Decision tree

from sklearn.datasets import make\_classification

from sklearn import tree

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

import matplotlib.pyplot as plt

X, t = make\_classification(n\_samples=100, n\_features=5, n\_classes=2, shuffle=True, random\_state=1)

X\_train, X\_test, t\_train, t\_test = train\_test\_split(X, t, test\_size=0.3, shuffle=True, random\_state=1)

model = tree.DecisionTreeClassifier()

model = model.fit(X\_train, t\_train)

predicted\_value = model.predict(X\_test)

plt.figure(figsize=(10, 8))

tree.plot\_tree(model, filled=True, feature\_names=[f"Feature\_{i}" for i in range(X.shape[1])], class\_names=["Class 0", "Class 1"])

plt.show()

zeroes = sum(t\_train == 0)

ones = sum(t\_train == 1)

print("Zeroes:", zeroes)

print("Ones:", ones)

prob\_zero = zeroes / len(t\_train)

prob\_one = ones / len(t\_train)

gini = 1 - (prob\_zero\*2 + prob\_one\*2)

print("Gini Impurity:", gini)

match = sum(predicted\_value == t\_test)

accuracy = match / len(t\_test)

print("Accuracy:", accuracy)

Zeroes: 36

Ones: 34

Gini Impurity: -1.0

Accuracy: 0.9333333333333333

Zeroes: 36

Ones: 34

Gini Impurity: -1.0

Accuracy: 0.9333333333333333

Output :

