**Experiment No. 2**

**Topic: Select the social media platforms of your choice ( Twitter, Facebook, LinkedIn, Youtube,Web blogs etc), connect to and capture social media data for business.**

**Group no. 33**

**Data source: Reddit API wrapper**

**Dataset link:** [**https://github.com/WanderLust08/SMA-Pracs/blob/main/reddit\_malnutrition\_data.json**](https://github.com/WanderLust08/SMA-Pracs/blob/main/reddit_malnutrition_data.json)

**Code:**

import praw

import json

import time

# Set up Reddit API credentials

reddit = praw.Reddit(

    client\_id="4wMHyfMxZUJZdX\_xQwPU0Q",

    client\_secret="76dH47Tja8nDn5PmFCYcdNN75KxmaQ",

    user\_agent="MalnutritionResearchBot"

)

# Define relevant subreddits and keywords (removed India-related searches)

subreddits = ["nutrition", "publichealth", "poverty", "globalhealth", "health"]

keywords = ["malnutrition", "undernutrition", "stunting", "nutritionaldeficiency", "globalmalnutrition", "hunger"]

# Function to fetch top comments from a post

def fetch\_comments(submission, comment\_limit=5):

    submission.comments.replace\_more(limit=0)  # Load all top-level comments

    sorted\_comments = sorted(submission.comments, key=lambda c: c.score, reverse=True)  # Sort by score

    comments = []

    for top\_comment in sorted\_comments[:comment\_limit]:  # Get highest-scoring comments

        if len(top\_comment.body) < 10 or (top\_comment.author and top\_comment.author.name == "AutoModerator"):  # Ignore short comments and AutoModerator

            continue

        comments.append({

            "comment\_text": top\_comment.body,

            "comment\_author": top\_comment.author.name if top\_comment.author else "Deleted",

            "score": top\_comment.score

        })

    return comments

# Function to fetch posts from Reddit

def fetch\_reddit\_posts(subreddit\_name, keyword, limit=500):

    posts = []

    subreddit = reddit.subreddit(subreddit\_name)

    for submission in subreddit.search(keyword, limit=limit):

        if not submission.selftext.strip():  # Ignore posts with no text

            continue

        posts.append({

            "title": submission.title,

            "text": submission.selftext,

            "url": submission.url,

            "author": submission.author.name if submission.author else "Deleted",  # Get post author

            "score": submission.score,

            "comments\_count": submission.num\_comments,

            "created\_utc": submission.created\_utc,

            "subreddit": submission.subreddit.display\_name,

            "comments": fetch\_comments(submission)  # Include top comments with authors

        })

    return posts

# Fetch posts for each subreddit and keyword

all\_posts = {}

for subreddit in subreddits:

    for keyword in keywords:

        key = f"{subreddit}\_{keyword}"  # Unique key for each search

        print(f"Fetching posts for r/{subreddit} with keyword '{keyword}'...")

        all\_posts[key] = fetch\_reddit\_posts(subreddit, keyword)

# Save data to JSON

json\_filename = "reddit\_malnutrition\_data.json"

with open(json\_filename, "w", encoding="utf-8") as f:

    json.dump(all\_posts, f, indent=4)

print(f"Reddit data saved to {json\_filename}")