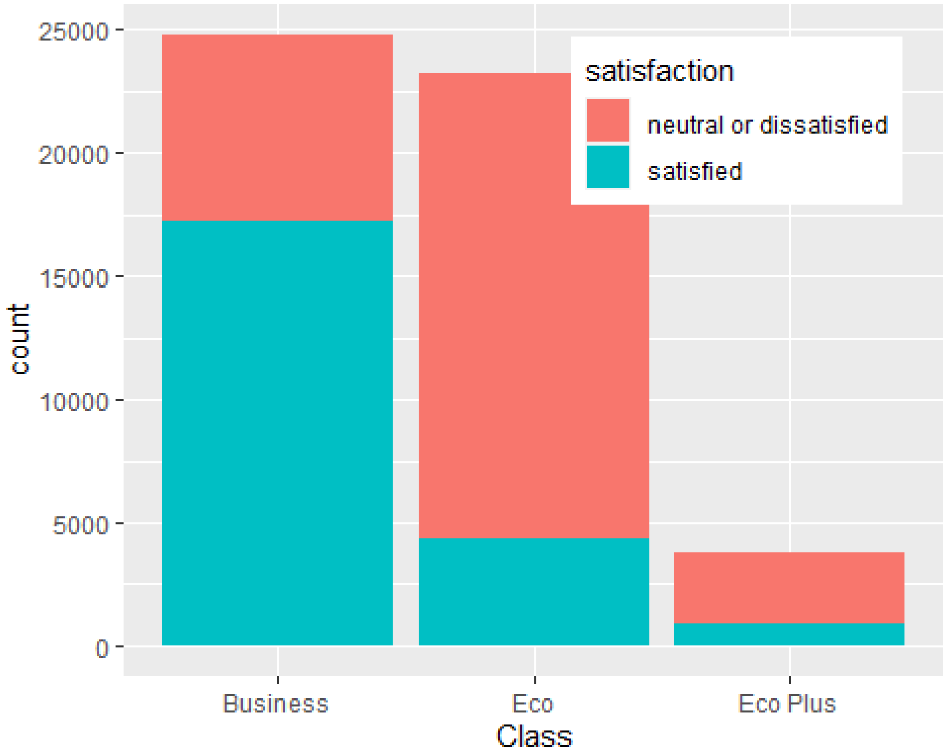
```
# Will take a few minutes to run
knn_fit <- train(satisfaction ~ ., method = "knn", data=train_set, tuneGrid =
data.frame(k=seq(73,133,20)))
knn_preds <- predict(knn_fit,test_set)
knn_acc <- mean(knn_preds==test_set$satisfaction)

plot(knn_fit)
```



As stated previously, the ticket class was the most important variable in our model calculations. By the bar chart based on the training set, it can be seen that Business class is very likely to be satisfied while the other classes are more likely to be neutral/dissatisfied.

This logic was applied to the test set as if a customer was in business class they would be satisfied and other classes would be noted as neutral/dissatisfied.

```
class_preds <- ifelse(test_set$Class=="Business","satisfied","neutral or
dissatisfied")
class_acc <- mean(class_preds==test_set$satisfaction)

train_set %>% ggplot(aes(Class,fill=satisfaction),lab="Class") + geom_bar() +
  theme(legend.position = c(.95, .95),
        legend.justification = c("right", "top")) #Business likely to be
dissatisfied/neutral
```



Results

Outputs from the models were similiar in accuracy results but yielded the QDA model to be the best fit to the training set then confirmed with test data. The accuracy achieved was 0.754 which was chosen to apply to the validation set.

```
# Model Results
Results <- data.frame(Models =
c("LDA","GLM","QDA","Rpart","Loess","Knn","Class Pred"),
                      Accuracy =
c(lda_acc,model_acc[[1]],model_acc[[2]],model_acc[[3]],
                                   loe_acc,knn_acc,class_acc)) %>%
arrange(desc(Accuracy))

kable(Results)
```

Models	Accuracy
QDA	0.754
GLM	0.753
LDA	0.752
Rpart	0.752
Class Pred	0.752
Loess	0.674
Knn	0.671

```
# Validation - QDA was most accurate model
qda_fit <- train(satisfaction ~  ., method = "qda", data=train_set)
val_preds <- predict(qda_fit,validation)
val_acc <- mean(val_preds==validation$satisfaction)
```

The QDA model applied to the validation dataset resulted in an accuracy of 0.749.

Conclusion

Overall the analysis performed consistently around 75% accuracy. 7 models were complete for this analysis that performed in close proximity on accuracy. Items that could further improve the models would be addition of airlines and jets used for each review. A limitation to getting a better idea of satisfaction was the grouping of neutral and dissatified together in the grading rubric. Other thoughts would be to just use a numeric score of 1 to 5 so a better sense of bias/effects that are involved in ratings (flight equipment, consumer and airlines). However, the final accuracy acheieved with the available data on the QDA model was 0.749.