B565: Homework 4

1. Download the "naive_bayes_binary.csv" data from the course web site. These data are for a 3-class classification problem with 10 binary variables. The true class is the 11th column of the data file.

   (a) Using the first half of the data set, train a naive Bayes classifier.

   (b) Using the 2nd half of the data set, classify each vector and construct the confusion matrix. We we have C different classes, then the confusion matrix is the $C \times C$ matrix where the ij entry counts the number of times an observation from class i was classified as from class j.

2. Download the *Student Performance Data Set* at the UCI Machine Learning Repository,

   https://archive.ics.uci.edu/ml/datasets/Student+Performance

   We will use the math data.

   (a) Create a class variable for each student by testing if the final score "G3" satisfies $G3 > 10$ or $G3 \leq 10$. Create a decision tree predicting this class using all other variables *except* "G1" and "G2." Prune the tree to avoid overfitting and submit a plot of your tree.

   (b) For your pruned tree, what is the error rate on the training data and what is the estimated *generalization error*. That is, what would you predict the error rate to be on data different from your training data but from the same population.

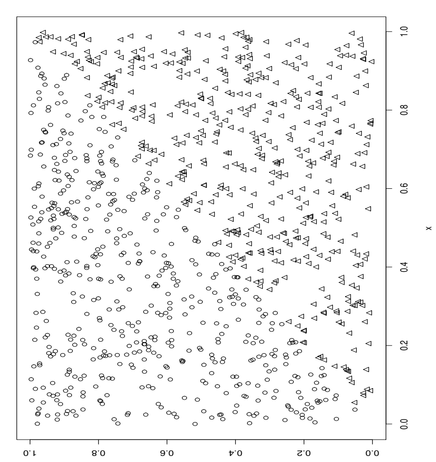
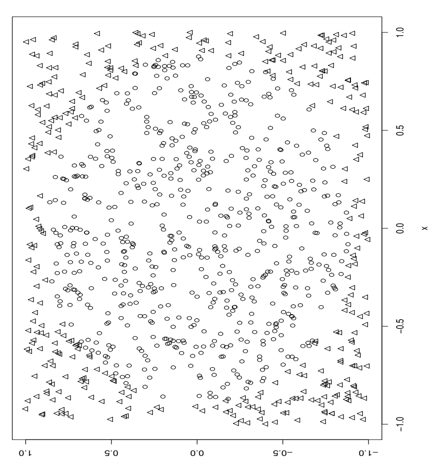
   (c) What is the most useful variable for prediction?

   (d) You can also use *rpart* to predict the score of a *continuous value*. That is, we can treat the problem as *regression* rather than classification. To do this with *rpart* just change the method to "anova" and use the original continuous "G3" variable. For regression the notion of "error rate" isn't meaningful since we are trying to predict a continuous value, so we use sum of squared errors (SSE) of the predition. That is, if y_i is the true value for the i the observation and $y(\hat{x_i})$ is our estimate of y, which depends on the features, then

   $$SSE = \sum_i (y_i - \hat{y}(x_i))^2$$

   Plot the tree and answer the same questions.

3. For the 2 cases below, find new features (functions of x and y) that would increase the efficiency of a classification tree.

4. Using the data in "strange_binary.csv," build a classification tree that distinguishes the "good" examples from the "bad" ones using no more than 3 splits.

   (a) Report the classification error rate on this training set. Is it reasonable to assume that your classification accuracy would be similar on test data from the same model?

(b) Introduce an additional feature that allows you to significantly decrease the error rate, still using only 3 splits. Report the training error rate for this new classifier. It should be possible to get about 80% correct on the training.

5. Jensen's inequality says that for a convex function, k, $E(k(X)) \geq k(E(X))$. Using the fact that $-\log$ is convex, it follows that

$$E(\log(X)) \leq \log(E(X))$$

(a) Use this inequality to show that the average entropy caused by a split is no greater than the original entropy. That is, if q_l and q_r are the proportions going to the left and right nodes and p, p_l, p_r are the class distributions at the original, left, and right nodes, then

$$q_l H(p_l) + q_r H(p_r) \leq H(p)$$

(b) Let C be the class of an example and T be the leaf node of the tree for that example, regarded both as random variables. Define the conditional entropy of the class given the tree, $H(C|T)$, to be

$$H(C|T) = \sum_t p_t H(C|T = t)$$

where p_t is the probability of reaching leaf node t and $H(C|T = t)$ is the entropy of the class distribution at leaf node t. Show that each split reduces $H(C|T)$. It is fine to think of all probabilities in this case as proportions.

(c) The joint entropy of the pair of random variables, (C, T), is defined to be $-\sum_{t,c} p_{t,c} \log p_{t,c}$. Show that

$$H(T, C) = H(T) + H(C|T)$$

This is a general fact about entropy or "information," not depending on the particular example of classification trees.

6. The "classification_accuracy.csv" table on canvas gives classification accuracy of 3 different classification techniques: decision trees, naive Bayes, and support vector machines. Compare each pair of techniques on each data set, deciding the comparision as a win, loss, or draw for the first technique of the pair. Produce a 3x3 table with rows labeled by the techniques and columns labeled by win/loss/draw, counting the number of data sets that fall into each cell.