

A decorative graphic in the top-left corner consisting of a network of grey nodes connected by thin lines, with several nodes highlighted in blue. 

# **Reddit Influence on the Stock Market**


A decorative graphic in the bottom-right corner consisting of a network of grey nodes connected by thin lines, with several nodes highlighted in blue.



# Problem Statement

**I am interested in exploring and understanding the influence of social media platforms, specifically reddit in this case, in effecting price movement in the general market as well as with individual securities. Do investing subreddits actually have enough influence and users with money to in a sense move the market, or is there evidence of only slight influence?**

**One step further, can reddit data from selected subreddits be scored by sentiment analysis and used to accurately forecast price movement through the use of time-series forecasting?**





# 185,000 submissions

That's a lot of submissions!

# 185.5 mb

Has to use AWS 48 Cores 96 GiB RAM

# 10 Hours of Mining

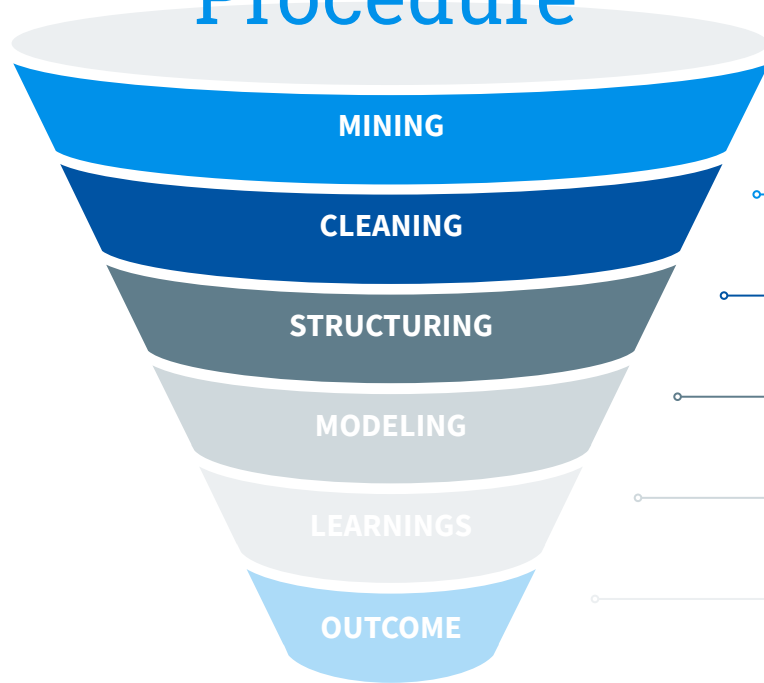
Thorough collection of submissions

# Just how relevant is it?

Being able to quantify how much weighting or relevancy social media has on the market would be a very powerful insight.



# Procedure



Mining

Cleaning

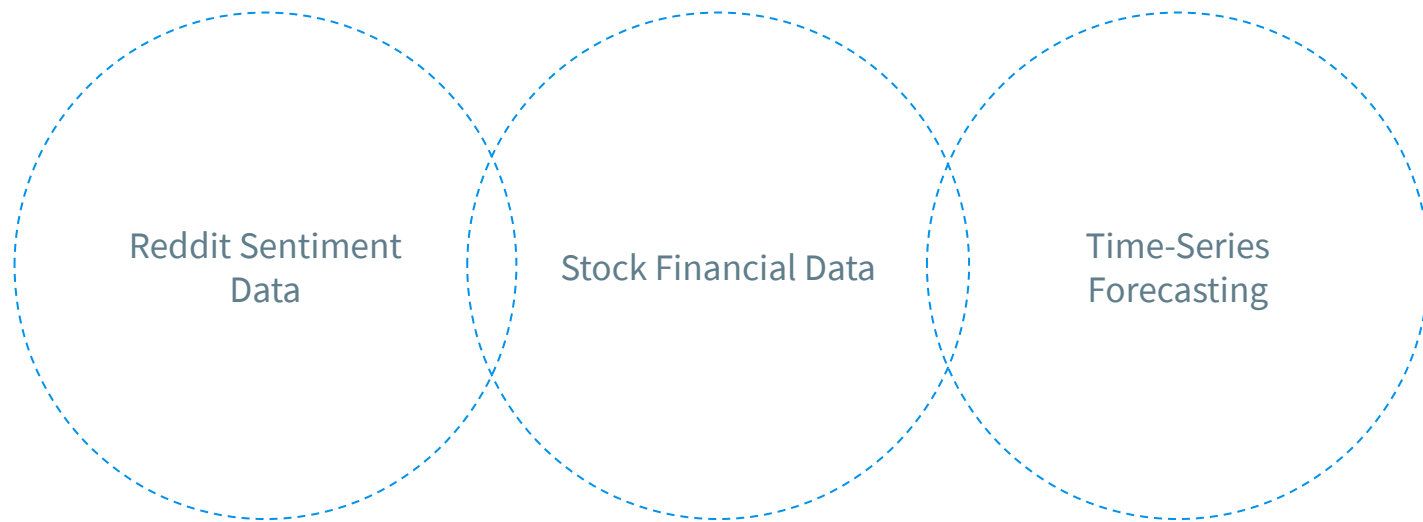
EDA + Data Vis + Structuring

Time Series Forecasting + RNN

Learnings + Revisions

Result: Quantify How Important Subreddits are at effecting price movement

# Structuring





# Mining Sources

- <https://www.reddit.com/r/investing/> Created Mar 15, 2008
- <https://www.reddit.com/r/GME/> Created May 30, 2012
- <https://www.reddit.com/r/wallstreetbets/> Created Jan 31, 2012
- <https://www.reddit.com/r/SecurityAnalysis/> Created Dec 8, 2010
- AlphaVantage API - Indexes + Individual Stocks

# Data Collection 1

I used the PushShift API to mine submissions and comments from the subreddits mentioned before. In total I managed to gather around 185,000 submissions and decided to set the comments aside for later use. Mining 4 separate subreddits 1 day at a time to ensure full collection was arduous and time consuming. I performed all mining on my local machine, which made it difficult to make significant progress with other areas of the project. Next time, I will set up separate AWS instances so I can mine simultaneously and have full CPU / RAM on my local machine. Further, PushShift only allows active use from 1 IP address, meaning with multiple instances with different IPs, I could have gathered data from multiple sources at once.





# Data Collection 2

**I used the AlphaVantage API to mine stock data on the daily range. I collected data on GME, Tesla, the SP500, Dow, and Nasdaq. I grouped the data in different data sets to maintain time intervals of 1 day and 1 week, and did the same with the subreddit data.**

# Data Cleaning 1

Working with such a large dataset, especially text data from reddit was challenging. I ran into a number of issues and when possible attempted to resolve the missing values to keep the submission text data. There were many unique characters that my string cleaning function didn't handle and I had to make a lot of considerations about how to handle unique instances. Through cleaning, my original 185,000 documents were reduced to 165,000.

The financial data from AlphaVantage was clean as can be, and was very easy to collect. I had 0 issues here, and the API works incredible fast.

# Data Cleaning 2

Since I am performing time-series analysis considering the sentiment scores, staying organized with so many different datasets was a challenge. I had to assign sentiment scores to each document, and then filter the entire dataset for the desired keyword (like GME), and then group by day or week. This proved more challenging than expected, but arriving at a dataset with sentiments was highly rewarding.

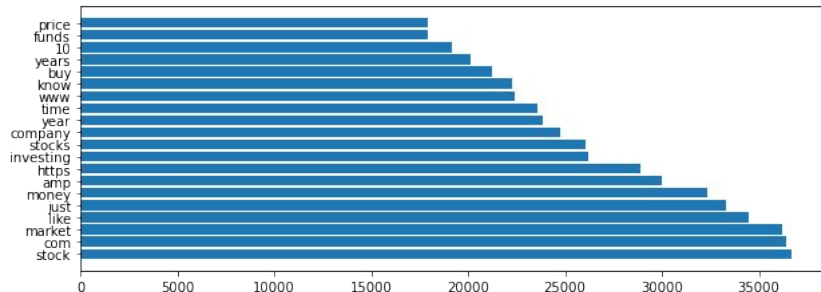
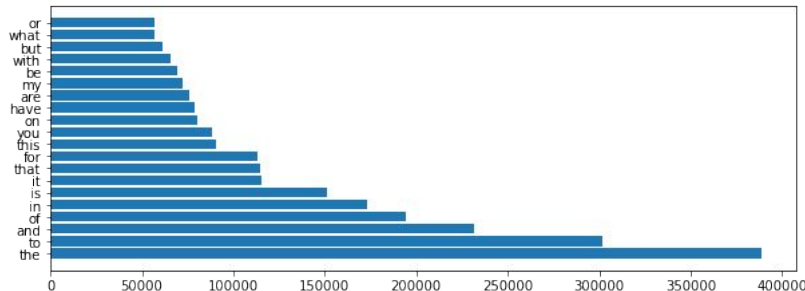


## EDA + Data Visualization

In the pursuit of thoroughly understanding the data, I went to great lengths in this segment of the project to thoroughly understand what I was working with. I have chosen the most relevant displays to communicate what I have discovered.

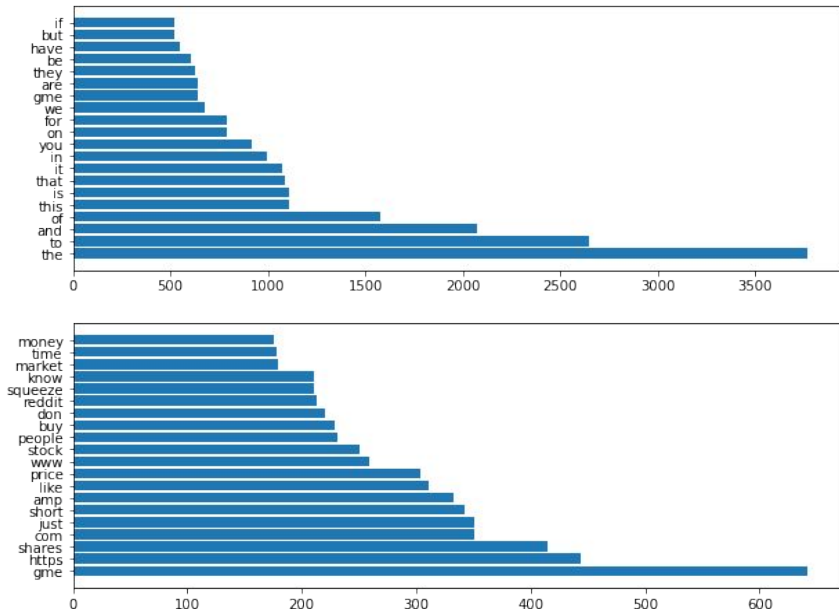
Count-vectorizer proved problematic with 165,000 documents.

Investment Submission Text Dataset: Text Before and After Removing Stopwords



The subreddit submissions was responsible for my most of my documents I mined. The word “the” showed up nearly 400,000 times in my dataset, “to” nearly 300,00 times, and “and” around 225,000 times. Even after cleaning the data of stopwords and the cleaning function I used, there were will meaningless words in the set, such as “com”, “like”, “https”, “year”, “time”, and “www”. This chart informed me of what custom stopwords I needed to use in order to ensure I was using meaningful data. This assisted me in further removing the noise.

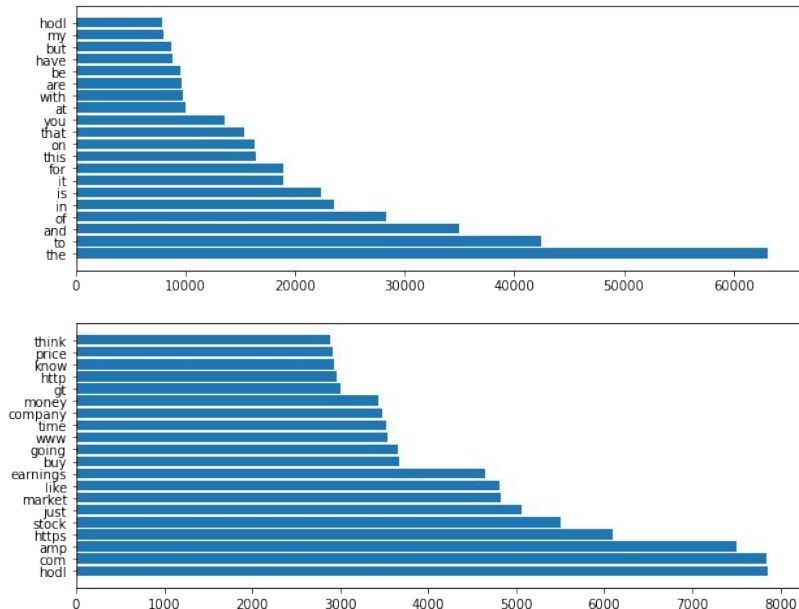
GME Submission Text Dataset: Text Before and After Removing Stopwords



Here, we see a very important element of the recent GME price activity, the word squeeze and short. While there aren't a significant amount of instances to be likely meaningful or impactful, I think it is still important to note.

This before and after shows me that even after cleaning the data of common stopwords, that further cleaning is still warranted and necessary.

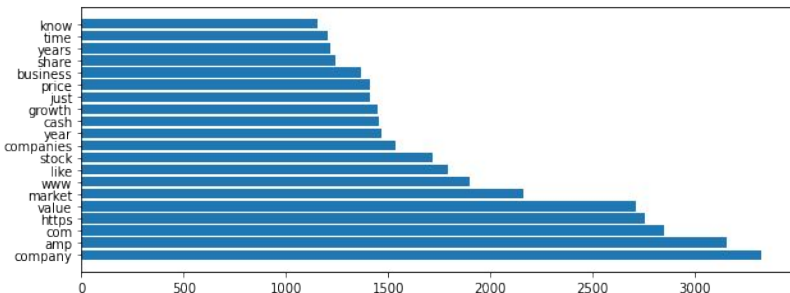
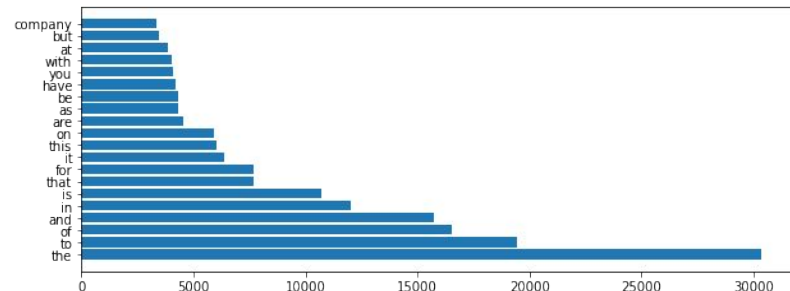
Wall Street Bets Submission Text Dataset: Text Before and After Removing Stopwords



This chart tells one interesting story. Before cleaning, the word `hodl` was 10th on the list in terms of frequency. After cleaning the data of common stopwords, `hodl` changed positions to be the most frequent word in documents. This makes plenty of sense considering the subreddit. `Hodl` means hold on for dear life, showing intent to hold through the peaks and valleys.

Once more further stopwords revision is necessary in order to aggregate meaningful words that actually effect sentiment.

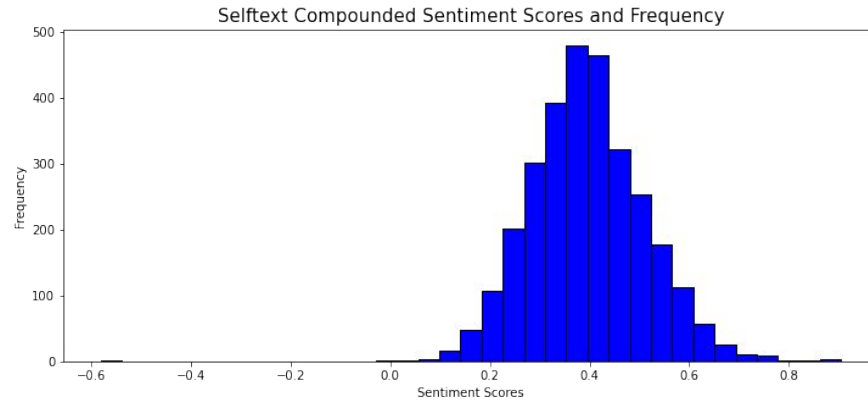
### Security Analysis Submission Text Dataset: Text Before and After Removing Stopwords



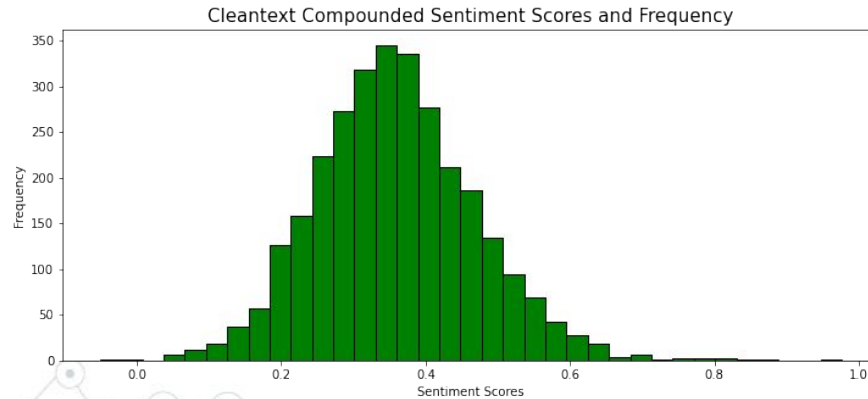
Security Analysis is known as a more reputable subreddit. There is more discussion around the fundamentals and what makes a stock valuable. The term hodl doesn't show up once here. Words like growth, market, value, and company are frequently used in the corpus.

One could assume that for those that use reddit as a source of trading information or ideas, that more senior and experienced investors would participate in this group over wallstreetbets or the subreddit GME.



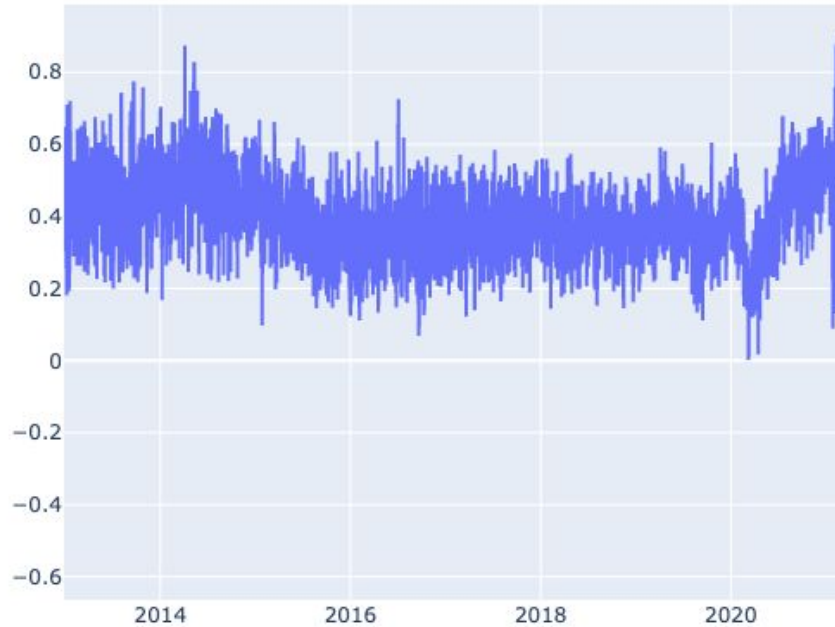


This chart was made after I consolidated all of the sentiment scores from selftext and cleantext into average sentiment by day. I used the Vader Sentiment Intensity Analyzer to gather these metrics.



Something that I find really interesting is that by removing the stopwords and cleaning up the data, the center of the distribution shifted negative, back from 0.4 to around 0.35. I didn't expect data cleaning to have such a impact on the generalized sentiment scores.

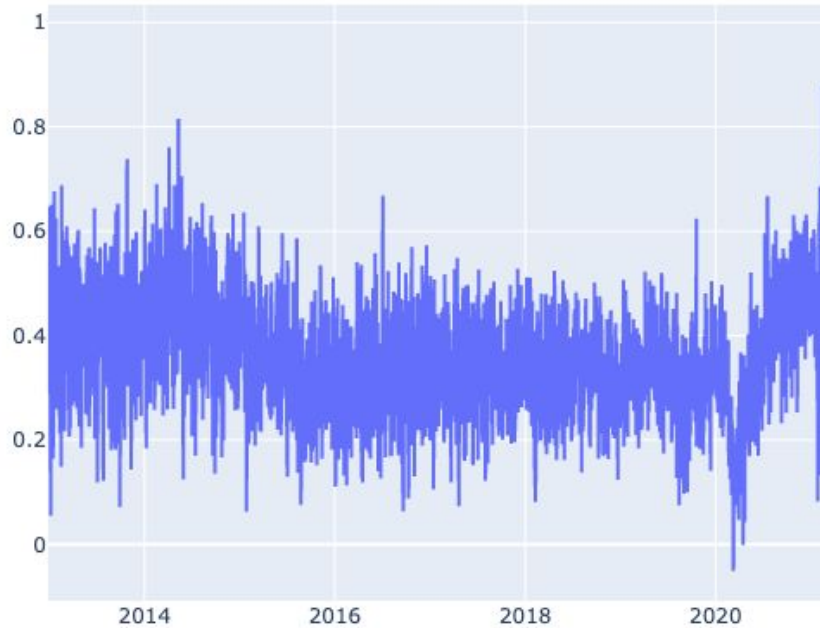
Selftext General Sentiment Over Time By Day



It is important to observe how cleaning the data affects the trends of the data and the general score changes overall.

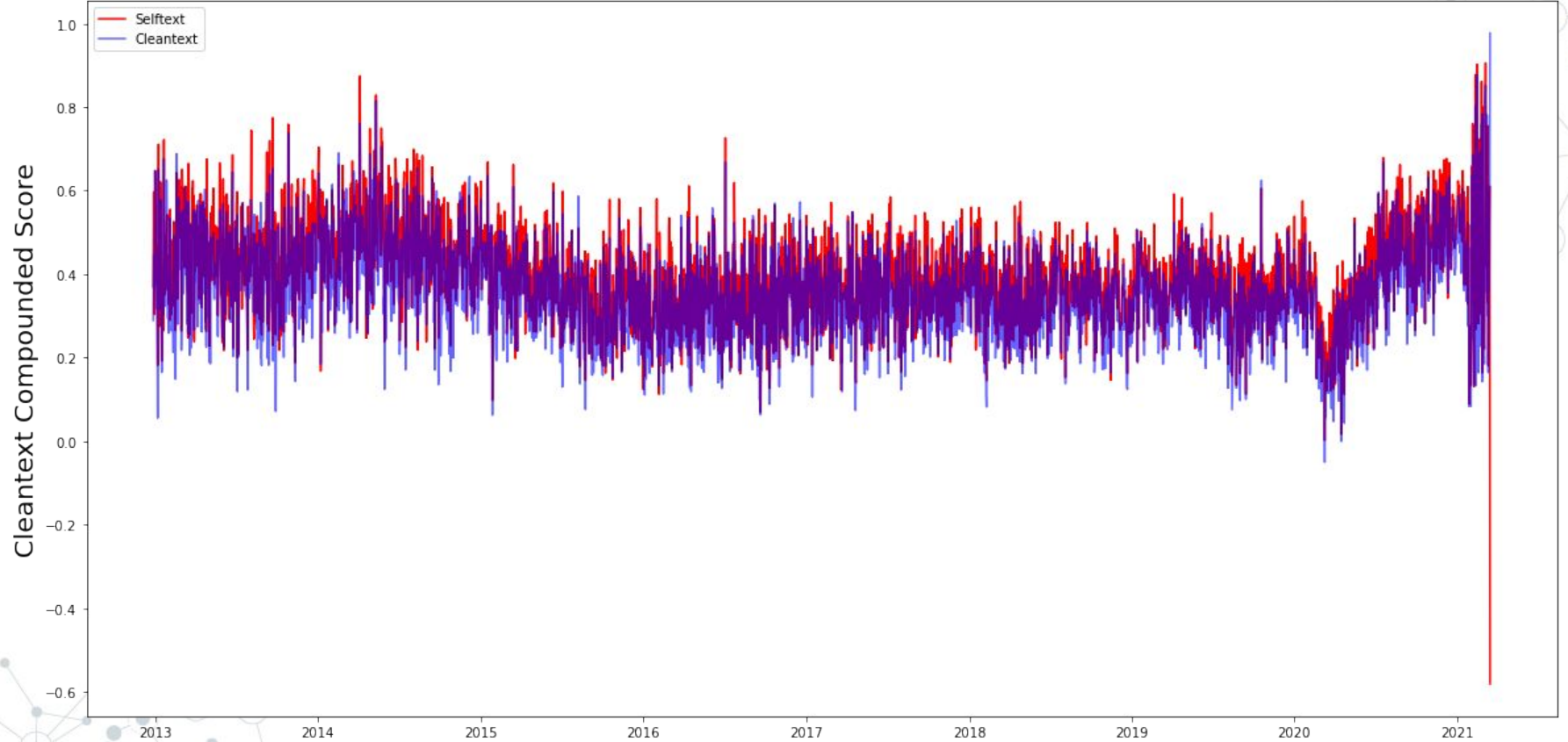
There is a noticeable drop off in the self text chart of sentiment over time by day.

Cleantext General Sentiment Over Time By Day

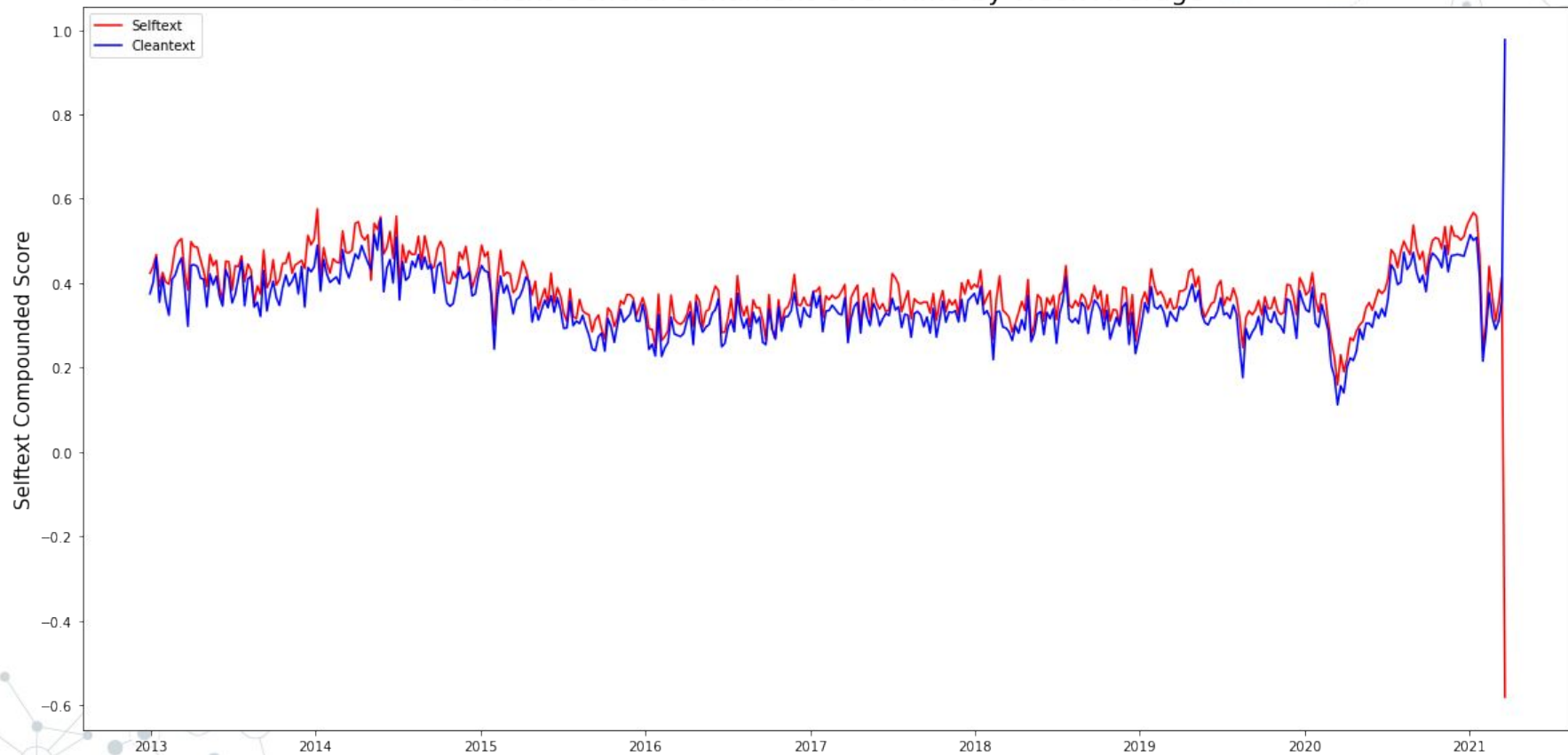


By cleaning the data, we actually see more volatility in general sentiment, and the range widens. Interesting enough, the major drop towards the end in the self text chart is now increasing and more centered in the middle of the distribution.

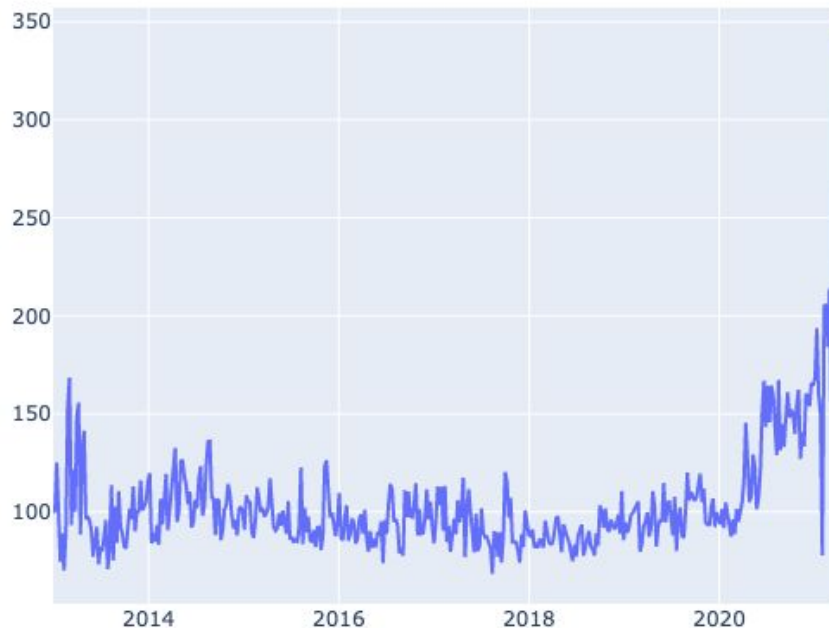
Selftext and Cleantext General Sentiment Over Time



Selftext General Sentiment Over Time by Week Average

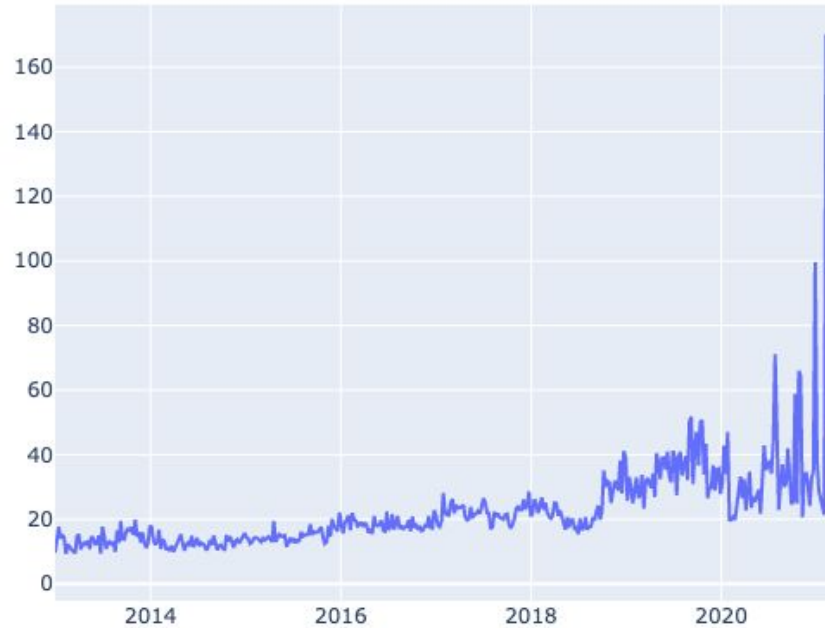


Self Text Wordcount Over Time By Week



Across the 4 different subreddits I gathered from, a very interesting trend emerges. The average word count in submissions is increasing, starting around 2020. This shows engagement behavior is changing and evolving, and that maybe more people are getting seriously involved in the market.

Self Text Number of Comments Over Time By Week



This chart shows that the number of comments in response to submissions is also increasing and getting highly volatile after the year 2020. Interesting behavior that warrants further investigation.

# Model Selection

For this particular problem, I chose to pursue a **RNN time-series** model, 1 **ARIMA** model, and 1 **SARIMA** model. My objective is to test how each model performs on daily stock data alone. Then, introduce the sentiment scores gathered relevant to the stock or index, and test the performance of the model once more. I am trying to quantify just how impactful or significant this reddit data can possibly be in effecting price. If the performance of the model improves, it will tell me that adding the sentiment scores on a daily basis actually has some significance.



# RNN: GME Stock Data

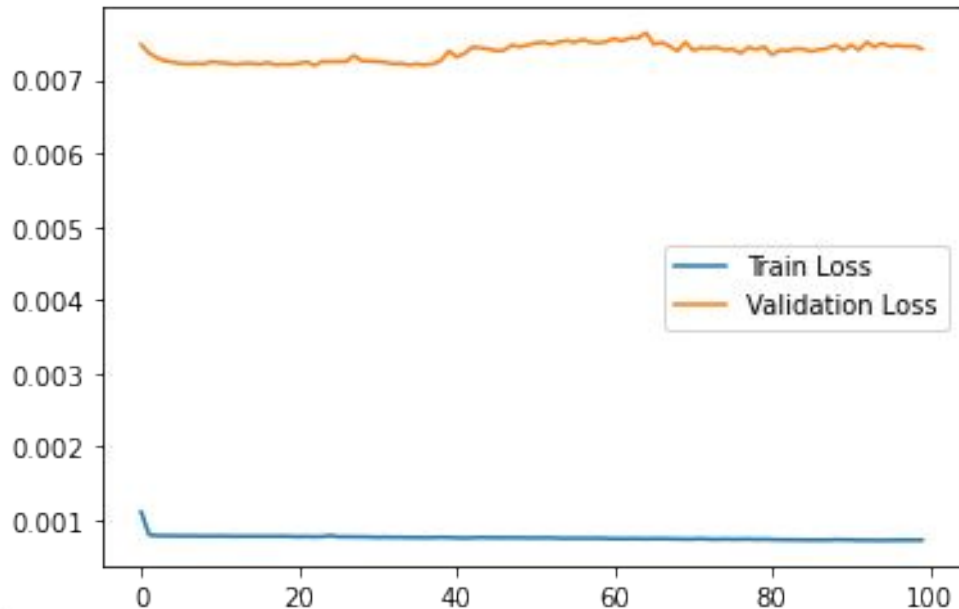
```
%%time
model = Sequential()
# 1st GRU Layer: Accepts a sequence of 7 observations
model.add(GRU(7, input_shape = input_shape, return_sequences = True)) # Output the sequence to
# 2nd GRU Layer: Accepts a sequence of 7 observations
model.add(GRU(7, return_sequences= False)) # In final recurrent layer output sequences = FALSE
# Hidden Layer 1
model.add(Dense(4, activation= "relu"))
# Hidden Layer 2
model.add(Dense(12, activation= "relu"))
# Output Layer
model.add(Dense(1, activation = "linear"))
# Compile the Model
model.compile(optimizer = Adam(lr = 0.0005),
              loss = "mse",
              metrics= ["MSE"])
history = model.fit(train_sequence,
                    validation_data = test_sequence,
                    epochs= 100,
                    batch_size=16,
                    verbose=1)
```

# RNN: GME Stock Data

The RNN architecture above allowed me to reach validation scores MSE of around 0.008. I need to do more research and create predictions on the test set and chart it out to see how well the RNN with financial data alone performs. Then test it against RNN performance with sentiment data.

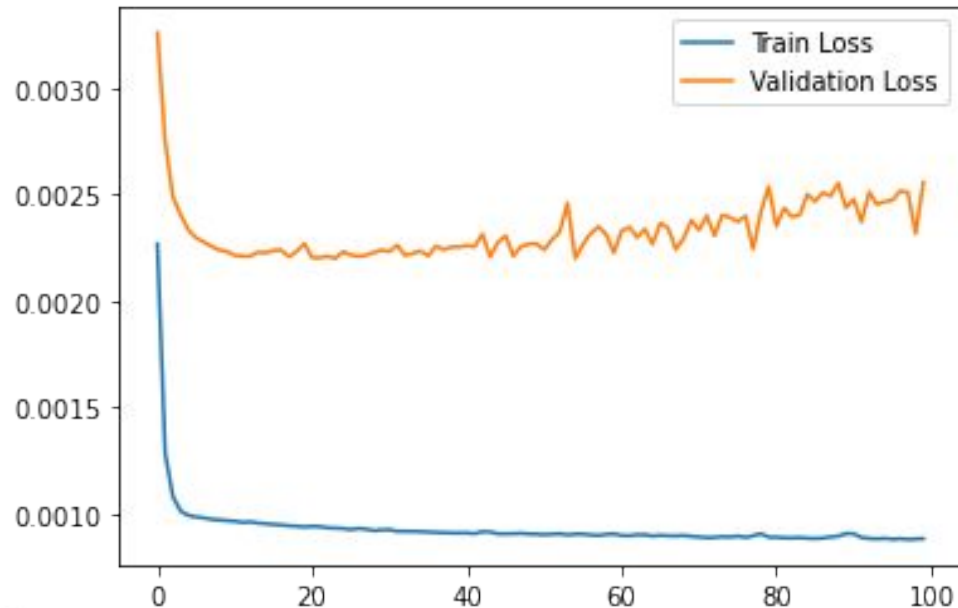
Tested early stopping, increased hidden layers, and varying node count.  
Tested varying layering techniques with GRU layers.

# RNN: GME Stock Data



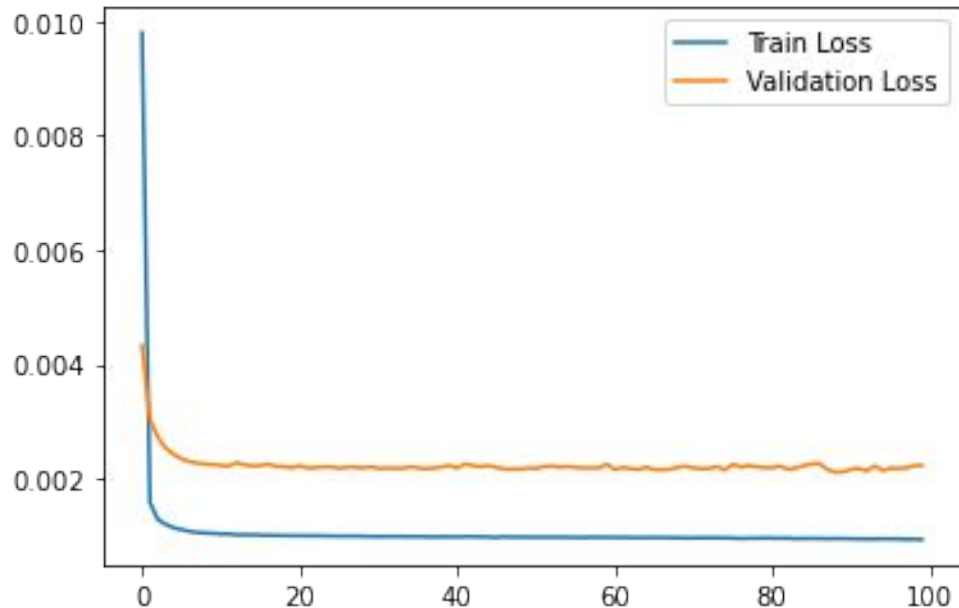
I have been unable to get the validation loss to converge with the train loss. Getting the validation loss to decrease has been challenging, but I have learned a lot about possible ways that I can either increase complexity, or add more financial market data to give the RNN more to work with.

# RNN: TSLA Stock Data



Something that I find very interesting is that with another stock like tesla, I was able to reach convergence and had more predictable results.

# RNN: TSLA Stock Data + Sentiment



Something that I find very interesting is that with another stock like tesla, I was able to reach convergence and had more predictable results.

# Summary and Evaluation

Thus far, predicting financial movement with financial data alone doesn't provide fidelity. I am setting up the RNN / SARIMA to consider sentiment for just GME, TSLA, SP500, Dow, and Nasdaq.

# Next Steps



## Web App

Acquire larger dataset and instantiate searchable web app.



## Much Larger Dataset

165,000 submissions doesn't even begin to capture the activity taking place on subreddit investment discussion.



## Add Financial Data

Considering gathering further financial, such as Investors Daily scores and analyst scores, as another feature for the models.



## Further Data Cleaning

Need to clear out the static and explore other ways of aggregating the meaning in submissions.



## Value Assignment

Custom dictionary with terms and associated values + weights based on financial knowledge. Test BERT.



## Add Comments

Comments are a significant aspect of discussion and conversation. Capturing this data would be essential for understanding the subreddit markets.

