VIT UNIVERSITY, ANDHRA PRADESH
School of CSE
CSE3008 - Introduction to Machine Learning
Lab Experiment-2
(Candidate - Elimination Algorithm)

Faculty-**Dr. B. SRINIVASA RAO**

Name-Neeraj Guntuku
R.No-18MIS7071
Slot-L55+L56

Date-13 February 2021

MACHINE LEARNING LAB - 2 (Candidate - Elimination Algorithm)

[1]
```
import numpy as np
import pandas as pd
```

[2]
```
# Loading Data from a CSV File
data = pd.read_csv('EnjoySport.csv', header=None)
print(data)
```

```
        0     1       2       3     4       5    6
0   Sunny  Warm  Normal  Strong  Warm    Same  Yes
1   Sunny  Warm    High  Strong  Warm    Same  Yes
2   Rainy  Cold    High  Strong  Warm  Change   No
3   Sunny  Warm    High  Strong  Cool  Change  Yes
```

[3]
```
# Separating concept features from Target
concepts = np.array(data.iloc[:,0:-1])
print(concepts)
```

```
[['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same']
 ['Sunny' 'Warm' 'High' 'Strong' 'Warm' 'Same']
 ['Rainy' 'Cold' 'High' 'Strong' 'Warm' 'Change']
 ['Sunny' 'Warm' 'High' 'Strong' 'Cool' 'Change']]
```

[4]
```
# Isolating target into a separate DataFrame
# copying last column to target array
target = np.array(data.iloc[:,-1])
print(target)
```

```
['Yes' 'Yes' 'No' 'Yes']
```

```
[5] def learn(concepts, target):

        '''
        learn() function implements the learning method of the Candidate elimination algorithm.
        Arguments:
            concepts - a data frame with all the features
            target - a data frame with corresponding output values
        '''

        # Initialise S0 with the first instance from concepts
        # .copy() makes sure a new list is created instead of just pointing to the same memory location
        specific_h = concepts[0].copy()
        print("\nInitialization of specific_h and general_h")
        print(specific_h)
        #h=["#" for i in range(0,5)]
        #print(h)

        general_h = [["?" for i in range(len(specific_h))] for i in range(len(specific_h))]
        print(general_h)
        # The learning iterations
        for i, h in enumerate(concepts):

            # Checking if the hypothesis has a positive target
            if target[i] == "Yes":
                for x in range(len(specific_h)):

                    # Change values in S & G only if values change
                    if h[x] != specific_h[x]:
                        specific_h[x] = '?'
                        general_h[x][x] = '?'
```

```
            # Checking if the hypothesis has a positive target
            if target[i] == "No":
                for x in range(len(specific_h)):
                    # For negative hyposthesis change values only  in G
                    if h[x] != specific_h[x]:
                        general_h[x][x] = specific_h[x]
                    else:
                        general_h[x][x] = '?'

            print("\nSteps of Candidate Elimination Algorithm",i+1)
            print(specific_h)
            print(general_h)

        # find indices where we have empty rows, meaning those that are unchanged
        indices = [i for i, val in enumerate(general_h) if val == ['?', '?', '?', '?', '?', '?']]
        for i in indices:
            # remove those rows from general_h
            general_h.remove(['?', '?', '?', '?', '?', '?'])
        # Return final values
        return specific_h, general_h
```

```
[6] s_final, g_final = learn(concepts, target)
    print("\nFinal Specific_h:", s_final, sep="\n")
    print("\nFinal General_h:", g_final, sep="\n")
```

```
Initialization of specific_h and general_h
['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same']
[['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'

Steps of Candidate Elimination Algorithm 1
['Sunny' 'Warm' 'Normal' 'Strong' 'Warm' 'Same']
[['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'

Steps of Candidate Elimination Algorithm 2
['Sunny' 'Warm' '?' 'Strong' 'Warm' 'Same']
[['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'

Steps of Candidate Elimination Algorithm 3
['Sunny' 'Warm' '?' 'Strong' 'Warm' 'Same']
[['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '

Steps of Candidate Elimination Algorithm 4
['Sunny' 'Warm' '?' 'Strong' '?' '?']
[['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '

Final Specific_h:
['Sunny' 'Warm' '?' 'Strong' '?' '?']

Final General_h:
[['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?']]
```

***