21/09/20

EXPERIMENT NO. 2

1KS17CS030

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

TASK: The candidate-Elimination Algorithm computes the version space containing all hypotheses from H that are consistent with an observed sequence of training examples.

Dataset: EnjoySports Training Examples:

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

ALGORITHM:

Initialize G to the set of maximally general hypothesis in H

Initialize S to the set of maximally specific hypothesis in H.

For each training example d, do

- If d is a positive example
  - Remove from G any hypothesis inconsistent with d.
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      - h is consistent with d, and some member of G is more general than h.
    - Remove from S any hypothesis that is more general than another hypothesis in S.

- If d is negative example
  - Remove from S any hypothesis inconsistent with d.
  - For each hypothesis g in G that is not consistent with d
    - Remove from G
    - Add to G all minimal specializations h of g such that
      - h is consistent with d, and some member of S is more specific than h.
    - Remove from G any hypothesis that is less general than another hypothesis in G.

PROGRAM:

```
import numpy as np
import pandas as pd
#Loading data from CSV file
data = pd.DataFrame(data = pd.read_csv('dataset.csv'))
# Separating concept features from Target
concepts = np.array(data.iloc[:,0:-1])
# Isolating target into a separate DataFrame
target = np.array(data.iloc[:,-1])

def learn(concepts, target):
    # creating a new list
    specific_h = concept[0].copy()
    general_h = [['?' for i in range len(specific_h))] for i in
      range (len (specific_h))]
    # learning iterations
    for i, h in enumerate(concepts):
```

```
    # positive target
    if target[i] == "Yes":
        for x in range(len(specific_h)):
            if h[x] != specific_h[x]:
                specific_h[x] = '?'
                general_h[x][x] = '?'
    if target[i] == "No":
        for x in range(len(specific_h)):
            if h[x] != specific_h[x]:
                general_h[x][x] = specific_h[x]
            else:
                general_h[x][x] = '?'
    # Indices having empty rows, those that are unchanged
    indices = [i for i, val in enumerate(general_h) if val==['?',
    '?', '?', '?', '?', '?']]
    for i in indices:
        general_h.remove(['?', '?', '?', '?', '?', '?'])
    return specific_h, general_h
s_final, g_final = learn(concepts, target)
print("Final S:", s_final, sep="\n")
print("Final G:", g_final, sep="\n")
```

OUTPUT:

```
Final_S:
['Sunny' 'Warm' '?' 'Strong' '?' '?']
Final_G:
[['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?']]
```