IMT 573 Problem Set 7 - Prediction

Neelam Purswani

Due: Tuesday, November 27, 2018

##### Collaborators: Sindhuja Dutta, Dhaval Chheda

##### Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the problemset7.Rmd file from Canvas. Open problemset7.Rmd in RStudio and supply your solutions to the assignment by editing problemset7.Rmd.
2. Replace the “Insert Your Name Here” text in the author: field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. If you are using more than just a standard function that you found from another source, please credit the source in the comments. For example:
4. Collaboration on problem sets is acceptable, and even encouraged, but students must turn in an individual write-up in their own words and their own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students’ responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click Knit PDF or Knit Word, rename the R Markdown file to YourLastName\_YourFirstName\_ps7.Rmd, knit a PDF or DOC and submit both the PDF/DOC and the Rmd file on Canvas.

##### Setup:

In this problem set you will need, at minimum, the following R packages.

# Load standard libraries  
library(tidyverse)  
library(caTools)  
library(pROC)  
library(randomForest)  
#library(dplyr)

In this problem set we will use the dataset from Yeh, I-Cheng, Yang, King-Jang, and Ting, Tao-Ming, “Knowledge discovery on RFM model using Bernoulli sequence,”Expert Systems with Applications, 2008 (<doi:10.1016/j.eswa.2008.07.018>). This dataset is currently being used for a competition on <http://www.DrivenData.org>. Information on the dataset and variables can be found at: <https://archive.ics.uci.edu/ml/machine-learning-databases/blood-transfusion/transfusion.names>.

# Load data called 'TransfusionData.csv'  
blood\_donors <- read.csv("TransfusionData.csv")  
#transfusionDataDF <- as.data.frame(transfusionData)  
str(blood\_donors)

## 'data.frame': 748 obs. of 5 variables:  
## $ Recency..months. : num 2 0 1 2 1 4 2 1 2 5 ...  
## $ Frequency..times. : int 50 13 16 20 24 4 7 12 9 46 ...  
## $ Monetary..c.c..blood. : int 12500 3250 4000 5000 6000 1000 1750 3000 2250 11500 ...  
## $ Time..months. : num 98 28 35 45 77 4 14 35 22 98 ...  
## $ whether.he.she.donated.blood.in.March.2007: int 1 1 1 1 0 0 1 0 1 1 ...

nrow(blood\_donors)

## [1] 748

ncol(blood\_donors)

## [1] 5

##### Question 1a

Describe each variable in the dataset. (Hint: use the reference listed in the above instructions). R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

##### Question 1b

Prepare the data for easier processing. Describe what you did and why.

#renaming the columns so that we understand it better while using it  
names(blood\_donors) <-  
 c("monthssincelast","frequency","volume","monthssincefirst","donated")  
#changing the data type of donated from int to factor as 0 indicates false and 1 indicates true  
blood\_donors$donated <- as.factor(blood\_donors$donated)

##### Question 1c

Provide some basic summary statistics to become more familiar with the dataset.

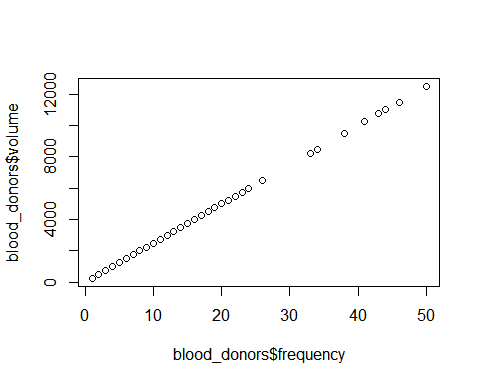
summary(blood\_donors)

## monthssincelast frequency volume monthssincefirst  
## Min. : 0.000 Min. : 1.000 Min. : 250 Min. : 2.00   
## 1st Qu.: 2.750 1st Qu.: 2.000 1st Qu.: 500 1st Qu.:16.00   
## Median : 7.000 Median : 4.000 Median : 1000 Median :28.00   
## Mean : 9.507 Mean : 5.515 Mean : 1379 Mean :34.28   
## 3rd Qu.:14.000 3rd Qu.: 7.000 3rd Qu.: 1750 3rd Qu.:50.00   
## Max. :74.000 Max. :50.000 Max. :12500 Max. :98.00   
## donated  
## 0:570   
## 1:178   
##   
##   
##   
##

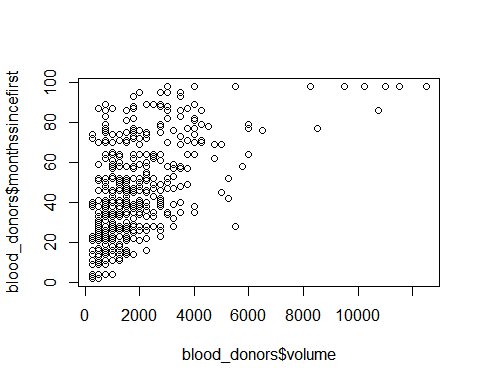
#lets find out the correlation between frequency and volume of blood  
cor(blood\_donors$frequency,blood\_donors$volume)

## [1] 1

plot(blood\_donors$frequency, blood\_donors$volume)



plot(blood\_donors$volume, blood\_donors$monthssincefirst)

 Here are the basic summary statistics for the dataset and correlation between frequency and volume is high. ####Question 2 As part of this assignment we will evaluate the performance of a few different statistical learning methods. We will fit a particular statistical learning method on a set of observations and measure its performance on a set of observations.

##### Question 2a

Discuss the advantages of using a training/test split when evaluating statistical models. To avoid overfitting: if we use all the data t train the model then there could arise a case of over-fitting of the model so we split it into Training and Test. Also, I think training test split helps us to evaluate if the model is predicting the right values for the test part based on its performance with the traning data. If the values from the test model match the actual values we can assert that the model is correct.

##### Question 2b

Split your data into a and set based on an 80-20 split, in other words, 80% of the observations will be in the training set. Use the code below (substituting in your own variable names) so that everyone has the same split.

# code adapted from https://rpubs.com/ID\_Tech/S1 AND https://stackoverflow.com/a/31634462  
  
# Set seed for reproducibility  
set.seed(112718)  
# splits the data in the ratio mentioned in SplitRatio. After splitting marks these rows as logical   
# TRUE and the the remaining are marked as logical FALSE  
sample = sample.split(blood\_donors$donated, SplitRatio = .8)  
# creates a training dataset named train with rows which are marked as TRUE  
donor\_train = subset(blood\_donors, sample == TRUE)  
# creates a training dataset named test with rows which are marked as FALSE  
nrow(donor\_train)

## [1] 598

names(donor\_train)

## [1] "monthssincelast" "frequency" "volume"   
## [4] "monthssincefirst" "donated"

donor\_test = subset(blood\_donors, sample == FALSE)  
nrow(donor\_test)

## [1] 150

#### Question 3

In this problem set our goal is to predict whether someone will donate blood in March 2007. First consider training a simple logistic regression model for whether an individual donated blood in March 2007 based on frequency of donations.

##### Question 3a

Fit the model described above using the function in R.

#logistic\_results <- glm(mansion ~ bedrooms+bathrooms+sqft\_living,family=binomial, data=train)  
#summary(logistic\_results)  
#length(donor\_train)  
logistic\_model\_transfusion <- glm(donated ~ frequency, data=donor\_train, family=binomial)  
summary(logistic\_model\_transfusion)

##   
## Call:  
## glm(formula = donated ~ frequency, family = binomial, data = donor\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9112 -0.7146 -0.6472 -0.6259 1.8583   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.60475 0.14004 -11.460 < 2e-16 \*\*\*  
## frequency 0.07399 0.01628 4.544 5.51e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 655.57 on 597 degrees of freedom  
## Residual deviance: 632.53 on 596 degrees of freedom  
## AIC: 636.53  
##   
## Number of Fisher Scoring iterations: 4

##### Question 3b

Describe in your own words your interpretation of the model summary. (Run a summary function of your model if you haven’t already.)

summary(logistic\_model\_transfusion)

##   
## Call:  
## glm(formula = donated ~ frequency, family = binomial, data = donor\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9112 -0.7146 -0.6472 -0.6259 1.8583   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.60475 0.14004 -11.460 < 2e-16 \*\*\*  
## frequency 0.07399 0.01628 4.544 5.51e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 655.57 on 597 degrees of freedom  
## Residual deviance: 632.53 on 596 degrees of freedom  
## AIC: 636.53  
##   
## Number of Fisher Scoring iterations: 4

#predict(logistic\_model\_transfusion, donor\_test, type="response")  
#head(donor\_test)

Since the p value is very less, we can say there is a statistically significant relationship between donated and frequency of donations for people. That menas we can use frequency to determine if the person has donated blood in March 2007

#### Question 4

Next, let’s consider the performance of this model.

##### Question 4a

Predict donations in March 2007 for each observation in your test set using the model fit in Question 3a. Save these predictions as y\_hat.

y\_hat <- predict(logistic\_model\_transfusion, donor\_test, type="response")  
length(y\_hat)

## [1] 150

y\_hat

## 1 7 17 26 28 32 39   
## 0.8904014 0.2522203 0.3614995 0.3614995 0.3280879 0.2963313 0.3614995   
## 53 54 57 58 61 79 81   
## 0.2385224 0.3280879 0.2664296 0.2522203 0.2522203 0.1889626 0.2385224   
## 83 95 101 102 106 110 112   
## 0.2126909 0.3119888 0.2664296 0.3446017 0.4141377 0.2963313 0.2811384   
## 122 125 127 129 134 136 140   
## 0.2253443 0.2126909 0.2664296 0.2664296 0.2385224 0.2005640 0.1889626   
## 146 148 154 160 161 170 171   
## 0.2005640 0.1889626 0.1778830 0.1778830 0.1778830 0.1778830 0.1778830   
## 172 175 179 192 194 204 209   
## 0.1778830 0.1778830 0.2385224 0.2522203 0.3119888 0.2522203 0.2385224   
## 220 222 223 224 227 230 243   
## 0.1778830 0.1778830 0.1778830 0.1778830 0.1778830 0.1778830 0.2005640   
## 244 248 254 255 260 261 272   
## 0.4141377 0.3280879 0.2005640 0.3446017 0.3280879 0.2126909 0.2522203   
## 273 275 276 278 289 292 297   
## 0.1889626 0.2253443 0.2126909 0.2005640 0.2253443 0.3280879 0.2253443   
## 307 310 312 321 326 327 332   
## 0.2253443 0.2005640 0.1889626 0.3963066 0.1889626 0.1889626 0.2385224   
## 340 345 350 352 353 363 366   
## 0.2005640 0.1778830 0.1778830 0.1889626 0.2253443 0.2005640 0.1889626   
## 368 372 377 384 385 386 388   
## 0.2126909 0.2005640 0.2253443 0.1778830 0.1778830 0.2522203 0.1778830   
## 391 401 402 412 417 422 426   
## 0.2126909 0.1889626 0.2253443 0.1778830 0.1778830 0.1889626 0.2253443   
## 430 434 437 450 457 458 477   
## 0.2005640 0.2126909 0.1889626 0.2005640 0.1778830 0.1778830 0.2385224   
## 483 484 487 492 505 511 522   
## 0.1778830 0.1778830 0.2385224 0.2005640 0.5790906 0.2126909 0.3119888   
## 526 528 530 533 539 540 545   
## 0.2126909 0.3787470 0.2385224 0.2664296 0.2664296 0.1889626 0.3614995   
## 548 552 555 558 559 562 565   
## 0.2005640 0.2385224 0.2126909 0.2385224 0.2253443 0.2811384 0.3614995   
## 566 567 576 582 589 592 602   
## 0.2811384 0.2385224 0.2385224 0.1778830 0.1778830 0.3119888 0.2385224   
## 604 624 625 630 643 644 645   
## 0.2664296 0.2385224 0.2005640 0.3119888 0.2385224 0.2253443 0.1889626   
## 647 659 663 664 669 671 681   
## 0.1889626 0.2005640 0.2005640 0.2664296 0.1778830 0.2005640 0.3614995   
## 690 701 703 708 710 711 715   
## 0.1778830 0.2005640 0.1778830 0.2126909 0.2385224 0.1889626 0.2126909   
## 727 735 742   
## 0.2385224 0.1778830 0.2522203

#head(y\_hat)  
#summary(y\_hat)

##### Question 4b

Use a threshold of 0.4 to classify predictions. Using a confusion matrix, what is the number of false positives on the test data? Interpret this in your own words.

table(donor\_test$donated, y\_hat > 0.4)

##   
## FALSE TRUE  
## 0 113 1  
## 1 33 3

The confusion matrix tells us that 113 values were correctly predicted by the model to be 0 and 33 were incorrectly predicted. For 1, the model correctly predicted 3 values and incorrectly predicted 1 value. #####Question 4c Calculate the accuracy rate of your y\_hat predictions.

#finding the percentage of 1s we correctly predicted  
3/(33+3)

## [1] 0.08333333

#finding the percentage of 0s we correctly predicted  
113/(113+1)

## [1] 0.9912281

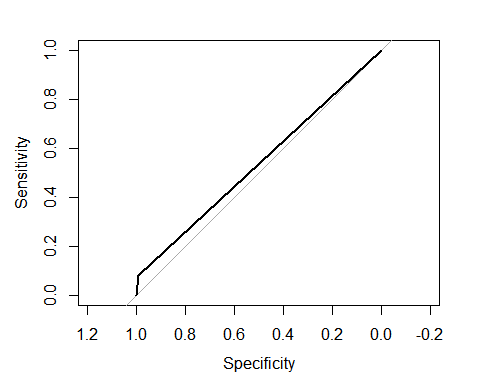
#accuracy  
(113+3)/(113+1+3+33)

## [1] 0.7733333

##### Question 4d

Using the function (or similar), plot the ROC curve for this model. Discuss what you find.

donor\_test$PredictedVal <- ifelse(y\_hat > 0.4, 1, 0)  
roc\_logistic <- roc(donor\_test$donated ~ donor\_test$PredictedVal)  
plot(roc\_logistic)



auc(roc\_logistic)

## Area under the curve: 0.5373

Area under the curve is : 0.5373 The ROC curve gives us complete sensitivity/specificity

#### Question 5

Suppose we use the data to construct a new predictor variable based on a donor’s average number of months between donations.

##### Question 5a

Why might this be an interesting variable to help predict whether an individual will donate in March 2007? The variable month\_span\_between\_donations will tell us the time lapse in visits for a donor in donating the blood which can guide us towatds whethere they donated the blood or not.

##### Question 5b

Write a function to add this predictor to your *full dataset*. Call this variable month\_span\_between\_donations.

#blood\_donations$month\_span\_between\_donations <-   
#(blood\_donations$monthssincefirst) #- blood\_donations$monthsincelast)/blood\_donations$frequency  
  
  
blood\_donors<-blood\_donors%>%  
 mutate(month\_span\_between\_donations=(monthssincefirst-monthssincelast)/frequency)  
head(blood\_donors)

## monthssincelast frequency volume monthssincefirst donated  
## 1 2 50 12500 98 1  
## 2 0 13 3250 28 1  
## 3 1 16 4000 35 1  
## 4 2 20 5000 45 1  
## 5 1 24 6000 77 0  
## 6 4 4 1000 4 0  
## month\_span\_between\_donations  
## 1 1.920000  
## 2 2.153846  
## 3 2.125000  
## 4 2.150000  
## 5 3.166667  
## 6 0.000000

Rerun the train test split

set.seed(112718)  
  
sample = sample.split(blood\_donors$donated, SplitRatio = .8)  
# creates a training dataset named train with rows which are marked as TRUE  
donor\_train = subset(blood\_donors, sample == TRUE)  
# creates a training dataset named test with rows which are marked as FALSE  
donor\_test = subset(blood\_donors, sample == FALSE)

##### Question 5c

Fit a second logistic regression model including *only* this new feature. Use the function to look at the model. How does this model compare to the model in question 3a?

logistic\_training2 <-glm(donated ~ month\_span\_between\_donations,  
 data = donor\_train, family=binomial)  
summary(logistic\_training2)

##   
## Call:  
## glm(formula = donated ~ month\_span\_between\_donations, family = binomial,   
## data = donor\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7881 -0.7699 -0.7253 -0.5411 2.0366   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.01002 0.12944 -7.803 6.05e-15 \*\*\*  
## month\_span\_between\_donations -0.03820 0.02235 -1.709 0.0874 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 655.57 on 597 degrees of freedom  
## Residual deviance: 652.37 on 596 degrees of freedom  
## AIC: 656.37  
##   
## Number of Fisher Scoring iterations: 4

We can say that there is a statistically significant relationship because p-value is less than 0.05.

##### Question 5d

Repeat questions 4a and 4b for this new model. Save these new predictions as y\_hat2. Interpret this new confusion matrix in your own words.

#Predicting results using above model  
y\_hat2=predict(logistic\_training2, type = "response",newdata =donor\_test)  
 #ifelse(donor\_test$y\_hat2 > 0.4, 1, 0)  
str(y\_hat2)

## Named num [1:150] 0.253 0.254 0.243 0.248 0.249 ...  
## - attr(\*, "names")= chr [1:150] "1" "7" "17" "26" ...

y\_hat2

## 1 7 17 26 28 32 39   
## 0.2528704 0.2543578 0.2431422 0.2481982 0.2487077 0.2494221 0.2386519   
## 53 54 57 58 61 79 81   
## 0.2428076 0.2370033 0.2469276 0.2401423 0.2391480 0.2595671 0.2439800   
## 83 95 101 102 106 110 112   
## 0.2416391 0.2595671 0.2398934 0.2266270 0.2397910 0.2232461 0.2335670   
## 122 125 127 129 134 136 140   
## 0.2353954 0.2245737 0.2296019 0.2279171 0.2301654 0.2267987 0.2347085   
## 146 148 154 160 161 170 171   
## 0.2381565 0.2347085 0.2669756 0.2669756 0.2669756 0.2669756 0.2669756   
## 172 175 179 192 194 204 209   
## 0.2669756 0.2669756 0.2234670 0.2189236 0.2464437 0.2133766 0.2475200   
## 220 222 223 224 227 230 243   
## 0.2669756 0.2669756 0.2669756 0.2669756 0.2669756 0.2669756 0.2179919   
## 244 248 254 255 260 261 272   
## 0.2381565 0.2207173 0.2547029 0.2408323 0.2393135 0.2036960 0.2543578   
## 273 275 276 278 289 292 297   
## 0.2347085 0.1942608 0.1960607 0.1990886 0.2465727 0.2463362 0.2465727   
## 307 310 312 321 326 327 332   
## 0.1758397 0.2595671 0.2487077 0.2368594 0.2595671 0.1474327 0.2439800   
## 340 345 350 352 353 363 366   
## 0.2094345 0.2669756 0.2669756 0.2312954 0.2128275 0.1343412 0.2416391   
## 368 372 377 384 385 386 388   
## 0.2245737 0.2267987 0.2166924 0.2669756 0.2669756 0.2097358 0.2669756   
## 391 401 402 412 417 422 426   
## 0.1886438 0.2487077 0.2522935 0.2669756 0.2669756 0.2279171 0.2064984   
## 430 434 437 450 457 458 477   
## 0.2115505 0.1842980 0.2522935 0.1910919 0.2669756 0.2669756 0.2126143   
## 483 484 487 492 505 511 522   
## 0.2669756 0.2669756 0.1940615 0.1632940 0.2457003 0.2669756 0.2413210   
## 526 528 530 533 539 540 545   
## 0.2504963 0.2372336 0.2428076 0.2451562 0.2347085 0.2595671 0.2198581   
## 548 552 555 558 559 562 565   
## 0.2451562 0.2279171 0.2296019 0.2358540 0.2669756 0.2290393 0.2255254   
## 566 567 576 582 589 592 602   
## 0.2108434 0.2301654 0.2267987 0.2669756 0.2669756 0.2490323 0.2179919   
## 604 624 625 630 643 644 645   
## 0.2559133 0.2463362 0.2644912 0.2150466 0.2312954 0.2272456 0.2669756   
## 647 659 663 664 669 671 681   
## 0.2669756 0.2312954 0.2547029 0.2139496 0.2669756 0.1229303 0.2416391   
## 690 701 703 708 710 711 715   
## 0.2669756 0.2223642 0.2669756 0.1663613 0.2347085 0.2669756 0.1960607   
## 727 735 742   
## 0.2370033 0.2669756 0.2034748

donor\_test$predict2 <- ifelse(y\_hat2 > 0.4, 1, 0)  
donor\_test$predict2

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [141] 0 0 0 0 0 0 0 0 0 0

nrow(donor\_test$predict2)

## NULL

summary(y\_hat2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1229 0.2258 0.2411 0.2373 0.2596 0.2670

#Creating confusion matrix for threshold of 0.4  
table(donor\_test$donated, y\_hat2>0.4)

##   
## FALSE  
## 0 114  
## 1 36

##### Question 5e

Use the function to fit a multiple logistic regression model with monetary and recency as predictors and make predictions for the test set. Save these predictions as y\_hat3.

head(donor\_train)

## monthssincelast frequency volume monthssincefirst donated  
## 2 0 13 3250 28 1  
## 3 1 16 4000 35 1  
## 4 2 20 5000 45 1  
## 5 1 24 6000 77 0  
## 6 4 4 1000 4 0  
## 8 1 12 3000 35 0  
## month\_span\_between\_donations  
## 2 2.153846  
## 3 2.125000  
## 4 2.150000  
## 5 3.166667  
## 6 0.000000  
## 8 2.833333

logistic\_training3 <-glm(donated ~ monthssincelast + volume ,  
 data = donor\_train, family=binomial)  
summary(logistic\_training3)

##   
## Call:  
## glm(formula = donated ~ monthssincelast + volume, family = binomial,   
## data = donor\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9593 -0.8020 -0.5057 -0.2828 2.5844   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -6.315e-01 1.909e-01 -3.308 0.000939 \*\*\*  
## monthssincelast -1.142e-01 1.825e-02 -6.257 3.92e-10 \*\*\*  
## volume 2.382e-04 6.746e-05 3.532 0.000413 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 655.57 on 597 degrees of freedom  
## Residual deviance: 581.60 on 595 degrees of freedom  
## AIC: 587.6  
##   
## Number of Fisher Scoring iterations: 5

#predicting results based on above model on test data  
y\_hat3=predict(logistic\_training3, type = "response",  
 newdata =donor\_test)  
  
summary(y\_hat3)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0392 0.1161 0.2633 0.2460 0.3360 0.8927

donor\_test$predict3 <- ifelse(y\_hat3 > 0.4, 1, 0)  
donor\_test$predict3

## [1] 1 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1  
## [106] 0 1 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [141] 0 0 0 0 0 0 0 0 0 0

#creating a confusion matrix for threshold of 0.4  
table(donor\_test$donated, y\_hat3>0.4)

##   
## FALSE TRUE  
## 0 106 8  
## 1 28 8

##### Question 5f

Calculate the accuracy rate of your y\_hat3 predictions.

#finding the percentage of 1s correctly predicted  
(28)/(28+8)

## [1] 0.7777778

#finding the percentage of 0s correctly predicted  
106/(106+8)

## [1] 0.9298246

#finding the accuracy  
(106+8)/(106+8+28+8)

## [1] 0.76

##### Question 5f

Create a correlation matrix for the donorData (without the donated March 2007 variable). What do you notice?

cor(subset(blood\_donors, select = c(monthssincelast,frequency, volume, monthssincefirst)))

## monthssincelast frequency volume monthssincefirst  
## monthssincelast 1.0000000 -0.1827455 -0.1827455 0.1606181  
## frequency -0.1827455 1.0000000 1.0000000 0.6349403  
## volume -0.1827455 1.0000000 1.0000000 0.6349403  
## monthssincefirst 0.1606181 0.6349403 0.6349403 1.0000000

frequency and volume have the highest correlation and frequency and months since first donation also have a strong correlation.

##### Question 5g

We have a very limited set of variables in this datset from which to build a model. If you could add in new data or create other calculated variables to aid in model building what would you add and why? (Must include at least 2 additional variables with explanation for full credit.) The objective of this model is to be able to predict if the person will donate blood: elibility for recent visits: storing this data will tell us if the person is eligible for donations. We can combine this with frequency to see the probability of them donating the blood. lastdonatedon: this will help us to understand if they will donate again based on when they last donated blood.

#### Question 6

Another very popular classifier used in data science is called a .

#### Question 6a

Use the function to fit a random forest model with monetary and recency as predictors. Make predictions for the test set using the random forest model. Save these predictions as y\_hat4.

set.seed(123)  
  
donor\_train$donated <- as.factor(donor\_train$donated)  
donor\_test$donated <- as.factor(donor\_test$donated)  
  
model\_random\_Forest <-randomForest(donated ~ monthssincelast + volume, data = donor\_train, family=binomial)  
summary(model\_random\_Forest)

## Length Class Mode   
## call 4 -none- call   
## type 1 -none- character  
## predicted 598 factor numeric   
## err.rate 1500 -none- numeric   
## confusion 6 -none- numeric   
## votes 1196 matrix numeric   
## oob.times 598 -none- numeric   
## classes 2 -none- character  
## importance 2 -none- numeric   
## importanceSD 0 -none- NULL   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 14 -none- list   
## y 598 factor numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## terms 3 terms call

y\_hat4=predict(model\_random\_Forest, type = "response",newdata =donor\_test)  
y\_hat4

## 1 7 17 26 28 32 39 53 54 57 58 61 79 81 83 95 101 102   
## 1 1 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1   
## 106 110 112 122 125 127 129 134 136 140 146 148 154 160 161 170 171 172   
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## 175 179 192 194 204 209 220 222 223 224 227 230 243 244 248 254 255 260   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## 261 272 273 275 276 278 289 292 297 307 310 312 321 326 327 332 340 345   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## 350 352 353 363 366 368 372 377 384 385 386 388 391 401 402 412 417 422   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## 426 430 434 437 450 457 458 477 483 484 487 492 505 511 522 526 528 530   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1   
## 533 539 540 545 548 552 555 558 559 562 565 566 567 576 582 589 592 602   
## 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0   
## 604 624 625 630 643 644 645 647 659 663 664 669 671 681 690 701 703 708   
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   
## 710 711 715 727 735 742   
## 0 0 0 0 0 0   
## Levels: 0 1

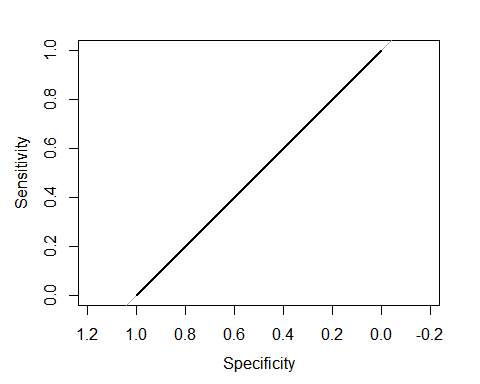
#donor\_test$predict4 <- ifelse(y\_hat4 > 0.4, 1, 0)  
#donor\_test$predict4  
  
summary(y\_hat4)

## 0 1   
## 136 14

#### Question 6b

Compare the accuracy of each of the models from this problem set using ROC curves. What have you learned about logistic regression and random forest from this dataset?

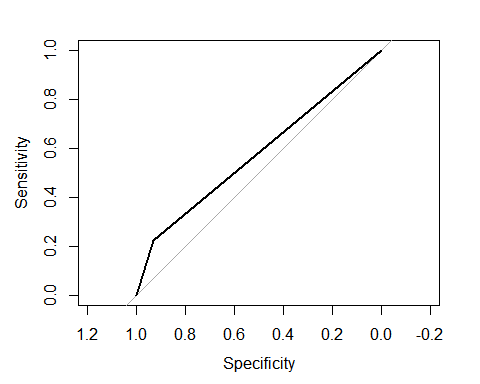
logistic\_roc2 <- roc(donor\_test$donated ~ donor\_test$predict2)  
plot(logistic\_roc2)



auc(logistic\_roc2)

## Area under the curve: 0.5

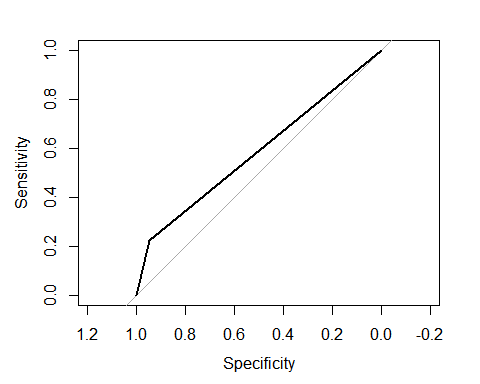
logistic\_roc3 <- roc(donor\_test$donated ~ donor\_test$predict3)  
plot(logistic\_roc3)



auc(logistic\_roc3)

## Area under the curve: 0.576

random\_roc <- roc(donor\_test$donated ~ as.numeric(y\_hat4))  
plot(random\_roc)



auc(random\_roc)

## Area under the curve: 0.5848

We see the ROC curves for monetary and recency are same for Logistic regression for that and randomForest method. Also, their area under the curve is also approximately the same. I think that means for both the models sensitivity specificity was the same.