IE 582 Statistical Learning for Data Mining
Homework 4, due December 14th, 2018

Instructions: Please solve the following exercises using R (http://www.r-project.org/). Also, you are requested to use certain R packages for this particular homework. You are expected to use GitHub Classroom and present your work as an html file (i.e. web page) on your progress journals (as you did for Homework 0).

Do not share your code (except the one in your progress journals)! As a fundamental principle for any educational institution, academic integrity is highly valued and seriously regarded at Boğaziçi University.

The aim of this homework is to compare the performance of penalized regression approaches, decision trees, support vector machines and tree-based ensembles.

1- You are asked to find 4 datasets for a classification task from different domains. UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) is a very good source to find such datasets. The datasets are required to have certain characteristics such as:

All of them should have
- A separate labeled test data (if there is no test data, please generate your test from the training data if you have reasonable number of training instances)
- Number of training samples and test samples that is larger than 200 (so that the comparison makes sense)
- More than 20 features
- A brief description of the task and features (UCI machine learning repository has nice examples)

At least
- One of them is regression problem
- One of them should be multi-class classification problem
- One of them has a class imbalance problem (a ratio of 3:1 will be enough)
- Two of them should have more than 50 features
- One of them has some number of categorical or ordinal features (i.e. not all numerical features)

2- Below is the specifications for the algorithms to use.

a. Penalized Regression Approaches (PRA): Use penalized regression approaches with lasso penalty. Parameter to be tuned is lambda in this case.

b. Decision Trees (DT): Use classification and regression trees (CART) for training. We are mainly interested in the depth of the tree since it controls the complexity. There are several options to control the depth of the tree but we will use only two criteria. They are "the minimal number of observations per tree leaf" and "complexity parameter". We assume that we do not consider any type of pruning (i.e. post-, pre-).

c. Support Vector Machines (SVM): Use SVM for training. There are two parameters of SVM: penalty parameter (referred to as *C*) and kernel type. We are mainly interested in two kernel types in this homework: *polynomial* and *Gaussian* (radial basis function kernel). Polynomial kernel has a degree parameter where Gaussian kernel has the bandwidth parameter (referred to as gamma or sigma as discussed in the class).

d. Random Forests (RF): Use Random Forests for training. In random forests, *J* trees are fit to bootstrap samples using a random sample of *m* features on which to split each node. Each tree is basically a classification and regression tree however the data used to train each tree is a random subsample of the whole training data (in general 2/3 is the preferred ratio for random selection). The second difference is that not all features are evaluated at each split decision. A random selection of *m* (which is smaller than the total number of features) is chosen independently for each node, and the best split for the selected predictors is used to split the node, where "best" is determined as for CART. The trees are grown large, and not pruned. Assume that we grow trees until we achieve "the minimal number of observations per tree leaf". In general, this parameter is set small enough to avoid underfitting. Assume that this value is set to 5 for this classifier (which is common in practice).

e. Stochastic Gradient Boosting (SGB): Use Gradient Boosted Trees for training. In this approach, we are mainly interested in "depth of the tree", "learning rate" (also known as shrinkage) and "number of trees". Inherited from the tree base learning, there is also "the minimal number of observations per tree leaf*".* In general, this parameter is set small enough to avoid underfitting. Assume that this value is set to 10 for this classifier (which is common in practice).

3- Specify the best set of parameters for each algorithm in item 2 based on cross-validation. Please try at least three at most six different levels for each parameter. To summarize:

a. For PRA: l_1 penalty, lambda.
b. For DT: the minimal number of observations per tree leaf and complexity parameter
c. For SVM: *C*, kernel type (as well as kernel parameters). Note that kernel types to be considered are given (you have only 2 levels).
d. For RF: only *m* (set other parameters as *J=500* and the minimal number of observations per tree leaf=5)
e. For SGB: depth, learning rate, number of trees.

4- Summarize the performance of the algorithms based on the cross-validation error on the training data. You are free to select the performance metric(s) you think useful. Use the best set of parameters to classify the test data. Compare and comment on the results. Some possible comments may answer the following questions:

a. Is the cross-validation error rate of different approaches consistent with the test error rate?
b. What is your observation about the performance of the classifiers over all datasets?
c. How would you compare training and test error? Is there any indication of underfitting or overfitting?
d. …
e. …