import tweepy

import pandas as pd

from textblob import TextBlob

from transformers import pipeline

# Twitter API keys (replace with your own or skip this if using offline data)

consumer\_key = 'YOUR\_CONSUMER\_KEY'

consumer\_secret = 'YOUR\_CONSUMER\_SECRET'

access\_token = 'YOUR\_ACCESS\_TOKEN'

access\_token\_secret = 'YOUR\_ACCESS\_TOKEN\_SECRET'

# Authenticate (optional - only if you want to stream live tweets)

auth = tweepy.OAuth1UserHandler(consumer\_key, consumer\_secret, access\_token, access\_token\_secret)

api = tweepy.API(auth)

# Function to fetch tweets based on a topic

def fetch\_tweets(query, count=100):

tweets = tweepy.Cursor(api.search\_tweets, q=query, lang='en').items(count)

tweet\_data = [{'text': tweet.text} for tweet in tweets]

return pd.DataFrame(tweet\_data)

# Function for basic sentiment analysis

def get\_sentiment(text):

analysis = TextBlob(text)

polarity = analysis.sentiment.polarity

if polarity > 0:

return 'positive'

elif polarity < 0:

return 'negative'

else:

return 'neutral'

# Function for emotion classification using a transformer

emotion\_classifier = pipeline("text-classification", model="j-hartmann/emotion-english-distilroberta-base", return\_all\_scores=False)

def get\_emotion(text):

try:

result = emotion