# Topic 5. Decision Trees

Case 3: Donor Recapture

using Transaction, Overlay, and Census Data



## Reading Assignment

Berry and Linoff (2000)

- Pages 111–120 Decision trees (review).



## The Plan

1. Review and augment the previous discussion of decision trees.

2. Discuss the interpretation of tree structure.

3. Describe interactions.

4. Show what overfitting does to lift charts.

5. Explore model differences.



## Review

Let us review the ideas behind decision trees ...

# Fitting Decision Trees

Decision trees are based on a simple idea: One tries all possible splits of each input variable into two groups and uses the mean of each group to predict the target. The variable and split that produces the smallest mean squared error is accepted.

One then does the same for each sub node of the tree.

One continues splitting until some termination rule suggests stopping.



# Control Parameters

The standard reference is Breiman, Leo, Jerome H. Friedman, Ronald A. Olshen, and Charlse J. Stone (1984), *Classification and Regression Trees*, Chapman and Hall, Boca Raton FL, ISBN 0-412-04841-8.

In their formulation, there is one major control parameter called the complexity parameter $cp$. It is the proportionate decrease in training sample mean squared error required for a new branch of the tree to be added.

The other control parameters are crude restrictions on structure that, when chosen sensibly, affect the speed of the algorithm without much affecting results. Usually program defaults for these are adequate.



# Tree Complexity, $cp$

In the least squares fit, the proportional decrease in mse.lrn due to adding the last variable was 0.00051, which provides guidance in the choice of $cp$.

Trying the values 0.0001, 0.0005, 0.0008, 0.001, and 0.01 for $cp$ one finds that $cp = 0.0008$ gives the best mean squared error in the validation sample and that $cp = 0.001$ and 0.0001 also give interesting results.

Fitting details follow ...



# Table 11. Features Available to Tree

| File | Feature | Type | Number of Dummies |
|---|---|---|---|
| 464 | LASTGIFT | num | |
| 75 | PEPSTRFL | chr | 1 |
| 4 | STATE | chr | 31 |
| 11 | RECP3 | chr | 1 |
| 8 | DOB | num | |
| 6 | MAILCODE | chr | 1 |
| 359 | MHUC2 | num | |
| 465 | LASTDATE | num | |
| 460 | MINRAMNT | num | |

## Table 12. Definitions of the Available Features

| File | Feature | Type | Definition |
|---|---|---|---|
| 464 | LASTGIFT | num | Dollar amount of most recent gift |
| 75 | PEPSTRFL | chr | Has given to three consecutive card mailings |
| 4 | STATE | chr | State of residence |
| 11 | RECP3 | chr | Has given to CTY's P3 program |
| 8 | DOB | num | Date of birth |
| 6 | MAILCODE | chr | Mailing address is correct |
| 359 | MHUC2 | num | Census tract homeowner cost w/out mortgage |
| 465 | LASTDATE | num | Date associated with the most recent gift |
| 460 | MINRAMNT | num | Dollar amount of smallest gift to date |



# Decision Tree: Results

### charity/tree/cty_tree_001.r.Rout

```
mse.lrn =  19.8910972250757
mse.val =  18.8846605863929
mse.tst =  18.0788772736459
```

### charity/tree/cty_tree_0008.r.Rout

```
mse.lrn =  19.8099228572865
mse.val =  18.8311786562636
mse.tst =  18.3128051798462
```

### charity/tree/cty_tree_0001.r.Rout

```
mse.lrn =  19.0171539150594
mse.val =  19.6427237028794
mse.tst =  18.9090330593579
```

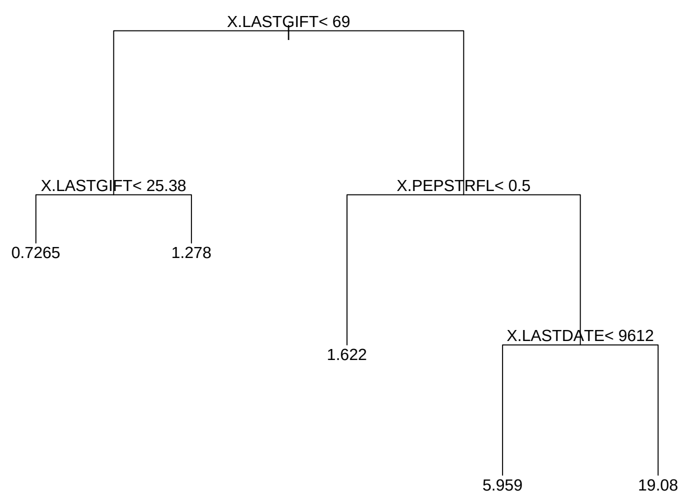


# Analysis of Results

First let's see what can be learned from the trees themselves ...



### Fig 61. Decision Tree, $cp = 0.001$




The left branch of the tree is the smaller side of the inequality; terminating values are the mean of the target at that leaf.

## Frequency Counts

PEPSTRFL is approximately a 50/50 split of the data.

But, if one looks at the frequency counts for LASTGIFT in file **lrn/num/464.frq** and LASTDATE in file **lrn/num/465.frq**, one learns that the $cp = 0.001$ tree is chopping close to the right hand edge of those two variables.

The number of observations in the right hand nodes of the tree could be too small.

Let's look ...



## Table 13. Tree Nodes, $cp = 0.001$

| Condition | Learning n | Learning mean | Validation n | Validation mean |
|---|---|---|---|---|
| $(\texttt{LASTGIFT} \geq 69)$ & $(\texttt{PEPSTRFL} \geq 0.5)$ | 173 | 7.86 | 52 | 4.33 |
| $(\texttt{LASTGIFT} \geq 69)$ & $(\texttt{PEPSTRFL} \geq 0.5)$ & $(\texttt{LASTDATE} \geq 9612)$ | 25 | 19.08 | 7 | 0 |



## Downright Suspicious!

It looks very much like the rightmost node of the tree is a learning mistake. The tree may not generalize well.

Also of interest is the dependence of the mean of the LASTGIFT cut on PEPSTRFL.

Let's cut closer to the middle of LASTGIFT and look ...



**Table 14. Gift Percentiles**

| Percentile | TARGET (Dollars) | LASTGIFT (Dollars) |
|---|---|---|
| min | 0 | 0 |
| 25 | 0 | 10 |
| 50 | 0 | 15 |
| 75 | 0 | 20 |
| 80 | 0 | 21 |
| 90 | 0 | 25 |
| 95 | 3 | 30 |
| 96 | 8 | 35 |
| 97 | 10 | 40 |
| 98 | 15 | 50 |
| 99 | 20 | 50 |
| max | 200 | 1000 |

Recall that these are lapsed donors so that one expects `LASTGIFT` to be larger than `TARGET`

**Table 15. LASTGIFT by PEPSTRFL**

| | PEPSTRFL $< 0.5$ | | PEPSTRFL $\geq 0.5$ | |
|---|---|---|---|---|
| | n | mean | n | mean |
| LASTGIFT $< 20$ | 17886 | 0.64 | 23971 | 0.74 |
| LASTGIFT $\geq 20$ | 17276 | 0.83 | 7767 | 1.23 |
| Difference | | 0.19 | | 0.49 |



# An Interaction!

We have learned something: There is an interaction.

An interaction is when the slope coefficient on one feature depends on the value of another feature.

A crude estimate of the slope coefficient on LASTGIFT in the learning sample is 0.019 when PEPSTRFL $= 0$ and 0.049 when PEPSTRFL $= 1$, because LASTGIFT changes by \$10 between groups.

The slope of LASTGIFT depends on PEPSTRFL!

More about this later.

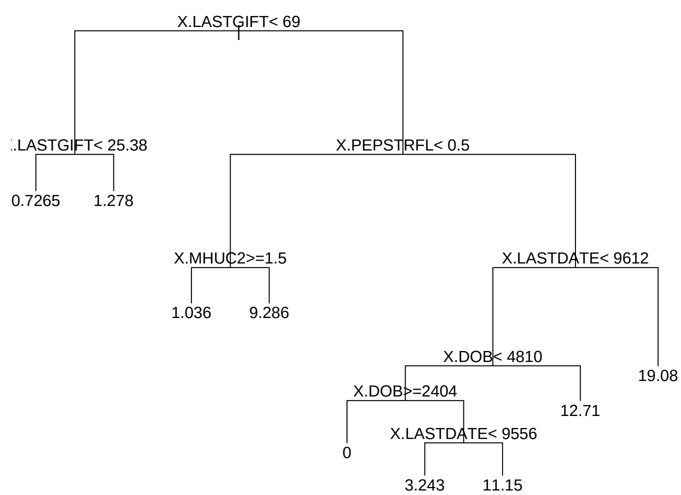


# Onward

The next two trees ...



**Fig 62. Decision Tree,** $cp = 0.0008$




The left branch of the tree is the smaller side of the inequality; terminating values are the mean of the target at that leaf.

## An Anomaly

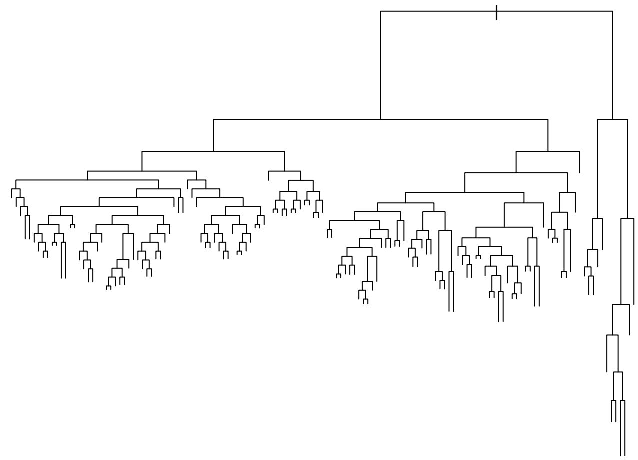
Recall that the tree with complexity $cp = 0.0008$ is the preferred tree according to mse.val.

The regression analysis put STATE in as the third most important variable.

Our preferred tree does not use any of the 31 STATE dummies.



**Fig 63. Decision Tree,** $cp = 0.0001$





## Too Complex

The tree with complexity $cp = 0.0001$ is too complex to make much sense of visually.

One can examine the printed output,
**tree/cty_tree_0001.r.Rout**,
to at least see what variables are included. A summary is in file
**tree/cty_tree_0001.cuts.txt**.

One learns that every variable in Table 12 is in the tree except MAILCODE and 14 of the STATE dummies.



## Why Trees are Popular

As we have just seen, trees are easy to interpret.

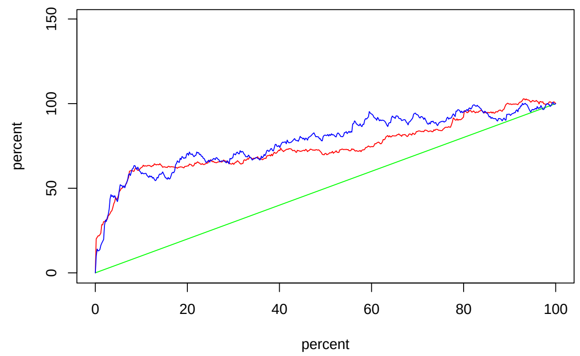
That is why the tool is so popular.

# Onward

Next are the lift charts for $cp = 0.0008$, which was our best tree according to MSE.



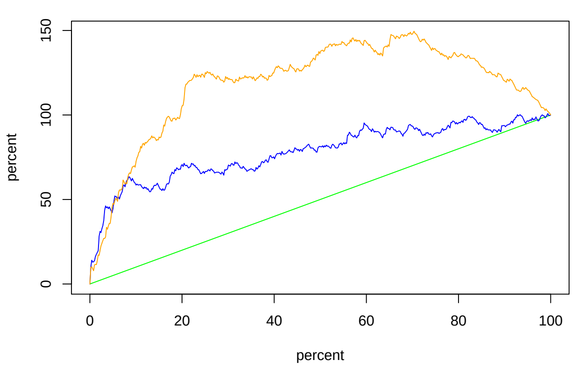
## Fig 64. Lift Charts, $cp = 0.0008$




The green curve shows net revenue if persons in the learning sample were mailed solicitations in random order. The red curve shows net revenue in the learning sample if persons are sorted by their predicted gift and mailed solicitations in sorted order, highest first; blue is the same for the validation sample. The plots are normalized so endpoints plot at (100,100). Net revenue is the gift less a mailing cost of \$0.68.



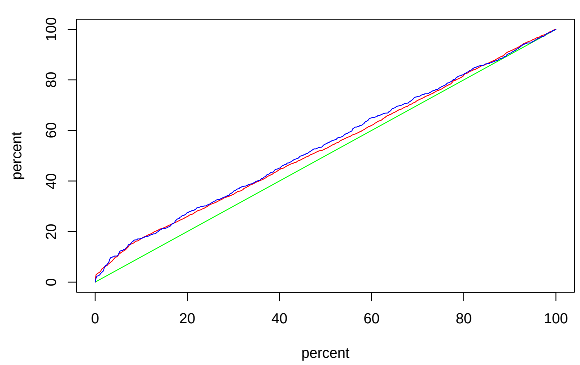
## Fig 65. Lift Charts, $cp = 0.0008$




Same as Fig 64 except that the orange line is the blue line from Fig 54, which shows the lift of the regression model in the validation sample.

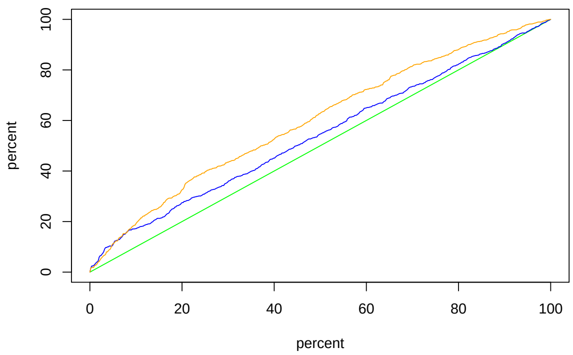


## Fig 66. Conventional Lift Charts, $cp = 0.0008$




Same as Fig 64 but gross revenue instead of net revenue.

## Fig 67. Conventional Lift Charts, $cp = 0.0008$




Same as Fig 66 except that the orange line is the blue line from Fig 55, which shows the lift of the regression model in the validation sample.



## Lift Charts

The decision trees are not providing as much lift as regression.

The trees seem to do a good job of identifying the largest donors. But after accurately predicting the largest 10%, they do not seem to be able to tell one donor from another.



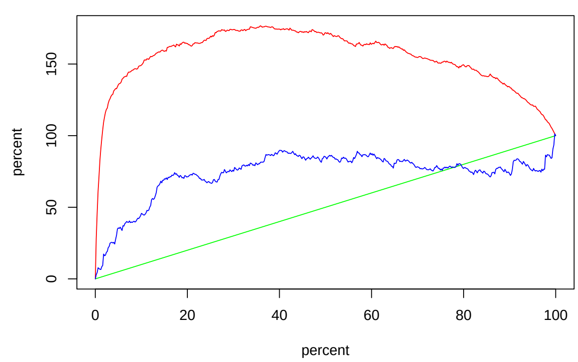
## Overfitting

Lift charts can cast the generalization failure that comes form over fitting into sharp relief.

The next two slides for complexity $cp = 0.0001$ illustrate ...

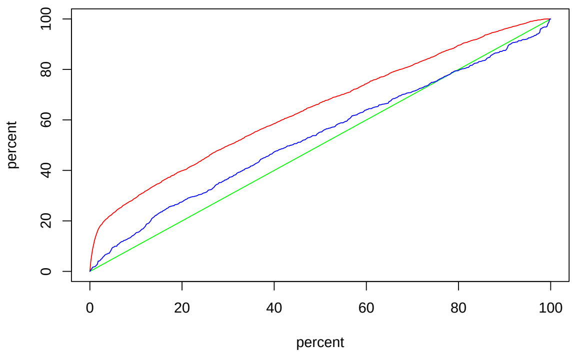


## Fig 68. Lift Charts, $cp = 0.0001$




The green curve shows net revenue if persons in the learning sample were mailed solicitations in random order. The red curve shows net revenue in the learning sample if persons are sorted by their predicted gift and mailed solicitations in sorted order, highest first; blue is the same for the validation sample. The plots are normalized so endpoints plot at (100,100). Net revenue is the gift less a mailing cost of \$0.68.

## Fig 69. Lift Charts, $cp = 0.0001$




Same as Fig 68 but gross revenue instead of net revenue.



## Overfitting

The drastically different shapes of the lift chart in the learning and validation samples display a massive generalization failure caused by over-fitting.



## Onward

The performance measures ...



## Table 16. Performance Measures

| | | Mean Squared Error | | |
|---|---|---|---|---|
| Model | Specification | Learning | Validation | Test |
| Mean | learning sample | 20.09922 | 18.82322 | 17.86605 |
| Regr | selected model* | 19.96083 | 18.67709 | 17.80003 |
| Nnet | 6 iter X 5 HU | 19.97731 | 18.72594 | 17.85258 |
| Tree | $cp = 0.001$ | 19.89110 | 18.88466 | 18.07888 |
| Tree | $cp = 0.0008$ | 19.80992 | 18.83118 | 18.31281 |
| Tree | $cp = 0.0001$ | 19.01715 | 19.64272 | 18.90903 |

$^*R^2 = 0.0068853$

## Trees Are Overfitting

Examination of the performance measures reveals that

- All decision trees achieved better in sample fits than the neural net models or the linear regression models.

- The performance of the decision trees in the validation sample is awful. The performance is actually worse than just using the mean of the data in the learning sample as the predictor in the validation sample.



## Can the Trees be Fixed?

We could probably fiddle with control parameters and get them to do better.

But even as it is, the trees did tell us that there was an interaction, which, as we shall soon see, is very useful information.



### How Different are the Models?

To find out, we can look at the correlations of predictions with each other and with the target ...



## Learning Sample

The Correlations:

| | target | yhat.regr | yhat.nnet | yhat.tree |
|---|---|---|---|---|
| target | 1.000000 | 0.0829767 | 0.0802823 | 0.1199726 |
| yhat.regr | 0.082977 | 1.0000000 | 0.7477238 | 0.3533951 |
| yhat.nnet | 0.080282 | 0.7477238 | 1.0000000 | 0.2718147 |
| yhat.tree | 0.119973 | 0.3533951 | 0.2718147 | 1.0000000 |

The $R^2$:

Rsquared.regr $= (0.08297671)^2 = 0.0068853$
Rsquared.nnet $= (0.08028232)^2 = 0.0064453$
Rsquared.tree $= (0.11997260)^2 = 0.0143934$

## Validation Sample

The Correlations:

| | target | yhat.regr | yhat.nnet | yhat.tree |
|---|---|---|---|---|
| target | 1.000000 | 0.0884041 | 0.0728670 | 0.0603211 |
| yhat.regr | 0.088404 | 1.0000000 | 0.7372246 | 0.3502917 |
| yhat.nnet | 0.072870 | 0.7372246 | 1.0000000 | 0.2792420 |
| yhat.tree | 0.060321 | 0.3502917 | 0.2792420 | 1.0000000 |

The $R^2$:

Rsquared.regr $= (0.08840412)^2 = 0.0078153$
Rsquared.nnet $= (0.07286999)^2 = 0.0053100$
Rsquared.tree $= (0.06032113)^2 = 0.0036386$



## Model Comparison

What is different about these models?

The $R^2$ suggest overfitting by trees and underfitting by nets, if one is willing to take the regression model as the benchmark.

Recall from the lift charts that the neural nets were making bizarre prediction errors at the left of the charts. This seems to have damaged generalization even though they look like underfits.

The model predictions are not highly correlated: The models are making different predictions.



## Decision Trees Main Points

1. Decision trees are popular because they are interpretable.
2. Our application appears to have an interaction that we discovered by means of trees.
3. Overfitting causes generalization failure that is apparent in lift charts.
4. Regression, nets, and trees are different tools and produce different results.
5. It is a wise precaution to use several tools in an application!



Blank page