Experiment -1

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 or .xlsx file.

Hypothesis – An assumption/ an idea/ an explanation/ statement of expectation or prediction that will be tested by research.

FindS algorithm:-

In order to understand Find-S algorithm, you need to have a basic idea of the following concepts as well:

1. Concept Learning
2. General Hypothesis
3. Specific Hypothesis

1. Concept Learning

Let's try to understand concept learning with a real-life example. Most of human learning is based on past instances or experiences. For example, we are able to identify any type of vehicle based on a certain set of features like make, model, etc., that are defined over a large set of features.

These special features differentiate the set of cars, trucks, etc from the larger set of vehicles. These features that define the set of cars, trucks, etc are known as concepts.

Similar to this, machines can also learn from concepts to identify whether an object belongs to a specific category or not. Any algorithm that supports concept learning requires the following:

- Training Data
- Target Concept
- Actual Data Objects

2. General Hypothesis

Hypothesis, in general, is an explanation for something. The general hypothesis basically states the general relationship between the major variables. For example, a general hypothesis for ordering food would be *I want a burger.*

G = { '?', '?', '?', …..'?'}

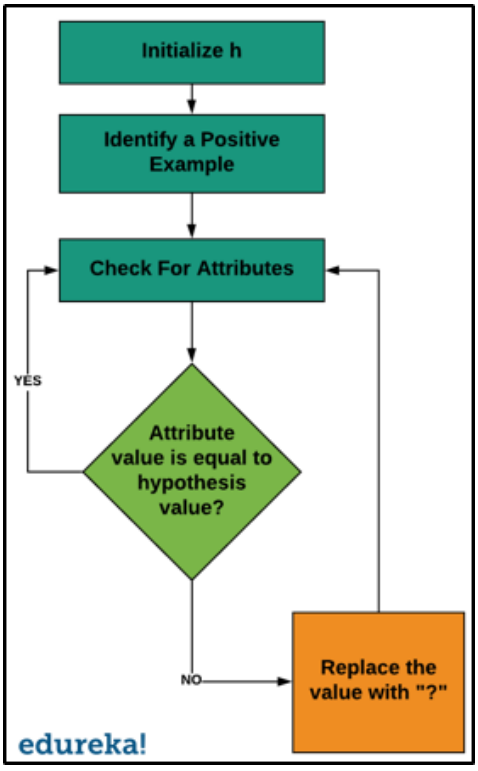
3. Specific Hypothesis

The specific hypothesis fills in all the important details about the variables given in the general hypothesis. The more specific details into the example given above would be *I want a cheeseburger with a chicken pepperoni filling with a lot of lettuce.*

S = {'Φ','Φ','Φ', ……,'Φ'}

The Find-S algorithm follows the steps written below:

1. Initialize ‘h’ to the most specific hypothesis.
2. The Find-S algorithm only considers the positive examples and eliminates negative examples. For each positive example, the algorithm checks for each attribute in the example. If the attribute value is the same as the hypothesis value, the algorithm moves on without any changes. But if the attribute value is different than the hypothesis value, the algorithm changes it to ‘?’.



1. The process starts with initializing ‘h’ with the most specific hypothesis, generally, it is the first positive example in the data set.
2. We check for each positive example. If the example is negative, we will move on to the next example but if it is a positive example we will consider it for the next step.
3. We will check if each attribute in the example is equal to the hypothesis value.
4. If the value matches, then no changes are made.
5. If the value does not match, the value is changed to ‘?’.
6. We do this until we reach the last positive example in the data set.

Implementation of Find-S Algorithm

To understand the implementation, let us try to implement it to a smaller data set with a bunch of examples to decide if a person wants to go for a walk.

The concept of this particular problem will be on what days does a person likes to go on walk.

Time	Weather	Temperature	Company	Humidity	Wind	Goes
Morning	Sunny	Warm	Yes	Mild	Strong	Yes
Evening	Rainy	Cold	No	Mild	Normal	No
Morning	Sunny	Moderate	Yes	Normal	Normal	Yes
Evening	Sunny	Cold	Yes	High	Strong	Yes

Looking at the data set, we have six attributes and a final attribute that defines the positive or negative example. In this case, yes is a positive example, which means the person will go for a walk.

So now, the general hypothesis is:

h_0 = {‘Morning’, ‘Sunny’, ‘Warm’, ‘Yes’, ‘Mild’, ‘Strong’}

This is our general hypothesis, and now we will consider each example one by one, but only the positive examples.

h_1= {‘Morning’, ‘Sunny’, ‘?’, ‘Yes’, ‘?’, ‘?’}

h_2 = {‘?’, ‘Sunny’, ‘?’, ‘Yes’, ‘?’, ‘?’}

Code:

```
import pandas as pd
import numpy as np

#to read the data in the csv file
data = pd.read_csv("data.csv")
print(data,"n")

#making an array of all the attributes
d = np.array(data)[:,:-1]
print("n The attributes are: ",d)

#segragating the target that has positive and negative examples
target = np.array(data)[:,-1]
print("n The target is: ",target)

#training function to implement find-s algorithm
def train(c,t):
    for i, val in enumerate(t):
        if val == "Yes":
            specific_hypothesis = c[i].copy()
            break

    for i, val in enumerate(c):
```

```
    if t[i] == "Yes":
        for x in range(len(specific_hypothesis)):
            if val[x] != specific_hypothesis[x]:
                specific_hypothesis[x] = '?'
            else:
                pass

    return specific_hypothesis

#obtaining the final hypothesis
print("n The final hypothesis is:",train(d,target))
```

Output:

```
      Time Weather Temperature Company Humidity    Wind Goes
0  Morning   Sunny        Warm     Yes     Mild  Strong  Yes
1  Evening   Rainy        Cold      No     Mild  Normal   No
2  Morning   Sunny    Moderate     Yes   Normal  Normal  Yes
3  Evening   Sunny        Cold     Yes     High  Strong  Yes


 The attributes are:  [['Morning' 'Sunny' 'Warm' 'Yes' 'Mild' 'Strong']
 ['Evening' 'Rainy' 'Cold' 'No' 'Mild' 'Normal']
 ['Morning' 'Sunny' 'Moderate' 'Yes' 'Normal' 'Normal']
 ['Evening' 'Sunny' 'Cold' 'Yes' 'High' 'Strong']]

 The target is:  ['Yes' 'No' 'Yes' 'Yes']

 The final hypothesis is: ['?' 'Sunny' '?' 'Yes' '?' '?']
```