Topic 11. AdaBoost

Case 3: Donor Recapture

using Transaction, Overlay, and Census Data

Reading Assignment

Berry and Linoff (2000)

- Pages 213–221. Boosting (review).

Boosting

Boosting is one of the most powerful learning ideas introduced in the last ten years.

Hastie, Tibshirani, and Friedman (2001)

Boosting

Boosting is the idea of using the scores from one tool as features for use by another tool. Berry and Linoff, pp. 213 – 221 describe four variants:

- **Segmented Input Combination Models** use different models for different parts of the input; only one model is used for a case.
- **Modeled Segmentation Combination Models** use the results from one model to segment the input, and then use another model to determine the output.
- **Error Fixing Combination Models** use high-confidence results from one model and build a separate model from the low confidence results.
- **Data Enhancement Combination Models** use the results of one model as input into another.

This is just the tip of the iceberg: Many other boosting methods have been proposed.

AdaBoost: The Idea

Use a weighted sum of classifiers as a classifier. It is like voting: If two classifiers think the item is in category A and the third thinks it is not, then A is what gets reported.

In training, AdaBoost tries to make subsequent classifiers learn from the mistakes of their predecessors. Predecessors mistakes are overweighted when the subsequent classifier trains.

A classifier's voting rights are proportional to its success in training.

It is also a perfect illustration of the disconnect. As we shall see, AdaBoost changes the training loss function. This has the effect of worsening performance on CER loss but bettering performance on lift charts.

AdaBoost: The Algorithm

1. Score the targets y_i as -1, 1 rather than the usual 0, 1.

2. Initialize the weights $w_i = 1/n$ for $i = 1, \ldots, n$.

3. For $m = 1$ to M :

   (a) Fit a classifier $C_m(x)$ to the training data using the weights w_i.

   (b) Let err_m be the sum of all w_i for which the classification was incorrect.

   (c) Compute $\alpha_m = \log\left[(1 - \text{err}_m)/\text{err}_m\right]$.

   (d) If the classification was correct, then let $w_i^* = w_i$, otherwise let $w_i^* = w_i\left[(1 - \text{err}_m)/\text{err}_m\right]$.

   (e) Set $w_i = w_i^* / \sum_{t=1}^{n} w_i^*$.

4. If $C(x) = \sum_{m=1}^{M} \alpha_m C_m(x)$ is positive, then output 1, otherwise output -1.

Why Does it Work?

The classifier $C(x)$ defined in step 4 of the algorithm is an additive dictionary type method

$$C(x) = \sum_{m=1}^{M} \gamma_m f(x, \beta_m)$$

like neural nets. The idea behind the boosting strategy is similar to the one we used to fit neural nets as seen from step 3 of the algorithm.

Thus, what the method is doing, at least in very large samples, is finding coefficients γ_m and β_m to minimize

$$\sum_{i=1}^{n} L\left[y_i, C(x_i)\right]$$

for some loss function L.

What is the Loss Function

First notice that because of the coding convention for y_i of -1, $+1$, a classification error occurs when the product

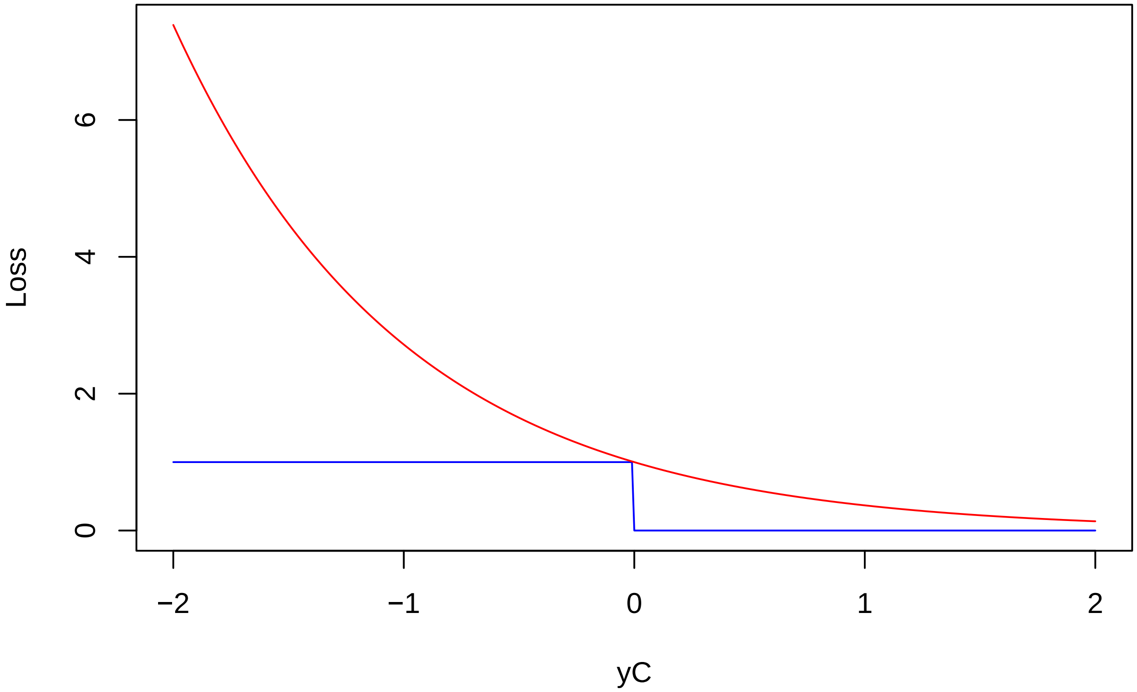
$$y_i C(x_i)$$

is negative. For AdaBoost the loss function is

$$L(y, C) = \exp(-yC),$$

seen graphically next slide.

Fig 115. AdaBoost Loss Function



The red curve shows the AdaBoost loss function $L(y, C) = \exp(-yC)$. The blue curve shows the CER (classification error rate) loss function.

AdaBoost for Classification Trees

Most classification trees are implemented as in Breiman, Friedman, Olshen, and Stone (1984), *Classification and Regression Trees*, Wadsworth. The major control parameter is called the complexity parameter cp. The tree stops growing if a split cannot reduce loss by

$$cp \cdot (\#\text{ leaves}) \cdot (\text{Loss at stump})$$

Another control parameter is maximum tree depth J. Hastie, Tibshirani, and Friedman (2001) recommend setting cp so small that the tree grows until J stops it.

The AdaBoost control parameters become M, the number of AdaBoost iterations, and J, the tree depth.

AdaBoost for Classification Trees

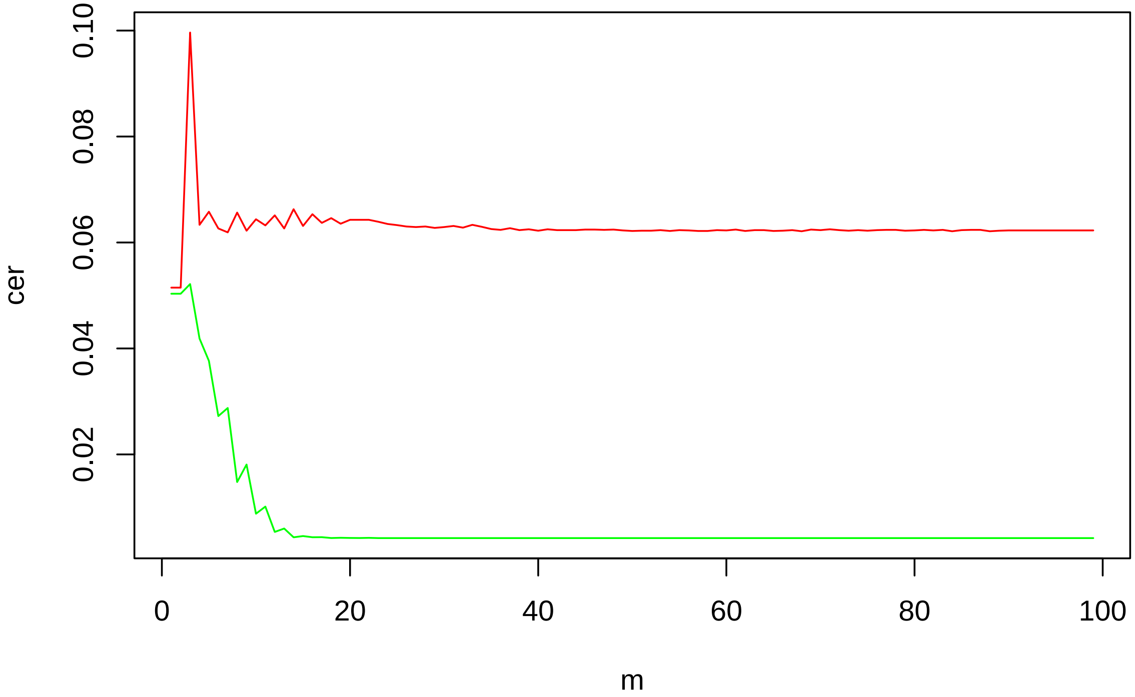
Hastie, Tibshirani, and Friedman (2001) recommend growing the tree on the training sample and selecting J and M to minimize loss in the validation sample.

For the Charity data, the tree will not grow at all unless $J \geq 25$, which conflicts with Hastie, Tibshirani, and Friedman's advice to keep $J < 8$. The minimum in the training set occurs at $M = 98$ when $J = 25$.

Hastie, Tibshirani, and Friedman warn that AdaBoost works best when the classification proportions are near 50-50. In the charity data, the classification proportions are 5-95.

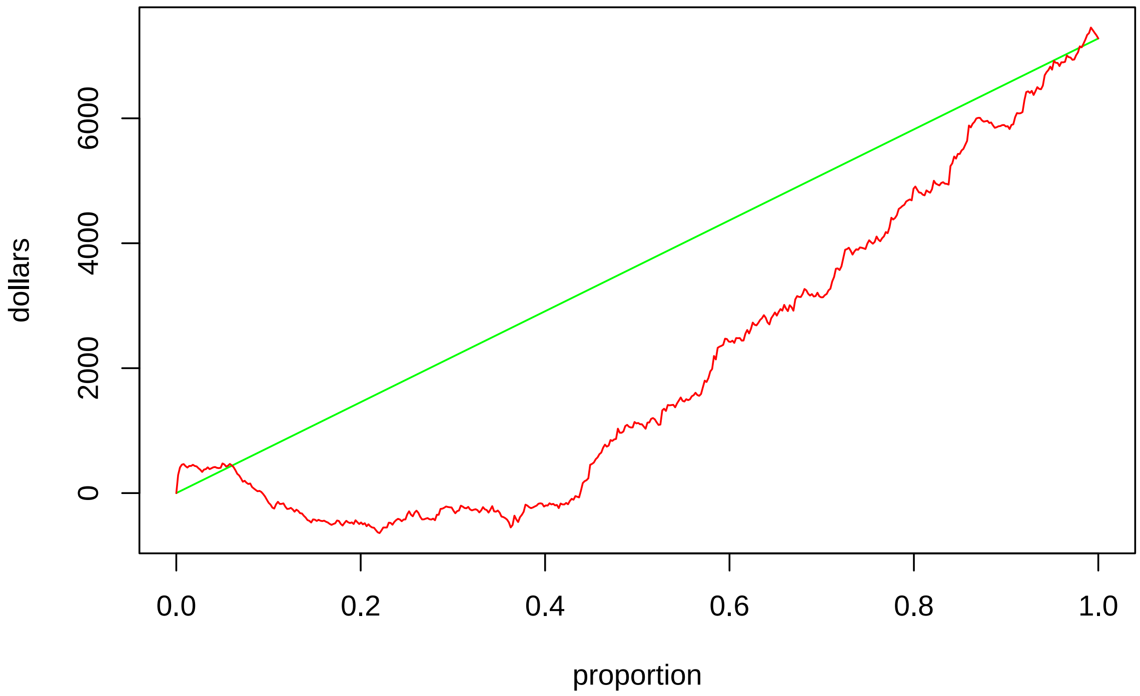
Watch the disconnect operate in the validation sample: CER loss is not improved by boosting but net revenue lift charts are.

Fig 116. AdaBoost Classification Error Rates for Charity Data



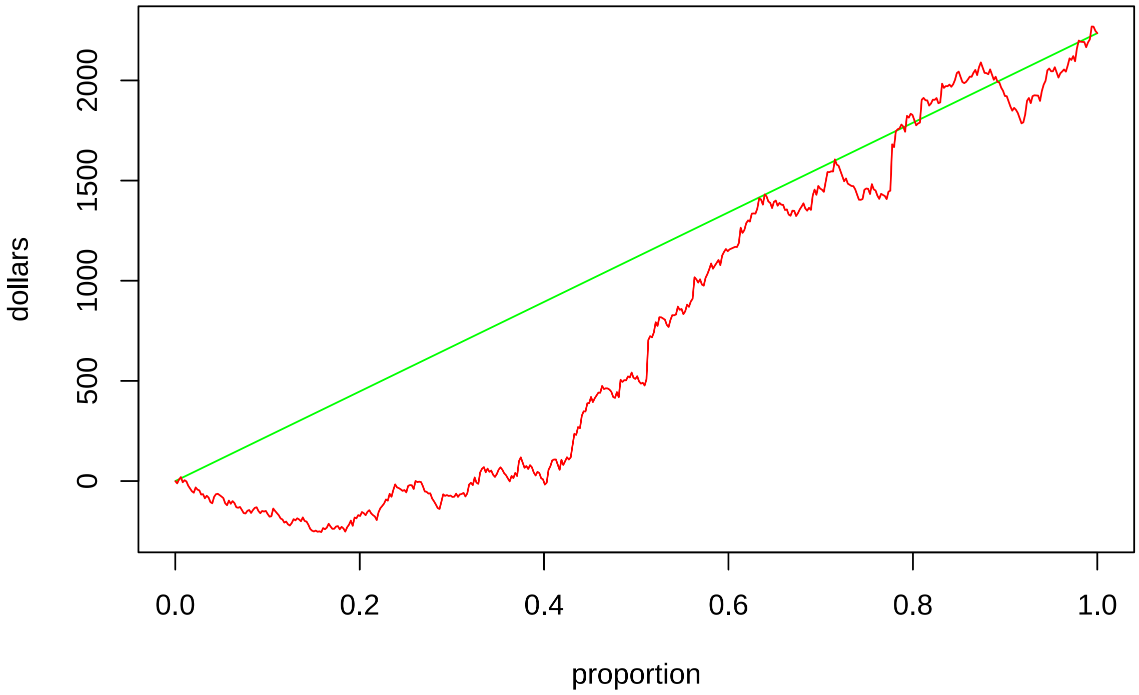
The red curve shows the AdaBoost classification error in the validation sample. The green curve shows the AdaBoost classification error in the training sample.

Fig 117. AdaBoost Lift Chart in Learning Set, $J = 25$, $M = 1$



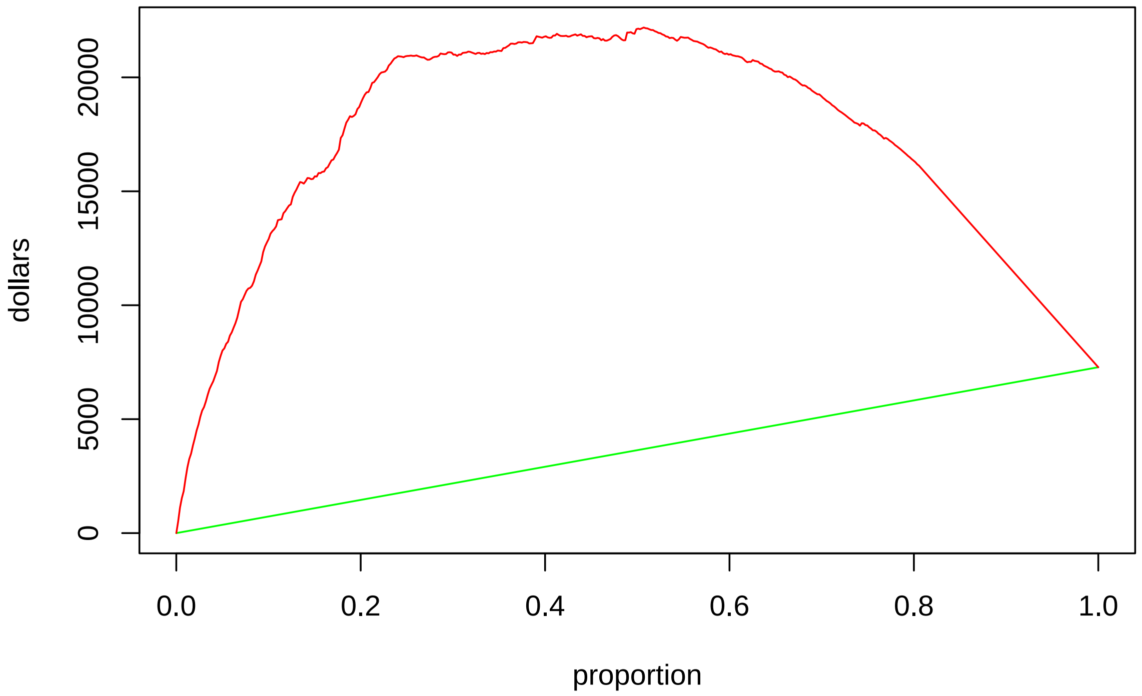
The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 118. AdaBoost Lift Chart in Validation Set, $J = 25$, $M = 1$



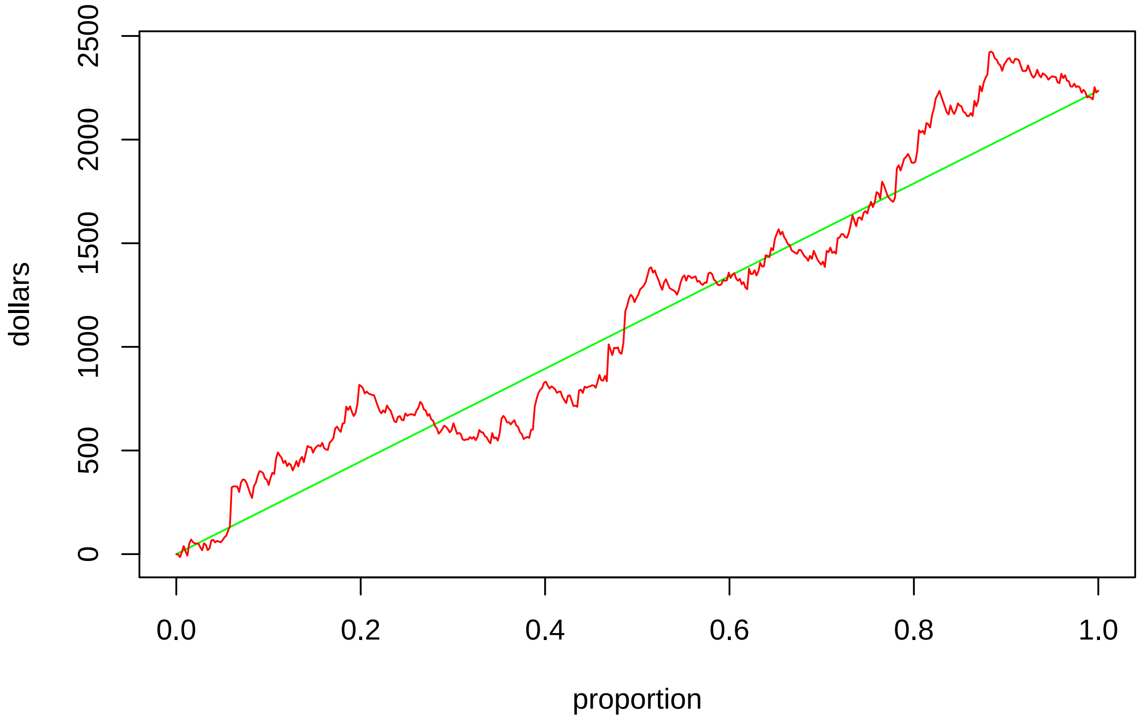
The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 119. AdaBoost Lift Chart in Learning Set, $J = 25$, $M = 10$



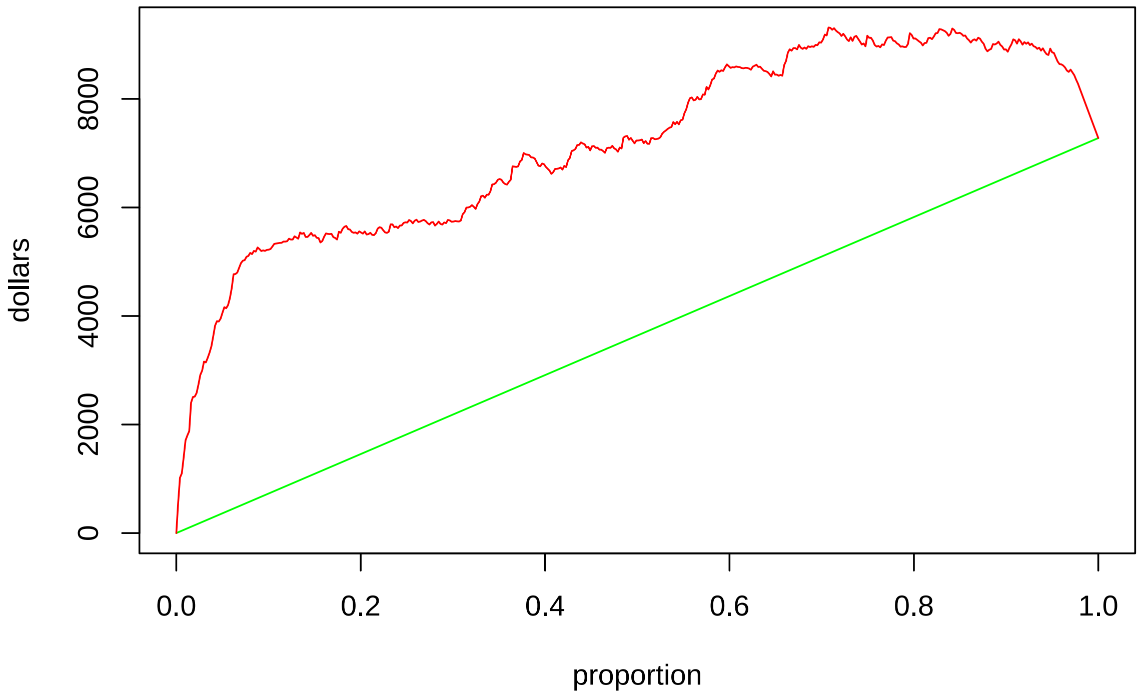
The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 120. AdaBoost Lift Chart in Validation Set, $J = 25$, $M = 10$



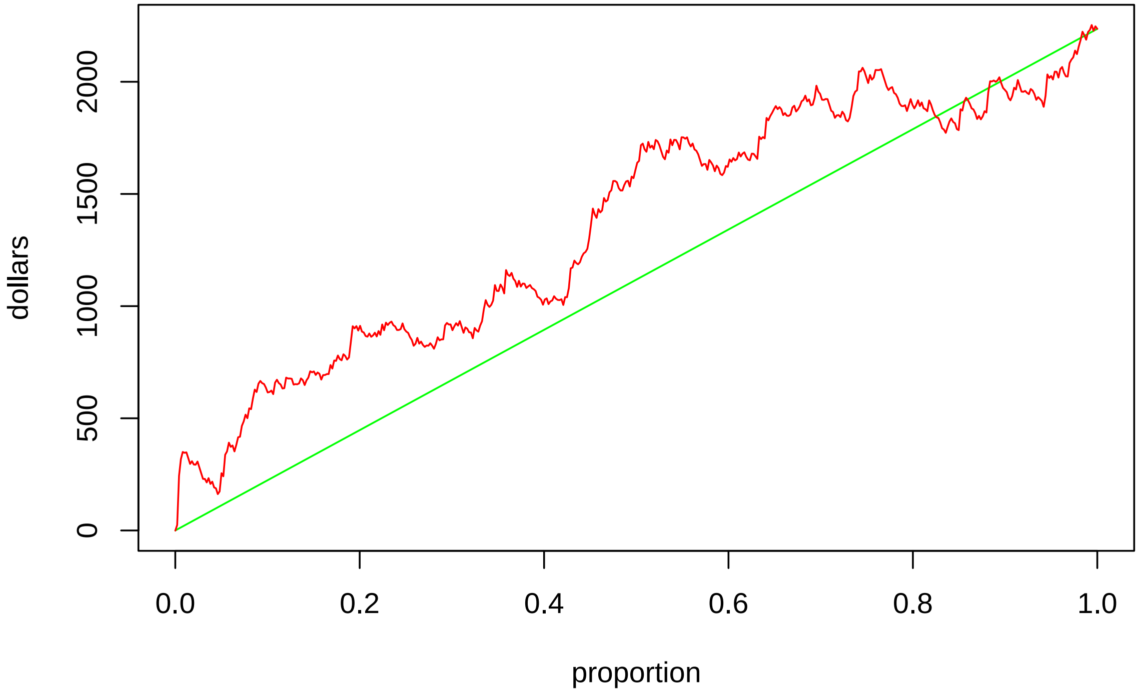
The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 121. AdaBoost Lift Chart in Learning Set, $J = 25$, $M = 50$



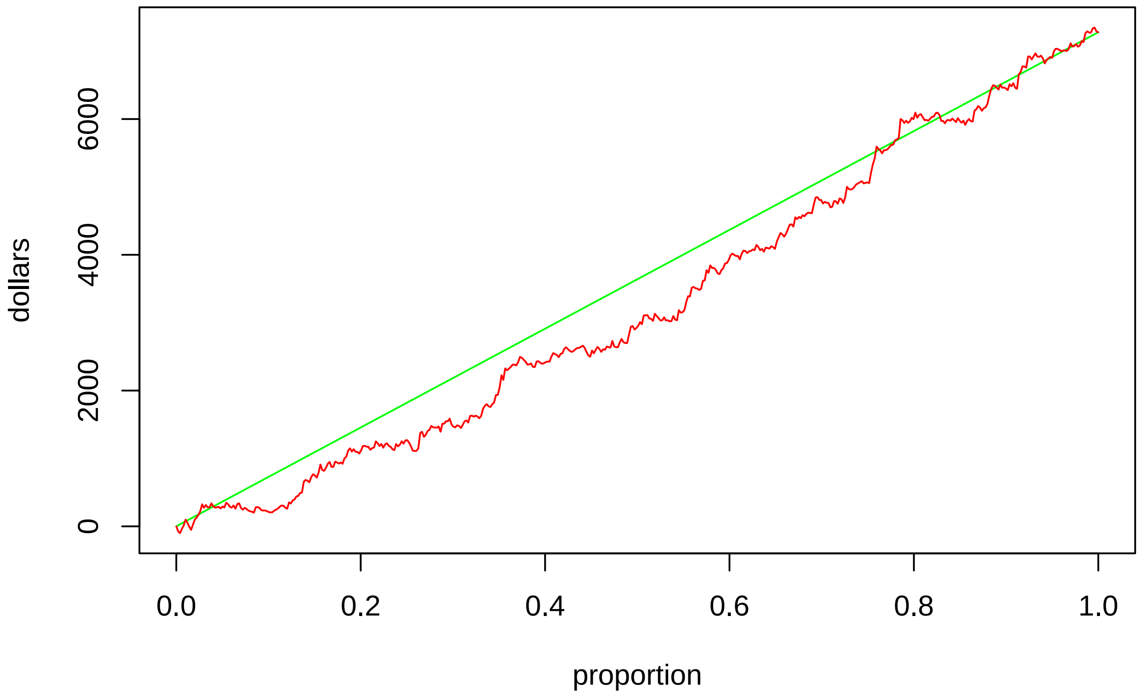
The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 122. AdaBoost Lift Chart in Validation Set, $J = 25$, $M = 50$



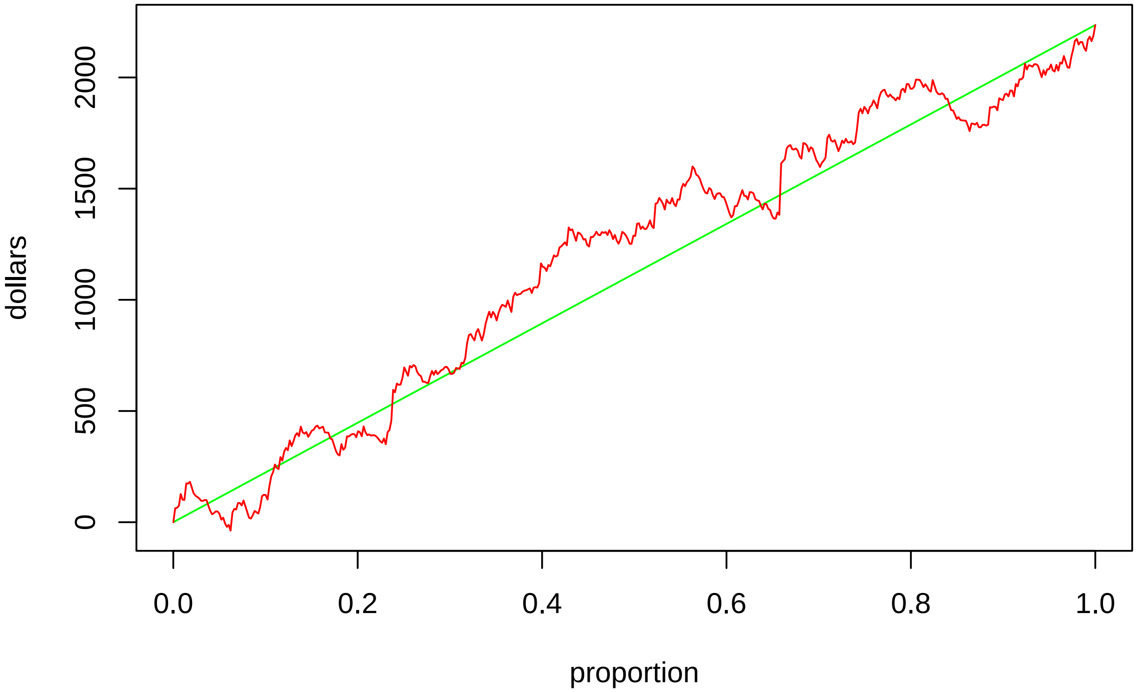
The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 123. AdaBoost Lift Chart in Learning Set, $J = 25$, $M = 90$



The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.

Fig 124. AdaBoost Lift Chart in Validation Set, $J = 25$, $M = 90$



The green curve shows expected net revenue if persons were mailed solicitations in random order. The red curve shows net expected revenue if persons are sorted according to the confidence scores of an AdaBoost classification tree.