# Webscraping in Python

# from selenium import webdriver

# from selenium.webdriver.chrome.options import Options

# from selenium.webdriver.common.action\_chains import ActionChains

# import time

# import random

# driver = webdriver.Chrome(executable\_path='driver/chromedriver')

#creates an excel file

import pandas as pd

from lxml import html

import requests

import re

df = pd.read\_csv('state-investor-dispute-timlove.csv')

url = df['URL\_CASE'][578]

page = requests.get(url)

tree = html.fromstring(page.content)

print(url)

https://investmentpolicy.unctad.org/investment-dispute-settlement/cases/405/zamora-gold-v-ecuador

#rips the following information from the UNCTAD website

def GetMetaData(url):

page = requests.get(url)

tree = html.fromstring(page.content)

meta\_data = {'respondent\_state':tree.xpath('//\*[@id="parties-content"]/div[1]/div/text()'),

'home\_state': tree.xpath('//\*[@id="parties-content"]/div[2]/div/text()'),

'details': tree.xpath('//\*[@id="Summary-content"]/div[1]/div/text()'),

'summary': tree.xpath('//\*[@id="summary-content"]/div[2]/div/text()'),

'sector': tree.xpath('//\*[@id="economic-sector-content"]/div[1]/div/text()'),

'subsector1':tree.xpath('//\*[@id="economic-sector-content"]/div[2]/div/text()[1]'),

'subsector2':tree.xpath('//\*[@id="economic-sector-content"]/div[2]/div/text()[2]'),

'arbitration\_rules':tree.xpath('//\*[@id="rules-institution-content"]/div[1]/div/text()'),

'adminstering\_institution':tree.xpath('//\*[@id="rules-institution-content"]/div[2]/div/text()'),

'president':tree.xpath('//\*[@id="tribunal-content"]/div[1]/div/text()'),

'arbitrator\_claimant':tree.xpath('//\*[@id="tribunal-content"]/div[2]/div/text()'),

'arbitrator\_respondent':tree.xpath('//\*[@id="tribunal-content"]/div[3]/div/text()'),

'outcome':tree.xpath('//\*[@id="status-content"]/div/text()'),

'claimed\_by\_investor': tree.xpath('//\*[@id="amounts-content"]/div[1]/div/text()'),

'awarded\_by\_tribunal': tree.xpath('//\*[@id="amounts-content"]/div[2]/div/text()'),

'IIA\_breaches\_alleged1': tree.xpath('//\*[@id="breaches-content"]/div[1]/div/text()[1]'),

'IIA\_breaches\_alleged2': tree.xpath('//\*[@id="breaches-content"]/div[1]/div/text()[2]'),

'IIA\_breaches\_alleged3': tree.xpath('//\*[@id="breaches-content"]/div[1]/div/text()[3]'),

'IIA\_breaches\_alleged4': tree.xpath('//\*[@id="breaches-content"]/div[1]/div/text()[4]'),

'IIA\_breaches\_alleged5': tree.xpath('//\*[@id="breaches-content"]/div[1]/div/text()[5]'),

'IIA\_breaches\_found1': tree.xpath('//\*[@id="breaches-content"]/div[2]/div/text()[1]'),

'IIA\_breaches\_found2': tree.xpath('//\*[@id="breaches-content"]/div[2]/div/text()[2]'),

'IIA\_breaches\_found3': tree.xpath('//\*[@id="breaches-content"]/div[2]/div/text()[3]'),

'IIA\_breaches\_found4': tree.xpath('//\*[@id="breaches-content"]/div[2]/div/text()[4]'),

'IIA\_breaches\_found5': tree.xpath('//\*[@id="breaches-content"]/div[2]/div/text()[5]'),

'type\_of\_follow\_up': tree.xpath('//\*[@id="follow-ups-content"]/div[1]/div/text()'),

'follow\_up\_status': tree.xpath('//\*[@id="follow-ups-content"]/div[3]/div/text()')

}

return meta\_data

df['meta\_data'] = df['URL\_CASE'].apply(lambda x: GetMetaData(x))

df = pd.concat([df.drop(['meta\_data'], axis=1), df['meta\_data'].apply(pd.Series)], axis=1)

copy = df

clean\_columns = df.columns[8:]

def Cleaner(text):

try:

if type(text) == list:

text = text[0]

except:

text = str(text)

pass

removers = ['\r','\n','[',']']

for remove in removers:

text = text.replace(remove,'')

return text.strip()

for col in clean\_columns:

try:

df[col] = df[col].apply(lambda x: Cleaner(x))

except Exception as e:

print(e)

print(col)

df['outcome'] = df['outcome'].apply(lambda x: x.strip())

df.to\_csv('state-investor-dispute.csv')