Problem 1: Annual Rainfall Prediction [20 marks]

Download the attached dataset called “annualrainfall.mat”. You can load it in Python using “loadmat” function and access its field. It contains a single matrix called XR that is 357x118. Each row refers to a location in India and each column refers to a year. Carry out the following steps:

1) For each year, calculate the total rainfall over all the locations. Calculate the mean (m) and standard deviation (s) of this quantity over all the years (columns). [2 marks]
2) Assign labels to each year according to the rule: if the total rainfall in any year is more than m+s, the label should be +1, if the total rainfall in any year is below m-s, the label should be -1, else the label should be 0. [1 mark]

3) For each year, try to predict the label of each year from the rainfall values at the different locations, using a Decision Tree of depth 10. Note which locations are chosen. Use the first 100 years for training using 5-fold cross validation, and the remaining 18 years for testing. Plot both training and testing errors. [4 marks]

4) Repeat the same analysis using a Random Forest. Do the results improve? [3 marks]

5) Now, for each location, define the mean and standard deviation of annual rainfall. Attach a label +1/0/-1 to each location, for each year, according to the rule in (2). [2 marks]

6) Use a Decision Tree of depth 10 to predict the label at each location on any given year, using the labels at the remaining locations in the same year. For each location, identify the top 10 predictor regions. Once again, use the first 100 years for training with 5-fold cross-validation, and the remaining years for testing. [6 marks]

7) Repeat (6) with random forest. [2 marks]

Problem 2: Handwritten Digit Prediction [10 marks]

Download the MNIST dataset of handwritten digits 0-9. In each image try to predict the label (0-9) using the pixel values. Train a decision tree of depth 20 and a random forest for this purpose. If the accuracy is not enough, increase the depth till 50. For each image class, use 75% images for training and the rest for testing. Use 10-fold cross validation.