Practical No: 2

Concept Learning

AIM: Implement and demonstrate the find-s algorithm for finding the most specific.

Description:

1. Training dataset table (input data):

	A	B	C	D	E	F	G
1	sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
2	Sunny	Warm	Normal	Strong	Warm	Same	Yes
3	Sunny	Warm	High	Strong	Warm	Same	Yes
4	Rainy	Cold	High	Strong	Warm	Change	No
5	Sunny	Warm	High	Strong	Cool	Change	Yes
6							

2.: Write the right hypothesis/function from historical data

One of the often-used statistical concepts in machine learning is the hypothesis. It is notably employed in supervised machine learning, where an ML model uses a dataset to train a function that most effectively translates input to related outputs.

In this code person enjoys sport if weather is sunny, airtemp is warm, wind is strong

3. How Does It Work?

It eliminates attribute that do not affect target column

Code with output

```
import csv
num_attributes = 6
a = []

print("\n The Given Training Dataset \n")
with open('Book1.csv','r') as csvfile:
  reader = csv.reader(csvfile)
  count = 0
  for row in reader:
    if count == 0:
      print(row)
      count+=1;
    else:
      a.append(row)
      print(row)
```

```
    count+=1

print("\n The initial value of hypothesis: ")
hypothesis = ['0'] * num_attributes
print(hypothesis)

for j in range(0,num_attributes):
  hypothesis[j]= a[0][j];
  print(hypothesis)

print("\n find S:finding a Maximally specific Hypothesis\n")
for i in range(0,len(a)):
  if a[i][num_attributes]=="Yes":
    for j in range(0,num_attributes):
      if a[i][j]!=hypothesis[j]:
        hypothesis[j]='?'
      else:
        hypothesis[j] = a[i][j]
  print("for training example no :{0} the hypothesis is".format(i),hypothesis)
```

```
 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', 'No']
['Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change', 'Yes']
```

```
 The initial value of hypothesis:
['0', '0', '0', '0', '0', '0']
```

```
['Sunny', '0', '0', '0', '0', '0']
['Sunny', 'Warm', '0', '0', '0', '0']
['Sunny', 'Warm', 'Normal', '0', '0', '0']
['Sunny', 'Warm', 'Normal', 'Strong', '0', '0']
['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', '0']
['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same']
```

```
 find S:finding a Maximally specific Hypothesis

for training example no :0 the hypothesis is ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same']
for training example no :1 the hypothesis is ['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']
for training example no :2 the hypothesis is ['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']
for training example no :3 the hypothesis is ['Sunny', 'Warm', '?', 'Strong', '?', '?']
```

```
['Sunny', 'Warm', '?', 'Strong', '?', '?']
```

```
import csv
a = []
with open('book2.csv', 'r') as csvfile:
    next(csvfile)
    for row in csv.reader(csvfile):
        a.append(row)

for x in a:
  print(x)

print("\nThe total number of training instances are : ",len(a))
num_attribute = len(a[0])-1
print("\nThe initial hypothesis is : ")
hypothesis = ['0']*num_attribute
print(hypothesis)

for i in range(0, len(a)):
    if a[i][num_attribute] == 'yes':
        print ("\nInstance ", i+1, "is", a[i], " and is Positive Instance")
        for j in range(0, num_attribute):
            if hypothesis[j] == '0' or hypothesis[j] == a[i][j]:
                hypothesis[j] = a[i][j]
            else:
                hypothesis[j] = '?'
        print("The hypothesis for the training instance", i+1, " is: " , hypothesis, "\n")
    if a[i][num_attribute] == 'no':
        print ("\nInstance ", i+1, "is", a[i], " and is Negative Instance Hence Ignored")
        print("The hypothesis for the training instance", i+1, " is: " , hypothesis, "\n")
print("\nThe Maximally specific hypothesis for the training instance is ", hypothesis)
```

```
['some', 'small', 'no', 'affordable', 'many', 'no']
['many', 'big', 'no', 'expensive', 'one', 'yes']
['some', 'big', 'always', 'expensive', 'few', 'no']
['many', 'medium', 'no', 'expensive', 'many', 'yes']
['many', 'small', 'no', 'affordable', 'many', 'yes']
```

```
The total number of training instances are :  5
```

```
The initial hypothesis is :
['0', '0', '0', '0', '0']
```

```
Instance  1 is ['some', 'small', 'no', 'affordable', 'many', 'no']  and is Negative Instance Hence Ignored
The hypothesis for the training instance 1  is:  ['many', '?', 'no', '?', '?']


Instance  2 is ['many', 'big', 'no', 'expensive', 'one', 'yes']  and is Positive Instance
The hypothesis for the training instance 2  is:  ['many', '?', 'no', '?', '?']


Instance  3 is ['some', 'big', 'always', 'expensive', 'few', 'no']  and is Negative Instance Hence Ignored
The hypothesis for the training instance 3  is:  ['many', '?', 'no', '?', '?']


Instance  4 is ['many', 'medium', 'no', 'expensive', 'many', 'yes']  and is Positive Instance
The hypothesis for the training instance 4  is:  ['many', '?', 'no', '?', '?']


Instance  5 is ['many', 'small', 'no', 'affordable', 'many', 'yes']  and is Positive Instance
The hypothesis for the training instance 5  is:  ['many', '?', 'no', '?', '?']
```

```
The Maximally specific hypothesis for the training instance is  ['many', '?', 'no', '?', '?']
```

Learnings

This Python code reads data from a CSV file and uses the Find-S algorithm for binary classification. It iterates through training instances, adjusting a hypothesis to correctly classify positive cases while minimizing errors. When negative cases are encountered, conflicting attributes are marked with '?' to ensure accuracy. The resulting 'hypothesis' is the most specific rule for the given training data.