Predict422-CharityProject Part4

Artur Mrozowski

May 31, 2017

```
# Load packages required for this code.
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.2.5
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.2.3
```

```
## Loaded glmnet 2.0-5
```

1. Import Data

a. Read the data into R from the CSV file valSample.csv the same as you did in Part 1 Exercise 1.
b. Subset the data to only those observations where DONR = 1. This set of observations will be the data used for this assignment (Part 2, The Regression Problem). Name this dataset valData.

```
inPath = file.path("C:\\playground\\Predict422\\Project\\Part4\\")

valDataIn = read.csv(file.path(inPath,"valSample.csv"),na.strings=c("NA"," "))
#valData = valDataIn[valDataIn$DONR == "1",]
valData=valDataIn
rm(valDataIn)
```

```
# Convert categorical variables to factors
# This is highly recommended so that R treats the variables appropriately.
# The lm() method in R can handle a factor variable without us needing to convert
# the factor to binary dummy variable(s).
valData$DONR = as.factor(valData$DONR)
valData$HOME = as.factor(valData$HOME)
valData$HINC = as.factor(valData$HINC)
```

2. Predictions on Validation Set

a. Review the data preparation steps you took in Part 2 of the project. Apply those same data preparation steps to valData.

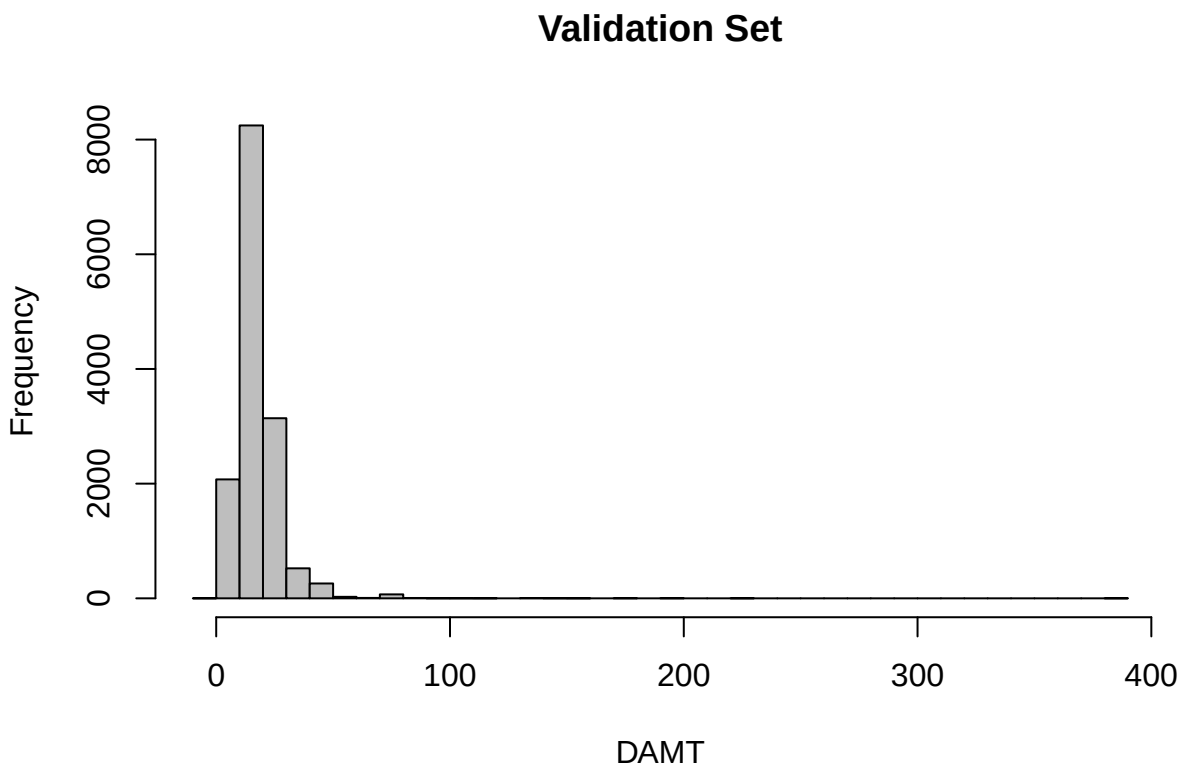
```
codePath = file.path("C:\\playground\\Predict422\\Project\\Part4\\")
source(file.path(codePath,"DataPreparation.R"))

valDataPart2 = processPart2(valData)
```

b. Using the model you chose from Part 2 (as trained on the Regression Training Set from Part 2), predict DAMT on the data coming from Step 2a.

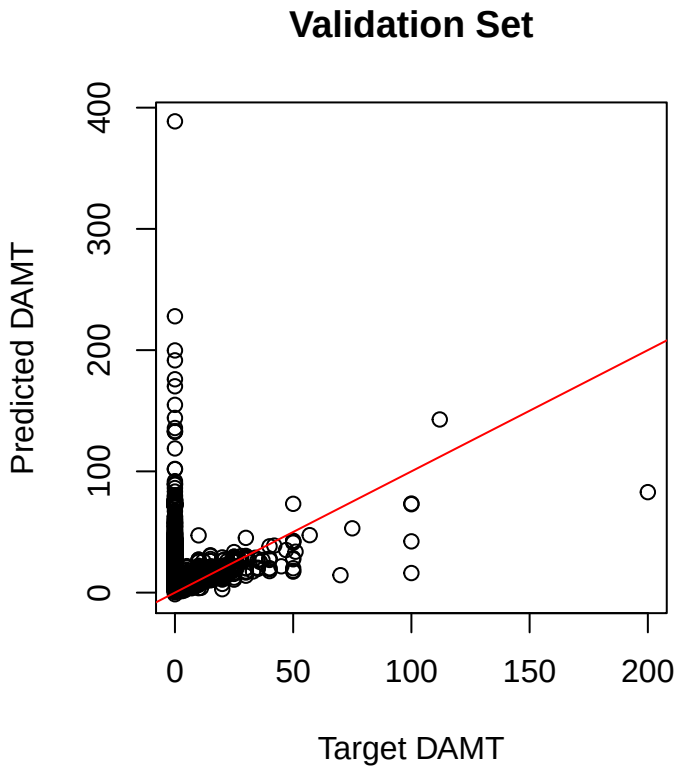
```
# Note that RFA_96_A for valData does not include the level "B". I had to do some
# investigating to track down an error that originated from this fact. Therefore, we
# will add the level so that we don't have problems with making predictions.
levels(valDataPart2$RFA_96_A) = c(levels(valDataPart2$RFA_96_A), "B")
```

```
modelPath = file.path("C:\\playground\\Predict422\\Project\\Part4\\")
load(file.path(modelPath,"modelPart2.RData"))
```

```
valData$DAMT.Pred = as.numeric(predict(modelPart2,newdata=valDataPart2))

# Check the predictions as a sanity check
hist(valData$DAMT.Pred,xlab="DAMT",main="Validation Set",col="gray",breaks=50)
```




```
par(pty="s")
plot(valData$DAMT,valData$DAMT.Pred,xlab="Target DAMT",ylab="Predicted DAMT",
     main="Validation Set")
abline(0,1,col="red")
```




```
par(pty="m")
```

c. Review the data preparation steps you took in Part 3 of the project. Apply those same data preparation steps to valData.

```
valDataPart3 = processPart3(valData)
levels(valDataPart3$RFA_96_A) = c(levels(valDataPart3$RFA_96_A), "B")
```

d. Using the model you chose from Part 3 (as trained on the Classification Training Set from Part 3), predict DONR and PDONR on the data coming from Step 2c.

```
load(file.path(modelPath,"modelPart3.RData"))
```

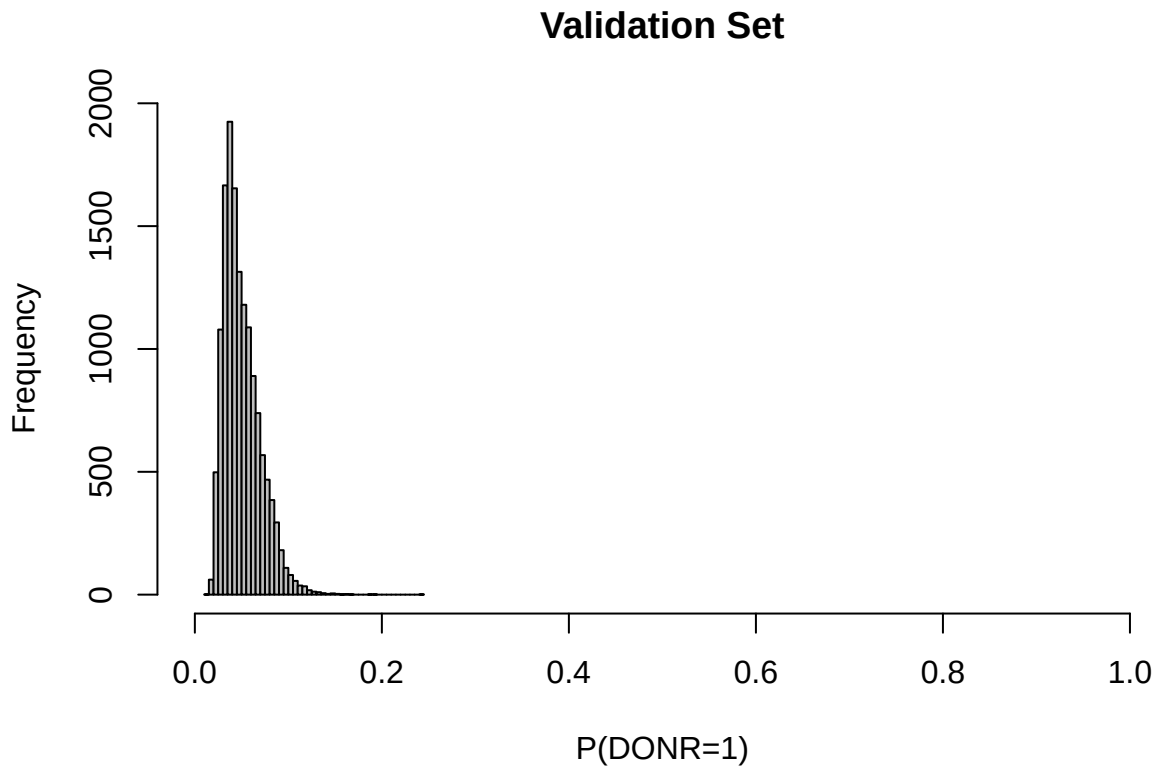
```
assignClass = function(probVals,threshVal)
{
  predVals = rep(0,length(probVals))
  predVals[probVals > threshVal] = 1
  predVals = factor(predVals)

  return(predVals)
}
```

```
# Further note that for a logistic regression model, the probabilities (PDONR) come
# from predict.glm and the classifications (DONR) come from applying the optimal threshold.
# Each predict method should have some means of obtaining the probabilities. You
# will have to check the documentation for the type of model you are using to
# determine the appropriate syntax.
valData$PDONR.Pred = predict(modelPart3,newdata=valDataPart3,type="response")
valData$DONR.Pred = assignClass(valData$PDONR.Pred,optThreshPart3)
```

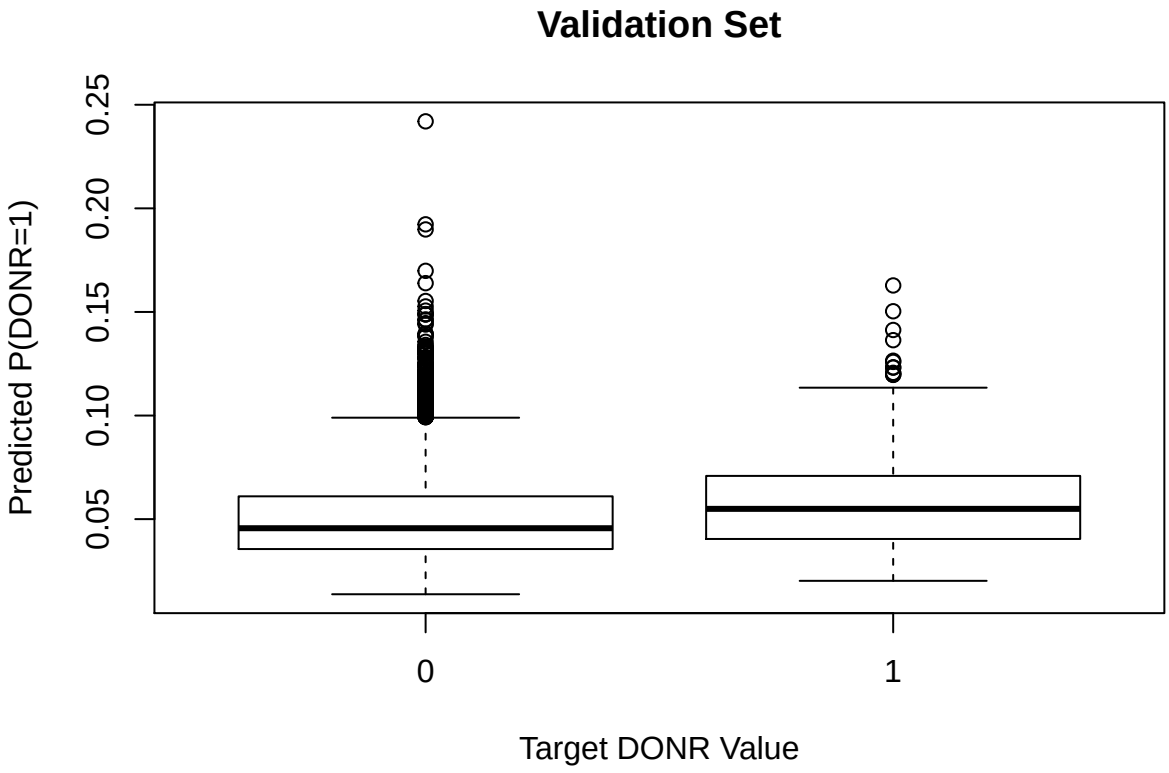
```
# Check the predictions as a sanity check
table(valData$DONR,valData$DONR.Pred,dnn=c("Target","Predicted"))
```

```
##       Predicted
## Target    0    1
##      0 8068 5566
##      1  313  424
```

```
hist(valData$PDONR.Pred,xlab="P(DONR=1)",main="Validation Set",col="gray",breaks=50,
     xlim=c(0,1))
```




```
plot(valData$DONR,valData$PDONR.Pred,xlab="Target DONR Value",
     ylab="Predicted P(DONR=1)",main="Validation Set")
```



3. Mailing List Selection

```
source(file.path(codePath,"RankedDonorOutput.R"))
```

a. The mailing list selection strategy illustrated in the sample code requires you to choose a score to rank and select a cutoff to use on that score. Evaluate this strategy by ranking various scores and calculating the profit obtained on the validation dataset. Scores that you might consider using include the predicted values of DONR, PDONR, and EXAMT. Summarize your findings with tables and figures as appropriate.

```
# Rank donors by PDONR.Pred
numBins = 10
out1 = outputForRankedDonors(numBins,rankVar="PDONR.Pred",dataToRank=valData)
print(out1$Donor.Table)
```

```
##    Num.Mailed Donors Donations Cum.Mailed Cum.Donors Cum.Donations
## 1        1437    139    1247.0       1437        139        1247.0
## 2        1437     93    1088.0       2874        232        2335.0
## 3        1437    101    1373.5       4311        333        3708.5
## 4        1437     75    1350.5       5748        408        5059.0
## 5        1437     69    1145.0       7185        477        6204.0
## 6        1437     57    1007.0       8622        534        7211.0
## 7        1437     60    1046.5      10059        594        8257.5
## 8        1437     45    1043.0      11496        639        9300.5
```

```
## 9        1437     57    1259.0      12933        696       10559.5
## 10       1438     41    1164.0      14371        737       11723.5
```

```
print(out1$Mailing.Table)
```

```
##    Bins.Mailed Num.Mailed Num.Donors Success.Rate Total.Cost
## 1     1 thru 1       1437        139     9.672930    1422.63
## 2     1 thru 2       2874        232     8.072373    2845.26
## 3     1 thru 3       4311        333     7.724426    4267.89
## 4     1 thru 4       5748        408     7.098121    5690.52
## 5     1 thru 5       7185        477     6.638831    7113.15
## 6     1 thru 6       8622        534     6.193459    8535.78
## 7     1 thru 7      10059        594     5.905160    9958.41
## 8     1 thru 8      11496        639     5.558455   11381.04
## 9     1 thru 9      12933        696     5.381582   12803.67
## 10   1 thru 10      14371        737     5.128384   14227.29
##    Total.Donations Total.Profit Average.Donation
## 1           1247.0      -175.63         8.971223
## 2           2335.0      -510.26        10.064655
## 3           3708.5      -559.39        11.136637
## 4           5059.0      -631.52        12.399510
## 5           6204.0      -909.15        13.006289
## 6           7211.0     -1324.78        13.503745
## 7           8257.5     -1700.91        13.901515
## 8           9300.5     -2080.54        14.554773
## 9          10559.5     -2244.17        15.171695
## 10         11723.5     -2503.79        15.907056
```

```
# Rank donors by EXAMT.Pred (expected donation amount)
# EXAMT.Pred = PDONR.Pred * DAMT.Pred
# (likelihood of donation * predicted donation amount)
valData$EXAMT.Pred = valData$PDONR.Pred * valData$DAMT.Pred
out2 = outputForRankedDonors(numBins,rankVar="EXAMT.Pred",dataToRank=valData)
print(out2$Donor.Table)
```

```
##    Num.Mailed Donors Donations Cum.Mailed Cum.Donors Cum.Donations
## 1        1437     83    2248.0       1437         83        2248.0
## 2        1437     69    1322.5       2874        152        3570.5
## 3        1437     75    1315.5       4311        227        4886.0
## 4        1437     77    1223.0       5748        304        6109.0
## 5        1437     70    1017.0       7185        374        7126.0
## 6        1437     70    1240.0       8622        444        8366.0
## 7        1437     54     756.0      10059        498        9122.0
## 8        1437     81    1030.0      11496        579       10152.0
## 9        1437     68     887.0      12933        647       11039.0
## 10       1438     90     684.5      14371        737       11723.5
```

```
print(out2$Mailing.Table)
```

```
##    Bins.Mailed Num.Mailed Num.Donors Success.Rate Total.Cost
## 1     1 thru 1       1437         83     5.775922    1422.63
## 2     1 thru 2       2874        152     5.288796    2845.26
```

```
## 3     1 thru 3       4311        227     5.265600    4267.89
## 4     1 thru 4       5748        304     5.288796    5690.52
## 5     1 thru 5       7185        374     5.205289    7113.15
## 6     1 thru 6       8622        444     5.149617    8535.78
## 7     1 thru 7      10059        498     4.950790    9958.41
## 8     1 thru 8      11496        579     5.036534   11381.04
## 9     1 thru 9      12933        647     5.002706   12803.67
## 10   1 thru 10      14371        737     5.128384   14227.29
##    Total.Donations Total.Profit Average.Donation
## 1           2248.0       825.37         27.08434
## 2           3570.5       725.24         23.49013
## 3           4886.0       618.11         21.52423
## 4           6109.0       418.48         20.09539
## 5           7126.0        12.85         19.05348
## 6           8366.0      -169.78         18.84234
## 7           9122.0      -836.41         18.31727
## 8          10152.0     -1229.04         17.53368
## 9          11039.0     -1764.67         17.06182
## 10         11723.5     -2503.79         15.90706
```

```
# Rank donors by DAMT.Pred (predicted donation amount)
out3 = outputForRankedDonors(numBins,rankVar="DAMT.Pred",dataToRank=valData)
print(out3$Donor.Table)
```

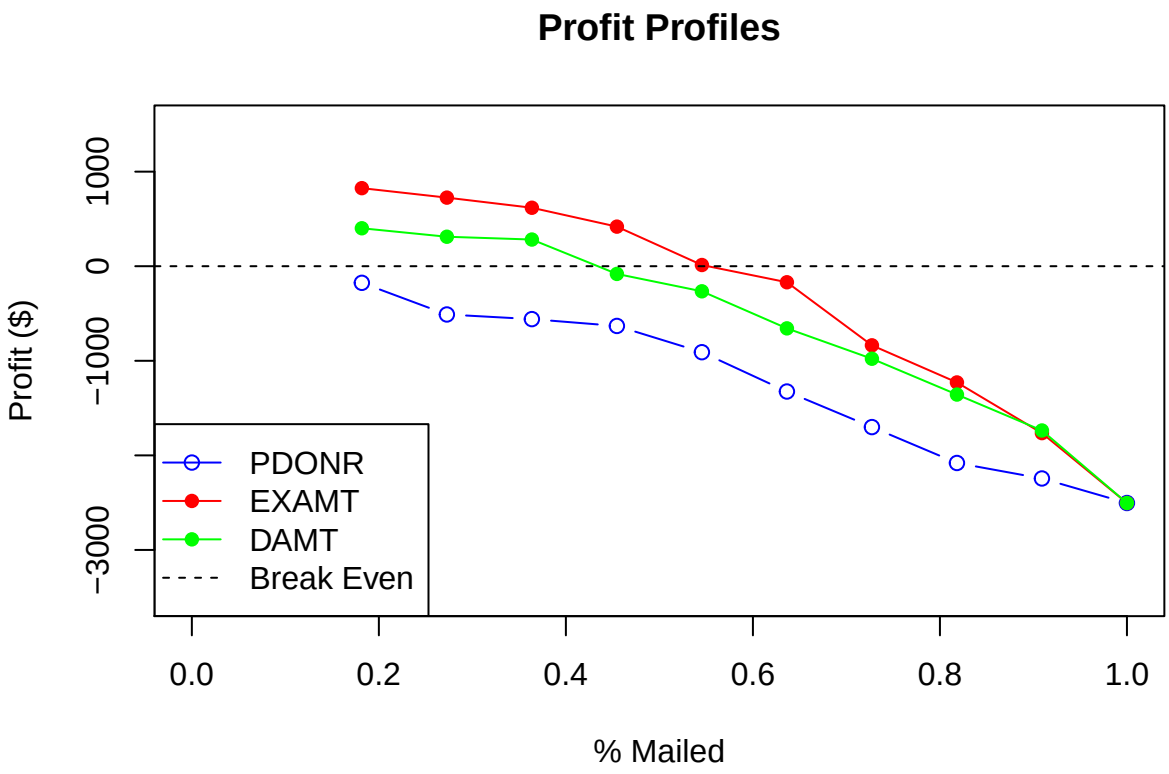
```
##    Num.Mailed Donors Donations Cum.Mailed Cum.Donors Cum.Donations
## 1        1437     40    1824.0       1437         40        1824.0
## 2        1437     55    1333.0       2874         95        3157.0
## 3        1437     70    1393.0       4311        165        4550.0
## 4        1437     57    1059.0       5748        222        5609.0
## 5        1437     66    1239.0       7185        288        6848.0
## 6        1437     66    1030.5       8622        354        7878.5
## 7        1437     74    1101.5      10059        428        8980.0
## 8        1437     90    1044.0      11496        518       10024.0
## 9        1437    110    1043.0      12933        628       11067.0
## 10       1438    109     656.5      14371        737       11723.5
```

```
print(out3$Mailing.Table)
```

```
##    Bins.Mailed Num.Mailed Num.Donors Success.Rate Total.Cost
## 1     1 thru 1       1437         40     2.783577    1422.63
## 2     1 thru 2       2874         95     3.305498    2845.26
## 3     1 thru 3       4311        165     3.827418    4267.89
## 4     1 thru 4       5748        222     3.862213    5690.52
## 5     1 thru 5       7185        288     4.008351    7113.15
## 6     1 thru 6       8622        354     4.105776    8535.78
## 7     1 thru 7      10059        428     4.254896    9958.41
## 8     1 thru 8      11496        518     4.505915   11381.04
## 9     1 thru 9      12933        628     4.855795   12803.67
## 10   1 thru 10      14371        737     5.128384   14227.29
##    Total.Donations Total.Profit Average.Donation
## 1           1824.0       401.37         45.60000
## 2           3157.0       311.74         33.23158
## 3           4550.0       282.11         27.57576
```

```
## 4           5609.0       -81.52         25.26577
## 5           6848.0      -265.15         23.77778
## 6           7878.5      -657.28         22.25565
## 7           8980.0      -978.41         20.98131
## 8          10024.0     -1357.04         19.35135
## 9          11067.0     -1736.67         17.62261
## 10         11723.5     -2503.79         15.90706
```

```
# Calculate percentiles of breakVals for each profile using the empircal CDF function.
fn1 = ecdf(out1$breakVals)
fn2 = ecdf(out2$breakVals)
fn3 = ecdf(out3$breakVals)
yLimits = c(-500+1000*floor(min(c(
              out1$Mailing.Table$Total.Profit,
              out2$Mailing.Table$Total.Profit,
              out3$Mailing.Table$Total.Profit
            ))/1000),
            500+1000*ceiling(max(c(
              out1$Mailing.Table$Total.Profit,
              out2$Mailing.Table$Total.Profit,
              out3$Mailing.Table$Total.Profit
            ))/1000))
plot(fn1(out1$breakVals)[-1],out1$Mailing.Table$Total.Profit,type='b',col="blue",
     xlab="% Mailed",ylab="Profit ($)",main="Profit Profiles",xlim=c(0,1),ylim=yLimits)
lines(fn2(out2$breakVals)[-1],out2$Mailing.Table$Total.Profit,col="red")
points(fn2(out2$breakVals)[-1],out2$Mailing.Table$Total.Profit,col="red",pch=16)
lines(fn3(out3$breakVals)[-1],out3$Mailing.Table$Total.Profit,col="green")
points(fn3(out3$breakVals)[-1],out3$Mailing.Table$Total.Profit,col="green",pch=16)
abline(h=0,lty=2)
legend(x="bottomleft",legend=c("PDONR","EXAMT","DAMT","Break Even"),
       col=c("blue","red","green","black"),
       lty=c(1,1,1,2),pch=c(1,16,16,NA))
```




```
cutOff = out3$breakVals[numBins+1-2]
valMailList = data.frame(ID=valData$ID[valData$DAMT.Pred >= cutOff])
length(valMailList$ID)
```

```
## [1] 2875
```

4. Predictions on Test Set In this exercise, you will make predictions on the Test Set data provided in testSample.csv. You will then select individuals from the Test Set to be mailed in the upcoming charity mailing campaign.

a. Repeat Exercise 1 of this assignment applied to the data in testSample.csv.

```
testData = read.csv(file.path(inPath,"testSample.csv"),na.strings=c("NA"," "))

testData$HOME = as.factor(testData$HOME)
testData$HINC = as.factor(testData$HINC)
```

b. Repeat Exercise 2 of this assignment applied to the data in testSample.csv.

```
# Note: The model.matrix method will not allow us to use a dataframe with "missing"
# columns. Therefore, we add dummy DAMT and DONR columns to testData.
testData$DAMT = -1
testData$DONR = -1
```

```
## Apply the Part 2 data processing steps to testData
testDataPart2 = processPart2(testData)
levels(testDataPart2$RFA_96_A) = c(levels(testDataPart2$RFA_96_A), "B")
```

```
## Predict DAMT for testData using your chosen model from Part 2
# Note that the model I am using is a glmnet model.
x = model.matrix(DAMT ~ .-ID,data=testDataPart2)[,-1]
testData$DAMT.Pred = as.numeric(predict(modelPart2,newdata = testDataPart2))
```

```
# Check the predictions as a sanity check
summary(testData$DAMT.Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.368  11.740  16.920  17.830  20.380 613.000
```

```
## Apply the Part 3 data processing steps to valData
testDataPart3 = processPart3(testData)
levels(testDataPart3$RFA_96_A) = c(levels(testDataPart3$RFA_96_A), "B")

## Predict DONR and PDONR for valData using your chosen model from Part 3
# Note that the model I am using is a glm model.
testData$PDONR.Pred = predict(modelPart3,newdata=testDataPart3,type="response")
testData$DONR.Pred = assignClass(testData$PDONR.Pred,optThreshPart3)

# Check the predictions as a sanity check
table(testData$DONR.Pred)
```

```
##
##     0     1
## 33226 23879
```

```
summary(testData$PDONR.Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01272 0.03578 0.04619 0.05057 0.06189 0.24200
```

c. Write your predictions out to a CSV file called projectPredictionsTEST.csv. This CSV file should contain the following columns: ID, DONR, PDONR, and DAMT.

```
## Part C - Write Test Set Predictions to CSV File
# Name the columns in the CSV file ID, DONR, PDONR, DAMT
testPredOut = data.frame(ID = testData$ID,
                         DONR = testData$DONR.Pred,
                         PDONR = testData$PDONR.Pred,
                         DAMT = testData$DAMT.Pred)

outPath = file.path("C:\\playground\\Predict422\\Project\\Part4\\")

write.csv(testPredOut,file=file.path(outPath,"projectPredictionsTEST.csv"),row.names=FALSE)
```

d. Apply the mailing list selection strategy that you chose in Exercise 3b to the Test Set.

```r
# Use cutoff selected above.
testMailList = data.frame(ID=testData$ID[testData$DAMT.Pred >= cutOff])
length(testMailList$ID)
```

```
## [1] 11339
```

e. Write the ID numbers of individuals selected for the mailing list to a CSV file called projectListTEST.csv. This CSV file needs only a single column: ID.

```r
write.csv(testMailList,file=file.path(outPath,"projectListTEST.csv"),row.names=FALSE)
```