CS 640
Scientific programming in Python, use of Weka
Relation of SNPs to disease

Single Nucleotide Polymorphisms (SNPs) may be quantified as predictor variables in order to look for associations with a given phenotype, such as a disease. Investigate the dataset disease.arff containing genomic data on 2000 individuals, some of whom have the disease and some of whom do not. Each row contains data on one individual. There are 10 predictor variables: these are SNPs encoded as number of mutated alleles for the gene. The class variable indicates for each person whether they have disease (1 if they have it, zero if not). The dataset, disease.arff, may be found on Canvas.

Explore and analyze the data using Python (Numpy, Scikit-learn, Pandas, Matplotlib). Train several models to classify the examples as described below. Also use the Weka tool to classify the data using a neural network. For each model you train, be sure to evaluate its performance. Include prediction accuracy, confusion matrix and area under the ROC curve (AUC).

Write a report in which you compare your model results and summarize your conclusions about the relationship of the SNPs to the disease. Your report should include plots that convey much of the information. The text you write for the report should be brief. You can paste relevant outputs into your report (ex: summary information and confusion matrix) and refer to them to support your conclusions. Use an appendix for lengthy outputs. You may create a jupyter notebook for this assignment. Alternatively you may write a python script and submit your report in Word, pdf, or html format.

Please complete the following tasks:

- Preprocess the data as you see fit (ex: R or Python).
- Use Python to explore the data. Produce summary statistics and a histogram of each variable.
- Use a random forest in Python to rank the variables by their importance in classifying disease.
- Construct models in Python to classify the examples by disease status. Use a support vector machine (SVM) and two different kernels (rbf and polynomial). Use cross-validation grid search to select the best C and gamma to train the model with an rbf kernel. Do your best to use cross-validation grid search to select the best C and degree for polynomial kernel. (Ex: you may limit the degree and engage in 2-step process, first using only a few values for C).
- Use a regression method in Python to model disease using the predictor variables. Produce plot(s) to visualize your results.
- Use a neural network in Weka to classify the examples by disease status. Specify parameters such as the number of hidden nodes, learning rate, momentum and training time.

Notes:

- scikit-learn documentation on SVM: http://scikit-learn.org/stable/modules/svm.html
- scikit-learn documentation on random forest: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html