```
#!/usr/bin/env python
# coding: utf-8

# In[77]:


## import pandas as pd
from bs4 import BeautifulSoup
from html_table_extractor.extractor import Extractor
import numpy as np
import numpy as np; np.random.seed(0)
import matplotlib.pyplot as plt
import seaborn as sns; sns.set_theme()
import csv
from collections import defaultdict

real = []

with open('nsltp_names_goood.csv', newline = "") as f:
    reader = csv.reader(f,  quotechar='|')
    real = list(reader)
    #print(real)

with open("C:\crossreactivity\sias_nsltp_clean.html", 'r') as f:
    sias_scam = f.read()
    soup = BeautifulSoup(sias_scam, 'lxml')

all_tables = soup.find_all('table')


# In[78]:


def scores(indx):
    table = all_tables[indx]
    #print(table)
    extractor = Extractor(table)
    extractor.parse()
    sias_list = extractor.return_list()
    sias_list = sias_list[:-3]
    sias_len = len(sias_list)
    sias_names = sias_list[sias_len - 1]
    sias_names = sias_names[1:]
    #print(sias_names)

    num_item = len(sias_names)
    #print(num_item)

    sias_final =  [[0] * num_item for _ in range(num_item)]

    for i in range(sias_len-1):
        row = sias_list[i]
        row = row[1:]
        #print(row)
        for j in range(len(row)):
            #rint(row[j])
            sias_final[i][j] = row[j]
    return sias_final


# In[79]:


identity = scores(1)
print(np.matrix(identity))

similarity = scores(4)
print(np.matrix(similarity))


# In[80]:


```

```
73
74 def percent(matrix):
75     num_final = [[0] * len(matrix) for _ in range(len(matrix))]
76     for i in range(len(matrix)):
77         for j in range(i+1):
78             num = matrix[i][j].strip("%")
79             num = float(num)
80             num_final[i][j] = num
81     return num_final
82
83 iden_final = percent(identity)
84 simi_final = percent(similarity)
85
86
87 arisc_final = [[0] * len(identity) for _ in range(len(identity))]
88
89
90
91
92 for i in range(len(iden_final)):
93         for j in range(i+1):
94             ide = iden_final[i][j]
95             sim = simi_final[i][j]
96             avg = (ide+sim)/2
97             arisc_final[i][j] = avg
98
99 arisc_matrix = np.matrix(arisc_final)
100
101 ax = sns.heatmap(arisc_matrix, cmap= "BuPu")
102
103 plt.show()
104
105
106 # In[81]:
107
108
109 for i in range(len(arisc_final)):
110     for j in range(len(arisc_final)):
111         arisc_final[i][j] = round(arisc_final[j][i], 4)
112
113 np.matrix(arisc_final)
114
115
116 # In[82]:
117
118
119 ptn = defaultdict(list)
120 ntp = defaultdict(list)
121 for k, v in real:
122     ptn[k]= v
123     ntp[v]= k
124
125 ptn
126
127
128 # In[83]:
129
130
131
132 names = all_tables[1]
133 extractor = Extractor(names)
134 extractor.parse()
135 names_list = extractor.return_list()
136 names_list = names_list[:-4]
137 len_names= len(names_list)
138
139 #print(names_list)
140
141 id_names = [[" "] * 2 for _ in range(len(names_list))]
142
143 for i in range(len(names_list)):
144     pname = names_list[i][0]
```

```
    if "|" in pname:
        pname = pname.split("|")[1]
    stripped_name = pname.split(".")[0];
    id_names[i][0] = stripped_name
    id_names[i][1] = ptn[stripped_name]

len(id_names)


# In[84]:


import mysql.connector
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="Iatwbofm21!",
  database="crossreactivity"
)

mycursor = mydb.cursor()
tbl_name = "crossreactivity_nsltp"
header = ""

for i in range(len(id_names)-1):
    header = header + id_names[i][0] + " FLOAT, "
header = header + id_names[len(id_names)-1][0] + " FLOAT"

#print(header)
query = "CREATE TABLE IF NOT EXISTS %s (name VARCHAR(255), origin VARCHAR(255), %s)"
% (tbl_name, header)
print(query)
mycursor.execute(query)




# In[85]:


query4 = "CREATE UNIQUE INDEX name_idx ON %s (name)" % (tbl_name)
mycursor.execute(query4)


# In[86]:


query5 = "CREATE UNIQUE INDEX origin_idx ON %s (origin, name)" % (tbl_name)
mycursor.execute(query5)


# In[87]:


column ="name, origin"

for i in range(len(id_names)):
    column = column + ", " + id_names[i][0]

for i in range(len(arisc_final)):
    print(i)
    row = str(arisc_final[i]).replace("[", "").replace("]", "")
    #print(id_names[i][1])
    query2 = "INSERT INTO %s (%s) VALUES ('%s', '%s', %s)" % (tbl_name, column,
id_names[i][0], id_names[i][1], row)
    print(query2)
    mycursor.execute(query2)


mydb.commit()
```

```
# In[88]:


mydb.close()


# In[ ]:




# In[ ]:




# In[ ]:


```