36-350: Data Mining   Handout 24   November 17, 2003

Logistic regression

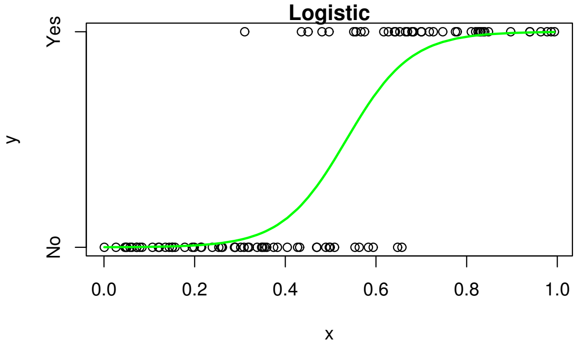
Logistic regression is a classification analog of linear regression. It is preferable to trees in the same situations that linear regression is preferable to regression trees:

- when effects are small
- when predictors contribute additively (no interactions)

The disadvantages are the same as well: logistic regression requires more assumptions than a tree and it is sensitive to outliers.

Logistic regression model—To compute the probability that the response equals a given value, take a linear combination of the predictors and force the result to be in $[0, 1]$ via the logistic function:

$$\begin{aligned}\text{logistic}(x) &= \frac{1}{1+e^{-x}} \\ p(y=1|\mathbf{x},\mathbf{a}) &= \text{logistic}(a_1x_1 + a_2x_2)\end{aligned}$$

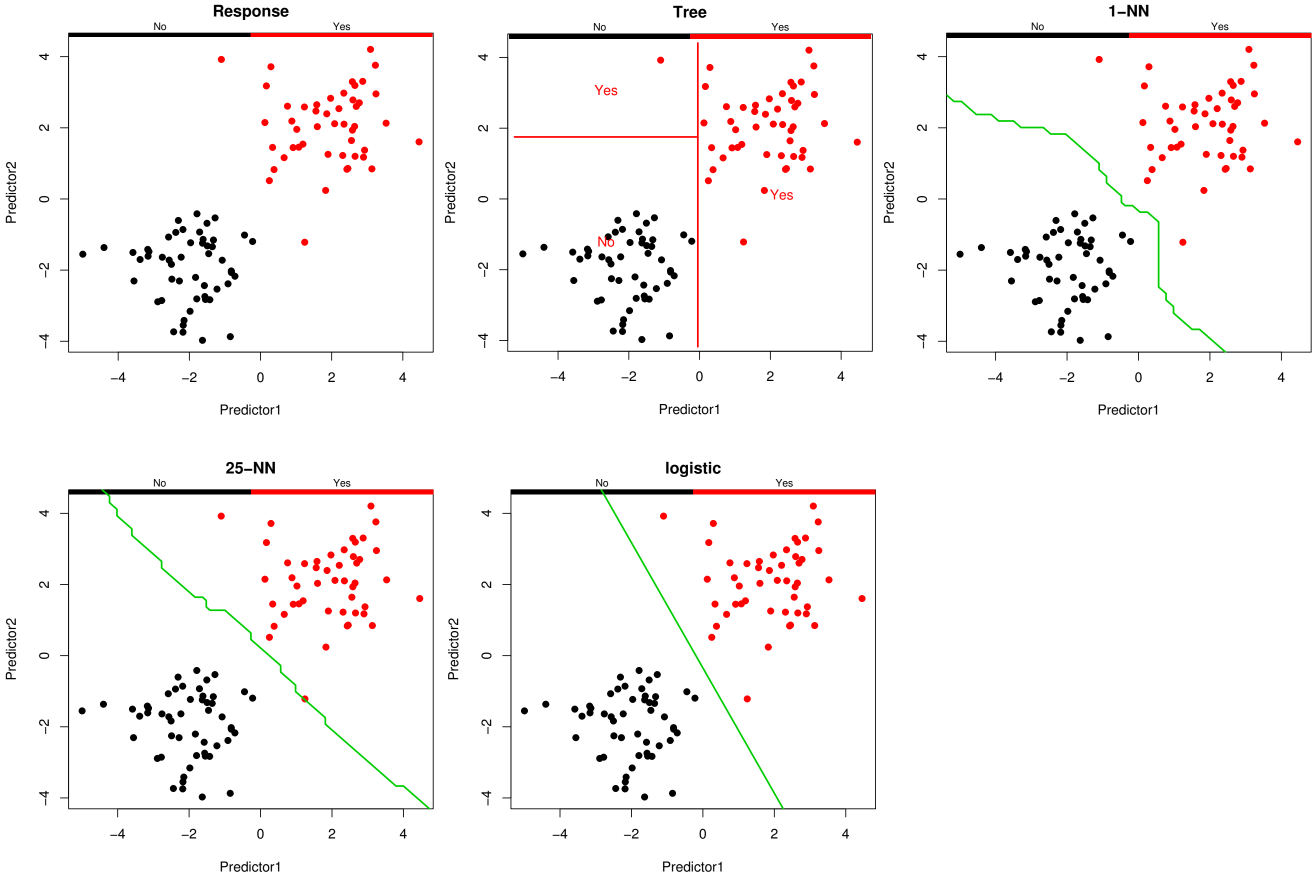


$$p(y = \text{Yes} \mid x) = \text{logistic}(14.137x - 7.565)$$

The vector $\mathbf{a}$ is the parameter of the model. The model is fit to training data by minimizing the deviance (handout 23), or equivalently by maximizing the predicted probability of the true outcomes:

$$\max_{\mathbf{a}} \prod_i p(y = y_i|\mathbf{x}_i, \mathbf{a})$$

Results for a simple dataset:



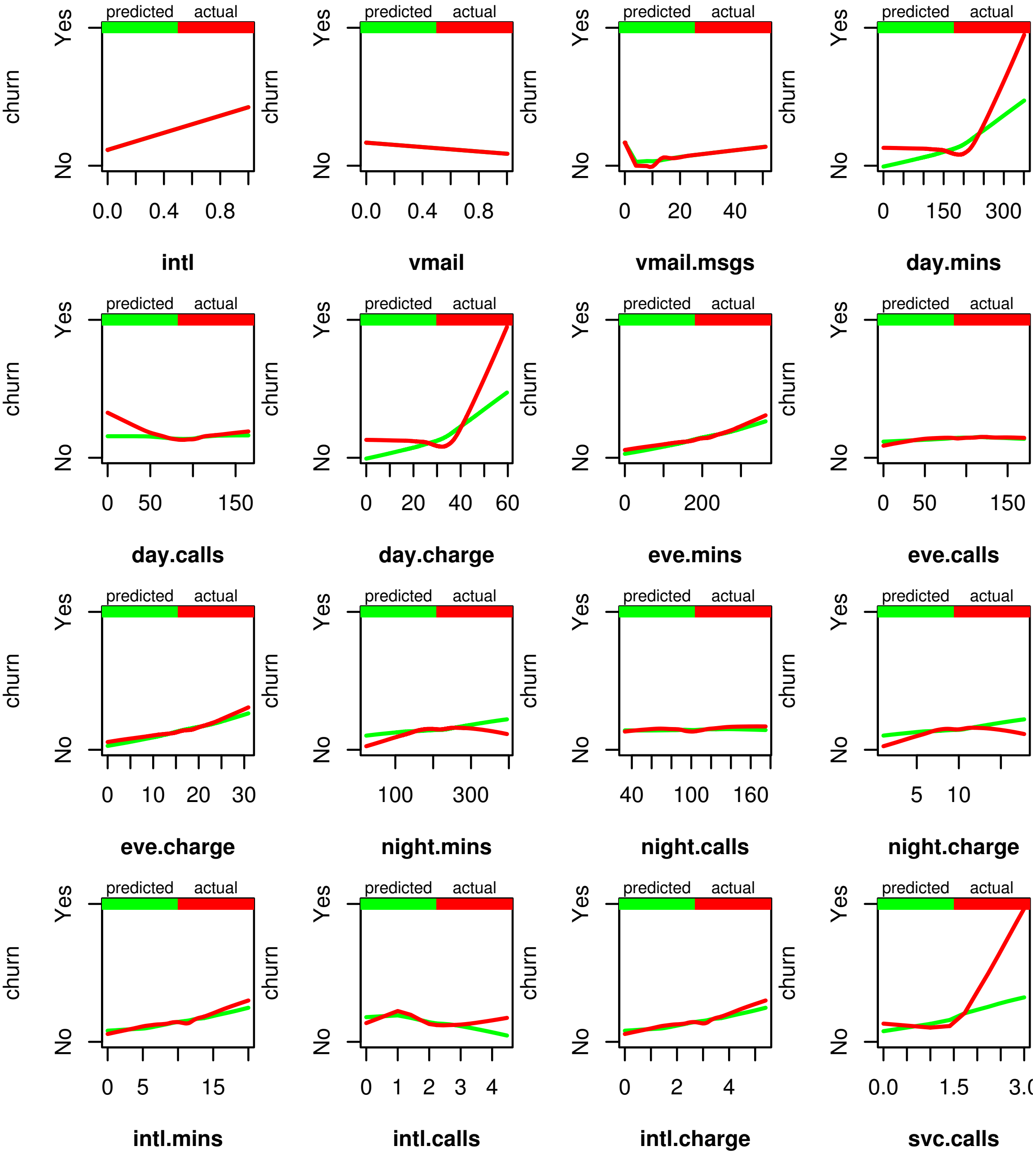
Just like linear regression, the predicted response has linear contours. The line where $p(y = 1|\mathbf{x}, \mathbf{a}) = 0.5$ is the decision boundary. As you move away from the boundary, the probability approaches zero or one. Whereas for the tree, the probability is constant in each cell.

Because it wants to give high probability to the correct outcome, logistic regression wants a **large margin** between the boundary and the data. The red point near the boundary causes logistic regression to tilt at a weird angle.

$k = 25$ was chosen by cross-validation, giving a straight boundary similar to logistic regression, but at a different angle.

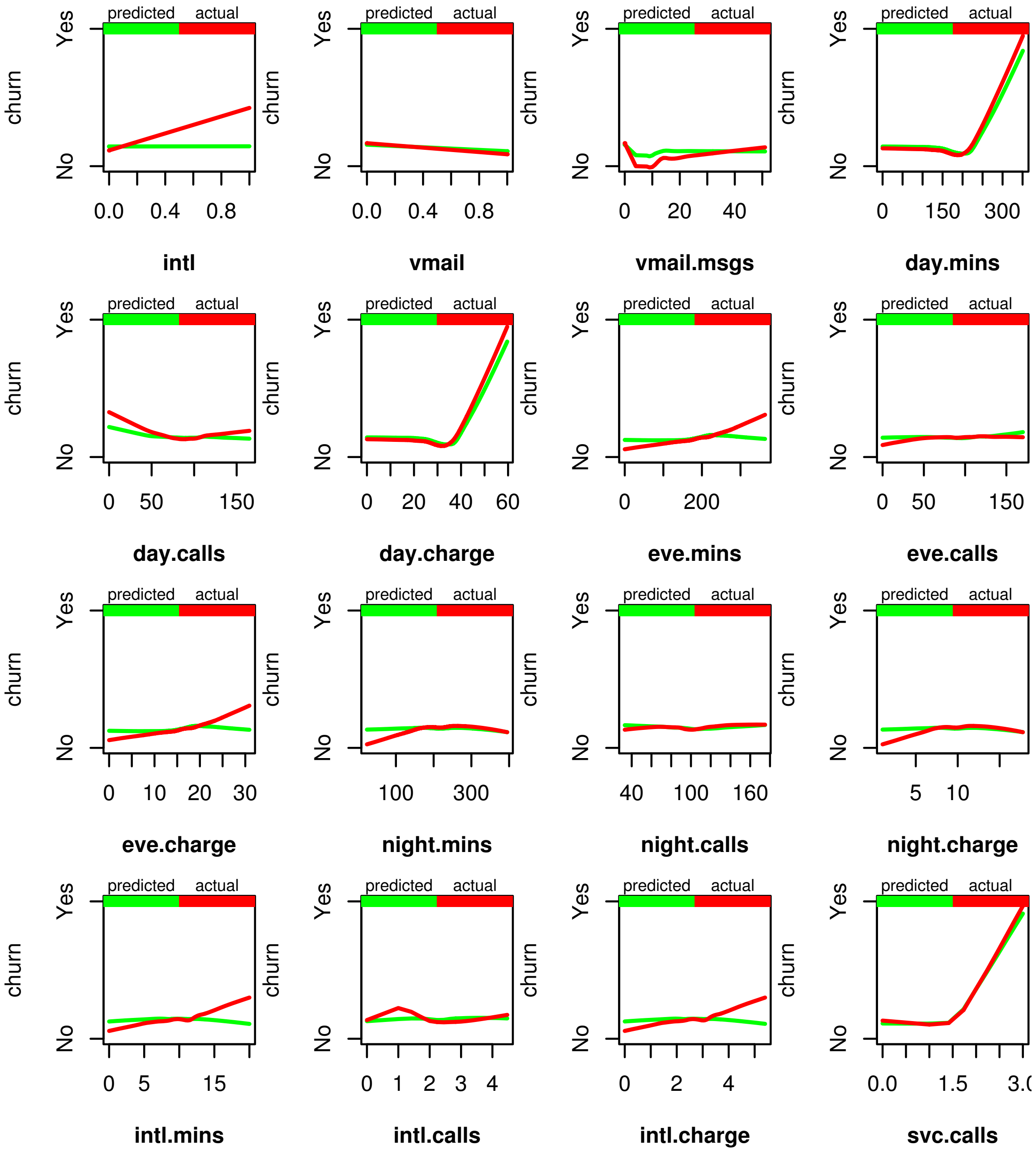
Churn dataset

The tree from handout 22 has a misclassification rate of 6%, while logistic regression has 13%. To see why, we look at **marginal model plots**, which overlay the predicted probability and the actual probability (estimated by lowess). These are used similarly to residual plots for linear regression.



Some of the curves are modeled very well. But the probability of churn is sharply nonlinear with respect to some of the variables, such as `day.mins`. Logistic regression tries to approximate this with a line, and performance suffers.

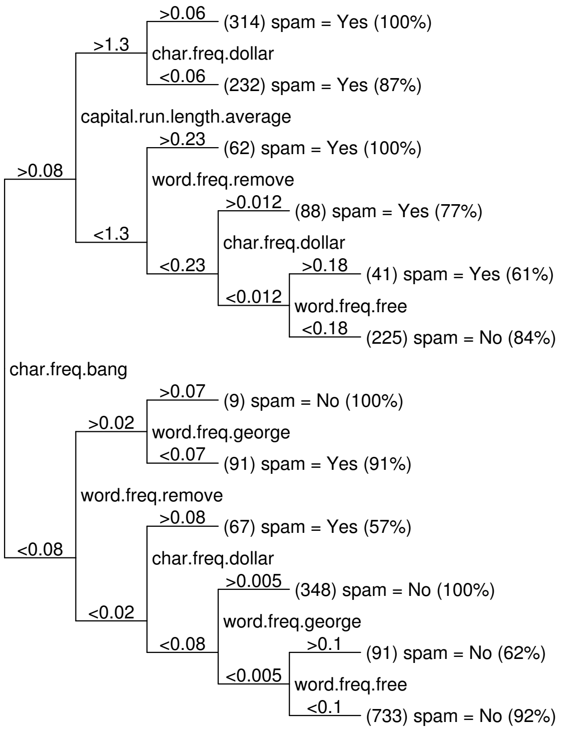
The marginal model plots for the tree:



Unsolicited bulk email, aka spam, is an annoyance for many people. Suppose you wanted to make a classifier to detect spam and filter it out of your mail. How would you do it? There are four basic steps:

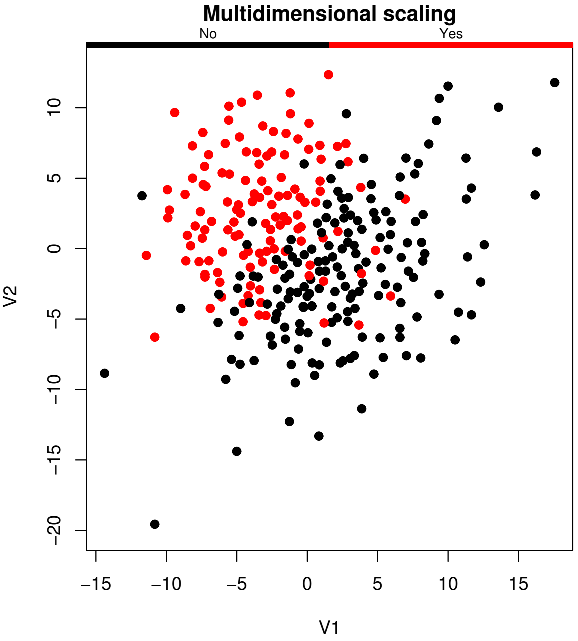
- Collect a dataset of email labeled as "spam" and "not spam".
- Define variables which can be used to discriminate the two classes.
- Divide the data into a training set and test set.
- Train a classifier and evaluate it on the test set.

George Forman, a researcher at Hewlett-Packard, has made a spam dataset. He took 4601 of his own email messages, labeled them, and extracted various features (57 in all). The features include the frequency of various words (e.g. "money"), special characters (e.g. dollar signs), and the use of capital letters in the message. The following refers to one particular 50% train-test split.



This is a subset of the best sized classification tree (37 leaves total). The tree has 5% misclassification error on the training set and 8% on the test set. This tree is suspicious because the same variables are being used, in essentially the same way, down each branch. It seems that we have multiple equally important predictors.

Multidimensional scaling reveals that the classes have a linear boundary, suggesting logistic regression should work well. Logistic regression turns out to have 5% error on the training set and 6% on the test set, an improvement which is statistically significant. It is important to transform the variables to get the best performance from logistic regression (these variables were very skewed to begin with).



5-NN has 6% error on the training set and 7% on the test set. Why does nearest neighbor do worse than logistic regression?

Suppose instead we use 5% of the data for training (116 points). From this reduced training set, a tree has 8% training error and 14% test error, while logistic regression has 0% training error and 23% test error. Logistic regression is badly overfitting, because there are only 57 predictors. Logistic regression typically needs 10 samples per coefficient to get a good fit. (9-NN has 10% test error, so it is the best here.)

References

[1] Spam data. `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/spambase/`