# **Indian Institute of Information Technology Surat**



# Lab Report on Machine Learning (CS 601) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty** 

Dr. Pradeep Kumar Roy Dr. Rajesh K. Ahir

Department of Computer Science and Engineering Indian Institute of Information Technology Surat Gujarat-394190, India

Jan-2024

# **Table of Contents**

Exp. No	Name of the Experiments	Page no	Date of Experiment	Date of Submission
1	Data Collection from E-Commerce, Twitter and Similar Platforms	3-9	03-01-2024	12-01-2024
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				

#### 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 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-1yad164.r-deolkf.r-homxoj.r-poiln3.r-7cikom.r-1ny41
31.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.st
rip()
        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/B07GQ
Y2RN2/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):

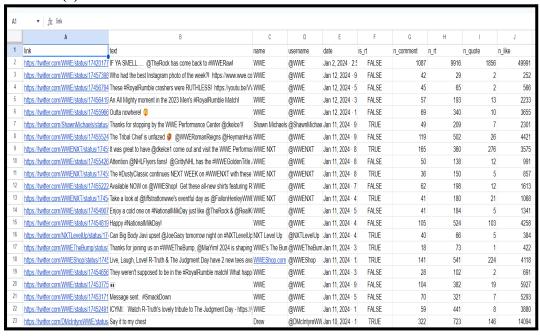


Figure 1.1 Output for Twitter Data Collection

#### For Task (b):

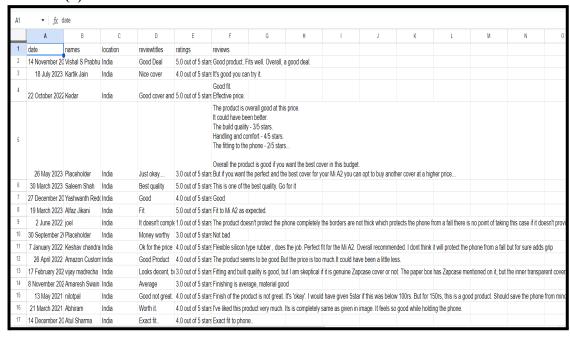


Figure 1.2 Output for Amazon Review Scraping

## **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.