Homework 7: Predicting Health using Behavioral Risk Factors

Dr. Borselli

4/13/2022

The problem

The problem we'll solve is a binary classification task with the goal of predicting an individual's health. The features are socioeconomic and lifestyle characteristics of individuals and the label is 0 for poor health and 1 for good health. This dataset was collected by the Centers for Disease Control and Prevention.

The Behavioral Risk Factor Surveillance System (BRFSS) is the nation's premier system of health-related telephone surveys that collect state data about U.S. residents regarding their health-related risk behaviors, chronic health conditions, and use of preventive services. Established in 1984 with 15 states, BRFSS now collects data in all 50 states as well as the District of Columbia and three U.S. territories. BRFSS completes more than 400,000 adult interviews each year, making it the largest continuously conducted health survey system in the world.

The objective of the BRFSS is to collect uniform, state-specific data on preventive health practices and risk behaviors that are linked to chronic diseases, injuries, and preventable infectious diseases in the adult population. Factors assessed by the BRFSS include tobacco use, health care coverage, HIV/AIDS knowledge or prevention, physical activity, and fruit and vegetable consumption. Data are collected from a random sample of adults (one per household) through a telephone survey.

This codebook (https://www.cdc.gov/brfss/annual_data/2015/pdf/codebook15_llcp.pdf) has info on each of the variables.

1) We will be trying to predict the _RFHLTH variable. Note that R will change this variable to be called X_RFHLTH since it doesn't like the fact that it starts with an underscore. Look in the codebook for more information. What does this variable measure? How are the different responses coded? Should we drop some rows based upon this?

_RFHLTH measures self-reported quality of health, either a 1 for good health or 2 for poor health. Missing data or responses were inputed as 9, so these should be dropped during the cleaning process.

2) Read in the data and keep only the columns 'HLTHPLN1', 'PERSDOC2', 'MEDCOST', 'CHECKUP1', 'SEX', 'MARITAL', 'EDUCA', 'RENTHOM1', 'VETERAN3', 'EMPLOY1', 'CHILDREN', 'INCOME2', 'INTERNET', 'SMOKE100', 'USENOW3', 'ALCDAY5', 'FRUITJU1', 'FRUIT1', 'FLUSHOT6', 'X_RFHLTH'. This last one will be our target variable.

In order to get a file that was easier to work with for students, I pre-cleaned the original 516 MB file into a more manageable CSV file that still required some pre-processing.

```
#keep = c( 'HLTHPLN1', 'PERSDOC2', 'MEDCOST', 'CHECKUP1', 'SEX', 'MARITAL', 'EDUCA',
'RENTHOM1', 'VETERAN3', 'EMPLOY1', 'CHILDREN', 'INCOME2', 'INTERNET', 'SMOKE100',
'USENOW3', 'ALCDAY5', 'FRUITJU1', 'FRUIT1', 'FLUSHOT6', 'X_RFHLTH')

#data = data[keep]
```

```
data = read.csv('2015new.csv')
data = data[,-1]
```

3) Perform data cleaning. Note that some variables are coded with things like 9 = no response. You want to delete those rows or possibly perform mean imputation if applicable, by replacing those "no response"s or "unsure"s with the mean of the column. Most columns should also be factors. Check to make sure everything is in good shape before going on. Include comments in your code explaining any changes you make.

```
#HLTHPLN1 - Do you have any kind of health care coverage, including health insurance,
prepaid plans such as HMOs, or government plans such as Medicare, or Indian Health Service?
 #Answers: yes(1) or no(2)
 #removed 7 and 9 bc they meant unsure or refused
data <- subset(data, data$HLTHPLN1 != 7 & data$HLTHPLN1 != 9)
data$HLTHPLN1 <- as.factor(data$HLTHPLN1)



#PERSDOC2 - Do you have one person you think of as your personal doctor or health care
provider? (If "No" ask "Is there more than one or is there no person who you think of as your
personal doctor or health care provider?".)
 #Answers:Yes, only one(1), More than one(2), No(3)
 #Removed 7 and 9 bc they meant unsure or refused
data <- subset(data, data$PERSDOC2 != 7 & data$PERSDOC2 != 9)
data$PERSDOC2 = ifelse(data$PERSDOC2 == 3, 0, data$PERSDOC2)
data$PERSDOC2 <- as.factor(data$PERSDOC2)


#MEDCOST - Was there a time in the past 12 months when you needed to see a doctor but
could not because of cost?
 #Answers: Yes(1), No(2)
 #Removed 7, 9 and Blank bc they meant unsure, refused, or not asked
data <- subset(data, data$MEDCOST != 7 & data$MEDCOST != 9)
data$MEDCOST <- as.factor(data$MEDCOST)


#CHECKUP1 - : About how long has it been since you last visited a doctor for a routine
checkup? [A routine checkup is a general physical exam, not an exam for a specific injury,
illness, or condition.]
 #Answers: Less than a year(1), 1-2 years(2), 2-5 years(3), 5+ years(4), never(8)
 #Removed 7, 9 and Blank bc they meant unsure, refused, or not asked
data <- subset(data, data$CHECKUP1 != 7 & data$CHECKUP1 != 9)
data$CHECKUP1 <- as.factor(data$CHECKUP1)


#SEX - Indicate sex of respondent.
 #Answers: male(1), female(2)
data$SEX <- as.factor(data$SEX)


#MARITAL - Are you: (marital status)
 #Answers: Married(1), Divorced(2), Widowed(3), Separated(4), Never married(5),
 member of unmarried couple(6)
 #removed 9 bc meant refused
data <- subset(data, data$MARITAL != 9)
data$MARITAL <- as.factor(data$MARITAL)
# ONLY MAKES SENSE AS A FACTOR
```

```r
#EDUCA - What is the highest grade or year of school you completed?
 #Answers: Never attended/up to kindergarten(1), Grade 1-8(2), Grade 9-11(3),
 Grade 12 or GED(4), College 1-3 years(5), College 4+ years(6)
 #removed 9 bc meant refused
data <- subset(data, data$EDUCA != 9)
data$EDUCA <- as.factor(data$EDUCA)


#RENTHOM1 - Do you own or rent your home?
 #Answers: Own(1), Rent(2), Other(3)
 #Removed 7 and 9 bc they meant unsure, refused, or not asked
data <- subset(data, data$RENTHOM1 != 7 & data$RENTHOM1 != 9 & data$RENTHOM1 != 3)
data$RENTHOM1 <- as.factor(data$RENTHOM1)


#EMPLOY1 - : Are you currently...?
 #Answers: Employed for wages(1), Self-employed(2), Out of work >1 year(3),
 Out of work <1 year(4), Homemaker(5), Student(6), Retired(7), Unable to work(8)
 #removed 9 bc meant refused
data <- subset(data, data$EMPLOY1 != 9)
data$EMPLOY1 <- as.factor(data$EMPLOY1)
# ONLY MAKES SENSE AS A FACTOR

#CHILDREN - How many children less than 18 years of age live in your household?
 #Answers: Number of Children(1-87), None(88)
 #Removed 99 and Blank bc they meant refused or not asked
data <- subset(data, data$CHILDREN != 99)
# 88 means no kids, so replace that with a 0
data$CHILDREN = ifelse(data$CHILDREN == 88, 0, data$CHILDREN)


#INCOME2 - Is your annual household income from all sources
 #Answers: <$10,000(1), <$15,000(2), <$20,000(3), <$25,000(4), <$35,000(5),
 <$50,000(6), <$75,000(7), >$75,000(8)
 #Removed 77, 99 and Blank bc they meant unsure, refused, or not asked
 ##(78,000 rows)
data <- subset(data, data$INCOME2 != 77 & data$INCOME2 != 99)


#INTERNET - Have you used the internet in the past 30 days?
 #Answers: Yes(1), No(2)
 #Removed 7, 9 and Blank bc they meant unsure, refused, or not asked
data <- subset(data, data$INTERNET != 7 & data$INTERNET != 9)
data$INTERNET <- as.factor(data$INTERNET)


#SMOKE100 - Have you smoked at least 100 cigarettes in your entire life?
 #Answers: Yes(1), No(2)
 #Removed 7, 9 and Blank bc they meant unsure, refused, or not asked
data <- subset(data, data$SMOKE100 != 7 & data$SMOKE100 != 9)
data$SMOKE100 <- as.factor(data$SMOKE100)
```

```r
#USENOW3 has variables listed as Don't Know/Not Sure and Blank Values
# 1 (every day), 2 (some days), 3 (not at all)
data = subset(data, data$USENOW3 != 7 & data$USENOW3 != 9)
data$USENOW3 <- as.factor(data$USENOW3)


#ALCDAY5 has responses that are Refused and Blank
data = subset(data, data$ALCDAY5 != 999 & data$ALCDAY5 != 777)
data$ALCDAY5 = ifelse(data$ALCDAY5 == 888, 0, data$ALCDAY5)
data$ALCDAY5 = ifelse(data$ALCDAY5 >199.5, (data$ALCDAY5 - 200), data$ALCDAY5)
data$ALCDAY5 = ifelse(data$ALCDAY5 >99.5, 4*(data$ALCDAY5 - 100), data$ALCDAY5)


#FRUITJU1 - During the past month, how many times per day, week, or month did you drink
100 percent PURE fruit juices?
 #Answers: Times per day(101-199), Times per week(201-299), <1 time per month(300),
 Times per month(301-399), Never(555)
 #Removed 777, 999 and Blank bc they meant unsure, refused, or not asked
 ##38,000 rows removed
data <- subset(data, data$FRUITJU1 != 777 & data$FRUITJU1 != 999)
#data$FRUITJU1 <- as.factor(data$FRUITJU1)
data$FRUITJU1 = ifelse(data$FRUITJU1 == 555, 0, data$FRUITJU1)
data$FRUITJU1 = ifelse(data$FRUITJU1 >299.5, (data$FRUITJU1 - 300), data$FRUITJU1)
data$FRUITJU1 = ifelse(data$FRUITJU1 >199.5,
    4*(data$FRUITJU1 - 200), ifelse(data$FRUITJU1>99.5,
    30*(data$FRUITJU1 - 100),data$FRUITJU1))


#FRUIT1 - n: During the past month, not counting juice, how many times per day, week,
or month did you eat fruit?
 #Answers: Times per day(101-199), Times per week(201-299), <1 time per month(300),
 Times per month(301-399), Never(555)
 #Removed 777, 999 and Blank bc they meant unsure, refused, or not asked
 ##35,000 rows removed
data <- subset(data, data$FRUIT1 != 777 & data$FRUIT1 != 999)
#data$FRUIT1 <- as.factor(data$FRUIT1)
data$FRUIT1 = ifelse(data$FRUIT1 == 555, 0, data$FRUIT1)
data$FRUIT1 = ifelse(data$FRUIT1 >299.5, (data$FRUIT1 - 300), data$FRUIT1)
data$FRUIT1 = ifelse(data$FRUIT1 >199.5,
    4*(data$FRUIT1 - 200), ifelse(data$FRUIT1>99.5,
    30*(data$FRUIT1 - 100),data$FRUIT1))



#FLUSHOT6 - During the past 12 months, have you had either a flu shot or a flu vaccine
that was sprayed in your nose?
 #Answers: Yes(1), No(2)
 #Removed 7, 9 and Blank bc they meant unsure, refused, or not asked
 ##43,000 rows removed
data <- subset(data, data$FLUSHOT6 != 7 & data$FLUSHOT6 != 9)
data$FLUSHOT6 <- as.factor(data$FLUSHOT6)
```

```
#X_RFHLTH - Adults with good or better health
 #Answers: Good/Better health(1), Fair/Poor health(2)
 #remove 9 bc meant unsure or refused or missing
data <- subset(data, data$X_RFHLTH != 9)
data$X_RFHLTH <- as.factor(data$X_RFHLTH)



#VETERAN3 - : Have you ever served on active duty in the United States Armed Forces?
 #Answers: Yes(1), No(2)
 #Removed 7, 9 and Blank bc they meant unsure, refused, or not asked
data <- subset(data, data$VETERAN3 != 7 & data$VETERAN3 != 9)
data$VETERAN3 <- as.factor(data$VETERAN3)
```

4) Perform some EDA. Come up with your own questions and try to answer them with EDA.

```
summary(data)
```

```
##  HLTHPLN1   PERSDOC2   MEDCOST    CHECKUP1   SEX        MARITAL    EDUCA
##  1:282087   0: 40548   1: 28825   1:225011   1:130774   1:170562   1:   296
##  2: 19428   1:237877   2:272690   2: 35282   2:170741   2: 42129   2:  5958
##             2: 23090              3: 20165              3: 34828   3: 12685
##                                   4: 18689              4:  5969   4: 77576
##                                   8:  2368              5: 39189   5: 83526
##                                                         6:  8838   6:121474
##
##  RENTHOM1   VETERAN3      EMPLOY1          CHILDREN          INCOME2
##  1:231735   1: 41929   1      :132794   Min.   : 0.0000   Min.   :1.000
##  2: 69780   2:259586   7      : 88376   1st Qu.: 0.0000   1st Qu.:5.000
##                        2      : 26431   Median : 0.0000   Median :7.000
##                        8      : 20061   Mean   : 0.5298   Mean   :5.935
##                        5      : 17122   3rd Qu.: 1.0000   3rd Qu.:8.000
##                        3      :  5758   Max.   :41.0000   Max.   :8.000
##                        (Other): 10973
##  INTERNET   SMOKE100   USENOW3       ALCDAY5         FRUITJU1
##  1:247563   1:133108   1:  5380   Min.   : 0.00   Min.   :   0.00
##  2: 53952   2:168407   2:  4092   1st Qu.: 0.00   1st Qu.:   0.00
##                        3:292043   Median : 1.00   Median :   3.00
##                                   Mean   : 5.05   Mean   :  10.45
##                                   3rd Qu.: 6.00   3rd Qu.:  15.00
##                                   Max.   :30.00   Max.   :2970.00
##
##      FRUIT1        FLUSHOT6   X_RFHLTH
##  Min.   :   0.00   1:146886   1:248985
##  1st Qu.:  12.00   2:154629   2: 52530
##  Median :  30.00
##  Mean   :  30.26
##  3rd Qu.:  30.00
##  Max.   :2970.00
##
```

```
table(data$X_RFHLTH, data$HLTHPLN1)
```

```
##
##          1      2
```

```
##   1 233958  15027
##   2  48129   4401
```

This table shows the relationship between an individual's health condition and whether or not they have health coverage. Out of the people without any health coverage, about 23% (4401/(4401+15027)) have bad health. And out of all the people with health coverage, about 17% have bad health (48129/(233958+48129)). The chance of having bad health if you have no health coverage is much higher than if you have health insurance.

```
table(data$X_RFHLTH, data$INCOME2)
```

```
##
##         1     2     3     4     5     6     7     8
##   1  7037  8471 13547 18520 25180 37217 44686 94327
##   2  6113  6531  7209  7120  6910  6694  5570  6383
```

This table shows the relationship between an individual's annual household income and their health condition. For example, out of all the people with an income less than 10,000, approximately 46% have bad health (6113/(6113+7037) = .4649). On the other hand, out of all the people with an income of 75,000 or more, about 6% have bad health (6383/(6383+94327) = .0634).

```
table(data$X_RFHLTH, data$SMOKE100)
```

```
##
##          1      2
##   1 103066 145919
##   2  30042  22488
```

This table shows the relationship between whether or not an individual has smoked up to 100 cigarettes in their lifetime versus their health condition. Out of all the people who have smoked up to 100 cigarettes, about 23% have bad health (30042/(30042+103066)) = .2257). However, out of all the people who have not smoked up to 100 cigarettes, about 13% have bad health (22488/(22488+145919)=.1335).

5) What type of problem is it? Supervised or Unsupervised Learning? Classification or Regression? Binary or Multi-class? Uni-variate or Multivariate?

This problem is supervised learning, binary classification, and it is multivariate.

6) Check to see if there is a class imbalance for the target variable. If so, what does that mean when we are evaluating our model?

```
table(data$X_RFHLTH)
```

```
##
##      1      2
## 248985  52530
```

Yes, there's a class imbalance. About 83% of survey respondants are in good health and only 17% in poor health.

7) Recall that certain classifiers will only be able to handle numeric predictor variables. If some variables can reasonably be converted to numeric, do that so you can use them with any classifier that requires numeric. If the categories are binary, then coding them as 0–1 is okay. But as soon as you get more than two categories, things get problematic. If the values are "Low", "Intermediate", and "High" (or more generally, if they at least have a natural order), then you can again make sense of coding them numerically as 1, 2, 3. But if the values are "Red", "Green", "Blue" (or more generally, something that has no intrinsic order), then simply coding them as integers won't work.

```
numeric_data = data
for (n in 1:ncol(numeric_data)){
        numeric_data[, n] = as.numeric(numeric_data[, n])
```

```
}
numeric_data$X_RFHLTH = factor(numeric_data$X_RFHLTH)
```

8) Split the data

```
library(caTools)
set.seed(123)
split = sample.split(data$X_RFHLTH, SplitRatio = 0.75)
training_set = subset(data, split == TRUE)
test_set = subset(data, split == FALSE)
```

```
split = sample.split(numeric_data$X_RFHLTH, SplitRatio = 0.75)
numeric_training_set = subset(numeric_data, split == TRUE)
numeric_test_set = subset(numeric_data, split == FALSE)

numeric_training_set[,-20] = scale(numeric_training_set[,-20])
numeric_test_set[,-20] = scale(numeric_test_set[,-20])
```

9) Build a few classifiers using our classification algorithms. Be careful about which variables each algorithm can handle. You can train classifiers that can used mixed variables using all predictors. You'll have to train classifiers that require only numeric with a subset of the predictor variables.

#Knn

```
library(class)
knn_preds = knn(train = numeric_training_set[,-20],
             test = numeric_test_set[,-20],
             cl = numeric_training_set[,20],
             k = 5,
             prob = TRUE)
knn_probs = attr(knn_preds, 'prob')
```

#Logistic Regression

```
logreg_class = glm(formula = X_RFHLTH ~ .,
                   family = binomial,
                   data = numeric_training_set)

logreg_probs = predict(logreg_class,
                       type = 'response',
                       newdata = numeric_test_set[,-20])

logreg_accuracy = ifelse(test = logreg_probs > .50,
                         yes = 2,
                         no = 1)

logreg_accuracy = factor(logreg_accuracy)
```

#Decision Tree

```
library(rpart)
dt_class = rpart(formula = X_RFHLTH ~ .,
                 data = training_set)

dt_preds = predict(dt_class,
                   newdata = test_set[,-20],
                   type = 'prob')
```

```
dt_probs = dt_preds[,2]

dt_accuracy = ifelse(test = dt_probs > .50,
                         yes = 2,
                         no = 1)

dt_accuracy = factor(dt_accuracy)
```

#Random Forest

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
rf_class = randomForest(x = training_set[-20], y = training_set$X_RFHLTH, ntree = 100)
rf_preds = predict(rf_class, newdata = test_set[,-20], type = 'prob')

rf_probs = rf_preds[,2]

rf_accuracy = ifelse(test = rf_probs > .50, yes = 2, no = 1)

rf_accuracy = factor(rf_accuracy)
```

10) Find the accuracy of each classifier.

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

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

```
confusionMatrix(table(numeric_test_set$X_RFHLTH, knn_preds))
```

```
## Confusion Matrix and Statistics
##
##    knn_preds
##         1     2
##   1 58522  3724
##   2  9660  3472
##
##                Accuracy : 0.8224
##                  95% CI : (0.8197, 0.8252)
##     No Information Rate : 0.9045
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.249
##
##  Mcnemar's Test P-Value : <2e-16
##
```

```
##             Sensitivity : 0.8583
##             Specificity : 0.4825
##          Pos Pred Value : 0.9402
##          Neg Pred Value : 0.2644
##              Prevalence : 0.9045
##          Detection Rate : 0.7764
##    Detection Prevalence : 0.8258
##       Balanced Accuracy : 0.6704
##
##        'Positive' Class : 1
##
```

```
confusionMatrix(table(numeric_test_set$X_RFHLTH, logreg_accuracy))
```

```
## Confusion Matrix and Statistics
##
##    logreg_accuracy
##         1     2
##   1 60334  1912
##   2 10169  2963
##
##                Accuracy : 0.8397
##                  95% CI : (0.8371, 0.8423)
##     No Information Rate : 0.9353
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2592
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8558
##             Specificity : 0.6078
##          Pos Pred Value : 0.9693
##          Neg Pred Value : 0.2256
##              Prevalence : 0.9353
##          Detection Rate : 0.8004
##    Detection Prevalence : 0.8258
##       Balanced Accuracy : 0.7318
##
##        'Positive' Class : 1
##
```

```
confusionMatrix(test_set$X_RFHLTH, dt_accuracy)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 60628  1618
##          2  9727  3405
##
##                Accuracy : 0.8495
##                  95% CI : (0.8469, 0.852)
##     No Information Rate : 0.9334
##     P-Value [Acc > NIR] : 1
```

```
##
##                   Kappa : 0.3084
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8617
##             Specificity : 0.6779
##          Pos Pred Value : 0.9740
##          Neg Pred Value : 0.2593
##              Prevalence : 0.9334
##          Detection Rate : 0.8043
##    Detection Prevalence : 0.8258
##       Balanced Accuracy : 0.7698
##
##        'Positive' Class : 1
##
```

```
confusionMatrix(test_set$X_RFHLTH, rf_accuracy)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 60155  2091
##          2  9244  3888
##
##                Accuracy : 0.8496
##                  95% CI : (0.8471, 0.8522)
##     No Information Rate : 0.9207
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3343
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8668
##             Specificity : 0.6503
##          Pos Pred Value : 0.9664
##          Neg Pred Value : 0.2961
##              Prevalence : 0.9207
##          Detection Rate : 0.7980
##    Detection Prevalence : 0.8258
##       Balanced Accuracy : 0.7585
##
##        'Positive' Class : 1
##
```

11) Plot the ROC curves and find the AUC for each classifier.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

```
##
##     cov, smooth, var
```

```
logregROC = roc(numeric_test_set$X_RFHLTH ~ logreg_probs, plot=TRUE, print.auc=TRUE,
col="green", lwd =2, legacy.axes=TRUE, main="ROC Curves")
```

```
## Setting levels: control = 1, case = 2
```

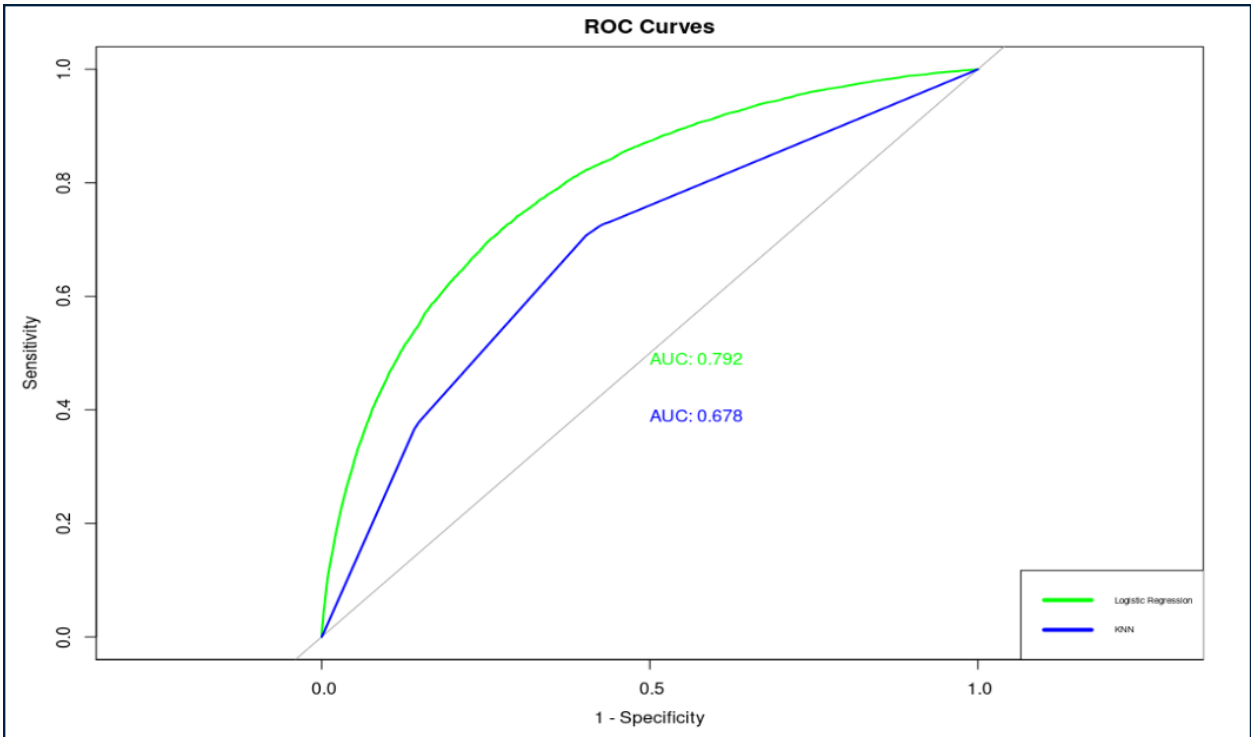
```
## Setting direction: controls < cases
```

```
knnROC = roc(numeric_test_set$X_RFHLTH ~ knn_probs, plot=TRUE, print.auc=TRUE, col="blue",
lwd = 2, print.auc.y=0.4, legacy.axes=TRUE, add = TRUE)
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls > cases
```

```
legend("bottomright",legend=c("Logistic Regression", "KNN"), col=c("green", "blue"),
lwd=4, cex = 0.5)
```




```
dtROC = roc(test_set$X_RFHLTH ~ dt_probs, plot=TRUE, print.auc=TRUE, col="green", lwd =2,
legacy.axes=TRUE, main="ROC Curves")
```

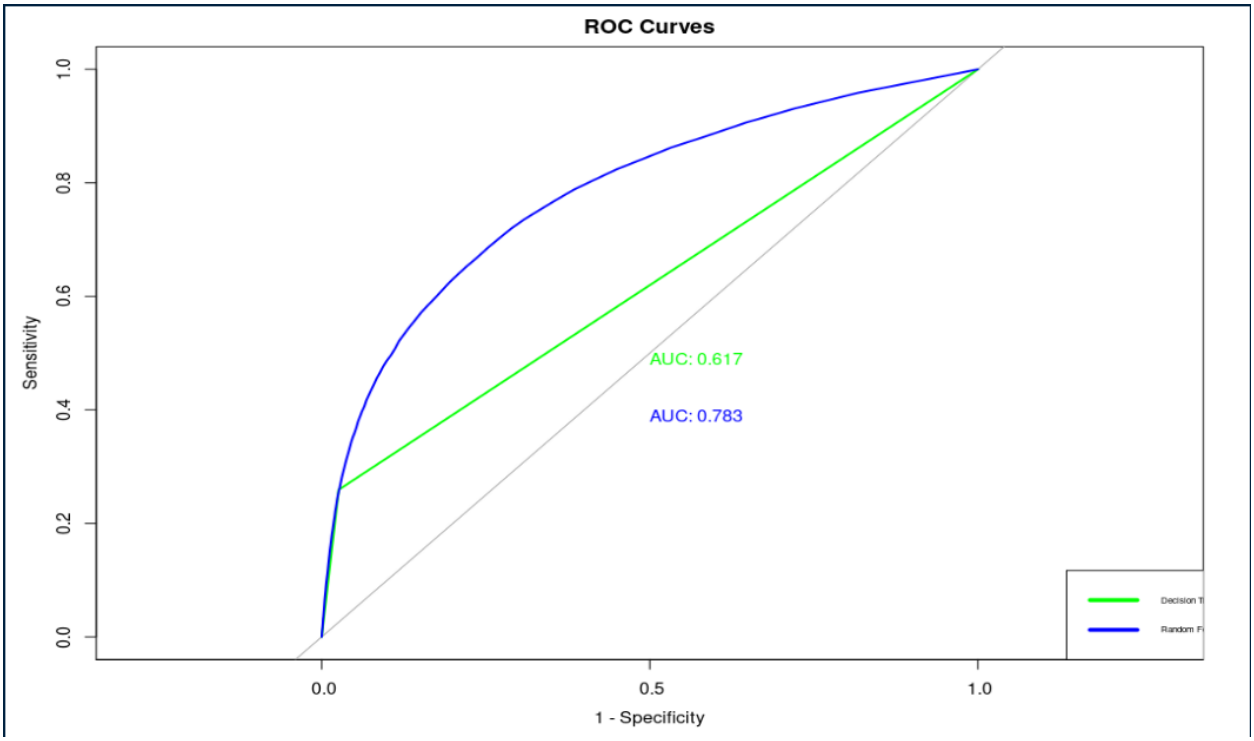
```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
rfROC = roc(test_set$X_RFHLTH ~ rf_probs, plot=TRUE, print.auc=TRUE, col="blue", lwd = 2,
print.auc.y=0.4, legacy.axes=TRUE, add = TRUE)
```

```
## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
```

```
legend("bottomright",legend=c("Decision Tree", "Random Forest"), col=c("green", "blue"), lwd=4,
cex = 0.5)
```




```
auc(knnROC)
```

```
## Area under the curve: 0.6783
```

```
auc(logregROC)
```

```
## Area under the curve: 0.7914
```

```
auc(dtROC)
```

```
## Area under the curve: 0.6166
```

```
auc(rfROC)
```

```
## Area under the curve: 0.7845
```

12) Summarize your findings.

Based off of the plots that were created and each area under the curve, the classifier that worked the best for predicting whether someone has good or bad health was logistic regression classification, with the area under the curve curve being 0.7923. The second best classifier was random forest, with an area under the curve of .7831. The third best was knn, with an area under the curve of 0.6783. The worst classifier was decision tree with an area under the curve of 0.6166.