IE 582 Statistical Learning for Data Mining
Homework 2
Due December 18, 2023

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) 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-steps917519326e41

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.

b) 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.

Please mention if you utilize Large Language Models (i.e. chatGPT, BART and etc). You are expected to provide your prompt and resulting response of these models at the end of your reports as Appendix if you have utilized any of these tools.

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 nearest neighbor approaches, decision trees and tree-based ensembles.

1- You are asked to find 5 datasets for a classification task from different domains. The datasets are required to have certain characteristics such as:

All of them should have
- A separate labeled test data
- 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 should be a regression problem
- Two of them should be multi-class classification problem
- One of them has a class imbalance problem (a ratio of 2:1 will be enough)
- Two of them should have more than 100 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. Nearest Neighbor (NN) Approaches: Use NN for prediction. There are two parameters to consider: the distance measure and the number of neighbors. We are mainly interested in two distance measures in this homework: *Euclidean and Manhattan Distance.*

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 one criterion. It is "the minimal number of observations per tree leaf". This criterion is implemented by all the software mentioned above. We assume that we do not consider any type of pruning (i.e. post-, pre-).

c. Random Forests (RF): Use Breiman's (2011) implementation. We are only interested in the effect of the ratio of the number of features evaluated at each split.

d. Gradient Boosted Trees (GBT): There are many boosting implementations (i.e. standart gradient boosting covered in the class, lightgbm and etc.). Please specify the implementation you choose. We are mainly interested in tuning the depth, the number of trees and the learning rate. You can keep the other parameters as default in your trials.

3- Specify the best set of parameters for each algorithm in item 2 based on cross-validation. For continuous parameters, evaluate at least 5 alternatives (in a reasonable range). To summarize:

a. For NN: Distance measure and the number of neighbors.

b. For DT: only the minimal number of observations per tree leaf. Set complexity parameter to zero and minimum number of observations to split as the twice as the minimal number of observations per tree leaf.

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

d. For GBT: The depth, the number of trees and the learning rate.

4- Summarize the performance of the algorithms based on the cross-validation error on the training data. 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 learners over all datasets?

c. …

d. …

e. …