

## Homework 2 (100 points)

### Ensemble Learning

***In this homework, you will solve the same problem as homework 1 using a different machine learning technique. Upload a .txt file with a link to your file as your submission on Submittity.***

***You need to perform the following tasks for this homework:***

**Task 1 (30 points):** Implement a Decision Tree Classifier for your classification problem. You may use a built-in package to implement your classifier. Try modifying one or more of the input parameters and describe what changes you notice in your results. Clearly describe how these factors are affecting your output.

**Task 2 (30 points):** From the Bagging and Boosting ensemble methods pick any one algorithm from each category. Implement both the algorithms using the same data. Use k-fold cross validation to find the effectiveness of both the models. Comment on the difference/similarity of the results.

**Task 3 (40 points):** Compare the effectiveness of the three models implemented above. Clearly describe the metric you are using for comparison. Describe (with examples) Why is this metric(metrics) suited/appropriate for the problem at hand? How would a choice of a different metric impact your results? Can you demonstrate that?

**The following task is for Graduate level only (6000 level):** This task is more open ended and emphasizes the research aspect of implementing a model. You will be exploring the impact of hyperparameter tuning which we haven't discussed in detail so far.

**Task (50 points):** For the same classification problem solved above, implement the XGBoost algorithm. If you picked XGBoost as one of the boosting algorithms in task 2, you may use the same implementation. Picking the best hyper-parameters is an important task for any Machine Learning implementation. In this task create an ***experiment*** that explores the impact of any 3 input parameters on your result. Describe the parameters in detail. Also describe how you compare the results.

An experiment can be defined as a systematic way of picking parameter values. This could be something that you come up with yourself or you may refer to the exiting literature on design of experiments for hyperparameter tuning. See:

<https://www.jeremyjordan.me/hyperparameter-tuning/>

This task will require you to perform some research into this open-source library yourself. A good place to start is here: <https://xgboost.readthedocs.io/en/latest/parameter.html#general-parameters>