Documentation:

1. All you need is to have python 3.12 as well as the following python modules
    a. Pandas: Used to build our csv files.
    b. Numpy: Used with Pandas to create our csv files.
    c. NLTK: Used for its sentiment analysis & text cleaning libraries, but has many useful NLP functions. Installation guide: https://www.nltk.org/install.html
    d. Instagrapi: Unofficial instagram api, chosen for ease of use and project replication. Used for scraping comments and post data from instagram users. Installation guide: https://subzeroid.github.io/instagrapi/getting-started.html
    e. Scikit-learn: Used for implementing models.
2. Follow along the README file to either explore the results of the project, or replicate it with your own set of users.

Documentation by folder:

Instagrambot

1. instabot.py: The python file uses instagrapi to create the instagram_data csv file.
    a. Client.delay_range = [n, m]: This line of code adds a delay between api calls to follow instagram's policy on web scraping. Users can choose the integers n and m.
    b. user_ids: It is a list of instagram's user ids. Users can choose different ids to get data from different users.
2. instagram_data.csv: Contains the data of certain users' last 20 instagram posts. The file includes media_id, media_type, number of followers, number of comments, number of likes.
3. Instagramcomments.py: The python file creates the comments csv file.
4. comments.csv: Contains the last 50 comments for the posts mentioned in instagram_data.csv
5. Instagramfollowers.py: The python file modifies the instagram_data csv file to get the follower count of the given user.

Sentimentanalysis

1. Clean_comments.py: Cleans the comments data in comments.csv and outputs comments_cleaned.csv. Cleaning methods discussed in the functions below. Uses NLTK, Pandas, and Numpy. This file also downloads two nltk files for tokenization and removing stopwords. Finally, a file emo_unicode.py is referenced for removing emojis.
    a. tokenize(sentence): Uses a downloaded NLTK file "punkt" to go through and tokenize comments. Comments are "untokenized" at the end of the file.
    b. remove_stopwords(tokens): Removes stopwords from the data by using the downloaded NLTK file "stopwords" to remove tokens which are in that file.

    c. remove_emoji(tokens): References emo_unicode to remove emojis, by seeing if the tokenized unicode is in emo_unicode.py. Unfortunately does not removed grouped emojis (emojis which are placed together without spaces, a very tricky problem to resolve)
    d. Emo_unicode.py: A file which contains a dictionary of all the unicode representations of emojis.
2. Comments_cleaned.csv: The output of clean_comments.py, just a collection of post id's and their last 50 comments after being cleaned.
3. Sentiment_analysis.py: Uses NLTK, Pandas, and Numpy to perform sentiment analysis on the comments in comments_cleaned.csv. It cleans out any comments with no positive or negative sentiment value, so results are more polarized. Takes the average of every comment's sentiment analysis score, and appends that to instagram_data.csv. This outputs final_data.csv.
4. Final_data.csv: a csv containing all relevant data for the machine learning portion of the project: Every line represents a post containing that post's number of likes, number of comments, number of followers, along with that post's sentiment analysis average. It also contains less relevant information, like the post's id & what type of media it is (photo, video, album, etc.)

Machinelearning

1. final_data.csv: The cleaned data from the Sentimentanalysis step and web scraping. I
2. sentiment_machine_learning.ipynb: Uses sci-kit to train a Gaussian Naive Bayes, Logistic Regression, and Support Vector Classification.
    a. Import all necessary Libraries
    b. Load in data
    c. Normalize the number of likes and comments
    d. Set a threshold at what is considered to be a positive or negative sentiment based on the sentiment average. This will used as a label
    e. Create training and test data sets
    f. Run Gaussian Naive Bayes, Logistic Regression, and Support Vector Classification models.
    g. Use confusion matrix to determine efficacy of models

Project contributions

1. Jaehong: Built the instagram bot, and built our instagram post dataset by writing instabot.py, instagramcomments.py, & instagramfollowers.py.
2. Jake: Researched the Instagram API the bot was built from, as well as helped run the instagramcomments.py file to build the comments.csv. Cleaned the data by writing clean_comments.py. Finally, performed sentiment analysis on the cleaned comment dataset with sentiment_analysis.py
3. Miles: Researched what machine learning algorithms would fit our project's goals best & implemented them in the sentiment_machine_learning.jupyter file. This also serves as a file which can be easily used to access and review the results of our project.