# A Custom Implementation of Decision Tree Classifier and Evaluation of Its Performance

Adel Hamad Alkhamisy
Department of Computer Science
George Mason University
Fairfax, USA
Email: aalkhami@gmu.edu

***Abstract*—This study presents a customized implementation of a decision tree classifier[1], [2], [3], a fundamental and well-known machine learning technique. We construct and test the classifier on a publicly available wine dataset, a multivariate data source based on the results of a chemical study of wines. The F1 score, recall, accuracy, and precision are just a few of the several metrics that are employed to comprehensively assess the tree-based model. We use K-fold cross-validation to obtain a more reliable estimation of the model's performance.**

## I. Introduction

In the present world of data, machine learning has become a powerful method for getting beneficial insights and producing intelligent forecasts. Due to their simplicity, interpretability, and robustness against noisy input, decision trees [2] have become very popular among the numerous machine learning methods. In this work, a custom decision tree classifier [1] implementation is examined. K-fold cross-validation [4] is used to evaluate the model's performance on a small dataset of wines, and it is shown how this validation strategy can improve the model's performance.

## II. Background

To operate, decision tree classifiers [2] divide the input space into regions and give each region a class name. The partitioning generates a tree structure with internal nodes for features and leaf nodes for class labels based on feature thresholds. The two metrics for splitting that are most frequently employed are gini impurity and information gain. Despite their ubiquitous use, decision trees frequently exhibit high variation, which, if improperly pruned or managed, can lead to overfitting and subpar model performance.

## III. Proposed Approach

The suggested decision tree classifier is created from scratch in Python [1] and has a number of programmable settings, including the minimum sample size to split and the maximum depth of the tree. The Gini [2] impurity serves as a splitting criteria for the classifier. On the wine dataset, the model is tested and evaluated. Performance is measured in terms of accuracy, precision, recall, and F1 score using K-fold cross-validation [5].

## IV. Experimental Results

In a K-fold cross-validation configuration, the custom decision tree classifier performed admirably on the wine dataset, obtaining an accuracy of 90%, mean accuracy of 0.83 (+/- 0.21), average precision of 0.88 (+/- 0.13), recall of 0.84 (+/- 0.22), and average f1 score of 0.82 (+/- 0.21). The cross-validation process [4], which involved training and testing the model multiple times on different subsets of the data, helped to reduce model variance and improve the robustness and generalizability of the model.

## V. Conclusions

The success and effectiveness of creating a unique decision tree classifier from scratch and evaluating its performance using K-fold cross-validation are demonstrated in this study. The evaluation's findings show that the model performs satisfactorily, and K-fold cross-validation is a useful tactic for minimizing overfitting and improving model performance. Future studies can investigate other validation strategies and apply the model on significant and intricate datasets.

## Acknowledgment

The author would like to thank N. Nerd for the informative video tutorial "Decision Tree Classification in Python (from scratch!)" [1], which greatly assisted the study, although any errors are our own and should not tarnish the reputations of these esteemed persons.

We also extend our thanks to S. J. Russell and P. Norvig for their comprehensive textbook "Artificial Intelligence: A Modern Approach" [2], which has been a significant reference in this research.

Finally, we express our gratitude to L. Buitinck et al. for their insightful paper "API Design for Machine Learning Software: Experiences from the Scikit-Learn Project" [3], which has enlightened our understanding of machine learning software design.

## References

[1] N. Nerd, "Decision tree classification in python (from scratch!)," 2021, accessed: 2023-04-28. [Online]. Available: https://youtu.be/sgQAhG5Q7iY

[2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, third edition ed. Prentice Hall Press, 2009.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," in *Wadsworth Statistics/Probability Series*. Taylor & Francis, 1984.

[4] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111–147, 1974. [Online]. Available: http://www.jstor.org/stable/2984809

[5] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "Api design for machine learning software: Experiences from the scikit-learn project," *ArXiv*, vol. abs/1309.0238, 2013.