```
CS578 Empirical Methods in Machine Learning and Data Mining
Course Project    Predictions Due Saturday December 6, 2003

The goal of this project is to apply decision trees, neural nets,
SVMs, and/or k-nearest neighbor to a data set using any/all of the
methods we studied in the course to improve performance.  These
include:

  - bagging and boosting
  - cross validation
  - model averaging (combining predictions from two or more
    models or learning methods)
  - early stopping
  - feature re-coding
  - feature selection
  - feature weighting
  - distance metric hacking
  - ...

You will be given two data files, a train set and a final test set.
The train set will contain 5,000 cases.  You can do anything you want
with the training set.  We strongly encourage you to use cross
validation to create your own test sets from this training set so that
you get unbiased estimates of the performance of the methods you try.
There are no missing values in the data.

The test set will not contain targets!  You will run your final model
on the test set and email the predictions to alexn@cs.cornell.edu.  We
will compute the performance of your method from these predictions and
use this as part of the grading.  The test set will contain 25,000
cases so that we can reliably estimate performance.

You may work on the project in groups of 1-4 students.  We encourage
you to work in groups of 3-4 because this will let you explore more
options.  If you work in a group, briefly document who does what.  For
example:

  "X was responsible for decision trees, implementing cross validation
  for all the experiments, and preprocessing the data.  Y did neural
  nets and k-NN and looked at feature weighting in k-NN (which helped,
  but not enough to make k-NN competitive with trees and neural
  nets). Z implemented feature selection and bagging, and generated
  most of the graphs in this report. We all decided how to do cross
  validation before starting the project, and which model performed
  best at the end of the project.

The project will be graded as follows:

50% TECHNICAL APPROACH:
    How well did you tackle the problem?
    What method(s) did you use to optimize performance?
    How well did you do them?
    How well did you interpret the results?
    The project is open ended and you are expected to think about
    how to find/train good models in the allotted time.  You can't
    try all possible combinations of methods.  It is important to
    create a plan for tackling the problem, and adjust the plan as
    you collect intermediate results.

25% WRITE-UP:
    Is your report clear, concise, and complete? The write-up should
    outline your plan for tackling the problem, and summarize the
    performance of all the models you trained.  The write-up should
    clearly state what model you think is best and how the final model
    is trained. You must include estimates of roughly how well you
    think the final model should perform on the final test set (based
    on the performance you observe on your test sets).  Unlike the
```



```
    homeworks, the quality of the final report is important.  Reports
    should be only as long as they need to be, and no longer.  Reports
    that seem unnecessarily long and rambling or that include a lot of
    uninteresting results will lose points.  Your graphs should be
    clean and to the point.  It's still fine to write by hand on your
    graphs, but you should think carefully about what you plot, how
    you plot it, the scales and labels on the axes, if the plot is
    intelligible, etc.  You will not get a chance to explain anything
    in your report to us later, so it is important your report be
    self contained and intelligible.

25% PERFORMANCE ON THE FINAL TEST SETS:
    We'll measure the accuracy, RMSE, and ROC Area of your predictions.
    Because a model that optimizes accuracy might not be optimal for
    ROC Area or RMSE, you will submit different predictions for
    accuracy, for RMS, and for ROC Area.  It is OK to if the predictions
    you submit for accuracy are the same as the ones you submit for
    RMSE and ROC Area.  (ROC will be explained in a few lectures, and
    we will provide you with c-code to calculate accuracy, RMSE, and
    ROCA so that everyone is calcualting performance the same way.)

    To submit your predictions, send one email to alexn@cs.cornell.edu
    with a subject line of "CS578 Predictions for Final Project". The
    email should have three attached files.  Each file should be named
    groupname.{acc|rms|roc}.  The groupname can be anything you want
    as long as it is unlikely to get duplicated.  Each file should
    observe the following format:

   - The 1st line of each file contains names of the project members.
   - The next line should be blank.
   - The next 25,000 lines should be the predictions we should use to
     compute accuracy, RMSE, or ROC Area. The predictions should be one
     entry per line: the probability your model predicts for class 1.

     IMPORTANT! You must return predictions to us in the same order as
     the cases in the unlabeled final test sets and it is critical that
     the format described above be followed exactly!

     Sample Email:
     ...
     Subject: CS578 Predictions for Final Project
     ...
     attachment: carnic.acc
     attachment: carnic.rms
     attachment: carnic.roc

     Sample File:

     Rich Caruana, Alex Niculescu, Radu Popovici, Casey Smith

     0.66
     0.09
     0.38
     ... 24,995 more of these
     0.99
     0.59

Comments about the format:
  - Probabilities can use any reasonable number of significant digits,
    but it's better if you use 6 or fewer figures to keep the files
    small.
  - The probability to give us is the probability the item is class 1!
  - The order of the predictions is critical!
```