CS410 Project Progress Report

1. **Which tasks have been completed?**
    1. Finished the scraper to retrieve and clean data
        1. ***./praw_scraper.py:*** Created a Python Reddit API Wrapper (PRAW) scraper that gets submissions and certain submission information. This was slow and limited to 1000 posts, so this scraper was not used to create the final datasets.
        2. ***./pushshift_scraper.py:*** Created a PushShift API scraper that gets submission titles and comments for given time ranges. In tandem with the PushShift Multithread API Wrapper (PMAW), scraping a lot of data was a lot faster; this scraper was used to retrieve the datasets.
    2. Performed Sentiment Analysis on datasets using NLTK
        1. ***./sentiment_analyzer.py:*** Cleaned datasets, implemented polarity score analysis, labelled resulting data, and outputted label counts to csv files.
2. **Which tasks are pending?**
    1. Get datasets and perform analysis for more time periods and more players (just have to re-run my scraper and analyzer, which just requires waiting).
    2. Creating a Flask API to provide the analyzed data.
    3. Building a front end to display analysis results.
3. **What challenges are you facing?**
    1. Getting quality data that is actually related to specific players has proven to be a lot more difficult than I previously thought.
        1. Scraping thousands of posts/comments through PRAW was taking an extremely long time, so I opted to use the pushshift Reddit API, which effectively is a database of all Reddit comments and submissions, allowing for faster scraping.
        2. Producing accurate query results is also a difficulty, as players go by many different aliases on the subreddit. Stephen Curry, for

example, could just go by “steph” or “curry”, and mentions of “curry” could conflict with his brother Seth Curry.

iii. Not only that, comments referring to a player may often not mention that player’s name, so some comments that could be related to a player were not identifiable. I tried a workaround by getting posts that included a player name and looking through all the comments in those posts, but that also resulted in many unrelated comments.

b. **Solution**: I have decided to go with a combined analysis of post titles and comments that all contain the full names of players. This way, there will not be conflicts from common names and the content will almost certainly be related to the given player in some way. Given the volume of posts/comments on the NBA subreddit, this still results in more than enough data.