Wine Quality Analysis with Decision Trees and Random Forests

Name: Bhavya Sandhu

Student ID: 23022333

GitHub Repository Link:
https://github.com/BhavyaSandhu/MachineLearning/blob/551c8ad9c0d827be3bda0d0296dea72897e9783a/Bhavya_ML_Tutorial_.ipynb

➢ INTRODUCTION

This tutorial is about the application of Decision Trees and Random Forests to analyze wine quality dataset and create models that can classify wines based on their quality. In the beverage industry, the wine quality assessment is essential as it influences the techniques of production, consumer preferences and market trends. By utilizing the different attributes, the goal is to develop reliable models that accurately differentiate between various quality levels.

Two major machine learning models: Decision Trees and Random Forests, are undertaken to focus specifically. Through an in-depth analysis of these models, this study identifies the critical factors that influence wine quality while optimizing model parameters to enhance predictive accuracy.

Additionally, this tutorial provides a practical demonstration of how machine learning can be applied to real-world challenges in the beverage industry. By uncovering the key determinants of wine quality and balancing model complexity with performance, this can be used by the winemakers, industrialists, researchers, and professionals to seek valuable insights. The study emphasizes the role of Decision Trees and Random Forests in supporting wine quality evaluation and improving decision-making processes based on given data.

➢ DATASET

The Wine Quality dataset is a widely used dataset for predictive modeling, which offers a wide range of attributes of wines, alongside their quality ratings. This dataset is particularly valuable for machine learning applications because it provides labeled data, enabling the development of supervised learning models to predict wine quality. The dataset includes essential features such as acidity, residual sugar, pH, and alcohol content, which have been shown to significantly influence wine quality.

The main reason behind using this dataset was its application in the beverage industry. It has got practical implications. The valuable insights can be used by the industry to predict wine quality using measurable attributes which can significantly streamline production processes and improve market competitiveness. Additionally, the dataset's inherent class imbalance and the intricate relationships between features make it ideal for exploring complex machine learning algorithms like Decision Trees and Random Forests. These models are particularly well-suited for identifying key features that impact quality and addressing over fitting challenges, ensuring robust and reliable predictions. By working with this dataset, valuable insights can be drawn for optimizing wine production and quality management in real-world scenarios.

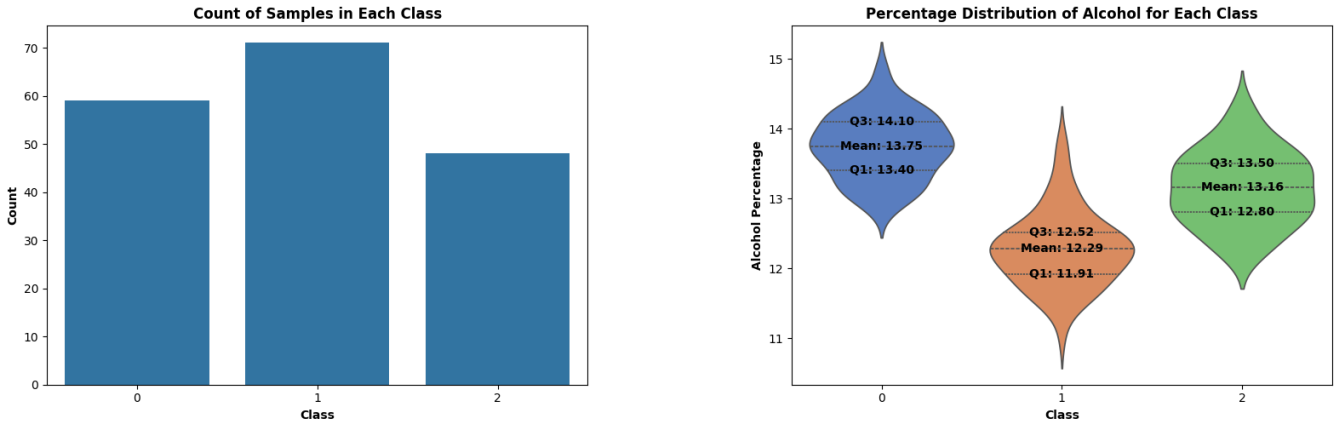


Figure 1 Exploratory Data Analysis

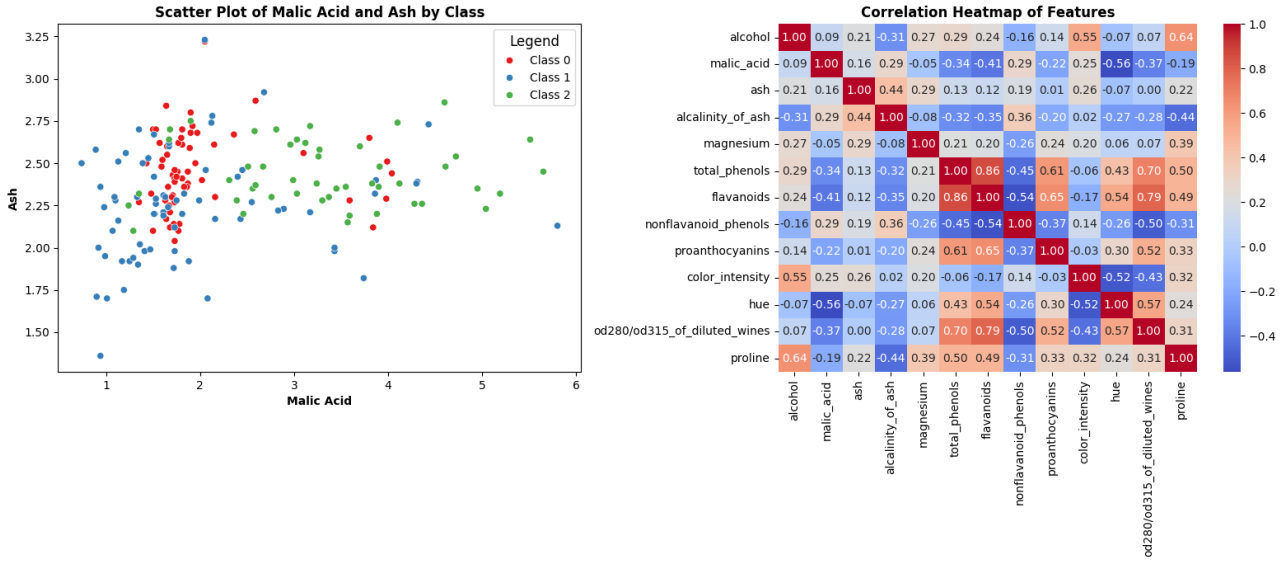


Figure 2 Exploratory Data Analysis

The exploratory data analysis was done and four plots were generated to analyze the dataset.

1. Bar plot

- The bar plot displays the distribution of wine quality classes (0, 1, and 2).
- It depicts the distribution of samples, highlighting that class 1 has been the most popular one.

2. Violin plot

- It shows the distribution of alcohol percentages for each wine quality class.
- Higher-quality wines (Class 0) generally have higher alcohol percentages (mean around 13.75%) compared to lower-quality classes (mean 12.29% for Class 1 and 11.91% for Class 2). This suggests alcohol content is a significant factor in determining wine quality.

3. Scatter Plot

- It shows relationship between malic acid and ash content, grouped by wine quality classes.
- A positive correlation is observed between malic acid and ash, particularly in Class 2. This plot highlights potential feature interactions that might influence wine quality classification.

4. Correlation Heatmap

- The heatmap visualizes pairwise correlations between all features in the dataset.

- Strong positive correlations are observed between some features, such as **alcohol** and **proline**, as well as **total phenols** and **flavanoids**.
- Negative correlations, such as between **malic acid** and **flavanoids**, suggest certain features might have opposing influences on wine quality.

➢ DATA PRE-PROCESSING

Data pre processing is done so as to ensure the data is prepared appropriately for machine learning models. These steps are essential to enhance the quality of the dataset, reduce biases, and improve model performance. The key preprocessing techniques used in this project are:

- **Handling Missing Values**: The dataset was inspected for any missing or inconsistent values

```
No missing values found in the dataset.

Class Distribution:
1    71
0    59
2    48
Name: count, dtype: int64
```

- **Feature Scaling**: The dataset features were standardized using **StandardScaler** to bring all numerical features to the same scale. This step ensures that no feature dominates the model training process due to differences in scale and improves convergence during training.

- **Encoding Categorical Data**: The dataset does not contain categorical variables.

- **Train-Test Split**: The dataset was split into **training** and **testing** subsets using an 80% and 20% ratio. This split ensures the model is trained on one subset and evaluated on another to assess its generalization capabilities effectively.

```
#Normalize features and split into training and testing sets
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3,
                                                    random_state=42)
```

➢ MODEL DEFINITION

1. RANDOM FOREST MODEL

Random Forest: Random Forest classifier is employed, which an ensemble is learning method based on decision trees. It combines multiple decision trees to improve generalization and reduce overfitting. It works well with both numerical and categorical data and can handle high-dimensional datasets effectively.

Architecture: Input features -----> Decision trees -----> Voting mechanism -----> Class prediction

Parameters: Number of Trees (n_estimators): Default value, Criterion: Gini impurity, Random State: 42

```
# Train the Random Forest model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train_pca, y_train)

# Make predictions
y_pred_rf = rf_model.predict(X_test_pca)

# Evaluate the model's performance
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print("\nRandom Forest Accuracy:", accuracy_rf)
print("\nRandom Forest Classification Report:")
print(classification_report(y_test, y_pred_rf))
print("\nRandom Forest Confusion Matrix:")
print(confusion_matrix(y_test, y_pred_rf))
```

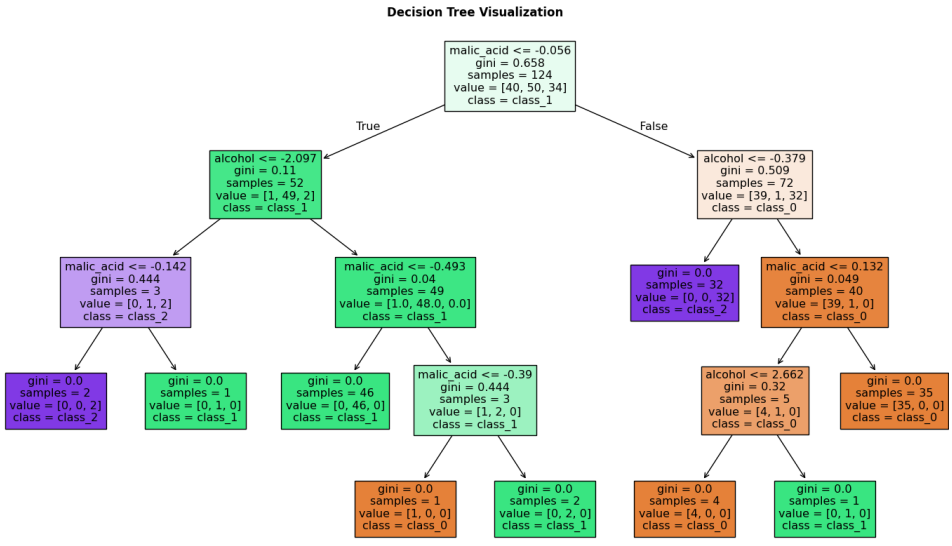
2. DECISION TREE

Decision Tree: Using CART algorithm to recursively partition the feature space. It makes decisions based on splitting criteria such as Gini impurity to maximize information gain. The resulting tree predicts the class labels of samples by traversing from the root to leaf nodes, where final predictions are made. This model captures complex decision boundaries.

Architecture: Input features -----> Tree structure (Internal nodes) -----> Splitting criteria -----> Leaf node prediction -----> Class prediction

Parameters: Criterion: Gini impurity, Splitter: Best, Maximum Depth: Default value, Random State: 42

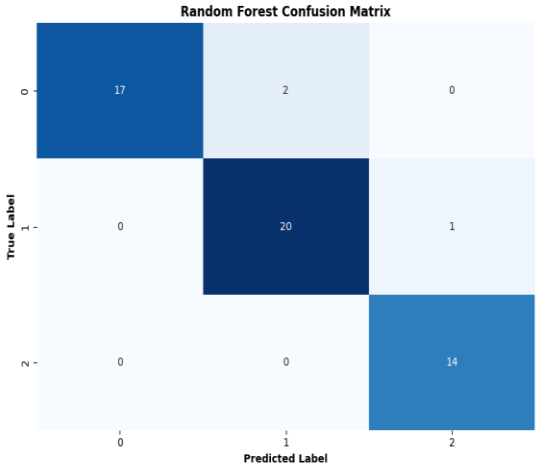
```
# Train a decision tree classifier
dt_model = DecisionTreeClassifier(random_state=42)
dt_model.fit(X_train_pca, y_train)

# Make predictions on the test set
y_pred_dt = dt_model.predict(X_test_pca)

# Calculate the accuracy of the decision tree model
accuracy_dt = accuracy_score(y_test, y_pred_dt)
print("Decision Tree Accuracy:", accuracy_dt)
```



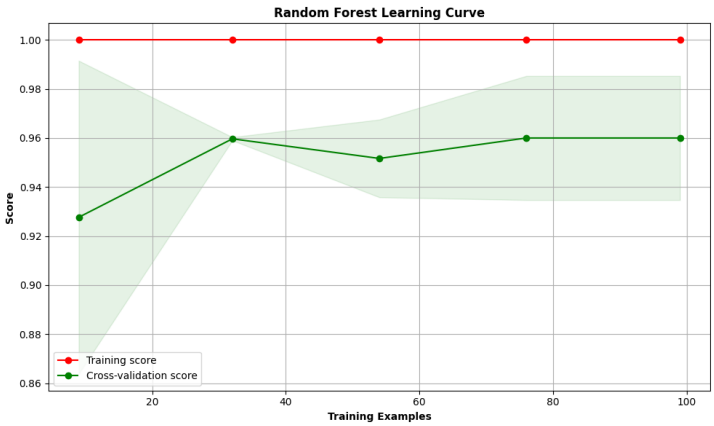
➢ RESULTS AND CONCLUSION

MODEL	ACCURACY
Random Forest	~94%
Decision Tree	~92%

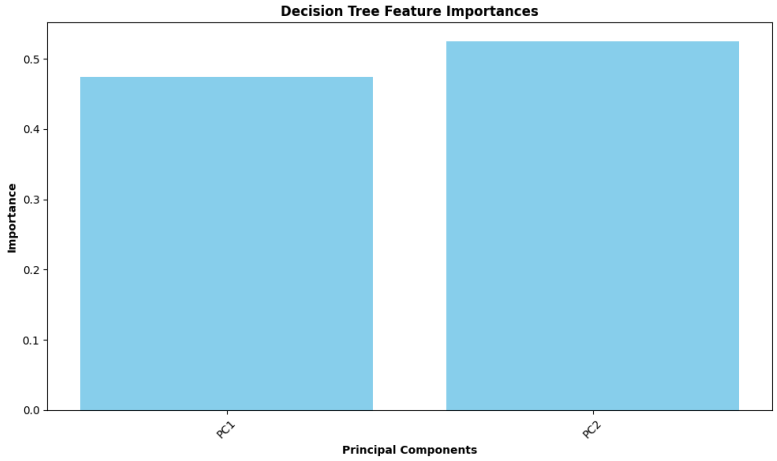
- The confusion matrix shows the classification performance of the Random Forest model. It highlights that most samples were correctly classified across all three classes (0, 1, 2), with very few misclassifications.



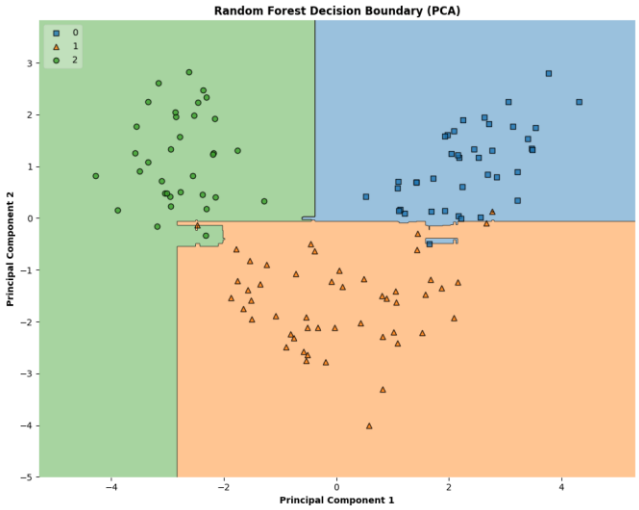
- The learning curve demonstrates that the Random Forest model achieves a near-perfect training score (1.0), while the cross-validation score stabilizes around 96%, indicating strong generalization with minimal overfitting.



- This feature importance chart displays the importance of the principal components (PC1 and PC2), showing their equal contribution to the Decision Tree model's classification process, emphasizing dimensional reduction efficiency.



- The decision boundary plot illustrates how the Random Forest model separates the classes based on two principal components (PC1 and PC2). The boundaries effectively capture the distribution of the three wine quality classes with distinct regions for each.



➢ LIMITATIONS AND IMPROVEMENTS

Model Performance: While the models achieved quite a decent accuracy of 94% for random forests and 92% for decision trees, there may be room for improvement in terms of performance metrics such as precision, recall, and F1-score. Fine-tuning hyperparameters or exploring more sophisticated algorithms could enhance performance.

```
Random Forest Accuracy: 0.9444444444444444

Random Forest Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.89      0.94        19
           1       0.91      0.95      0.93        21
           2       0.93      1.00      0.97        14
```

Feature Engineering: The effectiveness of machine learning models greatly depends on the quality of features. Exploring additional feature engineering techniques or incorporating domain knowledge could improve model performance.

➢ REFRENCES

-Breiman, L., 2001. Random Forests. Machine Learning, 45(1), pp.5-32. Available at: https://link.springer.com/article/10.1023/A:1010933404324.

-Dua, D. & Graff, C., 2019. UCI Machine Learning Repository. Available at: https://archive.ics.uci.edu/ml/datasets/wine+quality.

-Kuhn, M. & Johnson, K., 2013. Applied Predictive Modeling. New York: Springer. Available at: https://link.springer.com/book/10.1007/978-1-4614-6849-3.