

2 : Implement and demonstrate the Candidate-Elimination algorithm

```
import csv
```

```
data_list = []

with open('data.csv', 'r') as csvFile:
    reader = csv.reader(csvFile)
    for row in reader:
        data_list.append(row)
```

```
*first_sample, output = data_list[0]
num_attributes = len(first_sample)

S = ['0'] * num_attributes
G = ['?'] * num_attributes

print("The Initial value of hypothesis", end='\n\n')
print ("The most specific hypothesis S0 : ", S)
print ("The most general hypothesis G0 : ", G)
```

The Initial value of hypothesis

The most specific hypothesis S0 : ['0', '0', '0', '0', '0', '0']
The most general hypothesis G0 : ['?', '?', '?', '?', '?', '?']

```
# Comparing with First Training Example ( Assigning )
```

```
S = first_sample[:]
```

```

# Comparing with Remaining Training Examples of Given Data Set

print("Candidate Elimination algorithm  Hypotheses Version Space Computation", end='\n\n')

general_hypothesis_space = []
outer_index = 1

for *data, output in data_list:

    if output == 'Y':
        for index, attribute in enumerate(data):
            if attribute != S[index]:
                S[index] = '?'

        for general_hypothesis in general_hypothesis_space:
            for index, attribute in enumerate(general_hypothesis):
                if attribute not in {'?', S[index]}:
                    general_hypothesis_space.remove(general_hypothesis)
                    #remove it if it's not matching with the specific hypothesis

    elif output == 'N':
        for index, attribute in enumerate(data):
            if S[index] not in {'?', attribute}:
                # if not matching with the specific Hypothesis take it seperately and store it
                G[index] = S[index]
            general_hypothesis_space.append(G)
            # this is the version space to store all Hypotheses
            G = ['?'] * num_attributes
            # resetting

#-----printing section-----
print()
print("for training example no : {0}, S{0}: ".format(outer_index), S)

if ( len(general_hypothesis_space) == 0 ):
    print("for training example no : {0}, G{0}: ".format(outer_index), G)
else:
    print("for training example no : {0}, G{0}: ".format(outer_index), general_hypothesis_space)
print('-' * 90)
#-----

outer_index += 1

```

Candidate Elimination algorithm Hypotheses Version Space Computation

```

for training example no : 1, S1: ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same']
for training example no : 1, G1: ['?', '?', '?', '?', '?', '?']
-----

for training example no : 2, S2: ['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']
for training example no : 2, G2: ['?', '?', '?', '?', '?', '?']
-----

for training example no : 3, S3: ['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']
for training example no : 3, G3: [['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', 'Same']]
-----

for training example no : 4, S4: ['Sunny', 'Warm', '?', 'Strong', '?', '?']
for training example no : 4, G4: [['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?']]
-----

```

```

print("Specific hypothesis : ", S)
print()
print("General hypothesis : ", general_hypothesis_space)

```

Specific hypothesis : ['Sunny', 'Warm', '?', 'Strong', '?', '?']

General hypothesis : [['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?']]

```
def get_version_space(S, G):
    version_space = [S]      # Initialize the version space list and append S (Specific hypothesis)

    for i in range(len(S)):
        for general_hypothesis in G:
            if general_hypothesis[i] != S[i]:
                temp_hypothesis = list(general_hypothesis)
                temp_hypothesis[i] = S[i]
                if temp_hypothesis not in version_space:
                    version_space.append(temp_hypothesis)

    version_space.extend(G)   # Finally put hypotheses that exist in G
                             # extending = multiple appends

    return version_space

print("Version Space : ")
get_version_space(S, general_hypothesis_space)
```

Version Space :

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

```
# PS: Dataset for clarity
print(open('data.csv').read())
```

```
Sunny,Warm,Normal,Strong,Warm,Same,Y
Sunny,Warm,High,Strong,Warm,Same,Y
Rainy,Cold,High,Strong,Warm,Change,N
Sunny,Warm,High,Strong,Cool,Change,Y
```