

30.4.22

AIM:- Implement and demonstrate the Find-S algorithm for finding the most specific hypothesis based on a given dataset. Read the training data from a csv file or any other file format.

i) Dataset Description :-

- A) Name of dataset :- tennis.csv
- B) Description: This dataset consist of ~~weather~~ weather condition based on few parameters and choose whether someone will go out for playing tennis on a certain weather. This is a binary class dataset. The dataset is available on Kaggle website.
- C) Size :- The dataset consist of 14 rows and 5 columns.
- D) Attributes :- This dataset has 4 features or attributes based on which output is generated. The attributes are -
 - 1) Outlook :- This feature takes 3 values
 - a) Sunny b) Overcast c) Rainy
 - 2) Temp :- This feature takes 3 values
 - a) hot b) cool c) mild
 - 3) Humidity :- This feature takes 2 values
 - a) high b) normal
 - 4) Windy :- This feature takes 2 values
 - a) False b) True

- E) Label: This dataset has a target which is play, it's a categorical binary output which can be either yes or no.
- F) Type of dataset: This is a categorical dataset which has binary output.

2) Find-S algorithm:-

- A) It is a basic concept learning algorithm in machine learning.
- B) The find-S algorithm finds the most specific hypothesis that fits all the positive example. The algorithm only works on the positive training samples of the dataset.

c) Algorithm:-

Step 1: Initialize h to the most specific hypothesis in H

Step 2: for each positive training instance x
for each attribute constraint a , in h
If the constraint a , is satisfied by x
Then do nothing
Else replace a , in h by the next more general constraint that is satisfied by x

Step 3: Output hypothesis h

Surmon Singh

30.4.22

30.4.22

Result analysis:-

The final specific hypothesis for our training dataset is $(?, ?, ?, ?)$ which indicates that the attributes are capable of accepting any value.

Activities

Firefox Web Browser

Sat Apr 30 15:26

Find-S_Algorithm - Jupyter × Fish_Data - Jupyter Noteb × Price - Jupyter Notebook × +

localhost:8888/notebooks/Find-S_Algorithm.ipynb

110%

Trusted | Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help

Run

Markdown

Implement Find-S Algorithm

Load Libraries

```
In [1]: import pandas as pd
```

Load Dataset

```
In [2]: df = pd.read_csv('./tennis.csv')
```

Explore Dataset

```
In [3]: df.head()
```

```
Out[3]:
```

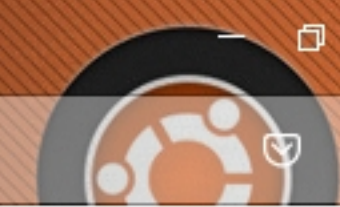
	outlook	temp	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes

```
In [4]: df.shape
```

```
Out[4]: (14, 5)
```

```
In [5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   outlook     14 non-null    object
1   temp        14 non-null    object
```



In [5]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   outlook     14 non-null     object
1   temp        14 non-null     object
2   humidity    14 non-null     object
3   windy       14 non-null     bool
4   play        14 non-null     object
dtypes: bool(1), object(4)
memory usage: 590.0+ bytes
```

In [16]:

```
for i in df.columns:
    print(f'{i} : {df[i].unique()}')

outlook : ['sunny' 'overcast' 'rainy']
temp : ['hot' 'mild' 'cool']
humidity : ['high' 'normal']
windy : [False True]
play : ['no' 'yes']
```

Split Dataset Into Attributes And Target

In [6]: result=df['play'].values

In [7]: attributes=df.drop('play',axis=1).values

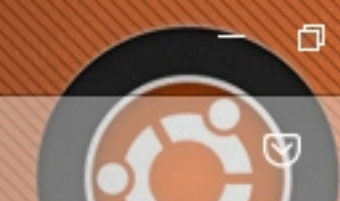
Initialization Of Specific Hypothesis

In [8]: H=['0']*attributes.shape[1]

In [9]: print(f'Initial Hypothesis is : {H}')

Initial Hypothesis is : ['0', '0', '0', '0']

Implement The Logic Of Find-S Algorithm



```
In [9]: print(f'Initial Hypothesis is : {H}')
```

```
Initial Hypothesis is : ['0', '0', '0', '0']
```

Implement The Logic Of Find-S Algorithm

```
In [10]: for i in range(attributes.shape[0]):
          if result[i]=='yes':
              for j in range(attributes.shape[1]):
                  if H[j]=='0':
                      H[j]=attributes[i][j]
                  elif H[j]!=attributes[i][j]:
                      H[j]='?'
          print(f'For Step-{i} : {H}')
```

```
For Step-0 : ['0', '0', '0', '0']
For Step-1 : ['0', '0', '0', '0']
For Step-2 : ['overcast', 'hot', 'high', False]
For Step-3 : ['?', '?', 'high', False]
For Step-4 : ['?', '?', '?', False]
For Step-5 : ['?', '?', '?', False]
For Step-6 : ['?', '?', '?', '?']
For Step-7 : ['?', '?', '?', '?']
For Step-8 : ['?', '?', '?', '?']
For Step-9 : ['?', '?', '?', '?']
For Step-10 : ['?', '?', '?', '?']
For Step-11 : ['?', '?', '?', '?']
For Step-12 : ['?', '?', '?', '?']
For Step-13 : ['?', '?', '?', '?']
```

Final General Hypothesis

```
In [11]: print(f'Final Hypothesis is : {H}')
```

```
Final Hypothesis is : ['?', '?', '?', '?']
```