S.SRIHARI - 2018103601
MACHINE LEARNING LAB EX-2
CANDIDATE ELIMINATION ALGORITHM

DATASET

Desktop/SEM-6/Machine_Learni | enjoysport.csv - Jupyter Text Edit | ML-Lab-Week2-CandidateElimin

localhost:8888/edit/Desktop/SEM-6/Machine_Learning/lab/lab-week2/enjoysport.csv

jupyter enjoysport.csv a few seconds ago

Logout

File Edit View Language current mode

```csv
Sky,AirTemp,Humidity,Wind,Water,Forecast,EnjoySport
Sunny,Warm,Normal,Strong,Warm,Same,Yes
Sunny,Warm,High,Strong,Warm,Same,Yes
Rainy,Cold,High,Strong,Warm,Change,No
Sunny,Warm,High,Strong,Cool,Change,Yes
```

enjoysport.csv - Jupyter Text Edit × ML-Lab-Week2-CandidateElimina ×

88/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [24]:
```
import numpy as np
import pandas as pd
```

In [25]:
```
def load_data(filename, target):
    dataset = pd.read_csv(filename, sep=",")  # read .csv file into dataset variable

    print(dataset,"\n")

    x = np.array(dataset.drop([target],1)) # x contains all the features. Does not include target
    y = np.array(dataset[target]) # y contains the target class

    print("\nX = \n",x)
    print("\nY = \n",y)

    print("\n\n")

    return (dataset,x,y,target)

inp = load_data("enjoysport.csv","EnjoySport")
print(inp[0])
```

```
     Sky AirTemp Humidity    Wind Water Forecast EnjoySport
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


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

Y =
 ['Yes' 'Yes' 'No' 'Yes']



     Sky AirTemp Humidity    Wind Water Forecast EnjoySport
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
```

enjoysport.csv - Jupyter Text Edit | ML-Lab-Week2-CandidateElimin

st:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [26]:
```
def convert_data(dataset,x,y,target):
    header_names = []
    for i in range(len(dataset.columns)):
        if dataset.columns[i] != target:
            header_names.append(dataset.columns[i])
    print("Header Names: ",tuple(header_names))

    #print("srihari",len(dataset))
    data = []
    for i in range(len(dataset)):
        new = [(tuple(x[i])),y[i]]
        data.append(tuple(new))

    print("\nDataset: ",data)

    return (data, header_names)


data_inp = convert_data(inp[0],inp[1],inp[2],inp[3])
```

```
Header Names:  ('Sky', 'AirTemp', 'Humidity', 'Wind', 'Water', 'Forecast')

Dataset:  [(('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same'), 'Yes'), (('Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Sam
e'), 'Yes'), (('Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change'), 'No'), (('Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Chang
e'), 'Yes')]
```

enjoysport.csv - Jupyter Text Edit × ML-Lab-Week2-CandidateElimin ×

ost:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [27]:
```
class holder():
    factors = {}
    attributes = ()

    def __init__(self,attr):
        self.attributes = attr
        for i in attr:
            self.factors[i] = []

    def add_values(self,data,header_names):
        for i in range(len(data)):
            for j in range(len(header_names)):
                # 3dimensions as 2 tuples in data
                if data[i][0][j] not in self.factors[header_names[j]]:
                    self.factors[header_names[j]].append(data[i][0][j])

f = holder(data_inp[1])
f.add_values(data_inp[0],data_inp[1])
print("\nFactors: ",f.factors)
print("\nAttributes: ",f.attributes)
```

```
Factors:  {'Sky': ['Sunny', 'Rainy'], 'AirTemp': ['Warm', 'Cold'], 'Humidity': ['Normal', 'High'], 'Wind': ['Strong'], 'Water':
['Warm', 'Cool'], 'Forecast': ['Same', 'Change']}

Attributes:  ['Sky', 'AirTemp', 'Humidity', 'Wind', 'Water', 'Forecast']
```

enjoysport.csv - Jupyter Text Edit ML-Lab-Week2-CandidateElimin

host:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [28]:
```
class CandidateElimination():
    Positive = {}
    Negative = {}

    def __init__(self,data,fact):
        self.num_factors = len(data[0][0])
        self.factors = fact.factors
        self.attr = fact.attributes
        self.dataset = data


    def run_algorithm(self):
        G = self.initializeG()
        #print(type(G[0]))
        S = self.initializeS()

        count = 0
        for trial_set in self.dataset:  # check for every input
            count+=1
            print("\nInput %d: "%(count),trial_set)
            if self.is_positive(trial_set):
                print("Positive")
                #print(trial_set[0])
                G = self.remove_inconsistent_G(G,trial_set[0])
                S_new = S[:]
                #print("S_new: ",S_new)
                for s in S:

                    if not self.consistent(s,trial_set[0]):
                        S_new.remove(s)
                        print(trial_set[0])
                        generalization = self.generalize_inconsistent_S(s, trial_set[0])
                        generalization = self.get_general(generalization,G)

                        if generalization:
                            S_new.append(generalization)
```

Learnin × | enjoysport.csv - Jupyter Text Edit × | ML-Lab-Week2-CandidateElimina × +

localhost:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
                S = S_new[:]
                S = self.remove_more_general(S)
            print("S%d: "%(count),S)
            print("G%d: "%(count),G)


        else:   # if negative input
            print("Negative")
            S = self.remove_inconsistent_S(S, trial_set[0])
            G_new = G[:]
            for g in G:
                if self.consistent(g,trial_set[0]):
                    G_new.remove(g)
                    specialization = self.specialize_inconsistent_G(g, trial_set[0])
                    #print("srihari",specialization)
                    specialization = self.get_specific(specialization, S)
                    if specialization != []:
                        G_new += specialization
                G = G_new[:]
                G = self.remove_more_specific(G)
            print("S%d: "%(count),S)
            print("G%d: "%(count),G)

    print("\nFinal S: ", S)
    print("Final G: ",G)


def initializeG(self):
    G = tuple(['?' for i in range(self.num_factors)])
    return [G]

def initializeS(self):
    S = tuple(['-' for i in range(self.num_factors)])
    return [S]
```

Learnin × | enjoysport.csv - Jupyter Text Edit × | ML-Lab-Week2-CandidateElimina × | +

localhost:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
    def is_positive(self,trial_set):
        if trial_set[1] == 'Yes' or trial_set[1] == 1:
            return True
        elif trial_set[1] == 'No' or trial_set[1] == 0:
            return False
        else:
            raise TypeError("Invalid Target Value")

    def remove_inconsistent_G(self,hypothesis,instance):
        G_new = hypothesis[:]
        for g in hypothesis:
            if not self.consistent(g,instance):
                G_new.remove(g)
        return G_new

    def remove_inconsistent_S(self,hypothesis,instance):
        S_new = hypothesis[:]
        for s in hypothesis:
            if self.consistent(s,instance):
                S_new.remove(s)
        return S_new


    def consistent(self, hypothesis, instance):
        for i,factor in enumerate(hypothesis):
            # enumerate returns index,val
            if not self.match_factor(factor, instance[i]):
                return False
        return True

    def match_factor(self, val1, val2):
        if val1 == '?' or val2 == '?':
            return True
        elif val1 == val2:
            return True
        return False
```

e_Learnin × | enjoysport.csv - Jupyter Text Edit × | ML-Lab-Week2-CandidateElimina × | +

localhost:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
    def generalize_inconsistent_S(self, hypothesis,instance):
        hypo = list(hypothesis)
        for i,factor in enumerate(hypo):
            if factor == '-':
                hypo[i] = instance[i]
            elif not self.match_factor(factor,instance[i]):
                hypo[i] = '?'
        return tuple(hypo)

    def specialize_inconsistent_G(self, hypothesis,instance):
        specializations = []
        hypo = list(hypothesis)
        for i,factor in enumerate(hypo):
            if factor == '?':
                values = self.factors[self.attr[i]]
                for j in values:
                    if instance[i] != j:
                        hyp = hypo[:]
                        hyp[i] = j
                        hyp=tuple(hyp)
                        specializations.append(hyp)
        return specializations

    def get_general(self, generalization, G):
        for g in G:
            if self.more_general(g,generalization):
                return generalization
        return None

    def get_specific(self, specializations, S):
        valid_specialization = []
        for hypo in specializations:
            for s in S:
                if self.more_specific(s,hypo) or s==self.initializeS()[0]:
                    valid_specialization.append(hypo)
        return valid_specialization
```

earnin × | enjoysport.csv - Jupyter Text Edit × | ML-Lab-Week2-CandidateElimin × | +

alhost:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
    def more_general(self, hypo1, hypo2):
        hyp = zip(hypo1,hypo2)
        for i,j in hyp:
            if i == '?':
                continue
            elif j == '?' and i != '?':
                return False
            elif i != j:
                return False
            else:
                continue
        return True

    def more_specific(self,hypo1,hypo2):
        return self.more_general(hypo2, hypo1)

    def remove_more_general(self, hypotheses):
        S_new = hypotheses[:]
        for old in hypotheses:
            for new in S_new:
                if old != new and self.more_general(new,old):
                    S_new.remove[new]

        return S_new

    def remove_more_specific(self, hypotheses):
        G_new = hypotheses[:]
        for old in hypotheses:
            for new in G_new:
                if old != new and self.more_specific(new,old):
                    G_new.remove(new)

        return G_new

#print(f)
output = CandidateElimination(data_inp[0],f)
output.run_algorithm()
```

OUTPUT

enjoysport.csv - Jupyter Text Edit × ML-Lab-Week2-CandidateElimina ×

:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (unsaved changes) Logout

e Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
Input 1:  (('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same'), 'Yes')
Positive
('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same')
S1:  [('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same')]
G1:  [('?', '?', '?', '?', '?', '?')]

Input 2:  (('Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same'), 'Yes')
Positive
('Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same')
S2:  [('Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same')]
G2:  [('?', '?', '?', '?', '?', '?')]

Input 3:  (('Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change'), 'No')
Negative
S3:  [('Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same')]
G3:  [('Sunny', '?', '?', '?', '?', '?'), ('?', 'Warm', '?', '?', '?', '?'), ('?', '?', '?', '?', '?', 'Same')]

Input 4:  (('Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change'), 'Yes')
Positive
('Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change')
S4:  [('Sunny', 'Warm', '?', 'Strong', '?', '?')]
G4:  [('Sunny', '?', '?', '?', '?', '?'), ('?', 'Warm', '?', '?', '?', '?')]

Final S:  [('Sunny', 'Warm', '?', 'Strong', '?', '?')]
Final G:  [('Sunny', '?', '?', '?', '?', '?'), ('?', 'Warm', '?', '?', '?', '?')]
```

In [ ]:

_Learnin ✕ | enjoysport.csv - Jupyter Text Edit ✕ | ML-Lab-Week2-CandidateElimina ✕ | +

localhost:8888/edit/Desktop/SEM-6/Machine_Learning/lab/lab-week2/enjoysport.csv

jupyter enjoysport.csv ✓ a few seconds ago

File Edit View Language

```
Sky,AirTemp,Humidity,Wind,Water,Forecast,EnjoySport
Sunny,Warm,Normal,Strong,Warm,Same,Yes
Sunny,Warm,High,Weak,Warm,Same,Yes
Rainy,Cold,High,Strong,Warm,Change,No
Sunny,Warm,High,Strong,Cool,Change,Yes
```

CHANGE IN THE DATASET

CHANGE IN THE OUTPUT

Learnin × | enjoysport.csv - Jupyter Text Edit × | ML-Lab-Week2-CandidateEliminc × | +

localhost:8888/notebooks/Desktop/SEM-6/Machine_Learning/lab/lab-week2/ML-Lab-Week2-CandidateEliminationAlgorithm.ipynb

jupyter ML-Lab-Week2-CandidateEliminationAlgorithm Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
Input 1:  (('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same'), 'Yes')
Positive
('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same')
S1:  [('Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same')]
G1:  [('?', '?', '?', '?', '?', '?')]

Input 2:  (('Sunny', 'Warm', 'High', 'Weak', 'Warm', 'Same'), 'Yes')
Positive
('Sunny', 'Warm', 'High', 'Weak', 'Warm', 'Same')
S2:  [('Sunny', 'Warm', '?', '?', 'Warm', 'Same')]
G2:  [('?', '?', '?', '?', '?', '?')]

Input 3:  (('Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change'), 'No')
Negative
S3:  [('Sunny', 'Warm', '?', '?', 'Warm', 'Same')]
G3:  [('Sunny', '?', '?', '?', '?', '?'), ('?', 'Warm', '?', '?', '?', '?'), ('?', '?', '?', '?', '?', 'Same')]

Input 4:  (('Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change'), 'Yes')
Positive
('Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change')
S4:  [('Sunny', 'Warm', '?', '?', '?', '?')]
G4:  [('Sunny', '?', '?', '?', '?', '?'), ('?', 'Warm', '?', '?', '?', '?')]

Final S:  [('Sunny', 'Warm', '?', '?', '?', '?')]
Final G:  [('Sunny', '?', '?', '?', '?', '?'), ('?', 'Warm', '?', '?', '?', '?')]
```