Veermata Jijabai Technological Institute, Mumbai 400019

Assignment No.: 02

Aim : Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

Name : Kiran K. Patil

Enrollment No.: 211070904

Branch : Computer Engineering

Course: Machine Learning Lab

Batch : IV

```
from google.colab import files
files.upload()
```

```
# Attributes
# survival - Survival (0 = No; 1 = Yes)
# class - Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
# name - Name
# sex - Sex
# age - Age
# sibsp - Number of Siblings/Spouses Aboard
# parch - Number of Parents/Children Aboard
# ticket - Ticket Number
# fare - Passenger Fare
# cabin - Cabin
# embarked - Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)
```

```
import pandas as pd
import numpy as np

#to read the data in the csv file
data = pd.read_csv("titanic_dataset.csv")
# print(data,"n")

data = data[5:15]
print(data)
```

```
    PassengerId  Survived  Pclass  \
5           897         0       3
6           898         1       3
7           899         0       2
8           900         1       3
9           901         0       3
10          902         0       3
11          903         0       1
12          904         1       1
13          905         0       2
14          906         1       1

                                            Name     Sex   Age  SibSp  \
5                          Svensson, Mr. Johan Cervin    male  14.0      0
6                               Connolly, Miss. Kate  female  30.0      0
7                        Caldwell, Mr. Albert Francis    male  26.0      1
8            Abrahim, Mrs. Joseph (Sophie Halaut Easu)  female  18.0      0
9                            Davies, Mr. John Samuel    male  21.0      2
10                                  Ilieff, Mr. Ylio    male   NaN      0
11                         Jones, Mr. Charles Cresson    male  46.0      0
12      Snyder, Mrs. John Pillsbury (Nelle Stevenson)  female  23.0      1
13                              Howard, Mr. Benjamin    male  63.0      1
14  Chaffee, Mrs. Herbert Fuller (Carrie Constance...  female  47.0      1

    Parch       Ticket     Fare Cabin Embarked
5       0         7538   9.2250   NaN        S
6       0       330972   7.6292   NaN        Q
7       1       248738  29.0000   NaN        S
8       0         2657   7.2292   NaN        C
9       0    A/4 48871  24.1500   NaN        S
10      0       349220   7.8958   NaN        S
11      0          694  26.0000   NaN        S
12      0        21228  82.2667   B45        S
13      0        24065  26.0000   NaN        S
14      0  W.E.P. 5734  61.1750   E31        S
```

```
data.drop('PassengerId', inplace=True, axis=1)
```

```
data.head()
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

```
data.drop('Name', inplace=True, axis=1)
data.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
5	0	3	male	14.0	0	0	7538	9.2250	NaN	S
6	1	3	female	30.0	0	0	330972	7.6292	NaN	Q
7	0	2	male	26.0	1	1	248738	29.0000	NaN	S
8	1	3	female	18.0	0	0	2657	7.2292	NaN	C
9	0	3	male	21.0	2	0	A/4 48871	24.1500	NaN	S

```
data.drop('Ticket', inplace=True, axis=1)
data.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
5	0	3	male	14.0	0	0	9.2250	NaN	S
6	1	3	female	30.0	0	0	7.6292	NaN	Q
7	0	2	male	26.0	1	1	29.0000	NaN	S
8	1	3	female	18.0	0	0	7.2292	NaN	C
9	0	3	male	21.0	2	0	24.1500	NaN	S

```
data.drop('Fare', inplace=True, axis=1)
data.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
5	0	3	male	14.0	0	0	NaN	S
6	1	3	female	30.0	0	0	NaN	Q
7	0	2	male	26.0	1	1	NaN	S
8	1	3	female	18.0	0	0	NaN	C
9	0	3	male	21.0	2	0	NaN	S

```
#making an array of all the attributes
d = np.array(data)[:,:]
print("\n The attributes are: ",d)
```

```
 The attributes are:  [[0 3 'male' 14.0 0 0 nan 'S']
 [1 3 'female' 30.0 0 0 nan 'Q']
 [0 2 'male' 26.0 1 1 nan 'S']
 [1 3 'female' 18.0 0 0 nan 'C']
 [0 3 'male' 21.0 2 0 nan 'S']
 [0 3 'male' nan 0 0 nan 'S']
 [0 1 'male' 46.0 0 0 nan 'S']
 [1 1 'female' 23.0 1 0 'B45' 'S']
 [0 2 'male' 63.0 1 0 nan 'S']
 [1 1 'female' 47.0 1 0 'E31' 'S']]
```

```
target = np.array(data)[:,0]
print("\n The target is: ",target)
```

```
 The target is:  [0 1 0 1 0 0 0 1 0 1]
```

```
print(data)
```

```
    Survived  Pclass     Sex   Age  SibSp  Parch Cabin Embarked
5          0       3    male  14.0      0      0   NaN        S
6          1       3  female  30.0      0      0   NaN        Q
7          0       2    male  26.0      1      1   NaN        S
8          1       3  female  18.0      0      0   NaN        C
9          0       3    male  21.0      2      0   NaN        S
10         0       3    male   NaN      0      0   NaN        S
```

```
    11        0       1    male  46.0      0      0   NaN        S
    12        1       1  female  23.0      1      0   B45        S
    13        0       2    male  63.0      1      0   NaN        S
    14        1       1  female  47.0      1      0   E31        S
```

```
def train(c,t):
    for i, val in enumerate(t):
        if val == 1:
            specific_hypothesis = c[i].copy()
            break

    for i, val in enumerate(c):
        if t[i] == 1:
            for x in range(len(specific_hypothesis)):
                if val[x] != specific_hypothesis[x]:
                    specific_hypothesis[x] = '?'
                else:
                    pass

    return specific_hypothesis
```

```
print("\n The final hypothesis is:",train(d,target))
```

```
 The final hypothesis is: [1 '?' 'female' '?' '?' 0 '?' '?']
```

The second Hypothesis After applying diffrent filters on the data

```
data_frame = pd.read_csv("titanic_dataset.csv")
data_frame.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S

```
# print(data_frame)
data_frame.drop('PassengerId', inplace=True, axis=1)
data_frame.drop('Name', inplace=True, axis=1)
data_frame.drop('Ticket', inplace=True, axis=1)
data_frame.drop('Fare', inplace=True, axis=1)
data_frame.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
0	0	3	male	34.5	0	0	NaN	Q
1	1	3	female	47.0	1	0	NaN	S
2	0	2	male	62.0	0	0	NaN	Q
3	0	3	male	27.0	0	0	NaN	S
4	1	3	female	22.0	1	1	NaN	S

```
data_frame.sort_values('Age')
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
354	1	3	female	0.17	1	2	NaN	S
201	0	3	male	0.33	0	2	NaN	S
281	0	3	male	0.75	1	1	NaN	S
307	0	3	male	0.83	0	1	NaN	S
250	1	2	female	0.92	1	2	NaN	S

```
valid_df=data_frame.dropna()
valid_df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
12	1	1	female	23.0	1	0	B45	S
14	1	1	female	47.0	1	0	E31	S
24	1	1	female	48.0	1	3	B57 B59 B63 B66	C
26	1	1	female	22.0	0	1	B36	C
28	0	1	male	41.0	0	0	A21	S

```
sorted_df=valid_df.sort_values('Age')
sorted_df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
117	1	3	female	1.0	1	1	G6	S
196	0	1	male	6.0	0	2	E34	C
326	1	2	female	12.0	2	1	F4	S
64	0	1	male	13.0	2	2	B57 B59 B63 B66	C
395	1	1	female	18.0	1	0	C31	S

```
sorted_df.head()
sorted_df=sorted_df[:25]
```

```
#making an array of all the attributes
d1 = np.array(sorted_df)[:,:]
print("\n The attributes are: ",d1)
```

```
 The attributes are:  [[1 3 'female' 1.0 1 1 'G6' 'S']
 [0 1 'male' 6.0 0 2 'E34' 'C']
 [1 2 'female' 12.0 2 1 'F4' 'S']
 [0 1 'male' 13.0 2 2 'B57 B59 B63 B66' 'C']
 [1 1 'female' 18.0 1 0 'C31' 'S']
 [1 1 'female' 18.0 1 0 'D30' 'S']
 [0 2 'male' 18.5 0 0 'F' 'S']
 [0 2 'male' 20.0 0 0 'D38' 'C']
 [1 1 'female' 22.0 0 1 'B36' 'C']
 [1 2 'female' 22.0 0 0 'F33' 'S']
 [1 1 'female' 23.0 0 1 'C54' 'C']
 [1 1 'female' 23.0 1 0 'B45' 'S']
 [0 1 'male' 23.0 0 0 'B24' 'S']
 [0 1 'male' 24.0 1 0 'C31' 'S']
 [0 1 'male' 24.0 1 0 'B45' 'S']
 [1 1 'female' 25.0 1 0 'E50' 'C']
 [0 3 'male' 25.0 0 0 'F E57' 'C']
 [0 3 'male' 25.0 0 0 'F G63' 'S']
 [1 1 'female' 26.0 1 0 'C89' 'C']
 [0 2 'male' 26.0 0 0 'F2' 'S']
 [1 1 'female' 27.0 1 1 'B58 B60' 'C']
 [0 1 'male' 27.0 1 0 'C89' 'C']
 [1 1 'female' 27.0 1 2 'B71' 'S']
 [1 1 'female' 28.0 3 2 'C23 C25 C27' 'S']
 [0 1 'male' 28.5 0 0 'D43' 'C']]
```

```
target1 = np.array(sorted_df)[:,0]
print("\n The target is: ",target1)
```

```
The target is:  [1 0 1 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 1 0 1 0 1 1 0]
```

```
print("\n The final hypothesis is:",train(d1,target1))
```

```
The final hypothesis is: [1 '?' 'female' '?' '?' '?' '?' '?']
```

Grouping bye Age with intewal of 10 and then applying Find-s on it

```
# sorted_df.where(sorted_df <= 9, 10, inplace=True)
sorted_df['Age'].values[sorted_df['Age'].values < 9] = 10
sorted_df['Age'].values[(sorted_df['Age'].values > 10) & (sorted_df['Age'].values < 20)] = 20
sorted_df['Age'].values[(sorted_df['Age'].values > 20) & (sorted_df['Age'].values < 30)] = 30
```

```
# sorted_df.head()
# print(sorted_df)
sorted_df=sorted_df[10:25]
# print()
sorted_df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
150	1	1	female	30.0	0	1	C54	C
12	1	1	female	30.0	1	0	B45	S
390	0	1	male	30.0	0	0	B24	S
50	0	1	male	30.0	1	0	C31	S
287	0	1	male	30.0	1	0	B45	S

```
#making an array of all the attributes
d2 = np.array(sorted_df)[:,:]
print("\n The attributes are: ",d2)

target2 = np.array(sorted_df)[:,0]
print("\n The target is: ",target2)
```

```
The attributes are:  [[1 1 'female' 30.0 0 1 'C54' 'C']
[1 1 'female' 30.0 1 0 'B45' 'S']
[0 1 'male' 30.0 0 0 'B24' 'S']
[0 1 'male' 30.0 1 0 'C31' 'S']
[0 1 'male' 30.0 1 0 'B45' 'S']
[1 1 'female' 30.0 1 0 'E50' 'C']
[0 3 'male' 30.0 0 0 'F E57' 'C']
[0 3 'male' 30.0 0 0 'F G63' 'S']
[1 1 'female' 30.0 1 0 'C89' 'C']
[0 2 'male' 30.0 0 0 'F2' 'S']
[1 1 'female' 30.0 1 1 'B58 B60' 'C']
[0 1 'male' 30.0 1 0 'C89' 'C']
[1 1 'female' 30.0 1 2 'B71' 'S']
[1 1 'female' 30.0 3 2 'C23 C25 C27' 'S']
[0 1 'male' 30.0 0 0 'D43' 'C']]

The target is:  [1 1 0 0 0 1 0 0 1 0 1 0 1 1 0]
```

```
print("\n The final hypothesis is:",train(d2,target2))
```

```
The final hypothesis is: [1 1 'female' 30.0 '?' '?' '?' '?']
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import files
files.upload()
```

```
df = pd.read_csv('titanic_dataset.csv')
```

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000

Drop unnecessary columns

```
df.drop(['Name', 'PassengerId'], axis=1, inplace=True)
```

```
df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	0	3	male	34.5	0	0	330911	7.8292	NaN	Q
1	1	3	female	47.0	1	0	363272	7.0000	NaN	S
2	0	2	male	62.0	0	0	240276	9.6875	NaN	Q
3	0	3	male	27.0	0	0	315154	8.6625	NaN	S
4	1	3	female	22.0	1	1	3101298	12.2875	NaN	S

```
df.drop(['Cabin'], inplace=True, axis=1)
df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	0	3	male	34.5	0	0	330911	7.8292	Q
1	1	3	female	47.0	1	0	363272	7.0000	S
2	0	2	male	62.0	0	0	240276	9.6875	Q
3	0	3	male	27.0	0	0	315154	8.6625	S
4	1	3	female	22.0	1	1	3101298	12.2875	S

Finding Bins

```
df.describe()
```

	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
50%	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200

Remove rows with null rows

```
df['Age'].isna().sum()
df.dropna(inplace=True)
```

```
df.isna().sum().sum()
```

```
0
```

Creating bins

```
bins1 = [0,5,12,20,30,60,100]
label1 = ['Infant','child','Teenager','Young Adult','Adult','Elderly']
df['Age Category'] = pd.cut(df['Age'], bins1, labels=label1)
df.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Age Category
0	0	3	male	34.5	0	0	330911	7.8292	Q	Adult
1	1	3	female	47.0	1	0	363272	7.0000	S	Adult
2	0	2	male	62.0	0	0	240276	9.6875	Q	Elderly
3	0	3	male	27.0	0	0	315154	8.6625	S	Young Adult
4	1	3	female	22.0	1	1	3101298	12.2875	S	Young Adult

```
bins2 = [0,200,400,600]
label2 = ['Low', 'Medium', 'High']
df['Fare Category'] = pd.cut(df['Fare'], bins2, labels=label2)
df.tail()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Age Category	Fare Category
409	1	3	female	3.0	1	1	SOTON/O.Q. 3101315	13.775	S	Infant	Low
411	1	1	female	37.0	1	0	19928	90.000	Q	Adult	Low
412	1	3	female	28.0	0	0	347086	7.775	S	Young Adult	Low
414	1	1	female	39.0	0	0	PC 17758	108.900	C	Adult	Low

```
bins2 = [-1,2,4,8]
label3 = ['Low', 'Medium', 'High']
df['Sibsp Category'] = pd.cut(df['SibSp'], bins2, labels=label3)
df.tail()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Age Category	Fare Category	Sibsp Category
409	1	3	female	3.0	1	1	SOTON/O.Q. 3101315	13.775	S	Infant	Low	Low
411	1	1	female	37.0	1	0	19928	90.000	Q	Adult	Low	Low
412	1	3	female	28.0	0	0	347086	7.775	S	Young Adult	Low	Low
414	1	1	female	39.0	0	0	PC 17758	108.900	C	Adult	Low	Low

```
df.drop(['Age', 'SibSp', 'Fare'], inplace=True,axis=1)
```

Final Dataset

```
df.head(10)
```

	Survived	Pclass	Sex	Parch	Ticket	Embarked	Age Category	Fare Category	Sibsp Category
0	0	3	male	0	330911	Q	Adult	Low	Low
1	1	3	female	0	363272	S	Adult	Low	Low
2	0	2	male	0	240276	Q	Elderly	Low	Low
3	0	3	male	0	315154	S	Young Adult	Low	Low
4	1	3	female	1	3101298	S	Young Adult	Low	Low
5	0	3	male	0	7538	S	Teenager	Low	Low
6	1	3	female	0	330972	Q	Young Adult	Low	Low
7	0	2	male	1	248738	S	Young Adult	Low	Low
8	1	3	female	0	2657	C	Teenager	Low	Low
9	0	3	male	0	A/4 48871	S	Young Adult	Low	Low

Find-S Algorithm

```
d = np.array(df)[:,1:]
print("n Attributes are: ",d)

target = np.array(df)[:,0]
print("n Target is: ",target)
```

```
n Attributes are:  [[3 'male' 0 ... 'Adult' 'Low' 'Low']
 [3 'female' 0 ... 'Adult' 'Low' 'Low']
 [2 'male' 0 ... 'Elderly' 'Low' 'Low']
 ...
 [3 'female' 0 ... 'Young Adult' 'Low' 'Low']
 [1 'female' 0 ... 'Adult' 'Low' 'Low']
 [3 'male' 0 ... 'Adult' 'Low' 'Low']]
n Target is:  [0 1 0 0 1 0 1 0 1 0 0 1 0 1 1 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1
 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 0
 1 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 1
 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 1 0 1 0 1 0 0 1 0 0 0 1 1 0 1 1 0 0 1 1 0
 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 1 1 0 1 0 1 0 0 0 0 1 0
 0 0 0 0 1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 1
 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0
 1 0 1 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 0 1 0 1 1 0 1 1 0 1 1 1 0 0 1 0 0 1
 1 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 1 1 1 1 0]
```

```
def train(c,t):
    for i, val in enumerate(t):
        if val == 0:
            specific_hypothesis = c[i].copy()
            break

    for i, val in enumerate(c):
        if t[i] == 0:
            for x in range(len(specific_hypothesis)):
                if val[x] != specific_hypothesis[x]:
                    specific_hypothesis[x] = '?'
                else:
                    pass

    return specific_hypothesis

#obtaining the final hypothesis
print("n The final hypothesis is:",train(d,target))
```

```
n The final hypothesis is: ['?' 'male' '?' '?' '?' '?' '?' '?']
```