Task 5: Decision Trees and Random Forests

✅ Task 5: Decision Trees and Random Forests — Full Explanation

This task teaches you how to use **tree-based machine learning models** for **classification** and **regression**, using tools like **Scikit-learn** (for modeling) and **Graphviz** (for visualization).

🎯 Objective

Learn how to build, evaluate, and interpret **Decision Trees** and **Random Forests** — two of the most popular machine learning models used in data science.

These models help in:

Classifying data into categories

Predicting continuous values

Understanding which features are most important

🛠 Tools Used

1. Scikit-learn

Library for building the models

Provides functions for training Decision Trees and Random Forests

Offers metrics for evaluation like accuracy, precision, etc.

2. Graphviz

Used to **visualize decision trees**

Helps you understand how the model splits the data step-by-step

📓 Hints / Mini Guide (Step-by-Step Explanation)

1. Train a Decision Tree Classifier and Visualize the Tree

Load a dataset (e.g., Iris, Titanic, etc.)
Split into training/testing
Train the model using DecisionTreeClassifier()

Use Graphviz to draw the tree structure
You will see:

Root node

Feature splits

Decision rules

2. Analyze Overfitting and Control Tree Depth

Decision trees tend to overfit if not controlled.
You can prevent this by setting parameters such as:
max_depth
min_samples_split
min_samples_leaf
This step teaches you:
What overfitting looks like
How pruning or limiting depth improves performance

3. Train a Random Forest and Compare Accuracy

A **Random Forest** is a collection of many decision trees.
Benefits:
Reduces overfitting

Gives more stable accuracy
Handles noisy data better
Train using:

```
RandomForestClassifier()
```

Then compare:

Accuracy of Decision Tree vs Random Forest

Usually Random Forest performs better

4. Interpret Feature Importances

Random Forests provide **feature importance scores** that show:
Which features contribute most to predicting the output
Helps in model understanding and feature selection
You'll produce a graph or list showing feature contributions.

5. Evaluate Using Cross-Validation

Cross-validation ensures the model performs well on unseen data.
You can use:

```
cross_val_score()
```

This gives more reliable model performance metrics.

📌 End Goal of This Task

By completing this task, you will learn to:
✓ Build and tune a Decision Tree
✓ Visualize a tree using Graphviz

✓ Understand overfitting and pruning
✓ Train a Random Forest
✓ Compare model performances
✓ Identify important features

✓ Evaluate models using cross-validation
If you want, I can also give you: ✅ Python code for each step

✅ A summary version
✅ A report-style explanation for your assignment
Just tell me!