Practical no.5

**Aim:** Train and fine-tune a Decision Tree for the Moons Dataset

# Theory

Decision trees are a popular supervised learning algorithm used for both classification and regression tasks. In decision tree classification, the algorithm tries to build a tree-like model of decisions and their possible consequences based on the input data. The moons dataset is a synthetic two-dimensional dataset that is commonly used for binary classification problems in machine learning. It consists of two interleaving half circles, making it a non-linearly separable dataset.

# Material

* sklearn

# Program

from sklearn.datasets import make\_moons

from sklearn.model\_selection import train\_test\_split from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score # Generate the Moons dataset

X, y = make\_moons(n\_samples=10000, noise=0.4, random\_state=42) # Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,random\_state=42)

# Create an instance of the Decision Tree classifier

tree\_clf = DecisionTreeClassifier(max\_depth=4, random\_state=42)

# e set the max\_depth hyperparameter to 4, but you can experiment with different values to see how it affects the accuracy of the model.

# Fit the model to the training data tree\_clf.fit(X\_train, y\_train)

# Make predictions on the testing data y\_pred = tree\_clf.predict(X\_test)

# Calculate the accuracy of the model accuracy = accuracy\_score(y\_test, y\_pred) print("Accuracy:", accuracy)

# Output

