S-Algorithm

```
import pandas as pd
import numpy as np


df = pd.read_csv('Tesla Deaths.csv')
df.head
```

```
<bound method NDFrame.head of      Case #  Year        Date  Country   State   \
0       294  2022   1/17/2023      USA       CA
1       293  2022    1/7/2023   Canada        -
2       292  2022    1/7/2023      USA       WA
3       291  2022  12/22/2022      USA       GA
4       290  2022  12/19/2022   Canada        -
..      ...   ...         ...       ...     ...
289       5  2014   7/14/2014      USA      CA
290       4  2014    7/4/2014      USA      CA
291       3  2014    7/4/2014      USA      CA
292       2  2013   11/2/2013      USA      CA
293       1  2013    4/2/2013      USA      CA

                           Description    Deaths   Tesla driver   \
0      Tesla crashes into back of semi          1              1
1                        Tesla crashes          1              1
2     Tesla hits pole, catches on fire          1             -
3               Tesla crashes and burns          1              1
4        Tesla crashes into storefront          1             -
..                                 ...       ...            ...
289           Tesla kills motorcyclist          1             -
290         Thief crashes stolen Tesla          1              1
291        Tesla rear ends stopped car          3             -
292                Tesla kills cyclist          1             -
293     Tesla veers into opposite lane          2             -

     Tesla occupant   Other vehicle   Cyclists/ Peds   TSLA+cycl / peds   \
0                 -               -                -                  1
1                 -               -                -                  1
2                 1               -                -                  1
3                 -               -                -                  1
4                 -               -                1                  1
..              ...             ...              ...                ...
289               -               1                -                  -
290               -               -                -                  1
291               -               3                -                  -
292               -               -                1                  1
293               -               2                -                  -

     Model   Autopilot claimed   Verified Tesla Autopilot Deaths  Unnamed: 15  \
0        -                   -                                 -           -
1        -                   -                                 -           -
2        -                   -                                 -           -
3        -                   -                                 -           -
4        -                   -                                 -           -
..     ...                 ...                               ...         ...
289      -                   -                                 -           -
290      -                   -                                 -           -
291      -                   -                                 -           -
292      -                   -                                 -           -
293      S                   -                                 -           -

     Autopilot deaths
0                   0
1                   0
2                   0
3                   0
4                   0
```

```
d = np.array(df) [:,:-1]
print(" The attributes are: ",d)
```

```
 The attributes are:  [[294 2022 '1/17/2023' ... ' - ' ' - ' ' - ']
 [293 2022 '1/7/2023' ... ' - ' ' - ' ' - ']
 [292 2022 '1/7/2023' ... ' - ' ' - ' ' - ']
 ...
 [3 2014 '7/4/2014' ... ' - ' ' - ' ' - ']
 [2 2013 '11/2/2013' ... ' - ' ' - ' ' - ']
 [1 2013 '4/2/2013' ... ' - ' ' - ' ' - ']]
```

```
target = np.array(df) [:, -1]
print("The target is: ",target)
```

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

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

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