Use of Decision Tree on Wine Quality Analysis (June 2024)

Kristina Ghimire (THA077BCT023)[1], Punam Shrestha (THA077BCT038)[1]

[1]Department of Electronics and Computer Engineering, IOE, Thapathali Campus, Kathmandu 44600, Nepal
Corresponding author: Kristina Ghimire(ghimirekristina10@gmail.com)

ABSTRACT

This paper explores the effectiveness of decision tree models in predicting wine quality based on physicochemical attributes. Through extensive experimentation and analysis, including data preprocessing, feature selection, and parameter optimization, we achieved high predictive accuracy. Our findings highlight the importance of addressing data integrity issues such as duplicates and outliers, categorizing quality scores, and handling class imbalance to enhance model performance. The decision tree model, particularly optimized with a maximum depth of 19 and cost complexity pruning, achieved an accuracy of 88%. This study underscores the robustness of decision trees in complex classification tasks, offering insights into effective predictive modeling for wine quality assessment.

INDEX TERMS Cost Complexity Pruning, Data Preprocessing, Decision Tree, Feature Selection.

I INTRODUCTION

Decision tree models are powerful tools for making decisions by representing choices and their possible outcomes in a clear, tree-like structure. Each branch in the tree represents a choice between different options, and each leaf (endpoint) signifies a final decision or classification. This intuitive format makes decision trees easy to understand and visualize, which is particularly useful for explaining decisions to others.

In a decision tree, decision nodes signify tests on a single feature, and each branch from a node indicates the outcome of that test. The primary objective is to split the data into groups that are as homogeneous (pure) as possible. To achieve this, decision trees employ criteria such as Gini impurity, entropy, and information gain. The process of splitting the data continues until it meets specific stopping criteria, such as a maximum tree depth or a minimum number of samples per leaf.

For classification tasks, an instance is classified by moving from the root node to a leaf node based on its features. Each decision node tests one feature of the instance, and the process continues down the tree until a leaf node is reached. The leaf node then provides the final classification. For regression tasks, the leaf node offers an average value of the target variable for all instances that reach that leaf, thereby providing a prediction.

Decision trees have several advantages. They can capture nonlinear relationships and handle both numerical and categorical features with minimal preprocessing. Moreover, their simplicity and transparency make them excellent for interpretability and explaining decision-making processes. However, decision trees are prone to overfitting, where they become too complex and capture noise in the training data. This complexity can result in poor generalization to new data. Additionally, decision trees can be unstable, as small changes in the data can lead to significant changes in the tree structure. Trees might also favor features with many levels, leading to biased splits.

Despite their limitations, decision trees are widely used in many fields. They are helpful for tasks like detecting spam emails, predicting when customers will leave a service, and diagnosing medical conditions. For regression tasks, decision trees can predict things like housing prices or stock prices. They are also useful for identifying important features in a dataset, which helps in selecting the most relevant data and understanding relationships within the data.

In summary, decision trees are a flexible and valuable tool in machine learning. They are simple to understand yet powerful for making decisions and predictions. Their ability to work with different types of data and produce clear, understandable results makes them a popular choice for many practical applications.

II RELATED WORK

Decision tree is widely studied and used in classification and regression problems. P. Apalasamy et al. [1] explores how to use decision trees, focusing on the Iterative Dichotomiser 3 (ID3)algorithm, in the R programming language. Decision trees are important in machine learning for classification and regression because they are easy to understand and interpret, making them useful for analysis and decision-making. This document provides

a detailed guide on how to implement decision trees in R, including information on relevant packages like rpart, party, and tree. It highlights the benefits of decision trees, such as their ability to handle both numerical and categorical data with little preprocessing, and their clear and easy-to-understand results. The paper also discusses common problems like overfitting and offers solutions such as pruning and advanced methods like Random Forests and gradient-boosted trees to improve performance. The paper looks at how decision trees can be applied in various fields such as finance, healthcare, and marketing, demonstrating their effectiveness in predictive modeling, classification, and regression tasks. It includes practical examples and case studies to show real-world applications of decision trees in R, with datasets, code examples, and result interpretations. Overall, this paper is a thorough resource for understanding and using decision trees, particularly the ID3 algorithm, in R. It provides valuable insights into both theory and practice. For more detailed information, examples, and code, refer to the original documents.
Dragana Radosavljevic et al. [2] explores how data mining techniques can be used to predict the quality of the wine. Data mining involves analyzing large sets of data to find patterns and make predictions. In this study, the authors collected a dataset of various wine characteristics, such as acidity, sugar content, and pH levels, and used these features to predict wine quality. The goal was to determine which attributes most significantly affect wine quality and to create a model that can accurately forecast it. By applying different data mining methods, such as classification and regression, they aimed to build a predictive model that winemakers and consumers could use to assess the potential quality of a wine before tasting it. The authors found that certain features, like alcohol content and acidity, played a more crucial role in determining wine quality. They also compared various algorithms to find the most effective one for this task. This research is important because it shows how modern technology and data analysis can aid traditional industries like winemaking. By better understanding what makes a good wine, producers can improve their processes and offer higher-quality products. The study's findings could lead to more efficient and reliable ways of predicting wine quality, ultimately benefiting both producers and consumers. This approach could also be applied to other types of food and beverages, demonstrating the broader potential of data mining in the food industry. Overall, the paper highlights the intersection of technology and agriculture, showcasing how data science can enhance traditional practices and contribute to better products and consumer experiences.
These works contribute to the understanding and use of decision trees to make more accurate predictions based on given features.

III METHODOLOGY

A DATASET DESCRIPTION

The performance evaluation of target classification using the Decision Tree involved conducting experiments on the Wine Quality dataset. Two datasets related to red and white wine samples are used with decision trees to classify the wines or predict their quality. The categories in these datasets are ranked but not evenly distributed, with a higher number of average wines compared to excellent or poor ones. These datasets can also be used to explore feature selection and outlier detection. There are 1599 Red Wine datasets. 11 features are used to predict the wine quality score ranging from 0 to 10. The use of the Decision Tree on this dataset was done to evaluate the prediction of the decision tree and different ways to improve the accuracy.
The Red Wine Quality dataset used in the study includes 11 features that help predict the quality of red wine. Here are the features explained:

1. **Fixed Acidity:**

   This measures the acids in wine that do not evaporate quickly. These acids give the wine its tart and sour taste.

2. **Volatile Acidity:**

   This measures the amount of acetic acid in wine, which can give it a vinegar taste if too high.

3. **Citric Acid:**

   A small amount of citric acid can add freshness to wine. It is often used as a preservative.

4. **Residual Sugar:**

   This measures the sugar left after fermentation stops. Wines with higher residual sugar are sweeter.

5. **Chlorides:**

   This measures the salt content in wine. High chloride levels can make the wine taste salty.

6. **Free Sulfur Dioxide (SO2):**

   This measures the amount of SO2 that is not bound and is available to act as a preservative to prevent spoilage.

7. **Total Sulfur Dioxide (SO2):**

   This measures the total amount of SO2, including both free and bound forms, in the wine.

8. **Density:**

   This measures the density of wine. It can provide clues about the alcohol and sugar content in the wine.

9. **pH:**

   This measures the acidity or alkalinity of the wine. A lower pH means higher acidity.

10. **Sulphates:**

    This measures the amount of sulfur compounds in the wine. Sulphates can contribute to the wine's bitterness and act as preservatives.

11. **Alcohol:**

    This measures the alcohol content in the wine, usually expressed as a percentage.

B THEORETICAL FORMULATION

Decision trees are a popular machine learning algorithm utilized for both classification and regression tasks. They operate by partitioning the data into subsets based on the values of input features, forming a tree-like model of decisions. The structure of a decision tree consists of various elements, including nodes, edges, and leaves, which collectively represent the decision-making process. Nodes are the points where data is split based on the values of an attribute. The topmost node in a decision tree is known as the Root Node, which initiates the process. Internal Nodes represent the features used for splitting the data, while Leaf Nodes (or terminal nodes) represent the final output, such as a class label in classification tasks or a value in regression tasks. Edges are the branches that connect nodes, illustrating the outcomes of decisions made at each node. The process of constructing a decision tree involves selecting the best attribute for splitting the data, which aims to maximize the separation of the data concerning the target variable. This process, known as Splitting, divides a node into two or more sub-nodes based on certain conditions. The selection of the best attribute is typically based on criteria such as Information Gain or Gini Index.

The construction of decision trees involves the following steps:

Step-1: **Select the Best Attribute**: Choose the attribute that best separates the training examples based on a specific criterion (e.g., Information Gain, Gini Index).

Step-2: **Split the Data**: Divide the dataset into subsets where each subset contains data with the same value of the selected attribute.

Step-3: **Recursive Process**: Apply steps 1 and 2 recursively to each subset until one of the stopping conditions is met:

i) All instances within a subset share the same class.

ii) There are no remaining attributes to split.

iii) A predefined stopping criterion (e.g., maximum tree depth) is reached.

1 Some Decision Tree Algorithms

Several decision tree algorithms are commonly used:

ID3

ID3 (Iterative Dichotomiser 3) is a classic decision tree algorithm primarily used for classification tasks. It begins by calculating the entropy of the dataset's target variable, which quantifies the uncertainty or impurity of the data. A higher entropy indicates more randomness and lack of predictability in the dataset. Next, ID3 computes the information gain for each attribute. Information gain measures how much a particular attribute contributes to reducing uncertainty about the target variable. The attribute that yields the highest information gain is selected as the root node for the decision tree, initiating the splitting process. The dataset is then partitioned into subsets based on the values of the selected attribute. This process of attribute selection and dataset splitting is applied recursively to each subset, treating each subset as a new dataset. The algorithm continues this recursive process until one of the stopping conditions is met: either all instances in a subset belong to the same class, no attributes remain to split further, or the dataset for a particular node is empty. ID3 constructs decision trees that are easy to interpret due to their hierarchical structure, making it suitable for tasks where understanding the reasoning behind predictions is crucial. However, it can be sensitive to noisy data and may overfit if not pruned or if the dataset is imbalanced.

C4.5

C4.5 is an extension of ID3 that can handle both categorical and continuous data. It uses a Gain Ratio for attribute selection and prunes trees after creation to remove unnecessary branches.

CART (Classification and Regression Trees)

CART (Classification and Regression Trees) is a versatile algorithm used for both classification and regression tasks in machine learning. It operates by recursively partitioning the dataset into subsets, aiming to maximize the homogeneity of the target variable within each subset. For classification tasks, CART uses the Gini Index as a measure of impurity to evaluate the quality of splits. The Gini Index quantifies how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. The algorithm selects the attribute and split point that minimizes the Gini Index, effectively separating the classes within the data. In regression tasks, CART uses the Mean Squared Error (MSE) to assess the quality of splits. MSE calculates the mean of the squared differences between predicted values and actual values within a subset. By iteratively choosing splits that minimize MSE, CART constructs a decision tree that predicts continuous target variables. Overall, CART's flexibility in handling both categorical and continuous data, along with its ability to produce interpretable decision trees, makes it a widely

used algorithm in various machine learning applications. In summary, decision trees are a versatile and intuitive method for making predictions in machine learning. They hierarchically structure decisions, allowing for clear and interpretable models. By recursively splitting data based on attributes that best separate the data concerning the target variable, decision trees can effectively model complex relationships in data.

2 Techniques to prevent Overfitting

Overfitting is a significant challenge in decision tree algorithms, primarily due to their ability to create highly complex models that fit the training data too closely, thereby reducing generalization to unseen data. To address overfitting, two main techniques are commonly employed: Early Stopping and Pruning.

Early Stopping

Early stopping involves limiting the growth of the decision tree during the training process to prevent it from becoming overly complex. Several strategies can be used for early stopping:

i. Depth Limitation:

   Restricting the maximum depth of the tree. A shallow tree is less likely to overfit as it captures simpler patterns in the data.

ii. Impurity Threshold:

   Stopping the growth of the tree when nodes no longer provide a sufficient decrease in impurity (e.g., Gini impurity or entropy). This prevents the tree from splitting further when additional splits do not significantly improve the model.

iii. Minimum Samples in a Node:

   Setting a threshold on the minimum number of samples required to split a node. Nodes with fewer samples than this threshold are not split further, thereby avoiding overly specific decisions based on too few instances.

Pruning

Pruning involves growing the full decision tree first and then removing or collapsing nodes that provide little predictive power, thus simplifying the final tree structure. The main technique used for pruning is: - **Cost Complexity Pruning**: Also known as ”weakest link pruning” or ”alpha pruning,” this method involves systematically varying a complexity parameter (often denoted as alpha) to balance the tree’s complexity against its ability to accurately predict the target variable. Nodes are pruned based on whether removing them improves the overall model performance on a validation set or through cross-validation.

Both early stopping and pruning aim to improve the decision tree’s ability to generalize to new data by preventing it from memorizing the training set. These techniques help achieve a more balanced trade-off between model complexity and predictive accuracy, thereby enhancing the robustness and reliability of decision tree models in practical applications.

C MATHEMATICAL FORMULAE

1 Criteria to determine the best split

Gini Index

The Gini index quantifies the impurity of a decision tree node, ranging from 0 (pure node) to 0.5 (maximum impurity). Lower values indicate effective splits, enhancing the tree’s ability to separate classes accurately for precise predictions.

Entropy

Entropy measures uncertainty in data, ranging from 0 (pure node) to 1 (maximum uncertainty). Minimizing entropy at each split in decision trees ensures effective classification by reducing uncertainty and creating purer child nodes.

Information Gain

Information gain measures entropy reduction when splitting a dataset based on an attribute. Higher information gain indicates more effective splits, leading to better classification accuracy and purer child nodes in decision tree construction.

Classification Error

Classification error quantifies incorrect predictions relative to total predictions made by a model. Minimizing classification error is crucial for developing accurate classification models that perform effectively in practical applications.

2 Performance Metrics

Precision

Precision measures the accuracy of positive predictions, reflecting the ratio of true positives to total predicted positives. High precision indicates reliable positive predictions with few false positives, essential for applications sensitive to incorrect classifications.

Recall

Recall (sensitivity) measures the proportion of true positive predictions relative to all actual positives. Balancing recall with precision is crucial; high recall ensures identifying most positives but may increase false positives, impacting overall model performance.

F1 Score

The F1 score combines precision and recall into a single metric using their harmonic mean. A higher F1 score indicates a balanced performance between precision and recall, making the model suitable for applications requiring both accurate identification of positives and low false positives.

Macro Average

Macro average calculates the mean performance metric (e.g., precision, recall, F1 score) across all classes in multi-class classification, treating each class equally. A higher macro average signifies consistent performance across diverse categories, indicating strong generalization capabilities.

Weighted Average

Weighted average considers class distribution by calculating a mean weighted by instances in each class. Higher weighted averages indicate robust model performance across all classes, demonstrating the ability to handle imbalanced data and make accurate predictions.

Accuracy

Accuracy measures correctly predicted instances relative to the total number of instances in a dataset. High accuracy indicates effective model performance suitable for real-world applications, accurately predicting both positive and negative instances.

D SYSTEM BLOCK DIAGRAM

In the system block diagram shown in figure 1, there is a dataset block that is divided into two parts: training data and test data. In the training data, decision trees are generated by selecting the best attribute in dataset X using ASM (Attribute Selection Measure) by calculating the Gini index or gain ratio, and by calculating the information gain. Afterward, the dataset X is divided into smaller subsets. This process is recursively repeated for each child node, selecting the best attribute in the dataset each time. In the test data, the model is evaluated to check if the data is overfitting or not. Finally, performance metrics are calculated using the formulas for accuracy and precision.

E DATA PREPROCESSING PIPELINES

1 Imputing Missing Values

A dataset often contains missing values, which can hinder the analysis as they do not contribute any meaningful information. To address this issue, missing values should be imputed using various methods such as filling them with the mean, median, or mode (highest frequency) of the available data. Imputing missing values ensures that the dataset remains complete and allows for more accurate and reliable statistical analyses, enhancing the overall quality and utility of the data.

2 Treating Outliers

Outliers are values that fall far outside the typical range of data points in a dataset. They can skew statistical analyses and lead to misleading results. To identify and treat outliers, several methods can be employed. One common approach involves using z-scores, where we calculate how many standard deviations a data point is away from the mean. Points beyond a certain threshold, often set at around ±3 standard deviations, are considered outliers and can be removed or adjusted. Another method involves using boxplots, which visually display the spread of data and help identify values that lie significantly beyond the whiskers, determined by the InterQuartile Range (IQR). Data points outside the calculated bounds of the IQR are typically treated as outliers and either corrected or excluded from further analysis to ensure the dataset remains reliable and representative.

3 Categorization of Numerical Target

Categorizing a numerical target variable involves transforming its continuous values into discrete categories or classes. This simplification is beneficial for predictive tasks, as it reduces complexity compared to predicting from a larger number of distinct values. For instance, predicting among three categorical classes is generally simpler than predicting from ten numerical classes. This categorization strategy is particularly useful in methods like clustering or decision trees, where handling discrete categories can streamline processing and improve prediction accuracy. By converting a regression problem (predicting numerical values) into a classification problem (predicting categories), categorization facilitates a clearer interpretation of results and enhances the overall efficiency of predictive models.

4 Rectifying class imbalance

In datasets, classes aren't always balanced. Some classes might have many more examples than others, causing models to favor those classes during predictions due to their sheer numbers. To fix this, we can balance the classes by resampling the data. Upsampling involves increasing the number of instances in the minority class, while downsampling decreases instances in the majority class. This ensures that each class contributes equally to the model training, improving its ability to predict all classes accurately, not just the majority ones.

5 Feature Selection Using Elastic Net

Feature selection and dimensionality reduction are crucial for simplifying models and reducing computational complexity and cost. Various methods exist for feature selection, such as PCA analysis, Lasso regularization, Elastic Net regularization, and more. PCA analysis summarizes multiple features into a smaller set of principal components, which may not directly preserve the original feature values but captures their variance effectively. Regularization techniques like Lasso and Elastic Net help in selecting the most relevant features by penalizing less important ones based on their coefficients. However, it's essential to consider that sometimes all features may be important, necessitating correlation analysis to understand relationships between features and ensure comprehensive model coverage. This approach ensures that models are efficient,

interpretable, and perform optimally without unnecessary computational burden.

6 Normalization of data

Normalization involves scaling numerical data into a standardized range. There are several normalization methods. Z-score normalization adjusts data to have a mean of 0 and a standard deviation of 1. On the other hand, min-max scaling transforms data to fit within a fixed range, typically between 0 and 1. These techniques ensure that data from different scales can be compared directly and are essential for many machine learning algorithms to perform effectively and efficiently.

7 Discretization of Continuous data

Using continuous data directly in classification tasks, such as with decision trees, isn't always ideal. Continuous data can vary across a wide range of values, making predictions uncertain and potentially less accurate. To address this, we can discretize continuous data by dividing it into groups or bins. This process, known as discretization, allows us to categorize continuous variables into distinct groups based on specified criteria, such as equal width or equal frequency bins. By doing so, we simplify the data representation and make it more suitable for classification tasks, improving the interpretability and effectiveness of models like decision trees.

IV INSTRUMENTATION DETAILS

Python is a high-level, versatile programming language known for its simplicity and readability. Python is an interpreted language, which means code is executed line by line, allowing for quick testing and debugging. Variables in Python do not require an explicit declaration to reserve memory space. The declaration happens automatically when a value is assigned to a variable. Jupyter Notebook is a web-based interactive development environment (IDE) that allows you to create and share live code, equations, visualizations, and narrative text documents. It supports over 40 programming languages, including Python, and is widely used in data science for its flexibility and ease of use. Matplotlib supports a wide variety of plots, including line plots, scatter plots, bar charts, histograms, pie charts, and more. It provides a wide range of plotting functions and customization options, making it suitable for creating publication-quality figures for data analysis and presentation. Seaborn is a powerful Python data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. It is particularly well-suited for visualizing complex datasets and creating aesthetic statistical plots. Seaborn is designed to work seamlessly with pandas' data structures and is commonly used for statistical data visualization.

V EXPERIMENTAL RESULTS

This study aims to evaluate the factors that influence wine quality based on physicochemical tests. There are two types of wine in this dataset but we will evaluate the quality of red wine. This dataset does not contain any missing values.

1 Duplicate Values Removal

Evaluating the dataset, 240 data are duplicated. So, we remove the duplicated data from the dataset reducing it from 1599 to 1359.

2 Outliers Removal

There are some outliers in the dataset. So we visualized a boxplot to check for outliers and remove them. After evaluating the boxplot of each feature, we calculated the InterQuartile Range (IQR) of each feature to determine the lower bound and upper bound of the data. The data less than the lower bound and upper bound are assumed as outliers and removed from the original dataset. The plot before removing outliers and after removing outliers is shown in Figure 2 and 3 respectively. After removing outliers, the dataset was reduced from 1359 to 1019.

3 Categorization of Wine Quality Score

The target for this dataset is wine quality score which ranges from 0 to 10. Using a decision tree for 10 classes is tedious. Among them, only 6 score datasets are available. Among 1599 instances, 42.59% are scored 5, 39.9% are scored 6, 12.45% scored 7 and the remaining are labelled 4, 8, 3. To make the prediction simple, we categorized the scores into bad, neutral, and good quality as shown in table 2.

4 Resampling

After the removal of outliers, bad quality contains 470 data whereas neutral quality contains 423, and good quality only contains 126. So, we need to resample the data. The plot before resampling classes with the majority class and after resampling is shown in figure 4 and 5.

5 Feature Selection Using Elastic Net

We use the Elastic Net Regularization method to select important features from the dataset. We convert the target classes to numbers using Label Encoding. Then, we fit the features and target to get the important features. The selected features are 'volatile acidity', 'chlorides', 'total sulfur dioxide', 'sulphates', and 'alcohol'. The confusion matrix based on feature selection is displaced in 6. We also performed a decision tree without removing any features which shows results in 7.

6 Discretization of Continuous data

We observed that all 5 features are continuous data. So, we performed discretization using 5 number of bins.

7 Training and Testing Decision Tree

The dataset was divided into 80% training data and 20% testing data. Using the DecisionTreeClassifier with the criteria "entropy" and "log loss," the decision tree achieved an accuracy of 84% without feature selection and 70% with feature selection. When using the "gini" criterion, the decision tree achieved an accuracy of 87% without feature selection and 70% with feature selection. The optimal max depth value was determined by evaluating the accuracy of the testing data with decision trees of max depths ranging from 1 to 20. It was observed that a max depth of 19 performed the best, with an accuracy of 87%, as shown in Figure 8. Grid Search with cross-validation was used to evaluate the minimum sample split, resulting in an accuracy of 85% with a minimum sample split of 5. Additionally, the cost complexity pruning method was used to enhance the performance of the decision tree. Different alpha values were evaluated, and an alpha value of 0.000877 resulted in an accuracy of 88%, as shown in Figure 9. However, when using parameters such as the "gini" criterion, max depth of 19, minimum samples split of 5, and cost complexity pruning, the accuracy was only 85

The classification report in Table 2 shows the precision, recall, and F1 scores for the classes "bad," "good," and "neutral" wines. The accuracy achieved was 88%, with macro and weighted averages of 0.88 for precision, recall, and F1 scores. Specifically, the precision for "bad" wines was 0.87, with a recall of 0.81 and an F1 score of 0.84. For "good" wines, the precision was 0.96, the recall was 0.96, and the F1 score was 0.96. For "neutral" wines, the precision was 0.81, the recall was 0.87, and the F1 score was 0.84.

Table 3 presents the accuracy and F1 scores for different methods used in the decision tree. The "entropy" and "log loss" criteria both achieved an accuracy and F1 score of 0.84. The "gini" criterion without any adjustments resulted in an accuracy and F1 score of 0.87. Using a max depth of 9, the "gini" criterion maintained an accuracy and F1 score of 0.87. When a minimum impurity decrease of 0.005 was applied, the accuracy and F1 score remained at 0.87. With a minimum sample split of 5, the "gini" criterion achieved an accuracy and F1 score of 0.85. Finally, with cost complexity pruning and an alpha value of 0.00069, the accuracy and F1 score improved to 0.88.

The results demonstrate the effectiveness of different decision tree parameters and methods in predicting wine quality, with the highest accuracy achieved through cost complexity pruning.

For easy visualization of the decision tree, we limited the depth to only 3 which is shown in figure 10.

The best accuracy is achieved using the parameters "gini" and cost complexity pruning which is shown in table **??**

VI DISCUSSION AND ANALYSIS

In this study, we embarked on a comprehensive exploration of using a decision tree model to predict wine quality based on physicochemical attributes. Our journey began with meticulous data preprocessing, a crucial step in ensuring the quality and reliability of our model's predictions. Initially, we addressed duplicate values within the dataset, identifying and removing 240 instances that were exact duplicates. This refinement reduced our dataset from 1599 to 1359 entries, ensuring that each data point was unique and contributing effectively to our analysis.

Following the removal of duplicates, we turned our attention to outliers, another potential source of model distortion. Using boxplots for visual inspection across each feature, we pinpointed outliers that could skew our analysis. Employing the InterQuartile Range (IQR) method, we defined upper and lower bounds for each feature and subsequently removed data points that fell outside these boundaries. This rigorous outlier removal process reduced our dataset further to 1019 instances, enhancing the robustness of our subsequent analyses.

A pivotal aspect of our study was the categorization of wine quality scores. Originally spanning a continuous range from 0 to 10, we simplified these scores into three distinct categories: bad, neutral, and good. This categorization not only streamlined our predictive task but also facilitated clearer insights into how physicochemical attributes correlate with overall wine quality. For instance, we observed that a significant proportion of wines fell within the neutral category, reflecting a balanced distribution in the dataset.

The issue of class imbalance then became pertinent, particularly after outlier removal. Our dataset exhibited varying frequencies across the categorized quality scores: bad, neutral, and good. With bad quality comprising 470 instances, neutral 423, and good only 126, a clear imbalance existed. To rectify this, we implemented data resampling techniques, ensuring that each quality category contributed proportionally to our model's training and testing phases. This rebalancing step was crucial in preventing the model from favoring the majority class and thereby achieving more equitable predictions across all quality categories.

Feature selection using Elastic Net regularization played a pivotal role in identifying the most influential attributes for predicting wine quality. Through this method, we identified 'volatile acidity,' 'chlorides,' 'total sulfur dioxide,' 'sulphates,' and 'alcohol' as the primary predictors. However, it was noteworthy that limiting feature selection to these variables alone resulted in a slight decrease in model accuracy to 70%. This observation underscored the importance of considering all 11 features during model training, where a more inclusive approach led to an improved accuracy of 84

Determining the optimal parameters for our decision tree model was another critical aspect of our study. Through systematic evaluation, we identified that a maximum depth of 19 and utilizing the ”gini” criterion provided optimal performance, yielding an accuracy of 87%. Moreover, employing cost complexity pruning with an alpha value of 0.000877 further refined our model, boosting accuracy to an impressive 88%. These findings highlight the efficacy of parameter tuning in enhancing model performance, ensuring that our decision tree could effectively capture and generalize patterns in the data.

The evaluation metrics employed, including accuracy, precision, recall, and F1 score, provided comprehensive insights into the decision tree's classification capabilities. Notably, the model demonstrated robust performance across all quality categories, achieving high precision and recall scores. For instance, precision scores of 0.87 for ”bad” wines, 0.96 for ”good” wines, and 0.81 for ”neutral” wines underscored the model's ability to correctly identify instances within each category. This balanced performance was further validated through confusion matrices, which illustrated the model's accurate predictions across different quality classes.

In conclusion, our study underscores the critical role of meticulous data preprocessing and parameter optimization in maximizing the predictive accuracy of decision tree models. By systematically addressing data quality issues, selecting pertinent features, and fine-tuning model parameters, we significantly enhanced our model's ability to predict wine quality based on physicochemical attributes. The insights gained from this study not only contribute to the field of wine quality assessment but also provide a framework for leveraging decision tree models effectively in similar predictive analytics contexts. Future research could explore additional ensemble methods or deep learning approaches to further refine predictive accuracy and generalize findings across diverse datasets.

VII CONCLUSION

In this study, we conducted a comprehensive examination of the decision tree model's efficacy in predicting wine quality based on physicochemical attributes. Our analysis commenced with rigorous data preprocessing, a critical step aimed at ensuring data integrity and enhancing model performance. This involved the removal of duplicate values, which reduced our initial dataset from 1599 to 1359 instances. Subsequent outlier detection and removal, guided by boxplot analysis and the InterQuartile Range (IQR) method, further refined the dataset to 1019 instances, mitigating potential biases and ensuring robust analysis.

A significant aspect of our preprocessing efforts included categorizing wine quality scores into distinct classes—bad, neutral, and good—to simplify the predictive task and improve interpretability. Addressing class imbalance through strategic resampling techniques ensured equitable representation across quality categories, crucial for training a balanced and effective model. Feature selection using Elastic Net regularization identified 'volatile acidity,' 'chlorides,' 'total sulfur dioxide,' 'sulphates,' and 'alcohol' as pivotal predictors, highlighting their substantial influence on wine quality predictions.

To optimize our decision tree model, we systematically tuned parameters such as maximum depth and alpha for cost complexity pruning. Through iterative testing, we determined that a maximum depth of 19 and an alpha value of 0.000877 yielded optimal performance, culminating in an impressive accuracy of 88%. Evaluation metrics including accuracy, precision, recall, and F1 score provided a comprehensive assessment of our model's classification capabilities. Notably, precision scores of 0.87 for ”bad” wines, 0.96 for ”good” wines, and 0.81 for ”neutral” wines underscored the model's ability to effectively discriminate between different quality classes.

Overall, our study highlights the pivotal role of rigorous preprocessing and parameter optimization in maximizing the predictive accuracy of decision tree models. By integrating these methodologies, we not only improved model performance but also demonstrated the model's robustness in handling complex and diverse datasets. The findings underscore the importance of leveraging all pertinent features and refining model parameters to achieve high accuracy and reliable predictions in predictive analytics. Future research could explore additional modeling techniques or incorporate domain-specific insights to further enhance predictive capabilities in wine quality assessment and beyond.

VIII APPENDICES

A Figures and Tables

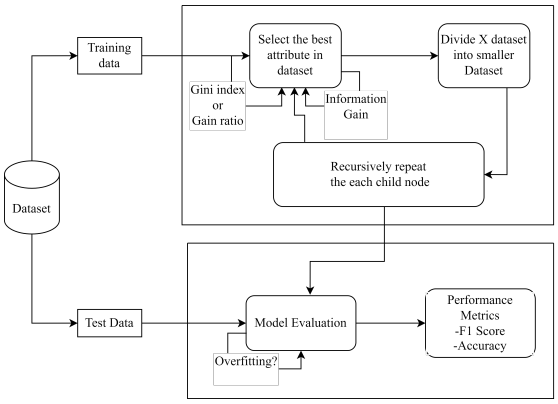

Figure 1: System Block Diagram

[1]

[1] *Using Cost Complexity

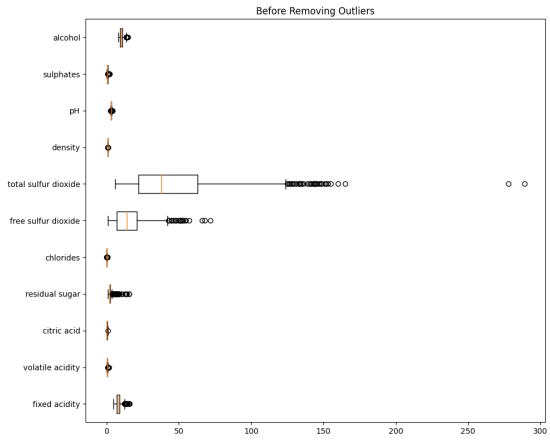

Figure 2: Plot Before Removing Outliers

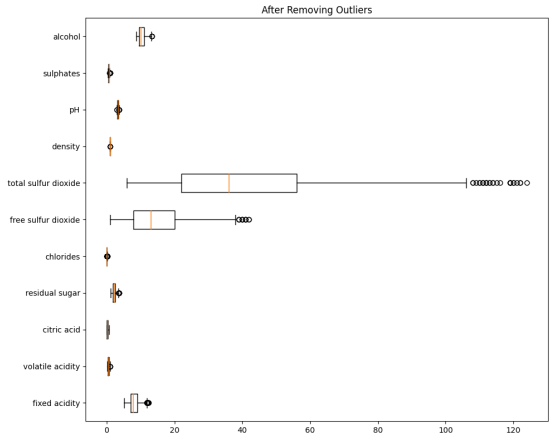

Figure 3: Plot After Removing Outliers

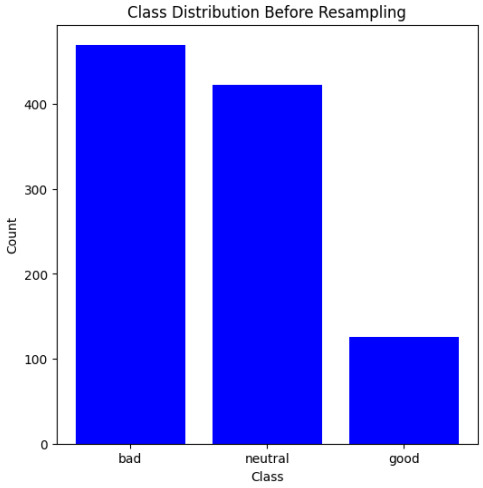

Figure 4: Plot of Class Distribution Before Resampling

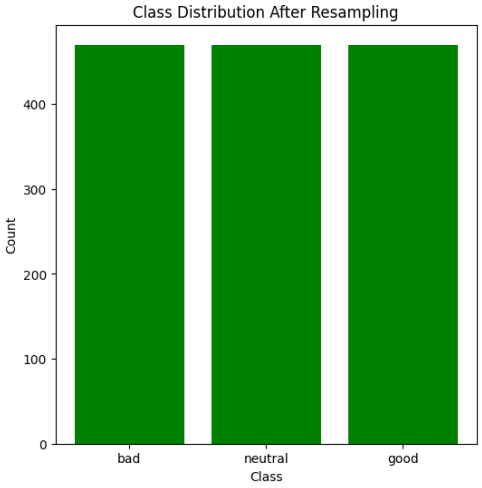

Figure 5: Plot of Class Distribution After Resampling

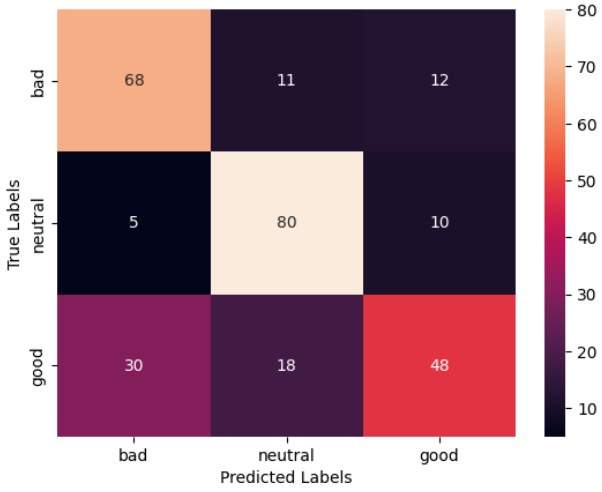

Figure 6: Confusion Matrix for selected Features

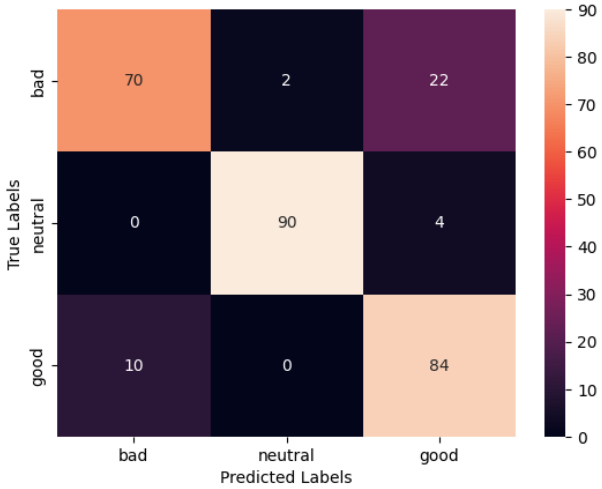

Figure 7: Confusion Matrix having all Features

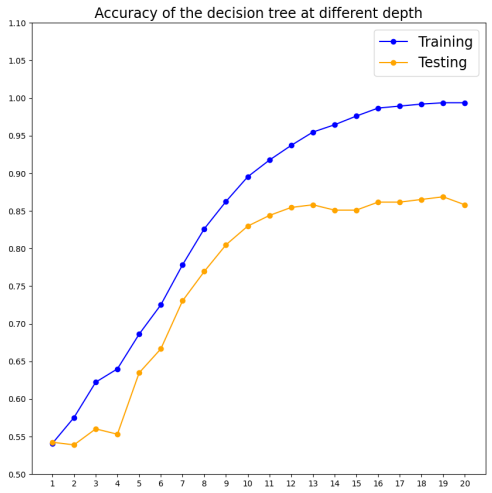

Figure 8: Plot of Accuracy Over Max Depth

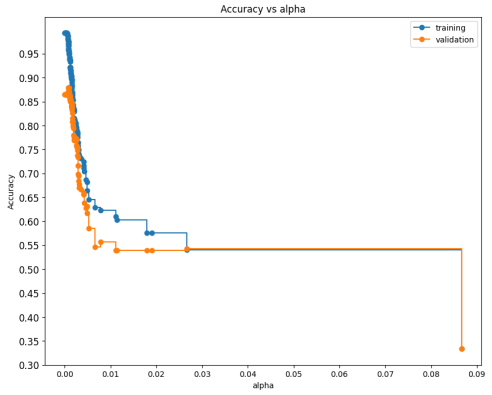

Figure 9: Plot of Accuracy over Cost Complexity Pruning

Table 1: Classification report of Best Accuracy

	Precision	Recall	F1 score	Support
bad	0.87	0.81	0.84	94
good	0.96	0.96	0.96	94
neutral	0.81	0.87	0.84	94
accuracy			0.88	282
macro avg	0.88	0.88	0.88	282
weighted avg	0.88	0.88	0.88	282

Table 2: Categorization of Wine Quality Score

Quality Score	Class
$<= 5$	bad
== 6	neutral
> 6	good

Table 3: Accuracy and F1 score of Decision tree

Methods	Accuracy	F1 score (Macro average)	F1 score (Weighted average)
”Entropy”	0.84	0.84	0.84
”Gini”	0.87	0.87	0.87
”Log loss”	0.84	0.84	0.84
”Gini” max depth = 9	0.87	0.87	0.87
”Gini” min impurity decrease = 0.005	0.87	0.87	0.87
”Gini” min samples split = 5	0.85	0.85	0.85
”Gini” Pruning* with alpha = 0.00069	0.88	0.88	0.88

B Equations

Gini Index

$$\text{Gini Index} = 1 - \sum_{i=1}^{J} p_i^2 \tag{1}$$

Entropy

$$\text{Entropy} = -\sum_{i=1}^{J} p_i \log_2(p_i) \tag{2}$$

Information Gain

$$\text{IG} = \text{Entropy(parent)} - \sum_{j=1}^{m} \frac{N_j}{N} \cdot \text{Entropy(child}_j\text{)} \tag{3}$$

Classification Error

$$\text{Classification Error} = 1 - \max(p_i) \tag{4}$$

Precision

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5}$$

Recall

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{6}$$

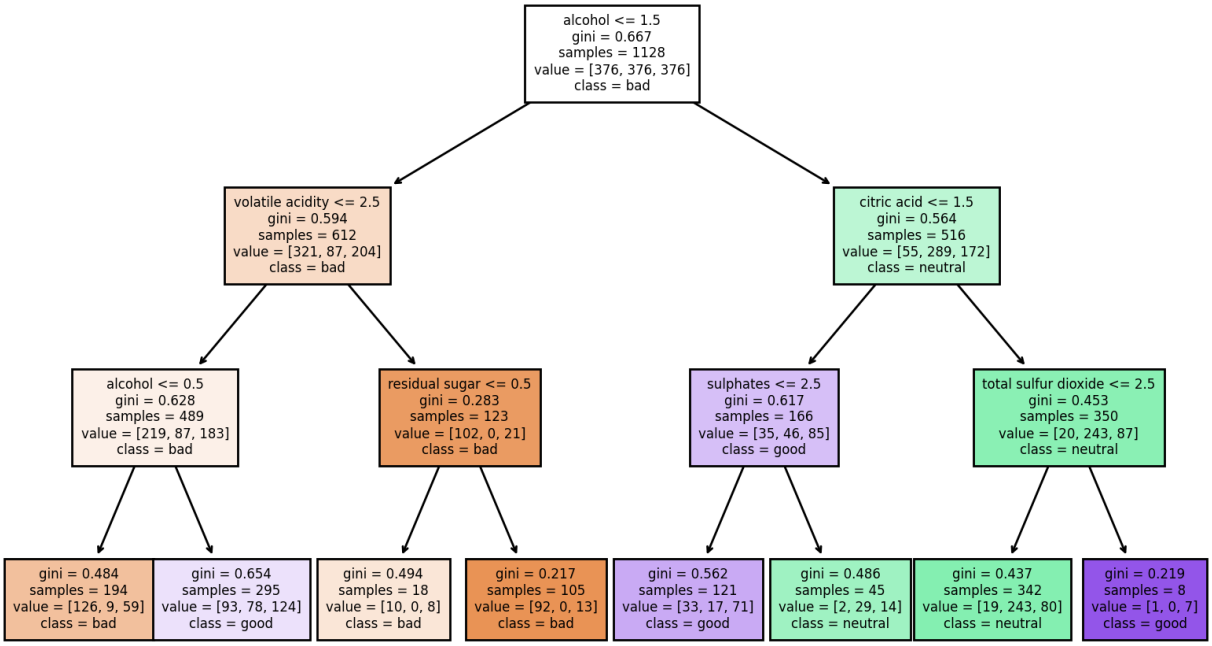

Figure 10: Decision Tree for Max Depth 3

F1 Score

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Macro Average

$$\text{Macro Average} = \frac{1}{J} \sum_{j=1}^{J} \text{Metric}_j \tag{8}$$

Weighted Average

$$\text{Weighted Average} = \sum_{j=1}^{J} \frac{N_j}{N} \cdot \text{Metric}_j \tag{9}$$

Accuracy

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{10}$$

REFERENCES

[1] P. Appalasamy, A. Mustapha, N. Rizal, F. Johari, and A. Mansor, "Classification-based data mining approach for quality control in wine production." 2012.

[2] D. Radosavljević, S. Ilić, and S. Pitulić, "A data mining approach to wine quality prediction," in *International Scientific Conference. Gabrovo*, 2019.

Kristina Ghimire is currently pursuing her undergraduate degree in Computer Engineering at IOE, Thapathali Campus. Her research interests encompass various areas, including data mining, machine learning, and deep learning.(THA077BCT023)

Punam Shrestha is currently pursuing her undergraduate degree in Computer Engineering at IOE, Thapathali Campus. Her research interests encompass various areas, including data mining, and web development.(THA077BCT038)