Modeling Competition

Austin Pesina

5/7/2021

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:

The goal is to improve the cost-effectiveness of the veterans' organization direct marketing campaign with data analysis. The objective of this effort is to develop a classification model that can effectively capture donors so that the expected net profit is maximized.

Data Sources and Data used:

For this project, we were given with a data sample. The dataset was already generated with weighted sampling as the original dataset/population has heavy non-responders. The sample given has almost equal number of donors and non-donors.

```
future <- read_rds("~/Data_Mining/data/future_fundraising.rds")
fund <- read_rds("~/Data_Mining/data/fundraising.rds")

set.seed(12345)

str(future)
```

```
## tibble [120 x 20] (S3: tbl_df/tbl/data.frame)
##  $ zipconvert2        : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ zipconvert3        : Factor w/ 2 levels "Yes","No": 1 2 2 2 1 2 1 2 1 1 ...
##  $ zipconvert4        : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ zipconvert5        : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 1 1 ...
##  $ homeowner          : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 2 2 ...
```

```
##  $ num_child          : num [1:120] 1 1 1 1 1 1 1 1 1 1 ...
##  $ income             : num [1:120] 5 1 4 4 2 4 2 3 4 2 ...
##  $ female             : Factor w/ 2 levels "Yes","No": 1 2 1 2 1 1 1 2 2 2 ...
##  $ wealth             : num [1:120] 9 7 1 8 7 8 1 8 3 5 ...
##  $ home_value         : num [1:120] 1399 1355 835 1019 992 ...
##  $ med_fam_inc        : num [1:120] 637 411 310 389 524 371 209 253 302 335 ...
##  $ avg_fam_inc        : num [1:120] 703 497 364 473 563 408 259 285 324 348 ...
##  $ pct_lt15k          : num [1:120] 1 9 22 15 6 10 36 25 19 14 ...
##  $ num_prom           : num [1:120] 74 77 70 21 63 35 72 68 55 59 ...
##  $ lifetime_gifts     : num [1:120] 102 249 126 26 100 92 146 98 66 276 ...
##  $ largest_gift       : num [1:120] 6 15 6 16 20 37 12 5 7 15 ...
##  $ last_gift          : num [1:120] 5 7 6 16 3 37 11 3 5 13 ...
##  $ months_since_donate: num [1:120] 29 35 34 37 21 37 36 32 30 33 ...
##  $ time_lag           : num [1:120] 3 3 8 5 6 5 5 9 9 10 ...
##  $ avg_gift           : num [1:120] 4.86 9.58 4.34 13 7.69 ...
```

```
str(fund)
```

```
## tibble [3,000 x 21] (S3: tbl_df/tbl/data.frame)
##  $ zipconvert2        : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 2 1 2 ...
##  $ zipconvert3        : Factor w/ 2 levels "Yes","No": 2 2 2 1 1 2 2 2 2 2 ...
##  $ zipconvert4        : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 1 ...
##  $ zipconvert5        : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 2 1 1 2 1 ...
##  $ homeowner          : Factor w/ 2 levels "Yes","No": 1 2 1 1 1 1 1 1 1 1 ...
##  $ num_child          : num [1:3000] 1 2 1 1 1 1 1 1 1 1 ...
##  $ income             : num [1:3000] 1 5 3 4 4 4 4 4 4 1 ...
##  $ female             : Factor w/ 2 levels "Yes","No": 2 1 2 2 1 1 2 1 1 1 ...
##  $ wealth             : num [1:3000] 7 8 4 8 8 8 5 8 8 5 ...
##  $ home_value         : num [1:3000] 698 828 1471 547 482 ...
##  $ med_fam_inc        : num [1:3000] 422 358 484 386 242 450 333 458 541 203 ...
##  $ avg_fam_inc        : num [1:3000] 463 376 546 432 275 498 388 533 575 271 ...
##  $ pct_lt15k          : num [1:3000] 4 13 4 7 28 5 16 8 11 39 ...
##  $ num_prom           : num [1:3000] 46 32 94 20 38 47 51 21 66 73 ...
##  $ lifetime_gifts     : num [1:3000] 94 30 177 23 73 139 63 26 108 161 ...
##  $ largest_gift       : num [1:3000] 12 10 10 11 10 20 15 16 12 6 ...
##  $ last_gift          : num [1:3000] 12 5 8 11 10 20 10 16 7 3 ...
##  $ months_since_donate: num [1:3000] 34 29 30 30 31 37 37 30 31 32 ...
##  $ time_lag           : num [1:3000] 6 7 3 6 3 3 8 6 1 7 ...
##  $ avg_gift           : num [1:3000] 9.4 4.29 7.08 7.67 7.3 ...
##  $ target             : Factor w/ 2 levels "Donor","No Donor": 1 1 2 2 1 1 1 2 1 1 ...
```

```
table(fund$target)
```

```
##
##    Donor No Donor
##     1499     1501
```

A quick look at our data:
n = 3000
p = 20
y = target

Note that our target is a factor variable with 2 levels: “Donor” and “No Donor”.

Our sample dataset has 3000 observations ($n = 3000$) and our production dataset has 120 observations. As previously mentioned, a 50-50 weighted sample was used. There are 1499 "Donors" and 1501 "No Donors" which is a rate of 49.7% of "Donors".

Because we are dealing with a classification problem, we use a weighted sample. A simple random sample could potentially be biased towards one group, so we use a weighted sample to make sure we have an apporoximately even distribution.

Type of Analysis Performed

Exploratory Data Analysis

Here is how our analysis was performed:

- *Summary* of the sample to understand how the predictors data is distributed, any significant outliers, etc.
- *Correlation* of the predictors with the response variable and with each other to understand which predictors can heavily influence the model.
- *Collineraity* of the predictors with each other to understand if any exclusions can be made for the final model.

```
#Summary of the training sample

summary(fund)
```

```
##  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
```

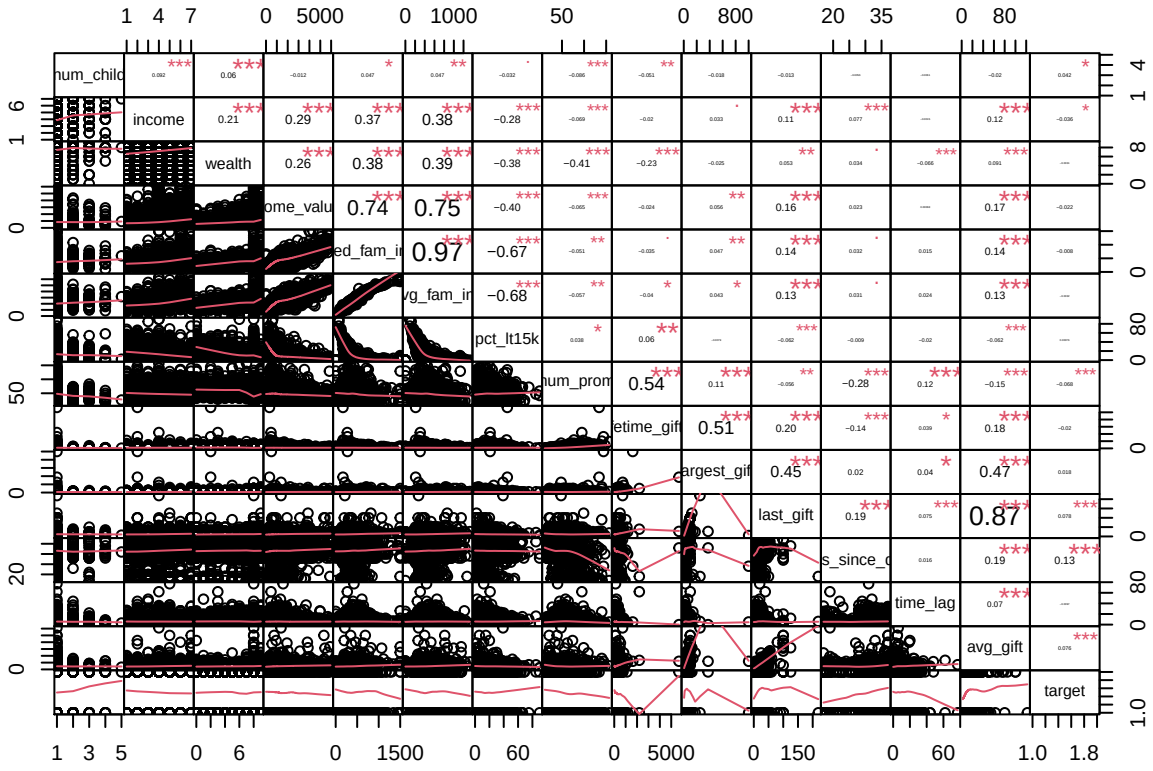
By looking at the summary, we see that our target is split almost evenly. We also see that `home_value`, `med_fam_inc`, `avg_fam_inc`, `pct_lt15k`, `num_prom`, `lifetime_gifts`, `largest_gift`, `last_gift`, `time_lag`, and `avg_gift` all look to contain outliers.

Despite being numerical, `num_child`, `income`, and `wealth` all seem to be categorical variables.

```
# Correlation

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

chart.Correlation(data, histogram=F, pch=19)
```



From the graphs, we see that few variables have a high correlation and only `num_child`, `income`, `num_prom`, `last_gift`, `months_since_donate`, and `avg_gift` are correlated with the response variable. We also see that `med_fam_inc` and `avg_fam_inc` are highly correlated with `income`.

```r
# Check collinearity
vif(as.data.frame(data))
```

```
##              Variables       VIF
## 1            num_child  1.027086
## 2               income  1.198361
## 3               wealth  1.509487
## 4           home_value  2.495523
## 5          med_fam_inc 18.433712
## 6          avg_fam_inc 20.709328
## 7            pct_lt15k  2.040823
## 8             num_prom  1.964372
## 9       lifetime_gifts  1.994304
## 10        largest_gift  1.715450
## 11           last_gift  4.155421
## 12 months_since_donate  1.159323
## 13            time_lag  1.032709
## 14            avg_gift  4.470251
## 15              target  1.030083
```

Here we see that `med_fam_inc` and `avg_fam_inc` are collinear. The only other collinear variables are `last_gift` and `avg_gift`.

Exclusions:

No variables were excluded.

Variable Transformations

As stated earlier, `home_value`, `med_fam_inc`, `avg_farm_inc`, `pct_lt15k`, `num_prom`, `lifetime_gifts`, `largest_gift`, `last_gift`, `time_lag`, and `avg_gift` all have large outliers.

Because we have values of 0, we can take the square root and then apply a log transformation on any non-zero values.

Methodology, Background, and Benefits

Partitioning

Two different approaches are used below.

```r
# Create partition
split = 0.80
trainIndex <- createDataPartition(fund$target,p=split,list=FALSE)
train <- fund[trainIndex,]
test <- fund[-trainIndex,]
train_control <- trainControl(method="repeatedcv",number=10,repeats=3)
```

- Cross Validation

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

Model Fit Approach:

Variable Importance:

Logistic regression was used to find the variables with significant p-values, <0.5.

```
glm_fit <- glm(target~., data = fund, family = "binomial")
summary(glm_fit)
```

```
##
## Call:
## glm(formula = target ~ ., family = "binomial", data = fund)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90432  -1.15349   0.00153   1.15919   1.79778
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.885e+00  4.595e-01  -4.102 4.10e-05 ***
## zipconvert2Yes      -1.365e+01  2.670e+02  -0.051  0.95924
## zipconvert3No        1.361e+01  2.670e+02   0.051  0.95934
## zipconvert4Yes      -1.365e+01  2.670e+02  -0.051  0.95922
## zipconvert5Yes      -1.365e+01  2.670e+02  -0.051  0.95922
## homeownerNo          4.957e-02  9.412e-02   0.527  0.59847
## num_child            2.752e-01  1.137e-01   2.422  0.01544 *
## income              -6.952e-02  2.595e-02  -2.679  0.00738 **
## femaleNo             5.995e-02  7.673e-02   0.781  0.43463
## wealth              -1.907e-02  1.800e-02  -1.059  0.28940
## home_value          -1.074e-04  7.141e-05  -1.503  0.13272
## med_fam_inc         -1.200e-03  9.303e-04  -1.289  0.19725
## avg_fam_inc          1.756e-03  1.010e-03   1.738  0.08226 .
## pct_lt15k           -9.519e-04  4.440e-03  -0.214  0.83024
## num_prom            -3.682e-03  2.317e-03  -1.589  0.11204
## lifetime_gifts       1.599e-04  3.721e-04   0.430  0.66743
## largest_gift        -1.773e-03  3.091e-03  -0.574  0.56629
## last_gift            9.923e-03  7.562e-03   1.312  0.18945
## months_since_donate  5.922e-02  1.003e-02   5.906 3.51e-09 ***
## time_lag            -6.174e-03  6.789e-03  -0.909  0.36311
## avg_gift             7.539e-03  1.106e-02   0.682  0.49526
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4158.9  on 2999  degrees of freedom
## Residual deviance: 4062.0  on 2979  degrees of freedom
## AIC: 4104
##
```

```
## Number of Fisher Scoring iterations: 12
```

```
coef(glm_fit)
```

```
##         (Intercept)      zipconvert2Yes       zipconvert3No      zipconvert4Yes
##       -1.884773e+00       -1.364510e+01        1.361190e+01       -1.365500e+01
##      zipconvert5Yes         homeownerNo           num_child              income
##       -1.365258e+01        4.956599e-02        2.752422e-01       -6.952306e-02
##            femaleNo              wealth          home_value         med_fam_inc
##        5.994674e-02       -1.907373e-02       -1.073676e-04       -1.199496e-03
##         avg_fam_inc           pct_lt15k            num_prom      lifetime_gifts
##        1.755584e-03       -9.519183e-04       -3.681972e-03        1.598657e-04
##        largest_gift          last_gift months_since_donate            time_lag
##       -1.772667e-03        9.923322e-03        5.921548e-02       -6.174095e-03
##            avg_gift
##        7.539422e-03
```

```
summary(glm_fit)$coef
```

```
##                          Estimate   Std. Error     z value     Pr(>|z|)
## (Intercept)         -1.884773e+00 4.595006e-01 -4.10178679 4.099720e-05
## zipconvert2Yes      -1.364510e+01 2.670209e+02 -0.05110124 9.592448e-01
## zipconvert3No        1.361190e+01 2.670209e+02  0.05097692 9.593439e-01
## zipconvert4Yes      -1.365500e+01 2.670209e+02 -0.05113831 9.592153e-01
## zipconvert5Yes      -1.365258e+01 2.670209e+02 -0.05112926 9.592225e-01
## homeownerNo          4.956599e-02 9.412356e-02  0.52660550 5.984676e-01
## num_child            2.752422e-01 1.136519e-01  2.42180072 1.544382e-02
## income              -6.952306e-02 2.595057e-02 -2.67905680 7.382987e-03
## femaleNo             5.994674e-02 7.672748e-02  0.78129420 4.346295e-01
## wealth              -1.907373e-02 1.800369e-02 -1.05943418 2.894021e-01
## home_value          -1.073676e-04 7.141352e-05 -1.50346346 1.327196e-01
## med_fam_inc         -1.199496e-03 9.302622e-04 -1.28941665 1.972533e-01
## avg_fam_inc          1.755584e-03 1.010277e-03  1.73772557 8.225918e-02
## pct_lt15k           -9.519183e-04 4.440127e-03 -0.21438986 8.302430e-01
## num_prom            -3.681972e-03 2.317003e-03 -1.58911007 1.120355e-01
## lifetime_gifts       1.598657e-04 3.720630e-04  0.42967362 6.674331e-01
## largest_gift        -1.772667e-03 3.090851e-03 -0.57352069 5.662922e-01
## last_gift            9.923322e-03 7.562225e-03  1.31222251 1.894451e-01
## months_since_donate  5.921548e-02 1.002632e-02  5.90600490 3.505036e-09
## time_lag            -6.174095e-03 6.788749e-03 -0.90945987 3.631074e-01
## avg_gift             7.539422e-03 1.105534e-02  0.68197131 4.952571e-01
```

```
glm_prob <- predict(glm_fit, type = "response")
contrasts(fund$target)
```

```
##          No Donor
## Donor           0
## No Donor        1
```

```
glm_pred <- rep("Donor", 3000)
glm_pred[glm_prob>0.5] = "No Donor"
table(glm_pred, fund$target)
```

```
##
## glm_pred   Donor No Donor
##   Donor      870      672
##   No Donor   629      829
```

```
mean(glm_pred==fund$target)
```

```
## [1] 0.5663333
```

Predictors for the final model:

1. We took the Top 10 predictors based on the random forest importance and left out any collinear predictors.

- `last_gift` is collinear with `avg_gift`
- `med_fam_inc` and `pct_lt15k` are both collinear with `income`

2. The predictors we used are:
    - num_child
    - income
    - home_value
    - months_since_donate
    - time_lag
3. time_lag was initially excluded from the model. Upon adding it back in, half the models performed 0.3% better on average.

Classification Models

For this analysis, several models were used including Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbor (KNN), Random Forest, and Support Vector Machines (SVM).

Model Performance and Validation Results

The final, best accuracy rate observed was 56.1% with logistic regression.

LDA (55.64%), SVM Linear (55.38), QDA (54.85), SVM Radial (54.35) were the other best performing models.

Logistic Regression

```
glm_fit_tl <- glm(target~num_child + income + home_value + months_since_donate + time_lag, data = train
glm_prob_tl <- predict(glm_fit_tl, type = "response")
glm_pred_tl <- rep("Donor", 2401)
glm_pred_tl[glm_prob_tl > 0.5] = "No Donor"
table(glm_pred_tl, train$target)
```

```
##
## glm_pred_tl Donor No Donor
##    Donor      679      533
##    No Donor   521      668
```

```
mean(glm_pred_tl==train$target)
```

```
## [1] 0.5610162
```

```
future_value <- predict(glm_fit_tl, future)
Value <- c("value", as.character(future_value))
Value <- if_else (Value > 0.5, "No Donor", "Donor")
write.csv(Value,file="~/final_glm.csv", row.names=F)
```

Prediction accuracy is 56.1%.

Support Vector Machines - Linear

```
set.seed(12345)
svm_fit2 <- train(target~num_child + income + home_value + months_since_donate, data = train, method =
future_value_svm <- predict(svm_fit2, future)
Value_svm <- c("value", as.character(future_value_svm))
write.csv(Value_svm,file="~/final_svm.csv", row.names=F)
```

The linear SVM had a 55.4% accuracy.