# Decision Tree Analysis on Iris Dataset

Jayesh Pamnani
*University of Maryland, College Park*
College Park, USA
jpamnani@umd.edu

Anjaneya Ketkar
*University of Maryland, College Park*
College Park, USA
aketkar@umd.edu

Janesh Hasija
*University of Maryland, College Park*
College Park, USA
jhasija@umd.edu

Divya Lnu
*University of Maryland, College Park*
College Park, USA
divlnu@umd.edu

***Abstract*—Decision Tree is an algorithm that is used for classification systems for developing algorithms to predict the target variable. This journal paper presents a guide to in-depth knowledge about decision trees and how we can generate a decision tree using the iris data set. An experiment was conducted to divide the data set into three parts and then use the data for training and testing. The purity measure of the dataset was changed three times and results were visualized.**

***Index Terms*—Decision Tree, Classifier, criterion**

## INTRODUCTION

In the vast realm of machine learning and data science, decision trees have emerged as a fundamental and versatile tool for solving classification problems. These elegant structures mimic the human decision-making process and have found applications in various domains, from finance to healthcare. The Iris Dataset [1], introduced by British biologist and statistician Ronald A. Fisher in 1936, has long been a touchstone in the field of data analysis and machine learning. The dataset provides an opportunity to investigate the efficiency of the decision trees and the authors have used 150 samples to classify Iris flowers into their respective species based on their sepal length, sepal width, petal length, and petal width. The results are later plotted using graphviz and are compared by changing the purity measure using criterion.

## METHODOLOGY

In this paper, the authors delve into the diverse concepts of decision trees within the Python programming ecosystem, employing them to classify species within the Iris dataset. Their methodology relies on Python version 3.11.2, NumPy version 1.25.2, and Pandas version 2.1.0. This study serves as an excellent start for those looking to initiate their exploration of classification techniques using the Python programming language.

The authors begin by pre-processing the data as the csv file is read using numpy, and the features and the target are extracted. Multiple experiments are conducted on the dataset, involving varying proportions for each split. Initially, only 20% of the dataset is allocated for testing purposes, with the remainder reserved for training. Subsequently, this allocation is increased to 30%, and so forth.

Further, changing the criterion of the decision tree can improve the classifier. For doing this model is trained with 3 different classifier criteria namely Gini, which tries to minimize the misclassification of data, entropy, aiming to maximize the information gain at each node, and log_loss[2]. Furthermore, we divided the dataset into training and test subsets, initially training the model. Subsequently, we refined its accuracy through retraining, utilizing 67% of the test data. Visualizations were generated using Graphviz, illustrating the decision trees that aid in predicting the Iris flower species based on their distinctive features.

## RESULTS

The paper explores various analyses of Decision Trees on the Iris Dataset. Starting with the test size of 20%, the accuracy was found to be 98%. On experimenting with the data, the accuracy was 96% when the test size was 30% and the accuracy was 95% when the test data size was kept at 40%. A tree diagram was created for the 96% accuracy model. Subsequently, a decision tree was generated using Graphviz, an open-source graph visualization software, as shown in Fig 1. The tree branches into several nodes, each representing a decision rule based on the selected features and thresholds. The leaf nodes correspond to the final predicted classes (Setosa, Versicolor, or Virginica).

To assess stability, we compare the trained Tree A (Fig 1) with an online reference Decision Tree, Tree B, that is generated and trained on the same dataset [3].

### *A. Visual Comparison*

Visualizing both trees using a tree plotting library like Graphviz, shows the differences in their structures, depth of both the trees, on what levels the leaf nodes are getting generated, starting feature, and comparison parameters.

- Both trees start with the same feature, i.e., petal width.
- Sample size of Tree A is 100, whereas the sample size of Tree B is 150.

### B. Summary Statistics

- Tree A has a maximum depth of 6 and has 8 leaf nodes.
- Tree B has a maximum depth of 6 and has 9 leaf nodes.

### C. Performance Metrics

- Gini value differs on comparing both trees on almost all levels.
- The last comparison parameter, having depth 6, is the Sepal Width in Tree A whereas it is the Petal Length in the case of Tree B.

Obtaining different Gini values for a decision tree when applying it to the same dataset can be due to the following factors:

- **Randomness in the Algorithm**: Decision trees can introduce randomness by randomly selecting subsets of data or features during training. This can lead to different tree structures and therefore different Gini values for each run.
- **Limited Data**: If the dataset is relatively small, the randomness in splitting data into training and validation sets can cause variations in model performance, including Gini values.
- **Data Preprocessing**: Differences in data preprocessing can affect the Gini index.
- **Tie-breaking Rules**: When calculating Gini impurity for a decision tree node, there can be tie-breaking rules for splitting nodes when multiple features have the same Gini impurity. Small variations in these rules or floating-point arithmetic can lead to slightly different trees and Gini values.

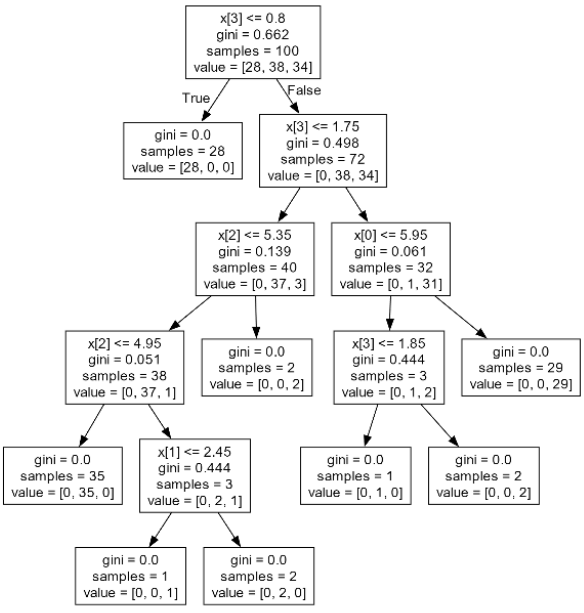


Fig. 1. Decision Tree with "Gini" Criterion

With the aim of maximizing the accuracy, the purity measure was modified by changing the criterion to "entropy." By doing so, the accuracy was found to be 95%. The subsequent section presents (Fig 2) the decision tree model for the same.

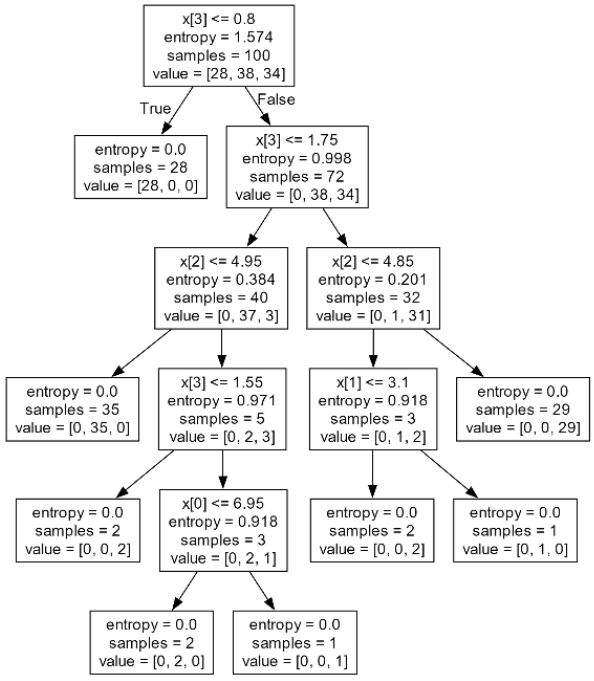


Fig. 2. Decision Tree with "Entropy" Criterion

Changing the criterion to "log loss", the accuracy was found to be 94%. Presented below in Fig 3 is the decision tree with the "log_loss" criterion.

When the decision tree was retrained with the "gini" criterion after splitting the dataset and allocating 33% for testing, it resulted in 100% accuracy. In the following section (Fig 4), the decision tree model for the same will be elucidated.

One of the key considerations in our analysis is the choice between the criteria. Practically, the suitability of each criterion depends on factors such as class distribution, dataset size, and domain-specific requirements.

## Comparison and Discussion

In the analysis presented in the paper, the authors evaluated three primary criteria for constructing decision trees: the "gini", "entropy" and "log-loss" criteria. These criteria were chosen to assess their impact on both model performance and structure. The difference in accuracies raises interesting considerations regarding the choice of criterion. It's essential to delve into the implications of these accuracy disparities, particularly in the context of the research question. The authors also observed that increasing the training data set would not always improve the prediction, it could add an unwanted bias if the data is not equally split among all the categories.

Further exploration is warranted to expand our understanding of decision tree criteria. Investigating additional criteria

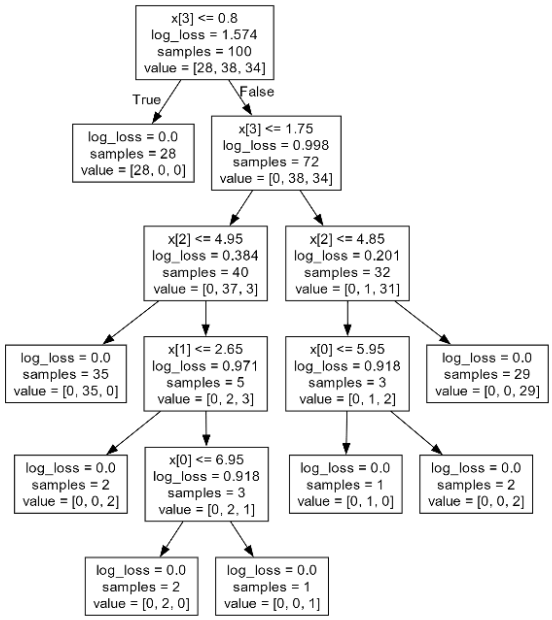


Fig. 3. Decision Tree with "Log_Loss" Criterion

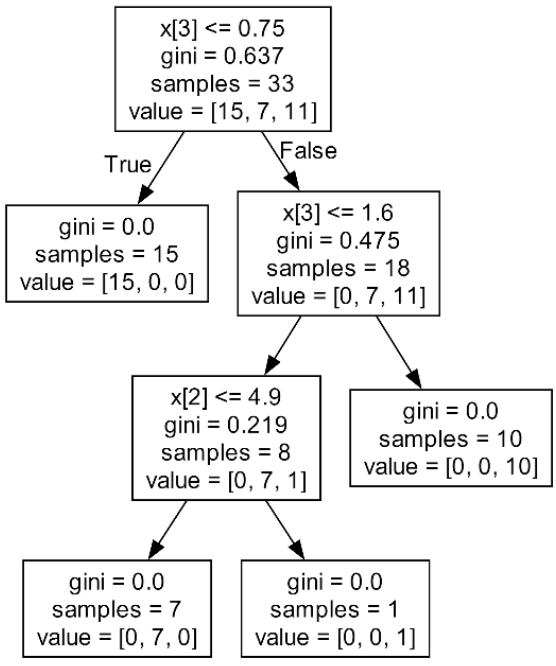


Fig. 4. Decision Tree with "Gini" Criterion after Retraining the model

and assessing their effects on model performance could provide a more comprehensive perspective on this critical aspect of decision tree modeling.

## CONCLUSION

The paper explored the application of decision tree classifiers to the Iris dataset using various criteria. The analysis encompassed constructing decision trees, visualizing tree structures, evaluating model performance, and comparing criteria. The choice of criterion significantly impacted accuracy. The findings emphasize the importance of criterion selection and the need for careful consideration of problem characteristics. Researchers and practitioners should assess the trade-offs between different criteria when applying decision tree classifiers to classification tasks.

## REFERENCES

[1] https://archive.ics.uci.edu/dataset/53/iris
[2] https://www.ibm.com/topics/decision-trees#
[3] https://scikit-learn.org/stable/modules/tree.html