| **CO1** | **Examine the basic concepts of data mining and machine learning concepts** |
| --- | --- |
| **Task 2:** | Demonstrate the FIND-S algorithm to find a most specific hypothesis and Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.  **Platform: Orange, Google co-lab, Anaconda navigator, Language: Python** |

**FIND-S Algorithm:**

**Objective:**

Given a dataset of weather conditions with binary attributes (Sunny/Rainy, Warm/Cold, Normal/High, Strong/Weak, Warm/Cool, Same/Change), the objective is to find the most specific hypothesis based on positive examples.

**Algorithm:**

1.- Import necessary libraries (`matplotlib.pyplot` and `drive` from `google.colab`).

- Mount Google Drive using `drive.mount('/content/drive')`.

2.- Define a function `visualize\_find\_s\_algorithm` that takes positive examples as input.

- Initialize the hypothesis with '0' for each attribute.

- Iterate through each positive example and update the hypothesis based on the FIND-S   
 algorithm.

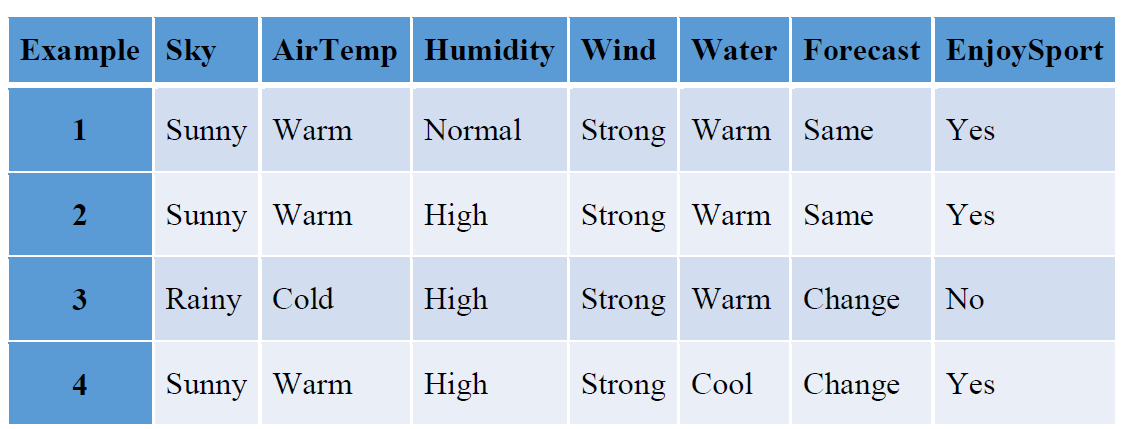
- Keep track of the visualization at each step.

3. - Define positive examples for the FIND-S algorithm.

4. - Apply the FIND-S algorithm to positive examples.

- Plot the hypothesis visualization using Matplotlib.

**Training Examples:**



**Program:**

import matplotlib.pyplot as plt

from google.colab import drive

drive.mount('/content/drive')

def visualize\_find\_s\_algorithm(positive\_examples):

hypothesis = ['0'] \* len(positive\_examples[0])

visualization = []

for idx, example in enumerate(positive\_examples, start=1):

for i in range(len(example)):

if hypothesis[i] == '0':

hypothesis[i] = example[i]

elif hypothesis[i] != example[i]:

hypothesis[i] = '?'

visualization.append((idx, " ".join(hypothesis)))

return visualization

positive\_examples\_find\_s = [

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

['Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same'],

['Rainy', 'Cold', 'High', 'Strong', 'Cool', 'Change'],

['Sunny', 'Hot', 'High', 'Strong', 'Cool', 'Change']

]

hypothesis\_find\_s\_visualization = visualize\_find\_s\_algorithm(positive\_examples\_find\_s)

plt.figure(figsize=(12, 6))

plt.plot([point[0] for point in hypothesis\_find\_s\_visualization], [point[1] for point in hypothesis\_find\_s\_visualization], marker='o')

plt.title('FIND-S Algorithm Visualization')

plt.xlabel('Example Index')

plt.ylabel('Hypothesis')

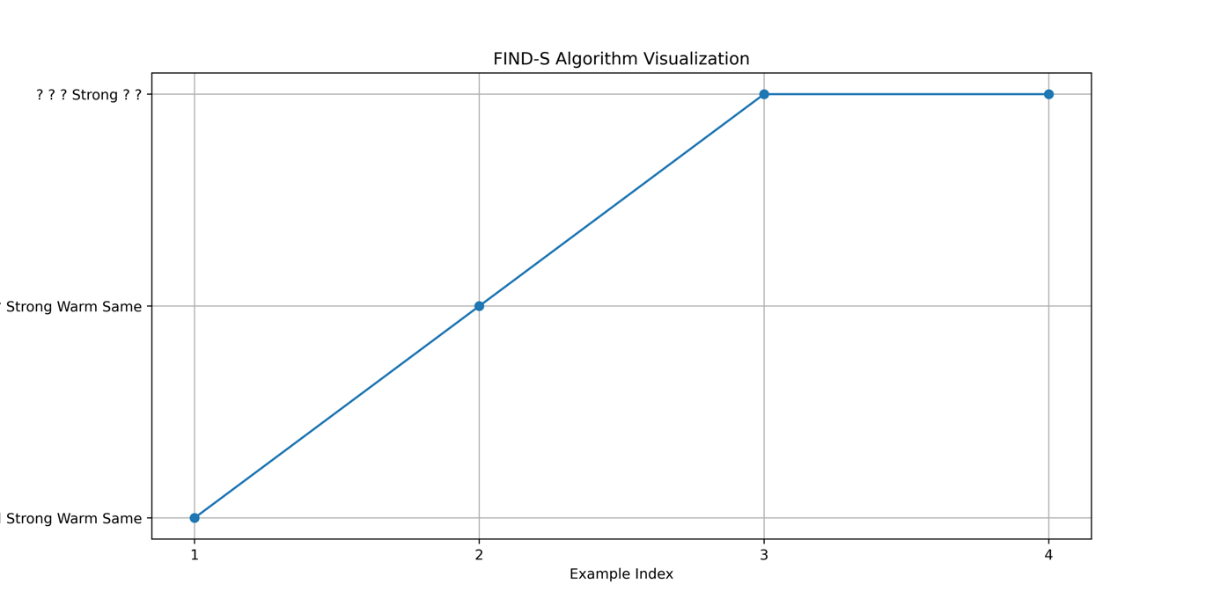
plt.xticks(range(1, len(positive\_examples\_find\_s) + 1))

plt.grid(True)

plt.savefig('/content/drive/MyDrive/ML Lab/ExNo02/FIND-S Algorithm Visualization.png', dpi=500)

plt.show()

Output:



**Candidate-Elimination Algorithm:**

**Scenario:**

Similar to the FIND-S scenario, but now we also have negative examples. The goal is to maintain a version space that includes all hypotheses consistent with positive examples and eliminates inconsistent hypotheses based on negative examples.

**Algorithm:**

1. Initialization:

- Initialize the specific hypothesis with '0's.

- Initialize the general hypothesis with '?'s.

2. Positive Examples Loop:

- For each positive example:

- Update the specific hypothesis:

- If an attribute in the specific hypothesis is '0', update it with the corresponding attribute   
 from the example.

- If an attribute is different, update it to '?' indicating variability.

- Update the general hypothesis:

- If an attribute in the example is different from the corresponding attribute in the specific   
 hypothesis, update the general hypothesis to match the specific hypothesis.

3. - For each negative example:

- Update the general hypothesis:

- If an attribute in the example is different from the corresponding attribute in the specific   
 hypothesis, update the general hypothesis to '?' indicating variability.

4. - Use matplotlib to visualize the evolution of the specific and general hypotheses.

- Plot the specific hypothesis as circles ('o') and the general hypothesis as 'x' marks with a   
 dashed line.

- Save the plot to a file and display it.

**Program:**

import matplotlib.pyplot as plt

def visualize\_candidate\_elimination\_algorithm(positive\_examples, negative\_examples):

specific\_hypothesis = ['0'] \* len(positive\_examples[0]) general\_hypothesis = ['?'] \* len(positive\_examples[0])

visualization\_specific = []

visualization\_general = []

for idx, example in enumerate(positive\_examples, start=1):

for i in range(len(example)):

if specific\_hypothesis[i] == '0':

specific\_hypothesis[i] = example[i]

elif specific\_hypothesis[i] != example[i]:

specific\_hypothesis[i] = '?'

for i in range(len(example)):

if example[i] != specific\_hypothesis[i]:

general\_hypothesis[i] = specific\_hypothesis[i]

visualization\_specific.append((idx, " ".join(specific\_hypothesis)))

visualization\_general.append((idx, " ".join(general\_hypothesis)))

for example in negative\_examples:

for i in range(len(example)):

if example[i] != specific\_hypothesis[i]:

general\_hypothesis[i] = '?' visualization\_general.append((idx + 1, " ".join(general\_hypothesis)))

return visualization\_specific, visualization\_general

positive\_examples\_candidate = [

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

['Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same'],

['Rainy', 'Cold', 'Normal', 'Strong', 'Cool', 'Change'],

['Sunny', 'Hot', 'High', 'Strong', 'Cool', 'Change']

]

negative\_examples\_candidate = [

['Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change'],

['Sunny', 'Warm', 'Normal', 'Weak', 'Cool', 'Same']

]

specific, general = visualize\_candidate\_elimination\_algorithm(positive\_examples\_candidate, negative\_examples\_candidate)

plt.figure(figsize=(12, 6))

plt.plot([point[0] for point in specific], [point[1] for point in specific], marker='o', label='Specific Hypothesis')

plt.plot([point[0] for point in general], [point[1] for point in general], marker='x', linestyle='--', label='General Hypothesis')

plt.title('Candidate-Elimination Algorithm Visualization')

plt.xlabel('Example Index')

plt.ylabel('Hypothesis')

plt.legend()

plt.xticks(range(1, len(positive\_examples\_candidate) + 2))

plt.grid(True)

plt.savefig('/content/drive/MyDrive/ML Lab/ExNo02/Candidate-Elimination Algorithm Visualization.png', dpi=500)

plt.show()

**Output:**

