

Assignment: Implementing a Decision Tree in Python (Unit IV)

Overview

This assignment allows you to explore how **Decision Trees** work and practice implementing one using **Python**.

You will research, code, and interpret results from a Decision Tree classifier using a real dataset.

Learning Outcomes

By completing this assignment, you should be able to:

- Explain the working principle of Decision Trees.
- Implement a Decision Tree model in Python using the `scikit-learn` library.
- Evaluate and interpret the performance of a Decision Tree classifier.

Instructions

1. Research Task:

Write a short explanation (in your own words) on how Decision Trees work and how they are implemented in Python.

2. Implementation Task:

Use **Python** and the **scikit-learn** library to implement a Decision Tree on a real dataset (e.g., *Iris*, *Titanic*, *Breast Cancer*, or any dataset from `sklearn.datasets` or Kaggle).

Your code should include:

- Importing and exploring the dataset.
- Splitting data into training and testing sets.
- Building and training a Decision Tree classifier (`DecisionTreeClassifier`).
- Making predictions and evaluating performance (accuracy, confusion matrix, etc.).
- Visualize the tree using `plot_tree()` or `graphviz`.

Submission Details

- **Submission format:** Upload **one file only** (PDF)
- **Group size:** 5 students per group
- **File name format:** `GroupName_DecisionTree_Assignment`
- **Submission deadline:** 5.11.2025
- **Late submission policy:** Marks will be deducted for late submissions unless prior approval is given.
- **No similar works are allowed.**

Grading Rubric (Total = 10 Marks)

Criteria	Description	Marks
1. Concept Explanation	Clear and accurate explanation of Decision Tree working principles	2
2. Python Implementation	Correct and functional implementation using scikit-learn	4
3. Evaluation & Interpretation	Includes accuracy score, confusion matrix, and explanation of results	2
4. Presentation & Originality	Code clarity, structure, and proper referencing of sources	2
Total		10 Marks

Tips for Success

- Use the `DecisionTreeClassifier` from `sklearn.tree`.
 - Experiment with different splitting criteria such as `gini` and `entropy`.
 - Try using small public datasets before moving to larger ones.
 - Make sure your program runs without syntax or runtime errors.
-