Multiclass Decision Tree – Full Explanation

# 🔍 Objective

The goal of this document is to explain how a Decision Tree handles Multiclass Classification problems, with a focus on understanding Gini Impurity calculations, splits, label encoding, and interpretation of tree output.

# 📊 Dataset Overview

We are working with a small fruit dataset where each fruit is characterized by two features: 'Color' and 'Shape'. The fruits belong to one of several classes: Apple, Banana, Kiwi, and Orange.

# 🧠 Label Encoding

Since Decision Trees work with numeric data, categorical values such as 'Red', 'Green', 'Yellow', 'Orange' for 'Color' are converted to numbers using Label Encoding. Here's an example encoding:

Red → 0  
Green → 1  
Yellow → 2  
Orange → 3

# 🌳 Tree Structure Interpretation

Below is the tree output that was generated using a Decision Tree Classifier. The tree uses 'Gini Impurity' as the splitting criterion. Each decision node and leaf node contains the following information:

* - Gini Impurity  
  - Number of samples  
  - Class distribution (value = [...])  
  - Predicted class at that node

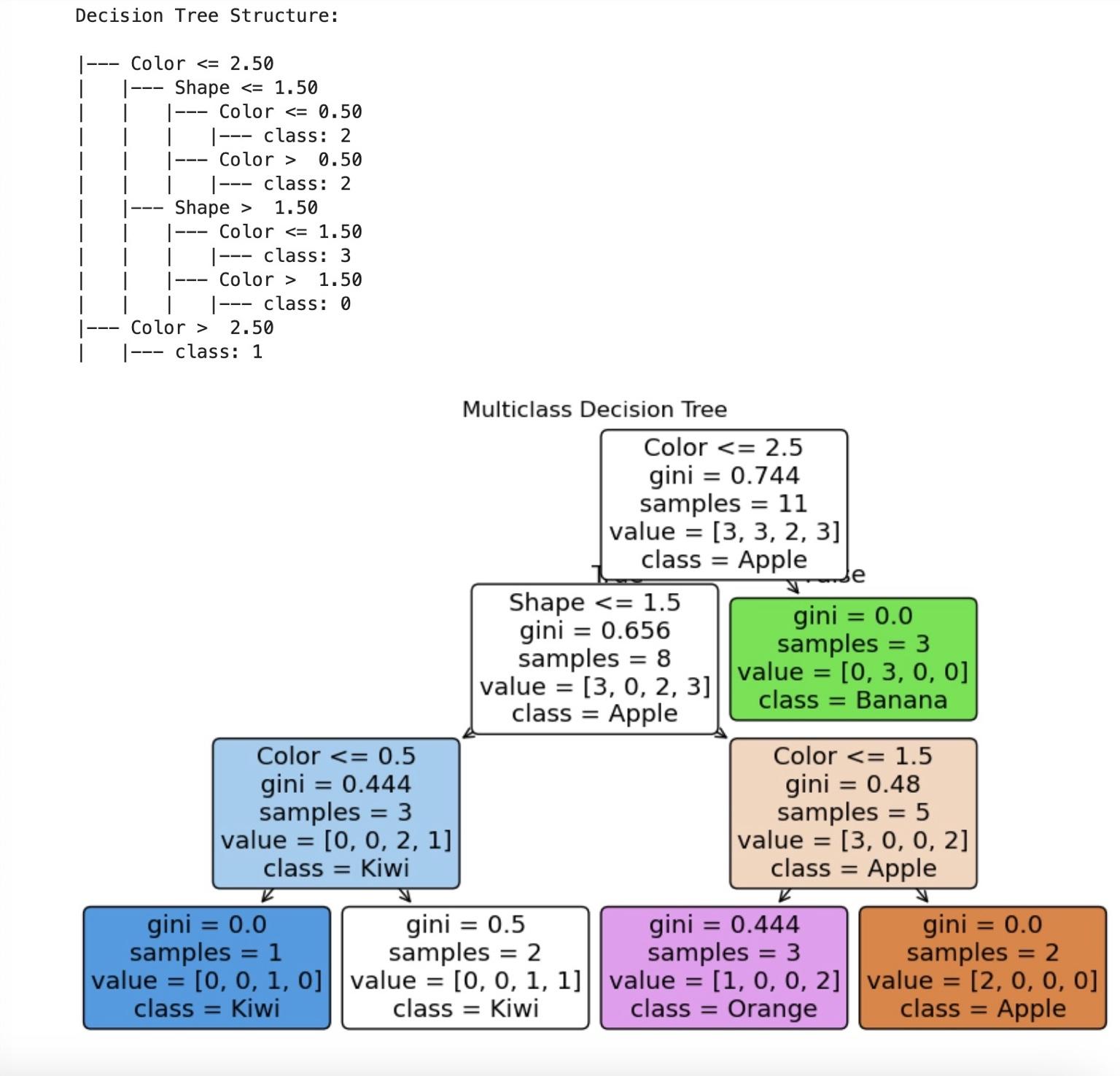


Figure: Visual and text representation of the multiclass decision tree

# 📐 Gini Impurity Explained

Gini Impurity is used to measure the 'impurity' or 'disorder' in a node. The formula is:  
Gini = 1 - Σ(p\_i)^2  
Where p\_i is the probability of class i in the node.  
Example calculation for parent node:  
- value = [3, 3, 2, 3] → total = 11  
- Gini = 1 - (3/11)^2 - (3/11)^2 - (2/11)^2 - (3/11)^2 = 0.744 (matches output)

# ❓ What does "Color <= 2.5" Mean?

The decision tree shows a condition like 'Color <= 2.5'. This doesn't refer to a direct numeric value from the original dataset. It is a threshold applied to the Label Encoded values of Color. For example:

- Encoded Color:  
 Red = 0  
 Green = 1  
 Yellow = 2  
 Orange = 3

'Color <= 2.5' translates to 'Red, Green, Yellow go left', and 'Orange goes right'. These thresholds are chosen by the tree to best split the dataset based on impurity.

# 🔍 Tree Splitting Strategy

At each node, the tree evaluates all possible splits using features like Color and Shape. It chooses the split that results in the highest decrease in impurity. For example:

- Splitting on Color might create two nodes with gini = 0.45 and 0.5 → weighted gini = 0.47  
- Splitting on Shape might result in weighted gini = 0.50  
Since 0.47 < 0.50, the tree chooses to split on Color.

# 📦 Interpreting Leaf Nodes

Each leaf node shows the predicted class (the most common one in that node), the number of samples, and the class distribution. A gini of 0.0 means the node is pure.

# ✅ Summary

This decision tree shows how multiclass classification is handled natively by scikit-learn's DecisionTreeClassifier. Label encoding helps convert categorical features. Gini Impurity is used to quantify disorder. Splits are chosen based on which feature + threshold combination reduces impurity the most.