Indian Institute of Technology Delhi
Department of Mathematics
II Semester 2023-2024
Assignment
Weightage 12% Due Date 15Th March 2024

You need to form a group of 3 members and jointly complete the assignment using Python Google Colab.

Q1. In this assignment, you will experiment with TensorFlow's Decision Forest (TF-DF) library to train, test and interpret **Random Forests** and **Gradient Boosted Trees** in TensorFlow running over the **Google colab**.

A Random Forest is a Collection of deep CART decision trees trained independently and without pruning. It uses bagging technique to train each Decision Tree (DT) on a random subset of the original training dataset (sampled with replacement). Whereas, in Gradient boosting, weak learners (DTs) are trained sequentially with each successive model trying to improve on the error from the previous model by assigning more weight to the incorrect predictions. These models are robust to Overfitting.

A decision tree decides on the best split feature by evaluating different features and determining which one, along with an associated threshold, maximally reduces impurity in the resulting subsets. As we discussed in the class details about the impurity measures like Entropy, I.G, Gini Index, and Gain ratio, in this assignment you will implement CART algorithm that uses Gini Index as the metric for deciding the best split.

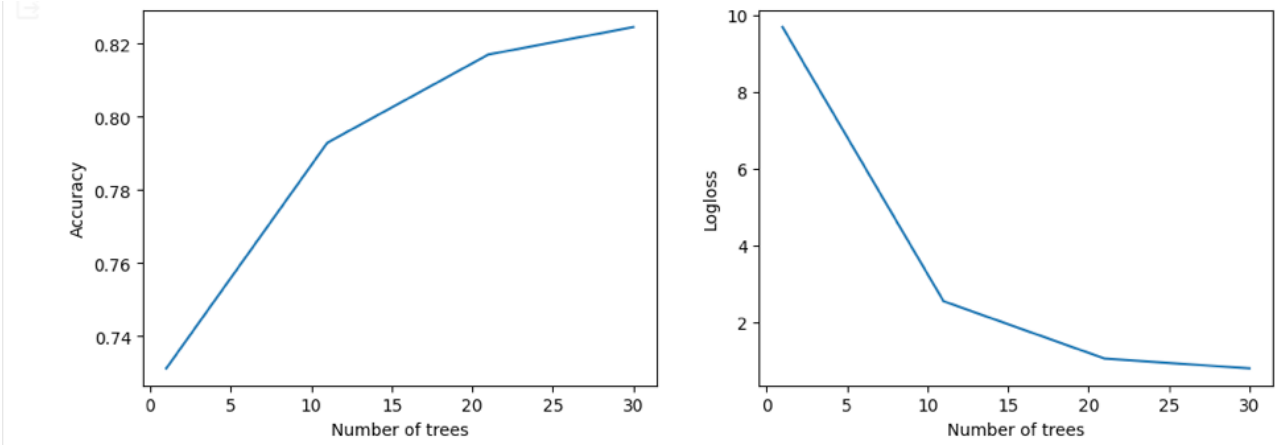
You are provided with a set of data items (Data-RF.xlsx) listing the scores of students in various components of Instructor's one of the earlier course offerings. Students were graded A, A-, B, …E out of a total score of 144 (for Mid-semester grading). For privacy reasons the ID numbers and names of students are removed from the original file. While TensorFlow's Decision Forest library supports both classification and regression tasks, here you need to implement a classifier using TensorFlow's Decision Forest library as discussed in the class. The class label represents the Grade in the course (last column).

Tasks:

- Load the provided excel file into the code converting it into a Pandas dataframe. Since our target column is 'Grade', we need to map the categorical values to numerical values, the reason being that the label column for classification gets mapped to numerical values when converting to TensorFlow datasets. As you can see Attendance and Grades are Ordinal data, therefore you might need to perform label encoding. However, Tensorflow decision forests natively handle numerical and categorical data and therefore no

encoding would be required. You can check this by training the decision tree with and without this preprocessing and then compare their accuracies.

- Next, split the dataset into training and testing using 70-30 rule. Train a random forest model using TensorFlow's decision forest, fit the model using training dataset generated earlier and evaluate the model using test set.
- Visualize the first tree in the trained
- Implement the Gradient boosted decision trees and compare their accuracies with the Random forest implementation having 30 DTs (RF: accuracy and log loss as shown below).



- Compare and contrast the training and testing accuracies of any one of these two types (RF or GBDT) and find out the number of trees and maximum depth hyper-parameters for a reasonable accuracy say, 85% and above. In the process find out if increasing n_trees hyper-parameter leads to more robust and accurate trees. Find out what impact a smaller and a larger value of max_depth hyper-parameter has on the performance of the model.