Experiment No:- 3

Name:- Dipesh Makdiya
Class:- D15C
Roll no:- 32

EXPERIMENT: Decision Tree & Random Forest for Classification

---

1. Dataset Source

Dataset Used: Iris Dataset
Type: Classification
Source:
https://scikit-learn.org/stable/datasets/toy_dataset.html#iris-dataset

Access Method:

```
from sklearn.datasets import load_iris
```

Reason for Selection:

- Real-world benchmark dataset
- Suitable for classification
- Commonly used for Decision Tree & Random Forest experiments

---

2. Dataset Description

2.1 Iris Dataset

Objective:
Classify iris flowers into species using Decision Tree and Random Forest algorithms.

Number of Instances: 150
Number of Features: 4

Features:

- Sepal length
- Sepal width
- Petal length
- Petal width

Target Variable:

- Species (Setosa, Versicolor, Virginica)
  (Binary or multiclass classification can be used; here we use multiclass)

Characteristics:

- Clean dataset (no missing values)
- Numerical features
- Suitable for tree-based classifiers

---

3. Mathematical Formulation of the Algorithms

3.1 Decision Tree Classifier

A Decision Tree splits data recursively based on feature values to maximize class separation.

Splitting Criteria:

Gini Index
$Gini=1-\sum_{i=1}cpi2$Gini=1−i=1∑cpi2

Entropy
$Entropy=-\sum_{i=1}cpi\log2(pi)$Entropy=−i=1∑cpilog2(pi)

Information Gain:

$IG=Entropy(parent)-\sum knknEntropy(childk)$IG=Entropy(parent)−k∑nnkEntropy(childk)

The feature with **maximum information gain** is selected for splitting.

3.2 Random Forest Classifier

Random Forest is an **ensemble learning method** that builds multiple Decision Trees and combines their predictions.

Key Concepts:

- Bootstrap sampling (random sampling with replacement)
- Random feature selection
- Majority voting for classification

Final Prediction:

y^=mode(h1(x),h2(x),...,hn(x))y^=mode(h1(x),h2(x),...,hn(x))

```
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
print(confusion_matrix(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf))
```

```
Random Forest Accuracy: 1.0
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

---

4. Algorithm Limitations

Decision Tree Limitations

- Prone to overfitting
- Sensitive to small variations in data
- Complex trees are hard to interpret

Random Forest Limitations

- Computationally expensive
- Less interpretable than a single tree
- Requires tuning for optimal performance

---

5. Methodology / Workflow

Step-by-Step Workflow

1. Import required libraries
2. Load Iris dataset
3. Data exploration
4. Feature and target selection
5. Train-test split
6. Train Decision Tree model
7. Train Random Forest model
8. Prediction
9. Performance evaluation
10. Compare results

Workflow Diagram (Conceptual):

```
Dataset → Preprocessing → Train/Test Split
→ Model Training → Prediction → Evaluation
```

---

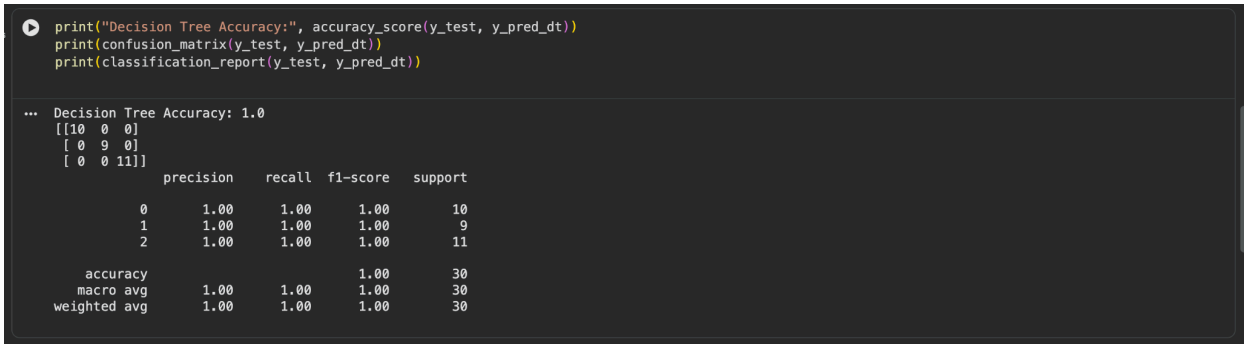
6. Performance Analysis

Metrics Used

- Accuracy
- Confusion Matrix
- Classification Report (Precision, Recall, F1-Score)

Interpretation:

- Higher accuracy → better classification
- Random Forest usually outperforms Decision Tree due to ensemble learning



```
print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))
print(confusion_matrix(y_test, y_pred_dt))
print(classification_report(y_test, y_pred_dt))
```

```
Decision Tree Accuracy: 1.0
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

Result and Conclusion

Result:

- Decision Tree achieved good classification accuracy.
- Random Forest achieved higher accuracy due to ensemble learning and reduced overfitting.

Conclusion:
Decision Tree and Random Forest algorithms were successfully applied for classification using the Iris dataset. Random Forest outperformed Decision Tree in terms of accuracy and robustness, demonstrating the effectiveness of ensemble methods in supervised learning tasks.
