**Title:** Building Decision Tree Classifier by given attribute selection measures and evaluate their performance.

**Objective/Aim:** To perform following tasks:

1. Build a menu driven GUI that allows user to upload datasets, select various tree induction methods and see metrics of performance for analysis.
2. Implement tree induction functions in the backend for the options of attribute selection as:
   1. Information Gain
   2. Gain Ratio
   3. Gini Index
3. Implement function for performance analysis for the metrics:
   1. Recognition rate
   2. Misclassification rate
   3. Sensitivity
   4. Specificity
   5. Precision and recall

**Introduction:**

This assignment is about induction of Decision Trees from given dataset. We divide the dataset into two parts, training dataset (80%) and testing dataset(20%). We put attributes of given training dataset at different levels of a tree following a measure for attribute selection. Once the tree is built, we test it’s performance using the testing dataset. For evaluating the performance we make use of various metrics provided in the PS. Lastly we built a suitable menu driven GUI platform for the user to interact with this functionality.

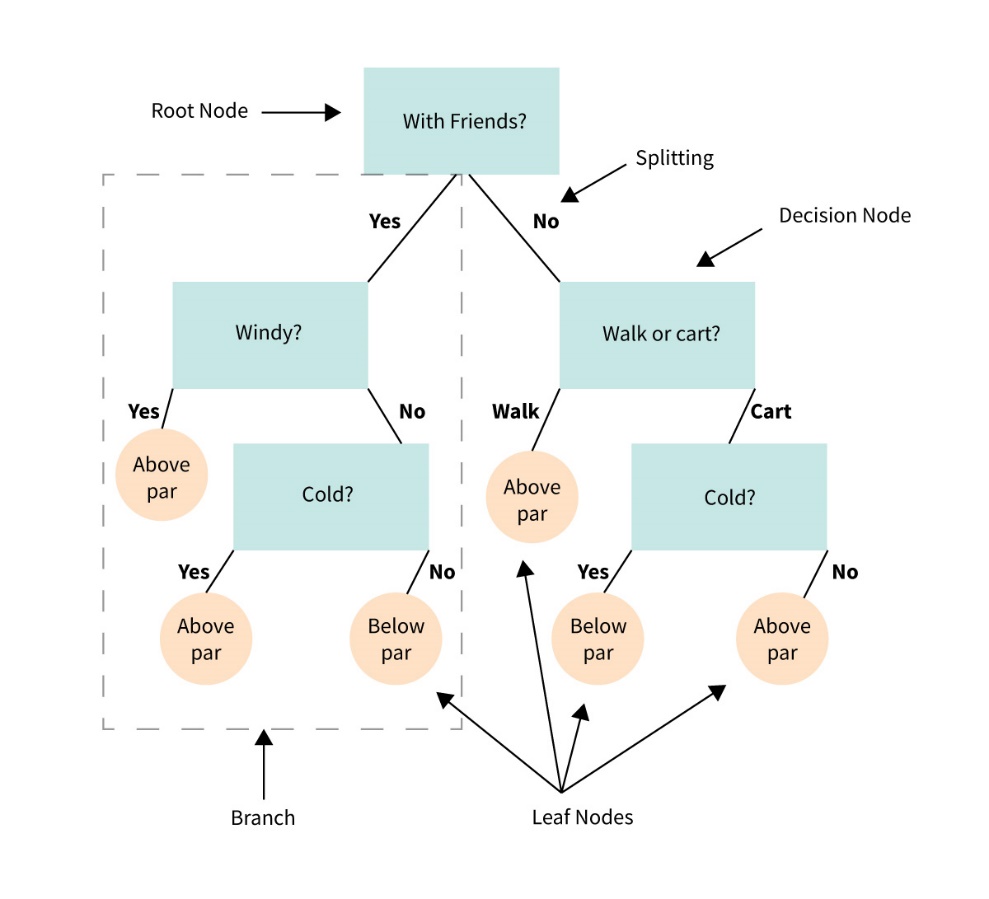
**Theory/Algorithm:**

Classification: classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

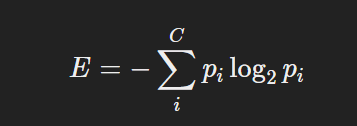
Decision Tree: A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

Decision trees are represented by internal nodes and external nodes

Internal Nodes: Rectangular External Nodes: Oval



3 Heuristics:

1. Entropy: Also known as information gain. It is given by:  … E = Info

Gain = Info (D) – InfoA (D)

Pi = |Ci, D| / |D|

1. Gain Ratio: Formula:

Gain Ratio = Gain (A)/SplitInfoA(D)

1. Gini Index: Formula

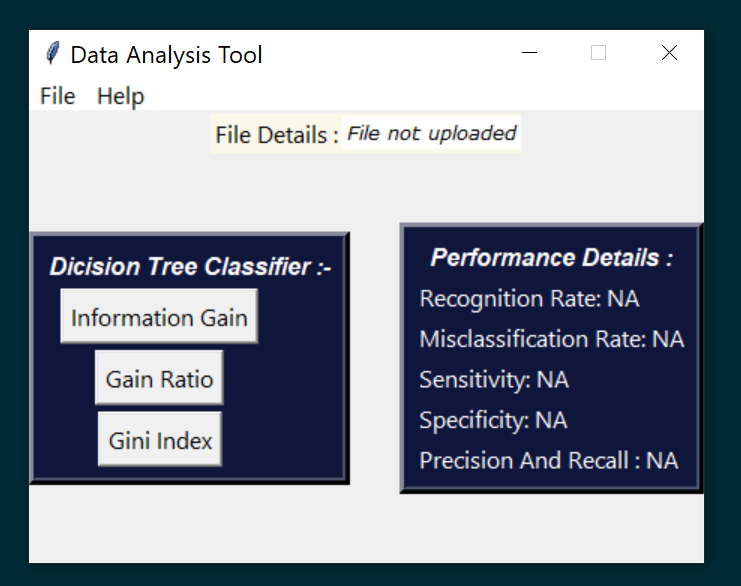
GiniA(D) = |D1| / |D| Gini (D1) + |D2| / |D1| Gini(D2)

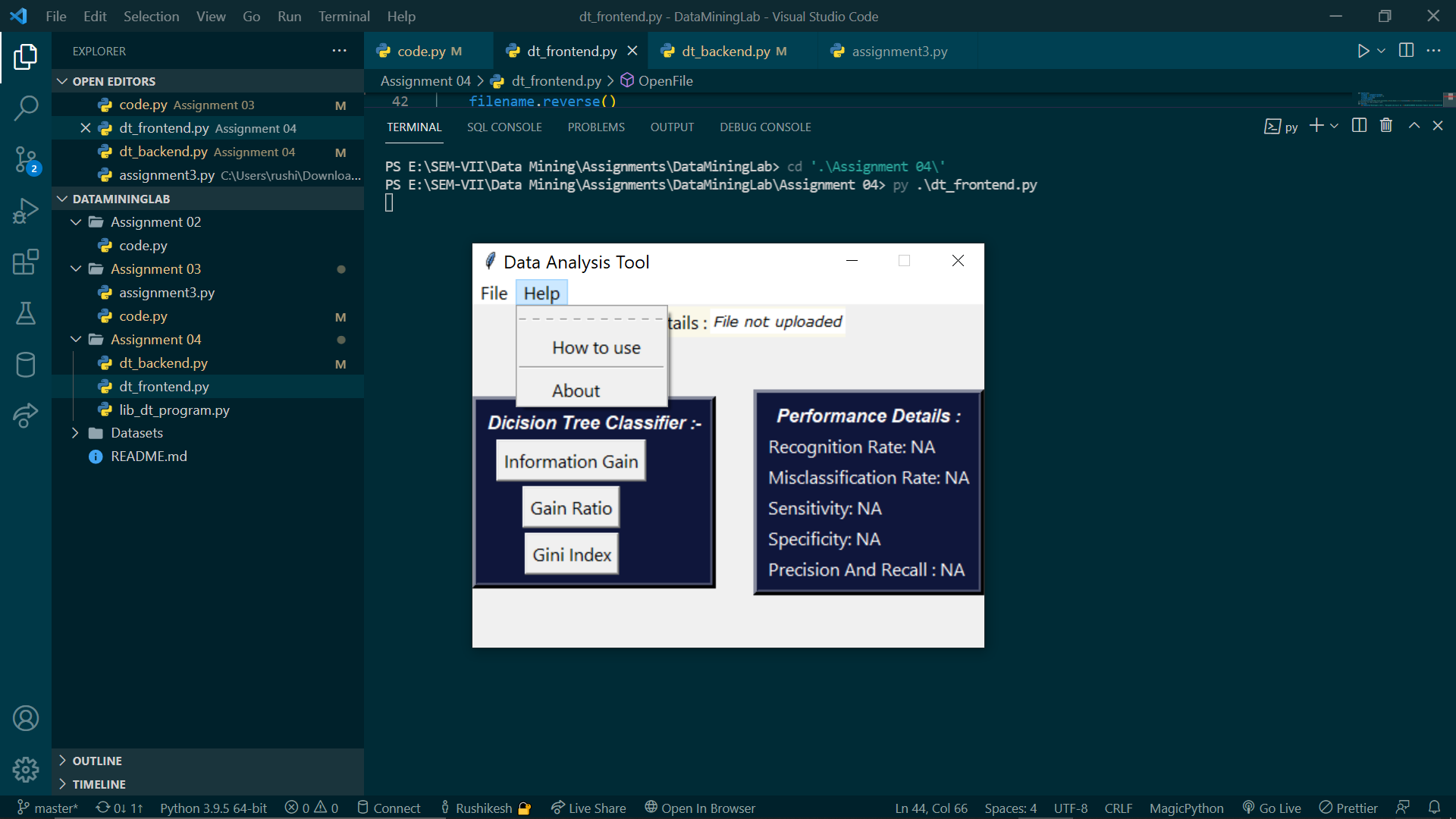
Performance metrics:

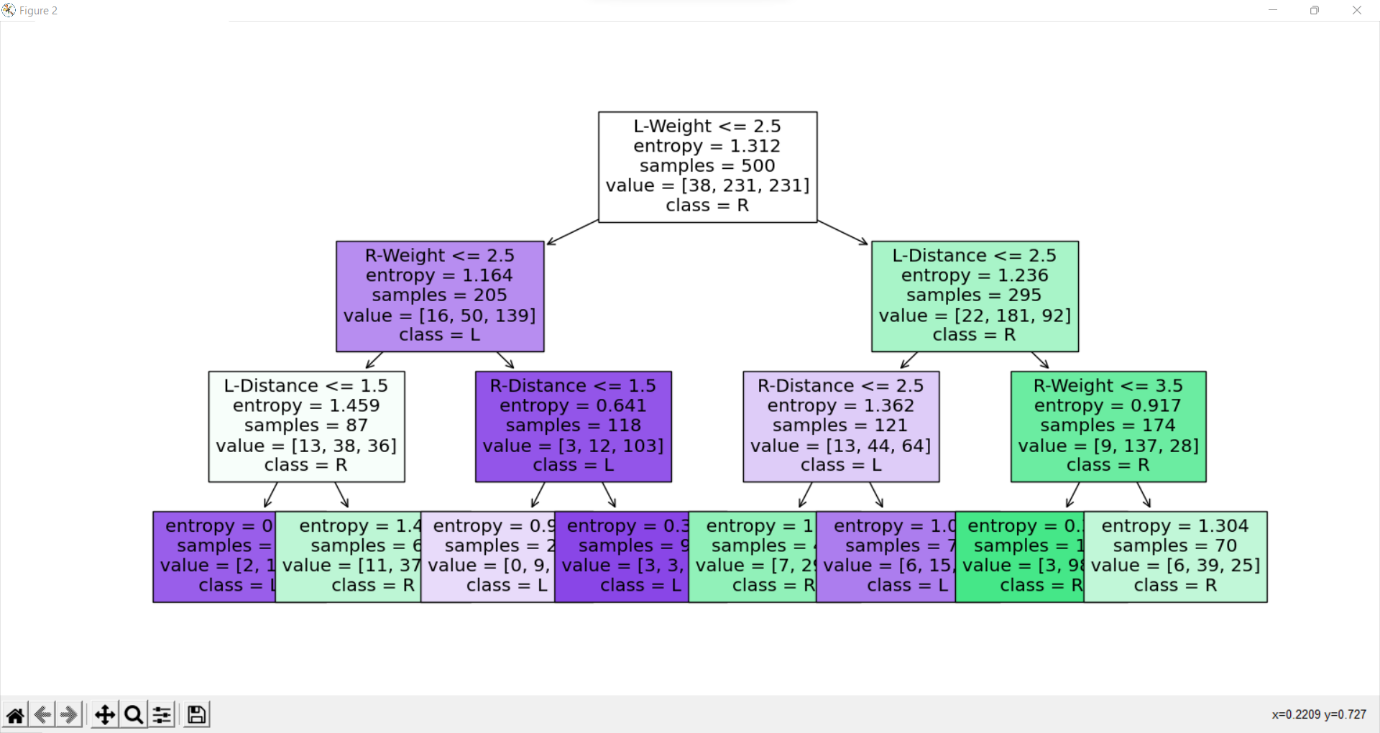
1. Recognition rate: TP + TN/ P+N
2. Misclassification rate: FP+FN/P+N
3. Sensitivity: TP/P
4. Specificity: TN/N
5. Precision: TP/TP+FP

**Actual Experimentation/ simulation/ result/ Observation:**

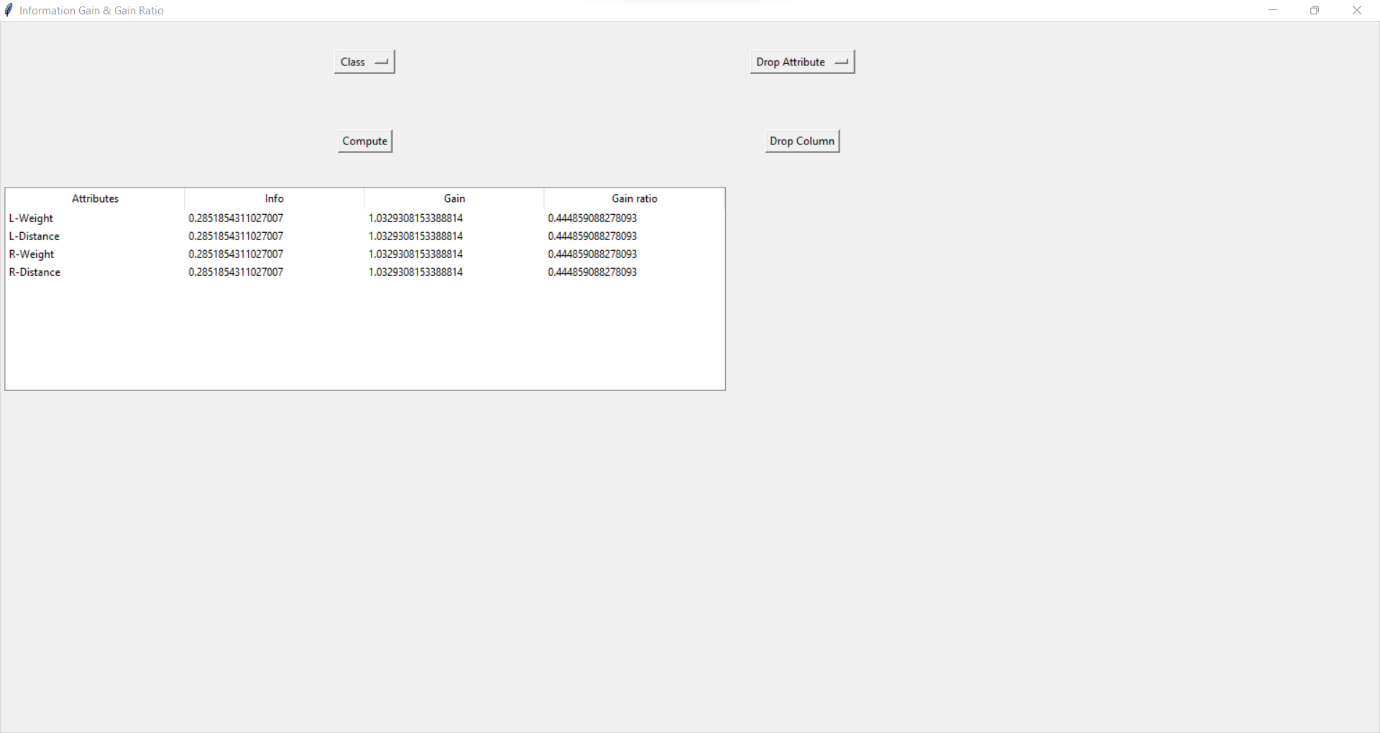
The GUI:











**Conclusion:** In this assignment we learnt:

1. How to build a dynamic gui using tkinter
2. For to parse .xcel, .csv files
3. Building Decision Tree using various attribute selection methods.
4. Performance evaluation of Decision trees.

**References:**

1. <https://www.vedantu.com/commerce/gaining-ratio>
2. <https://towardsdatascience.com/>
3. <https://online.datasciencedojo.com/blogs/a-comprehensive-tutorial-on-classification-using-decision-trees>