Experiment-1

Objective: Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file

Code:

import numpy as np

# Define a 2D array with 4 training examples of weather data

a = np.array([['Sunny','Warm','Normal','Strong','Warm','Same','Yes'],

['Sunny','Warm','High','Strong','Warm','Same','Yes'],

['Rainy','Cold','High','Strong','Warm','Change','No'],

['Sunny','Warm','High','Strong','Cool','Change','Yes']])

# Calculate the number of attributes in each training example

num\_attributes = len(a[0])-1

# Initialize the hypothesis with '0' values equal to the number of attributes

hypothesis = ['0']\*num\_attributes   #['0','0','0','0','0','0']

# Print the initial hypothesis

print(hypothesis)

# Set the initial value of the hypothesis to be equal to the first training example

for j in range(0,num\_attributes):

    hypothesis[j] = a[0][j];

# Print the initial value of the hypothesis

print("The initial value of hypothesis:",hypothesis)

# Iterate over all the training examples

for i in range(0,len(a)):

    # If the target attribute of the training example is 'Yes'

    if a[i][num\_attributes] == 'Yes':

        # Update the hypothesis by checking each attribute value

        for j in range(0,num\_attributes):

            # If the attribute value is not equal to the hypothesis value, set the hypothesis value to '?'

            if a[i][j]!= hypothesis[j]:

                hypothesis[j] = '?'

            # If the attribute value is equal to the hypothesis value, set the hypothesis value to the attribute value

            else:

                hypothesis[j] = a[i][j]

    # Print the hypothesis for each training example

    print("For training example no: {0} the hypothesis is".format(i),hypothesis)

# Print the final maximally specific hypothesis

print("The maximally specific hypothesis for a given training example:",hypothesis)

Result:

