**Abstract:** This documentation describes the implementation of a custom Decision Tree Classifier in Python using the scikit-learn library. The decision tree is designed to handle classification tasks and includes methods for building the tree, making predictions, and calculating accuracy. The classifier is applied to the Iris dataset to demonstrate its functionality.

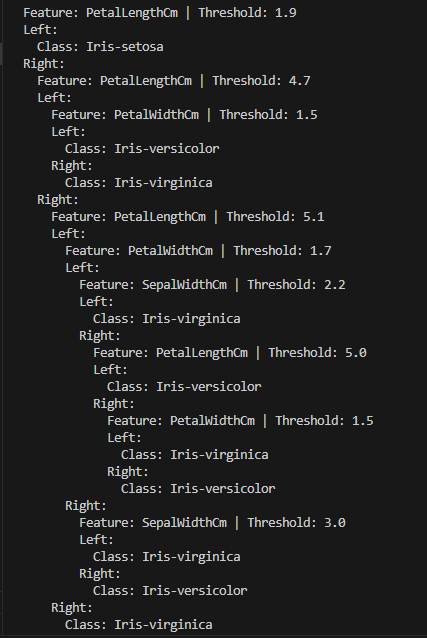
**Introduction:** Decision trees are popular machine learning models used for classification tasks. This document presents a custom Decision Tree Classifier implemented in Python. The classifier is constructed using the information gain criterion to determine the optimal splits in the data, and it includes features such as maximum depth and minimum samples for splitting to control the tree's complexity. The classifier is evaluated on the Iris dataset, a well-known dataset in the machine learning community.

**Related Work:** Decision trees are a fundamental part of machine learning, and various implementations exist in libraries like scikit-learn. This custom implementation aims to provide a simplified understanding of how decision trees work and how they can be applied to classification tasks.

**Description of Applied Method:** The custom Decision Tree Classifier consists of two main classes: **DecisionTreeNode** and **DecisionTreeClassifier**. The former represents nodes in the decision tree, while the latter manages the overall tree construction, prediction, and printing. The tree-building process involves recursively selecting the best split based on information gain until a stopping criterion is met. The classifier is trained on the Iris dataset, and the resulting tree is printed to visualize its structure.

The custom Decision Tree Classifier is applied to the Iris dataset, and the resulting tree is printed for examination. Additionally, the accuracy of the classifier is calculated on a test set using the **calculate\_accuracy** method.

Decision Tree:



Accuracy:



**Conclusion:** The custom Decision Tree Classifier demonstrates the basic principles of decision tree construction for classification tasks. By applying the classifier to the Iris dataset, users can gain insights into the decision-making process of the model. The implementation provides a starting point for understanding and customizing decision tree algorithms. Further improvements and optimizations can be explored to enhance the classifier's performance and versatility