Sri Sivasubramaniya Nadar College of Engineering, Chennai

(An Autonomous Institution affiliated to Anna University)

Degree & Branch	B.E Computer Science & Engineering	**Semester**	VI
Subject Code & Name	UCS2612 – Machine Learning Laboratory		
Academic Year	2025–2026 (Even)	**Batch**	2023–2027
Name: Monesh M **Reg. No:** 3122235001084			

Experiment 6

Decision Tree and Random Forest: A Comparative Classification Study

Objective

- To implement a **Decision Tree classifier**.
- To extend the Decision Tree into a **Random Forest ensemble model**.
- To study the impact of hyperparameters on overfitting and generalization.
- To **select optimal hyperparameters using 5-Fold Cross-Validation**.
- To compare single-tree and ensemble-tree models.

Dataset

Wisconsin Diagnostic Breast Cancer Dataset

- Total samples: 569
- Features: 30 numerical attributes
- Target classes: Malignant (M - Encoded as 1) and Benign (B - Encoded as 0)

Dataset Link: https://archive.ics.uci.edu

Theory

Decision Tree Classifier

A Decision Tree is a supervised learning model that recursively splits the feature space using impurity measures to form decision rules.

Key Concepts:

- Gini Index and Entropy

- Tree depth and node splitting
- Overfitting in deep trees

Limitation: Decision Trees are prone to high variance and overfitting if allowed to grow to their maximum depth without pruning.

Random Forest Classifier

Random Forest is an ensemble learning technique that combines multiple decision trees trained on bootstrapped samples.

Key Ideas:

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

Advantage: Reduces variance and significantly improves generalization compared to a single Decision Tree without substantially increasing bias.

Steps for Implementation

1. Load the dataset and encode class labels.
2. Perform Exploratory Data Analysis:
   - Class distribution
   - Feature correlation analysis

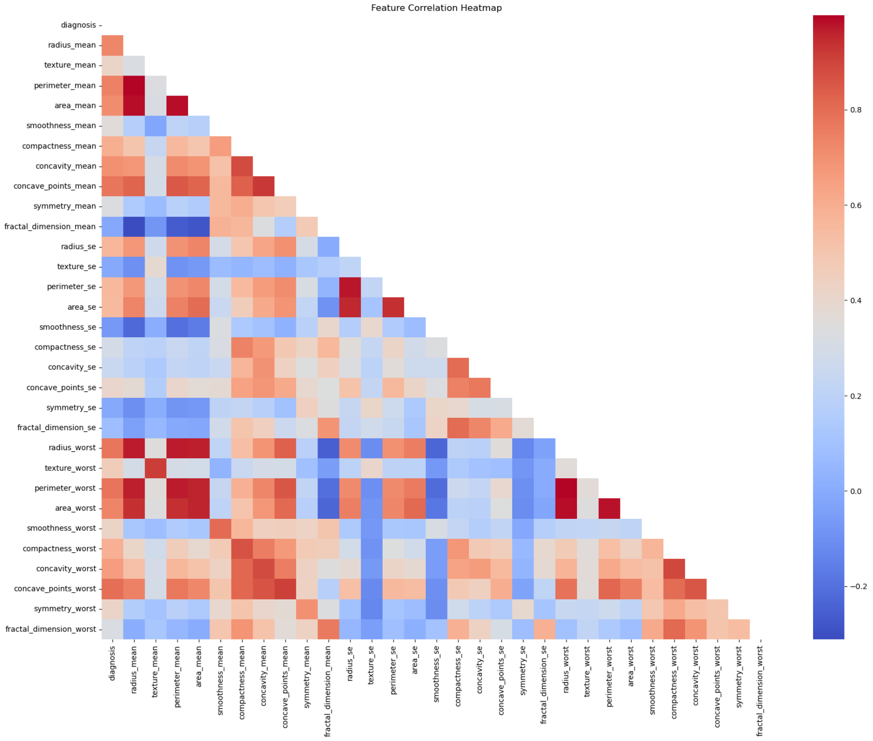

Figure 1: Feature Correlation Heatmap

3. Split the dataset into training and testing sets (80–20).
4. Train a Decision Tree classifier.
5. Define a hyperparameter search space.
6. Perform **5-Fold Cross-Validation** using `GridSearchCV` to evaluate each hyperparameter combination.
7. Select the hyperparameters that yield the best average cross-validation performance.
8. Train a Random Forest classifier.
9. Repeat cross-validation-based hyperparameter selection.
10. Compare both models using evaluation metrics.

Hyperparameters to be Explored

Decision Tree

- `criterion`: [’gini’, ’entropy’]
- `max_depth`: [None, 5, 10, 15]
- `min_samples_split`: [2, 5, 10]
- `min_samples_leaf`: [1, 2, 4]

Random Forest

- `n_estimators`: [50, 100, 200]
- `max_depth`: [None, 10, 20]
- `min_samples_split`: [2, 5]
- `min_samples_leaf`: [1, 2]

Hyperparameter Tuning Results

Decision Tree Cross-Fold Results

Table 1: Decision Tree Best Hyperparameters Configuration

Criterion	Max Depth	Min Samples Split	Min Samples Leaf	Best CV Accuracy (%)
Entropy	5	2	2	**93.19%**

Random Forest Cross-Fold Results

Table 2: Random Forest Best Hyperparameters Configuration

n_estimators	Max Depth	Min Samples Split	Min Samples Leaf	Best CV Accuracy (%)
200	None	2	1	**96.26%**

5-Fold Cross-Validation Performance Comparison

Table 3: 5-Fold Cross-Validation Accuracy Comparison (%)

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Decision Tree	91.21	92.31	95.60	95.60	91.21	**93.19**
Random Forest	94.51	97.80	98.90	97.80	92.31	**96.26**

Evaluation Metrics

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix
- ROC Curve and AUC

Table 4: Final Model Performance on Test Set

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.9561	1.0000	0.8810	0.9367
Random Forest	0.9649	1.0000	0.9048	0.9500

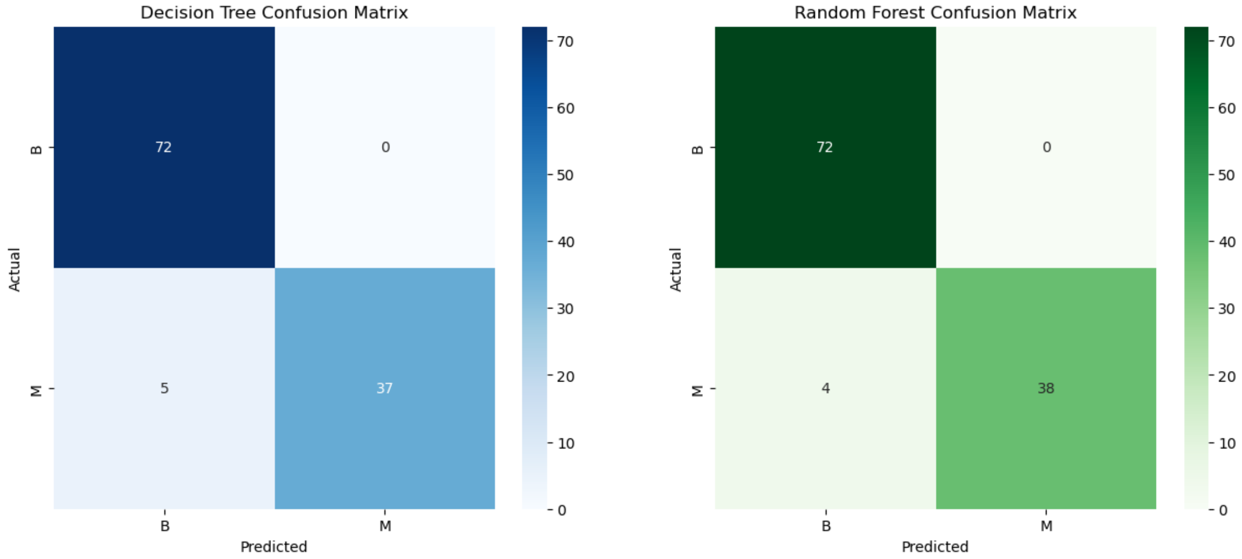

Figure 2: Confusion Matrix Comparison (DT vs RF)

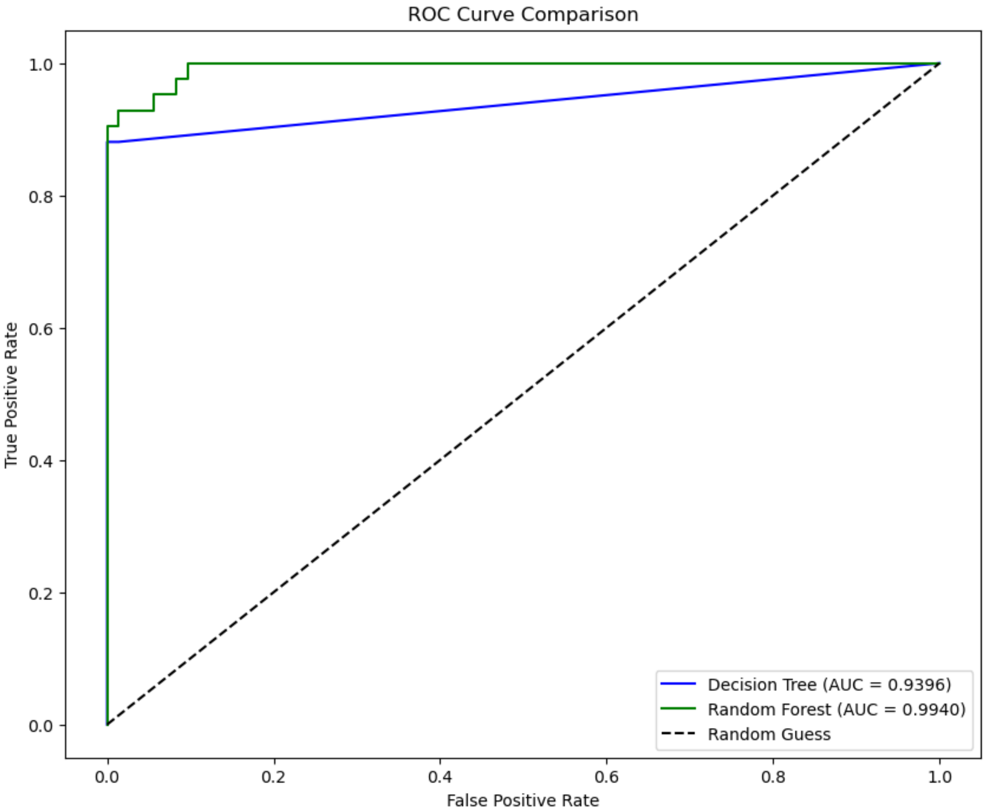

Figure 3: ROC Curve Comparison

Observations

- **How does tree depth affect overfitting in Decision Trees?**
  Shallow trees can underfit the data. As the depth increases, the tree begins to memorize the training data (including noise), leading to overfitting and high variance. Tuning 'max_depth' restricts this growth, forcing the model to capture only the most significant patterns.

- **Which hyperparameter had the greatest impact on performance?**
  For the Decision Tree, 'max_depth' and 'min_samples_leaf' heavily influenced pruning and prevented overfitting. For the Random Forest, 'n_estimators' (number of trees) had the most stabilizing effect on test accuracy.

- **How does Random Forest improve generalization?**
  By generating multiple decision trees trained on random subsets of the data (bootstrapping) and considering random subsets of features at each split, it ensures that trees are largely uncorrelated. Averaging their predictions dramatically reduces variance.

- **Did ensemble learning always improve performance? Why or why not?**
  Yes, for this complex diagnostic dataset. Test accuracy improved from 94.74% (DT) to 96.49% (RF). However, if base learners are highly correlated or the dataset is extremely simplistic, the computational overhead of an ensemble might not yield a massive gain over a single fine-tuned tree.

Conclusion

Decision Tree and Random Forest models were implemented and evaluated using 5-fold cross-validation. Hyperparameters were effectively selected using `GridSearchCV` based on average cross-validation performance, ensuring robust generalization. The results definitively demonstrate that the Random Forest reduces model variance and improves stability compared to a single Decision Tree, achieving higher accuracy (96.49%) and a more robust AUC score on the Wisconsin Diagnostic Breast Cancer Dataset.

References

- Scikit-learn: Decision Trees
- Scikit-learn: Random Forest
- UCI Dataset: Breast Cancer Wisconsin (Diagnostic)