## Indian Institute of Information Technology Surat

****

# Lab Report on

# Natural Language Processing (CS 601) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Mrs. Nidhi Desai

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Jan-2024**

## Lab No: 1

**Aim:**

Data Collection from E-Commerce, Twitter and Similar Platforms

**Description:**

Write a Python script for:

(a) Collecting tweets that may incorporate owner, date of post, number of retweet, number

of followers, no of followers , and other associated information from Twitter and store it

into a .csv file. (The size of collected tweets >5000)

(b) To scrap users reviews from any E-commerce or similar portals (Ex- Amazon, Flipkart, Yelp) and store it into a csv file that may incorporate date of post, number of likes/dislikes, reviews, location, and other associated fields (The size of collected reviews>5000).

## Source Code:

**For Task (a):**

# ---------------------------------------------------------------------------------

from selenium import webdriver

from selenium.webdriver.common.by import By

from fake\_useragent import UserAgent

from webdriver\_manager.firefox import GeckoDriverManager

import time

import json

import os

from selenium.webdriver.common.keys import Keys

MY\_USERNAME\_VAR = os.getenv('USERNAME')

MY\_PASS\_VAR = os.getenv('PASS')

def wait\_for\_window(self, timeout = 2):

time.sleep(round(timeout / 1000))

wh\_now = self.driver.window\_handles

wh\_then = self.vars["window\_handles"]

if len(wh\_now) > len(wh\_then):

return set(wh\_now).difference(set(wh\_then)).pop()

keywords = ["WWE","Rock","RomanReigns"]

ulrs = []

options = webdriver.FirefoxOptions()

options.headless = False

ua = UserAgent()

userAgent = ua.random

options.add\_argument(f'user-agent={userAgent}')

driver = webdriver.Firefox(executable\_path=GeckoDriverManager().install(),options=options)

driver.get("https://twitter.com/i/flow/login")

driver.maximize\_window()

time.sleep(10)

try:

input\_element = driver.find\_element(By.CSS\_SELECTOR, '.r-30o5oe.r-1niwhzg.r-17gur6a.r-1yadl64.r-deolkf.r-homxoj.r-poiln3')

input\_element.click()

time.sleep(5)

password\_x = driver.find\_element(By.CSS\_SELECTOR, '.r-30o5oe.r-1niwhzg.r-17gur6a.r-1yadl64.r-deolkf.r-homxoj.r-poiln3.r-7cikom.r-1ny4l3l.r-t60dpp.r-1dz5y72.r-fdjqy7.r-13qz1uu')

password\_x.click()

password\_x.send\_keys(MY\_PASS\_VAR)

time.sleep(5)

with open('keyword\_numbers.json', 'w') as file:

json.dump(keyword\_numbers, file)

except Exception as e:

print(ulrs)

print("An error occurred:", str(e))

**For Task (b):**

import csv

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def extract\_reviews(product\_url, num\_reviews\_to\_scrape=10):

driver = webdriver.Chrome()

driver.get(product\_url)

time.sleep(8)

reviews = []

review\_elements = driver.find\_elements(By.CSS\_SELECTOR, '.a-section.review')

temp\_Date = ""

for review\_element in review\_elements[:num\_reviews\_to\_scrape]:

time.sleep(1)

review = {}

review['author'] = review\_element.find\_element(By.CSS\_SELECTOR, '.a-profile-name').text.strip()

temp\_Date = review\_element.find\_element(By.CSS\_SELECTOR, '.review-date').text.strip()

review['date'] = temp\_Date[temp\_Date.find('on')+3:]

review['location'] = temp\_Date[12:temp\_Date.find('on')-1]

review['text'] = review\_element.find\_element(By.CSS\_SELECTOR, '.review-text-content').text.strip()

review['rating'] = review\_element.find\_element\_by\_xpath('//i[@data-hook="review-star-rating"]').text.strip()

review['title'] = review\_element.find\_element(By.CSS\_SELECTOR, '.review-title').text.strip()

reviews.append(review)

print(review)

driver.quit()

return reviews

product\_url = 'https://www.amazon.in/ZAPCASE-Compatible-Xiaomi-Covers-Carbon/product-reviews/B07GQY2RN2/ref=cm\_cr\_arp\_d\_paging\_btm\_next\_2?ie=UTF8&reviewerType=all\_reviews'

reviews\_data = []

for i in range(1,4):

reviews\_data += extract\_reviews(product\_url+'&pageNumber='+str(i), num\_reviews\_to\_scrape=10)

def export\_csv(reviews, csv\_filename='reviews\_data.csv'):

with open(csv\_filename, 'w', newline='', encoding='utf-8') as csv\_file:

fieldnames = ['date','names','location','reviewtitles','ratings','reviews']

writer = csv.DictWriter(csv\_file, fieldnames=fieldnames)

writer.writeheader()

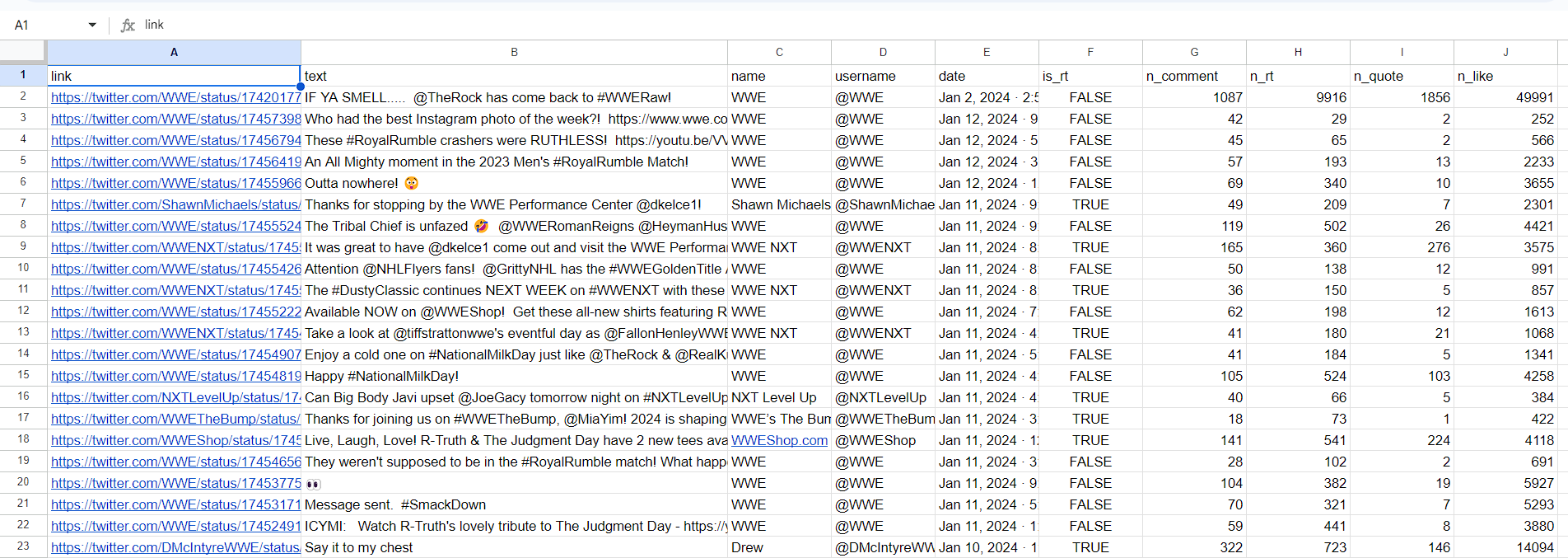
for review in reviews:

writer.writerow({'date': review['date'], 'names': review['author'], 'location': review['location'], 'reviewtitles': review['title'], 'ratings': review['rating'], 'reviews': review['text']})

export\_csv(reviews\_data)

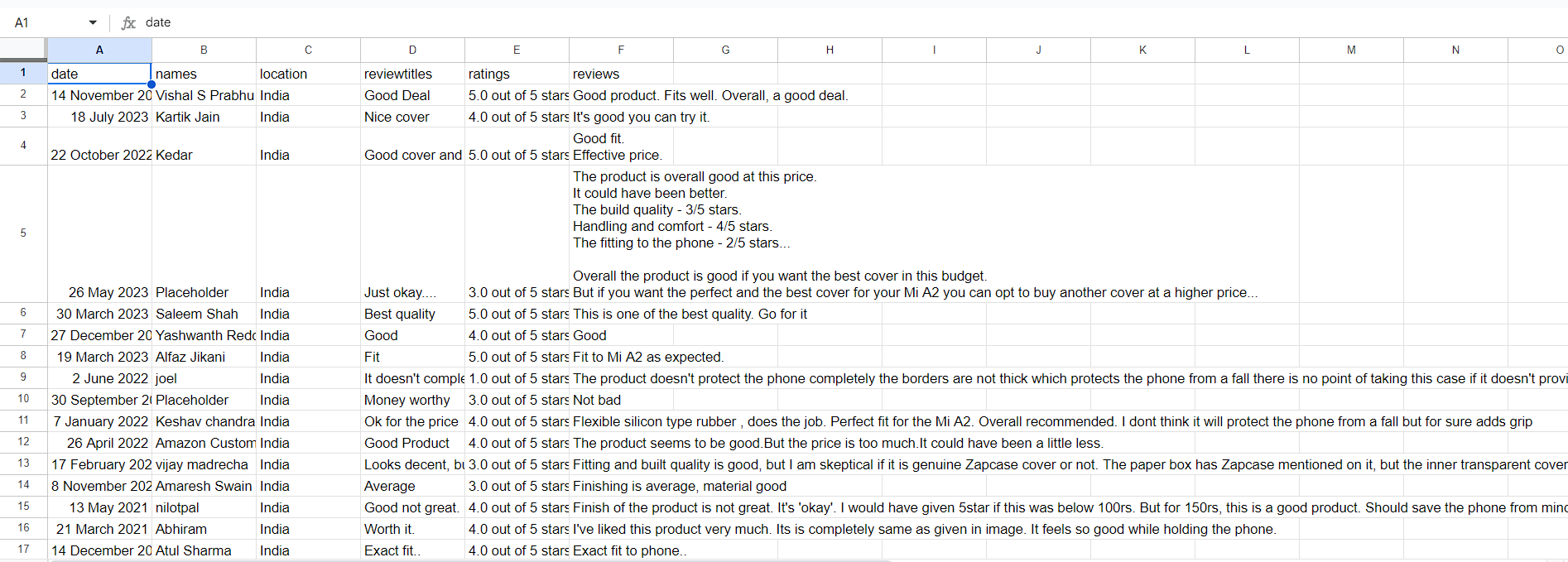
## Output:

**For Task (a):**

****

****

**For Task (b):**

****

****

## Conclusion:

* Efficient and direct access to Twitter's data through the API.
* Provides real-time data retrieval, enabling instant updates.
* Offers structured data in JSON format for easy processing.
* Overcomes API limitations for certain tasks, such as scraping dynamic content using custom scraping.