Implement the **Find-S algorithm** to:

* Learn the most specific hypothesis (H) from a given set of training examples.
* Read data from a **CSV file**.

**Training Dataset**

* Each row is a training instance with attributes like:

| **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 |

**Find-S Algorithm Logic**

1. **Initialize** the hypothesis h as the most specific (e.g., [0, 0, 0, 0, 0, 0]).
2. **Update h** for every positive example (EnjoySport = Yes):
   * If a value matches the hypothesis or it's uninitialized (0), keep it.
   * If it differs, generalize that feature to '?'.

**Python Code Breakdown**

import csv

a = []

# Reading CSV file

with open('enjoysport.csv', 'r') as csvfile:

for row in csv.reader(csvfile):

a.append(row)

# Display the data

print(a)

# Initialize

print("\n The total number of training instances are :", len(a))

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

print("\n The initial hypothesis is : ")

hypothesis = ['0'] \* num\_attribute

print(hypothesis)

# Apply Find-S

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

if a[i][num\_attribute] == 'yes': # Only positive examples

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("\n The hypothesis for the training instance {} is : \n".format(i+1), hypothesis)

# Final result

print("\n The Maximally specific hypothesis for the training instance is ")

print(hypothesis)

**Output Flow Summary**

1. Starts with hypothesis:  
   ['0', '0', '0', '0', '0', '0']
2. After instance 1 (positive):  
   ['sunny', 'warm', 'normal', 'strong', 'warm', 'same']
3. After instance 2 (positive):  
   Mismatch at humidity → becomes '?':  
   ['sunny', 'warm', '?', 'strong', 'warm', 'same']
4. Instance 3 is negative → **ignored**
5. After instance 4 (positive):  
   Mismatches at water and forecast → become '?':  
   ['sunny', 'warm', '?', 'strong', '?', '?']

**Final Hypothesis:**

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