

NAME: AFRIN FATHIMATOT SEC.: ROLL NO.: SUB.:

s. No.	Date	Title	<del>Pag</del> e	Teacher's Sign / Remarks
	31.7.2024	Paris Dukta D	marles	The
	01.4.0024	Basic Python Programmes	60	H. P.
2	\$.8.2024	in Google cotab Email Span setection	w	Ha
3	4/9/2024	N- Queens	w	76
4	11/9/2024	A* slarch	10	100
6	18 9 12024	Depth First search	10	***************************************
6,	81912024	AO* Algorithm	10	No.
7)	25/9/2024	Decision Tree	10	X
	9/10/2024	K-Means	10	1/9
8,	16/10/2024	Artificial Newal Network	10	Xo
9,	23/10/2024	Minikan	10	1
10,	30/10/2024	Introduction to Prolog	10	Ny Ny
11)	6/11/2024	Protog family Tree	10	X
		J. J		
		Street And I want to the street and		
1				

```
Palindrome:
        word = input()
         ij word == word [::-1]:
        print (word, " is a palindromi)
                          print (word, " is not a palindrome")
2) Prime Number : Prime vit mins ) tugal) in a co
                                                                          numbers to gent
           num = int (input ("Entir a number:"))
          flag = False
                                                                                              11. Education of a number:
          if num == 1;
                         print ( num, " is not a prime number")
                                                                                                                                             W. Deco.
           elif num >1:
                         for i in range (2, num):
                                               if (num ', i)= = 0:
flag = True
                                          ((" bolocak a restated) toget ) the a parameter
                        if plag: " is the special of the second of t
                                             print (num, " is not a prime number")
                                                                                                              rice fundadu Sara (rice):
                        else:
                                            print (num, "is a prime number")
           else:
                         print (num, is not a valid number to chuck
                     [ ] for prime")
                                                                                                                                            A HILLIAM A
            was the Cont input ( Enter mankers by speci +)
```

```
Fibonacci Svils:
   det fibonacci (n):
       for i in range (2, n):
           next-number = fib-sequence [-1] + fib-sequence[-2]
           fib-sequence. append (next-number)
       return fib-siquence
   n = int (Input ("Enter the number of fibonacci
                   numbers to generate !"
   print (fibonacci (n))
                                         PALINUT
As Factorial of a number:
   def factorial (n): " that's a loss the " manes ) the inter
      if n==0:
          for the shade (2) ments. I have
      else
         outwin n* factorial (n-1)
   number: int (input ("Enter a number:"))
   print ( Factorial of , n, is mumber ) is
   Bubble Sort : 10 5 0 500 & " ( prod. ) desired
50
   det Bubble Sort (an):
                                         · della.
      for i in range (n)?
         for j in range (0, n; -1):
         if (outj] > outj+i]:
                ansij, ausijii] = ansijii], ausij
of networn and
   numbers: list (map (int, input ("Enter numbers by spaces:")
                                  · split(1))
   sorted array = Bubble Sort (numbers)
```

print/ conted array)

```
6) Asimstrong Number:
num: int (input ("Enter a number"))
                     ment but ( should ? Supply a surplier
sum =0
              will are Corput (" Sugar a survibil")?
temp = num
while temp >0: (smust = = 1 must bas (sauni = 1 mont) p
   digit = temp 1.10
   sum + = digit **3
   temp/1 = 10
                             SHOULD I PLUMP
If num = = sum :
   print (num, "is an Armstrong number")
else:
   print (min, "is not an Axonstrong number")
                                         Statepart:
Output:
                                Citi Rodinun D Palate
Enter a number: 153
153 is an Armstrong number.
7) Sum of n natural numbers.
 num = int (input ("Enter a number"))
 of (num 20)
     print ("Enter a positive number")
 else
     sum =0
    while (num >0).
        sum + / num
        nun = num -1
    poure ("The sum is", sum)
  Output
  Enter a number: 10
                  les mens of in affect susapports : 10
  The sum is 55
                   g: books and h to mor my
```

```
8) Largest of
 num: int (input ("Enter a number"))
            3 nos:
 hum 2 = int (input (" Enter a number"))
  num 3 = int Cinput ("Entor a number"))
  4 (num1 > = num2) and (num1 > = num3):
       largest = num!
  elseif (num2>= num1) and (num2>= num3):
        largest = numa
  else
       largest = num3 portunish to the num i hours
  point ("The largest number is:", largest)
  Output:
  Enter a number: 12
  Enter a number: 5
                            se is an Amort one runner.
  Enter a number: 54
   The largest number is: 54
9) Swap 2 variables:
  x=5
  4=10
  temp = x
  n = y
  y = temp
  print ( 'The value of x after swapping: ', x'))
  print ('The value of y after swaping:', y')
 Output:
 The value of n after swapping: 10
```

```
101 LCM of 2 input numbers:
  def compute. ecm (2,y):
      if x > y:
        greater: x
      else:
          greater = 4
      while (True):
           if ((greater 1, n = = 0) and (greater 1, y == 0)):
                lon = greater break
          greater +=1
      return lem
   num 1 = 54
   num 2 = 24
   print (" The sem is: ", compute_lcon (num1, num2))
   Output:
    The LEM is: 216
```

```
04/09/2024
                       N - QUEENS
  PROGRAM:
  de is-safe (board, now, col):
  for i in range (col):
           if board [now][i] ==1:
               retwo False
     for i, j in zip (nange (now, -1, -1, nange (col, +, -1)).
          if board [i][j]==1:
                return False.
    for i, j in zip (range (now, len (board), 1), range (col,
          if board (i)[j] == 1:
               return False
     Retwin True
 def solve-n-queens (boowed, col):
      if col > = len (board):
            retwon True
     for i in range (len (board)):
            if is_safe (board, i, col):
                 board [i] [col] =1
            if solve_n-queens (board, col +1):
                 return Treu
            board [i][col] = 0
      return False
```

def print-board (board):

for now in board:

print (" ". join (str(x) for x in row))

```
def. solve ():
    n = 8
    board = [[ofor - in range (n)] for - in range [n)]
    if not sowe_n_queens (board, 0):
        print ("Solution does not exist")
        retwen False.
    print_board (board)
    section True
               parants [stance node] = stance node
solve()
                    while Levelopen set ) > 0:
Output:
  10000000 - DA 10000 N V 10)
if (m= none or 9[v] + here 2008 (9) 71
  0000100
  0000001
  01000000
 if (n= stop nock or grapho or dy or p o 600):
 00000100
                            bows pass
 00,00000
                                   : 3633
Thus the program was executed and the output was
 verified.
           in dored at ):
      glmJ= grnJ + assight
    it aling - [ulp + mader
    fruit - o [e] + andin
```

## A\* algorithm

ref [(1) people in rot of) - Klopy PROGRAM: astaralgo (start-node, stop-node): open-set = set (start-node) closed\_set = set() successor Fabre 9= 33 (beard Growed) pounts = 33 surry True g[start\_node] =0 parents [start\_node] = start\_node while len(open\_set) > 0: REPORT n=None for v in open-set: if (n== None or g[v] + hewistic(V) < 9 + hewistic(n): nIV If (n== stop-node or Graph=nodus[n] == None) pars else: for (m, weight) in get-neighbows (n): if (m not in open-set and m not in cosed-set): open-set. add(m) parcents [m] = n g[m]= g[n] + weight else: if g[m] > g[n] + weight: g[m] = g[n] + weight

parents [m] = n

```
if m in dosed-set:
          Closed-set. gemove (m)
                         Open-set. add (m)
       if (n== None):
            print ("path does not exist")
           retwen None
       if (n== stop-neals): ('n) A') appended
            part:[]
            while pavents [n] ! = n:
                                            FUSTUR
                path. append (n)
                 n = pounts [nj] ' A'] + bours Alest
            path. append ( start_node)
            path. suverse ()
            print (" Path Found: ??", format (path))
            return path
       open-set. seemove (n)
       closed - set add (n)
  print (" path does not exist!")
   return None
def get-neighbors (V):
    if v in Graph-nodus:
       return graph-nodes[v]
    else:
        Ketwin None
det hewistic (n):
   H-dipt= {'A': 11, 'B': 6, 'C': 99, 'D': 1, 'E': 7,
             'q':0,3
   return H_dist[n]
```

Graph\_nodus = { A : [(B,2), ('E',3)], B': [('C',1), ('Gi,9)], ( None, 'E': [('D'; 6)], il him Dong [('Gi', 1)], wing astarago ('A', 'Gi') ( whom got = 1) ] []: Mext OUTPUT : Path Found : ['A', E', D', on'] path. append ( stone note) print (" Path Found ! ? ] . Jonnas (path) receive path Open sel second (11) closed - set add (a) print (" park des not extent?) if y me Graph - nocks:[v] suction Groph-nodus[v] : 31/2 Thus the program was executed and output vas verified Carlotte H. dist [1]

### DEPTH FIRST SEARCH

```
PROJRAM:
dy add edge (adj, s, t):
    adj [s]. append (t)
    adj[t]. append (5)
dy ds_nec(adj, virited, s):
    Visited [S] = True
    print (s, end = "")
    for i in adj[s]:
        if not visited [i]:
              des_rec(adj, visited, i)
dy des (adj, s):
     visited = [False] * lin (adj)
     dfs-ruc (adj, visited, s)
V=5
adj = [[] for - in range (V)]
edges = [[1,2],[1,0],[2,0],[2,3],[2,4]]
for e in edges:
     add_edge (adj, e[o], e[i])
source = 1
Print ("DFS from source: ", source)
des (adj, source)
DES from cowice: 1
12084
RESULT !
    Thus the pragram was executed and the output
was verified
```

# IMPLEMENTING APTIFICIAL NEURAL NETWORKS FOR AN APPLICATION USING PYTHON - REGRESSION

EX NO: DETE 75/9/2024

AIM:

To implement outificial number number for an application in Regression using python.

#### EXPLANATION!

\* Generate synthetic sugression does using moss-signession with 1000 samples. 100 features and noise of 0.05.

\* 3 plit data into training (80%) and testing (20%) - suts using train-test-split.

\* Initialize the MLP Regressor model with a maximum of 1000 iterations.

\* Fit the model to the training data using of fit (X.Y)

\* Evaluate the model's performance by collecting by printing the R2 score for both the training and testing doesnots.

#### SOURCE CODE:

from sklean humal-network import HLP Regressor
from sklean model skeetion import train test-split
from execute import make-regression
import numpy as np.
import matploteib pyplot as plt
import seaborn as soss
'! matplotlib inline

X, y = make-regression (n-samples = 1000, hoise=0.05, n-features=

X-train, X-test, y-train, y-test = train-test-split (x, test-size = 0.2, Shufflerandom-state = 42)

elf = MLP Regressor (max\_iter=1000) elf.fit (x-train, y-train)

print (f" R2 Score for Training Data: ? clf. score(x-train, y-train)?

printf (f "R2 score for Test Data: {clf. score (X-test, y-test,

OUTPOT: AC IS PAGES THE PROPERTY SEEMING METERS !

R2 Score for Training Data: 0.9999275908849684
R2 Score for Test Data: 0.9808550368018144

RESULT:

The program was executed and output was verified

of palantin livings and analysis is her to per in the part of

Charges and the State of the State of

able of the same

المهاول الأطالعي

and the second of the

#### IMPLEMENTATION OF DECISION TREE CLASSIFICATION TECHNIQUES

EX. NO:

DATE : 9/10/2024

AIM:

To implement a decision true classification technique for gender classification using python.

#### EXPLANATION:

\* Import tree from sklearin

\* call the function Decision Tree Classifier () from tree

\* Assign values for x and y.

\* call the function predict for predicting con the basis

of given random values for each given feature.

\* simplay the output.

#### Source Code:

import pandas as pd

from sklearn tree import Decision Tree classifica

data: { 'Height': [152, 172, 185, 167, 180]

'weight': [45,57,72,85,68]

'Gender': ['Female', 'Female', 'Hale', 'Male', 'Female'

of = pd. Data France (data)

X = of [['Hight'/ Weight']]

Y = df [ 'Gendar' ]

classifier = DecisionTree Classifier ()

classifice. fit (x, y)

hight = 159

weight = 51

prediction-data = pd DataFrame ([[height, weight]]. columns: ['Height', 'weight'])

predicted gender = classifier predict (prediction Print (f"Predicted gender: {predicted-gender[0]}. OUTPUT: Predicted gender: Female leasing to a social control RESULT: The program was executed and output was very . O clustivis x untroids Feature - 1

#### IMPLEMENTATION OF CLUSTURING TECHNIQUES -

K-MEANS EX. NO: DATE: 16/10/2024 10 implement a k-means clustwing technique using python Language. EXPLANATION: \* Import K-means from sklearn. cluster \* Assign X cy . Yoursmust signed . Alond this is \* (all the function kneoms () \* Perform scatter operation and display the output STRIBURY - STRIBER - 118. SOURCE CODE: P. Salandaria C.S. import numpy as np import mat plotlib. pyplot as plt from Sklewin duster import KMeans from Skleaven. datasets import make-blobs. it said the self-doops or not self-grapi X, y-true = make\_blobs (n. samples = 300. centus = 3, cluster-std random-state=0) Committee the character K=3 kmeans = KMeans (n. clusters = K, random\_state = 0) y-kmeans: kmeans. fit-predict(x) Plt. fique (figsize = (8,6)) plt. Scatter (x[:/6], X[::,1]; C= y-kmeans, S=30, cmap= viridie (georgia label = "clusters") Centers = Kmeans. Cluster\_centers\_ plt. scatter (centers [:,0], centers [:,1], C='red', S=200, alpha = 0.75, marker = 'x', label = 'Centroids') title ('KHeans Clustering' Results')

plt. xlabel ('Feature 1')

REQUET: Thus the program was plt. legent()

Plt. legent()

Plt. Man. ()

## AO\* ALGORITHM

AIM:

implement AO\* algorithm using python Programming language.

CODE :

class graph:

det -init-(self, graph, heuristic):

self. graph = graph

self. heuristic : hewristic

self. solution 2 23

det ao-star (suf, hode):

print (+ " Expanding; Enode 3")

if node not in self-graph or not self-graph [node]:

Hetwin

children: self. graph [node]

best-path = None

min-cost = ploat ("inf")

for group in children:

Cos = sum (self. hewistic [child] for chil

in group)

if cost < min-cost:

min-cost = cost best-path-group

self-solution mode J= best-path

print ( g" Best path for Enode 3: { best path 3 w

cost { nin-cost2)

#### for child in best-path: self. ao-stor(inild)

def get-solution (self): nerum self. solution

graph = { 'A: [['B';'c',['D']],

B': [['E']],

"c': [['q']], s are affect of co to the start was

'D': [['H']],

C.C. PJ, C. all all Elong All All Annual

'G':[],

'H': []

heweistic = { 'A': D, 'B': 1, 'C': 2, 'D': 4, 'E': 1, 'G': 3, 'H':53

graph-obj = Graph (graph fewistic)

graph\_obj. ao\_stayl(A')

Solution = graph-obj. get-solution()

print ("solution:", solution)

Thus the program to implement 40+ algorithm and executed and the certicold · 1234 11974

OUTPUT:

Expanding: A

Best Path for A: ['B', 'C'] with cost 3

Expanding : B

Best pour for B:[E] with cost 1

Esyanding: E

Expanding: C

Best path for C: ['G'] with cost 3. [ ]

Expanding: 9

Solution: {'A': [B', 'C', J, B': [E'], 'C': [a']}

dos chier en best, person

RESULT:

Thus the program to implement AO+ algorithm was executed and the outpet was

Toph-obje Coople (graph)

1) MOIS . GO . STON ("A

age ifdo xaego i mon

#### MINI MAX ALGORITHM

AIM:

To implement their more algorithm using python language.

CODE :

import math

det minimare (depth, node-index, is-marimizer, Scores, height):

if depth == height:

survin scores [nodes\_index]

if is-maximizer:

retwen man (minimax (depth +1, node. index \*2, False, Scores, height),

minimax (depth +1, node\_index \*2+1, Falx, Scores, height))

else:

retwin nuin (minimax (depth +1, node-index \*2, True, scores, height),

minimax (dupth +1, node-index \* 2 +1), True,
Scores, height)

det calculate-très height (num-leaves): restroy math. ceil (math. log2(num-leaves))

Sorres = [3, 5, 6, 9, 1, 2, 0, -1]

tree-height = calculate - tree-height (len (scores))

Optimal\_score = minimax (0, 0, Tour, Scores, tree-hieght)

print (f" the optimal Score is: { optimal- 8 core 3")

The optimal sore is: 3 to the prime metropic series in the series minimum ( depth, soots, widow it maximum from a tapina a Button a same Borns Barrel Lang return to the midelinity shifty for 1), notice because Fritz Brewi Hardelin William minimal Super LES made, index + 2 - 1, False green with levisionan Ediphent, node whener For Other March St. C. 1500 C. S. L. Levie M. RESULT: + second short 14 de 15 commissions Thus the minimax algorithm was executed successfully and the output was verified. [P. M. S. 1 ? 1 ? 2] SUN ((130026) mily Well said - 13 mily 102 - Well Core in white care (0. 0 19 EL., Scores the might)

E Restry

bronder ( mia)

2 - Leoman (mia)

9 - Parity

# INTRODUCTION TO PROLOG

AIM:

To leaven prolog terminologies and write basic programs.

TERMINOLOGIES:

1) Atomic Terms:

Atomic Term is usually string made up of dower of upper case letters, digits and the undercore, starting with a lowercase letter; starting with a lowercase letter; starting with a lowercase letter;

2, Variables:

vovuiables are string of letters, digits, and the underscore starting with capital letter of underscore.

ex: Dog Apple-420

3, Compound Terms:

PROLOG atom 8 a no. of arguments (PROLOG turns)
and enclosed in paranthesis and separated by commus.

Ex: is bigger (elephant, 21)

4) Facts:

Ex: bigger\_animal (whale)

& rule of a head (a predicate) and a bo 5, Rules: (a sequence) Px: is smaller (x, v): is bigger (Y, x) SOURCE CODE ! KB1 : woman (ma) "woman (jody) printe planer is most stancia woman (gotandia) playshir Guitar (jody) - Horas al a River print party O/P : 2 - woman (mia) Capture the Carthelic of Saint Water 7- playshir quitar (mia) yabse 600 Mapple 4120 ? - party true Compound trivial ores wall 7 - concert procedure concert doesn't exist. he brigger ( employed and

```
SOURCE CODE :
KB2:
  happy (yobrda)
                                     ( de l'about ) bons
  listen 2 music (mia)
                                         feed ( pizz a)
  listen 2 music (yolandra): - happy (yolandra)
  playstrix (witar (mia): - listen 2 music(mia)
  playstir quitar (yolandra):-listen 2 muric (yolandra),
                                    food (pizza)
  2 - play Air quitar (mia)
      true
   ? - plays Ais (without ( yolandra) ) some (x) low
                                       a Kandonikch
      true
    likes (dan, Sally)
    likes (sally, dan)
    likes (john, bithney)
    married (x, y) - likes (x, y), likes (7, x)
    friends (x, y) - likes (x, y); likes (y, x)
                                 outan (care ( brown))
op:
     7 - likes (dan, x)
      x-sally
     ? - married (dan, Sally)
        true
     ?- married ( john , brittney)
                                      general (chang)
        jalse
                         in fund (ar Condition)
```

```
KBH:
   food (burger)
   food (sandwich)
   food ( pizza)
   lunch ( sandwich) ( some ) i love and ) i love
   dinnoc (pizza) invas monal : (ova) rusting site
   meal (x) - food(x) med: ( orbinary) modified
O/P:
   ? - food (pizza)
                            play (Air qui rot (much)
   7- meal(x), leunch(x)
   x = sandwidth
   ? - dinner (sandwiich)
       Jalse
                                 and (don solly)
KB5 :
                             was (son thing)
  Owns (jack, Car(bmw))
  owns ( john, Con (Chevy)).
  owns (divia, car(civic))
                           SLEWIS - ( X X) Warring
  Owns (jane, car(chevy)
  ordan (car (bmys))
                               (x con ( pros x)
   Sedan (car(cipic))
   truck (carlanery))
                                   x sales
                         married (dain saley)
OIP:
   ? - owns (john, x)
   x-con (chury) qualtied ado do barreson -
  7 - own (john,-)
     true
  ? - owns (who, car (chery))
     who = john
```

? - owns (jane, x), sedan(x) 7 - owns (jone, x), truck (x) x = car (chavy) the stance o govern the programme male (Merin) ( (Letter) drawn L perun Of ( chais, pues) personally ( hales, pites) present ( 1 ( Musis, better) RESULT: Thus the program was executed and output, was Worified. of providing (x) i male (x) provide (x) Generalings (F. Y) it provide (Y) presently (N. X. A.

# PROLOY FAMILY TREE

AIM:

To dwelop a jamily tree program using PRI with all possible facts, rules and queries.

SOURCE CODE :

KNOWLEDGE BASE:

male (peter)

male (John)

male (avris)

male (Kevin)

female (betty)

female (jerry)

female (lisa)

female (helen)

parent of (chris, peter)

parent Of ( Chris, betty)

parent Of ( helen, peter)

parent Of ( helph, betty)

parent of ( Kevin, avois)

parent of ( kwin, lisa)

parent Of ( jeny, john)

parent Of ( jerry, helen)

RULES :

father (x, y): - male (y), powert of (x, y)

mother (x, y): - female (y), parent of (x, y)

grandfather (x, y):- male (y), povently (x, z), parently grandmother (x, y):- female (y), povently (x, z), powertly (x, z), powertly (x, z)

brother (X,Y):- male (Y), female (X,Z), female (Y,W),

Z== W

sister (X,Y):- female (Y), female (X,Z), female (Y,W),Z

== W

male (Y), parent Of (X,Y)

X= chris, Y= peter

female (Y), parent Of (X,Y)

X= chris, Y= betty

male (Y), parent Of (X,Z), parent Of (Z,Y)

X= kurin, Y= peter, Z= chris

female (Y), parent Of (X,Z), parent Of (X,Y)

X= Kurin, Y= betty, Z= chris

RESULT

The program was executed and output was verified.