**Assignment 4 (Decision Tree Classifier technique)**

import numpy as np

import pandas as pd

from sklearn import tree

import matplotlib.pyplot as plt

data = {

'age': ['<21', '<21', '21-35', '>35', '>35', '>35', '21-35', '<21', '<21', '>35', '<21', '21-35', '21-35', '>35'],

'income':['high','high','high','medium','low','low','low','medium','low','medium','medium','medium','high','medium'],

'gender':['male','male','male','male','female','female','female','male','female','female','female','male','female','male'],

'marital\_status':['single', 'married', 'single', 'single', 'single', 'married', 'married', 'single', 'married','single','married','married','single','married'],

'buys':['no','no','yes','yes','yes','no','yes','no','yes','yes','yes','yes','yes','no']

}

df = pd.DataFrame.from\_dict(data)

X=df.iloc[:,[0,1,2,3]].values

y=df.iloc[:,-1].values

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

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

from sklearn.preprocessing import LabelEncoder,OneHotEncoder *#to convert categorical values to integer values*

encoder=LabelEncoder()

X\_train[:,0]=encoder.fit\_transform(X\_train[:,0])

X\_train[:,1]=encoder.fit\_transform(X\_train[:,1])

X\_train[:,2]=encoder.fit\_transform(X\_train[:,2])

X\_train[:,3]=encoder.fit\_transform(X\_train[:,3])

X\_test[:,0]=encoder.fit\_transform(X\_test[:,0])

X\_test[:,1]=encoder.fit\_transform(X\_test[:,1])

X\_test[:,2]=encoder.fit\_transform(X\_test[:,2])

X\_test[:,3]=encoder.fit\_transform(X\_test[:,3])

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier()

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

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

classifier.predict([[1,1,0,0]])

y\_pred

from sklearn.metrics import confusion\_matrix,accuracy\_score

cm=confusion\_matrix(y\_test,y\_pred)

cm

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

**Output:**

0.8