Final Project

Ravenna Collver, Isabelle Fang, and Purva Kapshikar

12/18/2020

Introduction

In recent years, reclining seats on flights have caused many confrontations, as many passengers consider the action to be rude. In our scenario, Berkeley Air has reached out to us, statistical analysts from UC Berkeley, to help them come up with a more quantitative way to determine whether they should implement a "no seat reclining" policy on all or some of its flights. They have given us responses from an online poll run by FiveThirtyEight and SurveyMonkey on August 29 and 30, 2014. In this dataset, some variables represent demographic information about the respondents, while most are answers to specific questions about which behaviors they consider rude on a flight. Our goal is to find a model using logistic regression that can be used to determine which flights should implement a "no seat reclining" policy based on the characteristics of passengers on that flight.

Regression Model

Because our outcome variable is binary, we are using logistic regression. In order to select the best model, we first removed variables that were found to be highly dependent with other variables. We then performed stepwise selection (both forward and backward) using AIC as our criterion and considering both interactive and non-interactive models. Most of these methods led us to the same model (see Additional Work for more information). Our final model gives the following equation, with the interpretation of each variable and coefficient in Table 1 below:

$$log\text{–}odds(recline\_elim) = X$$

Table 1: Interpretation of final model

Variable	Coefficient	Interpretation
recline_elim		Response variable which indicates whether or not the respondent supports banning reclining seats on airplanes entirely.
Intercept	-2.941	With all other variables having values of 0, the odds that a flyer wants to eliminate seat reclining is $e^{-2.941}$ = 0.052 on average.
get_up	-0.382	The odds that a flyer wants to eliminate seat reclining is multiplied by $e^{-0.382}$ =0.682 on average for every additional time a respondent believes it is acceptable for a person not in the aisle seat to get up on a 6 hour flight from NYC to LA.
switch_family somewhat	0.181	The `switch_family` variable contains "no", "somewhat", and "very" based on how rude the respondent believes it is to ask to switch seats on a flight in order to be closer to family. Our estimates show that the average odds that a flyer wants to eliminate seat reclining is multiplied by 1, $e^{0.181}$ = 1.198, or $e^{2.189}$ = 8.926, respectively, based on their response. For example, those who think it is very rude to switch seats to be closer to family are about 9 times more likely on average to want to eliminate seat reclining than those who do not think it is rude.
switch_family very	2.189	
age_30-44	0.008	Our `age` variable is bracketed into 18-29, 30-44, 45-60, and >60 groups. Our estimates show that the odds that a flyer would want to
age_45-60	0.448	

age_>60	0.614	eliminate seat reclining are multiplied by 1, $e^{0.008} = 1.008$, $e^{0.448} = 1.565$, or $e^{0.614} = 1.847$, respectively, based on the passenger's age bracket. This shows that as age increases, so does the average desire to eliminate reclining seats.
height	0.048	This variable encodes the respondent's height in inches. The coefficient estimate shows that the odds that a flyer wants to eliminate seat reclining is multiplied by $(e^{0.048}) = 1.049$ on average for every additional inch in the respondent's height. This shows that taller passengers are more likely to eliminate seat reclining.
has_children_ under_18 TRUE	-0.488	The average odds that a flyer wants to eliminate seat reclining is multiplied by $e^{0.488} = 1.629$ if the respondent has children under 18.

Overall, our `somewhat_switch_family` variable has the largest estimated coefficient, meaning that respondents' views on if it is rude to switch seats to be closer to family is the characteristic with the biggest effect on if flights should implement a "no seat reclining" policy.

Discussion

We fit our model using a training subset of the dataset. In order to determine the predictive power of our model on future passengers, we found the misclassification rate on the reserved testing subset of our data. We used a threshold of 0.5 to classify whether the passenger would be in favor or opposed to eliminating reclining seats based on the estimated probabilities found in the model. Further discussion of our threshold values is in our Additional Work section. Using this threshold led to a misclassification rate of 0.261. This is somewhat better than randomly guessing: the area under our ROC curve was 0.697, which is greater than the value of 0.5 that we would have gotten had we randomly guessed.[1] We recognize that the misclassification rate of 0.261 is still somewhat high. However, as all of our methods of model selection led to the same final model, we believe that this is the best classifier given our data. Our model has a null deviance of 756.04 and a residual deviance of 703.71. This supports our misclassification rate finding that while this model does not explain a lot of the variation in the odds, it does do better than the intercept-only model. A Likelihood Ratio Test between our model and the intercept-only model confirms this with a p-value less than 0.0001.

One limitation of our analysis is that we are not sure how the data was collected. In order to take our model a step further and predict whether a new individual will support the reclining ban, we have to assume that the respondents to this survey are predictive of the general population. Since this data came from a SurveyMonkey audience, it is probably not a representative sample of all Americans. However, it may be predictive of people similar to the respondents: perhaps Americans 18+ who use the Internet. We also recognize that there is a possibility of bias in the data, especially as Berkeley Air has not been very explicit with us about their methodology. We briefly discuss some potential sources of bias at the end of our report..

[1] An ROC curve plots true positive rate (TPR) against false positive rate (FPR). A perfect classifier would have an ROC curve close to the top left and an area under the curve of 1, as it would have a TPR of 1and an FPR of 0.

Simpler Model

Our final model is good for predicting which passengers would support a ban on reclining seats; however, some of the included variables are hard to gauge without asking the passenger directly. Ideally, Berkeley Air would be able to automatically make a decision on the policy for each flight based on data it has prior to the flight. We assume that the airline would have information about each adult passenger's age as well as a rough idea of whether the passenger has a child under the age of 18. Therefore, we fit a secondary model using only those two variables, with interpretations and coefficients in Table 2:

$$log\text{–}odds(recline\_elim) = X$$

Table 2: Interpretation of simple model

Variable	Coefficient	Interpretation
Intercept	-1.02	With all other variables having values of 0, the odds that a flyer wants to eliminate seat reclining is $e^{-1.02} = 0.36$ on average.
age_30-44	0.05	The older the passenger, the more likely they will be to support a ban on reclining seats. The odds are multiplied by 1, $e^{0.05} = 1.05$, $e^{0.40} = 1.49$, or $e^{0.56} = 1.75$, depending on the passenger's age bracket.
age_45-60	0.40	
age_>60	0.56	
children_under_18 TRUE	-0.51	Passengers with children under the age of 18 are less likely to support the policy.

When we used this model to predict support among the test data, we got a misclassification rate of 0.30 and an area under the ROC curve of 0.59. This suggests that using age and the indicator of having children allows Berkeley Air to guess whether that person will support the ban somewhat better than completely randomly. The airline can use that information, gathered on every passenger on the plane, to determine whether a flight will implement a "no seat reclining" policy.

Conclusion

We found that age, height, whether a passenger has children under 18, the number of times a passenger believes it's rude to get up during a flight, and whether a passenger thinks it's rude to request a seat change to sit closer to a family member are somewhat predictive of whether a passenger is in support of eliminating reclining seats. This model is the best we could come up with given the many variables provided to us by Berkeley Air. However, a simple model using only age and the indicator of children under 18 has only a moderately worse misclassification rate and is significantly easier to use in practice. Therefore, our recommendation to the airline is to fit our simple model to determine how many people on the flight are likely to support the ban on reclining seats.

Additional Work

EDA

Our first step was to visualize our data. Our primary method of visualization for this report was mosaic plots, since the majority of our variables are categorical or binary. We also used conditional density plots to visualize the relationship between numeric data and our binary outcome. The plots we made for the five variables that made it in our final model are below:

Figure 1: Variables used in final model variables

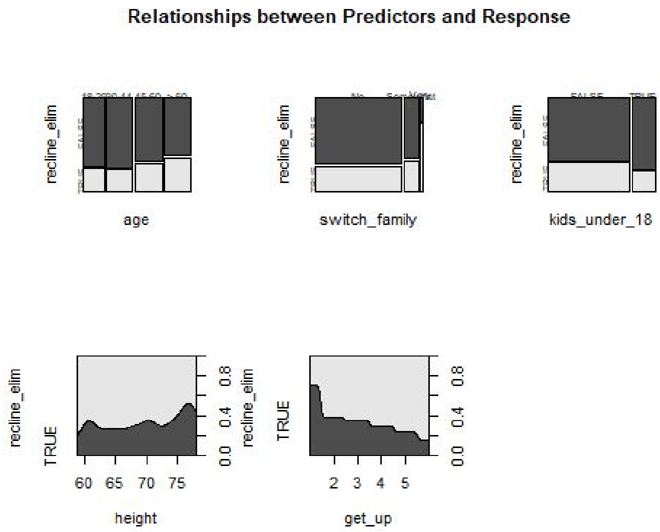
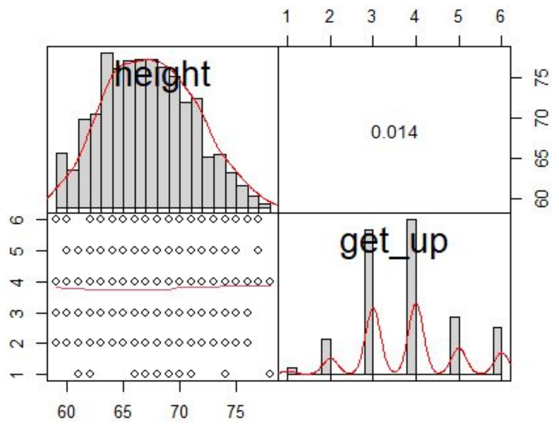


Figure 2: Correlation of numerical



These plots show that the signs of the coefficients in our model are reasonable: The likelihood of being in favor of the ban on reclining seats goes up with age, height, and how rude the passenger thinks it is to request a seat switch to be with family, and the likelihood goes down if the passenger has a child under the age of 18 and if the passenger thinks that it is rude to get up during a flight more times. We also see in Figure 2 that there is not a large correlation between the `height` and `get_up` variables, which justifies the lack of an interaction term for those two variables.

Our dataset had quite a few NA values, so we looked into some effects that they might have had on our model. 458 respondents out of 1040 had at least one NA value in their survey response. That is around 44.2% of our data, which we suspect is because 28 questions may be a hefty survey. The variable with the most NA values is `household_income`, with 329 people not responding. This could be due to privacy concerns or not knowing their household income. If `household_income` has a big effect on whether or not flyers would eliminate seat reclining, then we may not be able to uncover its full effect. The variables with the fewest NA values are age and gender, both with 33. These are both simple factual questions that don't require any thinking to respond to. Since we have more values in these variables, they may be overpowered when looking at their effects on our model.

Model Selection

During exploratory data analysis, we realized that there was quite a bit of collinearity. In order to choose the best model, we first had to resolve these issues. In order to determine which variables were dependent, we ran chi-squared tests of independence on every pair of variables. Once we had p-values from these tests, we found pairs and small groups of variables that were all dependent ($p < 0.05$) on each other. In forming these pairs or groups, we also considered which variables encoded similar opinions or traits. We then eliminated the variable or variables from each set that were the least likely to be dependent with our response variable. This cut the number of variables we considered from 27 to 13. While some dependency issues remained, we didn't want to continue cutting down variables too far, especially for pairs of variables that weren't obviously similar.

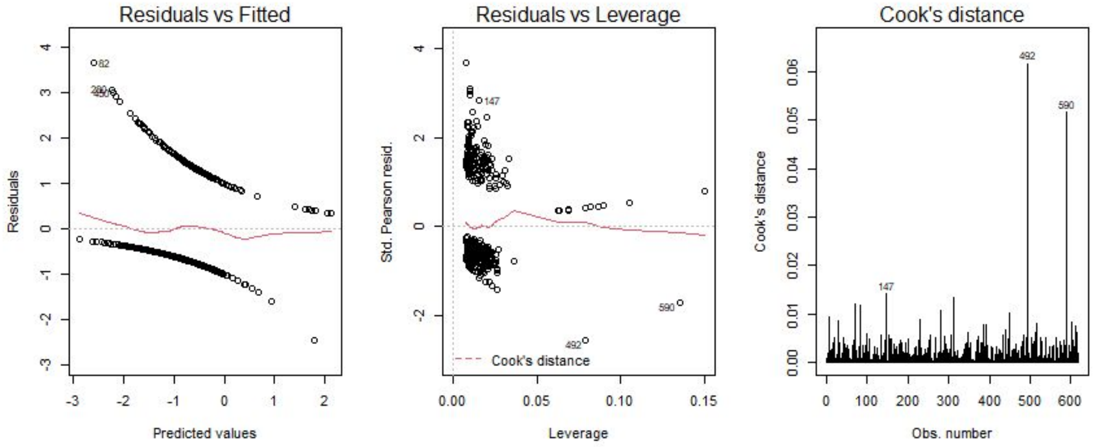
In order to choose the best model, we used stepwise selection with AIC as our criterion. To lessen the chance of finding a local minimum AIC, we ran both forward and backward selection and compared the results. When we conducted this selection over all models with only marginal effects, forward and backward selection found the same model. This basic model included `get_up`, `switch_family`, `age`, `height`, and `kids_under_18`.

We then looked at the relationships between these five explanatory variables using pairwise mosaic plots. We saw that interactions between some pairs of variables, such as `age` and `children_under_18` or `get_up` and `switch_family`, might be interesting to consider. However, searching over models with these interactions resulted in the same model as before. We also searched over all models with all possible interaction terms. Backwards selection in this case produced a very large and hard-to-interpret model, which we believe is because it got stuck in a local minimum for AIC. The model that forward selection found was again the same as the basic model above, so this is the model we chose as our final model.

Model Assessment

To assess our model, we looked at the deviance residual plot, the leverage plot, and the Cook's distance of each point. These plots revealed two data points with extremely unusual values of the predictors. They had Cook's distances of 1.67 and 1.71 , meaning that they had an extreme impact on the regression. Therefore, we removed these two points, assuming that they were respondents who didn't answer truthfully, given that their responses had leverages of 0.72 and 0.81. The plots for the final model are below:

Figure 3: Assessment of final model

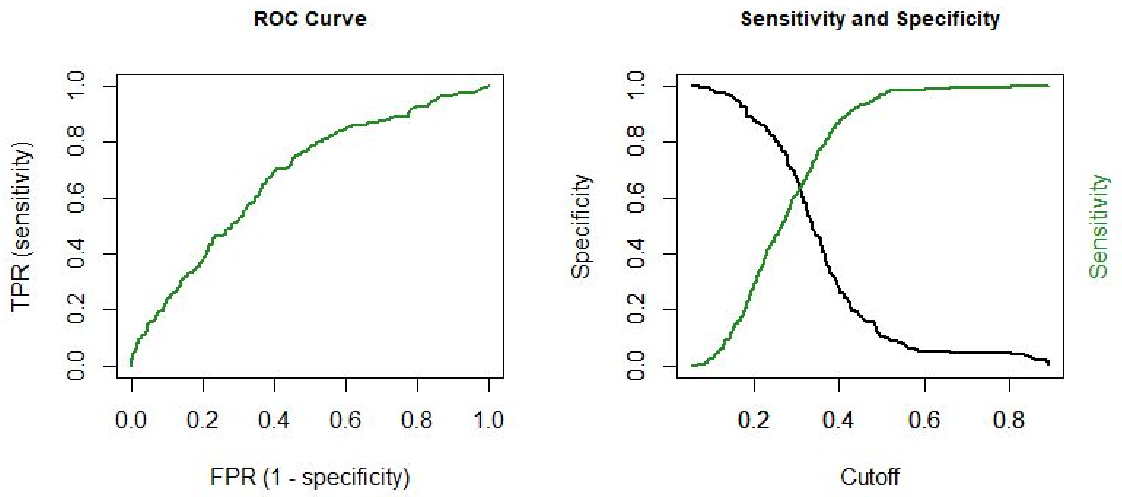


From these plots, we can see that no individual point is exerting undue influence on our model. While a few points have a larger Cook's Distance than the others, no distance is greater than 0.06, so the overall effect is small. There are also several points with a higher leverage, but none are larger than about 0.15. Also, these outlier points do not affect our results, since they do not also have large Cook's Distances.

Selecting Threshold for Classification

When evaluating our model on our test set, we decided to select a threshold value for classification by plotting the receiver operating characteristic (ROC) curve. This graph plots true and false positive rates for various choices of a threshold.

Figure 4: Determining optimal threshold value



We used Youden's J statistic, $J = sensitivity + specificity - 1$, and chose the threshold that yielded the highest J, which was $T = 0.294$. In this formula, sensitivity is equal to the true positive rate and sensitivity is equal to one minus the false positive rate.

However, using this threshold led to a misclassification rate of 0.369 on the test set, which is significantly higher than what we had got using the threshold of $T = 0.5$. We ultimately kept our original decision rule of using the 0.5 threshold that calculated a misclassification rate of 0.261 on the test set.

The following are a few comparisons for our two different threshold values:

Table 3: Numeric quantities calculated for different thresholds

	T = 0.5	T = 0.294
Accuracy	0.686	0.555
Precision	0.461	0.354
TPR	0.094	0.531

FPR	0.049	0.433

Generally, accuracy, or the proportion of points our classifier was able to classify correctly, is maximized with a threshold of 0.5. Precision, which penalizes false positives, tends to increase when the threshold increases, which explains why our precision for $T = 0.294$ is lower. However, there is a trade-off between precision and the true positive rate (TPR). TPR penalizes false negatives and increases when the threshold is lowered. We can see all of these relationships in the above table. Therefore, we had a choice between maximizing $sensitivity + specificity - 1$ by using 0.294 and maximizing precision and accuracy by using 0.5. We believe that it is most important to maximize accuracy and precision because we may lose customers if we act on false positives: If passengers are unhappy about this policy, they could choose to not fly with Berkeley Air. Thus, we want to prioritize true positives and negatives, which the threshold of 0.5 does.

Potential Sources of Bias

We considered that there could be potentially three types of bias present:

- Selection bias: Based on the demographics of respondents, this may not be occurring here. However, as the data is not from just their customers (some respondents have never flown before), our model might not be as applicable to Berkeley Air if their customers are quite different from the average airline passenger represented in this data.
- Response bias: There is some more personal data here, such as height and household income, which people may not always answer honestly or answer at all. Also, as some of the questions are somewhat similar, the ordering of questions could have influenced responses.
- Non-response bias: The poll was only offered online, so those who may not be as technologically adept may have decided not to participate.

References

"Deciding threshold for glm logistic regression model in R", Stack Overflow. (2014). https://stackoverflow.com/questions/23240182/deciding-threshold-for-glm-logistic-regression-model-in-r

Emerson, John W. and Walton A. Green. "Package 'gpairs'", Comprehensive R Archive Network. (2020). https://cran.r-project.org/web/packages/gpairs/gpairs.pdf

"Flying Etiquette Survey Data", FiveThirtyEight. (2014). https://github.com/fivethirtyeight/data/tree/master/flying-etiquette-survey

Hickey, Walt. "41 Percent Of Fliers Think You're Rude If You Recline Your Seat", FiveThirtyEight. (2014). https://fivethirtyeight.com/features/airplane-etiquette-recline-seat/

"How to remove ordering of the levels from factor variable in R?", Stack Overflow. (2013). https://stackoverflow.com/questions/17592524/how-to-remove-ordering-of-the-levels-from-factor-variable-in-r

"Package 'ROCR'", Comprehensive R Archive Network. (2020). https://cran.r-project.org/web/packages/ROCR/ROCR.pdf

Code Appendix

```r
knitr::opts_chunk$set(echo = FALSE)
library(tidyverse)
library(fivethirtyeight)
library(leaps)
library(MASS)
library(PerformanceAnalytics)
library(ROCR)


# load data
data("flying")
f <- flying

# change height to numeric inches
levels(f$height)[1] <- "4'11\""
levels(f$height)[20] <- "6'6\""
f <- f %>%
  separate(height, c("feet", "inches"), "'") %>%
  dplyr::mutate(inches = str_remove(inches, "\""),
                inches = (12 * as.numeric(feet)) + as.numeric(inches)) %>%
  dplyr::select(-feet)

# change get_up to numeric
levels(f$get_up) <- 0:5
f$get_up <- as.numeric(f$get_up)

# simplify column names
colnames(f)[c(1, 4, 5, 10, 11, 13:17)] <- c(
  "id","height", "kids_under_18", "recline_freq", "recline_oblig",
  "recline_elim", "switch_friends", "switch_family",
  "wake_bathroom", "wake_walk")

# make sure everything else is a factor
f$gender <- as.factor(f$gender)
f$location <- as.factor(f$location)
f$two_arm_rests <- as.factor(f$two_arm_rests)
f$middle_arm_rest <- as.factor(f$middle_arm_rest)
f$shade <- as.factor(f$shade)
f$kids_under_18 <- as.factor(f$kids_under_18)
f$recline_oblig <- as.factor(f$recline_oblig)
f$recline_elim <- as.factor(f$recline_elim)
f$electronics <- as.factor(f$electronics)
f$smoked <- as.factor(f$smoked)

# drop two outlying data points
f <- f %>% filter((id != 3432636547) &
                    (id != 3432169273))
```

```r
# create smaller data set, dropping col.s with high dependency
better_dat <- f %>%
  dplyr::select(-household_income, -switch_friends,
                -wake_walk, -wake_bathroom, -baby,
                -unruly_child,
                -middle_arm_rest, -shade,-talk_stranger,
                -unsold_seat, -smoked, -recline_freq,
                -recline_oblig, -recline_rude,
                -id) %>%
  na.omit()

# un-order the ordered factors
class(better_dat$age) <- "factor"
class(better_dat$education) <- "factor"
class(better_dat$frequency) <- "factor"
class(better_dat$switch_family) <- "factor"

# create test and training data sets
set.seed(15)
indexes = sample(1:nrow(better_dat), size=(3/4)*nrow(better_dat))
train <- better_dat[indexes,]
test <- better_dat[-indexes,]


# after model selection [see 'additional work'], we came up with:
final_mod <- glm(recline_elim ~ get_up + switch_family
                 + age + height + kids_under_18, train,
                 family = "binomial")


# information about our model
summary(final_mod)

# test against the null model
null <- glm(recline_elim ~ 1, train, family = "binomial")
anova(final_mod, null, test = "Chisq")

# find area under ROC curve (AUC)
predictions <- prediction(final_mod$fitted.values,
                          train$recline_elim)
auc <- performance(predictions, measure = "auc")
auc <- auc@y.values[[1]]


# simple model
simple <- glm(recline_elim ~ age + kids_under_18,
              train, family = "binomial")

# misclassification rate
```

```r
prob_fit_s <- predict(simple, newdata = test, type = "response")
y_hat_s <- prob_fit_s >= threshold
misclass_rate <- mean(y != y_hat, na.rm = TRUE)

# area under its ROC curve
predictions_s <- prediction(simple$fitted.values,
                          train$recline_elim)
auc_s <- performance(predictions_s, measure = "auc")
auc_s <- auc_s@y.values[[1]]


# plots of main 5 variables
par(mfrow = c(2, 3), oma = c(0, 0, 0.75, 0))
depvars <- c("age", "switch_family", "kids_under_18")
for (i in depvars){
  form2 <- paste("~", i, "+ recline_elim")
  mosaicplot(as.formula(form2), better_dat, color = TRUE,
             main = "")
}
cdplot(recline_elim ~ height, data=better_dat)
cdplot(recline_elim ~ get_up, data = better_dat)
title("Relationships between Predictors and Response",
      outer = TRUE)


# pairs plot of numeric variables
chart.Correlation(f[,c("height","get_up")], histogram=TRUE,
                  main = "Marginal Relationship of Numeric Predictors")


#sum of NA per column (variable)
NA.variable = sapply(f, function(x) sum(is.na(x)))
sort(NA.variable, decreasing = TRUE)


# assessing collinearity: making a table of Chisq test p-values
pvals <- expand.grid(x = colnames(f), y = colnames(f),
                     KEEP.OUT.ATTRS = FALSE)
pvals <- as.data.frame(pvals)
pvals$x <- as.character(pvals$x)
pvals$y <- as.character(pvals$y)

fnc <- function(xvec, yvec) {
  ps <- c()
  for(i in 1:length(xvec)) {
    x <- f[xvec[i]] %>% unlist()
    y <- f[yvec[i]] %>% unlist()
    p <- chisq.test(x, y, simulate.p.value = TRUE)$p.value
    ps <- c(ps, p)
  }
  return(ps)
}

pvals <- pvals %>%
  mutate(p = fnc(x, y)) %>%
  pivot_wider(names_from = y, values_from = p)


# model selection
## basic model selection
bigmod_1 <- glm(recline_elim ~ ., train,
                family = "binomial")
basemod_1 <- glm(recline_elim ~ 1, train,
                 family = "binomial")

fs_1 <- stepAIC(basemod_1, trace = FALSE, direction = "forward",
              scope = list(lower = basemod_1, upper = bigmod_1))
bs_1 <- stepAIC(bigmod_1, trace = FALSE, direction = "backward")
ss_1 <- stepAIC(fs_1, trace = FALSE, direction = "both")

## fully interactive
bigmod_2 <- glm(recline_elim ~ .*., train,
                family = "binomial")
basemod_2 <- glm(recline_elim ~ 1, train,
                 family = "binomial")

fs_2 <- stepAIC(basemod_2, trace = FALSE, direction = "forward",
              scope = list(lower = basemod_2, upper = bigmod_2))
bs_2 <- stepAIC(bigmod_2, trace = FALSE, direction = "backward")
ss_2 <- stepAIC(fs_2, trace = FALSE, direction = "both")


# finding various interaction sources
gpairs(data.frame(train$age, train$switch_family,
                  train$kids_under_18),
       mosaic.pars = list(shade =TRUE),
       outer.rot = c(0, 0))

# model selection with limited interaction terms
bigmod_3 <- glm(recline_elim ~ . * (age + switch_family), train,
                family = "binomial")
basemod_3 <- glm(recline_elim ~ 1, train,
                 family = "binomial")

fs_3 <- stepAIC(basemod_3, trace = FALSE, direction = "forward",
                scope = list(lower = basemod_3, upper = bigmod_3))
bs_3 <- stepAIC(bigmod_3, trace = FALSE, direction = "backward")
ss_3 <- stepAIC(fs_3, trace = FALSE, direction = "both")
```

```r
# Model assessment plots
par(mfrow = c(1, 3))
plot(final_mod, which = 1)
plot(final_mod, which = 5)
plot(final_mod, which = 4)


# ROC curve plots
predictions <- prediction(final_mod$fitted.values,
                          train$recline_elim)

par(mfrow=c(1,2), cex.main = 0.9)
perf <- performance(predictions, "tpr", "fpr")
plot(perf, lwd=2, col='forestgreen', main="ROC Curve",
     xlab="FPR (1 - specificity)", ylab="TPR (sensitivity)")

plot(unlist(performance(predictions, "sens")@x.values),
     unlist(performance(predictions, "sens")@y.values),
     type="l", lwd=2, ylab="Specificity", xlab="Cutoff",
     main="Sensitivity and Specificity", cex = 0.5)


par(new=TRUE)
plot(unlist(performance(predictions, "spec")@x.values),
unlist(performance(predictions, "spec")@y.values),
     type="l", lwd=2, col='forestgreen', ylab="", xlab="")
mtext("Sensitivity",side=4, padj=1, col='forestgreen')


# finding Youden's J-statistc
J_threshold <- function(predict, response) {
  perf <- ROCR::performance(ROCR::prediction(predict, response),
                            "sens", "spec")
  df <- data.frame(cut = perf@alpha.values[[1]],
                   sens = perf@x.values[[1]],
                   spec = perf@y.values[[1]])
  df[which.max(df$sens + df$spec), "cut"]
}


j_threshold = J_threshold(final_mod$fitted.values,
                          train$recline_elim)


# comparisons of threshold values
thresholds <- c(0.5, j_threshold)
results <- data.frame(`T = 0.5` = 1:5, `T = 0.294` = 1:5)
for (i in 1:2) {
  y_hat <- prob_fit >= thresholds[i]
  tp <- sum((y == y_hat & y == TRUE))   # true pos
  tn <- sum((y == y_hat & y == FALSE))  # true neg
  fp <- sum((y != y_hat & y == FALSE))  # false pos
  fn <- sum((y != y_hat & y == TRUE))   # false neg

  accuracy <- (tp + tn) / (tp + tn + fp + fn)
  precision <- tp / (tp + fp)
  recall <- tp / (tp + fn)
  fpr <- fp / (fp + tn)
  misclass_rate <- mean(y != y_hat, na.rm = TRUE)

  results[,i] <- c(accuracy, precision, recall, fpr, misclass_rate)
}


final = knitr::all_labels()
```