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”.

2. 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.

3. 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

4. Loss Rates of Test Data

The loss rates and their mean/variance of test data for the iterations is as follows:

The loss percentage of information gain technique in iteration 1 is 15.0%
The loss percentage of information gain technique in iteration 2 is 6.666666666666667%
The loss percentage of information gain technique in iteration 3 is 8.333333333333332%
The loss percentage of information gain technique in iteration 4 is 6.666666666666667%
The loss percentage of information gain technique in iteration 5 is 5.0%
The loss percentage of information gain technique in iteration 6 is 6.666666666666667%
The loss percentage of information gain technique in iteration 7 is 11.666666666666666%
The loss percentage of information gain technique in iteration 8 is 8.333333333333332%
The loss percentage of information gain technique in iteration 9 is 8.333333333333332%
The loss percentage of information gain technique in iteration 10 is 3.3333333333333335%
The loss percentage of gini impurity technique in iteration 1 is 15.0%
The loss percentage of gini impurity technique in iteration 2 is 6.666666666666667%
The loss percentage of gini impurity technique in iteration 3 is 6.666666666666667%
The loss percentage of gini impurity technique in iteration 4 is 6.666666666666667%
The loss percentage of gini impurity technique in iteration 5 is 5.0%
The loss percentage of gini impurity technique in iteration 6 is 6.666666666666667%
The loss percentage of gini impurity technique in iteration 7 is 11.666666666666666%
The loss percentage of gini impurity technique in iteration 8 is 8.333333333333332%
The loss percentage of gini impurity technique in iteration 9 is 8.333333333333332%

The loss percentage of gini impurity technique in iteration 10 is 3.3333333333333335%

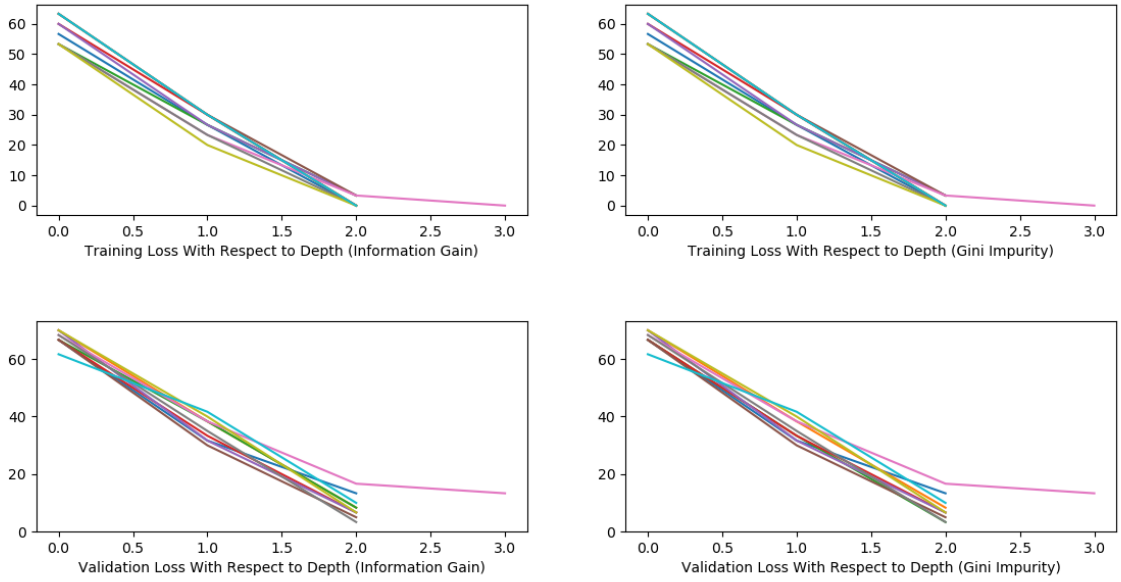
The mean of the loss with using information gain is 8.0%

The mean of the loss with using gini impurity is 7.833333333333333%

The variance of the loss with using information gain is 10.987654320987653%

The variance of the loss with using gini impurity is 11.141975308641975%

5. Loss Rates With Respect to Depth of the Trees



6. Conclusion

I have run my program so many times and noticed that the average error rate of test set is generally between 8% to 10%. Also the depth of the tree became at most 3 and often 2 which means that the intervals of the attributes of a class is not very distributed. For example, Iris Setosa has low petal width, Iris Versicolor has medium petal width, and Iris Virginica has high petal width; in general.

Furthermore, I compared the performance of Information Gain and Gini Impurity. I noticed that they perform very similar for the given, on average. However, Gini Impurity performed a little bit better for some iterations. I think that the both the formulas perform similar because they often give very similar results when you scale them to 1. Nevertheless, Gini Impurity is a little bit computationally efficient because the log operation in Information Gain is computationally expensive.