University of Ioannina,
Computer Science Department

Assignment 4

The deadline for the assignment is **December 27**. The assignments should be submitted either via turn-in, or via email. The details for the turn-in are on the web page of the course.

Question 1

In this question you will experiment with classification in WEKA. You will experiment with three different datasets, the Iris dataset, the Mushroom dataset, and the Spambase dataset, which you can get from the class web page. You can find more details for each dataset on the UCI repository page. You will experiment with three different classification algorithms of WEKA: Naïve Bayes, Logistic Regression, and SimpleCart decision tree. For each algorithm and each dataset, you will do 10-fold cross validation, and you will submit the summary produced by WEKA at the end. For the decision tree, you will also submit the tree produced by WEKA.

Together with the results you will also submit a short report discussing the results: how hard the classification problem for each dataset is, and which algorithm performs best. Examine the produced decision trees, and discuss which attributes seem to be the most important in the classification.

Question 2

For this question you will create a classifier for spam email. You will use the SpanAssassin dataset which you can find online (the link is on the class web page). To train the classifier, you need training examples which belong to the positive class (spam), and the negative class (non-spam), which you will download from the SpamAssasin site, and you will combine them into a single dataset (of at least 2000 examples). This dataset you will split into training (80%) and testing (20%) subsets.

Each example is the text of an email in a separate file. From this text, you will create features for your classifier. The features can be anything that you think can help in discriminating between spam and non-spam emails. For example, possible features are all the words used in the emails, or the frequent words, or the words in the subjects, or the size of the email, or the domain of the sender, etc. Combinations of the above are also possible. Your classifier should have at least 10 features (if you use the words as features it will be a lot more). Each email becomes now a vector in the feature space.

Once you decide on the features you will extract them from the text, and you will create the input for training and testing. For the classification algorithm, you can implement the Naïve Bayes Classifier (it works better with categorical features), or use the LibLinear package (the link is on the class web page) which implements Logistic Regression SVM classification (you would need to transform the input to what the program needs). After training, run the classifier you produced over the test set.

You should submit a report that explains the procedure above: What data you picked, how you split it into training and test subsets, what features you created, what algorithm you used, and the results of classifier on the test set. In the discussion of the results give the accuracy of the classifier, and the confusion matrix. Submit also your code and the input and output files, if you have changed their format.

Bonus: Download 50 spam, and 50 regular emails from your personal email and run your algorithm on these emails. Report the accuracy of the algorithm and the confusion table. Submit also the input emails.