Intelligent Data Analysis – Fall 2015
Homework #2
Due Date: Sept 24th, 2015

This homework uses a standard dataset available from UCI Machine Learning Data Repository. The dataset is called "*MAGIC Gamma Telescope Dataset.*" Details of this dataset along with the data can be found at: https://archive.ics.uci.edu/ml/datasets/MAGIC+Gamma+Telescope
The dataset is produced from a telescope by recording details of 19020 incident particles' flashes. An excel file containing the dataset is attached with this homework assignment. Ten features of each particle flash are recorded and these are stored in the first ten columns of the dataset. Names of these features are available at the above mentioned website. The eleventh column describes the class of each observed particle: a Gamma particle (class 1) or background noise (class 0). Your task is to create decision trees from this data by following the steps described below. Each step also describes the results that should be submitted. The goal of the decision tree model is to predict whether an observed particle flash is of a Gamma particle or of background noise.

1. Split the dataset into three parts by random selection: 13020 records for training, 3000 records for validation and tuning, and 3000 records for testing. You can read the attached .xlsx file by using the Matlab command: *data=xlsread('Magic04.xlsx');*
2. Split the training data into two tables: the first ten columns contained in a "Features" table and the last one column contained in the "ClassLabels" vector.
3. In Matlab environment, use fitctree command to generate decision tree (*dtr*) as follows: *dtr=fitctree(Features, ClassLabels, 'MinLeafSize', <N>);* In this command *MinLeafSize* parameter specifies the minimum number of records, '*N*', in each leaf node of the generated decision tree. If *N* is set to a high number, say 1200, then a node will not be split to grow the tree if its splitting results in a child node containing fewer than 1200 records. Therefore, high values of N will result in shallower trees and small values of N will result in deeper trees. You can view the generated decision trees by using the commands view(dtr) or view(dtr, 'Mode', 'graph').
4. A generated decision tree can be used to find the predicted class labels as follows. Create the 3000 by 10 matrix of features (say, it is called *TestFeatures*) from the validation or test data partition. Exclude the class labels from this table. Then use the command: *PredictLabels = predict(dtr, TestFeatures);* The predicted class labels can be compared to the original class labels to determine accuracy, precision, and recall values.
5. Generate decision tree from training data such that no leaf node has fewer than 1000 records.
    a. For this tree **submit** the graphical view of the decision tree.
    b. Test the tree against the training data itself and find the predicted labels. Compare the actual and the predicted labels and determine the accuracy, precision, and recall values. **Submit** these results along with the table containing the numbers of TP, FN, FP, and TN counts.
    c. Repeat the past (b) above for the 3000 records of the validation data partition.
6. Repeat #5 above for the case in which no leaf node has fewer than 20 records. **Submit** the results similar to those described in #5.
7. Find the accuracy of the training data and the validation data for the following values of *N* (minimum number of records at leaf nodes): 1000, 750, 500, 250, 125, 100, 50, 20, 10, 5. Plot these accuracy values on a single plot. Also plot the number of nodes in the decision tree for each of the above values of *N*. **Submit** both these plots along with your interpretation of these results.
8. As per the results obtained in #7 above which decision tree is the best model for the training data? How many nodes does it have? Find the accuracy, precision, and recall when this decision

tree model is tested against the 3000 records in the test partition of the dataset. **Submit** all these answers/results.