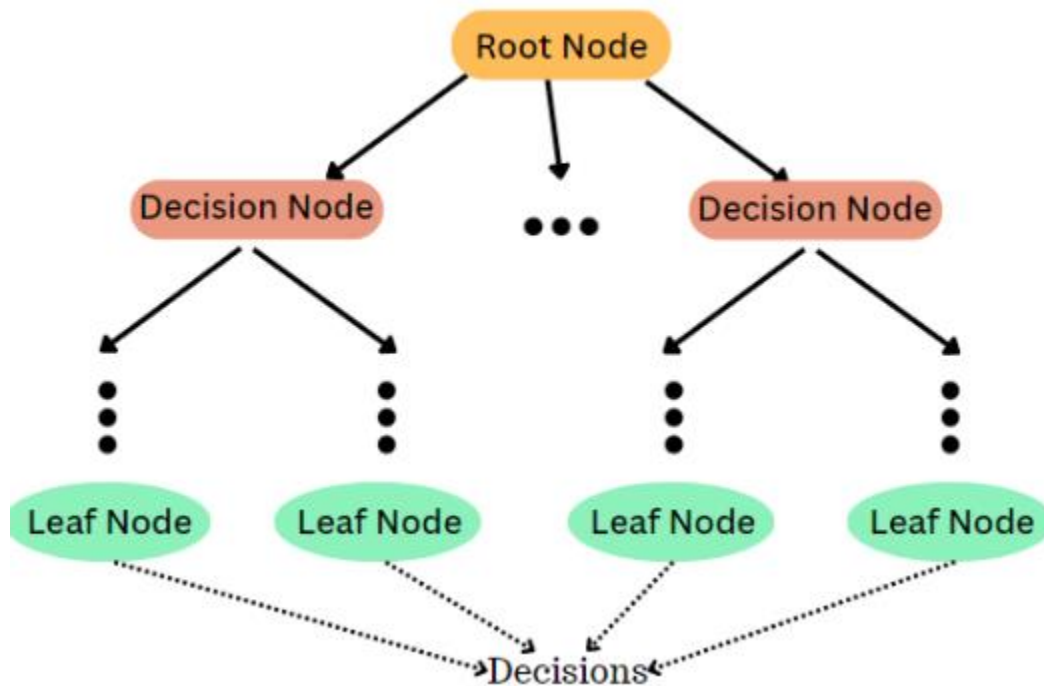# LAB No 3

Decision Tree Classifier

In this lab, students will study and implement the Decision Tree Classifier, a supervised machine learning algorithm used for classification tasks. The lab begins with a manual calculation of entropy and information gain to understand how decision trees choose splitting attributes. Students will then implement a decision tree on a small categorical dataset, followed by applying the algorithm to a real-world dataset (Iris) using Python and Scikit-learn. Through this lab, students will gain both theoretical clarity and practical experience in building, training, and visualizing decision tree models.

Introduction & Theory

A **Decision Tree Classifier** is a tree-structured model used to make decisions based on feature values. Each internal node represents a test on an attribute, each branch represents an outcome, and each leaf node represents a class label.

The construction of a decision tree is based on:

- **Entropy**: A measure of uncertainty or impurity in a dataset

- **Information Gain**: Reduction in entropy after splitting on an attribute

Decision trees are easy to interpret and visualize but may suffer from **overfitting**, especially on large or complex datasets.

Solved Examples:

Example 1: Entropy and Information Gain

Dataset

Student	Study Hours	Attendance	Result
S1	Low	Poor	Fail
S2	High	Good	Pass
S3	High	Poor	Pass
S4	Low	Good	Fail
S5	High	Good	Pass

1. Entropy of Target Variable (Result)

- Pass = 3
- Fail = 2

$$Entropy(Result) = -\left(\frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5}\right)$$

$$Entropy(Result) = -(0.6 \times -0.737 + 0.4 \times -1.322)$$

$$Entropy(Result) \approx 0.971$$

2. Information Gain for Study Hours

Study Hours = High

- Pass = 3, Fail = 0
- Entropy = 0

Study Hours = Low

- Pass = 0, Fail = 2
- Entropy = 0

Weighted Entropy:

$$= \frac{3}{5} \times 0 + \frac{2}{5} \times 0 = 0$$

$$IG(StudyHours) = 0.971 - 0 = 0.971$$

3. Root Node Selection

Since **Study Hours** provides the **maximum information gain**, it should be selected as the **root node**.

Example 2: Decision Tree on Small Dataset (Python Implementation)

Question

Build and visualize a decision tree using entropy.

Solution:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt

# Create dataset
data = {
    'StudyHours': ['Low', 'High', 'High', 'Low', 'High'],
    'Attendance': ['Poor', 'Good', 'Poor', 'Good', 'Good'],
```

```
    'Result': ['Fail', 'Pass', 'Pass', 'Fail', 'Pass']
}

df = pd.DataFrame(data)

# Encode categorical values
encoder = LabelEncoder()
for col in df.columns:
    df[col] = encoder.fit_transform(df[col])

# Features and target
X = df[['StudyHours', 'Attendance']]
y = df['Result']

# Train decision tree
model = DecisionTreeClassifier(criterion='entropy')
model.fit(X, y)

# Visualize tree
plt.figure(figsize=(10,6))
plot_tree(model, feature_names=X.columns, class_names=['Fail', 'Pass'], filled=True)
plt.show()
```

Example 3: Decision Tree Classifier on Iris Dataset

Objective

Apply decision trees to a real dataset and evaluate accuracy.

Solution

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Load dataset
iris = load_iris()
X = iris.data
y = iris.target

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
```

```
)

# Train model
model = DecisionTreeClassifier(criterion='entropy')
model.fit(X_train, y_train)


# Predict and evaluate
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)

# Visualize tree
plt.figure(figsize=(14,8))
plot_tree(model, feature_names=iris.feature_names,
        class_names=iris.target_names, filled=True)
plt.show()
```

LAB Assignment No. 3

Question 1

Entropy and Information Gain (Manual Calculation)

Given the dataset below about whether students pass an exam based on study time and attendance:

Student	Study Hours	Attendance	Result
S1	Low	Poor	Fail
S2	High	Good	Pass
S3	High	Poor	Pass
S4	Low	Good	Fail
S5	High	Good	Pass

1. Calculate the **entropy** of the target variable (Result).
2. Compute the **information gain** for the attribute Study Hours.
3. Which attribute should be selected for the root node based on maximum information gain?

Question No. 2

Implement Decision Tree Classifier on a Small Dataset

Build and visualize a simple decision tree.

Question:
Using the same dataset as above:

1. Use pandas to create a DataFrame.

2. Convert categorical values into numerical using LabelEncoder.

3. Train a **DecisionTreeClassifier** using **criterion='entropy'**.

4. Visualize the decision tree using plot_tree() from sklearn.tree.

Question 3

Decision Tree Classifier on Iris Dataset

Objective: Apply decision trees to a real dataset.

Question:

1. Load the **Iris dataset** using sklearn.datasets.load_iris.

2. Split it into training (70%) and testing (30%) sets.

3. Train a decision tree using **criterion='entropy'**.

4. Print the accuracy on the test set.

5. Visualize the tree and explain which feature provides the most information gain at the root.

Question 4

MNIST digit dataset (available via Keras / sklearn.datasets) as a baseline

Objectives

- Preprocess image data for classification

- Train a **Decision Tree Classifier** (or variants)

- Evaluate accuracy, confusion matrix, and discuss limitations

LAB Assessment

Student Name		LAB Rubrics	CLO3 , P5, PLO5
		Total Marks	10
Registration No		Obtained Marks	
		Teacher Name	Dr. Syed M Hamedoon
Date		Signature	