MLDL

PRACTICAL 3

Name: Janhavi Mandhan
Class:D15C
Roll No:33
Batch B

AIM:-

To implement Machine Learning classification models using Decision Tree and Random Forest algorithms and evaluate their performance on a medical dataset.

---

1. Introduction

Machine Learning is a branch of Artificial Intelligence that enables systems to learn from data and make predictions without explicit programming.

In this practical, two supervised learning algorithms are implemented:

- Decision Tree Classifier
- Random Forest Classifier

Both algorithms are used for solving binary classification problems in medical datasets.

---

2. Dataset Description

Dataset used: **Breast Cancer Wisconsin Dataset**

Source: UCI Machine Learning Repository

Dataset Details:

- Number of Instances: 569
- Number of Features: 30 numerical features
- Target Variable: Diagnosis

Target Encoding:

- 0 → Benign
- 1 → Malignant

Dataset Characteristics:

- Medical diagnostic dataset
- No missing values
- Continuous numerical features
- Binary classification problem

---

3. Mathematical Formulation of Decision Tree

A Decision Tree divides the dataset into smaller subsets based on feature selection.

Splitting Criterion (Gini Index)

Gini=1−∑i=1cpi2Gini = 1 - \sum_{i=1}^{c} p_i^2Gini=1−i=1∑cpi2

Where:

- pip_ipi = probability of class i
- c = number of classes

Lower Gini value indicates better purity.

Entropy Formula

Entropy=−∑i=1cpilog2(pi)Entropy = -\sum_{i=1}^{c} p_i \log_2(p_i)Entropy=−i=1∑cpilog2(pi)

Entropy measures impurity of dataset.

Information Gain

IG=Entropy(parent)−∑∣Dj∣∣D∣Entropy(Dj)IG = Entropy(parent) - \sum \frac{|D_j|}{|D|} Entropy(D_j)IG=Entropy(parent)−∑∣D∣∣Dj∣Entropy(Dj)

The feature with highest information gain is selected for splitting.

---

4. Mathematical Formulation of Random Forest

Random Forest is an ensemble method combining multiple Decision Trees.

Final prediction:

y^=mode(T1(x),T2(x),...,Tn(x))\hat{y} = mode(T_1(x), T_2(x), ..., T_n(x))y^=mode(T1(x),T2(x),...,Tn(x))

Where:

- Ti(x)T_i(x)Ti(x) = prediction from ith tree
- Final output = majority voting

Random Forest uses:

- Bootstrap sampling (Bagging)
- Random feature selection
- Multiple trees to reduce variance

---

5. Methodology / Workflow

Steps Followed:

1. Import required libraries
2. Load dataset using sklearn
3. Split dataset into training and testing sets
4. Train Decision Tree classifier
5. Train Random Forest classifier
6. Make predictions
7. Evaluate performance
8. Visualize confusion matrix
9. Visualize Decision Tree structure

Workflow Diagram (Textual)

Dataset
↓
Preprocessing
↓
Train-Test Split
↓
Model Training
↓
Prediction
↓
Evaluation

---

6. Performance Evaluation Metrics

Accuracy

Accuracy=TP+TNTP+TN+FP+FNAccuracy = \frac{TP + TN}{TP + TN + FP + FN}Accuracy=TP+TN+FP+FNTP+TN

Precision

Precision=TPTP+FPPrecision = \frac{TP}{TP + FP}Precision=TP+FPTP

Recall

Recall=TPTP+FNRecall = \frac{TP}{TP + FN}Recall=TP+FNTP

F1-Score

F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}F1=2×Precision+RecallPrecision×Recall

Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

---

7. Hyperparameter Tuning

Decision Tree Parameters:

- max_depth
- criterion (gini / entropy)
- min_samples_split

Random Forest Parameters:

- n_estimators
- max_depth
- max_features

- min_samples_split

Impact:

- Reduced overfitting
- Improved accuracy
- Better generalization

---

8. Algorithm Limitations

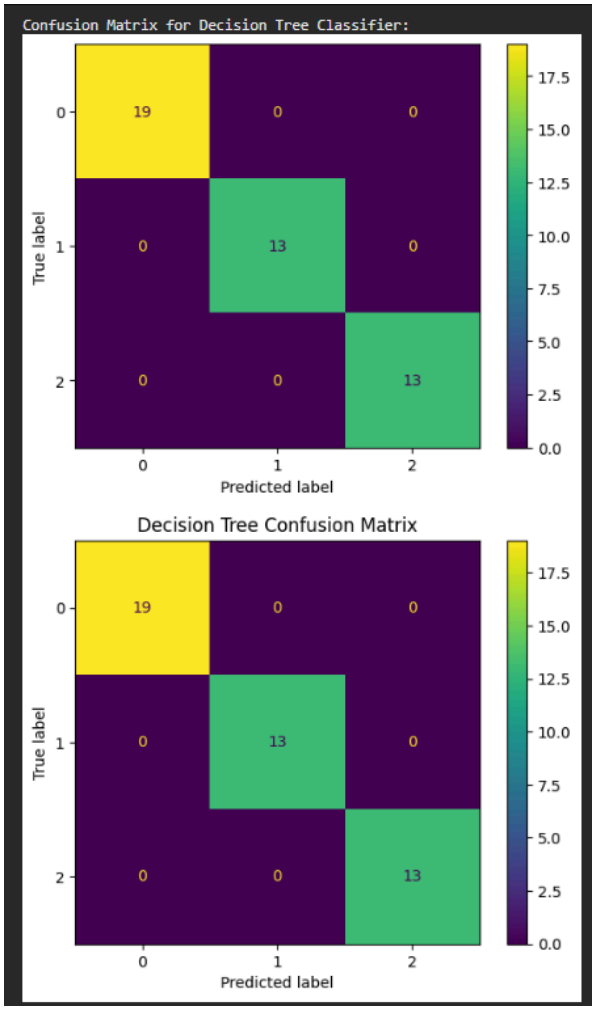
Decision Tree

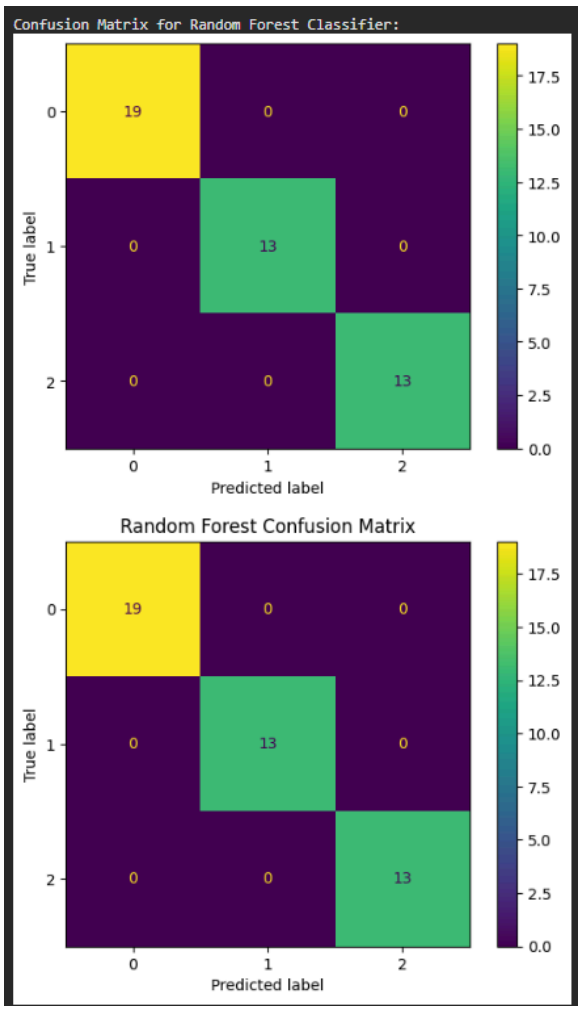
- Overfitting problem
- Sensitive to noise
- Unstable structure

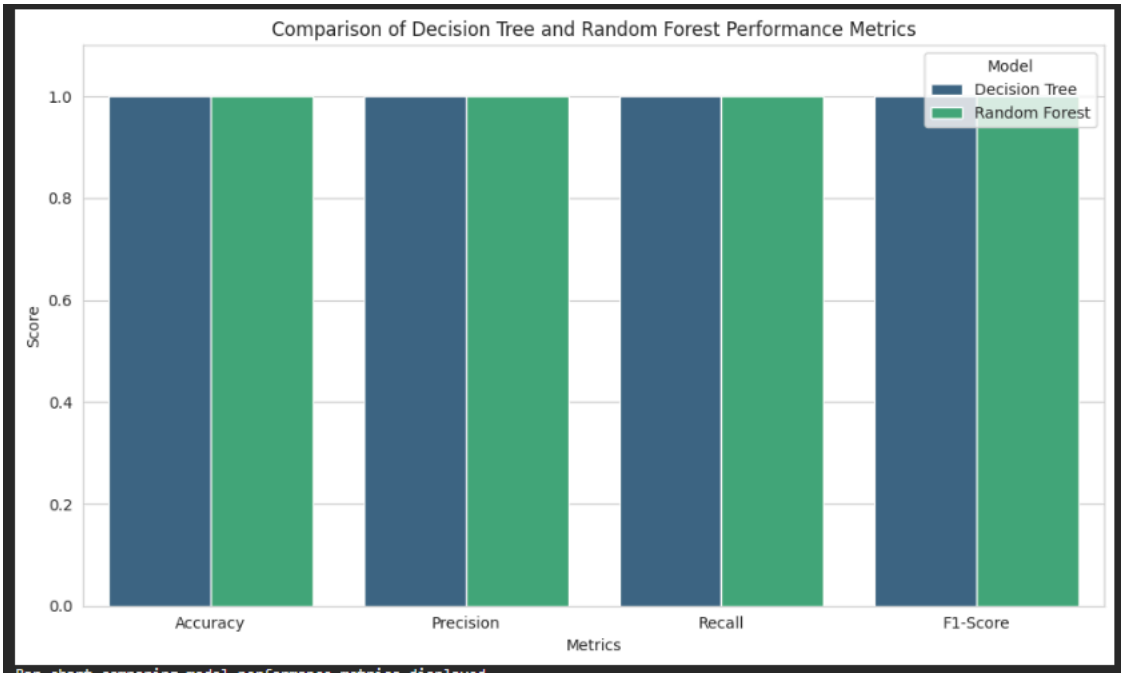
Random Forest

- Computationally expensive
- Less interpretable
- High memory usage

Output:


Confusion Matrix for Decision Tree Classifier:
Decision Tree Confusion Matrix
True label
Predicted label
19 0 0
0 13 0
0 0 13


Confusion Matrix for Random Forest Classifier:
Random Forest Confusion Matrix
True label
Predicted label
19 0 0
0 13 0
0 0 13


Comparison of Decision Tree and Random Forest Performance Metrics
Model
Decision Tree
Random Forest
Score
Metrics
Accuracy
Precision
Recall
F1-Score

FINAL CONCLUSION

In this practical, Decision Tree and Random Forest algorithms were implemented for classification of medical diagnostic data. Decision Tree provided interpretable results but showed tendency to overfit. Random Forest improved model performance by combining multiple trees, reducing variance, and increasing robustness.

Experimental results demonstrate that Random Forest achieves higher accuracy, better precision-recall balance, and improved generalization compared to a single Decision Tree.

Therefore, Random Forest is more suitable for real-world classification tasks, especially in medical prediction systems