CODE:

# Assignment 2: 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 hypothesis consistent with the training examples.  
  
  
from csv import reader  
  
  
with open('dataset.csv') as csv\_file:  
 data = reader(csv\_file)  
  
 attrs = next(data)[:-**1**]  
 print('\nAttributes:'**,** attrs**,** '\n')  
 attr\_range = range(len(attrs))  
 s = ['Φ' for \_ in attr\_range]  
 g = [['?' for \_ in attr\_range] for \_ in attr\_range]  
 print('Specific hypothesis:'**,** s)  
 print('General hypothesis:'**,** g**,** '\n')  
  
 # Finding and saving opposites of values (application- when first record is -ve)  
 data = list(data) # typecasting iterator to iterable  
  
set\_list**,** dict\_list = [set() for \_ in attr\_range]**,** [{} for \_ in attr\_range]  
for i in attr\_range:  
 set\_**,** dict\_ = set\_list[i]**,** dict\_list[i]  
 for record in data:  
 set\_.add(record[i])  
 if len(set\_) == **2**: # coz any attr can have 2 values at max  
 e1**,** e2 = set\_  
 dict\_[e1]**,** dict\_[e2] = e2**,** e1  
 break  
 else: # if opposite state of a value doesn't exist in (whole column of) dataset  
 dict\_[set\_.pop()] = '?'  
print('Opposites:'**,** dict\_list**,** '\n')  
  
# Assigning first record:  
for values in data: # Candidate Elimination considers both +ve as well as -ve records  
 print('Values:'**,** values)  
 if values[-**1**] == 'Yes': # +ve  
 s = values[:-**1**]  
 print('s:'**,** s**,** '\n')  
 break  
 else: # -ve  
 for i**,** value in enumerate(values[:-**1**]):  
 g[i][i] = dict\_list[i][value]  
 print('g:'**,** g**,** '\n')  
  
for values in data:  
 print('Values:'**,** values)  
 if values[-**1**] == 'Yes': # generalize  
 for i**,** value in enumerate(values[:-**1**]):  
 if s[i] != value:  
 s[i] = '?'  
 print('s:'**,** s)  
 else: # specify  
 for i**,** value in enumerate(values[:-**1**]):  
 if s[i] != value:  
 g[i][i] = s[i]  
 print('g:'**,** g)  
 print()  
  
  
# Syncing s and g:  
for i**,** g\_list in enumerate(g[:]): # https://stackoverflow.com/a/1207427/3064538  
 if s[i] == '?' or g\_list == ['?' for \_ in attr\_range]:  
 g.remove(g\_list)  
  
print('Generalised hypothesis:'**,** s)  
print('Specified hypothesis:'**,** g**,** '\n')  
  
# Making version space: (note- following logic needs to be reviewed)  
version\_space = set()  
for g\_list in g:  
 for i**,** g\_value in enumerate(g\_list):  
 if s[i] == g\_value:  
 for j**,** s\_value in enumerate(s):  
 if g\_value != '?' and s\_value != '?' and g\_value != s\_value:  
 temp = ['?' for \_ in attr\_range]  
 temp[i]**,** temp[j] = g\_value**,** s\_value  
 version\_space.add(tuple(temp)) # coz list is not hashable  
  
print('Version space:'**,** version\_space)

OUTPUT:

Attributes: ['Sky', 'Temp', 'Humidity', 'Wind', 'Water', 'Forecast']

Specific hypothesis: ['Φ', 'Φ', 'Φ', 'Φ', 'Φ', 'Φ']

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

Opposites: [{'Rainy': 'Sunny', 'Sunny': 'Rainy'}, {'Warm': 'Cold', 'Cold': 'Warm'}, {'Normal': 'High', 'High': 'Normal'}, {'Strong': '?'}, {'Warm': 'Cool', 'Cool': 'Warm'}, {'Change': 'Same', 'Same': 'Change'}]

Values: ['Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change', 'No']

g: [['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?'], ['?', '?', 'Normal', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', 'Cool', '?'], ['?', '?', '?', '?', '?', 'Same']]

Values: ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same', 'Yes']

s: ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same']

Values: ['Rainy', 'Cold', 'High', 'Strong', 'Warm', 'Change', 'No']

g: [['Sunny', '?', '?', '?', '?', '?'], ['?', 'Warm', '?', '?', '?', '?'], ['?', '?', 'Normal', '?', '?', '?'], ['?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', 'Cool', '?'], ['?', '?', '?', '?', '?', 'Same']]

Values: ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same', 'Yes']

s: ['Sunny', 'Warm', 'Normal', 'Strong', 'Warm', 'Same']

Values: ['Sunny', 'Warm', 'High', 'Strong', 'Warm', 'Same', 'Yes']

s: ['Sunny', 'Warm', '?', 'Strong', 'Warm', 'Same']

Values: ['Sunny', 'Warm', 'High', 'Strong', 'Cool', 'Change', 'Yes']

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

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

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

Version space: {('Sunny', 'Warm', '?', '?', '?', '?'), ('?', 'Warm', '?', 'Strong', '?', '?'), ('Sunny', '?', '?', 'Strong', '?', '?')}