IE 582 Statistical Learning for Data Mining
Homework 4, due December 16th, 2019

Instructions: Please solve the following exercises using R (http://www.r-project.org/) or Python (https://www.python.org/). You are expected to use GitHub Classroom and present your work as an html file (i.e. web page) on your progress journals. There are alternative ways to generate an html page for you work:

- A Jupyter Notebook including your codes and comments. This works for R and Python, to enable using R scripts in notebooks, please check:
  - https://docs.anaconda.com/anaconda/navigator/tutorials/r-lang/
  - https://medium.com/@kyleake/how-to-install-r-in-jupyter-with-irkernel-in-3-steps-917519326e41

  Things are little easier if you install Anaconda (https://www.anaconda.com/). Please export your work to an html file. Please provide your *. ipynb file in your repository and a link to this file in your html report will help us a lot.

- A Markdown html document. This can be created using RMarkdown for R and Python-Markdown for Python

Note that html pages are just to describe how you approach to the exercises in the homework. They should include your codes. You are also required to provide your R/Python codes separately in the repository so that anybody can run it with minimal change in the code. This can be presented as the script file itself or your notebook file (the one with *.ipynb file extension).

The last and the most important thing to mention is that academic integrity is expected! Do not share your code (except the one in your progress journals). You are always free to discuss about tasks but your work must be implemented by yourself. 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 and tree-based ensembles using the dataset you are working on for the project. Moreover, I would like you to think about possible features

1- Suppose we have two tasks for this task. One of them is to predict if total goals in a match will be larger than 2.5 where the other is to predict the total number of goals (i.e. the former is a classification problem where the latter is a regression one). Please extract relevant features other than odds (i.e. team average goals in last 5 games, average number of goals for home team in last 3 games, total points in the last 3 games and etc.,). You can still use the odds but some odds may be missing for certain games. You are expected to be creative in this statistical modeling part. Some people liked to state this iteration as "Feature Engineering". Please describe the features clearly and elaborate on your intent for extracting it. Use all the possible matches you have as instances.

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. 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).

d. 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 RF: only *m* (set other parameters as *J=500* and the minimal number of observations per tree leaf=5)
- **d.** 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.** …