CS 5661: Topics in Data Science
Homework1, Due Date: Mon, Feb 26

Instructor: Dr. Mohammad Pourhomayoun

Cancer Diagnosis Using Machine Learning

Write and submit your python codes in "Jupyter Notebook" to perform the following tasks. Make sure to provide proper descriptions as MarkDown for each section of your code (each section of the code must have a short meaningful description right before that, describing what this part of the code is supposed to do!).

In this homework, we work with a real dataset from UCI Dataset.

a- Read the dataset file "Cancer.csv" (you should download it from CSNS), and store it in a Pandas DataFrame. Check out the dataset. As you see, the dataset includes 9 numerical features. The last column is the binary label ("1" means it is a malignant cancer, "0" means it is a benign tumor). You will use all 9 features in this homework.

b- Use sklearn functions (see class tutorials for details) to split the dataset into testing and training sets with the following parameters: **test_size=0.3, random_state=2**.

c- Use "Decision Tree Classifier" to predict Cancer based on the training/testing datasets that you built in part (b). Then, calculate and report the accuracy of your classifier. Use this command to define your tree:
my_DecisionTree = DecisionTreeClassifier(random_state=2).

d- Now, we want to perform "Bagging" based on 19 "base decision tree classifiers".
Note: you should write your own code to perform Bagging (don't use scikit-learn functions for Bagging!)
To do so, you need to perform bootstrapping first. You can write a "for" loop with loop variable i=0…18.
In each iteration of the loop, you have to:
- make a bootstarp sample of the original "Training" Dataset (build in part(b)) with the size of **bootstarp_size = 0.8*(Size of the original dataset).** You can use the following command to generate a random bootstrap dataset ("*i*" is the variable of the loop, so the random_state changes in each iteration):
resample(X_train, n_samples = bootstarp_size , random_state=*i* , replace = True)
- Define and train a new base decision tree classifier on this dataset in each iteration:
Base_DecisionTree = DecisionTreeClassifier(random_state=2).
- Test "this base classifier" on the original "Testing" Dataset build in part(b), and save the prediction results for all testing samples.
- Perform Voting to make the final decision on each data sample based on the votes of all 19 classifiers.

Finally, calculate and report the accuracy of your Bagging method.

e- Use scikit-learn “Adaboost” classifier to predict Cancer based on the training/testing datasets that you built in part (b). Then, calculate and report the accuracy of your classifier. Use this command to import and define your classifier:
from sklearn.ensemble import AdaBoostClassifier
my_AdaBoost = AdaBoostClassifier(n_estimators = 19,random_state=2)

f- Use scikit-learn “Random Forest” classifier to predict Cancer based on the training/testing datasets that you built in part (b). Then, calculate and report the accuracy of your classifier. Use this command to import and define your classifier:
from sklearn.ensemble import RandomForestClassifier
my_RandomForest =
RandomForestClassifier(n_estimators = 19, bootstrap = True, random_state=2)