Name: Shubham Mahajan   NetID: smm3

I am working alone so I (Shubham Mahajan) will be the captain.

My topic is to provide a web application that will allow users to enter text with complex vocabulary and output text with simple vocabulary while trying to convey the same message as the original text. There are a lot of people in the world whose first language is not English and it can be difficult for them to understand articles that are written in English and have complex vocabulary. The goal of this project is to provide them with a service that can help them understand the article better by outputting text with simpler vocabulary which they will be more familiar with.

I plan to extract words from articles that I find on Kaggle. I will likely need to use multiple datasets as I need articles from a variety of categories (e.g sports, economics, politics). With these articles, I plan to calculate the term frequency for each term. I will assume that the more times a term appears the more familiar people will be with that term and can be categorized as "simpler vocabulary". I also plan to calculate the inverse document frequency of all terms to determine any stop words. These are words that I will not change in the text. I only plan to change the words in the input text with a low term frequency in the articles that I processed from Kaggle. Lastly, I plan to use nltk to determine synonyms of any complex words and replace these complex words with synonyms with a higher term frequency. I plan to use Flask to create the web application and host it with Github pages.

For evaluation, this will need to be a judgment call. I will need to find some new articles and test that as input to my program. I will need to make sure the text is conveying the same message while being easier to understand.

I primarily plan to use Python. I might use some HTML, CSS, and Javascript for the web application but the bulk of the work will be in Python.

Below is a breakdown of how much work I expect this project to take.

Find articles to process on Kaggle	2 hours
Extract all TF and IDF	5 hours
Utilize NLTK and processed data to output simple text	6 hours
Build web application	8 hours
Evaluate / Debug	5 hours
Total	**26 hours**