Introduction

The objective of this report is to build two machine learning models for a charitable organization that wishes to improve their costs and donations associated with an upcoming direct marketing campaign to prior donors. A classification model will be built to determine likely donors so that expected profit is maximized. A regression model will be built to predict expected donation amounts.

This report consists of the following steps, with the above goal in mind:
1. Exploratory Data Analysis (EDA)
2. Data Preparation
3. Model Building
4. Model Selection

I perform EDA to know how to best prepare the data in the Data Preparation step. In the Model Building step, I go over all considered models built off data prepared in the Data Preparation step. All models are then compared in the Model Selection step to select the optimal models. The optimal models are then used to predict donor data that have not been used in building and selecting the optimal models.

Data is divided into three splits: a train split, validation split, and test split. Models are built off the train split, evaluated and compared on the validation split, and then the selected models are used to make predictions on the test split.

Classification models either output the posterior probability that DONR is 1 (that someone will donate) or the prediction itself (1 or 0). Regression models output the predicted donation amount in dollars.

Classification models are evaluated based on their "maximum profit" metric on the validation set, as discussed in the Model Evaluation section. Regression Models are evaluated based on their means squared error (mse) on the validation set, as discussed in the Model Evaluation section.

R is used for this analysis.

I examine the data to answer the following questions:

1. What does the data look like?
2. Are there predictors that have either wide ranges, skewed distributions, or outliers?
3. Will some variables likely be un-predictive in both the classification and regression problem?

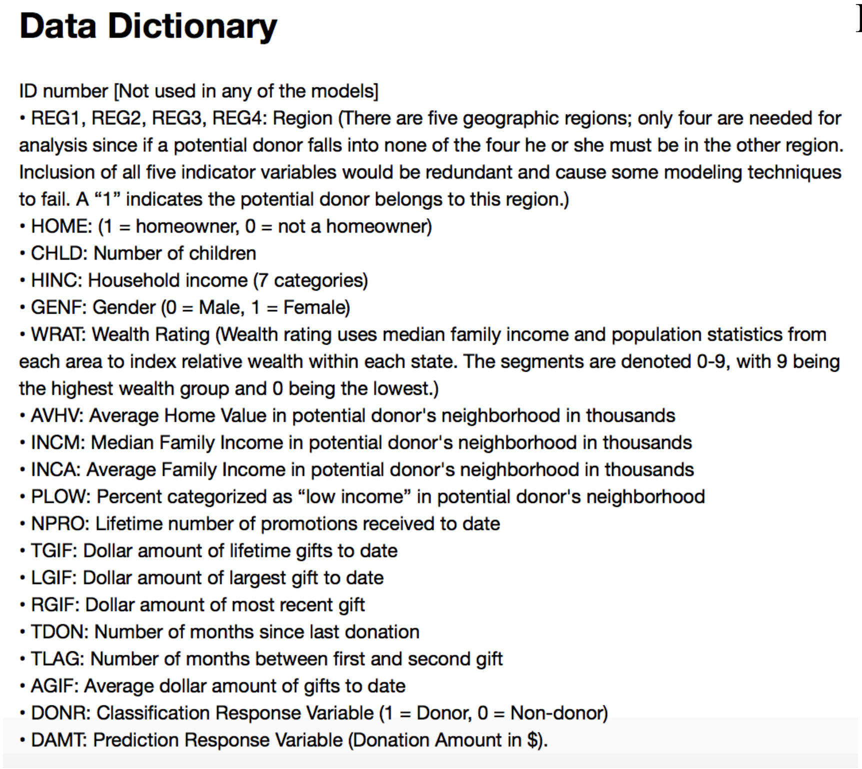
1. What does the data look like?

The dataset contains 8009 observations and 24 variables. 20 variables are considered in predicting the dependent variables, DONR and DAMT, for the classification and regression models, respectively. The ID variable contains no useful information and so is excluded from the analysis. The ‘part’ variable identifies which split the observation belongs to, either the train, validation, or test split.

8 of the predictors are categorical variables:
REG1, REG2, REG3, REG4, HOME, HINC, GENF, WRAT

12 of the predictors are quantitative variables:
CHLD, AVHV, INCM, INCA, PLOW, NPRO, TGIF, LGIF, RGIF, TDON, TLAG, AGIF

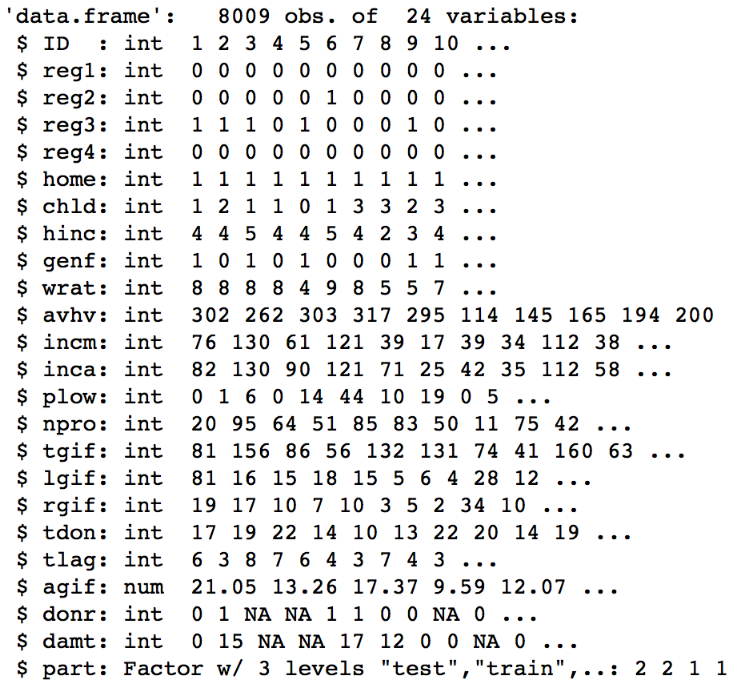
There are *no* missing values that need to be imputed.



Data Dictionary

ID number [Not used in any of the models]
- REG1, REG2, REG3, REG4: Region (There are five geographic regions; only four are needed for analysis since if a potential donor falls into none of the four he or she must be in the other region. Inclusion of all five indicator variables would be redundant and cause some modeling techniques to fail. A “1” indicates the potential donor belongs to this region.)
- HOME: (1 = homeowner, 0 = not a homeowner)
- CHLD: Number of children
- HINC: Household income (7 categories)
- GENF: Gender (0 = Male, 1 = Female)
- WRAT: Wealth Rating (Wealth rating uses median family income and population statistics from each area to index relative wealth within each state. The segments are denoted 0-9, with 9 being the highest wealth group and 0 being the lowest.)
- AVHV: Average Home Value in potential donor's neighborhood in thousands
- INCM: Median Family Income in potential donor's neighborhood in thousands
- INCA: Average Family Income in potential donor's neighborhood in thousands
- PLOW: Percent categorized as “low income” in potential donor's neighborhood
- NPRO: Lifetime number of promotions received to date
- TGIF: Dollar amount of lifetime gifts to date
- LGIF: Dollar amount of largest gift to date
- RGIF: Dollar amount of most recent gift
- TDON: Number of months since last donation
- TLAG: Number of months between first and second gift
- AGIF: Average dollar amount of gifts to date
- DONR: Classification Response Variable (1 = Donor, 0 = Non-donor)
- DAMT: Prediction Response Variable (Donation Amount in $).

Data Structure

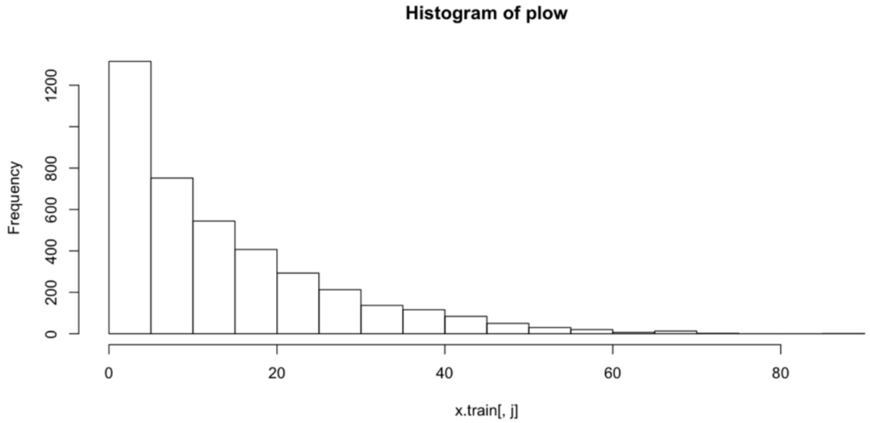


```
'data.frame':   8009 obs. of  24 variables:
 $ ID  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ reg1: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reg2: int  0 0 0 0 0 1 0 0 0 0 ...
 $ reg3: int  1 1 1 0 1 0 0 0 1 0 ...
 $ reg4: int  0 0 0 0 0 0 0 0 0 0 ...
 $ home: int  1 1 1 1 1 1 1 1 1 1 ...
 $ chld: int  1 2 1 1 0 1 3 3 2 3 ...
 $ hinc: int  4 4 5 4 4 5 4 2 3 4 ...
 $ genf: int  1 0 1 0 1 0 0 0 1 1 ...
 $ wrat: int  8 8 8 8 4 9 8 5 5 7 ...
 $ avhv: int  302 262 303 317 295 114 145 165 194 200
 $ incm: int  76 130 61 121 39 17 39 34 112 38 ...
 $ inca: int  82 130 90 121 71 25 42 35 112 58 ...
 $ plow: int  0 1 6 0 14 44 10 19 0 5 ...
 $ npro: int  20 95 64 51 85 83 50 11 75 42 ...
 $ tgif: int  81 156 86 56 132 131 74 41 160 63 ...
 $ lgif: int  81 16 15 18 15 5 6 4 28 12 ...
 $ rgif: int  19 17 10 7 10 3 5 2 34 10 ...
 $ tdon: int  17 19 22 14 10 13 22 20 14 19 ...
 $ tlag: int  6 3 8 7 6 4 3 7 4 3 ...
 $ agif: num  21.05 13.26 17.37 9.59 12.07 ...
 $ donr: int  0 1 NA NA 1 1 0 0 NA 0 ...
 $ damt: int  0 15 NA NA 17 12 0 0 NA 0 ...
 $ part: Factor w/ 3 levels "test","train",..: 2 2 1 1
```

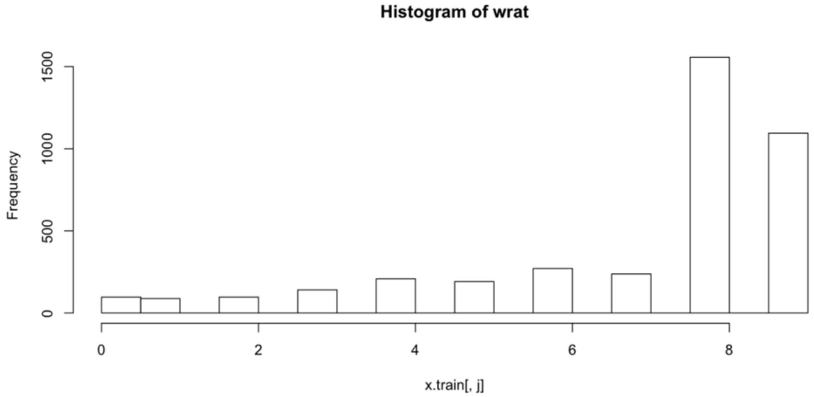
3. Are there quantitative predictors that have either wide ranges, skewed distributions, or outliers?

8 of the predictors have right skewed distributions in the train split, and so have outliers taking large, positive values: incm, inca, plow, tgif, lgif, rgif, tlag, agif. 1 predictor, wrat, has a left skewed distribution. Only 3 predictors have distributions that are approximately normal: wrat, tdon, and npro.

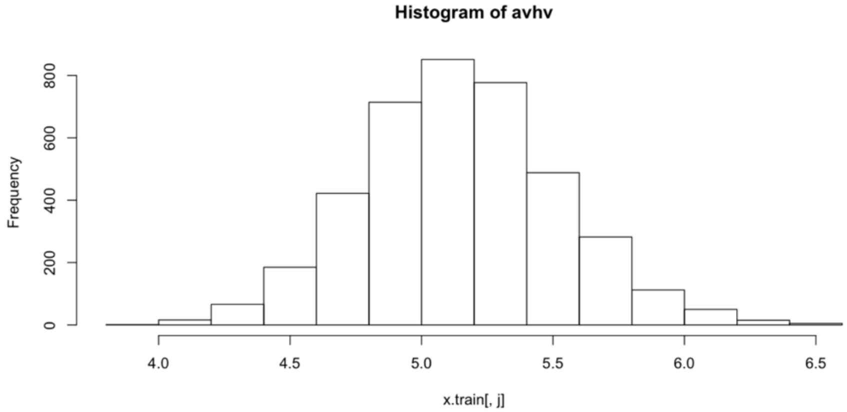
Most predictors, including plow, have a right skewed distribution.



Only the wrat predictor has a left skewed distribution.



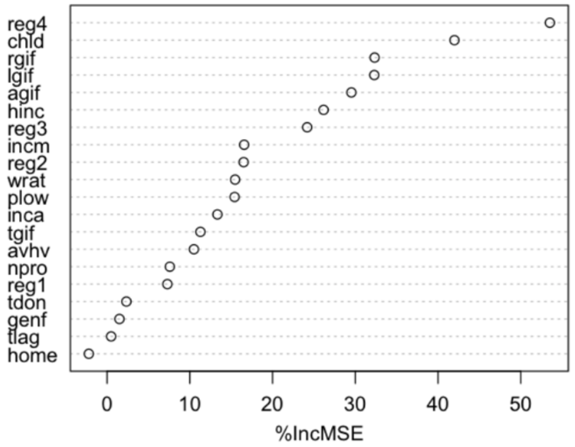
Only 3 variables, including avhv, have close to a normal distribution.

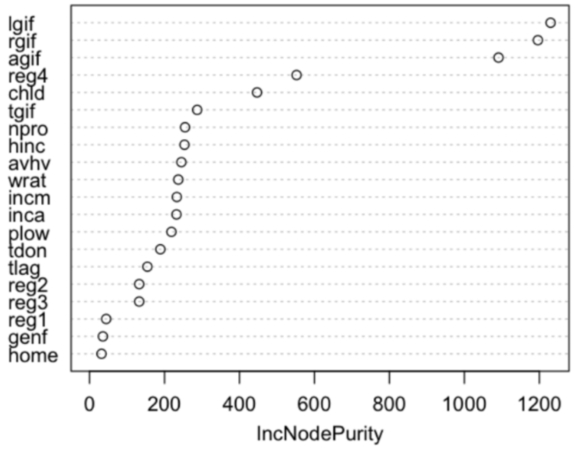


3. Will some variables likely be un-predictive in both the classification and regression problem?

I run a Random Forest Regressor in predicting damt from *standardized* train data (more on this later) to observe predictors that might be un-predictive of damt in the regression models. Two metrics are used to asses variable importances, %IncMSE and IncNodePurity. The higher the value, the more important the predictor. %IncMSE is based on the average increase in the mean squared error on out-of-bag observations when the predictor is not included in building the Random Forest. IncNodePurity estimates the average reduction in node impurity due to splitting on the predictor, across all the trees.

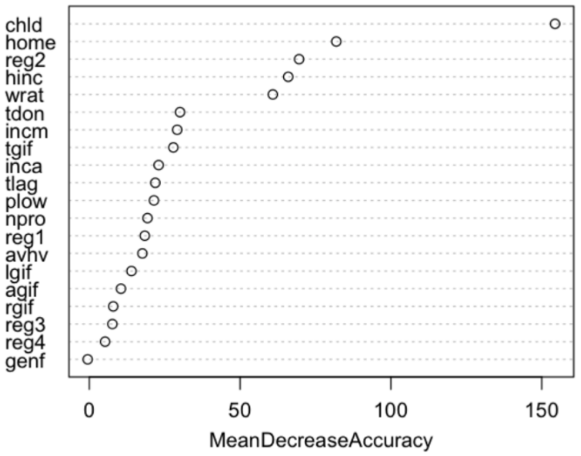
Observe that in both plots, home, genf, and reg1 are low on the totem pole of variable importances, so I consider excluding these variables for certain models.

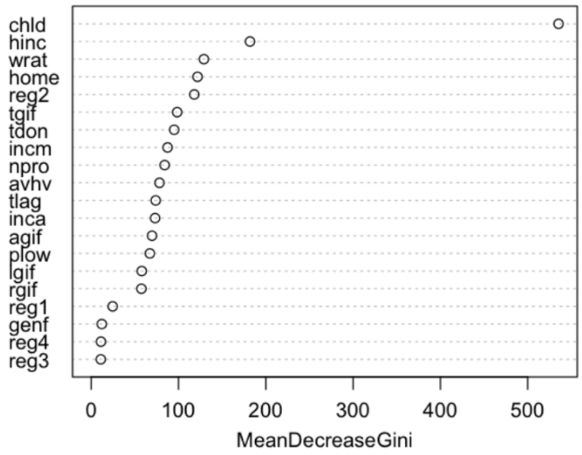




I run a Random Forest Classifier in predicting donr from standardized train data to observe predictors that might be un-predictive of donr in the classification models. Like the Random Forest Regressor's metrics, MeanDecreaseAccuracy and MeanDecreaseGini both measure the impact each variable has on the Random Forest. The higher the value, the better.

Observe that in both plots, genf, reg3, and reg4 have low variable importances, so I consider excluding these variables for certain models.





DATA PREPARATION

After exploring the data, I create four sets of predictor data.
Each set of predictor data is further broken up into train, validation, and test splits :

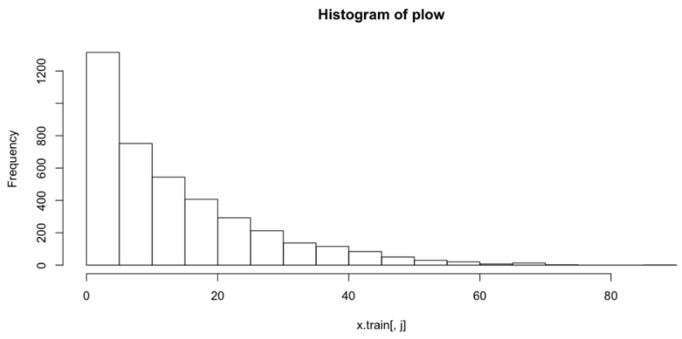
1. A set of standardized predictors to have mean zero and unit standard deviation.
2. A second set of standardized predictors where right-skewed predictors are log-transformed before having been standardized.
3. A third set of predictors for classification models, that are the same as the second set (directly above), but where predictors with low variable importances (reg3, reg4, genf) in the Random Forest Classifier are removed.
4. A fourth set of predictors for regression models, that are the same as the second set, but where predictors with low variable importances (home, genf, reg1) in the Random Forest Regressor are removed.

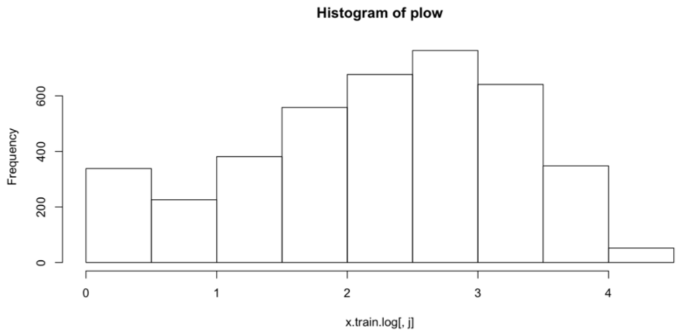
Dataset 1
Predictors are standardized to have mean zero and unit standard deviation: (x – mean)/sd. Predictors are thus placed on the same scale, and so the considered supervised learners will treat each predictor more equally. This is important for many of the considered methods, like Principal Component Regression, where predictors with relatively large values are likely to dominate the first several principal components.

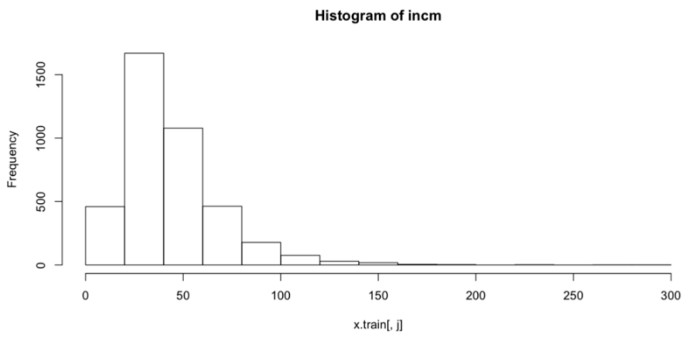
Dataset 2
For the second set, right-skewed predictors are log-transformed before being standardized. Observe the difference in distributions for the plow and incm predictors, which afterwards appear more normally distributed:

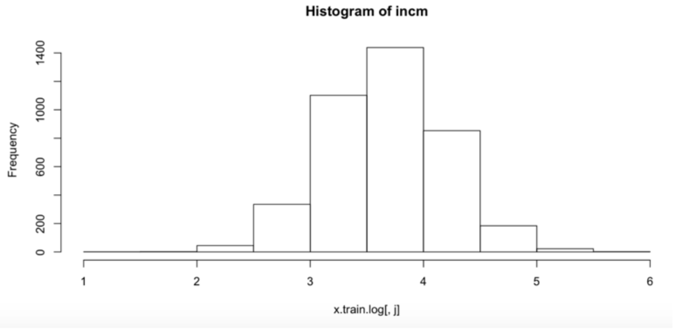
Before Log Transformation

After Log Transformation









MODEL BUILDING

Candidate Classification Models

Decision Trees, Boosting, Bagging, Random Forests, K-Nearest Neighbors, Support Vector Machines, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Logistic Regression

Candidate Regression Models

Least Squares Linear Regression, Best Subset Selection and Linear Regression
Ridge Regression, Lasso, Principal Components Regression, Neural Networks
Gradient Boosted Machines, XGBoost, Random Forests

Methodology

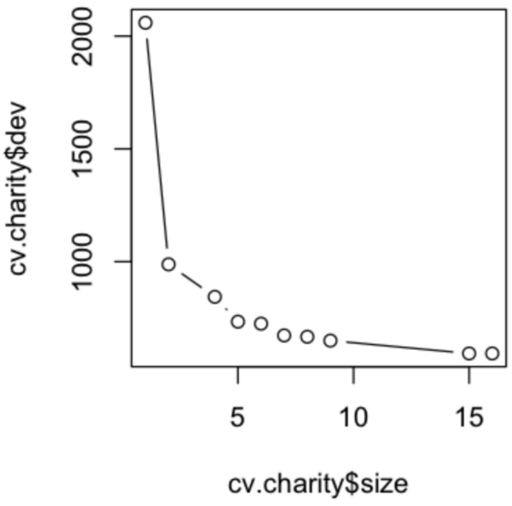
Each model is run off a train split of standardized predictors and a train split of standardized predictors with some having been log-transformed. The top 3 performing regression models and classifiers are also run off standardized, log-transformed predictors with some predictors having been removed (datasets 3 and 4, as discussed on the previous page). In the code, I've set the seed to 1 in R via set.seed(1) to allow for reproducible model results.

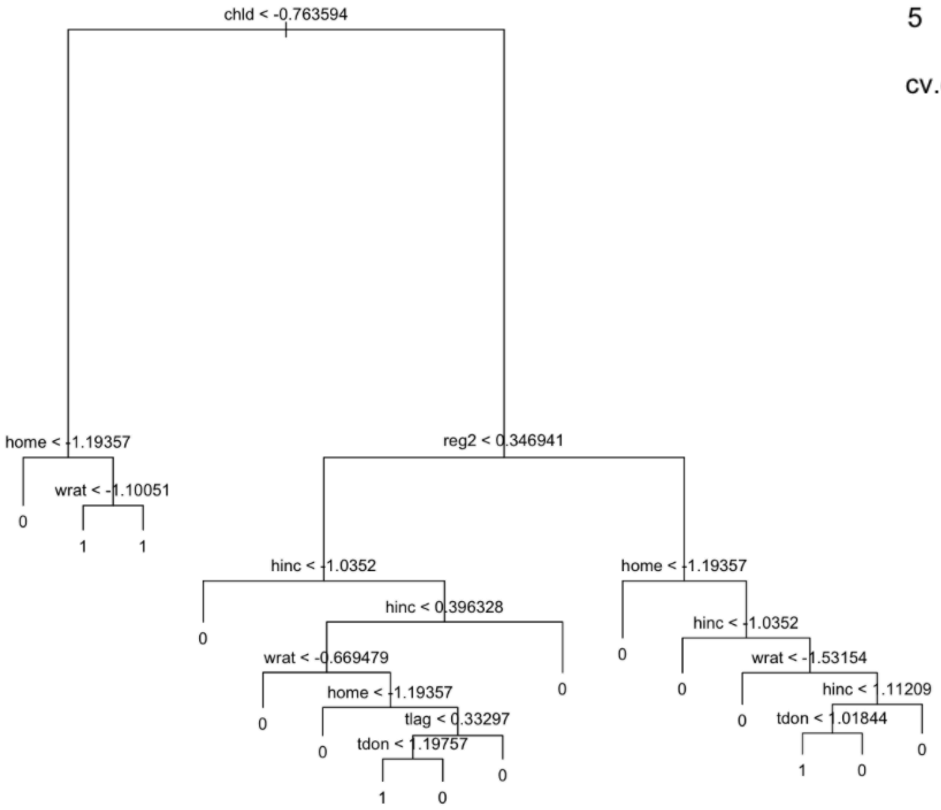
Classification

Decision Trees (DT)

The decision tree fit to the standardized predictor data produces the same results as the one fit to the log-transformed data. The chld variable (number of children) is selected as the root node (top-most node) and is thus considered as the best predictor. If the chld value is greater than .76 standard deviations (sd) *below* the mean, then the home variable's value is considered. If it's value is less than .76 sd *below* the mean, then the reg2 variable's value is considered, and so on.

I run 10-fold cross-validation to observe the number of misclassifications (cv error rate) from each of the 10 resultant models for each level of considered model complexity (k=1 to k=16 terminal nodes). The error rate is lowest when k=16, meaning when the model is the most complex. So there is no need to prune the tree.





Generalized Boosted Models (GBM)

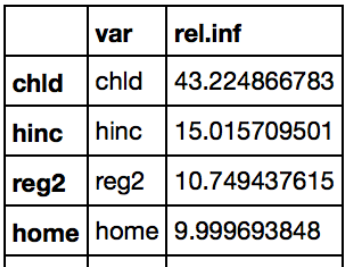
Generalized boosted classification trees are similar to generalized boosted regression trees. The latter is easier to explain, so I'll explain boosted regression trees. A generalized boosted regression tree starts out as just a regression tree that predicts y. Then, a regression tree that predicts the residuals of the model is created, and its residual predictions are added to each predicted y value, such that each yhat then equals

yhat1 = yhat + learning rate (predicted residual for yhat model)

Then another regression tree is created which models yhat1's residuals, and so on. This process is repeated for a predetermined number of boosting stages, meaning trees that improve the original tree.

All parameters are set to their default values except for the following:
Learning rate (shrinkage parameter): .001
Tree depth: 4
Number of trees: 5000
Distribution: Bernoulli (since this is a classification problem)

Log-transformed predictors do not change the results. I also run the model by removing reg3, reg4, genf, which improves accuracy, while causing the expected profit to drop by a point. Predictors with the highest relative influence measures are shown to the right, meaning predictors having the greatest impact on the model. Like the Decision Tree, chld, home, and reg2 are very influential.



	var	rel.inf
chld	chld	43.224866783
hinc	hinc	15.015709501
reg2	reg2	10.749437615
home	home	9.999693848

Random Forests (RF)

This model constructs a multitude of classification trees. Each tree is different because of having been created from two randomization processes: tree bagging and feature bagging. Each tree spits out a prediction and the final prediction is based on the predictions from all of the trees.

I produce a Random Forest (RF) using the default values in the RandomForest package, including 500 trees, and in the feature bagging step, only 4 predictors considered at each split. Using log-transformed predictors does change the result, but makes accuracy and profit worse on the validation set.

Bagging

Bagging is like Random Forests, but lacks the feature bagging step, so all features are considered at each split. Thus, at every split, I consider 20 features. 500 trees are constructed.

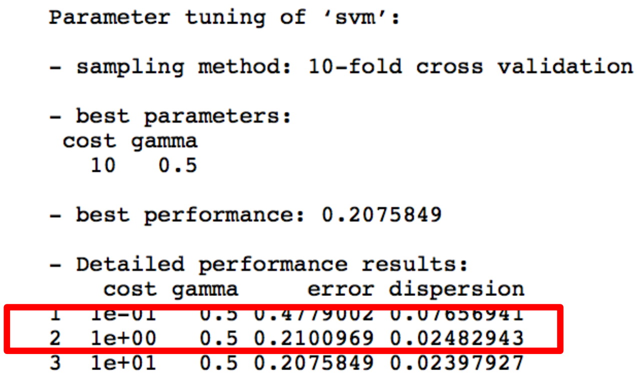
K-Nearest Neighbors (KNN)

The KNN Classifier considers the k closest points (in Euclidean distance) to the observation whose class we'd like to predict and assigns its class as the one that is the most frequent among the k closest points. I keep all other parameters at their default values. The predicted class in a tie is decided at random. I build 4 KNN models for k=3,4,5,6. k=5 produces the highest accuracy on the validation split, so I select k=5. Using log-transformed predictors does change the result, but slightly lowers profit.

Support Vector Machines (SVM)

SVMs fit non-linear decision boundaries between classes using non-linear kernels, and thus offer greater flexibility than the Maximum Margin Classifier and Support Vector Classifier. I use the radial kernel.

I tune the cost and gamma parameters over the following values by using grid search, which does 10-fold cross validation: (cost: .1, 1,10,100,1000), (gamma: .5,1,2,3,4). The optimal parameters, meaning the ones with the lowest cv error, are (cost: 10), (gamma:.5). The profit is awful for this model, unfortunately, so I do not bother observing the results for log transformed predictors. See the tuned parameter output below:



```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
   10   0.5

- best performance: 0.2075849

- Detailed performance results:
   cost gamma     error dispersion
1  1e-01   0.5 0.4779002 0.07656941
2  1e+00   0.5 0.2100969 0.02482943
3  1e+01   0.5 0.2075849 0.02397927
```

Linear Discriminant Analysis (LDA)

The LDA Classifier makes use of Bayes Theorem in that it assigns posterior probabilities based on the data, and then selects the class with the maximum likelihood. It assumes each observations for each class have a Gaussian distribution and the same variances, but classes have differing means. I run the model on all predictors and make it slightly more flexible by including a squared hinc term, which does not negatively impact the results. The model results improve after running it off log transformed predictors, and improve even more after removing reg3, reg4 and genf.

Quadratic Discriminant Analysis (QDA)

QDA is like LDA, but offers greater flexibility by assuming that classes have differing covariance matrices, and so has greater variance, but lower bias. I run the model on log-transformed predictors as well, though this reduces estimated maximum profit.

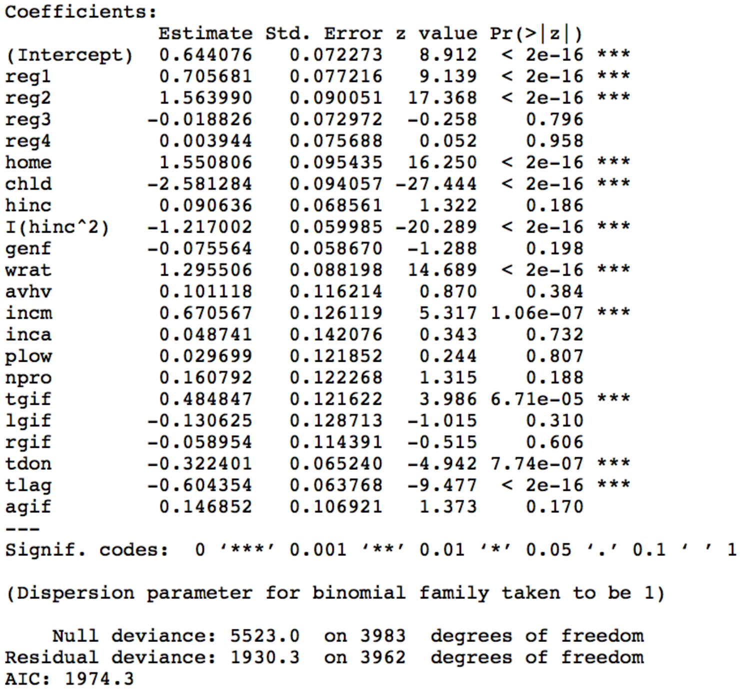
Logistic Regression

This method estimates the log odds of donr as a linear combination of the predictors:

ln (p/(1-p)) = linear predictor

I run the logistic model first off all predictors, then with log transformed predictors, then with reg3, reg4, and genf removed. I increase the model's flexibility by including the square of hinc as a predictor. Using log transformed predictors with no variables removed leads to the best profit. Observe below that many of the predictors are significant at the .001 level (as indicated by having three asterisks).

When the home variable increases by 1 standard deviation (remember, it has been standardized) , the odds of donating is expected to increase by exp(1.551)-1=372%, holding all other variables constant.



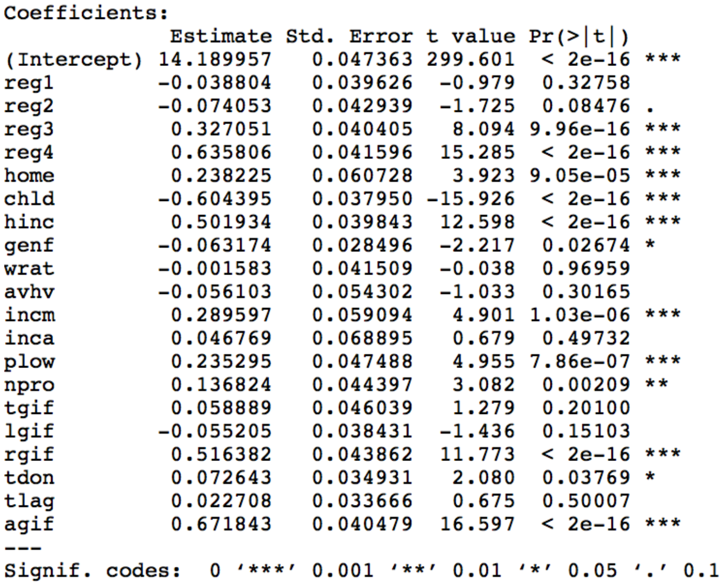
```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  0.644076   0.072273   8.912  < 2e-16 ***
reg1         0.705681   0.077216   9.139  < 2e-16 ***
reg2         1.563990   0.090051  17.368  < 2e-16 ***
reg3        -0.018826   0.072972  -0.258    0.796    
reg4         0.003944   0.075688   0.052    0.958    
home         1.550806   0.095435  16.250  < 2e-16 ***
chld        -2.581284   0.094057 -27.444  < 2e-16 ***
hinc         0.090636   0.068561   1.322    0.186    
I(hinc^2)   -1.217002   0.059985 -20.289  < 2e-16 ***
genf        -0.075564   0.058670  -1.288    0.198    
wrat         1.295506   0.088198  14.689  < 2e-16 ***
avhv         0.101118   0.116214   0.870    0.384    
incm         0.670567   0.126119   5.317 1.06e-07 ***
inca         0.048741   0.142076   0.343    0.732    
plow         0.029699   0.121852   0.244    0.807    
npro         0.160792   0.122268   1.315    0.188    
tgif         0.484847   0.121622   3.986 6.71e-05 ***
lgif        -0.130625   0.128713  -1.015    0.310    
rgif        -0.058954   0.114391  -0.515    0.606    
tdon        -0.322401   0.065240  -4.942 7.74e-07 ***
tlag        -0.604354   0.063768  -9.477  < 2e-16 ***
agif         0.146852   0.106921   1.373    0.170    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5523.0  on 3983  degrees of freedom
Residual deviance: 1930.3  on 3962  degrees of freedom
AIC: 1974.3
```

Regression

Least Squares Linear Regression

Results of running a regression on all predictors, with log transformed right skewed predictors include: an adjusted R-squared of 56.79%, meaning 56.79% of the variation in damt is explained by the predictors, and many of the predictors being significant at the .001 significance level.

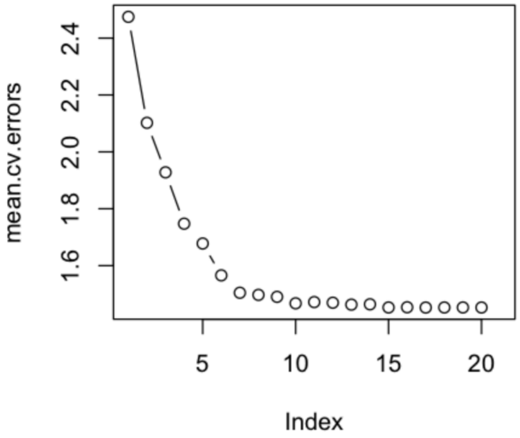
In interpreting coefficients, for example, for a one standard deviation increase in chld, we expect damt to decrease by -.6, holding all other predictors constant.



```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 14.189957   0.047363 299.601  < 2e-16 ***
reg1        -0.038804   0.039626  -0.979  0.32758    
reg2        -0.074053   0.042939  -1.725  0.08476 .  
reg3         0.327051   0.040405   8.094 9.96e-16 ***
reg4         0.635806   0.041596  15.285  < 2e-16 ***
home         0.238225   0.060728   3.923 9.05e-05 ***
chld        -0.604395   0.037950 -15.926  < 2e-16 ***
hinc         0.501934   0.039843  12.598  < 2e-16 ***
genf        -0.063174   0.028496  -2.217  0.02674 *  
wrat        -0.001583   0.041509  -0.038  0.96959    
avhv        -0.056103   0.054302  -1.033  0.30165    
incm         0.289597   0.059094   4.901 1.03e-06 ***
inca         0.046769   0.068895   0.679  0.49732    
plow         0.235295   0.047488   4.955 7.86e-07 ***
npro         0.136824   0.044397   3.082  0.00209 ** 
tgif         0.058889   0.046039   1.279  0.20100    
lgif        -0.055205   0.038431  -1.436  0.15103    
rgif         0.516382   0.043862  11.773  < 2e-16 ***
tdon         0.072643   0.034931   2.080  0.03769 *  
tlag         0.022708   0.033666   0.675  0.50007    
agif         0.671843   0.040479  16.597  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
```

Best Subset Selection and Linear Regression

5-Fold Cross Validation is used with best subset selection to determine a subset of variables to use for linear regression. For each of the 5 folds, best subset selection is performed and the best 1-predictor through 20-predictor models are produced. Then each of the 5 cv errors (cv mse's) associated with each j-predictor model are averaged and the j-predictor model that has the lowest mean cv score is selected.

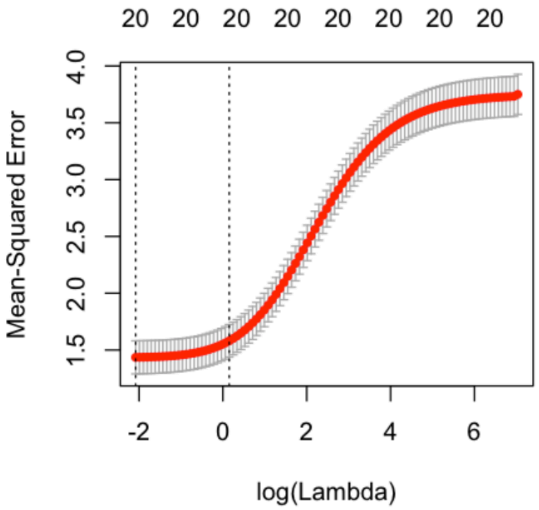


Observe how the mean cv score *decreases* as the number of predictors in the model *increases*. The number of predictors that minimize the mean cv score in this plot (for models built of log transformed right skewed predictors) is 17, and so I run a linear regression with these 17 predictors off the log-transformed data.

The adjusted R-squared is 62.1%, meaning that 62.1% of the variation in damt is explained by the selected predictors. Many predictors are statistically significant at the .001, like home, which has a coefficient of .242. This means that for a one standard deviation increase in home, we expect the donation amount to increase by .242, holding all other variables constant.

Ridge Regression

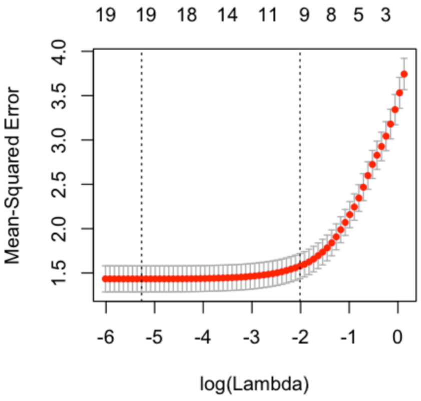
This method is a shrinkage method that attempts to reduce the complexity (variance) of a least squares solution by more than the corresponding increase in bias, which thus improves predictions. It does this by shrinking coefficients towards zero by increasing a lambda parameter, which controls the level of shrinkage. Lower values mean less shrinkage and a solution that is closer to least squares regression. I tune lambda via cross validation and observe the mean squared errors to select the optimal lambda, which turns out to be .125. This means that not much shrinkage is happening and the solution is close to the least squares solution.



The plot to the right shows that the 10-fold cross validation mse is minimized for lower values of lambda. This plot is for building a model off log transformed right skewed predictors.

Lasso

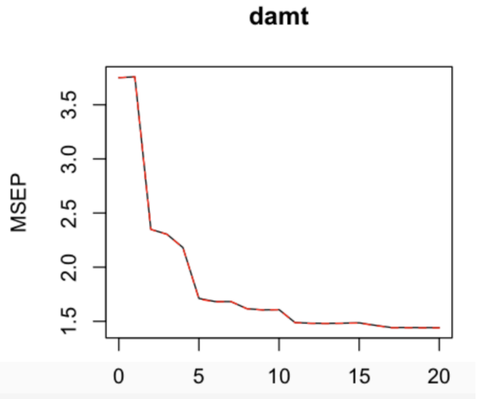
This method is a shrinkage method that is similar to ridge regression, but can do variable selection by shrinking coefficients to zero. I run 10-fold cross validation to tune lambda, and the optimal lambda is .134 for building a model off log transformed right skewed predictors.



Principal Components Regression (PCR)

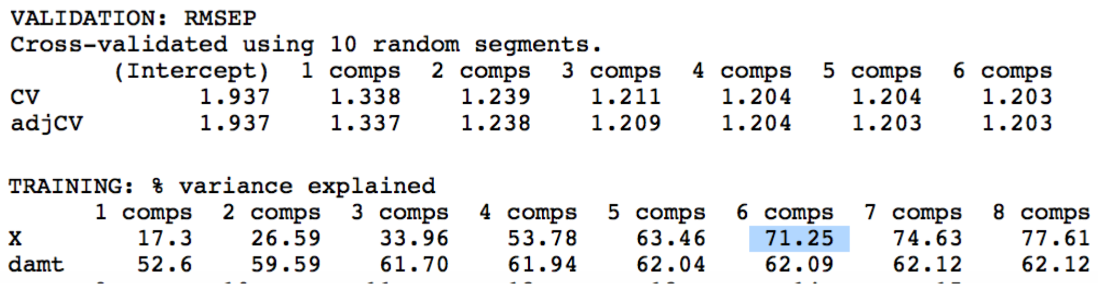
This method reduces the dimensionality of the dataset by creating new predictors called "principal components" that represent the original predictors.

In choosing how many principal components to use, I perform cross validation and look at mean squared errors (of regressing damt on the principal components) for differing numbers of components. I select 17 components for the log transformed right skewed predictor data, which produces the lowest cv score, even though 17 components explains 97.8% of the variance in the predictors. Observe how the cv mse score decreases as the number of principal components increases (for log transformed right skewed variable data).



Partial Least Squares Regression (PLSR)

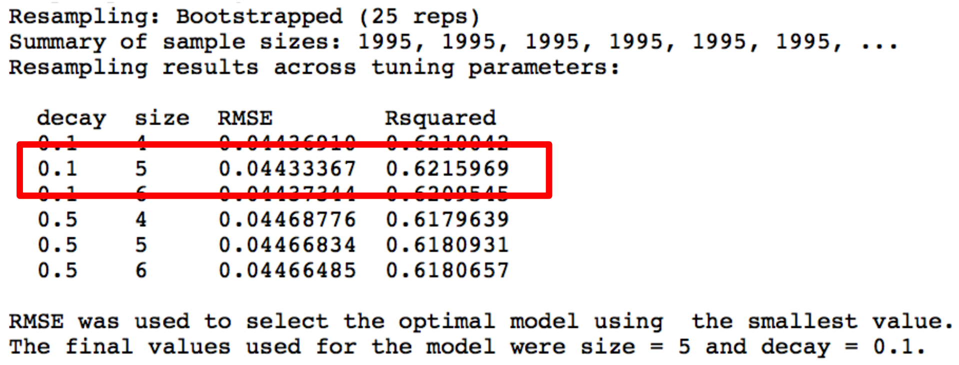
PLSR is similar to PCR, except new features are constructed in a way that not only represents the original features, but also explain the dependent variable. I use cross validation and observe that 6 components produce a low cv score while accounting for 71.25% of the variance in the data. Below are shown CV scores and % variance explained when training the model off log-transformed right skewed predictors:



```
VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           1.937    1.338    1.239    1.211    1.204    1.204    1.203
adjCV        1.937    1.337    1.238    1.209    1.204    1.203    1.203

TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
X        17.3    26.59    33.96    53.78    63.46    71.25    74.63    77.61
damt     52.6    59.59    61.70    61.94    62.04    62.09    62.12    62.12
```

Neural Networks (NN)

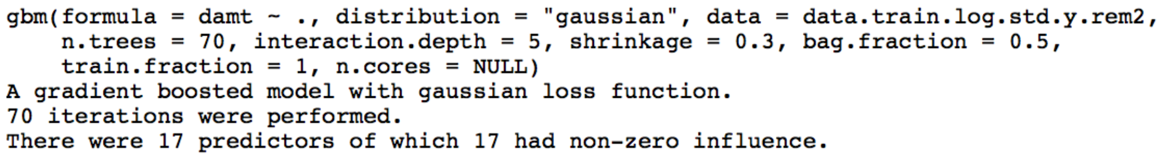
I run a Neural Network with a single hidden layer and tune the size and decay hyper-parameters. Output of the resampling optimization process is shown below for running the model off log-transformed predictors, which produces a lower mse than just using standardized predictors. Size=5 and decay = .1 generate the lowest rmse metric. I use these in the final model.



```
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1995, 1995, 1995, 1995, 1995, 1995, ...
Resampling results across tuning parameters:

  decay  size  RMSE        Rsquared
  0.1    4     0.04436910  0.6210042
  0.1    5     0.04433367  0.6215969
  0.1    6     0.04437344  0.6209545
  0.5    4     0.04468776  0.6179639
  0.5    5     0.04466834  0.6180931
  0.5    6     0.04466485  0.6180657

RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were size = 5 and decay = 0.1.
```

Generalized Boosted Models (GBM)

Unlike in the classification setting, I use differing parameters. The distribution is set to 'gaussian' since we're no longer dealing with a classification problem. I use less trees, which does not seem to impact performance and speeds up the algorithm. Running the model on log-transformed right skewed predictors, but with home, genf, and reg1 removed, yields the lowest mse on the validation split. Model parameters are shown below:



```
gbm(formula = damt ~ ., distribution = "gaussian", data = data.train.log.std.y.rem2,
    n.trees = 70, interaction.depth = 5, shrinkage = 0.3, bag.fraction = 0.5,
    train.fraction = 1, n.cores = NULL)
A gradient boosted model with gaussian loss function.
70 iterations were performed.
There were 17 predictors of which 17 had non-zero influence.
```

XGBoost

I implement a new, popular method called XGBoost, which is known for winning Kaggle competitions. The method was designed to enhance GBMs through regularization. The method performs best on all predictors, but with log-transformed right skewed predictors. I do not do any hyperparameter tuning and implement it with 70 boosting rounds.

Random Forests (RF)

The Random Forest Regressor, consisting of 500 trees, produces a lower mse on log-transformed right-skewed predictors than on just the original standardized predictors set.

Notes

I only run the top 3 performing classification models on data that excludes the 3 unimportant variables as identified by the Random Forests model.

Classification Model Selection

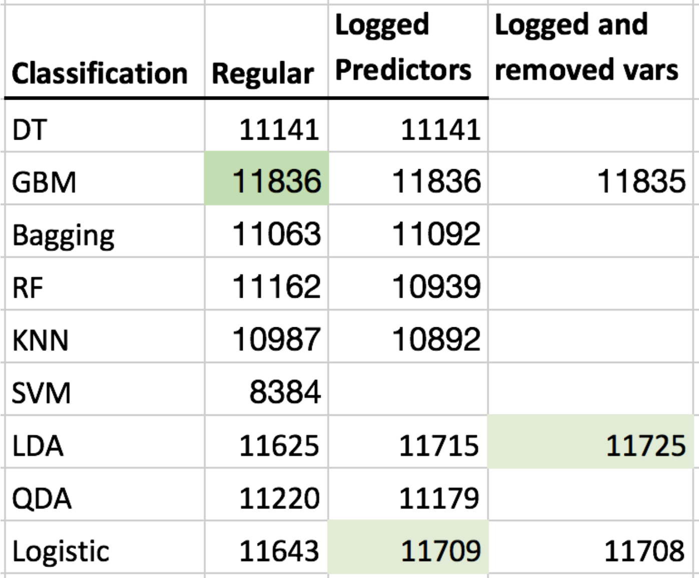
I calculate 'maximum profit' for each classification model applied to the validation set and select the model with the highest maximum profit. Maximum profit is calculated by rank ordering observations by predicted probability (from highest to lowest), and finding the subset of these observations whose cumulative sum of (14.5*DONR-2) leads to maximum profit). 14.5 is the average donation per mailing and 2 is the cost per mailing.

For models that only return predictions and not posterior probabilities, I calculate a proxy for maximum profit as 14.5*(number of true positives) – 2(true positives + false positives) = 14.5*(mailings that donate) – 2(mailings that donate + mailings that don't donate), as constructed from the confusion matrix. These models include: DT, Bagging, RF, KNN, SVM

Observe the maximum profits in the table on the right. Among the classification models, the top 3 models based on their expected profit are:

1. **Generalized Boosted Models (GBM)** run on standardized predictors produced a maximum profit of $11836.
2. **Linear Discriminant Analysis (LDA)** run on standardized predictors, with right-skewed predictors having been first log-transformed, and with reg3, reg4, genf having been removed. Max profit is $11725
3. **Logistic Regression** run on standardized predictors, with right-skewed predictors having been first log-transformed. Max profit is $11709.

I therefore select the GBM model run on standardized predictors.



Classification	Regular	Logged Predictors	Logged and removed vars
DT	11141	11141	
GBM	11836	11836	11835
Bagging	11063	11092	
RF	11162	10939	
KNN	10987	10892	
SVM	8384		
LDA	11625	11715	11725
QDA	11220	11179	
Logistic	11643	11709	11708

MODEL SELECTION

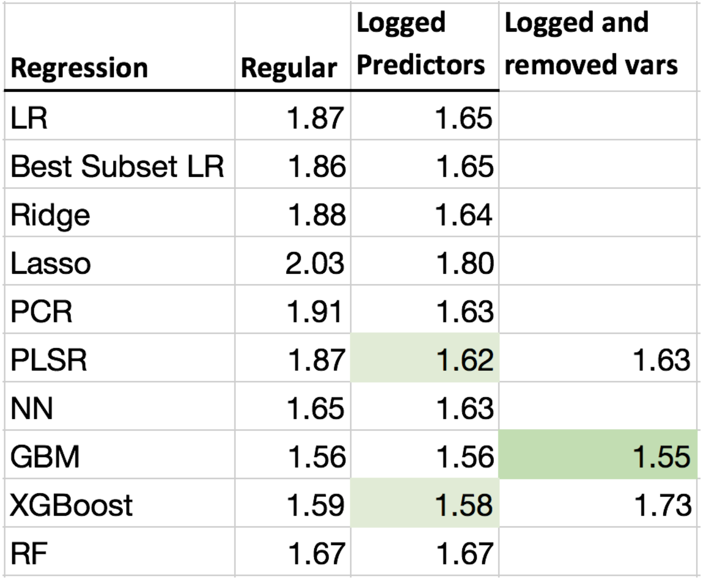
Regression Model Selection

I calculate mean squared error (mse) for each regression model applied to the validation set and select the model with the lowest mse.

Observe the mse metrics in the table on the right. The top 3 models include:

1. **Generalized Boosted Models (GBM)** on standardized predictors with right skewed predictors having first been log-transformed, and with home, genf, reg1 having been removed. MSE = 1.55
2. **XGBoost** run on standardized predictors with right-skewed predictors having first been log-transformed. MSE = 1.57
3. **Partial Least Squares Regression** run on standardized and right-skewed predictors having first been log-transformed. MSE = 1.62

I therefore select the GBM model run on standardized predictors, with right-skewed predictors having been log-transformed, and home, genf, reg1 having been removed.



Regression	Regular	Logged Predictors	Logged and removed vars
LR	1.87	1.65	
Best Subset LR	1.86	1.65	
Ridge	1.88	1.64	
Lasso	2.03	1.80	
PCR	1.91	1.63	
PLSR	1.87	1.62	1.63
NN	1.65	1.63	
GBM	1.56	1.56	1.55
XGBoost	1.59	1.58	1.73
RF	1.67	1.67	

CONCLUSION

Boosted classification trees and boosted regression trees, also called Generalized Boosted Models or Gradient Boosted Machines (GBM) or simply Boosting, appear to have generated the best metrics for both the classification and regression problems!

The selected classification method, GBM run on standardized predictors, maximizes expected profit on the validation set, so this method will be applied to the test set.

The selected regression method, GBM run on standardized predictors with right skewed predictors having first been log-transformed, and with home, genf, reg1 having been removed, will also be applied to the test set. Note that the test set must also have the same predictors log-transformed and home, genf, and reg1 removed.

In applying the selected methods to the test set: an adjustment has been applied to the classification model's results. This adjustment accounts for "over-sampling", since the validation response rate is 5 times greater than the test set's response rate. Observe the code for more details.