Members:
Nabdeep Shrestha <nabdeep2@illinois.edu>
Rishab Sriramoju <rishabs3@illinois.edu>
Kishore Iyer <kiyer3@illinois.edu>
Aarthi Mahesh <aarthim2@illinois.edu>

1) An overview of the function of the code (i.e., what it does and what it can be used for).

The code displays the sentiment scores of the most controversial posts in a subreddit and also which of the most controversial posts are objective vs subjective. This information can help understand the subreddit and its users more (i.e. are positive sentiment or negative sentiment posts more controversial, are objective posts or opinions more controversial, etc.)

2) Documentation of how the software is implemented with sufficient detail so that others can have a basic understanding of your code for future extension or any further improvement.

Python:

Our Python code sets up a web server using the Flask framework, in which we host our main functionality at the '/hot-posts' route/endpoint. Our main function, in which the bulk of the functionality lies, fetches and analyzes the controversial posts from a user specified subreddit.

Once receiving a response from the backend via praw (Python wrapper library for Reddit), we look at each post's sentiment score and appropriately add it to our objective/subjective classifier dict 'objective_subjective_dict', and our sentiment score data store 'sentiment_dict'. Following all iterations, we return both data stores in a JSONified format.

sentiment_dict is organized into 8 bins based on the sentiment range (i.e. [-1, -0.6) includes everything greater than or equal to -1 and less than -0.6). Sentiment values of 0 are not included in this dict because they are objective. Objective_subjective_dict has 2 bins (objective or subjective) where nonzero sentiment values are in the subjective column and zero sentiment values are in the objective column. Because of this, the sum of all columns in sentiment_dict is the number in the ‘subjective’ column of objective_subjective_dict.

HTML:

Our HTML code sets up our UI for the front-end as well as the bar charts (1. Objective vs Subjective; 2. Sentiment Analysis Scores) which are populated with the sentiment data gathered from our JSON dictionaries. We used the Bootstrap framework for styling, and incorporated the

Chart.js library to dynamically render and update the charts based on our user subreddit input. There is also dynamic exception handling in the case where the subreddit does not exist.

3) Documentation of the usage of the software including either documentation of usages of APIs or detailed instructions on how to install and run a software, whichever is applicable.

Note: We had to change our original proposal of sentiment analysis with X posts because the twitter libraries we tried (i.e. tweepy) weren't able to properly authenticate to the X developer account we created.

To retrieve posts, we used the PRAW library. After creating a reddit developer account, we were able to generate the proper authentication for PRAW to retrieve post titles.

For sentiment analysis, we used the Textblob library which assigns sentiment scores from -1 to 1 (-1 being the most negative and 1 being the most positive).

For the front end, we used a Flask HTML template that is hosted through the Flask dev server.

To run this project, do the following steps:

1. Clone this repo: https://github.com/kish0620/410-Final-Project
2. Install the required packages with pip3 install -r requirements.txt
3. Run python3 main.py in the home directory
4. Navigate to where the server is running (http://127.0.0.1:5000)
5. Enter a subreddit, click submit and wait 5-10 seconds for the result to show (if the subreddit doesn't exist an error message will appear)

4) Brief description of contribution of each team member in case of a multi-person team

Kishore	1. Created reddit developer account and established connection to it using PRAW library 2. Retrieved post titles and organized them into dictionaries on the backend
Nabdeep	1. Initiated the overall Flask project and designed the skeleton for the front and back end 2. Connected the bar chart to the back end to display the number of posts within each sentiment score range
Rishab	Created skeletal code for displaying sentiment

	scores bar chart. Added in logic for extracting user-inputted subreddit (from the text field) from front-end in to the HTTP request.
Aarthi	Created a bar chart to display the number posts within a subreddit that are objective or subjective