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

# In[105]:


# !pip3 install requests
import requests
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')


# In[106]:


df_top = pd.read_csv('./step0-topsites.csv', names =['rank', 'site'])
df_ran = pd.read_csv('./step0-randomsample.csv', names =['rank', 'site'])


# In[107]:


df_testing = df_top[:10]
df_testing.info()


# In[118]:


def check_access(args):
    site = args[1]
    website_sec = 'https://' + site
    website_dir = 'http://' + site

    # FOR SECURE HTTPS
    try:
        r1 = requests.get(website_sec, timeout = 5, allow_redirects = True)
        sc1 = r1.status_code
        url1 = r1.url

        # HTTPS worked means sc1 == 200
        # HTTPS redirected means sc1 == 301 or 302 (happens when r.url !=
website_sec)


    except Exception as e:
        # HTTPS didnt work
        sc1 = None


    # FOR DIRECT HTTP
    try:
        r2 = requests.get(website_dir, timeout = 5, allow_redirects = True)
        sc2 = r2.status_code
        url2 = r2.url

        # HTTP worked means sc2 == 200
        # HTTP redirected means sc2 == 301 or 302 (happens when r2.url !=
website_dir)

    except Exception as e:
        sc2 = None


    # sc1 -> for HTTPS
    # sc2 -> for HTTP

    print(sc1, sc2, website_dir)

    # Both accessible without redirections
    if sc2 == 200 and sc1 == 200:
        return 'b'

    # Neither accessible
    elif not sc2 and not sc1:
        return 'd'

    # Only HTTPS accessible (HTTPS gives 200, but HTTP redirects)
    elif (sc2 in [301, 302] or r2.url != website_dir or not sc2) and sc1 == 200:
        return 'a'

    # Only HTTP accessible (HTTPS gave error)
    elif sc2 == 200 and not sc1:
        return 'c'

    else:
        return ''




# In[121]:


print("------------------------------------------------------")
df_ran['category'] = df_ran.apply(check_access, axis = 1)
df_top['category'] = df_top.apply(check_access, axis = 1)

# In[127]:


np.unique(df_ran.category, return_counts = True)


# In[125]:


df_ran['category'].replace('', 'd', inplace=True)
df_top['category'].replace('', 'd', inplace=True)

# In[128]:


df_ran.to_csv('./step1-randomsample.csv', index=False)
df_top.to_csv('./step1-topsites.csv', index=False)
```