MATH 189 Final Project

Spam Classification

Consider an email spam dataset that consists of 4601 email messages, from which 57 features have been extracted. These features are described as follows:

- 48 features giving the percentage of certain words (e.g., "business", "free", "george") in a given message
- 6 features giving the percentage of certain characters (; ([! $ #)
- feature 55: the average length of an uninterrupted sequence of capital letters
- feature 56: the length of the longest uninterrupted sequence of capital letters
- feature 57: the sum of the lengths of uninterrupted sequences of capital letters

The data set contains a training set of size 3065 (link), and a test set of size 1536 (link). One can perform several types of preprocessing to this data. Try each of the following separately:

1) Standardize the columns so that they all have zero mean and unit variance;
2) Transform the features using $\log(x_{ij} + 1)$;
3) Discretize each feature using $I(x_{ij} > 0)$.

(a) For each version of the data, visualize it using the tools introduced in the class.

(b) For each version of the data, fit a logistic regression model. Interpret the results, and report the classification errors on both the training and test sets. Do any of the 57 features/ predictors appear to be statistically significant? If so, which ones?

(c) Apply both linear and quadratic discriminant analysis methods to the standardized data, and the log-transformed data. What are the classification errors (training and test)?

(d) Apply linear and nonlinear support vector machine classifiers to each version of the data. What are the classification errors (training and test)?

(e) Apply tree-based classifiers to this data. What are the classification errors (training and test)?

Report classification errors using different methods and different preprocessed data in a table, and comment on the different performances.

Finally, use either a single method with properly chosen tuning parameter or a combination of several methods to design a classifier with test error rate as small as possible. Describe your recommended method and its performance.