Chris Pelkey
PREDICT 422-55
Final Exam

## Introduction

For the course of this paper, we will look to develop a machine learning model to improve the cost-effectiveness of a direct marketing campaign to donors. Taken directly from the description of the program we learn that the average donation is $14.50. We further know that each direct mailing costs $2.00 to produce and send. With an average response rate of around 10%, the campaign is not cost-efficient to send to all donors; in fact it will actually lose the company money to send it out like this.

The course of this paper will outline several machine learning techniques that will help to determine a model to drive profits up. In addition, we will look to estimate how much donors will donate when placed in the "will donate" classification. Finally, we will look to see how much profit will be gleaned from taking this approach.

## Exploratory Data Analysis

A cursory glance at the data gives us a few easy to spot points of information. The entire dataset consists of 3984 training observations with an additional 2018 validation observations, and 2007 test observations. We further know that the data provided has oversampled a positive response variable with the oversampling resulting in a 50% response rate when the campaign typically yields a 10% response rate normally. The data contains 21 indicator variables and 2 response variables, DONR and DAMT. An explanation of the variables and what they represent can be found below in Table 1.

| VAR Name | Explanation |
|---|---|
| ID number | ID number, solely used to identify, not for modeling |
| 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 |

| VAR Name | Explanation |
|---|---|
| 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 $). |

**Table 1 - An explanation of variables found in the Charity dataset**

Turning to the exploratory data analysis portion of the exercise, it appears that the only variables with missing values are the observations in the test data set, which are missing DONR and DAMT. Charts 1 and 2 show the variable importance for both DONR and DAMT respectively. You can see that CHLD, COME, HINC and WRAT all play a role in lowering the MSE for DONR. Whereas for DAMT CHLD, RGIF, LGIF, AGIF and HINC play a fairly large role in reducing MSE..

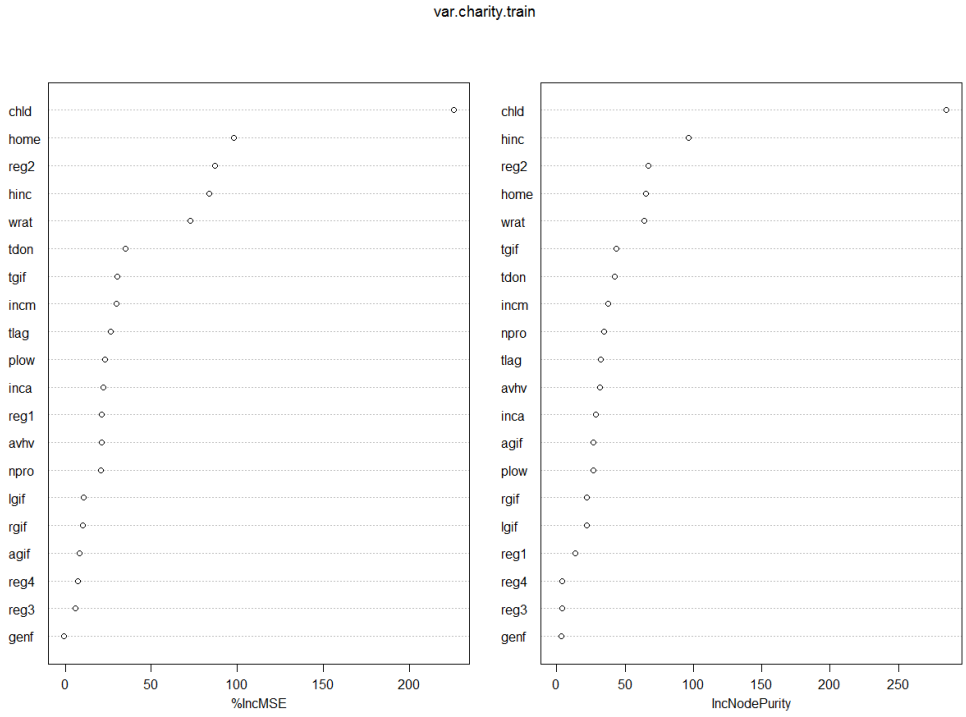


**Chart 1 – Chart showing variable importance for DONR**

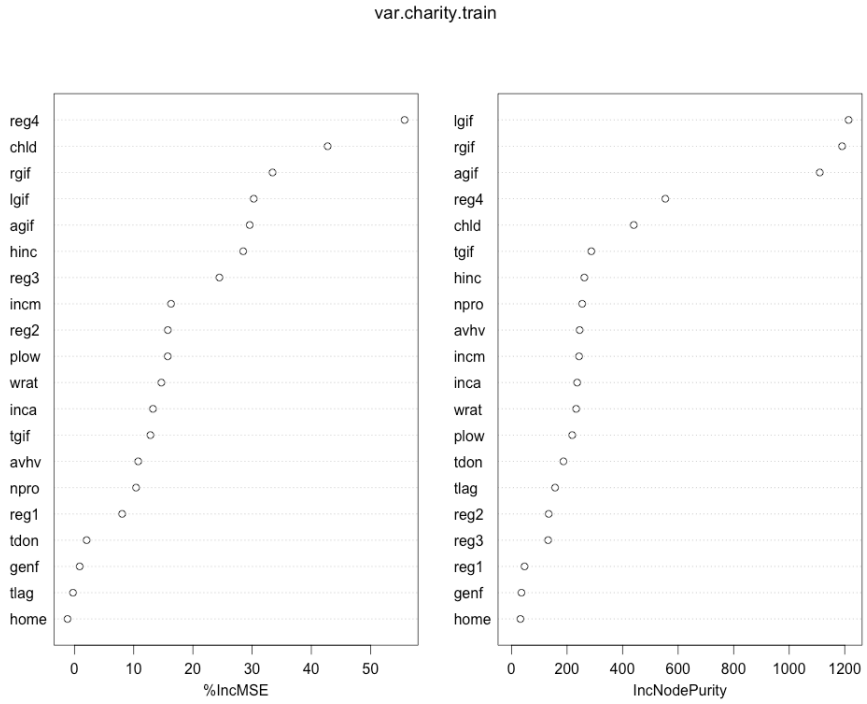


**Chart 2 – Chart showing variable importance for DONR**

## DONR Modeling

### LDA model

The first model tested is the linear discriminant analysis model. While this model isn't typically used with qualitative variables in modeling, it does enable us to get a base line to measure the other models and sometimes it is the model not thought to be used that has the best outcomes!

This was not the case for the linear discriminant analysis however. The model correctly guessed 1,660 of the validation data. The model guessed that a total of **1,329 would respond to the mailer**, but 344 of those were incorrectly identified. Only 14 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.lda1 | 0 | 1 |
| 0 | 675 | 14 |
| 1 | 344 | 985 |

check n.mail.valid = 344+985 = 1329
check profit = 14.5*985-2*1329 = 11624.5

The model returned a profit of **$11,624.50**.

## Logistic regression model

The second model tested is the logistic regression model. This model is the standard for predicting binary outcomes like whether someone will donate or they won't donate. It uses selected variables as input and then adds those up to determine if a candidate will donate. Typically it rounds up from .5 and down for below that.

The model correctly guessed 1,690 of the validation data. The model guessed that a total of **1,291 would respond to the mailer**, but 310 of those were incorrectly identified. 18 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.log1 | 0 | 1 |
| 0 | 709 | 18 |
| 1 | 310 | 981 |

check n.mail.valid = 310+981 = 1291
check profit = 14.5*981-2*1291 = 11642.5

The model returned a profit of **$11,642.50**. This is a slight improvement over the previous model.

## Bagging models

The third and fourth models run involved the process of bagging to determine which may be an optimal mailing campaign. The two different models used different amounts of variables at each split. The first model used 13 variables to consider at each split and the second model considered 6 variables at each split.

The model(13) correctly guessed 1,719 of the validation data. The model guessed that a total of **1,270 would respond to the mailer**, but 285 of those were incorrectly identified. 14 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.bag1 | 0 | 1 |
| 0 | 734 | 14 |
| 1 | 285 | 985 |

check n.mail.valid = 285+985 = 1270
check profit = 14.5*985-2*1270 = 11742.5

The model(13) returned a profit of **$11,742.50**. This is a slight improvement over the previous models.

The model(6) correctly guessed 1,756 of the validation data. The model guessed that a total of **1,265 would respond to the mailer**, but 278 of those were incorrectly identified. 23 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.bag2 | 0 | 1 |
| 0 | 769 | 23 |
| 1 | 278 | 987 |

check n.mail.valid = 278 + 987 = 1265
check profit = 14.5*976-2*1267 = 11781.5

The model(6) returned a profit of **$11,781.50**. This is a slight improvement over the previous models.

## Boosting models

The fifth and sixth models were generated using a boosting model. The difference between these models is the interaction depth. The first model allows for four splits per tree node and the second model allows for seven. Each of the models has 5,000 trees.

The model(4) correctly guessed 1,756 of the validation data. The model guessed that a total of **1,284 would respond to the mailer**, but 278 of those were incorrectly identified. 23 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.boost1 | 0 | 1 |
| 0 | 769 | 23 |
| 1 | 278 | 987 |

check n.mail.valid = 278 + 987 = 1284
check profit = 14.5*987-2*1284 = 11845

The model(4) returned a profit of **$11,845**. This is a slight improvement over the previous models.

The model(7) correctly guessed 1,757 of the validation data. The model guessed that a total of **1,242 would respond to the mailer**, but 252 of those were incorrectly identified. 9 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.boost | 0 | 1 |
| 0 | 767 | 9 |
| 1 | 252 | 990 |

check n.mail.valid = 252 + 990 = 1242
check profit = 14.5*990-2*1242 = 11871

The model(7) returned a profit of **$11,871**. This is a slight improvement over the previous models.

### Support vector machines

The seventh model was generated using a support vector machines. The model was run through with varying costs and gamma levels using cross-validation and the best model of all was selected to help predict the validation set.

The model(4) correctly guessed 1,684 of the validation data. The model guessed that a total of **1,023 would respond to the mailer**, but 179 of those were incorrectly identified. 155 of the desired donors in the validation set were missed.

| | c.valid | |
|---|---|---|
| chat.valid.svm | 0 | 1 |
| 0 | 840 | 155 |
| 1 | 179 | 844 |

check n.mail.valid = 179 + 844 = 1023
check profit = 14.5*844-2*1023 = 10192

The model returned a profit of **$10,192**. This was the worst model run.

### Overall DONR model numbers

The models run to predict the best mailing campaign to drive fundraising are below in Table 2. As was outlined in the paper above, the second boost model was the model that performed best, using 1,242 mailers and returning a profit of $11,871 on the validation data.

| n.mail | Profit | Model |
|---|---|---|
| **1329** | 11624.5 | LDA1 |
| **1291** | 11642.5 | Log1 |
| **1270** | 11742.5 | Bag1 |
| **1267** | 11748.5 | Bag2 |
| **1940** | 10576.5 | Tree |
| **1284** | 11845 | Boost |
| ***1242*** | *11871* | *Boost2* |
| **1023** | 10192 | Support Vector Machine |

Table 2 – Models run for DONR classification

Chris Pelkey
PREDICT 422-55
Final Exam

## DAMT Modeling

### Linear regression model

To begin a simple linear regression model with all variables was run to start out a baseline. With all 20 models included, there will be some background noise that will make the model a little less clear and a little harder to predict.

This model resulted in an MSE of 1.8675 with a standard error of 0.1697. This model was not overly problematic, but still had noise.

```
mean((y.valid - pred.valid.ls1)^2) # mean prediction error
# 1.867523
sd((y.valid - pred.valid.ls1)^2)/sqrt(n.valid.y) # std error
# 0.1696615
```

Another model was run using linear regression to show an example of removing the noise in the model. This model removed HOME, TDON, GENF, WRAT and TLAG. This cleaned the model up and presented it with a MSE of 1.864 with a standard error of 0.169.

```
mean((y.valid - pred.valid.ls2)^2) # mean prediction error
# 1.864044
sd((y.valid - pred.valid.ls2)^2)/sqrt(n.valid.y) # std error
# 0.1690677
```

### Tree based model

A third model was created using tree based methods. Cross-validation was used to determine which would be the appropriate number of end nodes. Seven nodes were selected using cross-validation and the tree can be seen below.

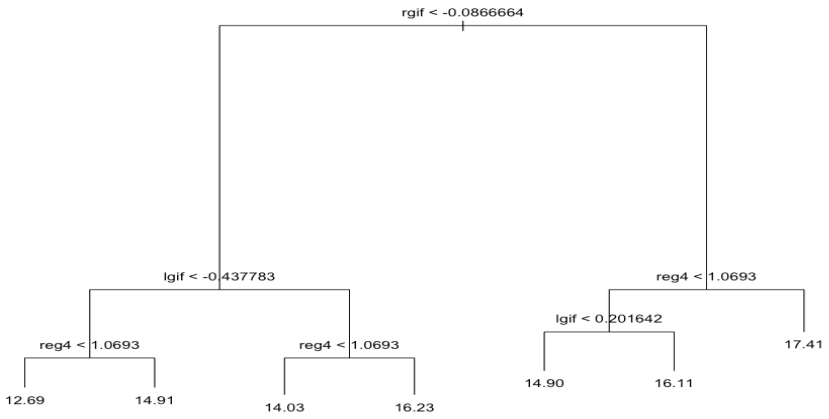


**Chart 3 – Tree constructed model**

This model gave us a MSE 2.3788 and a standard error of 0.1869. This model was worse than the least squares regression models.

```
Mean((y.valid – pred.valid.tree)^2) # mean prediction error
# 2.378834
sd((y.valid – pred.valid.tree)^2)/sqrt(n.valid.y) # std error
# 0.1868944
```

## Best subset selection model

A fourth model was created using cross-validation to determine the appropriate number of variables to use in model building and what those variables are. The cross-validation stated that six variables should be used and the following is the model created.

| (Intercept) | reg3 | reg4 | chld | hinc | rgif | agif |
|---|---|---|---|---|---|---|
| 14.2931010 | 0.3736171 | 0.6889575 | -0.5731359 | 0.5028017 | 0.4766170 | 0.6545193 |

This model gave us a MSE 1.980 and a standard error of 0.1689. This model was worse than the least squares regression models, but better than the tree based model.

```
mean((y.valid - pred.bestselect.model)^2) # mean prediction error
# 1.980266
sd((y.valid - pred.bestselect.model)^2)/sqrt(n.valid.y) # std error
# 0.1688628
```

## Bagging model

A fifth model was generated using bagging methods. The number of variables considered at each split was 20. This means that a very large tree was created, using all variables in true bagging approach.

This model gave us a MSE 1.712 and a standard error of 0.1751. This model was best so far, but has a pretty large variance in comparison.

```
mean((y.valid - pred.bagdamt)^2) # mean prediction error
# 1.711844
sd((y.valid - pred.bagdamt)^2)/sqrt(n.valid.y) # std error
# 0.1750859
```

## Random forest model

A sixth model was created using the random forest approach. Slightly different from bagging, rather than testing each variable at a split, this model only tested 5 variables at each node split of the tree.

This resulted in both a better MSE for the prediction set and a better standard error. The MSE for the random forest model was 1.666 and a standard error of 0.1734. This is the best model in the run so far.

```
mean((y.valid - pred.rf)^2) # mean prediction error
# 1.666401
sd((y.valid - pred.rf)^2)/sqrt(n.valid.y) # std error
# 0.1733991
```

## Boosting model

A seventh and final model was created to predict the amount of money a given donor would donate to the mailing campaign if they were to donate. This boosting model used 5,000 trees with an interaction depth of 4. This means that it built 5,000 models with 4 interactions considered per split.

This was the best model that we were able to run. It resulted in a MSE of 1.539 and a standard error of 0.1668.

```
mean((y.valid - post.valid.boostdamt)^2) # mean prediction error
# 1.539542
sd((y.valid - post.valid.boostdamt)^2)/sqrt(n.valid.y) # std error
# 0.1667702
```

## Overall DAMT model numbers

The models run to predict the amount of money that would be donated to the mailing campaign are below in Table 3. As was outlined in the paper above, the boost model was the model that performed best, with a mean prediction error of .539 and a standard error of 0.1668.

| MPE | Model |
|---|---|
| **1.867523** | LS1 |
| **1.867433** | LS2 |
| **2.378834** | Tree |
| **1.980266** | Best subset selection |
| **1.711844** | Bagging |
| **1.666401** | Random forest |
| **1.539542** | Boosting |

Table 3 – Models run for DAMT regression

## Conclusion

The task at hand was to create machine learning models that would help to drive cost effectiveness in a marketing mailing campaign. This included determining the appropriate amount of people to mail to, and determining who those people are. In addition, we determined how much money would be donated by the people we are predicting will respond to the mailing campaign.

The model selected was a boosting model in both predicting whether someone would donate and how much that person will donate when they respond. The boost model had an 87% accuracy rate on the validation data in the donor prediction data. The other boosting model had mean prediction error of 1.5359 in the data for the predicted donation amount.

Taken directly from the description of the program we know that the average donation per response is $14.50. We further know that each direct mailing costs $2.00 to produce and send. With an average response rate of around 10%, the campaign is not cost-efficient to send to all donors; in fact it will actually lose the company money to send it out like this. Running the numbers we know that if accurately predicted, we will earn net profit of $4,107.37 for mailing to 331 people.