#Decision tree classifier using sklearn

import pandas as pd

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

import numpy as np

from sklearn.tree import DecisionTreeClassifier

from sklearn.preprocessing import LabelEncoder

#Load the data, display the type and contents of the dataset

dataset = pd.read\_csv("tennis.csv")

print(dataset)

all\_cols=dataset.columns

feature\_cols=all\_cols[1:5]

print(feature\_cols)

#feature encoding

for label in dataset.columns:

dataset[label]=LabelEncoder().fit\_transform(dataset[label])

# formating into numpy array

data=dataset.drop(['play'],axis=1)

data=np.array(data.drop(['day'],axis=1))

target=np.array(dataset['play'])

#Splitting the dataset

X\_train,X\_test,y\_train,y\_test=train\_test\_split(data, target, test\_size=0.2)

# Train Decision Tree Classifier

id3 = DecisionTreeClassifier()

id3 = id3.fit(X\_train,y\_train)

#Predict the response for test dataset

y\_pred = id3.predict(X\_test)

print("Actual output:",y\_test)

print("predicted output:",y\_pred)

#Classify a new example

New\_example=np.array(['sunny','Hot','High','Weak'])

encoded=LabelEncoder().fit\_transform(New\_example)

encoded=encoded.reshape(1,-1)

new\_pred=id3.predict(encoded)

print(new\_pred)