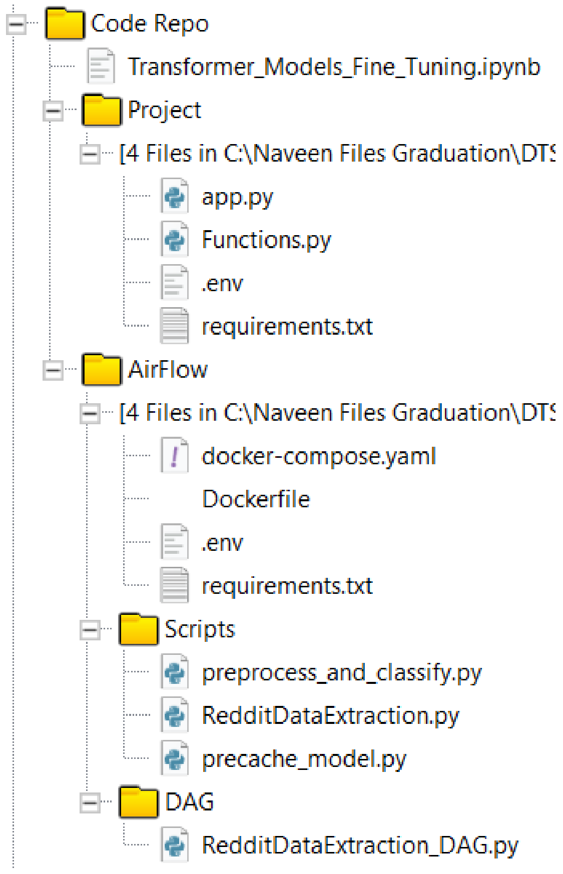
Installation Guide:

- The project can be divided into 2 main components. First is the data extraction, scheduled using Airflow, and the second component is the rest of the project i.e. data preprocessing, fine-tuning classification models, sentiment analysis, Topic Modeling, SHAP and web application.
- The organization of my code is shown in Fig-1



```
Code Repo
    Transformer_Models_Fine_Tuning.ipynb
    Project
        [4 Files in C:\Naveen Files Graduation\DTS
            app.py
            Functions.py
            .env
            requirements.txt
    AirFlow
        [4 Files in C:\Naveen Files Graduation\DTS
            docker-compose.yaml
            Dockerfile
            .env
            requirements.txt
        Scripts
            preprocess_and_classify.py
            RedditDataExtraction.py
            precache_model.py
        DAG
            RedditDataExtraction_DAG.py
```

Fig-1: Code organization

Transformer_Models_Fine_Tuning.ipynb

- This notebook contains the code to preprocess and finetune 4 transformer based classification models.
- It uses the csv file titled "Reddit_Posts.csv" collected from HuggingFace. Upload this dataset to the colab environment before code.

- First, cell uninstall the conflicting libraries and reinstall the right versions. So, there will be no library conflicts.
- When you run the third cell, you will be asked to enter the HuggingFace API Token with "WRITE" access. Just paste the following token "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", uncheck "Use as Git Credential" and you are good to run the rest of the code. (You can find the same token in the training_args as well)

AirFlow:

- The airflow folder contains code regarding the extraction and processing of live reddit data from Reddit API.
- It has 3 folders. Scripts and DAG.
    - Scripts – This folder have 3 files
        - RedditDataExtraction.py – Used to extract most recent data (past 1 day) using the reddit API.
        - precache_model.py – This file contains the code to download and cache the best performing classification model from my huggingface hub (pushed by my ipynb file) .
        - preprocess_and_classify.py – preprocess, classify, assign sentiment score to the live data using the loaded model and VADER. It will also upload the data to a google sheet. Data from each subreddit will be uploaded to different worksheets.
    - dags – This folder contains a script called RedditDataExtraction_DAG.py to schedule run the above 3 scripts everyday to accumulate daily posts. We define the tasks, what command it should run and working directories as well.

    - Config – This folder contains a "*.json" file related to my google service account. This contains different keys to get write access to the google sheet I created to upload the data. This will be used by the preprocess_and_classify.py in Scripts folder.
- Dockerfile – This will tell you how to build the airflow image using the base image along with the customizations made to accommodate the data extraction.
- docker-compose.yaml – This acts as an orchestrator of the local environment. This will necessarily tell how to run multiple containers (if any) simultaneously.
- requirements.txt – This text file has all the required libraries to perform data extraction
- .env – this file has all the secrets to support the data extraction (like HF token, reddit API ID and secret)

How to set up the data extraction pipeline:

- To set up the data extraction pipeline, make sure you have docker desktop installed.
- Save the AirFlow folder to somewhere on local device.
- Inside that, ensure that you created the following folders "Documents", "logs", "plugins", "RedditData" along with the existing folders and files (if not created automatically).
- Don't change the organization of the files since they were already set in the docker-compose.yaml file
- Open a terminal, navigate to the "AirFlow" folder and run the following 3 commands. The first command might return an error if the docker container is not already running.

  - docker compose down
  - docker compose build
  - docker compose up -d
- You can navigate to localhost:8080 to view the scheduled DAGs, previous runs etc. You can also manually trigger the DAGs

Project Folder:

- requirements.txt – contains all the libraries to run the code in app.py
- .env – contains secrets from google service account, huggingface, and reddit API
- Functions.py – contains all the code required to support app.py. It contains functions related to model loading, data extraction, preprocessing, sentiment score assignment (in case of page-2 live data extraction), time series analysis, topic modeling, and SHAP.
- app.py – contains code to build the web application.

How to set up the web application:

- Create a new conda environment
- Activate that conda environment.
- Run "pip install -r requirements.txt" to install all the required libraries
- Navigate to the "Project" folder.
- Run "streamlit run app.py" to launch the web application in a browser.
- As long as the code is running, you can also navigate to "localhost:8501" to open the website.
- In the terminal where you ran the code, you can also view the network URL which anyone on the same internet can use to interact with the web application.