**PRN : 117A1002**

**EXPERIMENT**

**AIM :** To implement ID3 Decision Tree algorithm

**PROGRAM:**

import math

age=[]

competition=[]

T=[]

profit=[]

for i in raw\_input("Enter dataset").split("\n"):

(a,c,t,p)=i.split(" ")

age.append(a),competition.append(c),T.append(t),profit.append(p)

info=0

for i in set(profit):

p= profit.count(i)/len(profit)

info-=p\*(math.log(profit.count(i),2)-math.log(len(profit),2))

def calsub(label,profit):

d={}

for i in set(label):

d[i]=[label.count(i),0,0]

for i in range(len(profit)):

if profit[i]=="D":

d[label[i]][1]+=1

else:

d[label[i]][2]+=1

#calinfo (d,profit)

s=0.0

for i in d:

i=d[i]

if 0 in i:

s+=0

continue

f=(float(i[0])/len(profit))

mf=0.0

for j in range(1,3):

mf-=(math.log(i[j],2)-math.log(i[0],2))\*i[j]/i[0]

s+=mf\*f

return(1-s)

def findroot(profit,age,competition,T):

a=calsub(age,profit)

b=calsub(competition,profit)

c=calsub(T,profit)

ele=max(a,b,c)

#print 'ele= ',ele

if ele==a:

root,root\_n=age,'age'

elif ele==b:

root,root\_n=competition,'competition'

else:

root,root\_n=T,'T'

return root,root\_n

def findnext(profit,age,competition,T,current,current\_n):

types=[i for i in set(current)]

n=len(profit)

#print 'types',types

for i in types:

subprofit=[]

for j in range(len(profit)):

if current[j]==i:

subprofit.append(profit[j])

#print('subprofit=',subprofit)

if len(set(subprofit))==1

print 'if ',current\_n,':',i,'=>',subprofit[-1]

else:

'''for j in range(n)

if age[j]!=i:

print(j,"$",age[j])

age.pop(j)

competition.pop(j)

profit.pop(j)

T.pop(j)

print age

print current'''

print 'if ',current\_n,':',i,'=>'

root,root\_n=findroot(profit[3:7],age[3:7],competition[3:7],T[3:7])

#print('root=',root,'root\_n',root\_n)

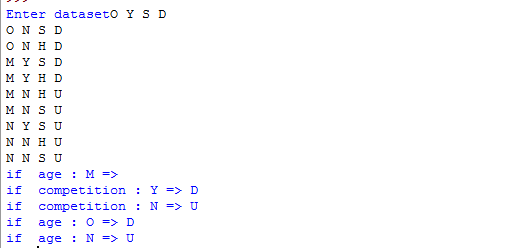
findnext(profit[3:7],age[3:7],competition[3:7],T[3:7],root,root\_n)

root,root\_n=findroot(profit,age,competition,T)

#print root

findnext(profit,age,competition,T,root,root\_n)

**OUTPUT:**



**CONCLUSION:**

Thus the ID3 Decision Tree algorithm has been implemented.