**Assignment on Platform Research**

**Task #1: Scrapping from TikTok**

Task 1 in is all about creating a python script to scrape tiktok profile data. Written carefully in python considering edge cases. Besides the flexibility of scraping 100 profile by changing the main url from where we want to gather data.

**Methodology:**

First step in this process was selecting a suitable library for the task. I started with the Beautifulsoup library , but faced a few problems, like the same Scrapy library, maybe just because of a few internal problems. However, I overcame this problem by using the Selenium library to get the source code of the site with the help of chrome browser and driver. And to make this happen created an account. Through which I got the urls of the suggested profile and later scraped data from those urls. And this url list can be changed to get the data of the desired profile. Further main task started from loading necessary libraries, setting up the web browser, driver and creating driver for my profile page ended up with suggested profile urls, file creates the python script to scrape the data storing in a csv file.

Step by step technical solutions:

Step 1:

It Started with Importing necessary library.

import os

import re

import numpy as np

import pandas as pd

from selenium import webdriver

from bs4 import BeautifulSoup

import time

import csv

import urllib.request

import selenium

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from webdriver\_manager.chrome import ChromeDriverManager

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

Step 2:

Setting up the web driver and initializes the driver(Chrome Browser)

options = webdriver.ChromeOptions()

options.add\_experimental\_option("excludeSwitches", ["enable-logging"])

options.add\_argument('--headless')

options.add\_argument('--disable-gpu')

#THIS INITIALIZES THE DRIVER (AKA THE WEB BROWSER)

path = Service(r'C:\Users\Sudhakor\Scrapy\chromedriver.exe')

driver = webdriver.Chrome(options=options, service = path)

Step 3:

Generating the source code for the main page of the site in the profile section and this URL can be changed to get the desired data. Here Green color code is defining the main page

driver.get("https://www.tiktok.com/")

source\_code = driver.page\_source

def url\_body(source\_code):

soup = BeautifulSoup(source\_code, 'html.parser')

url\_body = soup.find\_all('div', class\_ = 'tiktok-1mo2fkg-DivUserLinkContainer e797se20')

return url\_body

body = url\_body(source\_code)

Step 4.

Grabbing all the available url and from the main body part and store in url\_list.

body = url\_body(source\_code)

url\_list = []

def url\_generator(body):

for i in body:

url = 'https://www.tiktok.com' + i.find('a')['href']

url\_list.append(url)

url\_generator(body)

Step: 5

Then comes the main function which will scrape all the required data from a url. And this portion will handle all the necessary adjustments. Set the post limit will be 50 for each and set None otherwise(empty). And Converted the engagement number into thousand(K) standard, calculate average by dividing average by the total numbers of post available in the profile must be less then 50. And ended up returning the fields.

def finding\_content(url):

driver.get(url)

source\_code = driver.page\_source

soup = BeautifulSoup(source\_code, 'html.parser')

name = soup.find('h2', class\_ = "ekmpd5l5").text.strip()

about = soup.find('h2', class\_ = 'e1457k4r3').text

about = re.sub('[^a-zA-Z0-9]', ' ', about)

about = re.sub(' +', ' ', about).strip()

img\_link = soup.find('img', class\_ = 'e1e9er4e1')['src']

following = soup.find('strong', title = 'Following').text

followers = soup.find('strong', title = 'Followers').text

likes = soup.find('strong', title = 'Likes').text

all\_posts = []

engagement = 0

posts\_main = []

try:

posts\_main = soup.find\_all('div', class\_ = 'e19c29qe7')

if len(posts\_main)> 50:

posts\_main = posts\_main[:50]

else:

pass

for post in posts\_main:

post\_link = post.find('a')['href']

views = post.find('strong', class\_ = 'video-count tiktok-1nb981f-StrongVideoCount e148ts222').text

if views.endswith('M'):

views = views.split('M')[0]

try:

fract = views.split('.')[1]

le =(3-len(fract))

zero = ''

for i in np.zeros(le):

zero += '0'

views = (views + zero).replace('.', '')

except:

views = str(views) + ('000')

elif views.endswith('B'):

try:

fract = views.split('.')[1]

le =(6-len(fract))

zero = ''

for i in np.zeros(le):

zero += '0'

views = (views + zero).replace('.', '')

except:

views = str(views) + ('000000')

elif views.endswith('K'):

views = views.split('K')[0]

try:

views = views.replace('.', '')

except:

pass

else:

views = int(views)/1000

engagement+=int(views)

all\_posts.append(post\_link)

except:

all\_posts = []

average\_engagement = str(np.round(engagement/len(posts\_main),2))+'K'

return name, img\_link, about, likes, followers, following, all\_posts, average\_engagement

**Step 6:**

In this last iterate through all the url from the url list, store the scraped values onwards in a dictionary. Last but not the least, converted the dictionary into pandas dataframe and proceeded into a csv file.

profile\_list = []

for url in url\_list:

name, img\_link, about, likes, followers, following, all\_posts, average\_engagement = finding\_content(url)

profile = {'Username': name,

'Image':img\_link,

'About':about,

'Likes': likes,

'Followers' : followers,

'Following' : following,

'Posts' : all\_posts,

'Average engagement' : average\_engagement

}

print(profile)

profile\_list.append(profile)

print('Saving profile: ',name)

df = pd.DataFrame(profile\_list)

df.to\_csv('TIKTOK\_DATA.csv', index = False)

Dataset Link: <https://github.com/Sudhakordas/ActiveFence-assignmetn/blob/master/TIKTOK_DATA.csv>