**U19EC046 | ML | LAB 2**

**AIM**

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.

**THEORY**

The candidate elimination algorithm incrementally builds the version space given a hypothesis space H and a set E of examples. The examples are added one by one; each example possibly shrinks the version space by removing the hypotheses that are inconsistent with the example. The candidate elimination algorithm does this by updating the general and specific boundary for each new example.

* You can consider this as an extended form of Find-S algorithm.
* Consider both positive and negative examples.
* Actually, positive examples are used here as Find-S algorithm (Basically they are generalizing from the specification).
* While the negative example is specified from generalize form.

**ALGORITHM**

Step1: Load Data set

Step2: Initialize General Hypothesis  and Specific  Hypothesis.

Step3: For each training example

Step4: If example is positive example

if attribute value == hypothesis value:

Do nothing

else:

replace attribute value with '?' (Basically generalizing it)

Step5: If example is Negative example

Make generalize hypothesis more specific.

**CODE**

|  |
| --- |
| **import numpy as np**  **import pandas as pd**  **class CandiateElimination:**  **def \_\_init\_\_(self, csvPath):**  **self.csvPath = csvPath**  **self.data = pd.read\_csv(self.csvPath)**  **self.specificHypothesis = []**  **self.generalHypothesis = []**  **self.initailizeSpecificHypothesis()**  **self.hypothesisLength = len(self.specificHypothesis)**  **self.initalizeGeneralHypothesis()**    **def seriesToList(self, series):**  **return [value for \_, value in series.items()]**    **def initailizeSpecificHypothesis(self):**  **for i, r in self.data.iterrows():**  **if r[-1] == 'Yes':**  **self.specificHypothesis =  self.seriesToList(r)[:-1]**  **break**    **def initalizeGeneralHypothesis(self):**  **for i in range(self.hypothesisLength):**  **self.generalHypothesis.append(**  **['?' for \_ in range(self.hypothesisLength)])**    **def updateSpecificHypothisis(self, instance):**  **for i in range(len(self.specificHypothesis)):**  **if self.specificHypothesis[i] != '?'**  **and self.specificHypothesis[i] != instance[i]:**  **self.specificHypothesis[i] = '?'**  **def updateGeneralHypothisis(self, instance):**  **for i in range(self.hypothesisLength):**  **if self.specificHypothesis[i] != '?'**  **and self.specificHypothesis[i] != instance[i]:**  **self.generalHypothesis[i][i] = self.specificHypothesis[i]**    **def discardContradicting(self):**  **\_generalHypothesis = []**  **for i in range(self.hypothesisLength):**  **if self.generalHypothesis[i][i] == self.specificHypothesis[i] and self.specificHypothesis[i] != '?':**  **\_generalHypothesis.append(self.generalHypothesis[i])**  **self.generalHypothesis = \_generalHypothesis**    **def fit(self, instance):**  **currInstance = self.seriesToList(instance)**  **if currInstance[-1]=='Yes':**  **self.updateSpecificHypothisis(currInstance)**  **else:**  **self.updateGeneralHypothisis(currInstance)**  **def getHypothesis(self):**  **for i, r in self.data.iterrows():**  **self.fit(r)**  **self.discardContradicting()**  **return (self.specificHypothesis, self.generalHypothesis)**    **myHypothesis = CandiateElimination('cea.csv')**  **(specific, general) = myHypothesis.getHypothesis()**  **print(f"\**  **Specific Hypothesis : {specific}\n\**  **General Hypothesis : {general}\**  **")** |

**OUTPUT**

Specific Hypothesis :

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

General Hypothesis :

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

**CONCLUSION**

In this practical we have implemented python code for Candidate elimination algorithm and found specific and general hypothesis for given dataset.