**CLASSIFICATION REPORT**

**Group NO: 55**

**21CS10086 : Swapnil Ghsoh**

**23MT61R15 : Vivekananda Giri**

**Project Number : BSDT**

**Project Name : Predicting the Position of Balance Scale**

**Introduction:**

The primary objective of the problem statement is to develop a decision tree learning model for the Predicting the Position of Balance Scale according to important characteristics by using one of the two decision tree learning approaches.

1. utilising information gained through entropy

2. utilising criteria for attribute selection based on the Gini index.

The assignment also includes applying pruning techniques for decision trees in order to reduce overfitting. The final objective is to evaluate decision tree models' performance both before and after pruning, and to compare these outcomes with the outcomes of an earlier decision tree learning algorithm from a package like sci-kit learn.

**Dataset Description:**

The attribute Information is given as follows.

● right-distance: categorial - {1, 2, 3, 4, 5}

● left-distance: categorial - {1, 2, 3, 4, 5}

● right-weight: categorial - {1, 2, 3, 4, 5}

● left-weight: categorial - {1, 2, 3, 4, 5}

Output Classes: L, R, B

Total number of data rows = 625

Total number of data columns = 5

Column1 - describe on output class

Column2 - describe on left distance

Column3 - describe on left weight

Column4 - describe on right distance

Column5 - describe on right weight

If (right-distance \* right-weight) = (left-distance \* left-weight), then output should be Balance(‘B’)

If (right-distance \* right-weight) > (left-distance \* left-weight), then output should be Balance(‘R’)

If (right-distance \* right-weight) < (left-distance \* left-weight), then output should be Balance(‘L’)

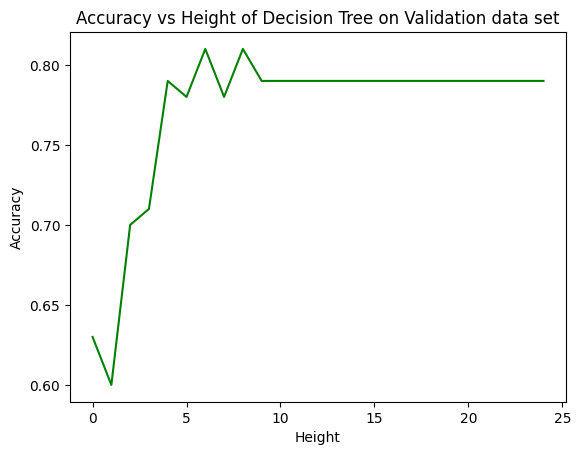
**Objective and goals**:

* So our objective is to create a model of decision tree based on information gain through entropy, so that in the given test dataset we can predict the output class {B,R,L}
* Then find out the accuracy of our model and calculate precision, recall, f1 score and support. And need to compare between 3 models (without pruning, with pruning, with sci-kit learn package in python)

**Program Output**:

| **Model Name** | **Without Pruning** | **With Pruning** | **With Sci-kit learn** |
| --- | --- | --- | --- |
| **Accuracy(%)** | 82 | 81 | 80.32 |
| **Precision** | B - 0.13  L - 0.88  R - 0.94 | B - 0.09  L - 0.81  R - 0.96 | B - 0.08  L - 0.87  R - 0.96 |
| **Recall** | B - 0.22  L - 0.88  R - 0.85 | B - 0.11  L - 0.91  R - 0.81 | B - 0.17  L - 0.84  R - 0.85 |
| **F1 score** | B - 0.17  L - 0.88  R - 0.89 | B - 0.10  L - 0.86  R - 0.88 | B - 0.11  L - 0.85  R - 0.90 |
| **Support** | B - 9  L - 57  R - 59 | B - 9  L - 57  R - 59 | B - 12  L - 87  R - 89 |

**Hyperparameter Tuning:**



To prune the tree, we have used Reduced Error Pruning. For pruning, we have used the accuracy on validation split for selecting the best pruned tree.