**Final Project**

The project is to have some hands-on experience with machine learning techniques using the software WEKA for solving real-world data mining problems, as well as to improve your understanding of some of the algorithmic issues in designing / applying various machine learning algorithms. This is a team project. Each team has two people. (If you prefer to do it by yourself or have difficulty to find a teammate, please let me know.)

1. Browse the UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/>) and select four datasets with the following characteristics / constraints:
2. Default task: Classification. Try to limit yourself to binary classification problems (for easier comparison and discussion of performance evaluation).
3. Number of instances: avoid choosing datasets with very few instances. (You may have one dataset with less than 100 instances.) If your dataset size is between 100 and 1000 instances, you should do 10-fold cross validation. If it is larger than 1000 instances, use 2/3 for training and 1/3 for testing.
4. Number of attributes: You should have at least one data set with more than 10 attributes. If it contains more than 100 attributes (and a large number of instances), you may consider performing feature selection to further improve classification accuracy.
5. (Write a short program to) convert the chosen dataset into ARFF format that can be used by WEKA. You do not need to make it a very fancy program and are not required to submit this program. Alternatively, you can download some UCI datasets in ARFF format here: <http://www.cs.waikato.ac.nz/ml/weka/>. (Not sure if the download contains at least three datasets satisfying the criteria above.)
6. Apply at least three classification schemes (e.g., Decision Tree [trees/J48], KNN [lazy/IBk], Naïve

Bayes [bayes/NaiveBayes], Logistic Regression [functions/Logistc], SVM [functions/SMO], etc.) to each data set and evaluate / compare their performance. For SVM, you can choose from three types of kernels (linear kernel, polynomial kernel, and RBF kernel), and each one counts for one scheme. You are also welcome to try any algorithms in WEKA (such as Bagging and Boosting) that we did not cover in class, or implement some simple variations of the existing algorithms. In that case you need to briefly discuss the idea of the algorithm and relate it to one of the algorithms we have learned in class.

1. Do some experiments (on the Biggreen with command lines) for each classification scheme-dataset pair, include the following in your report:
2. Observe the output classification model (no model for KNN) and figure out its meaning (e.g. how are discrete/categorical attribute values handled in the given algorithm?)
3. How does the training accuracy / kappa / AUC compare with the testing accuracy? (Use flag –i to turn on outputting statistics.) Which accuracy metric may be more meaningful for the particular data set given its class distribution and the purpose of the application?
4. How does varying some of the parameters affect the results? (e.g. for decision tree, you may try to vary the confidenceFactor and minNumObj to control pruning; for SVM, you may try to vary c or some parameters for the kernel function. Do some research if you are not sure of the meaning of some the parameters.) Do the changes meet or not meet your expectations and why?
5. For each data set, compare the classification schemes for their performance using paired t-test (on your Laptop with Weka GUI Experimenter)
6. Decide on a set of parameters to be used for each classification scheme-data set pair based on the results above in step 4.
7. Run 10-fold cross validation (or split data into 2/3 for training and 1/3 for testing for large dataset) for at least 20 times.
8. Observe and collect the following information from the output, which you will need to discuss in the project report. Its performance on testing data (e.g., accuracy, kappa statistics and ROC area etc.). You do not need to present and compare all of them, as most likely they will be consistent. However, in some cases you may want to present more than one measurement when there is inconsistency or there is reason to choose one over the other. For example, if class distribution is extremely unbalanced, accuracy is less useful than kappa; AUC and accuracy may be inconsistent when comparing multiple algorithms; etc.
9. Compute the average and standard deviation of each of the measures mentioned above. Plot/tabulate and compare the performance of the three algorithms on each dataset. Discuss why an algorithm worked significantly better / worse than the other algorithms on each particular dataset, or discuss why all algorithms did poorly and what aspects of the data mining process could be improved to get a better model (e.g., size and quality of samples? Data transformation? Feature selection?).

Project Report: write a report to summarize your experimental results. Your report must be between 5-10 pages excluding references. Single spaced. Font size 11-12. One-inch margin all around.

Your report needs to have the following sections:

1. A short introduction describing the purpose of the study and some rationale for your choice of the datasets / algorithms. (Approximately 0.5-1 page.)
2. Data set description and characteristics (including information such as classification task, data sources, dataset sizes, types of attributes, missing values, and how those were handled by each algorithm.) Discuss any other relevant data issues, for example, class distribution, data normalization / standardization, etc. Tables and plots can be included (or preferred). (~1 page).
3. Results and Discussion (~4 pages).

a. One subsection devoted to the discussion of the results from step 4.

b. One subsection devoted to the discussion of the results from step 5.

1. Conclusions (0.5 – 1 page).
2. References (for data set, algorithms, data analysis techniques, etc.)

Presentation at Dead Week

Each team should present their work to the class. Each student is required to present its own part of work as a team.