<u>**Practical No. - 7**</u>

Name- Bhavesh Kewalramani

Roll No.- A-25

Section- A

Semester- 6th

Shift- 1st

Aim:

Write a program to implement Decision Tree Algorithm. Your aim is to find root node of decision tree for the given dataset. (using Gain and Entropy)

Code:

```
import csv

import time

import math



def IndependentEntropy(row1,row2,unq,dec):

    dict1 = {}

    dict2 = {}

    for i in range(len(unq)):

        ent=0

        for k in range(len(dec)):

            count1=0
```

```
            count2=0
            for j in range(len(row1)):
                if unq[i] == row1[j] and row2[j] == dec[k]:
                    count1+=1
                if unq[i] == row1[j]:
                    count2+=1
            if count1==0:
                ent-=0
            else:
                ent -= ((count1/count2)*math.log(count1/count2,2))
                dict2.update({unq[i]:count2})
        dict1.update({unq[i]:ent})
    return dict1,dict2



a = []


#filepath = input("Enter Path of CSV File : ")
filepath = "C:\\Users\\bhave\\Downloads\\PlayTennis.csv"


print("\n")


start = time.time()
```

```
with open(filepath,'r')as file:
    filecontent=csv.reader(file)
    for row in filecontent:
        print(row)
        a.append(row)

headings = a[0]
unique=[]
d={}

a.remove(a[0])

temp = [[] for _ in range(len(headings))]

for i in range(len(a)):
    for j in range(len(a[i])):
        temp[j].append(a[i][j])

n=len(headings)
m=len(temp[0])

temp1 = [[] for _ in range(len(headings))]
```

```
for i in range(len(temp)):
    for j in range(len(temp[i])):
        if temp[i][j] not in temp1[i]:
            temp1[i].append(temp[i][j])
    d.update({headings[i]:temp1[i]})

n1 = temp1[-1]
temp1.remove(temp1[-1])


print("n1",n1)
print("temp1",temp1)
print("temp",temp)


d1={}
d2={}
for i in range(len(n1)):
    count=0
    for j in range(len(temp[-1])):
        if n1[i]==temp[-1][j]:
            count+=1
    d1.update({n1[i]:count})

print(d1) #Total Yes/No
```

```
tot=0

for k,v in d1.items():
    tot+=d1[k]

totalEntropy=0
for k,v in d1.items():
    totalEntropy -= ((v/tot)*math.log(v/tot,2))

print("\n Total Entropy : ",totalEntropy)

print("\n\n",temp[0])
print("\n\n",temp[-1])
print("\n\n",temp1[0])


dict3={}
for i in range(len(temp)-1):
    dict1,dict2 = IndependentEntropy(temp[i],temp[-1],temp1[i],n1)
    gain=totalEntropy
    t1=0
    for k,v in dict2.items():
```

```
        t1+=v
    for k,v in dict1.items():
        gain-=(dict2[k]/t1)*dict1[k]

    dict3.update({headings[i]:gain})

print(dict3)


maxx=-99999
root=""
for k,v in dict3.items():
    if maxx < dict3[k]:
        maxx=dict3[k]
        root=k

print("\n\n")
print("Root Node is ",root)
print("Value : ",maxx)
end=time.time()
print("\n\n")
print("Time Taken By The Algorithm : ",end-start)
```

Output:

```
['Outlook', 'Temperature', 'Humidity', 'Wind', 'Play Tennis']
['Sunny', 'Hot', 'High', 'Weak', 'No']
['Sunny', 'Hot', 'High', 'Strong', 'No']
['Overcast', 'Hot', 'High', 'Weak', 'Yes']
['Rain', 'Mild', 'High', 'Weak', 'Yes']
['Rain', 'Cool', 'Normal', 'Weak', 'Yes']
['Rain', 'Cool', 'Normal', 'Strong', 'No']
['Overcast', 'Cool', 'Normal', 'Strong', 'Yes']
['Sunny', 'Mild', 'High', 'Weak', 'No']
['Sunny', 'Cool', 'Normal', 'Weak', 'Yes']
['Rain', 'Mild', 'Normal', 'Weak', 'Yes']
['Sunny', 'Mild', 'Normal', 'Strong', 'Yes']
['Overcast', 'Mild', 'High', 'Strong', 'Yes']
['Overcast', 'Hot', 'Normal', 'Weak', 'Yes']
['Rain', 'Mild', 'High', 'Strong', 'No']
n1 ['No', 'Yes']
temp1 [['Sunny', 'Overcast', 'Rain'], ['Hot', 'Mild', 'Cool'], ['High', 'Normal'], ['Weak', 'Strong']]
temp [['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Rain', 'Overcast', 'Sunny', 'Sunny', 'Rain', 'Sunny', 'Overcast', 'Overca
st', 'Rain'], ['Hot', 'Hot', 'Hot', 'Mild', 'Cool', 'Cool', 'Cool', 'Mild', 'Cool', 'Mild', 'Mild', 'Mild', 'Hot', 'Mild'], ['H
igh', 'High', 'High', 'High', 'Normal', 'Normal', 'Normal', 'High', 'Normal', 'Normal', 'Normal', 'High', 'Normal', 'High'],
['Weak', 'Strong', 'Weak', 'Weak', 'Weak', 'Strong', 'Strong', 'Weak', 'Weak', 'Weak', 'Strong', 'Strong', 'Weak', 'Strong'],
['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'No']]
{'No': 5, 'Yes': 9}

 Total Entropy :  0.9402859586706309
```

```
 Total Entropy :  0.9402859586706309


 ['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Rain', 'Overcast', 'Sunny', 'Sunny', 'Rain', 'Sunny', 'Overcast', 'Overcast',
'Rain']


 ['No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'Yes', 'No']


 ['Sunny', 'Overcast', 'Rain']
{'Outlook': 0.2467498197744391, 'Temperature': 0.029222565658954647, 'Humidity': 0.15183550136234136, 'Wind': 0.048127030408269
32}



Root Node is  Outlook
Value :  0.2467498197744391



Time Taken By The Algorithm :  0.0031003952026367188
```

]: