Date. 29/9/20.....

Expt. No.....1..... Page No.....1.....

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data sample. Read the training data from csv file.

```
import random
import csv
attributes = [['sunny','rainy'],['warm','cold'],['normal',
    'high'],['strong','weak'],['warm','cool'],['same',
        'change']]
print(attributes)
num_attributes = len(attributes)
print(num_attributes)
print("\n the most general hypothesis : ['?','?','?',
        '?','?','?']\n")
print("\n the most specific hypothesis : ['0','0','0',
        '0','0','0']\n")
a=[]
print("\n the given training dataset\n")
with open('C:\Users\Admin\Desktop\ML\lab1\data.csv',
        'r') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        a.append(row)
        print(row)
print("\n the initial value of hypothesis:")
h = ['0']*num_attributes
print(h)
for j in range(0,num_attributes):
```

Teacher's Signature : .....

Date.....

Expt. No.....1..... Page No.....2.....

```
    b[j] = a[i][j]
for i in range(1, len(a)):
    if (a[i][num_attributes] == 'Yes'):
        for j in range(num_attributes):
            if(h[j] == '0' or h[j] == a[i][j]):
                h[j] = a[i][j]
            else:
                h[j] = '?'
print("\n for training examples : {0} the
    hypothesis") format(i+1), h)
```

Teacher's Signature : .....

Date.....

Expt. No..... Page No.....

Dataset:

Sl. No	sky	AirTemp	Humidity	wind	water	forecast	enjoy sport
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

output:-

[['Sunny','Rainy'],['warm','cold'],['normal','high'],['strong','weak'],['warm','cool'],['same','change']]

b.

the most general hypothesis: ['?','?','?','?','?','?']

the most specific hypothesis: ['0','0','0','0','0','0']

The given training dataset

['Sky','AirTemp','Humidity','wind','water','forecast','EnjoySport']

['Sunny','warm','normal','strong','warm','same','Yes']

['Sunny','warm','high','strong','warm','same','Yes']

['Rainy','cold','High','strong','warm','change','Yes']

The initial hypothesis is:

['0','0','0','0','0','0']

Date.....

Expt. No.....

Page No.....

for training example : {0} the hypothesis =

['sunny','warm','?','strong','?','?']

Date.06.10.2020....

Expt. No.....2.....

Page No.....3.....

2. for a given set of set of training data example stored in a .csv file, implement and demonstrate the candidate-elimination algorithm to output a description of the set of all hypothesis consistent with the training example.

```
import csv
with open('c:\\Users\\Admin\\Desktop\\MLBDA\\all.csv')
as f:
csv_file = csv.reader(f)
data = list(csv_file)
print(data)
s= data[1][:-1]
g= [['?' for i in range(len(s))] for j in range(
    len(s))]
for i in data:
    if i[-1]=="Yes":
        for j in range(len(s)):
            if i[j]!= s[j]:
                s[j] = '?'
                g[j][j]='?'
    elif i[-1]=="No":
        for j in range(len(s)):
            if i[j]!= s[j]:
                g[j][j] = s[j]
            else:
                g[j][j] = "?"
print("\n steps of candidate elimination algorithm", data.index(i)+1)
```

Teacher's Signature : .....

Date.....

Expt. No.....2.....

Page No.....4.....

```
Print(s)
print(g)
gh=[]
for i in g:
    for j in i:
        if 'j'!='?':
            gb.append(i)
            break
printf("\n Final specific hypothesis:\n",s)
print("\n Final general hypothesis:\n",gh)
```

Teacher's Signature : .....

Date.....

Expt. No..... Page No.....

Dataset:

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	warm	change	Yes

Output:

steps of candidate elimination algorithm 1

['Sunny', 'warm', '?', 'strong', '?', '?']

[['sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?']]

steps of candidate elimination algorithm 2.

['sunny', 'warm', '?', 'strong', '?', '?']

[['sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?']]

steps of candidate elimination algorithm 3.

['Sunny', 'warm', '?', 'strong', '?', '?'].

[['sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?']]

Date.....

Expt. No.....

Page No.....

steps of candidate elimination algorithm :

['sunny', 'warm', '?', 'strong', '?', '?']

[['sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?', '?'],

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

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

final specific hypothesis

['sunny', 'warm', '?', 'strong', '?', '?']

final general hypothesis:

[['sunny', '?', '?', '?', '?', '?'], ['?', 'warm', '?', '?', '?', '?']]