IMT 573 Problem Set 7 - Prediction

Naga Soundari Balamurugan

Due: Tuesday, November 27, 2018

Collaborators: Dhaval Chedda, Jayashree Raman

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 chunks, 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(caret)
```

Data: In this problem set we will use the `TransfusionData` 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'
transfusionData <- read.csv("TransfusionData.csv")
```

Question 1a

Describe each variable in the dataset. (Hint: use the reference listed in the above instructions). >The dataset has 748 rows and 5 columns. The columns are as follows: >Recency..months: months since last

donation >Frequency..times : total number of donation >Monetary..c.c..blood: total blood donated in c.c >Time..months: months since first donation and >whether.he.she.donated.blood.in.March.2007: 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.

```
colnames(transfusionData) <- c("Recency", "Frequency", "Monetary", "Time", "Donated")
```

I have changed the column names of the dataset for easy access and more readability.

Question 1c

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

```
summary(transfusionData)
```

```
##     Recency         Frequency         Monetary          Time
##  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
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.238
##  3rd Qu.:0.000
##  Max.   :1.000
```

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 *training* observations and measure its performance on a set of *test* observations.

Question 2a

Discuss the advantages of using a training/test split when evaluating statistical models.

> Using the whole dataset for both training and test would overfit the data and might not work well(predict) for the any new data. Splitting the dataset into training/test data gives us the advantage of training over one set and testing over the same dataset but not the exact same datapoints. Thus it helps us in generalizing and building up a better model.
>
> As honest assessments of the performance of our predictive models can be done using the split, it could help us to compare the performances of different predictive modeling procedures.

Question 2b

Split your data into a *training* and *test* 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(transfusionData$Donated, SplitRatio = .8)
# creates a training dataset named train with rows which are marked as TRUE
donor_train = subset(transfusionData, sample == TRUE)
# creates a training dataset named test with rows which are marked as FALSE
donor_test  = subset(transfusionData, sample == FALSE)
```

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 `glm` function in R.

```
#Logistic model
model_glm <- glm(Donated ~ Frequency, family = "binomial", data = donor_train)
summary(model_glm)
```

```
##
## 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.)

> From the summary statistics, we can see that we have a very significant z-scores of Frequency(less than 0.05). The AIC score is 636.53 but it cannot be used to predict the goodness of fit as the number itself is not meaningful. If one have more than one similar candidate models, then he/she should select the model that has the smallest AIC. The null deviance shows how well the response variable is predicted by the model that includes only the intercept (grand mean). We have a value of 655.566 on 597 degrees of freedom. Including the independent variable (Frequency) decreased the deviance to 632.529 points on 596 degrees of freedom, a significant reduction in deviance. The Residual Deviance has reduced by 23 with a loss of one degrees of freedom.

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`.

```
#Predictions
donor_test$y_hat <- predict(model_glm, donor_test, type = "response")
```

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.

```
donor_test$Prediction <- ifelse(donor_test$y_hat > 0.4, 1, 0)
confMatrixResults <- confusionMatrix(data = factor(donor_test$Prediction),
                                     reference = factor(donor_test$Donated))
confMatrixResults
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 113  33
##          1   1   3
##
##                Accuracy : 0.7733
##                  95% CI : (0.6979, 0.8376)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.3931
##
##                   Kappa : 0.1071
##  Mcnemar's Test P-Value : 1.058e-07
##
##             Sensitivity : 0.99123
##             Specificity : 0.08333
##          Pos Pred Value : 0.77397
##          Neg Pred Value : 0.75000
##              Prevalence : 0.76000
```

```
##          Detection Rate : 0.75333
##    Detection Prevalence : 0.97333
##       Balanced Accuracy : 0.53728
##
##        'Positive' Class : 0
##
```

There is **1 false positive** in the prediction using the logistic model. From the confusion matrix, we can see that out of total 150(113 + 33 + 1 + 3) predictions, 116(113 + 3) predictions are done right and 34(33 + 1) are wrong. Among the incorrect 34 predictions, 1 is false positive and 33 are false negatives.

Question 4c

Calculate the accuracy rate of your y_hat predictions.

```
tot_correct_predictions <- 113 + 3  #total correct predictions
tot_predictions <- 113 + 33 + 1 + 3 #total predictions made
accuracy <- (tot_correct_predictions/tot_predictions) * 100
accuracy
```

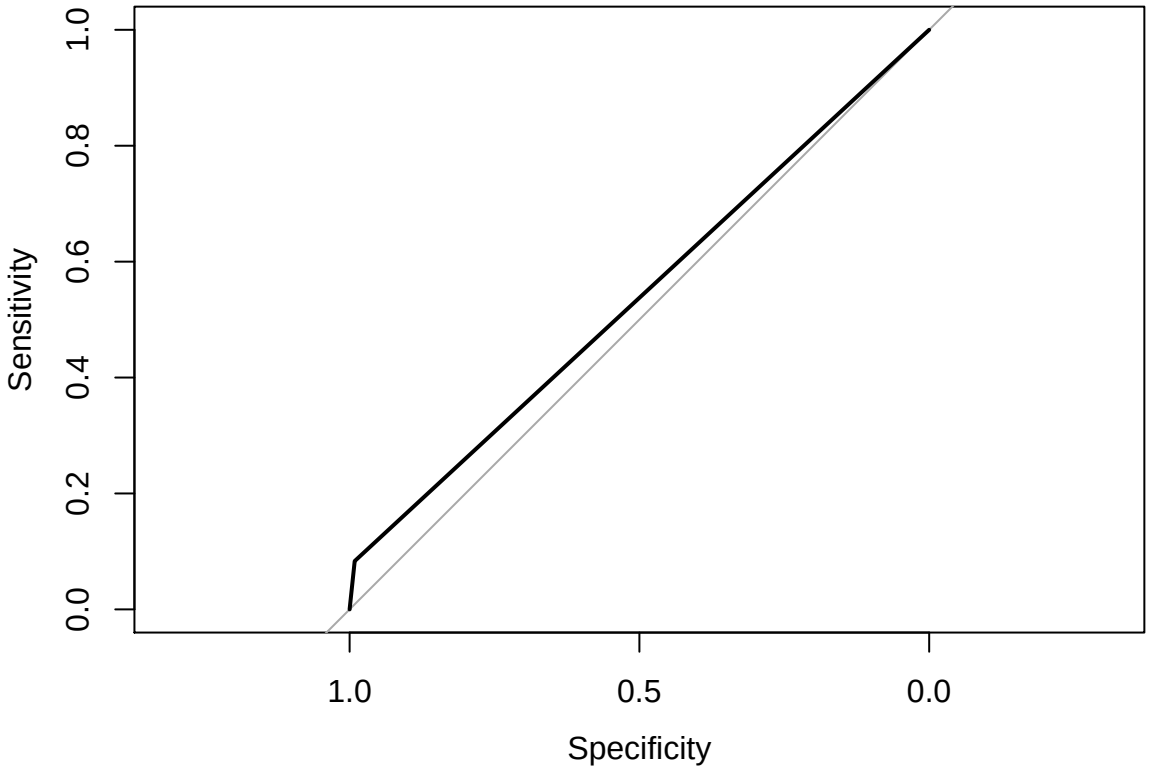
```
## [1] 77.33333
```

The accuracy rate of the predictions based on the above logistic model is **77.33%**.

Question 4d

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

```
#ROC curve
roc_glm <- roc(donor_test$Donated ~ donor_test$Prediction)
plot(roc_glm)
```




```
#Determine the area under the ROC curve
auc(roc_glm)
```

```
## Area under the curve: 0.5373
```

> The area under the curve is 0.5372. The area is not close to 1 and we cannot see a top left curve, but has a fair value.

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?

> As we know the time interval between each donation from the above variable, based on the last donation we can know if the person would donate blood in March 2007. Hence this would be more apt to predict if the person would donate blood.

Question 5b

Write a function to add this predictor to your *full dataset*. Call this variable `month_span_between_donations`.

```
transfusionData$month_span_between_donations <- transfusionData$Time/transfusionData$Frequency
```

Rerun the train test split

```
# creates a training dataset named train with rows which are marked as TRUE
donor_train <- subset(transfusionData, sample == TRUE)
# creates a training dataset named test with rows which are marked as FALSE
donor_test <- subset(transfusionData, sample == FALSE)
```

Question 5c

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

```
model_glm_avg_months <- glm(Donated ~ month_span_between_donations, family = "binomial",
                            data = donor_train)
summary(model_glm_avg_months)
```

```
##
## Call:
## glm(formula = Donated ~ month_span_between_donations, family = "binomial",
##     data = donor_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0661  -0.8278  -0.5766  -0.1064   2.7653
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.12335    0.18045  -0.684    0.494
## month_span_between_donations -0.14424    0.02468  -5.844 5.09e-09 ***
## ---
## 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: 603.52  on 596  degrees of freedom
## AIC: 607.52
##
## Number of Fisher Scoring iterations: 5
```

> This model is better compared to the model that was built with the independent variable 'Frequency' in question 3a, as the AIC value is lower($607.52 < 636.53$).

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.

```
#Predictions
donor_test$y_hat2 <- predict(model_glm_avg_months, donor_test, type = "response")
```

```
donor_test$Prediction2 <- ifelse(donor_test$y_hat2 > 0.4, 1, 0)
confMatrixResults <- confusionMatrix(data = factor(donor_test$Prediction2),
                                     reference = factor(donor_test$Donated))
confMatrixResults
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   0   1
##          0 114  35
##          1   0   1
##
##                Accuracy : 0.7667
##                  95% CI : (0.6907, 0.8318)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.4685
##
##                   Kappa : 0.0416
##  Mcnemar's Test P-Value : 9.081e-09
##
##             Sensitivity : 1.00000
##             Specificity : 0.02778
##          Pos Pred Value : 0.76510
##          Neg Pred Value : 1.00000
##              Prevalence : 0.76000
##          Detection Rate : 0.76000
##    Detection Prevalence : 0.99333
##       Balanced Accuracy : 0.51389
##
##        'Positive' Class : 0
##
```

There is **no false positive** in this prediction using the logistic model built with average number of months between donation. From the confusion matrix, we can see that out of total 150(114 + 35 + 0 + 1) predictions, 115(114 + 1) predictions are done right and 35 are wrong. All the 35 incorrect predictions are false negatives. The accuracy rate of the predictions based on the above logistic model is **76.66%**.

Question 5e

Use the `glm` 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`.

```
model_glm_multiple <- glm(Donated ~ Monetary + Recency, family = "binomial",
                          data = donor_train)
summary(model_glm_multiple)
```

```
##
## Call:
## glm(formula = Donated ~ Monetary + Recency, 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 ***
## Monetary     2.382e-04  6.746e-05   3.532 0.000413 ***
## Recency     -1.142e-01  1.825e-02  -6.257 3.92e-10 ***
```

```
## ---
## 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
```

```
#Predictions
donor_test$y_hat3 <- predict(model_glm_multiple, donor_test, type = "response")

#Classification
donor_test$Prediction3 <- ifelse(donor_test$y_hat3 > 0.4, 1, 0)
confMatrixResults <- confusionMatrix(data = factor(donor_test$Prediction3),
                                     reference = factor(donor_test$Donated))
confMatrixResults
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 106  28
##          1   8   8
##
##                Accuracy : 0.76
##                  95% CI : (0.6835, 0.8259)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.544587
##
##                   Kappa : 0.1877
##  Mcnemar's Test P-Value : 0.001542
##
##             Sensitivity : 0.9298
##             Specificity : 0.2222
##          Pos Pred Value : 0.7910
##          Neg Pred Value : 0.5000
##              Prevalence : 0.7600
##          Detection Rate : 0.7067
##    Detection Prevalence : 0.8933
##       Balanced Accuracy : 0.5760
##
##        'Positive' Class : 0
##
```

Question 5f

Calculate the accuracy rate of your y_hat3 predictions.

> From the summary of confusion matrix, we can see that the accuracy rate of y_hat3 predictions is **76%**.

Question 5f

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

```
#Find the correlation between each variables
corr_Matrix <- cor(transfusionData)
corr_Matrix
```

```
##                                Recency  Frequency   Monetary        Time
## Recency                      1.0000000 -0.1827455 -0.1827455  0.16061809
## Frequency                   -0.1827455  1.0000000  1.0000000  0.63494027
## Monetary                    -0.1827455  1.0000000  1.0000000  0.63494027
## Time                         0.1606181  0.6349403  0.6349403  1.00000000
## Donated                     -0.2798689  0.2186334  0.2186334 -0.03585441
## month_span_between_donations  0.6832801 -0.3203575 -0.3203575  0.23565908
##                                  Donated month_span_between_donations
## Recency                     -0.27986887                     0.6832801
## Frequency                    0.21863344                    -0.3203575
## Monetary                     0.21863344                    -0.3203575
## Time                        -0.03585441                     0.2356591
## Donated                      1.00000000                    -0.2520972
## month_span_between_donations -0.25209720                     1.0000000
```

From the correlation matrix, we notice that there is no significant high correlation between donated variable with any other variable. The highest correlation of donated variable is with Recency which is a negative correlation of 0.2798. There is a high positive correlation between month_span_between_donations and Recency variable which is intutive. Also there is a good positive correlation between Frequency/Monetary variable with Time. The correlation between Frequency and Time was to be anticipated. The more frequent the people have donated, the earlier they started donating.

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.)

stopped_donate: Recency/Time - If this number comes around 1, it means that the person has not donated blood since a long time. So, value closer to 1 would have stopped donating blood.

donation_rate: Frequency/Time - This provides us the average number of donations per month. Thus higher the rate, more likely to the person to continue blood donation.

Question 6

Another very popular classifier used in data science is called a *random forest*[1].

Question 6a

Use the `randomForest` 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`.

```
model_rf <- randomForest(Donated ~ Monetary + Recency, data = donor_train,
                         importance=TRUE, proximity=TRUE)
```

[1] https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
summary(model_rf)
```

```
##                 Length Class  Mode
## call                 5 -none- call
## type                 1 -none- character
## predicted          598 -none- numeric
## mse                500 -none- numeric
## rsq                500 -none- numeric
## oob.times          598 -none- numeric
## importance           4 -none- numeric
## importanceSD         2 -none- numeric
## localImportance      0 -none- NULL
## proximity       357604 -none- numeric
## ntree                1 -none- numeric
## mtry                 1 -none- numeric
## forest              11 -none- list
## coefs                0 -none- NULL
## y                  598 -none- numeric
## test                 0 -none- NULL
## inbag                0 -none- NULL
## terms                3 terms  call
```

```
#Predictions
donor_test$y_hat4 <- predict(model_rf, donor_test, type = "response")
```

```
#Classification
donor_test$Prediction4 <- ifelse(donor_test$y_hat4 > 0.4, 1, 0)
confMatrixResults <- confusionMatrix(data = factor(donor_test$Prediction4),
                                     reference = factor(donor_test$Donated))
confMatrixResults
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 97 19
##          1 17 17
##
##                Accuracy : 0.76
##                  95% CI : (0.6835, 0.8259)
##     No Information Rate : 0.76
##     P-Value [Acc > NIR] : 0.5446
##
##                   Kappa : 0.3294
##  Mcnemar's Test P-Value : 0.8676
##
##             Sensitivity : 0.8509
##             Specificity : 0.4722
##          Pos Pred Value : 0.8362
##          Neg Pred Value : 0.5000
##              Prevalence : 0.7600
##          Detection Rate : 0.6467
##    Detection Prevalence : 0.7733
```

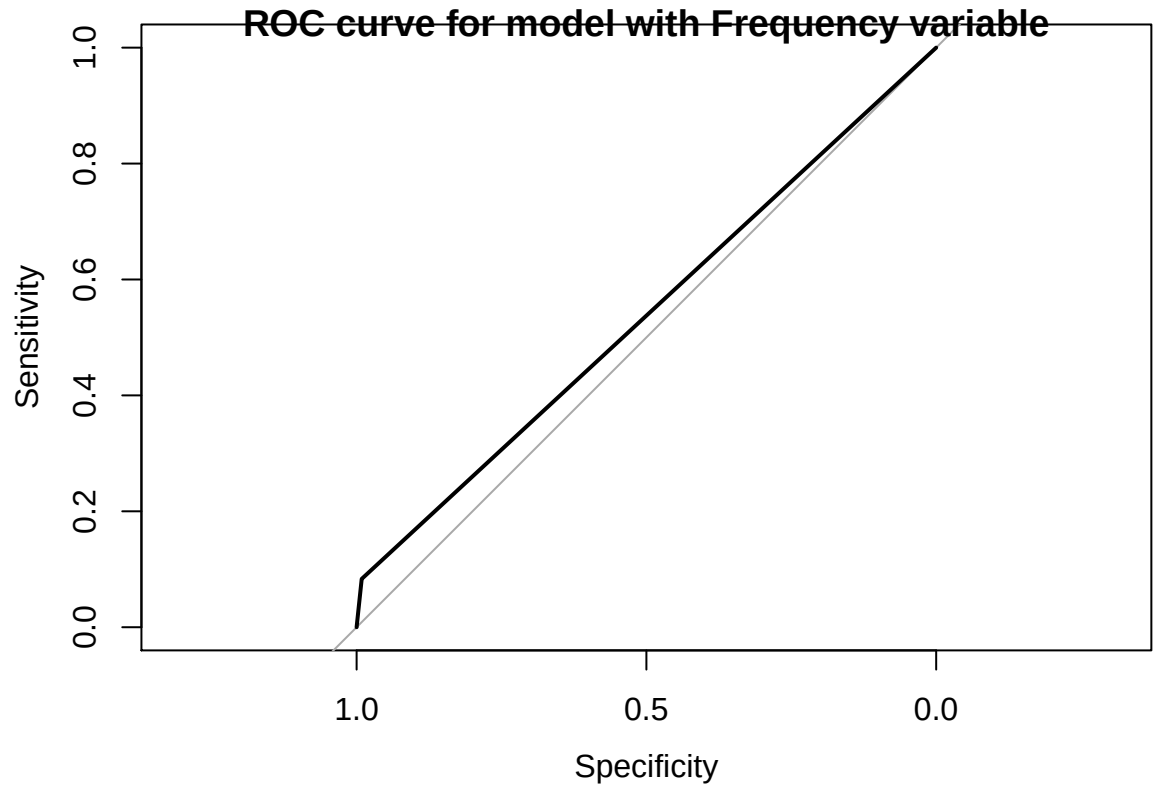
```
##        Balanced Accuracy : 0.6615
##
##         'Positive' Class : 0
##
```

The accuracy rate of the model built using random forest with the variables monetary and recency is **76%**.

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?

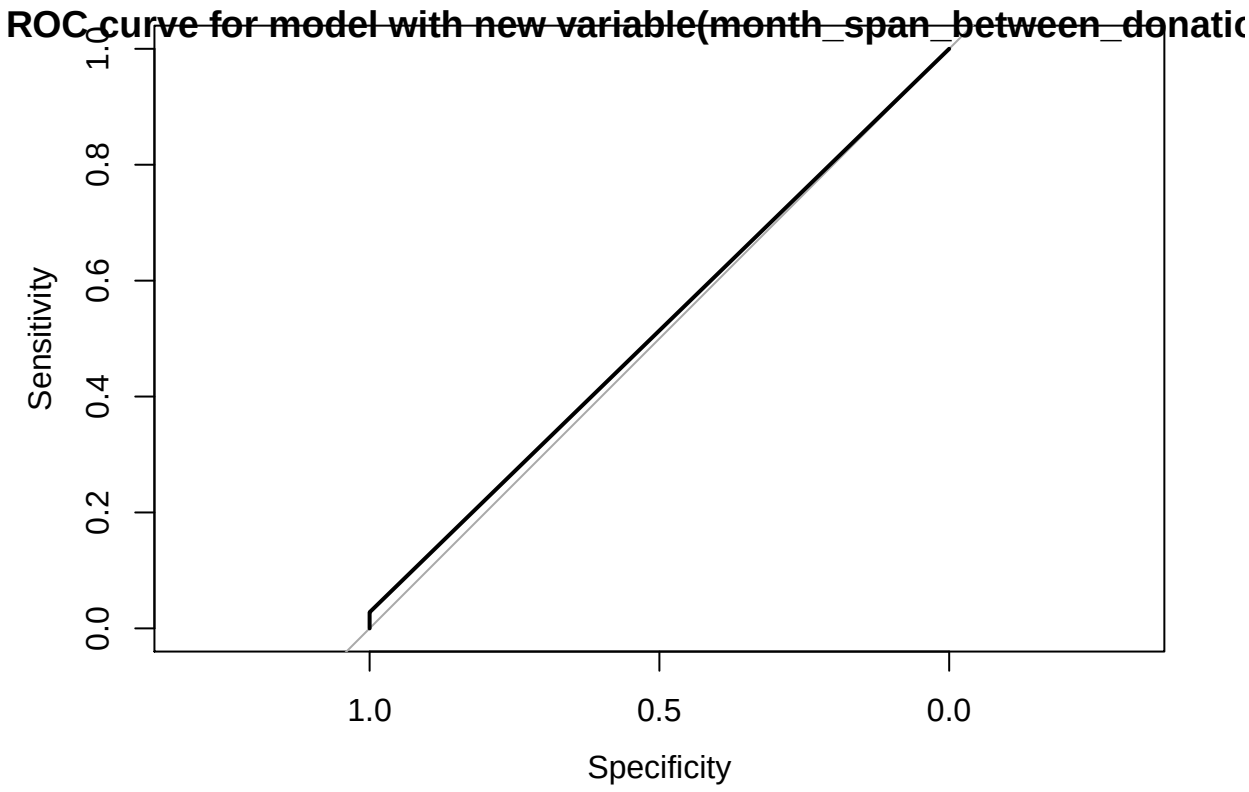
```
roc_glm_2 <- roc(donor_test$Donated ~ donor_test$Prediction2)
roc_glm_3 <- roc(donor_test$Donated ~ donor_test$Prediction3)
roc_glm_4 <- roc(donor_test$Donated ~ donor_test$Prediction4)

plot(roc_glm)
title("ROC curve for model with Frequency variable")
```




```
auc(roc_glm)
```

```
## Area under the curve: 0.5373
```

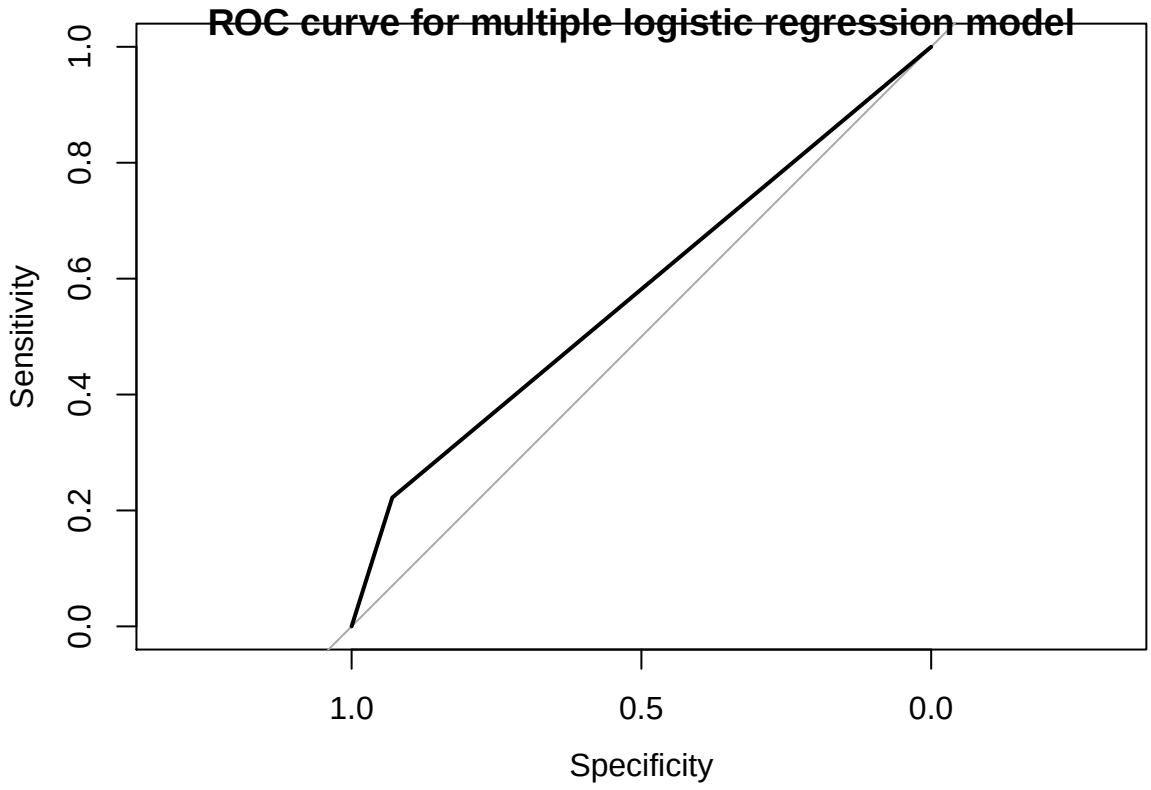
```
plot(roc_glm_2)
title("ROC curve for model with new variable(month_span_between_donations)")
```




```
auc(roc_glm_2)
```

```
## Area under the curve: 0.5139
```

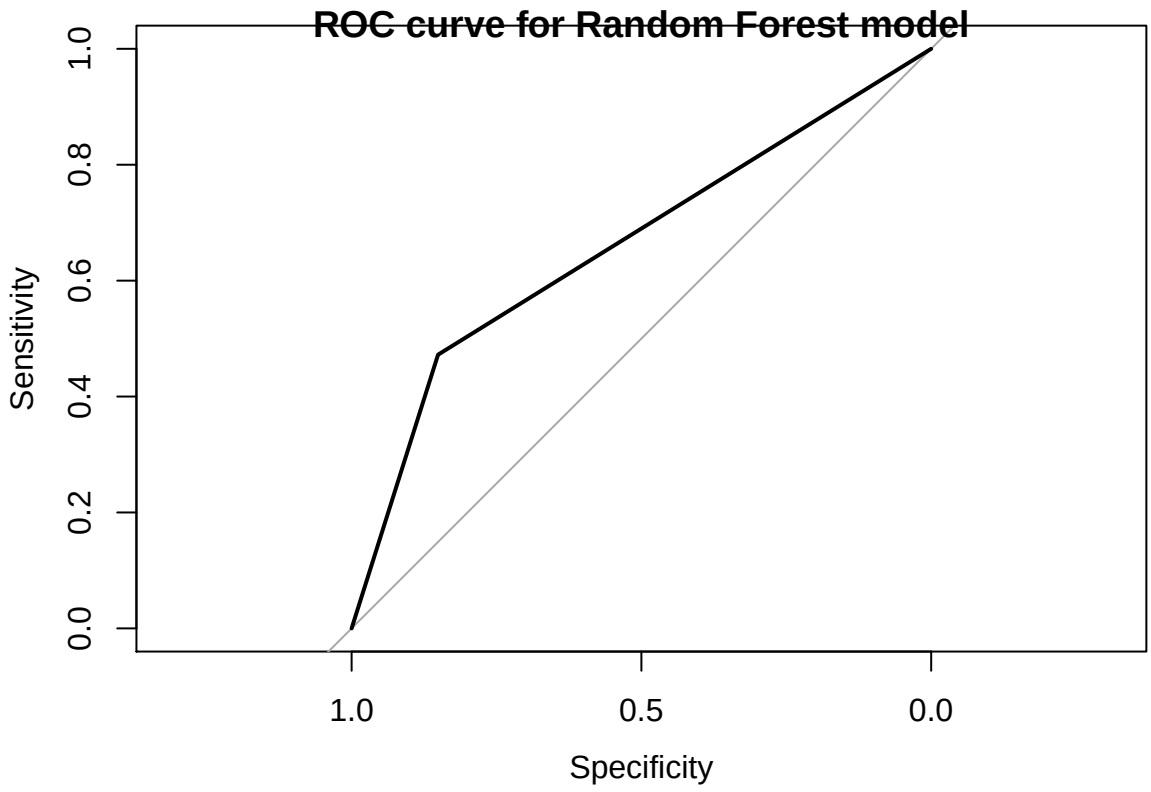
```
plot(roc_glm_3)
title("ROC curve for multiple logistic regression model")
```




```
auc(roc_glm_3)
```

```
## Area under the curve: 0.576
```

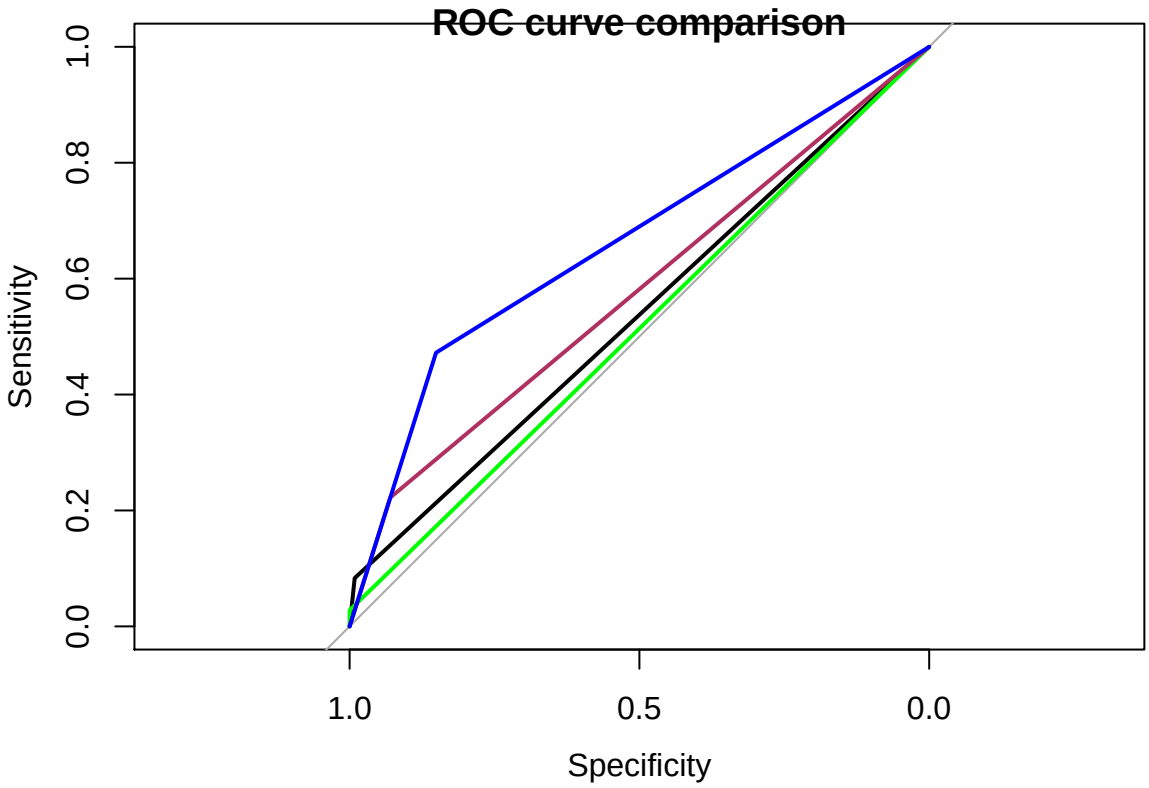
```
plot(roc_glm_4)
title("ROC curve for Random Forest model")
```




```
auc(roc_glm_4)
```

```
## Area under the curve: 0.6615
```

```
plot(roc_glm)
plot(roc_glm_2, add=TRUE, col='green')
plot(roc_glm_3, add=TRUE, col='maroon')
plot(roc_glm_4, add=TRUE, col='blue')
title("ROC curve comparison")
```



Though the accuracy rates are similar for all the models, the area under curve is comparatively high for random forest than logistic regression. But the number of false positives are less in logistic regression model than random forest model. Also the balanced accuracy rate is higher for random forest compared to other models. Random forest has a balanced accuracy rate of 66.15% while Logistic Regression has a Balanced Accuracy rate of around 51%-54% as per the values calculated earlier.