**Program 1 (Dataset from the sklearn library)**

import pandas as pd

import numpy as np

from sklearn.datasets import load\_iris

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

data=load\_iris()

x=data.data

y=data.target

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,random\_state=50,test\_size=0.25)

**#Do any one of the following numbered 1 to 3 during exam**

**#if you are doing DecisionTreeClassifier() in the plot\_tree you can use classifier as first argument**

**#if you are doing DecisionTreeClassifier(criterion='entropy') in the plot\_tree you can use #classifier\_entropy as first argument**

**#if you are doing DecisionTreeClassifier(criterion='entropy',min\_samples\_split=50) in the plot\_tree #you can use classifier\_entropy1 as first argument**

**#1.default criterion gini**

classifier=DecisionTreeClassifier()

classifier.fit(x\_train,y\_train)

y\_pred=classifier.predict(x\_test)

print('accuracy on train data using gini:',accuracy\_score(y\_train,classifier.predict(x\_train)))

print('accuracy on test data using gini:',accuracy\_score(y\_test,y\_pred))

**#2.entropy**

classifier\_entropy=DecisionTreeClassifier(criterion='entropy')

classifier\_entropy.fit(x\_train,y\_train)

y\_pred\_entropy=classifier\_entropy.predict(x\_test)

print('accuracy on train data using entropy:',accuracy\_score(y\_train,classifier\_entropy.predict(x\_train)))

print('accuracy on test data using entopy:',accuracy\_score(y\_test,y\_pred\_entropy))

**#3.entropy with min\_samples\_split**

classifier\_entropy1=DecisionTreeClassifier(criterion='entropy',min\_samples\_split=50)

classifier\_entropy1.fit(x\_train,y\_train)

y\_pred\_entropy1=classifier\_entropy1.predict(x\_test)

print('accuracy on train data using entropy:',accuracy\_score(y\_true=y\_train,y\_pred=classifier\_entropy1.predict(x\_train)))

print('accuracy on test data using entopy:',accuracy\_score(y\_true=y\_test,y\_pred=y\_pred\_entropy1))

#To visualize import these libraries

from sklearn.tree import plot\_tree

import matplotlib.pyplot as plt

# Visualize the decision tree

plt.figure(figsize=(15, 10))

plot\_tree(classifier, feature\_names=data.feature\_names, class\_names=data.target\_names, filled=True,rounded=True)

plt.title("Decision Tree Visualization (Gini Criterion)")

plt.show()

**Program 2 (Dataset saved in the working directory)**

import pandas as pd

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 the dataset

data = pd.read\_csv('diagnosis.csv') # Ensure this file is in the working directory

# Features (X) and Target (y)

X = data[['sore\_throat', 'fever', 'swollen\_glands','congestion', 'headache']]

y = data['diagnosis'] # Target column

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=50, test\_size=0.25)

**#Do any one of the following numbered 1 to 3 during exam**

**#if you are doing DecisionTreeClassifier() in the plot\_tree you**

**# can use classifier\_gini as first argument**

**#if you are doing DecisionTreeClassifier(criterion='entropy') in the plot\_tree you**

**# can use classifier\_entropy as first argument**

**#if you are doing DecisionTreeClassifier(criterion='entropy',min\_samples\_split=50) in the plot\_tree you**

**# can use classifier\_entropy\_split as first argument**

**#1.Default Decision Tree using Gini criterion**

classifier\_gini = DecisionTreeClassifier()

classifier\_gini.fit(X\_train, y\_train)

y\_pred\_gini = classifier\_gini.predict(X\_test)

print('Accuracy on training data using Gini:', accuracy\_score(y\_train, classifier\_gini.predict(X\_train)))

print('Accuracy on test data using Gini:', accuracy\_score(y\_test, y\_pred\_gini))

**#2.Decision Tree using Entropy criterion**

classifier\_entropy = DecisionTreeClassifier(criterion='entropy')

classifier\_entropy.fit(X\_train, y\_train)

y\_pred\_entropy = classifier\_entropy.predict(X\_test)

print('Accuracy on training data using Entropy:', accuracy\_score(y\_train, classifier\_entropy.predict(X\_train)))

print('Accuracy on test data using Entropy:', accuracy\_score(y\_test, y\_pred\_entropy))

**#3.Decision Tree using Entropy with min\_samples\_split = 50**

classifier\_entropy\_split = DecisionTreeClassifier(criterion='entropy', min\_samples\_split=50)

classifier\_entropy\_split.fit(X\_train, y\_train)

y\_pred\_entropy\_split = classifier\_entropy\_split.predict(X\_test)

print('Accuracy on training data using Entropy (min\_samples\_split=50):',

accuracy\_score(y\_train, classifier\_entropy\_split.predict(X\_train)))

print('Accuracy on test data using Entropy (min\_samples\_split=50):',

accuracy\_score(y\_test, y\_pred\_entropy\_split))

**# Visualize the Decision Tree (using Gini criterion as an example)**

plt.figure(figsize=(15, 10))

plot\_tree(classifier\_gini,

feature\_names=X.columns,

class\_names=y.unique().astype(str),

filled=True,

rounded=True)

plt.title("Decision Tree Visualization (Gini Criterion)")

plt.show()

**Program 3 (Dataset is created using dataframe)**

import pandas as pd

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

# Creating a simple dataset

data = {

'Age': [25, 30, 35, 40, 45, 50, 55, 60],

'Income': [50000, 60000, 70000, 80000, 90000, 100000, 110000, 120000],

'Purchased': [0, 0, 0, 1, 1, 1, 1, 1] # 0: No, 1: Yes

}

df = pd.DataFrame(data)

# Splitting the data into features (X) and target (y)

X = df[['Age', 'Income']]

y = df['Purchased']

# Splitting into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

#Do any one of the following numbered 1 to 2 during exam

#if you are doing DecisionTreeClassifier() in the plot\_tree you

# can use classifier as first argument

#if you are doing DecisionTreeClassifier(criterion='entropy') in the plot\_tree you

# can use classifier\_entropy as first argument

#1.Train the decision tree classifier with Gini criterion

classifier = DecisionTreeClassifier()

classifier.fit(X\_train, y\_train)

# Predict and evaluate

y\_pred = classifier.predict(X\_test)

print("Accuracy on train set (Gini):", accuracy\_score(y\_train, classifier.predict(X\_train)))

print("Accuracy on test set (Gini):", accuracy\_score(y\_test, y\_pred))

#2.Train another decision tree classifier with Entropy criterion

classifier\_entropy = DecisionTreeClassifier(criterion='entropy', random\_state=42)

classifier\_entropy.fit(X\_train, y\_train)

# Predict and evaluate for entropy

y\_pred\_entropy = classifier\_entropy.predict(X\_test)

print("Accuracy on train set (Entropy):", accuracy\_score(y\_train, classifier\_entropy.predict(X\_train)))

print("Accuracy on test set (Entropy):", accuracy\_score(y\_test, y\_pred\_entropy))

# Visualizing the decision tree (Gini)

plt.figure(figsize=(12, 8))

plot\_tree(classifier, feature\_names=['Age', 'Income'], class\_names=['No', 'Yes'], filled=True, rounded=True)

plt.title("Decision Tree Visualization (Gini Criterion)")

plt.show()