USN : 1KS17CS030

NAME : INDRASENA KALYANAM

SEM : 7 A

SUB : MACHINE LEARNING LAB.

SUB CODE: 17CSL76

14/09/20

EXPERIMENT No. 1

1KS17CS030

AIM: 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.

Task: Find-S algorithm is used to find a maximally specific hypothesis.

Dataset: Enjoy Sports Training Examples,

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	NO
4	Sunny	Warm	High	Strong	Cool	Change	Yes

ALGORITHM:

Step 1: Initialize h to the most specific hypothesis in H.

Step 2: For each positive training instance x,

- For each attribute constraint a_i in h.
  - If the constraint a_i is satisfied x Then do nothing
  - Else replace a_i in h by the next more general constraint that is satisfied by x.

Step 3: Output the hypothesis h.

PROGRAM:

```
import numpy as np
import pandas as pd
data = pd.read_csv('FindS.csv')
#Print(data)
concepts = data.iloc[:,0:-1].values
print("-----------------------")
print(concepts)
print("-----------------------")
target = data.iloc[:,-1].values
print(target)
def train(concepts, target):
    count = 0
    specific_h = concepts[0]
    for i,h in enumerate(concepts):
        print(i,h)
        if target[i] == "Yes"
            for x in range(len(specific_h)):
                if h[x] == specific_h[x]:
                    pass.
                else:
                    specific_h[x] = "?"
        Count+=1
        printf("Hypothesis after after sample number:
        {count} processed: {specific_h}")
```

```
        else:
            pass
            count += 1
            print("Negative sample number: {count}
            Same Hypothesis: {specific_h}")

    return specific_h

specific_h = train(concepts, target)
```

Input : finds.csv

Output :

```
      Sky AirTemp Humidity    Wind Water Forecast EnjoySport
0   Sunny    Warm   Normal  Strong  Warm     Same        Yes
1   Sunny    Warm     High  Strong  Warm     Same        Yes
2  Cloudy    Cold     High  Strong  Warm   Change         No
3   Sunny    Warm     High  Strong  Cool   Change        Yes
----------------------------------------
[['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same']
 ['Sunny' 'Warm' 'High' 'Strong' 'Warm' 'Same']
 ['Cloudy' 'Cold' 'High' 'Strong' 'Warm' 'Change']
 ['Sunny' 'Warm' 'High' 'Strong' 'Cool' 'Change']]
----------------------------------------
['Yes' 'Yes' 'No' 'Yes']
0 ['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same']
Hypothesis after sample number:1 processed: ['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same']
1 ['Sunny' 'Warm' 'High' 'Strong' 'Warm' 'Same']
Hypothesis after sample number:2 processed: ['Sunny' 'Warm' '?' 'Strong' 'Warm' 'Same']
2 ['Cloudy' 'Cold' 'High' 'Strong' 'Warm' 'Change']
Negative sample number:3 Same Hypothesis: ['Sunny' 'Warm' '?' 'Strong' 'Warm' 'Same']
3 ['Sunny' 'Warm' 'High' 'Strong' 'Cool' 'Change']
Hypothesis after sample number:4 processed: ['Sunny' 'Warm' '?' 'Strong' '?' '?']
```