PROGRAM-04

import numpy as np

import pandas as pd

# Loading Data from a CSV File

data = pd.DataFrame(data=pd.read\_csv('finds.csv'))

# Separating concept features from Target

concepts = np.array(data.iloc[:,0:-1])

# Isolating target into a separate DataFrame

target = np.array(data.iloc[:,-1])

def learn(concepts, target):

'''

learn() function implements the learning method of the Candidate elimination algorithm.

Arguments:

concepts - a data frame with all the features

target - a data frame with corresponding output values

# Initialise S0 with the first instance from concepts

# .copy() makes sure a new list is created instead of just pointing to the same memory location '''

specific\_h = concepts[0].copy()

# Initialises G0 using list comprehension

# Creates as many lists inside as there are arguments,

# that which later will be replaced with actual parameters

# G0 = [['?', '?', '?', '?', '?', '?'],

# ['?', '?', '?', '?', '?', '?'],

# ['?', '?', '?', '?', '?', '?'],

# ['?', '?', '?', '?', '?', '?'],

# ['?', '?', '?', '?', '?', '?'],

# ['?', '?', '?', '?', '?', '?']]

general\_h = [["?" for i in range(len(specific\_h))] for i in range(len(specific\_h))]

# The learning iterations

for i, h in enumerate(concepts):

# Checking if the hypothesis has a positive target

if target[i] == "Yes":

for x in range(len(specific\_h)):

# Change values in S & G only if values change

if h[x] != specific\_h[x]:

specific\_h[x] = '?'

general\_h[x][x] = '?'

# Checking if the hypothesis has a positive target

if target[i] == "No":

for x in range(len(specific\_h)):

# For negative hyposthesis change values only in G

if h[x] != specific\_h[x]:

general\_h[x][x] = specific\_h[x]

else:

general\_h[x][x] = '?'

# find indices where we have empty rows, meaning those that are unchanged

indices = [i for i,val in enumerate(general\_h) if val == ['?', '?', '?', '?', '?', '?']]

for i in indices:

# remove those rows from general\_h

general\_h.remove(['?', '?', '?', '?', '?', '?'])

# Return final values

return specific\_h, general\_h

s\_final, g\_final = learn(concepts, target)

print("Final S:", s\_final, sep="\n")

print("Final G:", g\_final, sep="\n")

data.head()