

Program 3

For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples

- Candidate Elimination is a supervised technique for learning concepts for data.
- This algorithm incrementally builds the version space given a hypothesis space H and a collection of E instances.
- The examples are introduced one by one, with each example shrinking the version space by removing the hypotheses that are inconsistent with the example.
- The candidate performs this by updating the general and specific boundary for each example.

Algorithm

- 1) Load the data set
- 2) Initialize the general and specific hypothesis
- 3) For each training example
 - if example is positive example
 - if attribute value == hypothesis value:
 - Do nothing
 - else:
 - Replace attribute value with '?'
 - if example is negative example
 - Make general hypothesis more specific

Program

```
import csv
with open('trainingexamples.csv') as f:
    csv_file = csv.reader(f)
    data = list(csv_file)
    specific = data[0][:-1]
    general = [['?' for i in range(len(specific))] for j in range(len(specific))]
    for i in data:
        if i[-1] == "Yes":
            for j in range(len(specific)):
                if i[j] != specific[j]:
                    specific[j] = "?"
                    general[j][j] = "?"

        elif i[-1] == "No":
            for j in range(len(specific)):
                if i[j] != specific[j]:
```

```

        general[j][i] = specific[i]
    else:
        general[j][i] = "?"

    print("Step " + str(data.index(i)+1) + " of Candidate Elimination Algorithm")
    print(specific)
    print(general)
    print()

    gh = []
    for i in general:
        for j in i:
            if j != "?":
                gh.append(i)
                break

    print("\nFinal Specific hypothesis:\n", specific)
    print("\nFinal General hypothesis:\n", gh)

```

Result

Step 1 of Candidate Elimination Algorithm

```

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

```

Step 2 of Candidate Elimination Algorithm

```

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

```

Step 3 of Candidate Elimination Algorithm

```

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

```

Step 4 of Candidate Elimination Algorithm

```

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

```

Final Specific hypothesis:

```

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

```

Final General hypothesis:

```

[['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?']]

```