**CMPE 480 Project 3**

**Decision Tree Implementation**

Seyfi Kutay Kılıç

2015400042

1. **Project Description**

The aim of the project is to develop a decision tree learning algorithm to classify different types of flowers by using a data which contains numerical attributes such as length and width of different parts of the flowers.

Also, we are expected to report/plot statistics such as: Training and validation loss with respect to the depth of the trees during training, loss in the test data for each iteration; also giving the means and variances of two different entropy metrics which are Information Gain and Gini Impurity.

We are given a data set which consisting of some flower (iris) data. The given data files which has lines in the following format:

<sepal\_length>,<sepal\_width>,<petal\_length>,<petal\_width>,<class>

where <class> has the following values: “Iris Setosa”, “Iris Versicolour”, and “Iris Virginica”.

1. **Software Design**

My implementation relies on binary decision tree as the learning model.

While it creating the decision tree it takes the training and validation set. Then it starts to split the nodes in breadth first manner. It decides how to split by calculating the minimal splitting entropy by trying all the attributes and their values. The decision tree calculates the entropy by using the injected entropy calculator function (Information Gain or Gini Impurity). Also, it decides to whether split a node by using the following algorithm: It first split the node with the best split and computes the error rate in the validation set and if the new validation error rate is greater than the previous then it recovers the split.

**The design of the code is as follows:**

Global Functions:

main(): It is the main function of the program. It reads the data from the given data file. Then starts to build the decision trees by using 10 iterations (At each iteration the data is shuffled). After that, it calls the functions of Output Utility to print and plot the necessary statistics.

readData(): It reads the data from the given file and convert them to a list of datums.

shuffleAndSplitToTrainingValidationTest(): It takes the datum array and shuffle then split into 3 sets (training, validation, test) with 20%, 40%, 40% proportions respectively.

DecisionTree Class:

It creates the nodes in a breadth first manner. It splits the nodes until the error in the validation set saturated. Also, it stores the training and validation losses at each depth of the tree so that they can be used.

DecisionTreeNode Class:

It is an extended Binary Tree Node which has pointers to left and right children; the plurality class; and the splitting information.

DecisionTreeEvaluator Class:

It is a utility class that contains functions to compute the error rate of a decision by using a data set.

EntropyCalculator Class:

It is a utility class that contains the calculators for Information Gain and Gini Impurity.

OutputUtility Class:

It is a utility class that contains the functions to print and plot the needed statistics.

1. **How to Run the Code**

First of all, if you do not have Python 3.73 or the library “matplotlib then you need to install them. After that you can run the code by typing:

python3 decision\_tree.py <data\_file>

For example: python3 decision\_tree.py iris.data

1. **Loss Rates of Test Data**

The loss rates of the test data for the iterations is as follows:

The loss percentage of information gain technique in iteration 1 is 5.0%

The loss percentage of information gain technique in iteration 2 is 11.666666666666666%

The loss percentage of information gain technique in iteration 3 is 8.333333333333332%

The loss percentage of information gain technique in iteration 4 is 13.333333333333334%

The loss percentage of information gain technique in iteration 5 is 10.0%

The loss percentage of information gain technique in iteration 6 is 11.666666666666666%

The loss percentage of information gain technique in iteration 7 is 11.666666666666666%

The loss percentage of information gain technique in iteration 8 is 3.3333333333333335%

The loss percentage of information gain technique in iteration 9 is 1.6666666666666667%

The loss percentage of information gain technique in iteration 10 is 8.333333333333332%

The loss percentage of gini impurity technique in iteration 1 is 5.0%

The loss percentage of gini impurity technique in iteration 2 is 15.0%

The loss percentage of gini impurity technique in iteration 3 is 5.0%

The loss percentage of gini impurity technique in iteration 4 is 11.666666666666666%

The loss percentage of gini impurity technique in iteration 5 is 13.333333333333334%

The loss percentage of gini impurity technique in iteration 6 is 16.666666666666664%

The loss percentage of gini impurity technique in iteration 7 is 6.666666666666667%

The loss percentage of gini impurity technique in iteration 8 is 5.0%

The loss percentage of gini impurity technique in iteration 9 is 10.0%

The loss percentage of gini impurity technique in iteration 10 is 10.0%

The mean of the loss with using information gain is 8.5%

The mean of the loss with using gini impurity is 9.833333333333334%

The variance of the loss with using information gain is 15.709876543209875%

The variance of the loss with using gini impurity is 18.796296296296294%

1. **Loss Rates With Respect to Depth of the Trees**