LAB MANUAL
PART A
(PART A : TO BE REFFERED BY STUDENTS)

Experiment No-02

A.1 Aim:

Data Collection-Select the social media platforms of your choice (Twitter, Facebook, LinkedIn, YouTube, Web blogs etc), connect to and capture social media data for business (scraping, crawling, parsing).

Lab Objective	To understand the fundamental concepts of social media networks
Lab Outcome	Collect, monitor , store and track social media data

A-2 Prerequisite

Data Mining, Data Analytic

A.3 OutCome

Students will able to Collect, monitor, store and track social media data

A.4 Theory:

Having quality data in the proper format is usually more than half of the battle. For those who can gain direct access to a well-maintained customer database, the data collection and preparation process is relatively painless. However, for researchers who want to study text information that exists in a public forum such as FlyerTalk.com, data collection can be more complex and usually involves web scraping.

Web scraping (or screen scraping) is a technique used to extract data from websites that display output generated from another program. There are many commercially available applications that can scrape a website and turn the blogs or forum messages into a data table.

Web Scraping Process

Crawl

• Crawl the website and scrape for topic, ID and thread initiator.

Download

• Use topic ID from the first step as part of the URL query string to download messages.

Store

- Web crawl and store message display pages.

Screen Scrape

- Screen scrape stored web pages and extract data into a structured format.

Access Facebook data using Graph API

The Graph API is the primary way to get data into and out of the Facebook platform. It's an HTTP-based API that apps can use to programmatically query data, post new stories, manage ads, upload photos, and perform a wide variety of other tasks. The Graph API is named after the idea of a "social graph" — a representation of the information on Facebook. It's composed of nodes, edges, and fields. Typically you use nodes to get data about a specific object, use edges to get collections of objects on a single object, and use fields to get data about a single object or each object in a collection.

Link

- Link extracted posts with topics from the first step, along with other extracted fields to create the final dataset

Extraction of Tweets using Tweepy

Twitter is a popular social network where users share messages called tweets. Twitter allows us to mine the data of any user using Twitter API or Tweepy. The data will be tweets extracted from the user. The first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication.

Steps to obtain keys:

– Login to twitter developer section

– Go to “Create an App”

– Fill the details of the application.

– Click on Create your Twitter Application

– Details of your new app will be shown along with consumer key and consumer secret.

– For access token, click” Create my access token”. The page will refresh and generate access token.

Tweepy is one of the library that should be installed using pip. Now in order to authorize our app to access Twitter on our behalf, we need to use the OAuth Interface. Tweepy provides the convenient Cursor interface to iterate through different types of objects. Twitter allows a maximum of 3200 tweets for extraction.

These all are the prerequisite that have to be used before getting tweets of a user.

PART B
(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

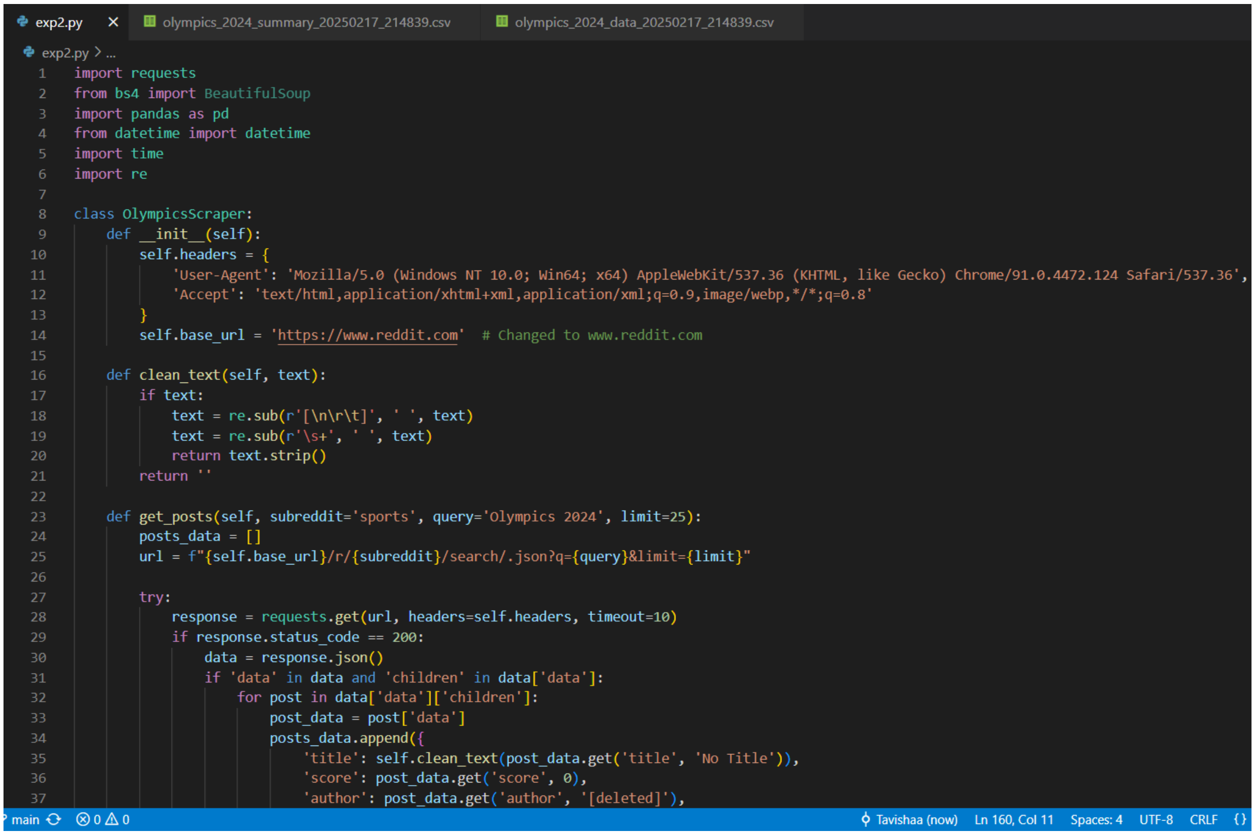
Roll. No.: C5	Name: Tavishaa Jaiswal
Class: BE-Comps	Batch: C1
Date of Experiment:	Date of Submission:
Grade:	

B.1.Study the fundamentals of social media platform and social media tools:

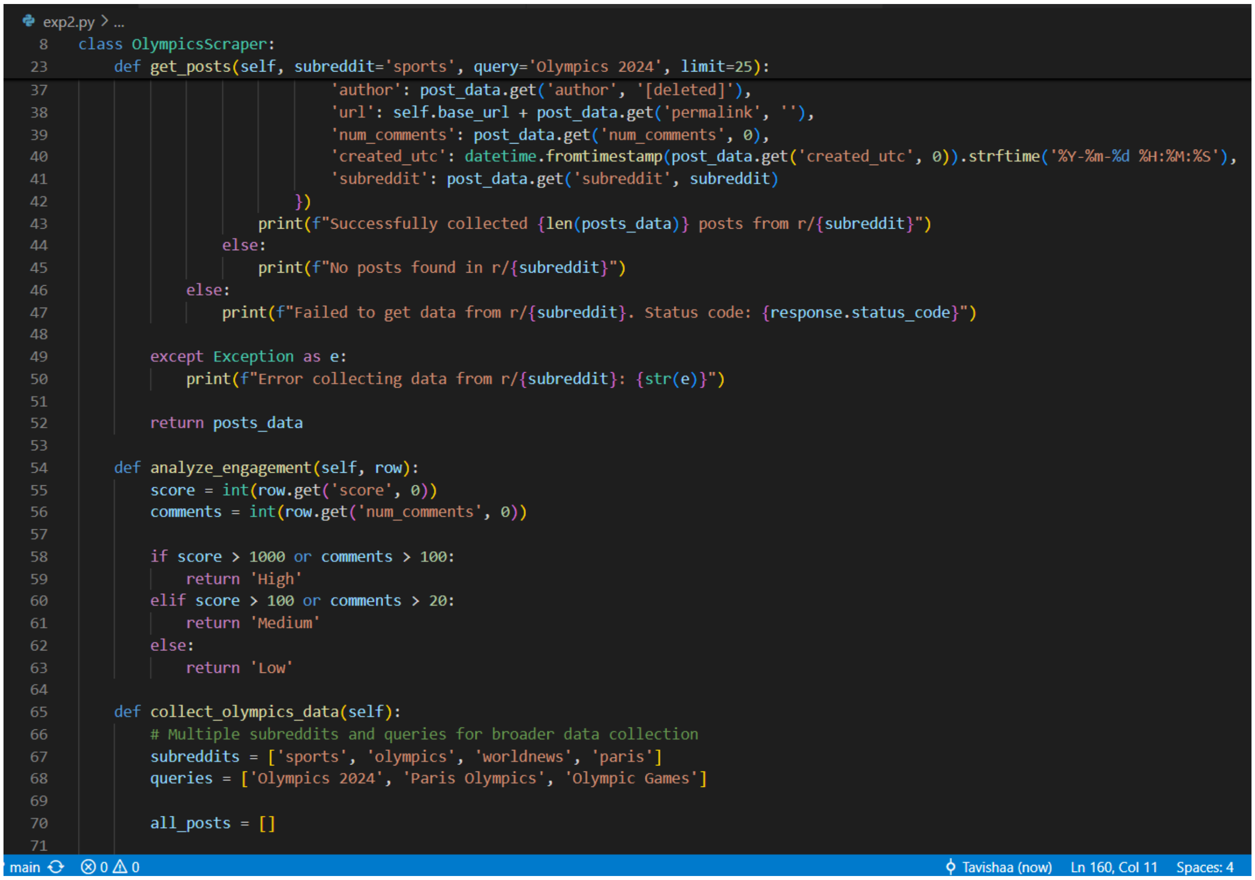
The fundamentals of social media platforms like Reddit involve users sharing content, engaging through upvotes and comments, and forming communities (subreddits). Tools such as web scrapers and APIs allow for automated data collection from these platforms, enabling the analysis of engagement and trends. The script exemplifies how these tools can be used to gather and categorize discussions around the 2024 Olympics, providing valuable insights into public sentiment and popular topics.

B.2 Input and Output:

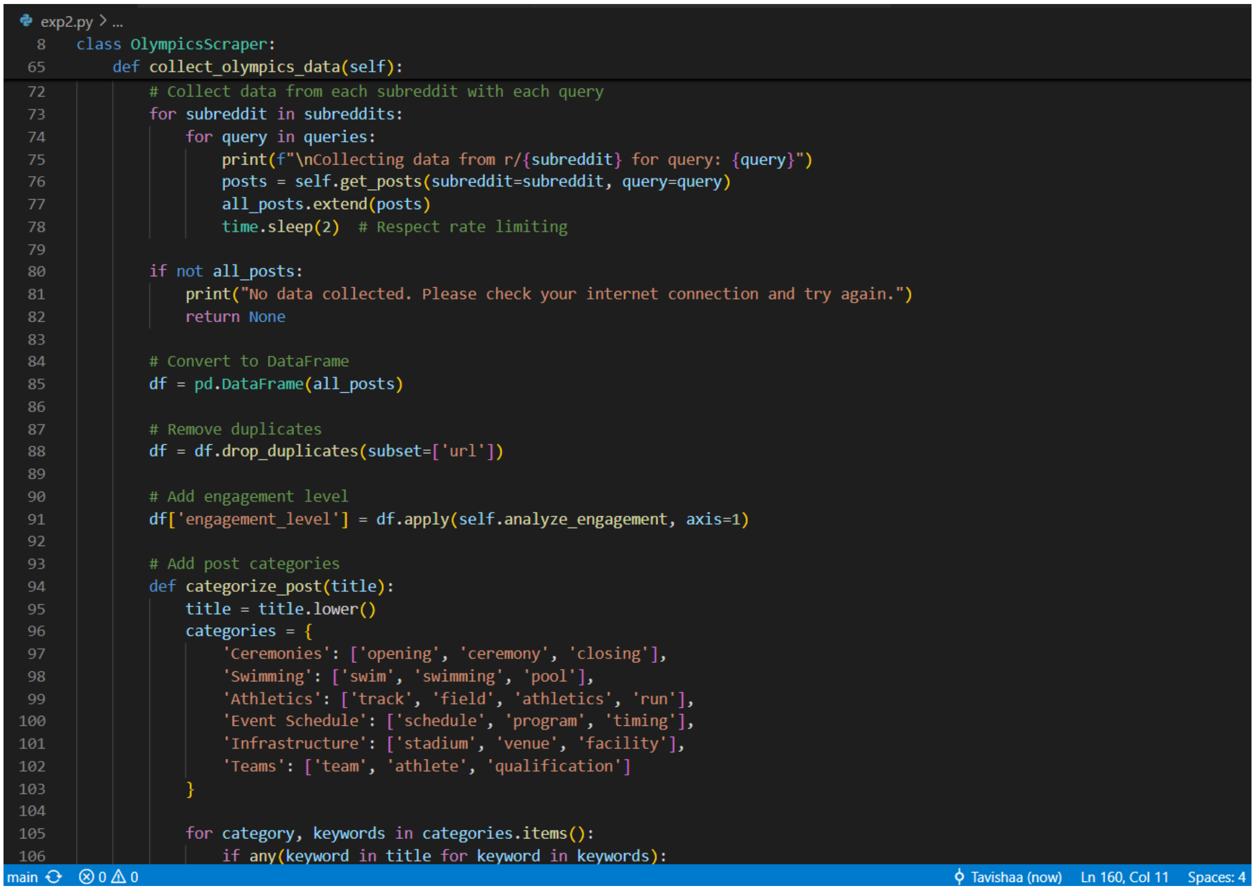
INPUT:


exp2.py
olympics_2024_summary_20250217_214839.csv
olympics_2024_data_20250217_214839.csv
exp2.py > ...
main
0 0
Tavishaa (now)
Ln 160, Col 11
Spaces: 4
UTF-8
CRLF
{}

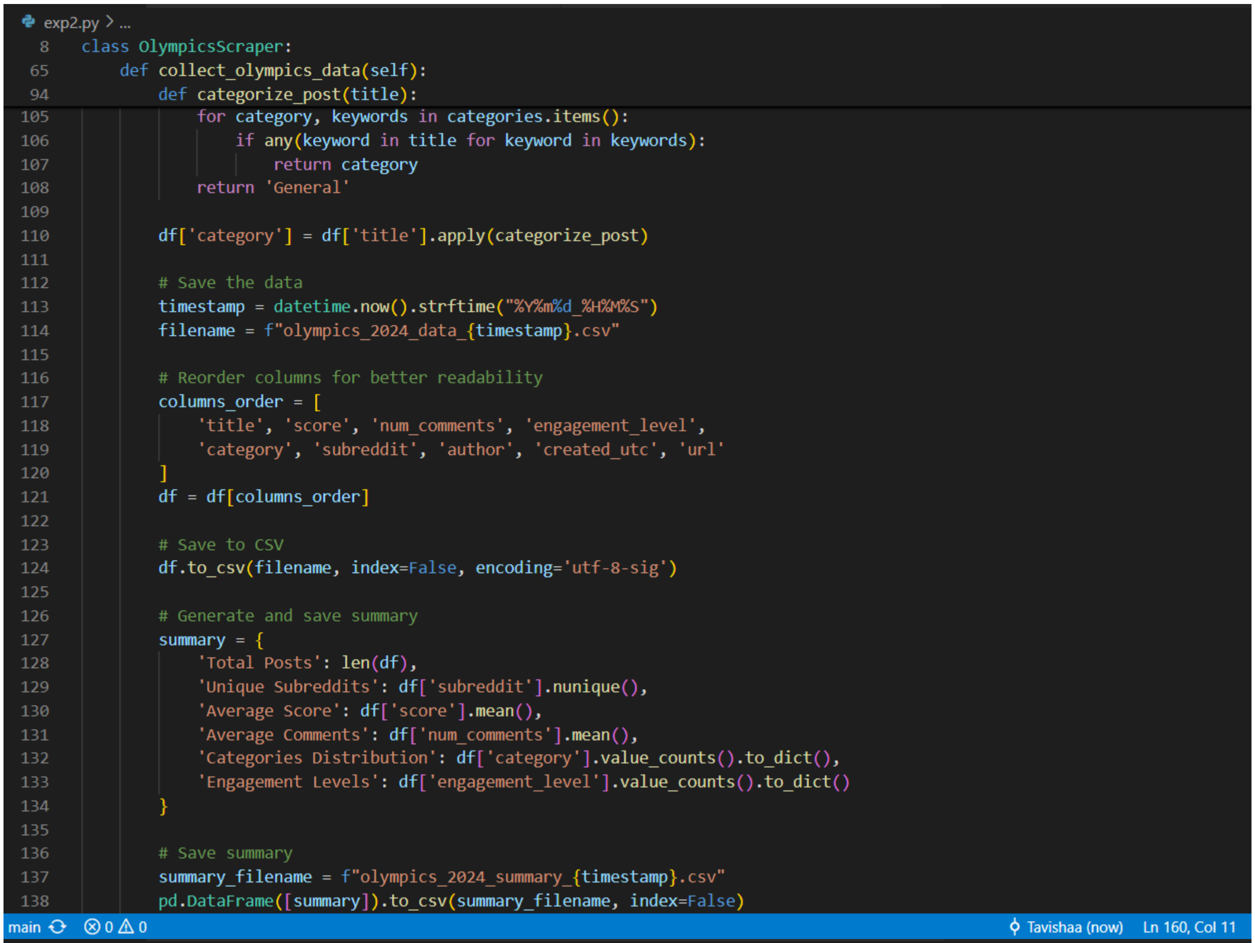
```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
from datetime import datetime
import time
import re

class OlympicsScraper:
    def __init__(self):
        self.headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36',
            'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8'
        }
        self.base_url = 'https://www.reddit.com'  # Changed to www.reddit.com

    def clean_text(self, text):
        if text:
            text = re.sub(r'[\n\r\t]', ' ', text)
            text = re.sub(r'\s+', ' ', text)
            return text.strip()
        return ''

    def get_posts(self, subreddit='sports', query='Olympics 2024', limit=25):
        posts_data = []
        url = f"{self.base_url}/r/{subreddit}/search/.json?q={query}&limit={limit}"

        try:
            response = requests.get(url, headers=self.headers, timeout=10)
            if response.status_code == 200:
                data = response.json()
                if 'data' in data and 'children' in data['data']:
                    for post in data['data']['children']:
                        post_data = post['data']
                        posts_data.append({
                            'title': self.clean_text(post_data.get('title', 'No Title')),
                            'score': post_data.get('score', 0),
                            'author': post_data.get('author', '[deleted]'),
```



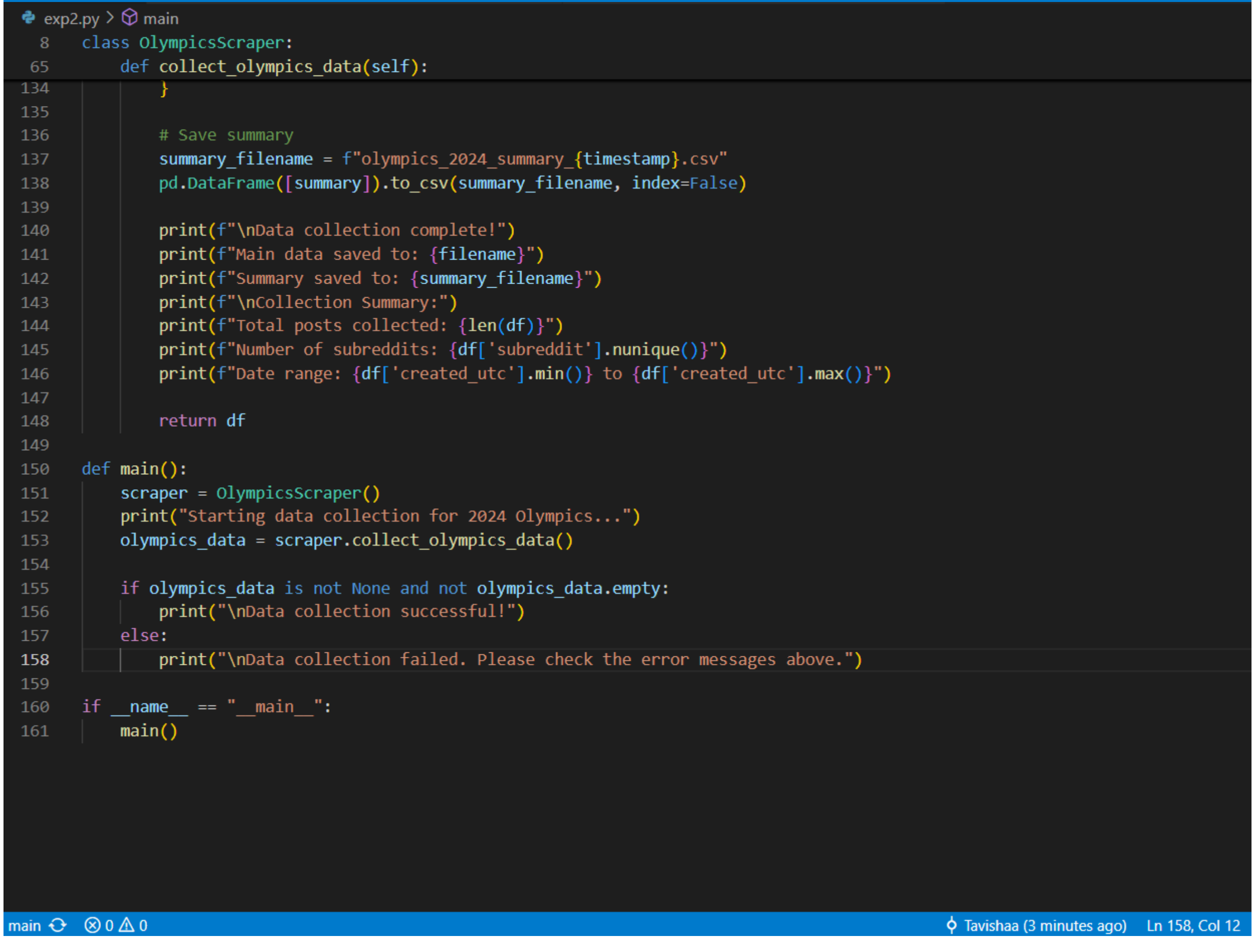
```python
class OlympicsScraper:
    def get_posts(self, subreddit='sports', query='Olympics 2024', limit=25):
                            'author': post_data.get('author', '[deleted]'),
                            'url': self.base_url + post_data.get('permalink', ''),
                            'num_comments': post_data.get('num_comments', 0),
                            'created_utc': datetime.fromtimestamp(post_data.get('created_utc', 0)).strftime('%Y-%m-%d %H:%M:%S'),
                            'subreddit': post_data.get('subreddit', subreddit)
                        })
                    print(f"Successfully collected {len(posts_data)} posts from r/{subreddit}")
                else:
                    print(f"No posts found in r/{subreddit}")
            else:
                print(f"Failed to get data from r/{subreddit}. Status code: {response.status_code}")

        except Exception as e:
            print(f"Error collecting data from r/{subreddit}: {str(e)}")

        return posts_data

    def analyze_engagement(self, row):
        score = int(row.get('score', 0))
        comments = int(row.get('num_comments', 0))

        if score > 1000 or comments > 100:
            return 'High'
        elif score > 100 or comments > 20:
            return 'Medium'
        else:
            return 'Low'

    def collect_olympics_data(self):
        # Multiple subreddits and queries for broader data collection
        subreddits = ['sports', 'olympics', 'worldnews', 'paris']
        queries = ['Olympics 2024', 'Paris Olympics', 'Olympic Games']

        all_posts = []

```



```python
class OlympicsScraper:
    def collect_olympics_data(self):
        # Collect data from each subreddit with each query
        for subreddit in subreddits:
            for query in queries:
                print(f"\nCollecting data from r/{subreddit} for query: {query}")
                posts = self.get_posts(subreddit=subreddit, query=query)
                all_posts.extend(posts)
                time.sleep(2)  # Respect rate limiting

        if not all_posts:
            print("No data collected. Please check your internet connection and try again.")
            return None

        # Convert to DataFrame
        df = pd.DataFrame(all_posts)

        # Remove duplicates
        df = df.drop_duplicates(subset=['url'])

        # Add engagement level
        df['engagement_level'] = df.apply(self.analyze_engagement, axis=1)

        # Add post categories
        def categorize_post(title):
            title = title.lower()
            categories = {
                'Ceremonies': ['opening', 'ceremony', 'closing'],
                'Swimming': ['swim', 'swimming', 'pool'],
                'Athletics': ['track', 'field', 'athletics', 'run'],
                'Event Schedule': ['schedule', 'program', 'timing'],
                'Infrastructure': ['stadium', 'venue', 'facility'],
                'Teams': ['team', 'athlete', 'qualification']
            }

            for category, keywords in categories.items():
                if any(keyword in title for keyword in keywords):
```

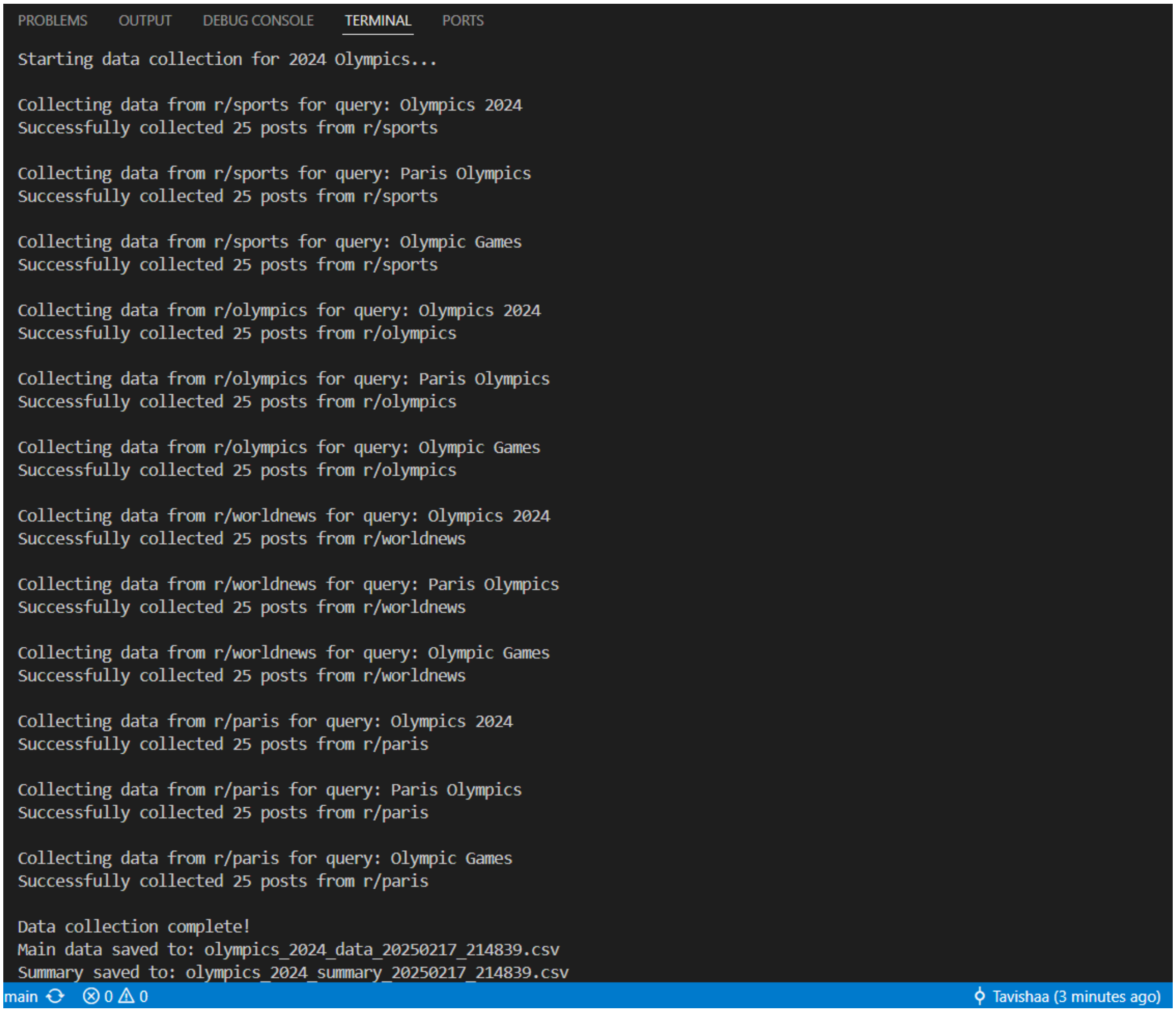


```python
class OlympicsScraper:
    def collect_olympics_data(self):
        def categorize_post(title):
            for category, keywords in categories.items():
                if any(keyword in title for keyword in keywords):
                    return category
            return 'General'

        df['category'] = df['title'].apply(categorize_post)

        # Save the data
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        filename = f"olympics_2024_data_{timestamp}.csv"

        # Reorder columns for better readability
        columns_order = [
            'title', 'score', 'num_comments', 'engagement_level',
            'category', 'subreddit', 'author', 'created_utc', 'url'
        ]
        df = df[columns_order]

        # Save to CSV
        df.to_csv(filename, index=False, encoding='utf-8-sig')

        # Generate and save summary
        summary = {
            'Total Posts': len(df),
            'Unique Subreddits': df['subreddit'].nunique(),
            'Average Score': df['score'].mean(),
            'Average Comments': df['num_comments'].mean(),
            'Categories Distribution': df['category'].value_counts().to_dict(),
            'Engagement Levels': df['engagement_level'].value_counts().to_dict()
        }

        # Save summary
        summary_filename = f"olympics_2024_summary_{timestamp}.csv"
        pd.DataFrame([summary]).to_csv(summary_filename, index=False)
```



```python
class OlympicsScraper:
    def collect_olympics_data(self):
        }

        # Save summary
        summary_filename = f"olympics_2024_summary_{timestamp}.csv"
        pd.DataFrame([summary]).to_csv(summary_filename, index=False)

        print(f"\nData collection complete!")
        print(f"Main data saved to: {filename}")
        print(f"Summary saved to: {summary_filename}")
        print(f"\nCollection Summary:")
        print(f"Total posts collected: {len(df)}")
        print(f"Number of subreddits: {df['subreddit'].nunique()}")
        print(f"Date range: {df['created_utc'].min()} to {df['created_utc'].max()}")

        return df

def main():
    scraper = OlympicsScraper()
    print("Starting data collection for 2024 Olympics...")
    olympics_data = scraper.collect_olympics_data()

    if olympics_data is not None and not olympics_data.empty:
        print("\nData collection successful!")
    else:
        print("\nData collection failed. Please check the error messages above.")

if __name__ == "__main__":
    main()
```

OUTPUT:



```
Starting data collection for 2024 Olympics...

Collecting data from r/sports for query: Olympics 2024
Successfully collected 25 posts from r/sports

Collecting data from r/sports for query: Paris Olympics
Successfully collected 25 posts from r/sports

Collecting data from r/sports for query: Olympic Games
Successfully collected 25 posts from r/sports

Collecting data from r/olympics for query: Olympics 2024
Successfully collected 25 posts from r/olympics

Collecting data from r/olympics for query: Paris Olympics
Successfully collected 25 posts from r/olympics

Collecting data from r/olympics for query: Olympic Games
Successfully collected 25 posts from r/olympics

Collecting data from r/worldnews for query: Olympics 2024
Successfully collected 25 posts from r/worldnews

Collecting data from r/worldnews for query: Paris Olympics
Successfully collected 25 posts from r/worldnews

Collecting data from r/worldnews for query: Olympic Games
Successfully collected 25 posts from r/worldnews

Collecting data from r/paris for query: Olympics 2024
Successfully collected 25 posts from r/paris

Collecting data from r/paris for query: Paris Olympics
Successfully collected 25 posts from r/paris

Collecting data from r/paris for query: Olympic Games
Successfully collected 25 posts from r/paris

Data collection complete!
Main data saved to: olympics_2024_data_20250217_214839.csv
Summary saved to: olympics_2024_summary_20250217_214839.csv
```

Link to the csv files:

- https://github.com/Tavishaa/SMA-analysis/blob/main/olympics_2024_data_20250217_214839.csv
- https://github.com/Tavishaa/SMA-analysis/blob/main/olympics_2024_summary_20250217_214839.csv

B.3 Observations and learning:

Observation:

The script successfully collects and organizes Olympics-related posts from various subreddits.

Learnings:

- Web scraping from multiple subreddits ensures a broader view of topics.
- Cleaning data is crucial for consistency and clarity in analysis.
- Analyzing engagement helps identify popular and influential posts.
- Categorizing posts by topics aids in spotting trends and areas of interest.

B.4 Conclusion:

The script effectively gathers and organizes data from Reddit, focusing on the 2024 Olympics. By cleaning the data, analyzing engagement, and categorizing posts, it provides valuable insights into the discussions surrounding the event. The process emphasizes the importance of efficient data collection, preprocessing, and categorization in drawing meaningful conclusions from large datasets.

B.5 Question of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Q1. Explain in details; why social media data collection is important

Ans:

- **Understanding Public Sentiment**: Analyzes public opinions and feelings in real-time, helping to gauge reactions to events or issues.
- **Identifying Trends**: Tracks emerging topics and conversations, aiding in trend prediction and adapting strategies accordingly.
- **Targeted Marketing**: Allows businesses to create personalized campaigns based on audience engagement and interests.
- **Crisis Management**: Quickly detects negative feedback or PR issues, enabling timely responses to manage reputation.
- **Supporting Research**: Provides data for academic and behavioral studies, offering insights into human interactions and societal shifts.

Q2. Explain: What social media data should you track?

Ans:

1. **Engagement Metrics**: Track likes, comments, shares, mentions, and tags to gauge content interaction and reach.
2. **Content Performance**: Measure post reach, impressions, and click-through rate (CTR) to assess visibility and effectiveness.
3. **Sentiment Analysis**: Analyze positive, neutral, and negative sentiments to understand public opinion.
4. **Hashtag Tracking**: Monitor hashtag mentions and popularity to identify trends and key topics.
5. **Audience Demographics**: Track age, gender, location, and interests to better understand and target your audience.

Q3. What is social listening?
Ans:

1. **Monitoring Conversations**: Social listening involves tracking online discussions across social media platforms to understand public sentiment and opinions.
2. **Identifying Trends**: It helps identify emerging trends, popular topics, and changes in user behavior.
3. **Analyzing Brand Perception**: Social listening tracks how a brand, product, or event is perceived by the public through comments, reviews, and mentions.
4. **Competitor Insights**: It provides insights into competitors' activities and how they are being discussed by users.
5. **Improving Strategy**: Social listening allows businesses to adjust their marketing, customer service, and product strategies based on real-time feedback.

Q4. What is facebook pixel? Explain the working of facebook pixel with suitable case study.
Ans:
Facebook Pixel: A piece of code placed on a website to track user actions and gather data for improving Facebook ad campaigns.

How Facebook Pixel Works:

1. **Tracking User Actions**: It tracks actions like page views, clicks, purchases, or sign-ups on a website.
2. **Data Collection**: The pixel collects data on user behavior (e.g., what products they viewed or added to cart).
3. **Optimizing Ads**: This data is used to optimize Facebook ads by targeting users who are more likely to complete the desired actions.
4. **Retargeting**: Facebook Pixel helps retarget users who previously visited the website but didn't take action, showing them ads to encourage conversions.
5. **Analyzing Campaign Effectiveness**: It provides insights into the success of Facebook ads and how effectively they drive conversions.

Case Study Example:

- **E-commerce Business**: An online store installs the Facebook Pixel to track users who visit product pages but don't complete a purchase. Based on this data, the store runs retargeting ads, showing ads to users who abandoned their carts. As a result, conversions increase by 20%, demonstrating the pixel's effectiveness in driving sales and optimizing ad campaigns.