Performance Analysis of Decision Tree Classifier on Wine Dataset (July 2023)

Bishwambhar Dahal[1] and Sirjana Bhatta[1]

[1]Department of Electronics and Computer Engineering, IOE, Thapathali Campus, Kathmandu 44600, Nepal
Corresponding author: Bishwambhar Dahal(dahalbishwambhar@gmail.com)

ABSTRACT Decision tree classifiers are widely used in various domains, dataset classification. In this study, we explore the application of decision tree classifier in Wine dataset. Here, we utilize both Gini impurity and entropy for tree construction and drawing decision trees of different depths. The wine dataset used in this analysis contains samples characterized by attributes such as Alcohol, Malic Acid, Ash, Alkalinity of Ash, Magnesium, Total Phenols, Flavonoid, and so on. By leveraging Gini impurity and entropy measures, we aim to explore their effectiveness in constructing decision trees and visualizing the classification process at different tree depths. We evaluate the performance of decision trees generated using both impurity measures and compare their interpretability and accuracy. This study provides insight into the applicability of Gini impurity and entropy for decision tree construction in wine dataset classification, shedding light on the relationship between tree depth, impurity measures, and classification performance. This study contributes to a better understanding of decision tree classifiers and their visualization techniques.

Keywords *Entropy, Decision Tree Classifier, Gini, Tree depth.*

I Introduction

The use of machine learning algorithms [1] has become increasingly prevalent in various fields, including the analysis and classification of complex datasets. In the field of machine learning [2], decision tree classifiers [3] have proven to be very efficient tools for classification tasks. Based on input features, they can create hierarchical decision structures which makes them particularly useful for analyzing and classifying complex datasets. The wine dataset is an interesting domain for applying decision tree classification techniques.

The graphical representation of decision trees [4] as flowchart-like structures facilitates understanding and provides valuable insights into the decision-making process. Furthermore, decision trees can handle both categorical and numerical data, making them well-suited for analyzing the diverse attributes present in the wine dataset.

In this study, we focus on the application of a decision tree classifier on the wine dataset to predict the type of wine based on a set of key attributes and assess its effectiveness in predicting the wine types based on the available attributes. We seek to identify the most important features that contribute to the classification process. We aim to evaluate the performance of the decision tree classifier.

By accomplishing these objectives, we can gain a deeper understanding of the wine dataset, uncover the significant attributes influencing the wine types, and contribute to the existing knowledge in wine classification. Additionally, the insights and methodologies derived from this study can be extended to other similar classification problems in various domains.

This report focuses on application of decision tree classifiers on a wine dataset. The wine dataset used in this study comprises of various attributes such as Alcohol, Malic Acid, Ash, Alkalinity of Ash, Magnesium and so on. By using decision tree classifiers, we aim to classify the wines based on their characteristics. The primary objective of this analysis is to evaluate the performance of decision tree classifiers on the wine dataset and assess its ability to accurately classify wine samples when given the required characteristics. We will explore different aspects of decision tree classifiers including tree depth and impurity measures. We will also uncover the advantages and drawbacks of decision tree classifiers in wine dataset classification.

Overall, the decision tree classifier is a powerful machine learning algorithm capable of achieving high accuracy in various tasks. With its interpretability and versatility in handling different data types, it is widely used in the field of machine learning. In the subsequent sections, we will outline the methodology employed for training and evaluating decision tree classifiers on the wine dataset. We will also present the experimental results and will discuss on the implications of our findings.By leveraging the interpretability and versatility of decision trees, we aim to uncover the attribute relationships and to provide valuable insights into the classification process.

II Methodology

A Brief Theory

Entropy [5]is a measure of impurity or uncertainty in a set of data. In the context of machine learning and decision trees, it is commonly used as a criterion to evaluate the quality of a split based of certain feature in a dataset. The entropy of a set is calculated based on the proportion of different classes or categories present in that set. A higher entropy indicates , . In machine learning, high entropy typically refers to higher impurity or uncertainty in a set of data. The high entropy of a set of data indicates that the data points are distributed more evenly among the different classes, and there is no clear separation between them. A lower entropy suggests a more pure or certain set.

The Gini index [6], similar to entropy, is another measure of impurity commonly used in decision tree algorithms. It calculates the inequality or impurity in a set by evaluating the probability of misclassifying a randomly chosen element from the set. The Gini index ranges from 0 to 1, where 0 represents a perfectly pure set (all elements belong to the same class), and 1 represents maximum impurity (the set contains an equal proportion of each class).

Classification error, also known as misclassification error or misclassification rate, is a simple measure that determines the error rate in a set. It is calculated by dividing the number of misclassified instances by the total number of instances in the set. In decision tree algorithms, it can be used as a criterion for evaluating the quality of a split, aiming to minimize the misclassification error.

The impurity measures like entropy and Gini index are used to evaluate the homogeneity or impurity of a set of samples at each node in the decision tree. They help determine the best splitting criterion for feature selection.

A decision tree is a graphical model used for decision-making which is represented in a flowchart-like structure. In supervised machine learning, it is one of most effective algorithm used for classification and regression tasks. The decision tree classifier applies the decision tree model to make predictions.The depth of a decision tree refers to the length of the longest path from the root node to any leaf node. The decision tree classifier algorithm constructs a model in the form of a tree structure. The tree is composed of nodes and branches, where each node represents a feature or attribute of the wine dataset and the branches of the tree correspond to the possible values or ranges of those attributes/features.

Tree depth, also known as maximum depth of a decision tree, refers to the length or number of levels from the root node to the farthest leaf node in the tree. It represents the maximum number of splits or decisions made along a path from the root to a leaf node. Shallower decision trees with limited depth tend to be more interpretable and less complex as they have a smaller number of decision rules and are easier to visualize and comprehend.

There are three types of nodes in a decision tree: the root node, internal nodes, and leaf nodes. The root node is the topmost node, representing the attribute or feature from which the data is initially split. The root node represents the attribute that best separates the wine samples based on their types. The algorithm determines the best attribute by evaluating various criteria, such as information gain, Gini impurity, or entropy. These measures quantify the amount of information provided by an attribute for classifying the wine samples. Once the root node is determined, the algorithm proceeds recursively to create internal nodes that split the dataset based on different attribute values. This splitting process continues until the algorithm reaches the leaf nodes, which represent the predicted wine types. Internal nodes are intermediate points in the tree corresponding to specific features, while leaf nodes, located at the bottom, represent the class labels. The algorithm determines the best features to split by using different methods, such as Gini impurity and Entropy. Classification error is also employed occasionally for this purpose. These measures quantify the amount of information provided by an attribute for classifying the wine samples and how well a feature can separate classes. To construct an optimal decision tree classifier, the algorithm seeks to minimize impurity or maximize information gain at each internal node. Impurity refers to the degree of randomness or mixture of wine types in a given subset of the dataset. The goal is to create homogeneous subsets in terms of wine types by selecting the most discriminative attributes.

Decision tree classifiers are capable of handling both categorical and numerical values. For categorical values, unique branches are created for each unique value, while for numerical values, data is handled by splitting based on thresholds or ranges. Ensemble methods like Random Forest and Gradient Boosting are applied to enhance the performance of decision tree classifiers. Overfitting may occur when the model learns the training data too well, resulting in poor generalization to unseen data. In decision trees, there is higher risk of overfitting when the tree becomes complex and deep. To mitigate overfitting, techniques such as pruning, setting maximum depth, controlling the minimum number of samples required to split a node and selecting the best attribute to split a given node are employed.

B Mathematical Formulae

Let's assume we have a dataset 'X' with m records and n attributes i.e. having dimension $m \times n$. If the whole dataset is divided into k classes, the formula for entropy is given by:

$$Entropy = \sum_{i=0}^{k} p_i \log_2(p_i) \tag{1}$$

In Equation 1, p_i represents the probability of instances belonging in class i within the dataset, and k represents the total number of classes.

When a node p is split into k partitions, the quality of split is computed as:

$$\text{Entropy}_{\text{split}} = \sum_{i=1}^{k} \left(\frac{n(i)}{n} \cdot \text{Entropy}(i) \right) \tag{2}$$

where,
n_i=Number of records at child i,
n=Number of records at node p.
The formula for calculating information gain is as follows:

$$\text{Inf\_Gain} = \text{Entropy(p)} - \sum_{i=1}^{k} \left(\frac{n(i)}{n} \cdot \text{Entropy}(i) \right) \tag{3}$$

where,
Entropy(p)= Entropy of parent,
Inf_Gain=Information Gain.

The formula for Gini index is given by:

$$Gini = 1 - \sum_{i=0}^{k} (\log_2(p_i))^2 \tag{4}$$

In Equation 4, $Gini$ represents the Gini index, k represents the total number of classes and p_i represents the probability of instances belonging in class i within the dataset.
The equation 2 can be modified for gini index and hence the quality of split is computed as:

$$\text{Gini}_{\text{split}} = \sum_{i=1}^{k} \left(\frac{n(i)}{n} \cdot \text{gini}(i) \right) \tag{5}$$

where,
n_i=Number of records at child i,
n=Number of records at node p.

The equation 6 for information gain using entropy can also be modified to calculate information gain using GINI index as:

$$\text{Inf\_Gain} = \text{Gini(p)} - \sum_{i=1}^{k} \left(\frac{n(i)}{n} \cdot \text{Gini}(i) \right) \tag{6}$$

where,
Gini(p)= Gini of parent,
Inf_Gain=Information Gain.

Classification error at node t is calculated as:

$$\text{Error}(t) = 1 - \max P(i|t) \tag{7}$$

C System Block Diagram

The block diagram of the system is shown in figure 1. Taking dataset as input, we constructed decision tree by fitting decision tree classifier on our dataset.

D Working Principle

Decision tree classifier is widely used algorithm for classification task. It is supervised machine learning algorithm [7]. We have implemented this algorithm using python programming language. We first load wine dataset available in scikit-learn (sklearn).. Wine dataset contains 178 records having 13 features for each record. The whole dataset is divided into 3 classes. Then, we split our dataset into training set and test set with 77% training set and a 33% testing set. In our next step, we train our classifier on the training data.
Initially, we compute the entropy of the parent node, which serves as a reference point. Then, While training the classifier, we focus on calculating the entropy and information gain for each attribute, which are essential measures for determining the structure of the decision tree. Initially, we compute the entropy of the parent node, which serves as a reference point. Then, we calculate the entropy of each attribute by following a series of steps. Firstly, we calculate the entropy of each category or class present within that attribute. This involves analyzing the distribution of data points among the different categories and quantifying the uncertainty or randomness within each category. By summing up the entropies of all categories within an attribute, we obtain the entropy value for that specific attribute. This process is repeated for every attribute in the dataset. Then with the help of entropy we calculate information gain for each attribute. Information gain gives us information about reduction of entropy we achieve by splitting data based on particular attribute. The attribute with highest information gain is chosen as root node. Then we split our data based on root node. In

our next step, we calculate entropy followed by calculation of information gain for each split data individually. For each section we choose attribute with highest information gain as intermediate node. Then we again split our data. This step is repeated until we reach leaf node in each branch. Here, leaf node gives us class labels. After getting leaf node in each section our training process is completed. Additionally, we have the flexibility to use different impurity measures such as the Gini index or misclassification error [8] in place of entropy.

Using trained model, we predict result on test set. To visualize the performance of a classification model, we construct confusion matrix and generate the classification report. The confusion matrix and classification report provide a summary of the predictions made by the model on a test dataset. Classification report calculate different parameter like accuracy, precision, recall, f1 score, etc. to evaluate model performance. In last step, we construct decision tree using `tree.plot_tree()` function. If desired, we can also fix max depth of tree using standard function of scikit-learn library [9].

E Instrumentation Details

In our study, we utilized the Python programming language to implement the decision tree classifier algorithm. The implementation was done in Jupyter Notebook, which is an interactive computing notebook environment. We imported various libraries for our study, with the most important one being the scikit-learn library. To load the dataset for our machine learning task, we used the `load_wine()` function from the sklearn.datasets module, which offers a variety of datasets. Additionally, we employed the `train_test_split()` function from the `sklearn.model_selection` module to split our dataset into training and test sets. This function takes parameters such as data, target, test_size, and random_state. For evaluating the performance of our model, we utilized the `confusion_matrix()` function from the sklearn.metrics module to construct a confusion matrix. Furthermore, we generated a classification report using the `classification_report()` function from the same module.

To create an instance of the decision tree classifier, we used the `DecisionTreeClassifier()` function from the sklearn.tree module. This function allows us to specify various parameters such as criterion (e.g., entropy, gini, or classification error) and max_depth (the maximum depth of the tree). For the training and prediction tasks, we utilized the `.fit()` and `.predict()` functions of the `DecisionTreeClassifier()` class, respectively. The. `fit()` function takes the training data and the ground truth values, while the .predict() function takes the test data and returns the predicted class values. In order to visualize the confusion matrix, we used the `heatmap()` function from the seaborn library. We also employed the pandas library to represent the data in data frames. Additionally, we used the matplotlib library to construct the decision tree, using various functions to specify the figure size, labels, title, spacing, and other details.

Overall, these libraries and functions were employed to implement and evaluate the decision tree classifier algorithm, as well as to visualize the results in a clear and informative manner.

F Dataset Description

In our study, we have used wine dataset. This dataset is provided by scikit-learn library. This dataset is popular for machine learning task. It consists of chemical analysis of wines from three different cultivars. These cultivars are grown in same region of Italy. It consists of 178 records with 13 attributes: Alcohol, Malic Acid, Ash, Alkalinity of Ash, Magnesium, Total Phenols, Flavonoid, Nonflavonoid Phenols, Proanthocyanins, Color Intensity, Hue, OD280/OD315 of diluted wines and Proline. These 178 records are divided into 3 classes: Class 0, Class 1, Class 2. It is a balanced dataset as number of samples of three groups are nearly equal. It is real world dataset and commonly used for classification task. This dataset gives us the information about relation of chemical composition and associated varieties of wine. Researcher can use this dataset in different classification algorithm and can get insights of key attribute or factor that can differentiate wines. This dataset is useful for benchmarking different classification algorithm.

III EXPERIMENTAL RESULTS

In this study, we investigated the performance of a decision tree classifier using two commonly used impurity measures: entropy and Gini index. We have generated the decision tree structures with different values of maximum depth including depths of 2,3 and without any maximum depth.We were specifically focused on evaluating the effect of different maximum depths on the classification accuracy of the decision tree model. The experiments were conducted on a labeled wine dataset containing various features and corresponding target labels.

Both the entropy and Gini impurity measures yielded comparable classification accuracies. As the maximum depth of the decision tree increased, the accuracy of the classifier generally improved, up to a certain point. This trend suggests that increasing the complexity of the decision tree by allowing more splits and branches can lead to better classification performance.

In our experiments, both entropy and Gini impurity measures performed similarly, with Gini slightly outperforming entropy across most maximum depth values. This outcome aligns with the common observation that Gini impurity is more sensitive to large class probabilities, making it effective in scenarios where classes are imbalanced or when the goal is to minimize misclassifications.

We observed a slight drop in accuracy when the maximum depth exceeded a certain threshold. This decline indicates a potential for overfitting the training data. Overfitting occurs when the decision tree becomes too complex and starts capturing noise or irrelevant patterns specific to the training data, leading to poorer generalization on unseen data. Therefore, it is important to carefully choose the maximum depth to strike a balance between model complexity and generalizability.

In any decision tree structure, we can see that every node except leaf node consists of 5 parameters (one in each row). The 1st row or simply first parameter(for e.g od280/od315_of_diluted_wines¡=2.19 in Figure 2) signifies that the branches of that node are split on the basis of od280/od315_of_diluted_wine attribute of the dataset and the branch is formed by taking reference value of 2.19 of that attribute. The records whose od280/od315_of_diluted_wine value is less than or equal to 2.19 lies in left branch of the root node while the records whose od280/od315_of_diluted_wine value is greater than 2.19 lies in right branch. This row is present in all other internal nodes but not in leaf nodes in decision tree. This suggests that further splitting is not necessary in leaf nodes. The decision tree has reached a point where it has made the final predictions, and there is no need for additional splits based on that particular feature.

The next or second row (for e.g entropy=1.57 in figure 2) indicates the entropy of the split when dataset is split on the basis of od280/od315_of_diluted_wine attribute. This is present in every node of a tree.

The third row (for e.g samples=119 in figure 2) refers to the number of instances or data points that reached a particular node in the decision tree during the training process. This is present in every node of a tree.

The fourth row in each non-leaf node represents the number of samples or instances that fall into different classes or categories at that particular node.For e.g in figure 2 value=[39,47,33] signifies that within that node, there are 39 samples belonging to Class 0, 47 samples belonging to Class 1, and 33 samples belonging to Class 2.

The last row in each node represents the class label that is most prevalent or has the highest frequency among the samples within that particular node. By identifying the dominant or majority class in a node, the decision tree can make predictions based on the prevalent class label for instances that reach that specific node during the decision-making process. This is present in every node of a tree.

We have constructed a decision tree structure using entropy without any maximum depth. The resulting decision tree structure, as shown in Figure 2, reveals that the total depth of the tree is 4 levels. It means that the tree reached a depth of 4 levels before stopping the splitting process. Similarly we also constructed a tree structure using Gini index without specifying maximum depth, as shown in Figure 5 Then we constructed decision trees using both entropy and gini by limiting the maximum depth to 3 as in figure 3 and 6. This means that the decision trees were constructed to have a maximum depth of three levels, ensuring that the tree does not exceed this depth during the splitting process. Similarly, we also constructed trees for maximum depth set to 2 using both entropy and gini, as shown in figure 4 and 7.

Since entropy and Gini index capture different aspects of impurity, they can lead to different feature selections and splitting criteria at the root node. This can be visualized from figure 2 and 5. In our context, it seems that when constructing a decision tree using entropy as the impurity measure, the root node is determined to be the feature ”od280/od315_of_diluted_wine.” On the other hand, when using the Gini index as the impurity measure, the root node is determined to be the feature ”color_Intensity. Due to the variation in nodes of the two trees which are constructed using entropy and gini, there might also be difference in performance between them. We visualized their performances using confusion matrices [10] and classification reports.

We drew confusion matrices to visualize the prediction of the classifier. Figure 8 and 11 are the confusion matrices for the classifier using entropy and gini respectively when trained without setting any maximum depth. Figure 9 and 12 are the confusion matrices of the tree classifier with maximum depth 3 trained on entropy basis and gini basis respectively. Similarly,figure 9 and 12 are the confusion matrices of the tree classifier with maximum depth 3 trained on entropy basis and gini basis respectively.

We also generated classification reports for all the trees, including the classifier using entropy and Gini impurity measures, as well as the classifier with varying depths. The accuracy results for entropy-based decision trees were evaluated under different conditions. The first case involved a decision tree without any depth limit, resulting in an accuracy of 83%. Secondly when we apply depth limit of three and two, then we obtained accuracy of 88% and 92% respectively. Similarly, accuracy for gini-based decision tree is 97%,97% and 88% for tree with out depth limit, with depth limit of three and with maximum depth two respectively. In a classification report, various parameters are calculated to evaluate the performance of a model. These parameters include recall, f1-score, support, macro average, and weighted average. Together, they provide valuable information about the model's effectiveness.

IV Discussion and Analysis

In the previous sections, we successfully implemented decision tree classifiers using both the entropy and Gini index as impurity measures. Additionally, we generated and visualized decision trees for different depth limits. Initially, we constructed trees without setting any depth limit, allowing them to grow until all the samples were perfectly classified. Subsequently, we set the depth limit to 3 and then to 2, creating more shallow trees. We visualized these trees using both entropy and Gini index to gain insights into their structure and decision-making process. It is important to note that the optimal maximum depth might vary depending on the dataset and problem domain. It is not always the case that the deeper trees perform better than shallow one. From figure 14, we can see that the performance of entropy based tree is high when the maximum depth is set to 2. So, to make the tree most effective further experimentation and tuning may be necessary to choose the correct maximum depth.

When we draw trees without any depth limit, we get that trees are formed with depth 4. It means that the tree was unable to grow further due to the nature of the dataset. In this case, the depth of 4 indicates that the tree has reached a moderate level of complexity, capturing patterns and making decisions based on up to four levels of splits. The decision tree might have considered multiple features and thresholds to create the branching structure. The number of samples at each node helps provide insights into the distribution and composition of the dataset as it flows through the decision tree. By examining these numbers, you can understand how many instances are being considered at different stages of the decision-making process. Analyzing the number of samples at each node can also help you identify nodes with low sample counts, which might indicate regions of the data with sparse coverage or potential areas of uncertainty.

Limiting the depth of the decision tree to 3 levels helps maintain simplicity and enhances interpretability. A shallower tree with fewer levels is easier to understand and visualize, making it more accessible for interpretation and communication.

If we compare accuracy values achieved from entropy and gini, we can see that the Gini index outperformed entropy in our dataset, it suggests that the Gini index was better at capturing the patterns and classifying the instances accurately. However, it's important to note that the performance of different impurity measures can vary depending on the dataset and the specific characteristics of the problem.

Based on the examination of the diagonal values in the confusion matrices, it is observed that the Gini-based decision tree has higher values in the diagonal compared to the entropy-based decision tree. This suggests that the Gini-based tree performs better in terms of overall correct predictions. However, when analyzing the performance of the classifier for each individual class, a more nuanced picture emerges. Specifically, it is observed that the entropy-based trees work better for the prediction of records belonging to class_0, while the Gini-based tree seems to perform better for predicting records belonging to class_1. Additionally, for class_2, the Gini-based trees exhibit better performance compared to the entropy-based trees.

These observations indicate that the performance of the decision trees can vary across different classes. While the Gini-based tree shows better overall performance, the entropy-based tree may be more effective for specific class predictions.

We can observe that not all parameters play a significant role in the construction of a tree. In our dataset, which contains 13 attributes, we can successfully construct a tree using only a subset of key attributes. For instance, in figure 2, We can see a well-constructed tree using only seven attribute. This approach highlights the importance of identifying the essential attributes that contribute significantly to the tree's structure and decision-making process. By focusing on these key attributes, we can simplify the tree construction process and potentially improve its interpretability.

Decision tree provides better visualization and analysis of data. Decision trees offer an intuitive representation of the decision-making process by visually tracing the path through the tree. We can also observe that decision trees perform well even with a limited amount of data. In our case, we had only 119 data points for training, yet our model achieved a satisfactory accuracy of more than 80% in every case. This demonstrates the effectiveness of decision trees in handling small datasets.

In summary, Decision tree classifiers have a distinct advantage over other machine learning algorithms as they provide a clear and easily understandable depiction of decision rules. This algorithm can handle interaction and dependencies between features in the data. Decision tree classifier can give information on which feature are important on classification and which are not. This algorithm has capacity to handle missing data efficiently through surrogate splits. However, this algorithm is prone to overfitting and sensitive to small change. If depth of tree large then this algoritm might be computationally expensive.

V Conclusion

We have successfully implemented the decision tree classification algorithm on the wine dataset, achieving remarkable accuracy in the classification task. Throughout our experiment, we utilized various impurity measures to construct the classification tree and compared their performance using different metrics. In the majority of cases, the Gini impurity measure outperformed the entropy impurity measure, yielding superior results. The decision tree classifier remains a powerful machine learning algorithm with extensive applications in classification tasks. Its versatility and efficacy have made it increasingly popular across various domains. Notably, this algorithm exhibits exceptional accuracy even when trained on relatively small datasets. Its efficiency and low resource requirements further contribute to its ease of use and widespread adoption. By appropriately selecting the depth of the tree, we can achieve optimal results while minimizing computational overhead. Our experiment also highlighted the capability of the decision tree classifier to identify key features that effectively distinguish between classes. This feature selection ability greatly contributes to the construction of the classification tree, ensuring its efficiency and accuracy. In addition to its classification capabilities, the decision tree classifier offers valuable features for data visualization and analysis. By visualizing the constructed tree, we can gain insights into the decision-making process and understand the hierarchy of features that contribute to classification .

Overall, our successful implementation of the decision tree classification algorithm on the wine dataset demonstrates its effectiveness and relevance in modern machine learning applications.

REFERENCES

[1] B. Mahesh, “Machine learning algorithms-a review,” *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[2] I. El Naqa and M. J. Murphy, *What is machine learning?* Springer, 2015.

[3] P. H. Swain and H. Hauska, “The decision tree classifier: Design and potential,” *IEEE Transactions on Geoscience Electronics*, vol. 15, no. 3, pp. 142–147, 1977.

[4] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, pp. 81–106, 1986.

[5] K. Boonchuay, K. Sinapiromsaran, and C. Lursinsap, “Decision tree induction based on minority entropy for the class imbalance problem,” *Pattern Analysis and Applications*, vol. 20, pp. 769–782, 2017.

[6] S. Nembrini, I. R. König, and M. N. Wright, “The revival of the gini importance?” *Bioinformatics*, vol. 34, no. 21, pp. 3711–3718, 2018.

[7] V. Nasteski, “An overview of the supervised machine learning methods,” *Horizons. b*, vol. 4, pp. 51–62, 2017.

[8] J. Kuha, C. Skinner, and J. Palmgren, “Misclassification error,” *Encyclopedia of biostatistics*, vol. 5, 2005.

[9] E. Bisong and E. Bisong, “Introduction to scikit-learn,” *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 215–229, 2019.

[10] R. Susmaga, “Confusion matrix visualization,” in *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM ‘04 Conference held in Zakopane, Poland, May 17–20, 2004*. Springer, 2004, pp. 107–116.

Bishwambhar Dahal is a fourth-year student of Electronics, Communication, and Information Engineering. With a deep fascination for Artificial Intelligence (AI), He is driven by the potential of AI to transform industries and tackle complex challenges. His academic journey has equipped him with a strong foundation in AI concepts, including machine learning and data analysis. He possess a relentless curiosity and is always eager to explore the latest advancements in AI. His goal is to apply his knowledge and skills in AI to make meaningful contributions to research and development, pushing the boundaries of what is possible with intelligent algorithms.(THA076BEI009)

Sirjana Bhatta is a fourth-year student of Electronics, Communication, and Information Engineering with a keen interest in the field of Artificial Intelligence (AI). She possess a strong academic foundation and practical skills in machine learning, deep learning, and data analysis. Her passion lies in leveraging AI to revolutionize industries and solve complex problems. She is a motivated learner, constantly staying updated with the latest advancements in AI. She is seeking opportunities to contribute to AI research and development and make a positive impact on society. (THA076BEI036)

Appendix

A Figures

1 Block Diagram

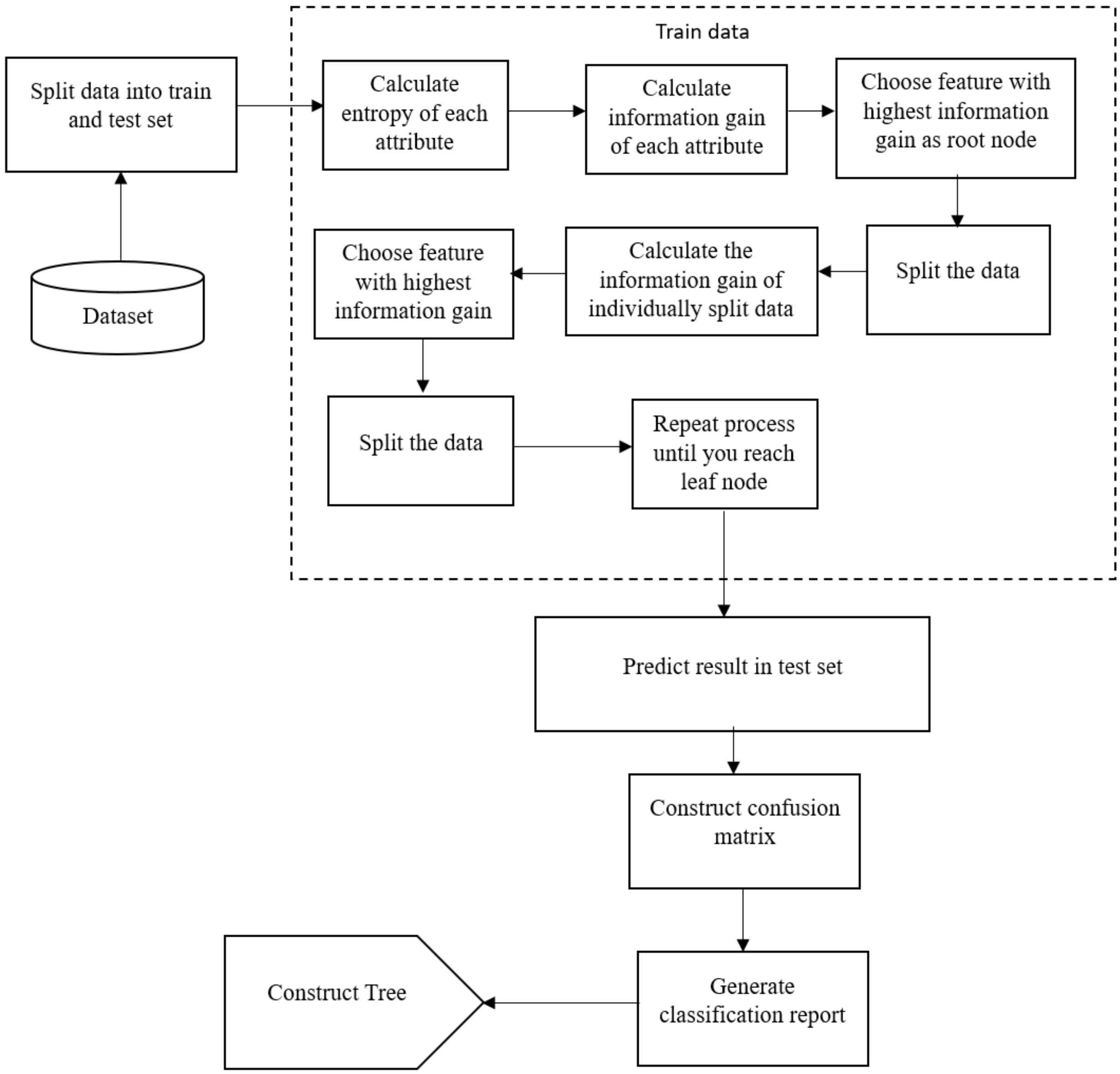

Figure 1: Block Diagram of the system

2 Figure of Trees

Entropy and Gini based Trees

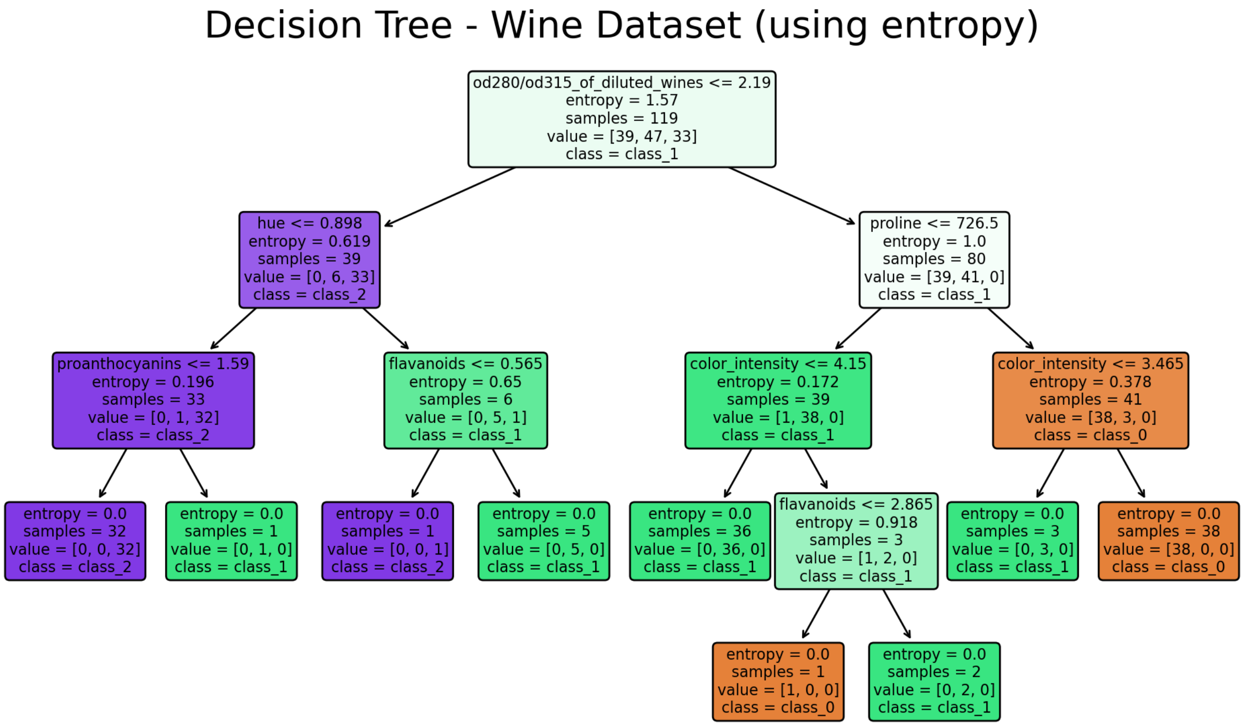

Figure 2: Decision tree based on Entropy

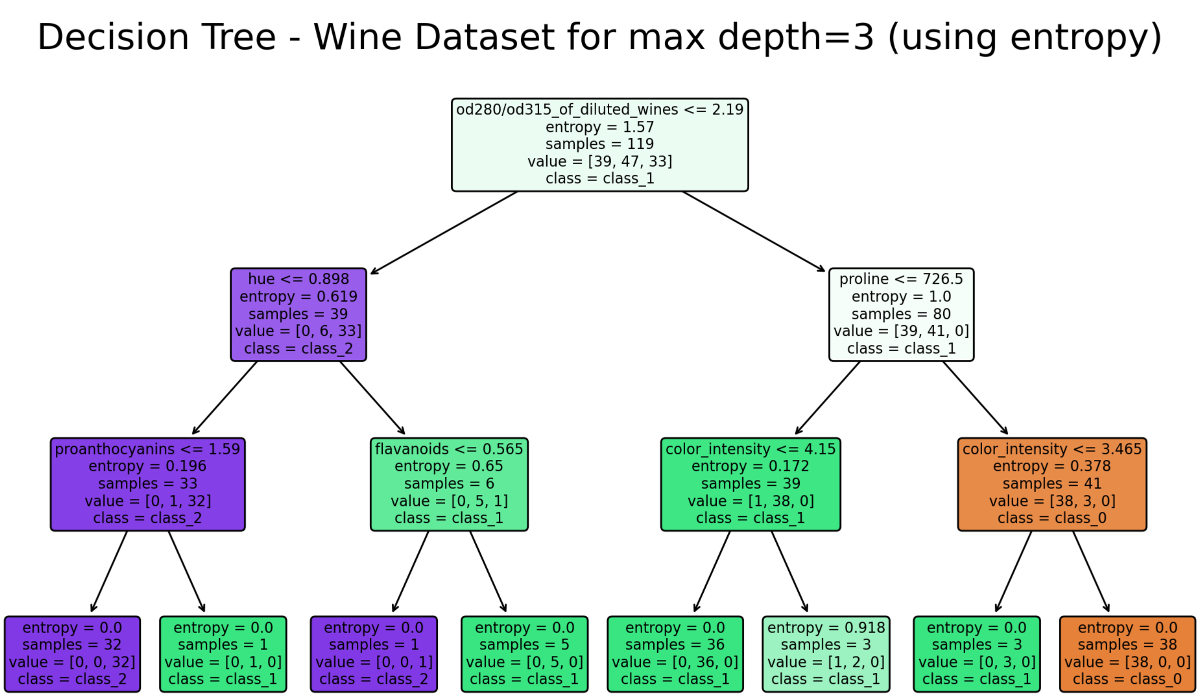

Figure 3: Entropy based Tree with maximum depth=3

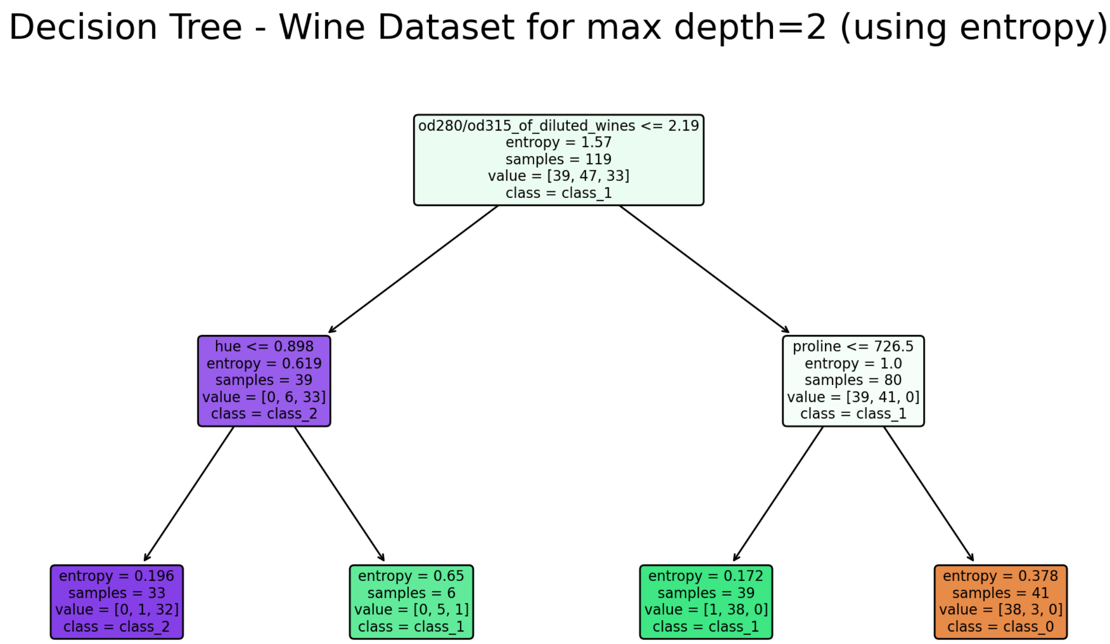

Figure 4: Entropy based Tree with maximum depth=2

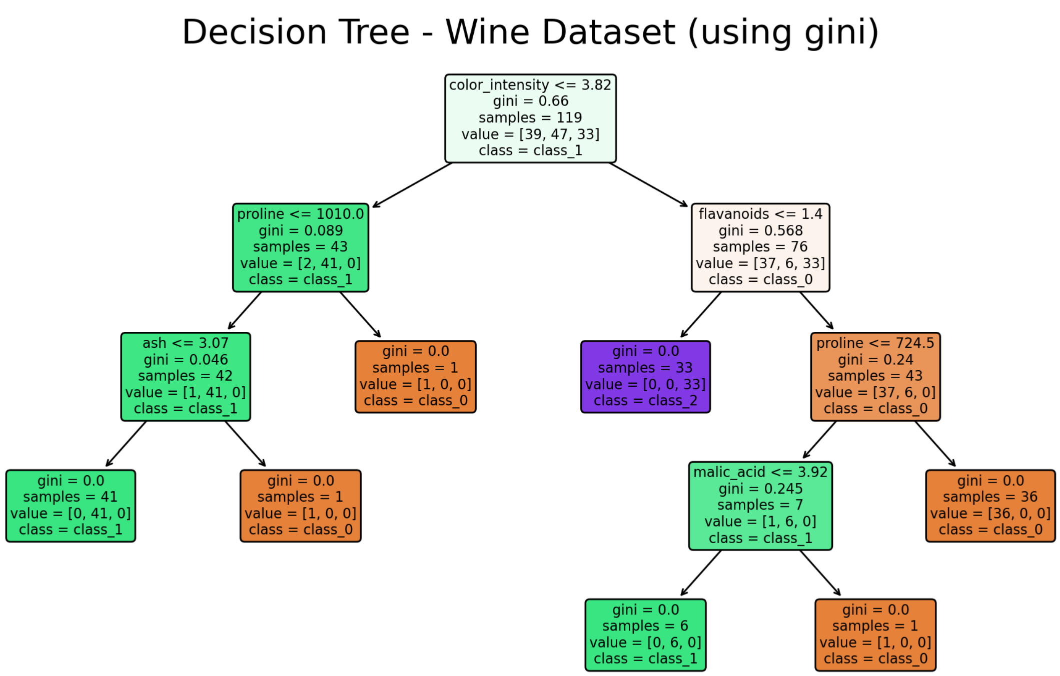

Figure 5: Gini based tree

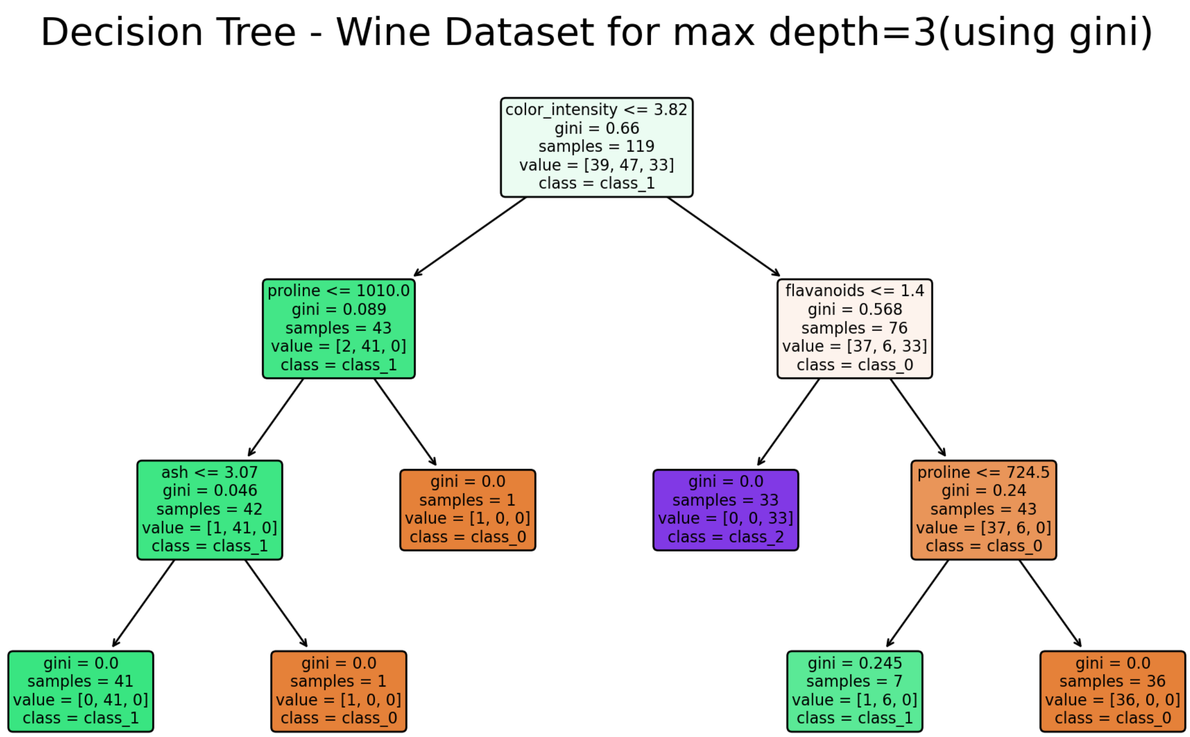

Figure 6: Gini based Tree with maximum depth=3

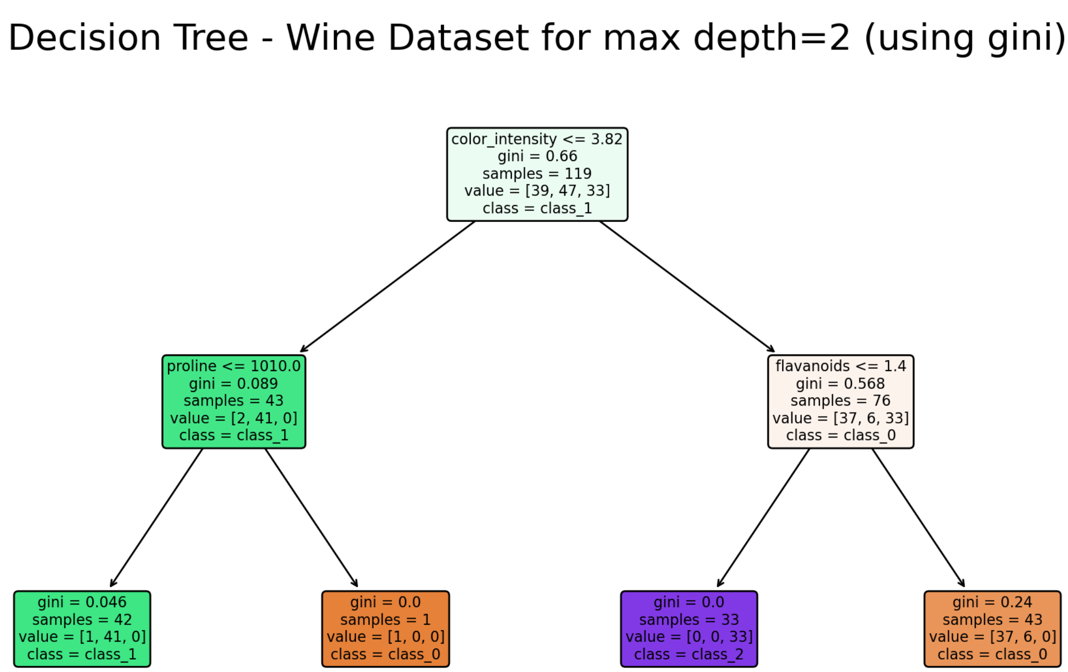

Figure 7: Gini based Tree with maximum depth=2

3 Confusion Matrix

Confusion matrices for Entropy Based Trees

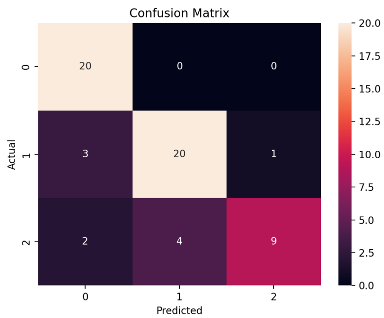

Figure 8: Confusion matrix for Entropy based Tree

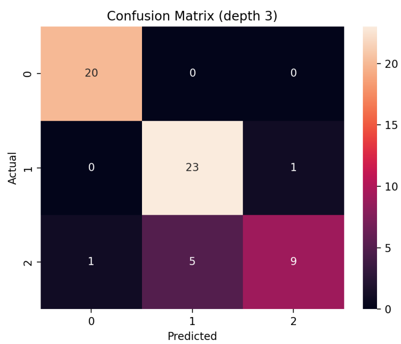

Figure 9: Confusion matrix for max_depth=3

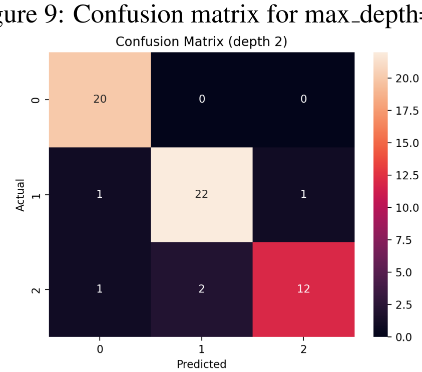

Figure 10: Confusion matrix for max_depth=2

Confusion matrix for Gini Based Trees

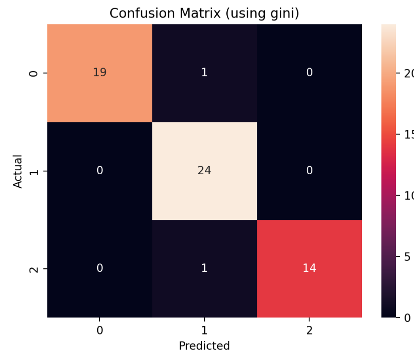

Figure 11: Confusion matrix for gini based tree

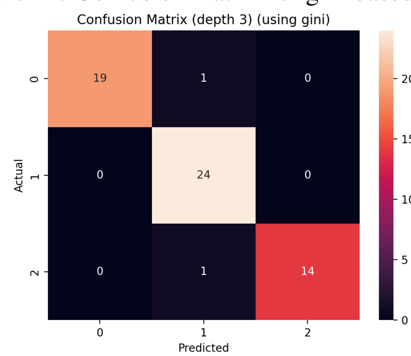

Figure 12: Confusion matrix for max_depth=3

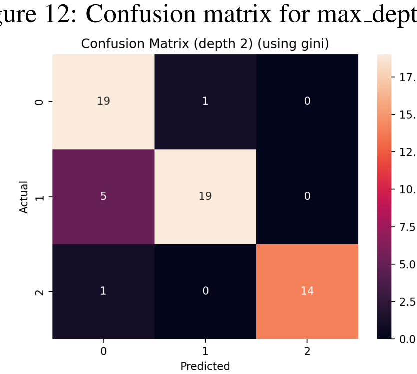

Figure 13: Confusion matrix for max_depth=2

4 Classification Report

Classification report

	precision	recall	f1-score	support
0	0.80	1.00	0.89	20
1	0.83	0.83	0.83	24
2	0.90	0.60	0.72	15
accuracy			0.83	59
macro avg	0.84	0.81	0.81	59
weighted avg	0.84	0.83	0.82	59

Classification report of depth 3

	precision	recall	f1-score	support
0	0.95	1.00	0.98	20
1	0.82	0.96	0.88	24
2	0.90	0.60	0.72	15
accuracy			0.88	59
macro avg	0.89	0.85	0.86	59
weighted avg	0.89	0.88	0.87	59

Classification report of depth 2

	precision	recall	f1-score	support
0	0.91	1.00	0.95	20
1	0.92	0.92	0.92	24
2	0.92	0.80	0.86	15
accuracy			0.92	59
macro avg	0.92	0.91	0.91	59
weighted avg	0.92	0.92	0.91	59

Figure 14: Classification report for Entropy based Tree

Classification report (using gini)

	precision	recall	f1-score	support
0	1.00	0.95	0.97	20
1	0.92	1.00	0.96	24
2	1.00	0.93	0.97	15
accuracy			0.97	59
macro avg	0.97	0.96	0.97	59
weighted avg	0.97	0.97	0.97	59

Classification report of depth 3 (using gini)

	precision	recall	f1-score	support
0	1.00	0.95	0.97	20
1	0.92	1.00	0.96	24
2	1.00	0.93	0.97	15
accuracy			0.97	59
macro avg	0.97	0.96	0.97	59
weighted avg	0.97	0.97	0.97	59

Classification report of depth 2 (using gini)

	precision	recall	f1-score	support
0	0.76	0.95	0.84	20
1	0.95	0.79	0.86	24
2	1.00	0.93	0.97	15
accuracy			0.88	59
macro avg	0.90	0.89	0.89	59
weighted avg	0.90	0.88	0.88	59

Figure 15: Classification report for gini based Tree