EXPERIMENT 3

Implementation of Decision Tree and Random Forest for Classification

Aim of the Experiment

To implement **Decision Tree** and **Random Forest** algorithms for classification and compare their performance using accuracy, confusion matrix, and classification report.

Theory

Decision Tree

A Decision Tree is a supervised machine learning algorithm used for classification and regression.

It works by splitting the dataset into subsets based on feature values. Each internal node represents a feature, each branch represents a decision rule, and each leaf node represents the final class label.

The tree splits the data in such a way that the classes become more pure at each level.

Advantages:

- Easy to understand and interpret
- Requires little data preprocessing
- Works for both categorical and numerical data

Disadvantages:

- Can easily overfit
- Sensitive to small data variations

Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees.

Instead of using one tree, it:

- Creates multiple trees
- Uses random subsets of data
- Uses random subsets of features
- Final prediction is based on majority voting

Advantages:

- Reduces overfitting
- More accurate than single Decision Tree
- Handles large datasets well

3. Mathematical Formulation of the Algorithm

Decision Tree (Using Gini Index)

The Gini Index measures impurity in a dataset.

$$Gini = 1 - \sum_{i=1}^{n} p_i^2$$

Where:

- p_i = probability of class i

Lower Gini value → better split.

Information Gain:

$$IG = Gini(parent) - \sum \frac{N_{child}}{N_{parent}} \times Gini(child)$$

The feature with highest Information Gain is selected for splitting.

Random Forest Mathematical Idea

Random Forest builds multiple decision trees.

For each tree:

For each tree:

1. Bootstrap sampling:

$$D_i \subset D$$

2. Random feature selection:
   Instead of using all features, select random subset m from total M.

Final prediction:

$$\hat{y} = \text{majority vote of all trees}$$

For classification:

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

Where:

- $T_i(x)$ = prediction of ith tree

METHODOLOGY / WORKFLOW

Step 1: Data Collection

- Load dataset
- Separate features and target

Step 2: Data Preprocessing

- Handle missing values
- Encode categorical variables
- Feature scaling (if required)
- Split into training and testing sets

Step 3: Model Training

Train:

- Decision Tree classifier

- Random Forest classifier

Step 4: Model Evaluation

Evaluate using:

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

Step 5: Performance Comparison

Compare:

- Accuracy
- Misclassification rate
- Model stability

Real-Life Example

Imagine predicting whether a student will pass or fail based on:

- Study hours
- Attendance
- Previous marks

Decision Tree:

It asks:

- If attendance > 75% → go left

- If study hours > 3 → go right
- Final decision at leaf node

Single tree makes final decision.

Random Forest:

Instead of one teacher (tree), imagine:

- 100 teachers give their opinion.
- Final result is based on majority vote.

This reduces mistakes.

RESULTS (From Your Output)

Decision Tree Accuracy = 92.98%
Random Forest Accuracy = 96.49%

Random Forest performed better because ensemble learning reduces overfitting and improves generalization.

CONCLUSION

In this experiment, Decision Tree and Random Forest classifiers were successfully implemented. The Decision Tree model achieved an accuracy of 92.98%, whereas the Random Forest model achieved a higher accuracy of 96.49%.

Random Forest outperformed Decision Tree because it combines multiple trees and reduces overfitting through ensemble learning. This experiment demonstrates that ensemble techniques improve classification performance compared to single-tree models.

Final Comparison Table

Model	Accuracy
Decision Tree	92.98%
Random Forest	96.49%