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 0.7884. All the inputs were utilized as a check point in an alternate calculation that yielded a maximum accuracy of 0.865. This increased the time to compute the models for an increase of ~7% accuracy.

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()
```

```r
library(dslabs)
library(broom)
library(caret)
```

```
## Loading required package: lattice
```

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

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

```r
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
```

```r
options(digits = 4)


# 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 ...
```

Technical Analysis

All Inputs Method

To reflect the accuracy if all available variables were utilized the following model was created. The Arrival Delay variable was changed from num into factors of levels late/ontime. The alteration 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 the flights are late. All the NA values were removed to allow calculations to be performed.

```
# 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"))
practice$Arrival.Delay.in.Minutes <-
as.factor(ifelse(practice$Arrival.Delay.in.Minutes>0, "Late","OnTime"))

# Remove all NA values from data tables
practice2 <- na.omit(setDT(practice), cols = c(1:25))
```

```
validation2 <- na.omit(setDT(validation), cols = c(1:25))

# Form test and train set within the practice dataset
set.seed(1, sample.kind = "Rounding")

## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
sampler
## used

test2_index <-
createDataPartition(practice2$satisfaction,times=1,p=0.5,list=FALSE)
train2_set <- practice2[-test2_index,]
test2_set <- practice2[test2_index,]
```

LDA was run first for speed and variable importance was determined. Amongst all variables available, the most important variable was Online.boarding.

```
# Regressions
lda2_fit <- train(satisfaction ~ ., method = "lda", data=train2_set)
lda2_pred <- predict(lda2_fit, test2_set)
lda2_acc <- mean(lda2_pred==test2_set$satisfaction)

varImp(lda2_fit) #shows online.boarding was most important

## ROC curve variable importance
##
##   only 20 most important variables shown (out of 24)
##
##                                    Importance
## Online.boarding                        100.00
## Class                                   80.13
## Inflight.entertainment                  71.66
## Type.of.Travel                          67.39
## Seat.comfort                            63.71
## On.board.service                        59.53
## Leg.room.service                        57.73
## Cleanliness                             53.75
## Inflight.wifi.service                   52.87
## Flight.Distance                         48.79
## Baggage.handling                        48.52
## Inflight.service                        47.36
## Checkin.service                         41.56
## Food.and.drink                          36.79
## Ease.of.Online.booking                  32.57
## Age                                     27.21
## Customer.Type                           22.87
## Arrival.Delay.in.Minutes                13.81
## Departure.Delay.in.Minutes              10.75
## Departure.Arrival.time.convenient        8.81
```

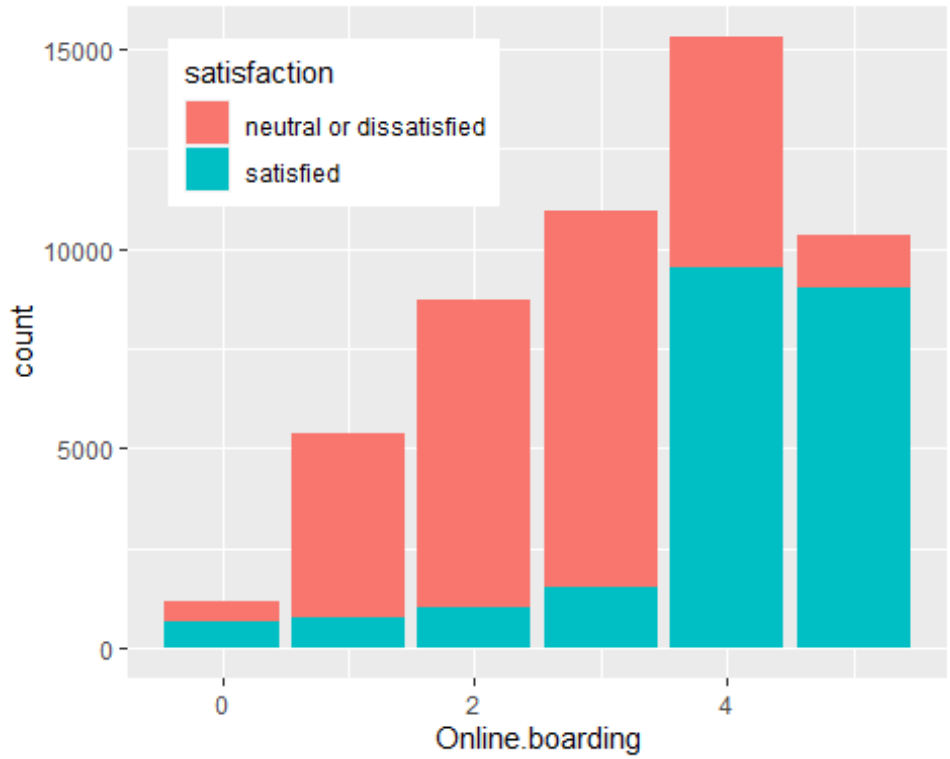
The remaining methods were performed in a single section. The knn method took the longest on computation and the k value was determined by taking the sqrt(N)/2 as this is a large dataset with many variables, which resulted in a target k=113. In a second, simplified analysis the k value is tuned to show confirmation of the estimate. Random forest was not performed due to hardware limitations.

```
mods = c("glm","qda","rpart")
mod_acc <- map(mods, function(mod){
  fit2 <- train(satisfaction ~ ., method = mod, data=train2_set)
  pred <- predict(fit2,test2_set)
  mean(pred==test2_set$satisfaction)
  })

loess_fit <- train(satisfaction ~
Flight.Distance+Age+Online.boarding+On.board.service+
                     Inflight.entertainment+Seat.comfort, method =
"gamLoess", data=train2_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

loess_pred <- predict(loess_fit,test2_set)
loess_acc <- mean(loess_pred==test2_set$satisfaction)

# Will take a few minutes to run
knn2_fit <- train(satisfaction ~ ., method = "knn", data=train2_set,
                  tuneGrid = data.frame(k=113))
knn2_preds <- predict(knn2_fit,test2_set)
knn2_acc <- mean(knn2_preds==test2_set$satisfaction)
```

As stated previously, the online boardiing was the most important variable in our model calculations. By the bar chart based on the training set, it can be seen that a rating of 4 or better was likely to be satisfied. This logic was applied to the test set as if a customer rated online boarding as a 4 or higher as satisfied and other ratings would be noted as neutral/dissatisfied.

```
train2_set %>% ggplot(aes(Online.boarding, fill=satisfaction), lab="Online
Boarding Rating") +
```

```
  geom_bar() + theme(legend.position = c(.05, .95), legend.justification =
c("left", "top"))
```




```
OnlineB_preds <- ifelse(test2_set$Online.boarding >3,"satisfied","neutral or
dissatisfied")
OnlineB_acc <- mean(OnlineB_preds==test2_set$satisfaction)
```

Simplified Model – Objective Inputs

It was decided to remove customer inputs as they would basically tell the customer's satisfaction based on ratings 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 <- validation2 %>% dplyr::select(satisfaction, Age, Class,
                              Arrival.Delay.in.Minutes, Type.of.Travel,
                              Customer.Type, Flight.Distance)
practice <- practice2 %>% dplyr::select(satisfaction, Age, Class,
                          Arrival.Delay.in.Minutes, Type.of.Travel,
                              Customer.Type, Flight.Distance)
```

The simplified 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,]
```

The first model performed was the LDA method as before and a variable importance function noted that ticket class was most important.

```
# 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 class of ticket most important

## ROC curve variable importance
##
##                            Importance
## Class                           100.0
## Type.of.Travel                   80.8
## Flight.Distance                  52.7
## Age                              20.2
## Customer.Type                    13.7
## Arrival.Delay.in.Minutes          0.0
```

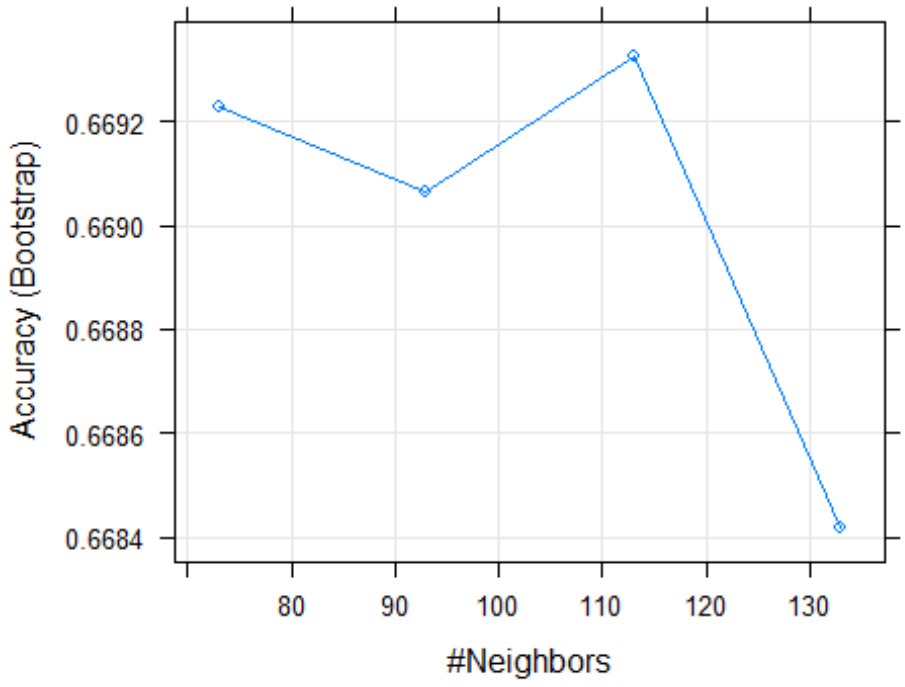
A function was developed to run models for all the other methods that receive varying classes of inputs in one step by mapping in the methods. Loess model was ran separately on only the int variables to create predictions.

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

The larger dataset in the previous Knn model was maxing out the hardware capacity available but this smaller set allowed tuning to occur to confirm the number used previously.

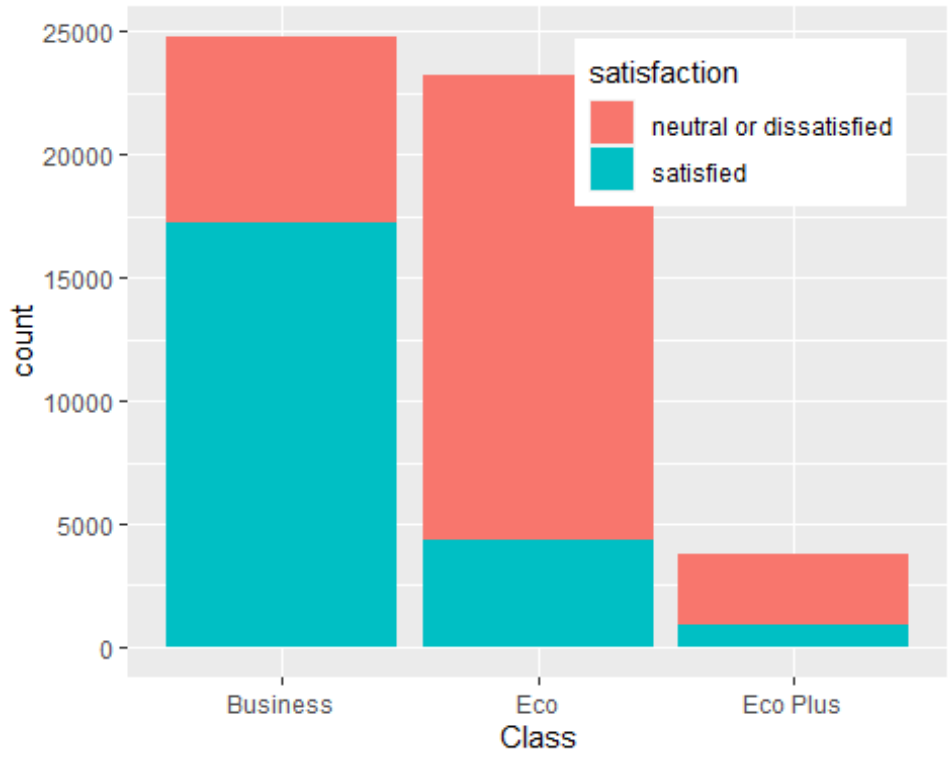
```
# Will take a few minutes to run
knn_fit <- train(satisfaction ~ ., method = "knn", data=train_set,
                 tuneGrid = data.frame(k=seq(73,133,20)))
knn_preds <- predict(knn_fit,test_set)
knn_acc <- mean(knn_preds==test_set$satisfaction)
```

```
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(test2_set$Class=="Business","satisfied","neutral or
dissatisfied")
class_acc <- mean(class_preds==test2_set$satisfaction)

train2_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
```



Outputs from the simplified models were accumulated into a single dataframe and reflected mostly similiar accuracies between 75 and 80%. However, the LDA model proved to be the best fit with an accuracy of `0.788`.

```
# 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
LDA	0.7884
GLM	0.7884
QDA	0.7819
Rpart	0.7750
Class Pred	0.7523
Loess	0.6744
Knn	0.6715

```
# Validation - LDA was most accurate model
val_preds <- predict(lda_fit,validation)
val_acc <- mean(val_preds==validation$satisfaction)
```

The simplified LDA model applied to the validation dataset resulted in an accuracy of 0.785.

Similiarly the compounded accuracies of the full dataset are displayed below with accuracies ranging mostly from 80 to 90%. Which confirms our suggestion that with ratings from customers, the prediction of satisfaction can be predicted even more accurately. The most correct model was GLM and was applied to the validation set.

```
Results2 <- data.frame(
  Models = c("LDA","GLM","QDA","Rpart","Loess","Knn","OnlineBoarding Pred"),
  Accuracy = c(lda2_acc,mod_acc[[1]],mod_acc[[2]],mod_acc[[3]],
               loess_acc,knn2_acc,OnlineB_acc)) %>% arrange(desc(Accuracy))

kable(Results2)
```

Models	Accuracy
GLM	0.8763
LDA	0.8723
QDA	0.8566
Rpart	0.8438
Loess	0.8180
OnlineBoarding Pred	0.7886
Knn	0.5899

```
# Validation - GLM was most accurate model
val2_fit <- train(satisfaction ~ ., method="glm", data=train2_set)
val2_preds <- predict(val2_fit,validation2)
val2_acc <- mean(val2_preds==validation2$satisfaction)
```

This GLM model when applied to validation data yields `0.874` accuracy.

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 0.785 when only incorporating object variables (inputs without customer feedback). Alternatively, when the calculations were ran with all available inputs gives an increased accuracy to `0.874` .