```
LAB - 2: - 103 Algorithm: ( annula some side)
 import numpy as no
 import mapplot116. pyplot as plb
                            ine also beforesport can be
import pardos as pd
import math
import copy
                              eno xbitore aug xora motor
 dataset = pd need - esv ( / content / weather (34)
X = datoset iloc [:, :] . values
attribute = ['outlook," Temp', 'Humidity', 'wind']
                                      to billion of the
class Node Cobject):
          sey. volve = None 1= 5 word 6 topper
     del _init- (sell):
          gey decision = None by - [ you ] tiber 389
          sey, child = None
del find Entropy (data, rows):
                                             to your us hay so
      yes = 0
       no = 0
       ans = -1
      idz=len(data[o])-1
       enhopy=0
       for i in grows:
           if data[i][idx] == 'Yes':

yer=gert1
                                              d= 40/ (do 10)
       y = no | (yest no) | (yest tribum) = + mug

y = no | (yest no) | (60) Spal mon by

if x! = 0 and y! = 600 Spal mon by
                                          0= 16 pur 0= 1x 8
             en tropy = - 10 ( or & math. log 2 (2) + y * math. log 2(y))
                                         return moxtain, rendx, ans
         nehm enhopy, ans
```

```
Backspace
def findmex quin (data, snows, whemas):
     moxGain = 0
                              did so daigled dilladgeory
     entropy, and = find Entropy (data, siows)
     if enhopy = =0:
        greturn max gain, sretidx, ans
                    v2) water (motors) ) v23- know by
    for j' in columns :
                                 soulov. [: : ] soli toxolox
        mydict = 23
        idx - [ bullook remp Humidaly world ] - steel
        for I in slows:
           key = dotalillidx]
           if key not in mydict: :(198) sin
                                       101 _ mit _ (3016):
           else mydict[key] = mydict[key] + prob 100
      gain = entropy
                                  (definity (data, rows):
      for key in mydict:
          yes=0
           no =0
          for kin grows:
             if data[k][j] = key: i- (tolotob) rat =
                  if data [h][-1] == 'Yes':
                  else jest som i roj
                      no=noti - [sbi][i] otalo [i
         x = yes (Cyertno)
         g = no/(yes +no)
                                   1+00 = 04
        of x1,00 and y1=0:
            gain to (my dict [key] * (24 moth log 2(x) +
                    y + moth . dog 2(y))) / 14
      gain margain : de dome se ) of - popular
           maxigain = goun
          retida = j
rehm moxigain, retida, ans
```

surprise makes ours

```
dep buildince (data, 910 ws, columns).
    morgain, idx, ans = find mox Gain (x, snows, columns)
     good = Node()
     goot childs = []
     il morgain = = 0:
         il anser 1%
            good value = 'Yes'
            root value = 'No'
        noturn scoot
                                  system : Stort value control
     groot value = attribute [idx]
                                    I busion. Sunny, Value, Hum
     mydict - 23
                                   Deesson Hope voles was
    for i in grows:
                                  - Deason women volue : " Ex
         key > = data[i][ida]
                                    - Down overcost, volue to
         if key not in mydict:
                                     - Ocusion Rom volue world
            mydict[hey] = 1
                                    Pougnos: Week, Value : 408
                                    Decision strong value No
         else mappiet [ney] +=1
    new columns = copy. deep copy (whomns)
    newcolumns. siemove (ida)
    for key in mydict
          new grows = []
          for in 90005:
              i) data[i][id1] == key:
                newsous-append(i)
          temp = build Tree (data, new Grows, new columns)
         temp. decision = key
         groot childs append (temp)
    relum root.
del traverse (root, level = 0):
    print (1° of indent of + Decision: Egroot. decision y, value: Egroot. values)
    for i, child in enumerate (900+.childs):
          if i = = len ( 900+ childs) -1:
                troverse (child, level +1)
                          (1) leveltl)
```

dej Calendate () 9lows = [for ? in gange (0,14)] columns = [i for i in garge (0,4)] groot = buildree (x, grows, columns) groot decision = 'start' 191 - July too moverse (groot) Calculate () 'ma' = sulov, four Joone and output: I- beusion: Sunny, value: Humiday & bid students a wiley + Decision: Stort, value: Loutlook t peusion: High, value: No - Deutsion: Normal, value: Yes usu @ dota[i][idx]. + Deursian: overeast, valuer yes Leasy por in mydict 1- neurion: Rain, volue storne mydid[hey] = 1 I neers non: week Trable 1488 1=+ (post tibus 1=1 + Decision: 3trong, value: No umns s copy-deep why laboren unins. Sumove (181) if dato[i][dr] == key: hewrows opport mp = build Tree (dota, newspans, newalumns) mp. decision = key of Unilds append (temp) : (0 - loval toor (heart) + Decision: Emost decision & volue: Local volues) ald in enomerate (moot childs): 1- ablib took I not -! troverse (child, level+1) (itland, blid) or