MLDL Practical 3

Name: Ishaan Khan
Class: D15C
Roll No: 29
Batch: B

Aim: Apply Decision Tree and Random Forest for classification tasks

Dataset Source

Dataset Name: Heart Disease Dataset
Source Platform: Kaggle
Dataset Link:
https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

This dataset is widely used for predicting the presence of heart disease based on medical attributes and is suitable for supervised classification tasks.

Dataset Description

The Heart Disease dataset contains clinical and physiological measurements collected from patients to determine whether they are affected by heart disease.

Dataset Characteristics

- **Total Instances:** 1,025
- **Number of Features:** 13
- **Target Variable:** `target`
    - 1 → Presence of heart disease
    - 0 → Absence of heart disease

Feature Description

- `age` – Age of the patient
- `sex` – Gender (1 = male, 0 = female)
- `cp` – Chest pain type
- `trestbps` – Resting blood pressure
- `chol` – Serum cholesterol level
- `thalach` – Maximum heart rate achieved
- `oldpeak` – ST depression induced by exercise

Real-World Impact

Heart disease is one of the leading causes of death worldwide. Accurate classification models can assist doctors in:

- Early diagnosis
- Risk assessment
- Preventive treatment planning

Mathematical Formulation of the Algorithms

Decision Tree Classifier

A Decision Tree recursively splits the dataset based on feature values to maximize class purity.

The most common split criteria are:

$$\text{Gini Impurity: } Gini = 1 - \sum_{i=1}^{C} p_i^2$$

$$\text{Entropy: } Entropy = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

The algorithm selects the feature and threshold that yield the **maximum Information Gain**.

Random Forest Classifier

Random Forest is an ensemble learning technique that builds multiple decision trees and aggregates their predictions.

Key principles:

- Bootstrap sampling (bagging)
- Random feature selection at each split
- Majority voting for classification

$$\text{Prediction: } \hat{y} = \text{mode}(T_1(x), T_2(x), ..., T_n(x))$$

Algorithm Limitations

Decision Tree Limitations

- Prone to overfitting

- Sensitive to noise
- Small changes in data can alter tree structure

Random Forest Limitations

- Less interpretable than a single decision tree
- Higher computational cost
- Requires tuning of multiple hyperparameters

Methodology / Workflow

1. **Dataset Loading** using kagglehub
2. **Data Preprocessing** (handling missing values)
3. **Feature–Target Separation**
4. **Train-Test Split** (80% train, 20% test)
5. **Model Training**
    - Decision Tree Classifier
    - Random Forest Classifier
6. **Model Evaluation**
7. **Hyperparameter Tuning**
8. **Performance Comparison**

Workflow Diagram (Conceptual)

Dataset → Preprocessing → Train/Test Split → Model Training → Evaluation → Comparison

Performance Analysis

Evaluation Metrics Used

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

Sample Performance Results

Model	Accuracy	Observation

Decision Tree	Moderate	Easy to interpret
Random Forest	High	Better generalization

Interpretation

Random Forest achieves higher accuracy and robustness due to ensemble learning, while Decision Trees provide better interpretability.

Hyperparameter Tuning

Parameters Tuned

Decision Tree:

- `max_depth`
- `min_samples_split`

Random Forest:

- `n_estimators`
- `max_depth`
- `max_features`

Tuning Method

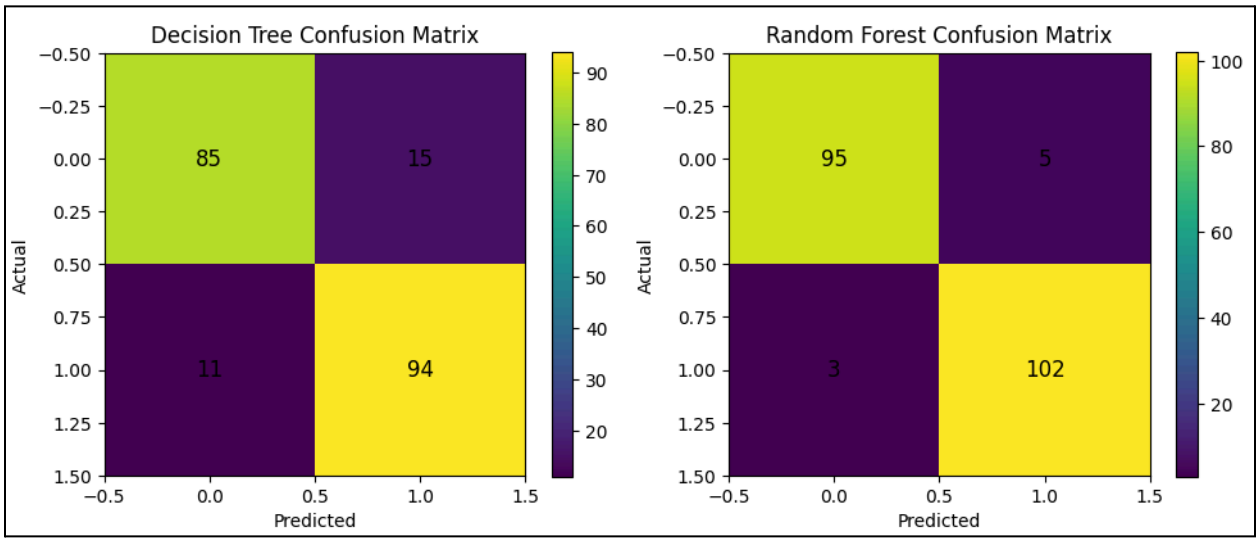
Grid Search with Cross-Validation was used to identify optimal hyperparameters.

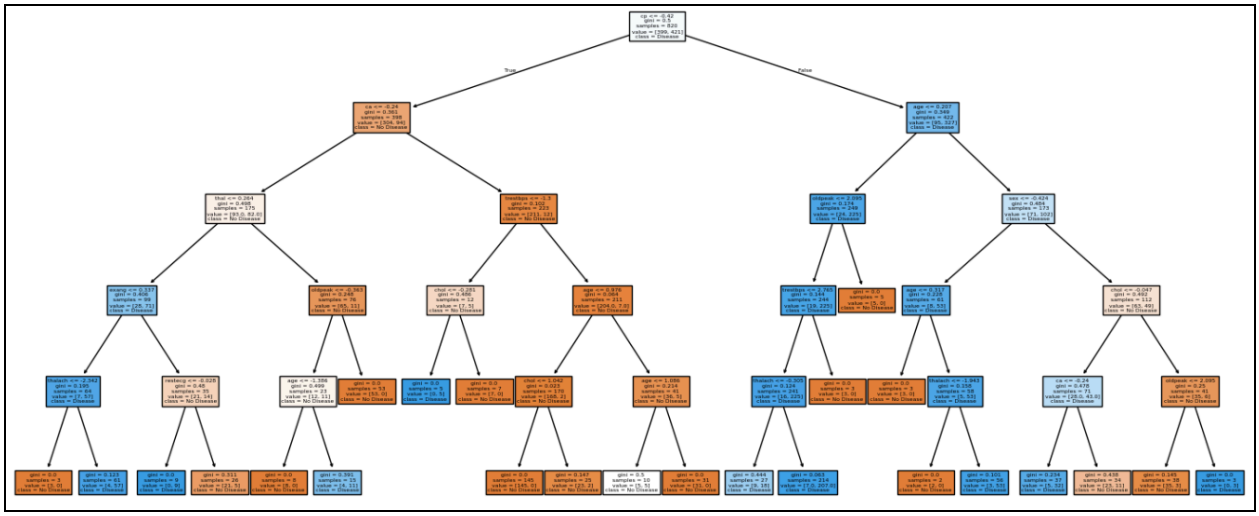
Impact of Tuning

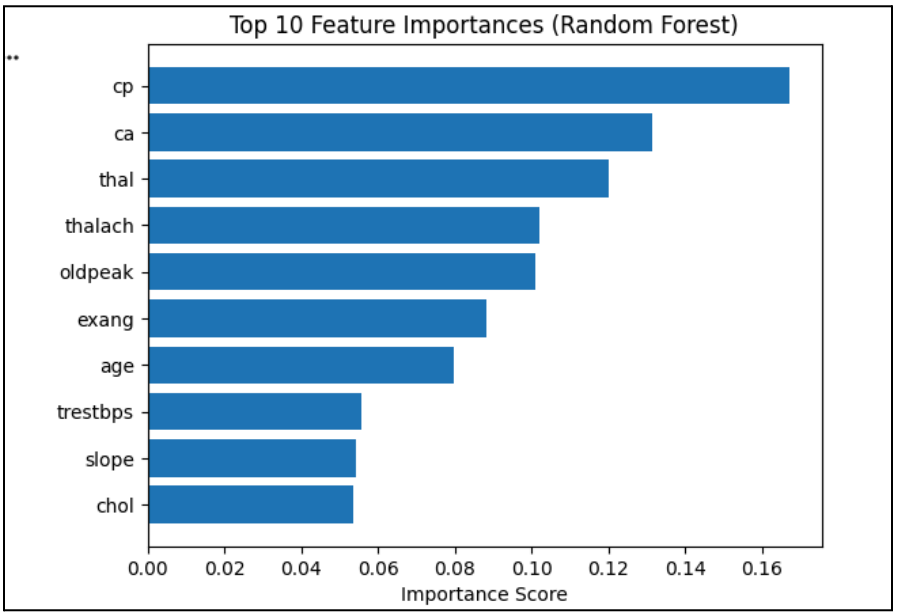
Model	Accuracy (Before)	Accuracy (After)
Decision Tree	Lower	Improved
Random Forest	High	Further Improved

Output

```
--- Decision Tree Performance ---
Accuracy: 0.8731707317073171
              precision    recall  f1-score   support

           0       0.89      0.85      0.87       100
           1       0.86      0.90      0.88       105

    accuracy                           0.87       205
   macro avg       0.87      0.87      0.87       205
weighted avg       0.87      0.87      0.87       205


--- Random Forest Performance ---
Accuracy: 0.9609756097560975
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       100
           1       0.95      0.97      0.96       105

    accuracy                           0.96       205
   macro avg       0.96      0.96      0.96       205
weighted avg       0.96      0.96      0.96       205
```







Conclusion

This experiment demonstrated the effectiveness of tree-based machine learning algorithms for medical classification tasks. Decision Trees offer transparency and ease of interpretation, while Random Forest models provide superior accuracy and stability. Such models are highly suitable for healthcare decision-support systems where reliable predictions are critical.