LAB No 3

Decision Tree Classifier

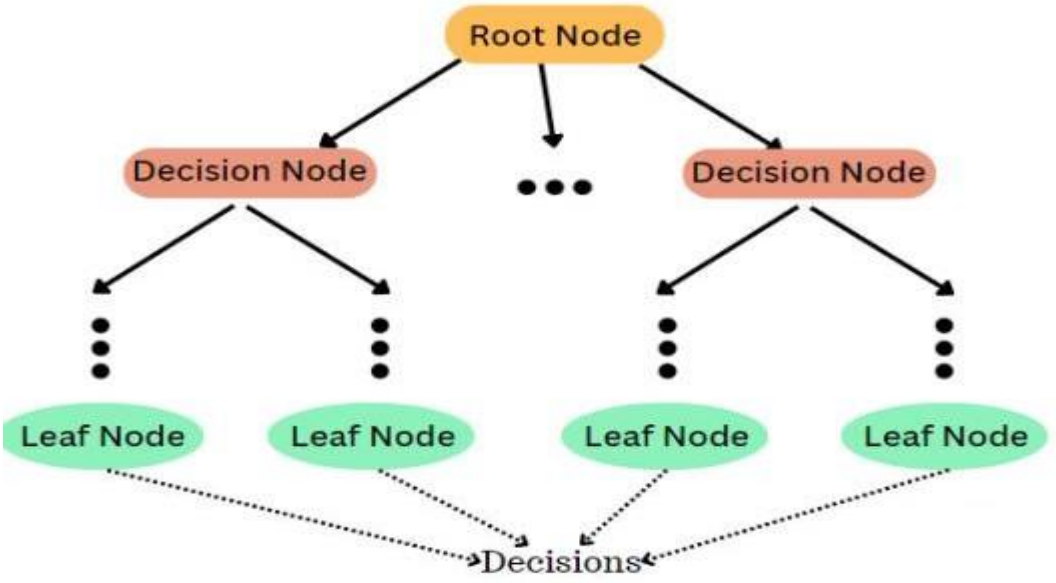
In this lab, students will study and implement the Decision Tree Classifier, a supervised machine learning algorithm used for classification tasks. The lab begins with a manual calculation of entropy and information gain to understand how decision trees choose splitting attributes. Students will then 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)

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)

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:

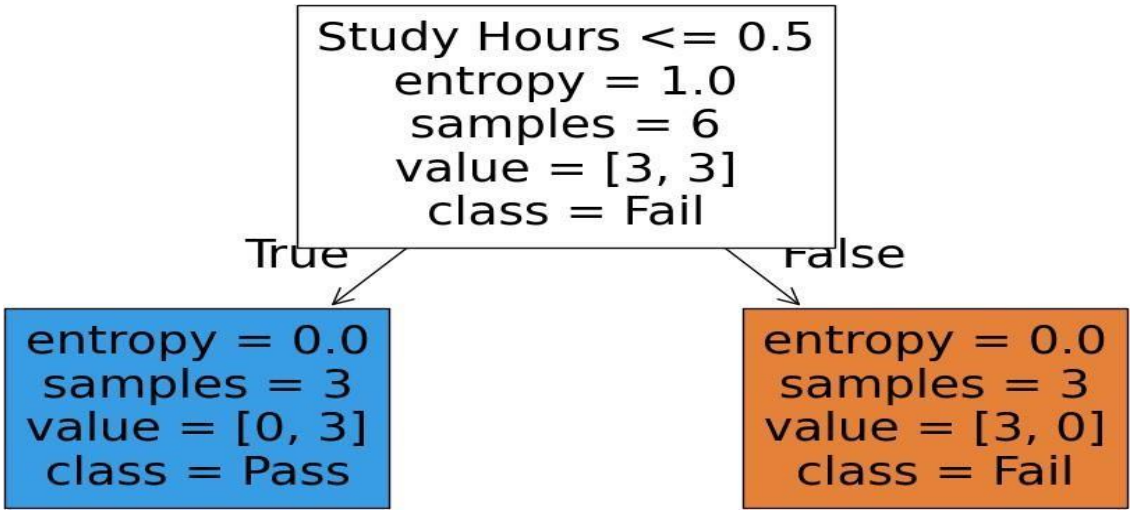
1. Use pandas to create a DataFrame.
2. Convert categorical values into numerical using LabelEncoder.
3. Train a **DecisionTreeClassifier** using **criterion='entropy'**.
4. Visualize the decision tree using plot_tree() from sklearn.tree.

Code:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier, plot_tree import matplotlib.pyplot as plt
data = {
'Study Hours': ['Low', 'Low', 'High', 'High', 'Low', 'High'],
'Attendance': ['Good', 'Poor', 'Good', 'Poor', 'Good', 'Poor'],
'Result': ['Fail', 'Fail', 'Pass', 'Pass', 'Fail', 'Pass']
}
df = pd.DataFrame(data) le = LabelEncoder()
df['Study Hours'] = le.fit_transform(df['Study Hours']) df['Attendance'] =
le.fit_transform(df['Attendance']) df['Result'] = le.fit_transform(df['Result']) X =
df[['Study Hours', 'Attendance']] y = df['Result']
model = DecisionTreeClassifier(criterion='entropy') model.fit(X, y) plt.figure(figsize=(10, 6))
plot_tree(model, feature_names=['Study Hours', 'Attendance'], class_names=['Fail', 'Pass'],
filled=True)
plt.show()
prediction = model.predict([[0, 0]])
result = 'Pass' if prediction[0] == 1 else 'Fail'
print("Prediction for Study Hours=Low and Attendance=Good:", result)
```

Output:

Prediction for Study Hours=Low and Attendance=Good: Pass



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.

Code:

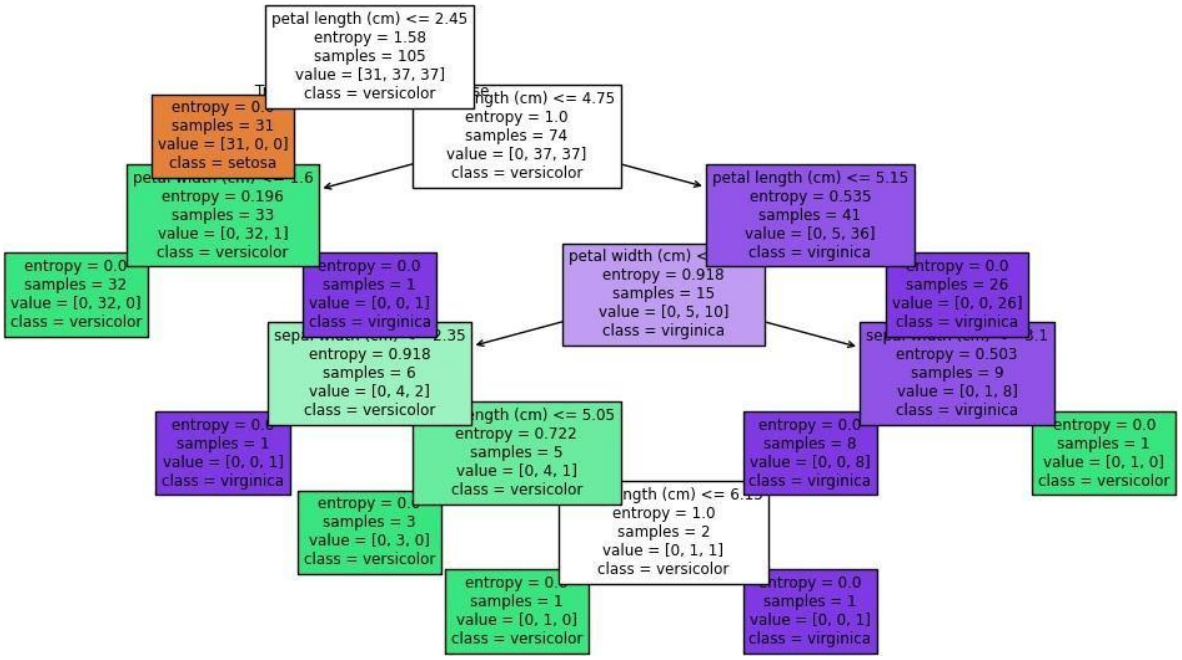
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics
 import accuracy_score
import matplotlib.pyplot as plt iris = load_iris()
X = iris.data y = iris.target
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42) model = DecisionTreeClassifier(criterion='entropy', random_state=42)
model.fit(X_train, y_train) y_pred = model.predict(X_test)
print("✓ Accuracy on Test Set:",
accuracy_score(y_test, y_pred)) plot_tree(
model,
feature_names=iris.feature_names, class_names=iris.target_names,
filled=True) plt.show()
sample = [[5.1, 3.5, 1.4, 0.2]] # example flower measurements
 predicted_class = model.predict(sample)
print(f"Prediction for sample {sample}: {iris.target_names[predicted_class][0]}")
```

Output:

✓ Accuracy on Test Set: 0.G777777777777777

Prediction for sample [[5.1, 3.5, 1.4, 0.2]]: setosa



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

Code:

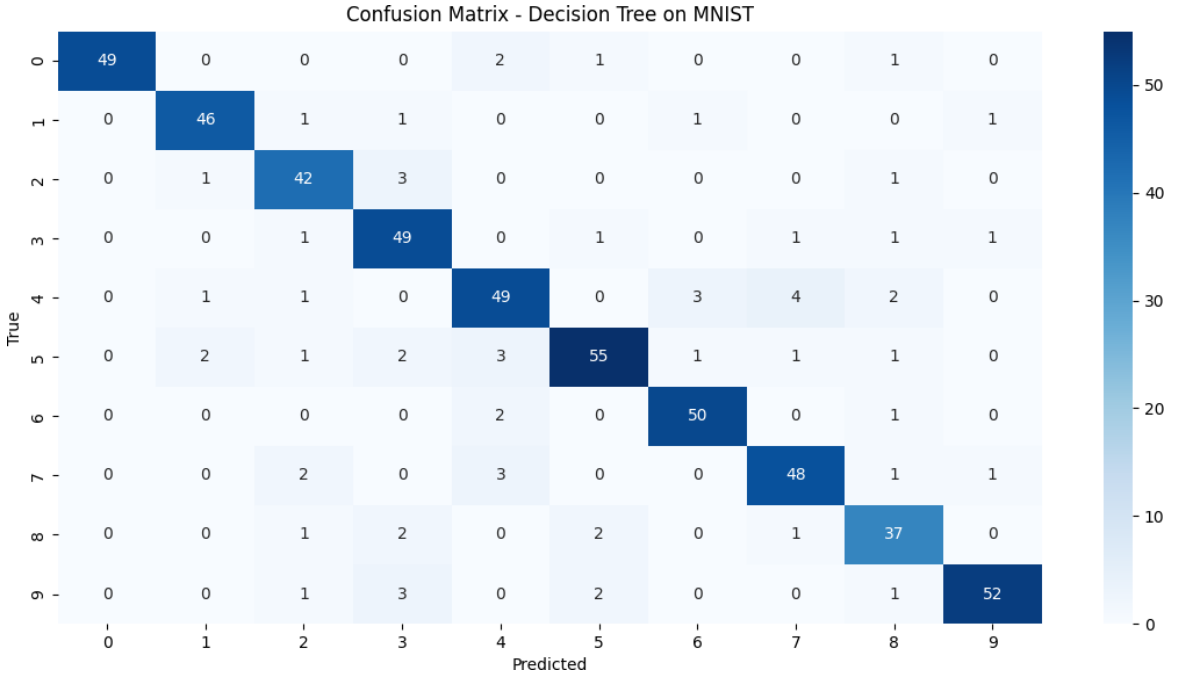
```
from sklearn.datasets import load_digits

from sklearn.model_selection import train_test_split from sklearn.tree import
DecisionTreeClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns

import matplotlib.pyplot as plt digits = load_digits()

X = digits.data# Flattened 8x8 images (64 features per sample) y = digits.target # Labels 0–9

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=42) model
= DecisionTreeClassifier(criterion='entropy', random_state=42)

model.fit(X_train, y_train) y_pred = model.predict(X_test)

acc = accuracy_score(y_test, y_pred) print("■✓ Accuracy on Test Set:", acc)

print("\n■ # ]¡ Classification Report:\n", classification_report(y_test, y_pred)) cm =
confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8,6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues') plt.title("Confusion Matrix - Decision
Tree on MNIST") plt.xlabel("Predicted")

plt.ylabel("True") plt.show()

plt.figure(figsize=(10,2))

for index, (image, label) in enumerate(zip(digits.images[:5], digits.target[:5])): plt.subplot(1,
5, index + 1)

plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest') plt.title(f'True: {label}')

plt.show()
```

Output:

Accuracy on Test Set: 0.8833333333333333

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.G2	0.G6	53
1	0.G2	0.G2	0.G2	50
2	0.84	0.8G	0.87	47
3	0.82	0.G1	0.86	54
4	0.83	0.82	0.82	60
5	0.G0	0.83	0.87	66
6	0.G1	0.G4	0.G3	53
7	0.87	0.87	0.87	55
8	0.80	0.86	0.83	43
G	0.G5	0.88	0.G1	5G
accuracy			0.88	540
macro avg	0.88	0.8G	0.88	540
weighted avg	0.8G	0.88	0.88	540



LAB Assessment

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

Laboratory Work Assessment Rubrics

Sr. No.	Performance Indicator	Excellent (5)	Good (4)	Average (3)	Fair (2)	Poor (1)
1	**Theoretical knowledge 10%**	Student knows all the related concepts about the theoretical background of the experiment and rephrase those concepts in written and oral assessments	Student knows most of the related concepts about the theoretical background of the experiment and partially rephrase those concepts in written and oral assessments	Student knows few of the related concepts about the theoretical background of the experiment and partially rephrase those concepts in written and oral assessments	Student knows very little about the related concepts about the theoretical background of the experiment and poorly rephrase those concepts in written and oral assessments	Student has poor understanding of the related concepts about the theoretical background of the experiment and unable to rephrase those concepts in written and oral assessments
2	**Application Functionality 10%**	Application runs smoothly and operation of the application runs efficiently	Application compiles with no warnings. Robust operation of the application, with good recovery.	Application compiles with few or no warnings. Consideration given to unusual conditions with reasonable	Application compiles and runs without crashing. Some attempt at detecting and correcting errors.	Application does not compile or compiles but crashes. Confusing. Little or no error detection or correction.
3	**Specifications 10%**	The program works very efficiently and meets all of the required specifications.	The program works and meets some of the specifications.	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	The program produces correct results but does not display them correctly.	The program is producing incorrect results.
4	**Level of understanding of the learned skill 10%**	Provide complete and logical answers based upon accurate technical content to the questions asked by examiner	Provide complete and logical answers based upon accurate technical content to the questions asked by examiner with few errors	Provide partially correct and logical answers based upon minimum technical content to the questions asked by examiner	Provide very few and illogical answers to the questions asked by examiner.	Provide no answer to the questions asked by examiner.
5	**Readability and Reusability 10%**	The code is exceptionally well organized and very easy to follow and reused	The code is fairly easy to read. The code could be reused as a whole or each class could be reused.	Most of the code could be reused in other programs.	Some parts of the code require change before they could be reused in other programs.	The code is poorly organized and very difficult to read and not organized for reusability.

6	**AI System Design 10%**	Well-designed AI models. Code is highly maintainable	Good designed AI models and Little code duplications	Some attempt to make AI models. Code can be maintained with significant effort	Little attempt to design AI models and less understanding of code	Very poor attempt to design AI models and its code
7	**Responsiveness to Questions/ Accuracy 10%**	1. Responds well, quick and very accurate all the time. 2. Effectively uses eye contact, speaks clearly, effectively and confidently using suitable volume	1. Generally Responsive and accurate most of the times. 2. Maintains eye contact, speaks clearly with suitable volume and pace.	1. Generally Responsive and accurate few times. 2. Some eye contact, speaks clearly and unclearly in different portions.	1. Not much Responsive and accurate most of the times. 2. Uses eye contact ineffectively and fails to speak clearly and audibly	. 1. Non Responsive and inaccurate all the times. 2. No eye contact and unable to speak 3. Dresses inappropriately
8	**Efficiency** **10%**	The code is extremely efficient without sacrificing readability and understanding	The code is fairly efficient without sacrificing readability and understanding	Some part of the code is efficient and other part of the code is not understandable and work properly	The code is brute force and unnecessarily long	The code is huge and appears to be patched together
9	**Delivery** **10%**	The program was delivered in time during lab.	The program was delivered in Lab before the end time.	The program was delivered within the due date.	The code was delivered within a day after the due date.	The code was delivered more than 2 days overdue.
10	**Awareness of Safety Guidelines 10%**	Student has sufficient knowledge of the laboratory safety SOPs and protocol and is fully compliant to the guidelines	Student has sufficient knowledge of the laboratory safety SOPs and protocol and is Partially compliant to the guidelines	Student has little knowledge of the laboratory safety SOPs and protocol and is Partially compliant to the guidelines	Student has little knowledge of the laboratory safety SOPs and protocol and is non-compliant to the guidelines	Student has no knowledge of the laboratory safety SOPs and protocol and is non-compliant to the guidelines