Considering the drastic changes in the economy currently, I worked on predicting the stock market depending on the daily news headlines.

* 1000 top new headlines, upvoted by Reddit users was chosen to tally it with DJIA historical stock data.
* With DJIA being the index for blue-chip stocks in the USA and since the economy is equally going bad as many other nations I decided to start this analysis by choosing on USA News from Reddit. [1]
* Reddit has hot, new, controversial, top and gilded sections of news. I chose the top 1000 news headlines. At this point, using Python I scraped this data from Reddit. I referenced Felippe Rodrigues article [2] to complete this step.
* These 1000 headlines were from 8-30-2014 to 04-05-2020. Thus, I extracted DJIA historical data also of the same timeline.
* Learning from Aaron7sun's [3] work I too combined the News headlines and the DJIA data by labeling it. The days when the Adj closing price decreased, labeled 0, the days when it increased labeled them 1 and the days for which the historical stock price was not available labeled those days 2.
* Then uploaded these two files on Google Colab. Since here text was going to be analyzed, natural language processing came into picture. Thus, count vectorizer was first implemented. Count Vectorizer, counts the number of times the keyword appears.
* Next Logistic Regression accuracy score was 0.408 Logistic Regression was chosen since it is classification scenario.
* Next again accuracy score was calculated after implementing TF-IDF : term frequency - inverse dense frequency.
* This was chosen because frequently used had to be omitted to avoid redundancy. TF-IDF function has parameters which modulated and filter words. Words having less than 0.01 frequency can be omitted and words having greater than 0.6 frequency can be omitted. The maximum count of keywords can also be set. I tried changing these parameters but surprisingly the accuracy score again came 0.408
* Next, I ran Naive Bayes on these datasets. It is also a supervised classification algorithm. This algorithm is built on Bayes theorem. The probability that a feature is dependent on another feature. The accuracy score improved a little, 0.4113 now.
* I understand the accuracy score is not upto the mark, this means my search for a best fit model continues. I will next implement LSTM.

References:

[1] Reddit USA News, <https://www.reddit.com/r/usanews/>

[2] How to scrape Reddit with Python, <https://www.storybench.org/how-to-scrape-reddit-with-python/>

[3] Reddit Time Machine, <https://github.com/sjhddh/reddit_time_machine>