```
!pip install asyncpraw torch torchvision torchaudio nest_asyncio matplotlib pandas numpy
```

```
Requirement already satisfied: asyncpraw in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: torchvision in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: torchaudio in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.11/dist-
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-
Requirement already satisfied: aiofiles in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: aiohttp<4 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: aiosqlite<=0.17.0 in /usr/local/lib/python3.11/
Requirement already satisfied: asyncprawcore<3,>=2.4 in /usr/local/lib/python3
Requirement already satisfied: update_checker>=0.18 in /usr/local/lib/python3.
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/pyt
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/loca
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/p
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/p
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/li
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/li
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in /usr/local/
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/pyth
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/py
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/local/l
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dis
Requirement already satisfied: huggingface-hub<1.0,>=0.26.0 in /usr/local/lib/
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/pytho
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/d
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/
```

```
    Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.1
    Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/
```

```
import asyncpraw
import asyncio
import torch
import torch.nn.functional as F
import nest_asyncio
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from collections import Counter
from transformers import AutoTokenizer, AutoModelForSequenceClassification


client_id = "HNUKCrL6c_OOcpLjBCY5WQ"
client_secret = "twvnKPIbXVHZru7csyRMudPcLsIHYw"
user_agent = "Depsenanal/0.1 by u/KrishRON_TheDon"

async def get_reddit_user_activity(username):
    """Fetches last 10 Reddit comments + their subreddits."""
    reddit = asyncpraw.Reddit(client_id=client_id, client_secret=client_secret, user_agen

    try:
        redditor = await reddit.redditor(username)
        comments, subreddits = [], []

        async for comment in redditor.comments.new(limit=10):
            comments.append(comment.body)
            subreddits.append(comment.subreddit.display_name)

        return comments, subreddits

    except Exception as e:
        print(f"Error: {e}")
        return None, None
    finally:
        await reddit.close()



device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

model_ckpt = "bhadresh-savani/distilbert-base-uncased-emotion"
tokenizer = AutoTokenizer.from_pretrained(model_ckpt)
model = AutoModelForSequenceClassification.from_pretrained(model_ckpt).to(device)

emotion_labels = ["sadness", "joy", "love", "anger", "fear", "surprise"]
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: Use
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access pu
  warnings.warn(
tokenizer_config.json: 100%                       291/291 [00:00<00:00, 13.9kB/s]
config.json: 100%                                 768/768 [00:00<00:00, 70.1kB/s]
vocab.txt: 100%                                 232k/232k [00:00<00:00, 5.27MB/s]
special_tokens_map.json: 100%                     112/112 [00:00<00:00, 7.55kB/s]
model.safetensors: 100%                         268M/268M [00:02<00:00, 131MB/s]
```

```python
def get_prediction(text):
    """Predicts emotion of a Reddit comment."""
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding="max_length",

    with torch.no_grad():
        output = model(**inputs)

    probabilities = F.softmax(output.logits, dim=1)
    pred = torch.argmax(probabilities, dim=1).item()

    return emotion_labels[pred]


def show_single_frame(comments, emotions, subreddits):
    """Creates a single frame with a BIGGER comment table on top and two side-by-side gra

    # Subreddit Frequency
    top_subreddits = Counter(subreddits).most_common(6)
    subreddit_labels, subreddit_counts = zip(*top_subreddits)

    # Emotion Count
    emotion_counts = {label: emotions.count(label) for label in emotion_labels}

    # Create Figure
    fig = plt.figure(figsize=(16, 14), facecolor="#121212")
    gs = fig.add_gridspec(3, 2, height_ratios=[2.5, 1, 1])

    fig.suptitle("💖 Emotions of Text 💖", fontsize=26, fontweight="bold", color="hotpin

    # ==== 📜 Table: BIGGER Top-Half ====
    ax_table = fig.add_subplot(gs[0, :])
    ax_table.axis("off")

    table_data = [[comments[i][:150] + "...", subreddits[i], emotions[i]] for i in range
    col_labels = ["📜 Reddit Comment", "🏠 Subreddit", "🎭 Predicted Emotion"]

    table = ax_table.table(
        cellText=table_data, colLabels=col_labels, cellLoc="center", loc="center",
        colColours=["#ff69b4", "#ff00ff", "#ff1493"], bbox=[0, 0, 1, 1]
    )

    table.auto_set_font_size(False)
    table.set_fontsize(14)  # Bigger text
    table.scale(1.4, 1.4)  # Bigger table
    for cell in table.get_celld().values():
        cell.set_edgecolor("black")
        cell.set_facecolor("#222222")
        cell.get_text().set_color("white")

    # ==== 🎭 Emotion & Subreddit Pie Charts (Side-by-Side) ====
    ax1 = fig.add_subplot(gs[1:, 0])
    ax2 = fig.add_subplot(gs[1:, 1])

    # 🎭 Emotion Pie
    colors = ["#ff69b4", "#ff1493", "#ff4500", "#ff6347", "#ff00ff", "#ffb6c1"]
    wedges, texts, autotexts = ax1.pie(
        emotion_counts.values(), labels=emotion_labels, autopct="%1.1f%%",
        colors=colors, textprops={"fontsize": 14, "weight": "bold", "color": "white", "f
        wedgeprops={"edgecolor": "black", "linewidth": 2, "antialiased": True},
        startangle=140
    )

    for text in texts + autotexts:
        text.set_fontsize(16)

    ax1.set_title("🌀 Emotion Analysis 🌀", fontsize=18, color="hotpink", fontweight="bo

    # 🔥 Subreddit Pie
    colors_sub = ["#8a2be2", "#00ffff", "#ff4500", "#00ff7f", "#ff6347", "#ffff00"]
    wedges, texts, autotexts = ax2.pie(
        subreddit_counts, labels=subreddit_labels, autopct="%1.1f%%",
        colors=colors_sub, textprops={"fontsize": 14, "weight": "bold", "color": "white"
        wedgeprops={"edgecolor": "black", "linewidth": 2, "antialiased": True},
        startangle=140
    )

    for text in texts + autotexts:
        text.set_fontsize(16)

    ax2.set_title("🔥 Subreddit Activity 🔥", fontsize=18, color="hotpink", fontweight="

    plt.tight_layout(rect=[0, 0, 1, 0.94])
    plt.show()


async def main():
    username = "yuz_HUNKAI15"
    comments, subreddits = await get_reddit_user_activity(username)

    if not comments or not subreddits:
        print("No data found! Try again.")
        return

    # Get Emotions
    emotions = [get_prediction(comment) for comment in comments]

    # Display Table + Side-by-Side Pie Charts
    show_single_frame(comments, emotions, subreddits)


nest_asyncio.apply()

def run_asyncio_main():
    loop = asyncio.get_event_loop()
    if loop.is_running():
        return asyncio.ensure_future(main())  # Workaround for Colab
    else:
        return asyncio.run(main())
```

```
run_asyncio_main()
```

```
<Task pending name='Task-1' coro=<main() running at <ipython-input-8-
766ae61436fc>:1>>
```

```
!pip install asyncpraw transformers torch matplotlib nest_asyncio pandas
```

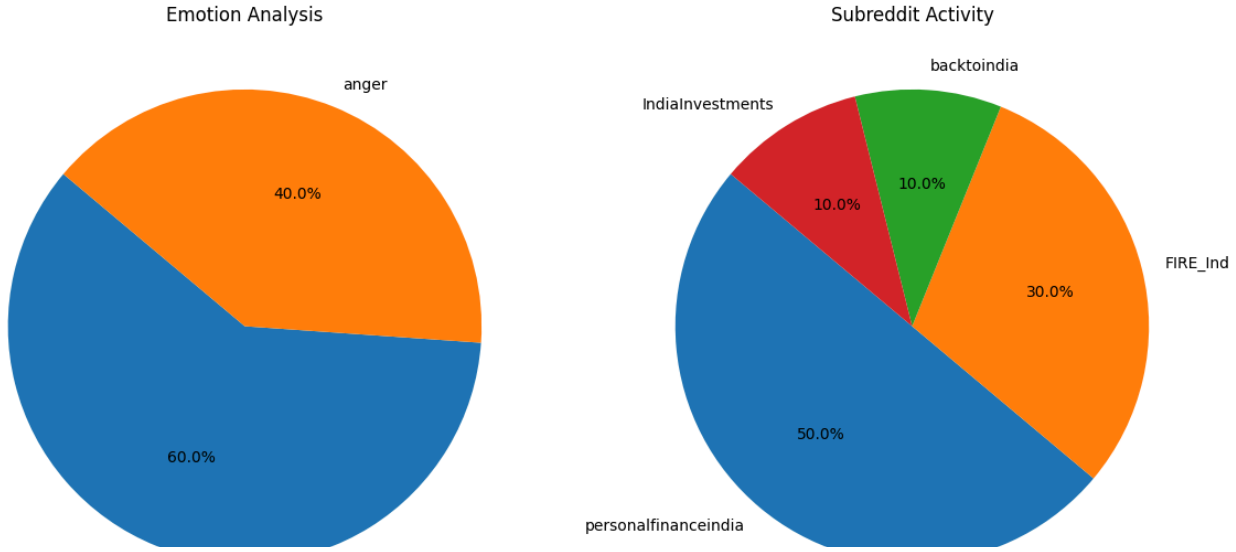
```
Requirement already satisfied: asyncpraw in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: aiofiles in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: aiohttp<4 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: aiosqlite<=0.17.0 in /usr/local/lib/python3.11/
Requirement already satisfied: asyncprawcore<3,>=2.4 in /usr/local/lib/python3
Requirement already satisfied: update_checker>=0.18 in /usr/local/lib/python3.
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: huggingface-hub<1.0,>=0.26.0 in /usr/local/lib/
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/pyt
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/loca
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/p
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/p
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/li
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/li
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in /usr/local/
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/pyth
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/py
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/local/l
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dis
```

```
    Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/pytho
    Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/d
    Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist
    Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/
    Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.1
    Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/d
    Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/
    Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-pack
```

```
import torch
import torch.nn.functional as F
import matplotlib.pyplot as plt
import pandas as pd
import asyncio
import asyncpraw
import nest_asyncio
from collections import Counter
from transformers import AutoTokenizer, AutoModelForSequenceClassification

# Apply nest_asyncio to handle async functions
nest_asyncio.apply()

# Set device (GPU if available)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# ========== Load Emotion Analysis Model ==========
emotion_model_ckpt = "bhadresh-savani/distilbert-base-uncased-emotion"
emotion_tokenizer = AutoTokenizer.from_pretrained(emotion_model_ckpt)
emotion_model = AutoModelForSequenceClassification.from_pretrained(emotion_model_ckpt).to

# Emotion labels
emotion_labels = ["sadness", "joy", "love", "anger", "fear", "surprise"]



# Reddit API Credentials
client_id = "HNUKCrL6c_OOcpLjBCY5WQ"
client_secret = "twvnKPIbXVHZru7csyRMudPcLsIHYw"
user_agent = "Depsenanal/0.1 by u/KrishRON_TheDon"

async def get_reddit_user_activity(username):
    """Fetches last 10 Reddit comments of a user."""
    reddit = asyncpraw.Reddit(client_id=client_id, client_secret=client_secret, user_agen
    try:
        redditor = await reddit.redditor(username)
        comments, subreddits = [], []
        async for comment in redditor.comments.new(limit=10):
            comments.append(comment.body)
            subreddits.append(comment.subreddit.display_name)
        return comments, subreddits
    except Exception as e:
        print(f"Error: {e}")
        return None, None
    finally:
        await reddit.close()



def get_prediction(text):
    """Predicts emotion of a Reddit comment."""
    inputs = emotion_tokenizer(text, return_tensors="pt", truncation=True, padding="max_l
    with torch.no_grad():
        output = emotion_model(**inputs)
    return emotion_labels[torch.argmax(F.softmax(output.logits, dim=1)).item()]



def generate_graphs(emotions, subreddits):
    """Generates pie charts for Emotion Analysis & Subreddit Activity."""
    fig, axes = plt.subplots(1, 2, figsize=(12, 6))

    # Emotion Analysis Pie Chart
    emotion_counts = Counter(emotions)
    axes[0].pie(emotion_counts.values(), labels=emotion_counts.keys(), autopct="%1.1f%%"
    axes[0].set_title("Emotion Analysis")

    # Subreddit Activity Pie Chart
    subreddit_counts = Counter(subreddits)
    axes[1].pie(Counter(subreddits).values(), labels=Counter(subreddits).keys(), autopct=
    axes[1].set_title("Subreddit Activity")

    plt.tight_layout()
    plt.show()



# Ask for Reddit username
username = input("Enter Reddit username: ")

# Fetch Reddit data
comments, subreddits = asyncio.run(get_reddit_user_activity(username))

if comments:
    # Predict emotions
    emotions = [get_prediction(comment) for comment in comments]

    # Display DataFrame
    df = pd.DataFrame({"Comment": comments, "Subreddit": subreddits, "Emotion": emotions
    display(df)

    # Generate Graphs
    generate_graphs(emotions, subreddits)
else:
    print("No data found! Check if the username is correct.")
```

```
Enter Reddit username: SriNivesh
```

	Comment	Subreddit	Emotion
0	To answer the NRI question, the transmitted un...	personalfinanceindia	joy
1	Please review your statement and modify it. T...	personalfinanceindia	anger
2	There have been many accounts of made-in-India...	FIRE_Ind	anger
3	There is one thing about mutual funds that no ...	personalfinanceindia	anger
4	THIS. \n\nIf we dig deep enough, there would d...	FIRE_Ind	joy
5	I wanted to comment on the 'at least one to tw...	backtoindia	joy
6	It is good to see many positive comments. I h...	personalfinanceindia	anger
7	20k with a risk capital - loan amount - of 10 ...	personalfinanceindia	joy
8	I had chosen to cross-post in this sub. I hav...	FIRE_Ind	joy
9	Some of you may know of my second career as a ...	IndiaInvestments	joy


Emotion Analysis
anger
Subreddit Activity
backtoindia
IndiaInvestments
FIRE_Ind
personalfinanceindia