Situation:
A national veterans organization frequently solicits donations through direct mail campaigns to its database of current and prospective donors. They believe in a ‘test-and-learn’ culture and sent out a ***test*** mailing to collect data about who responds and who does not. They believe that machine learning can help them ***learn*** how to identify their strategy of how to better target prospective donors in the future.

Complication:
There are many practical challenges in dealing with noisy real world data. This dataset has many unknown values for many columns. Also there is very significant class imbalance in the outcome variable. As you will see there are 90,569 non-responders and only 4,843 responders in the dataset, giving us a 5% response rate from the test mailing. This is typical of many real world classification settings. How can we effectively learn from such a skewed dataset with respect to the outcome?

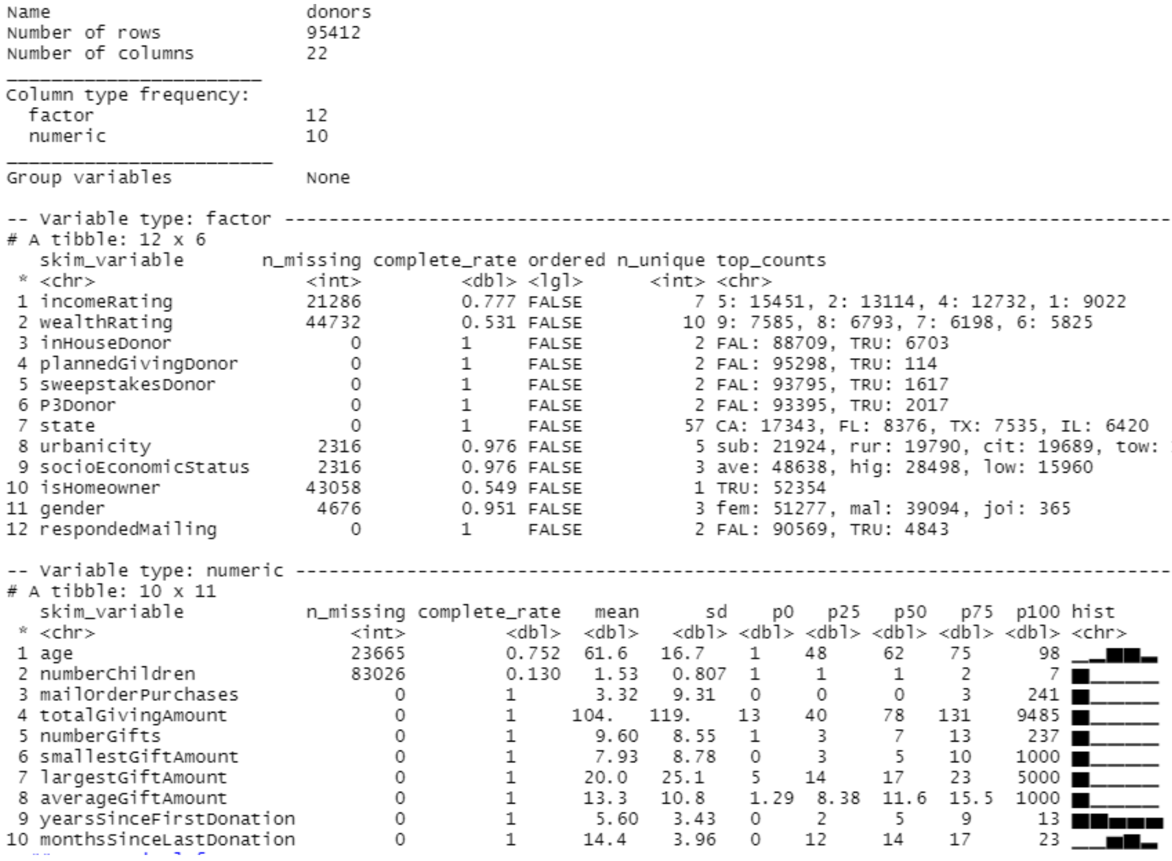
Key Question:
Can we build a model that informs this veterans organization how to effectively target future donors.

Data[1]:
Here is a quick snapshot of the entire dataset using the skim() function in R (part of the skimr package).

donors.csv

Lets spend a few minutes seeing what’s interesting about this dataset.

[1] Second International Knowledge Discovery and Data Mining Tools Competition - details at https://kdd.ics.uci.edu/databases/kddcup98/epsilon_mirror/cup98dic.txt



```
Name                       donors
Number of rows             95412
Number of columns          22
_______________________
Column type frequency:
  factor                   12
  numeric                  10
________________________
Group variables            None

-- Variable type: factor ---------------------------------------------------------------------------------
# A tibble: 12 x 6
   skim_variable       n_missing complete_rate ordered n_unique top_counts
 * <chr>                   <int>         <dbl> <lgl>      <int> <chr>
 1 incomeRating            21286         0.777 FALSE          7 5: 15451, 2: 13114, 4: 12732, 1: 9022
 2 wealthRating            44732         0.531 FALSE         10 9: 7585, 8: 6793, 7: 6198, 6: 5825
 3 inHouseDonor                0         1     FALSE          2 FAL: 88709, TRU: 6703
 4 plannedGivingDonor          0         1     FALSE          2 FAL: 95298, TRU: 114
 5 sweepstakesDonor            0         1     FALSE          2 FAL: 93795, TRU: 1617
 6 P3Donor                     0         1     FALSE          2 FAL: 93395, TRU: 2017
 7 state                       0         1     FALSE         57 CA: 17343, FL: 8376, TX: 7535, IL: 6420
 8 urbanicity               2316         0.976 FALSE          5 sub: 21924, rur: 19790, cit: 19689, tow: 19527
 9 socioEconomicStatus      2316         0.976 FALSE          3 ave: 48638, hig: 28498, low: 15960
10 isHomeowner             43058         0.549 FALSE          1 TRU: 52354
11 gender                   4676         0.951 FALSE          3 fem: 51277, mal: 39094, joi: 365
12 respondedMailing            0         1     FALSE          2 FAL: 90569, TRU: 4843

-- Variable type: numeric --------------------------------------------------------------------------------
# A tibble: 10 x 11
   skim_variable           n_missing complete_rate   mean      sd    p0   p25   p50   p75  p100 hist
 * <chr>                       <int>         <dbl>  <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
 1 age                         23665         0.752  61.6   16.7   1     48     62    75     98 ▁▁▃▇▇▂
 2 numberChildren              83026         0.130   1.53   0.807 1      1      1     2      7 ▇▁▁▁▁▁
 3 mailOrderPurchases              0         1       3.32   9.31  0      0      0     3    241 ▇▁▁▁▁▁
 4 totalGivingAmount               0         1     104.   119.   13     40     78   131   9485 ▇▁▁▁▁▁
 5 numberGifts                     0         1       9.60   8.55  1      3      7    13    237 ▇▁▁▁▁▁
 6 smallestGiftAmount              0         1       7.93   8.78  0      3      5    10   1000 ▇▁▁▁▁▁
 7 largestGiftAmount               0         1      20.0   25.1   5     14     17    23   5000 ▇▁▁▁▁▁
 8 averageGiftAmount               0         1      13.3   10.8   1.29  8.38  11.6  15.5  1000 ▇▁▁▁▁▁
 9 yearsSinceFirstDonation         0         1       5.60   3.43  0      2      5     9     13 ▇▇▅▅▅▅
10 monthsSinceLastDonation         0         1      14.4    3.96  0     12     14    17     23 ▁▁▃▇▇▂
```

Deck:

Dealing with highly skewed outcome class distributions using SMOTE (Synthetic Minority Over-sampling TEchnique)

R Code:

Lets try to use our template from Case 5 (targeting promotions at Universal bank) to fit a k-nearest neighbors model to this dataset. Note that we will have to deal with the missing values as well as the categorical variables such as State which have 57 levels (so far we only dealt with numerical data for k-NN, but this is not always the case in the real world).

See folder for R file

Discussion questions and observations:

1. We can use what's known as one-hot encoding to get R to create categorical variables into as many dummy variables as the number on levels in the categorical variable. If a person is from Texas than that dummy variable corresponding to TX will be = 1 and all 56 other dummy variables will be 0. Thus, all our predictors (X data) will be in a 0 to 1 scale (after we normalize the numeric variables) and we use k-NN.
2. Did SMOTE (Synthetic Minority Over-sampling TEchnique) help us learn better from the highly skewed (95% non donors -5% donors) dataset. Keep in mind that we only

oversample in the training dataset. The test data cannot be touched. Its what we will predict on, and as such is unseen by the model building process.

3. We will primarily use logistic regression, although I will also give you the code for k-NN
4. One aspect with lots of a categorical X variables is that due to randomization you may not have levels in your model built on the training set but they may exists in the test set. We will learn how to deal with this (delete those rows in test set).
5. 10 fold cross validation with 3 repetitions for k = 1 to 10 takes a long time on my laptop. Often in these situations I spin up an RStudio instance (see https://towardsdatascience.com/how-to-run-rstudio-on-aws-in-under-3-minutes-for-free-65f8d0b6ccda) with multiple cores (say 32) and many gigabytes of RAM on AWS and it takes a few seconds. Its costs only a few bucks and saves a lot of time.