STA 6543 Final Project

Krischelle Joyner

8/8/2021

Background

A national veterans' organization wishes to develop a predictive model to improve the cost-effectiveness of their direct marketing campaign. The organization, with its in-house database of over 13 million donors, is one of the largest direct-mail fundraisers in the United States. According to their recent mailing records, the overall response rate is 5.1%. Out of those who responded (donated), the average donation is $13.00. Each mailing, which includes a gift of personalized address labels and assortments of cards and envelopes, costs $0.68 to produce and send. Using these facts, we take a sample of this dataset to develop a classification model that can effectively capture donors so that the expected net profit is maximized. Weighted sampling was used, under-representing the non-responders so that the sample has equal numbers of donors and non-donors.

Business Objectives and Goals

Find a classification model that targets donors to gain a higher return on investment. This model will be used in the next donation campaign.

Data Sources and Data used:

```
#install.packages("pacman")
pacman::p_load('ISLR', 'corrgram', 'glmnet', 'pls', 'tidyverse', 'ggthemes',
'ggthemr', 'caret', 'modelr', 'leaps', 'psych', 'pastecs', 'e1071',
'randomForest', 'gbm', 'ROCR', 'recipes', 'broom', 'scales', 'outliers',
'MASS', 'VIF')

set.seed(12345)
```

The key to overcoming classification problems is the use of weighted samples. Otherwise, the disparity in frequency of observed classes can have a significant negative impact on model fitting. A random sample can be biased if there are a lot of non-responders, so we should use a weighted sampling to avoid this. Otherwise, we would have inaccurate results.

Exclusions:

```
f1 <- read_rds("fundraising.rds")
future_fundraising <- read_rds("future_fundraising.rds")

any(is.na(f1))

## [1] FALSE

any(is.na(future_fundraising))
```

```
## [1] FALSE
```

Since there weren't any missing values, we did not exclude any data.

Variable transformations:

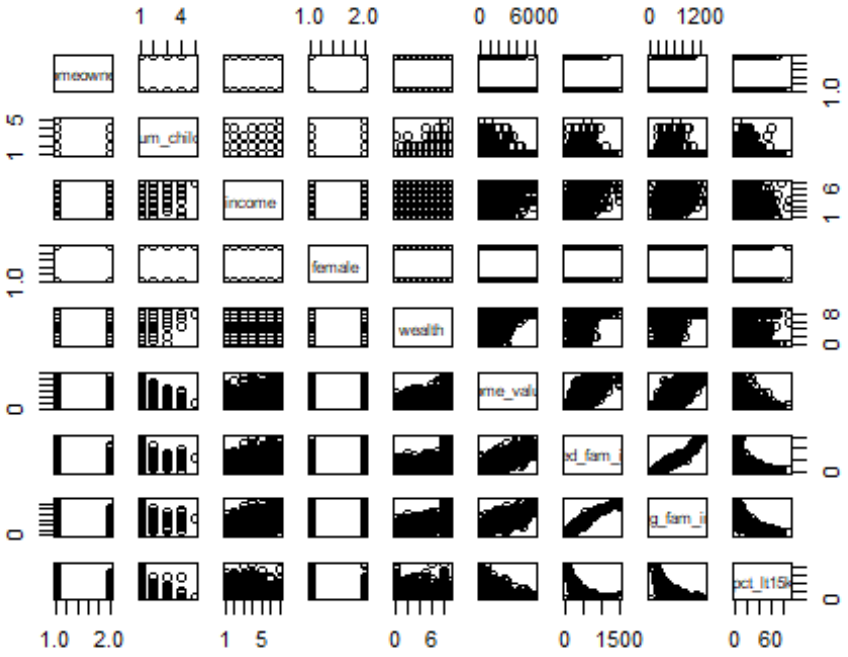
Due to some fields having zeros as the minimum value, it would be best to apply sqrt transformations to them and log transforms to the ones that do not include any 0's.The application of transformations seems useful, especially to predictors that will go into the final model. Ideally, some imputation should have been applied for these predictors. However, I did not do any transforms in my model.

Type of Analysis performed: what, why, findings.

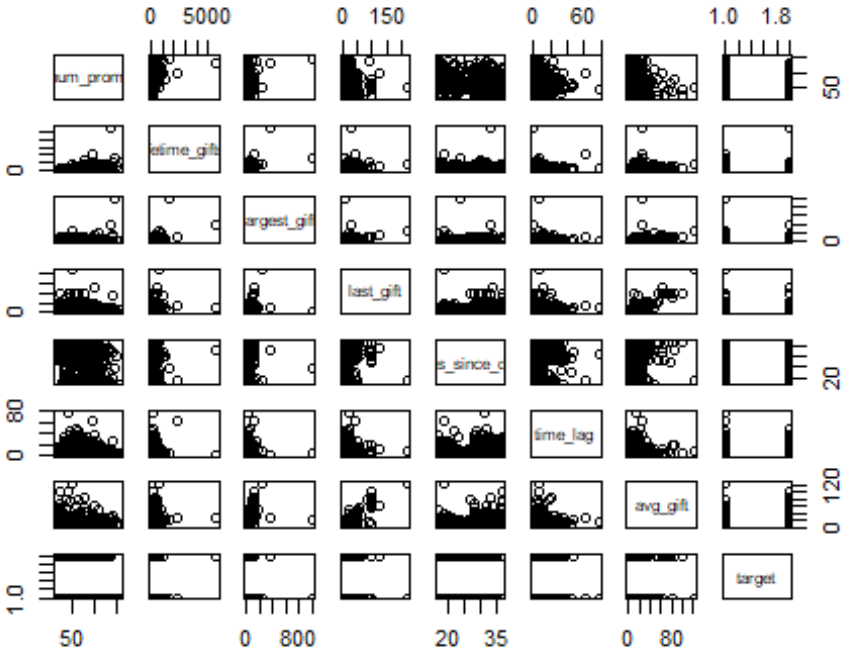
Exploratory Data Analysis:

```
summary(f1)

##  zipconvert2 zipconvert3 zipconvert4 zipconvert5 homeowner    num_child
##  No :2352    Yes: 551    No :2357    No :1846    Yes:2312   Min.   :1.000
##  Yes: 648    No :2449    Yes: 643    Yes:1154    No : 688   1st Qu.:1.000
##                                                             Median :1.000
##                                                             Mean   :1.069
##                                                             3rd Qu.:1.000
##                                                             Max.   :5.000
##      income      female          wealth        home_value      med_fam_inc
##  Min.   :1.000   Yes:1831   Min.   :0.000   Min.   :   0.0   Min.   :
0.0
##  1st Qu.:3.000   No :1169   1st Qu.:5.000   1st Qu.: 554.8   1st Qu.:
278.0
##  Median :4.000              Median :8.000   Median : 816.5   Median :
355.0
##  Mean   :3.899              Mean   :6.396   Mean   :1143.3   Mean   :
388.4
##  3rd Qu.:5.000              3rd Qu.:8.000   3rd Qu.:1341.2   3rd Qu.:
465.0
##  Max.   :7.000              Max.   :9.000   Max.   :5945.0   Max.
:1500.0
##   avg_fam_inc       pct_lt15k        num_prom      lifetime_gifts
##  Min.   :   0.0   Min.   : 0.00   Min.   : 11.00   Min.   :  15.0
##  1st Qu.: 318.0   1st Qu.: 5.00   1st Qu.: 29.00   1st Qu.:  45.0
##  Median : 396.0   Median :12.00   Median : 48.00   Median :  81.0
##  Mean   : 432.3   Mean   :14.71   Mean   : 49.14   Mean   : 110.7
##  3rd Qu.: 516.0   3rd Qu.:21.00   3rd Qu.: 65.00   3rd Qu.: 135.0
##  Max.   :1331.0   Max.   :90.00   Max.   :157.00   Max.   :5674.9
##   largest_gift       last_gift      months_since_donate    time_lag
##  Min.   :   5.00   Min.   :  0.00   Min.   :17.00       Min.   : 0.000
##  1st Qu.:  10.00   1st Qu.:  7.00   1st Qu.:29.00       1st Qu.: 3.000
##  Median :  15.00   Median : 10.00   Median :31.00       Median : 5.000
##  Mean   :  16.65   Mean   : 13.48   Mean   :31.13       Mean   : 6.876
```

```
##  3rd Qu.:  20.00   3rd Qu.: 16.00   3rd Qu.:34.00       3rd Qu.: 9.000
##  Max.   :1000.00   Max.   :219.00   Max.   :37.00       Max.   :77.000
##     avg_gift           target
##  Min.   :  2.139   Donor   :1499
##  1st Qu.:  6.333   No Donor:1501
##  Median :  9.000
##  Mean   : 10.669
##  3rd Qu.: 12.800
##  Max.   :122.167

corr_1 <- (f1[,c(6:7,9:21)])
corr_1$target <- as.numeric(corr_1$target)

pairs(f1[5:13])
```




```
pairs(f1[14:21])
```



Positive Correlation: There seems to be a positive correlation between home_value and med_fam_inc. There also seems to be a positive correlation between med_fam_inc and avg_fam_inc. Lastly, there seems to be a positive correlation between home_value and avg_fam_inc.

Negative Correlation: We see a negative correlation between med_fam_inc and pct_lt15k. We also see a negative correlation between avg_fam_inc and pct_lt15k.

```
library(VIF)
vif(as.data.frame(corr_1))

## [1] "m should be less than or equal to n"

## [1] 0
```

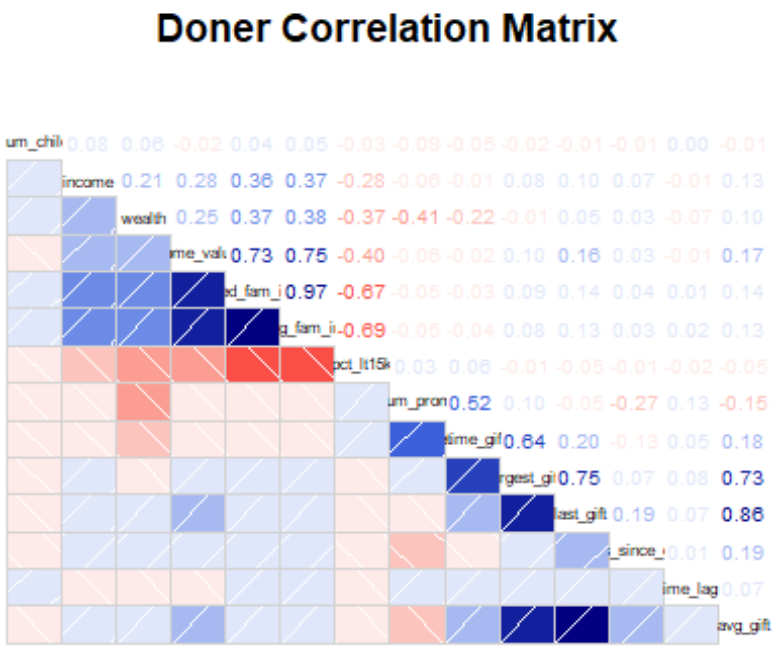
Methodology used, background, benefits:

From the company background we know the average donation to the national veterans organization is $13.00, and the average cost of supplies is $0.68. We are also given that approximately 50% of the data is 'Donors' and 50% is 'Non-donors'. Given these facts, we can calculate the maximum return on investment for the test set.

(13.00*299)-(0.68*300) = $3,683

So we can see that the maximum return on investment is $3,683.

```
D <- 299
ND <- 300
(13.00 * D)-(0.68 * ND)

## [1] 3683

training <- createDataPartition(f1$target,p=.8,list=FALSE)
train_data <- f1[training,]
test_data <- f1[-training,]
nrow(train_data)

## [1] 2401

train_control <- trainControl(method="repeatedcv",number=10,repeats=3)

corrgram(train_data, upper.panel=panel.cor, main="Doner Correlation Matrix")
```



We ran the correlation matrix to see which variables are highly correlated. We found that the following are highly correlated: med_fam_inc, home_value, avg_fam_inc, pct_lt15k, lifetime_gifts, last_gift, avg_gift.

LDA

We used LDA or Linear Discriminant Analysis which assumes the feature covariance matrices of both classes are the same resulting in a linear decision boundary. This LDA model uses all 20 variables.

```
library(caret)
lda.fit = train(target~., data=train_data, method='lda',trControl =
train_control)

pred.lda <- predict(lda.fit,test_data)

confusionMatrix(pred.lda,test_data$target)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Donor No Donor
##   Donor      174      147
##   No Donor   125      153
##
##                Accuracy : 0.5459
##                  95% CI : (0.5051, 0.5863)
##     No Information Rate : 0.5008
##     P-Value [Acc > NIR] : 0.01513
##
##                   Kappa : 0.0919
##
##  Mcnemar's Test P-Value : 0.20291
##
##             Sensitivity : 0.5819
##             Specificity : 0.5100
##          Pos Pred Value : 0.5421
##          Neg Pred Value : 0.5504
##              Prevalence : 0.4992
##          Detection Rate : 0.2905
##    Detection Prevalence : 0.5359
##       Balanced Accuracy : 0.5460
##
##        'Positive' Class : Donor
##
```

We can see from the confusion matrix that there are a total of 174 Donors and 153 Non Donors. The test accuracy is 54.59%.

```
D <- 174
ND <- 153
(13.00 * D)-(0.68 * ND)
```

```
## [1] 2157.96
```

The return on investment for LDA 1 is $2,157.96.

Let's look at 10 of the 20 variables for a second LDA analysis:

```
##Train Model
lda.fit2=train(target~ avg_gift + lifetime_gifts + med_fam_inc + avg_fam_inc
```

```
+ home_value + num_prom + pct_lt15k + months_since_donate + time_lag +
last_gift, data=train_data, method='lda',trControl = train_control)
##Calculate Predictions
pred.lda2<-predict(lda.fit2,test_data)
##Estimate Accuracy
confusionMatrix(pred.lda2,test_data$target)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction Donor No Donor
##   Donor      171      158
##   No Donor   128      142
##
##                Accuracy : 0.5225
##                  95% CI : (0.4817, 0.5632)
##     No Information Rate : 0.5008
##     P-Value [Acc > NIR] : 0.15351
##
##                   Kappa : 0.0452
##
##  Mcnemar's Test P-Value : 0.08638
##
##             Sensitivity : 0.5719
##             Specificity : 0.4733
##          Pos Pred Value : 0.5198
##          Neg Pred Value : 0.5259
##              Prevalence : 0.4992
##          Detection Rate : 0.2855
##    Detection Prevalence : 0.5492
##       Balanced Accuracy : 0.5226
##
##        'Positive' Class : Donor
##
```

We can see from the confusion matrix that there are a total of 174 Donors and 153 Non Donors. The test accuracy is 52.25%.

```
D2 <- 171
ND2 <- 142
(13.00 * D2)-(0.68 * ND2)

## [1] 2126.44
```

The return on investment for LDA 2 is $2,126.44.

QDA

We used QDA or Quadratic Discriminant Analysis which allows different feature covariance matrices for different classes, which leads to a quadratic decision boundary. This QDA model uses all 20 variables.

```
qda.fit = train(target~ homeowner +  num_child + income + female + home_value
+ med_fam_inc + avg_fam_inc + pct_lt15k + num_prom + lifetime_gifts +
largest_gift + last_gift + months_since_donate + time_lag + avg_gift,
data=train_data, method='qda',trControl = train_control)

pred.qda <- predict(qda.fit,test_data)

confusionMatrix(pred.qda,test_data$target)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Donor No Donor
##   Donor      227      214
##   No Donor    72       86
##
##                Accuracy : 0.5225
##                  95% CI : (0.4817, 0.5632)
##     No Information Rate : 0.5008
##     P-Value [Acc > NIR] : 0.1535
##
##                   Kappa : 0.0458
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7592
##             Specificity : 0.2867
##          Pos Pred Value : 0.5147
##          Neg Pred Value : 0.5443
##              Prevalence : 0.4992
##          Detection Rate : 0.3790
##    Detection Prevalence : 0.7362
##       Balanced Accuracy : 0.5229
##
##        'Positive' Class : Donor
##
```

We can see from the confusion matrix that there are a total of 227 Donors and 86 Non Donors. The test accuracy is 52.25%.

```
D3 <- 227
ND3 <- 86
(13.00 * D3)-(0.68 * ND3)
```

```
## [1] 2892.52
```

The return on investment for LDA 1 is $2,892.52.

Let's look at 10 of the 20 variables for a second QDA analysis:

```
##Train Model
qda.fit2 = train(target~ avg_gift + lifetime_gifts + med_fam_inc +
avg_fam_inc + home_value + num_prom + pct_lt15k + months_since_donate +
time_lag + last_gift, data=train_data, method='qda',trControl =
train_control)
##Calculate Predictions
pred.qda2<-predict(qda.fit2,test_data)
##Estimate Accuracy
confusionMatrix(pred.qda2,test_data$target)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Donor No Donor
##   Donor      201      183
##   No Donor    98      117
##
##                Accuracy : 0.5309
##                  95% CI : (0.49, 0.5714)
##     No Information Rate : 0.5008
##     P-Value [Acc > NIR] : 0.07632
##
##                   Kappa : 0.0622
##
##  Mcnemar's Test P-Value : 5.414e-07
##
##             Sensitivity : 0.6722
##             Specificity : 0.3900
##          Pos Pred Value : 0.5234
##          Neg Pred Value : 0.5442
##              Prevalence : 0.4992
##          Detection Rate : 0.3356
##    Detection Prevalence : 0.6411
##       Balanced Accuracy : 0.5311
##
##        'Positive' Class : Donor
##
```

We can see from the confusion matrix that there are a total of 174 Donors and 153 Non Donors. The test accuracy is 53.09%.

```
D4 <- 201
ND4 <- 117
(13.00 * D4)-(0.68 * ND4)
```

```
## [1] 2533.44
```

The return on investment for LDA 2 is $2,533.44.

Model performance and Validation Results:

We used the LDA and QDA classification models to training and validation. The model performance can be based on the two following factors:

Accuracy (Showing Top 2) -LDA 1 with 54.59 accuracy rate. -QDA 2 with 53.09 accuracy rate.

Return on Investment (Showing Top 2) -QDA 1 with an ROI of $2,892.52. -QDA 2 with an ROI of $2,533.44.

Cut-Off Analysis:

Due to the weighted sampling, we didn't use ROC curves and AUC when adjusting the threshold. The threshold is calculated using a default threshold or cutoff of 0.5.

Recommendations:

Our recommendation is to use the QDA 2 model, as it has the second highest ROI projection and the second highest accuracy rate based on the top two models in each category. Lastly, the models presented still have room for improvement; we suggest adding additional variables in the next campaign to allow ROI to continue to grow. These new variables like social media, could help to further target our audience.