Sentiment analysis of financial tweets to predict change in stock prices

Nicholas Szczepura, James Ko, Yefim Shneyderman

Abstract

We attempted to predict daily changes in stock prices using sentiment-labeled Twitter data from the corresponding time period. We scraped 26,321 tweets, each of which referenced one of 6 different companies, and performed sentiment analysis on them using VADER. We then fed the VADER sentiment scores, along with other tweet metadata such as number of comments, number of retweets, and number of likes, into various machine learning models that attempted to (1) classify the direction of stock price change (up or down) and (2) predict the amount of change. Overall, we found no statistically significant indication that our model was successful at predicting either metric, but have identified many important bottlenecks to tackle in future research.

Introduction

While lots of data on public opinion and company news is publicly available on social media platforms, newspapers, and other forums, the stock market remains highly unpredictable, even with many indicators of future movement and trades being based on this very same content. On May 1st 2020, Tesla CEO Elon Musk tweeted “Tesla stock price is too high imo” and received 196k likes with 35.1k comments -- coinciding with a drop in Tesla stock price of 17.9% within 24 hours.[1] In 2017 John McAfee, CEO of McAfee Associates, began investing in cryptocurrencies and would post his "coin of the day" for his 655,000 followers to pick up. On December 15, for instance, McAfee tweeted [2] about a coin called SAFEX which then spiked 92% from $0.014 to $0.027.[3] It is evident from further examples just like these that opinions and information expressed over microblogging sites like Twitter can play a significant role in actually determining what the price of a mentioned stock might look like within a short period of time and potentially make those that capitalize off of it significantly wealthy. Consumers frequently take to Twitter to express complaints about a company involved in some product which could drive prices down, or to generate hype about a recent announcement that could be expected to drive up the prices.

This paper investigates whether or not it is possible to directly predict the daily change in a company's stock price based on the contents of tweets mentioning the company during the corresponding time period. In doing so, we are also looking at how much of a measurable effect the day-to-day transactions of Twitter have on the stock market, and whether dramatic changes like the ones given as examples are common occurrences or merely flukes that rarely happen. We examine 6 companies that have a large presence on Twitter, 2 of which are high cap, 2 of which are medium cap, and 2 of which are low cap. We try to determine if the stock prices of these different company types are more susceptible to change based on tweet sentiments. Ultimately, the research question we aim to answer is: is there a correlation between tweet sentiments and daily stock changes or not? If there is, how effectively can we predict the stock prices?

Related Work

In 2016 Tahir Nisar and Man Yeungi published a paper titled "Twitter as a tool for forecasting stock market movements” [7] where they examined tweets from a time window preceding a UK political event and collected 60k tweets around certain hashtags in the elections to look at daily changes of

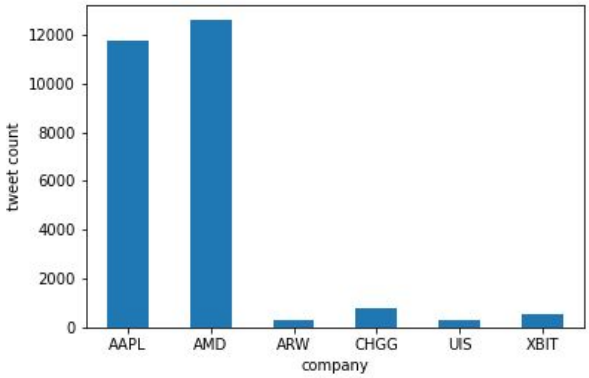
the stock market. They found that there is evidence of a small correlation between the mood of the general public and short-term market changes, but could not prove anything of significance due to small sample sizes. Their approach in using a sentiment classifier Umigon, is similar to our approach to use VADER, but they performed further filtering to get 21.44% positive, 14.30% negative, and 64.27% neutral tweets. The hashtags they used were targeted and very political hashtags such as #toryelectionfraud, #mayoralelections, #VoteConservative, #iVoted, #PollingDay and #LondonElects -- all of which would be expected to have some very sentimental tweets. In addition to filtering data by location, they also went through the steps to remove fake accounts, spam (anything containing 3 or more hashtags and URLs), bots, or anything else that they thought would not resemble an actual human's opinion. While they noted that there was a positive trend in correlation between mood and change based on statistical analysis, they also cautioned that the small sample size of their experiment prevented them from confidently drawing any conclusions. This conclusion is quite similar to ours as neither research project seems to get concrete proof of a definite correlation, but their dataset had a far better spread of sentiment than the financial tweets we scraped.

There are similar projects on Kaggle that attempt to make use of sentiment data to predict stock prices. For example, the Two Sigma challenge asks users to predict the stock movement of certain companies given news headlines and past market data about them.[4] Various notebooks make use of sentiment data that is then fed to deep learning models in order to make their predictions for this challenge.[5] [6]

Data

We selected 2 large cap, 2 mid cap, and 2 small cap companies to focus on for our study because we wanted to compare the performance of our model on different types of companies. We gathered a total of 26,321 tweets tagging these companies made in the 20 weeks before the start of the COVID-19 pandemic, from 09/29/2019 to 02/15/2020 inclusive, as we expected that stocks after the start of the pandemic would be far more unpredictable. Each of these tweets use a Twitter feature called *cashtags*. Cashtags such as "$AAPL" and "$AMD" allow users posting tweets related to financial instruments to link their tweets to the respective company. The number of tweets for each company is shown in the figures below.

STOCK	# TWEETS
AMD	12608
APPL	11774
CHGG	819
ARW	282
UIS	312
XBIT	526
Total	**26321**



Number of tweet counts per company

We obtained 576 rows of stock data for the given time period, which included opening and closing prices, from Yahoo Finance. Unfortunately, 140*6-576 = 264 rows of data were missing, so we could not train on these or generate predictions for certain companies on these days, and chose to not include tweets falling within the corresponding time period in our model inputs.

DAILY CANDLESTICK PRICE CHARTS:

AMD

AAPL

ARW

CHGG

UIS

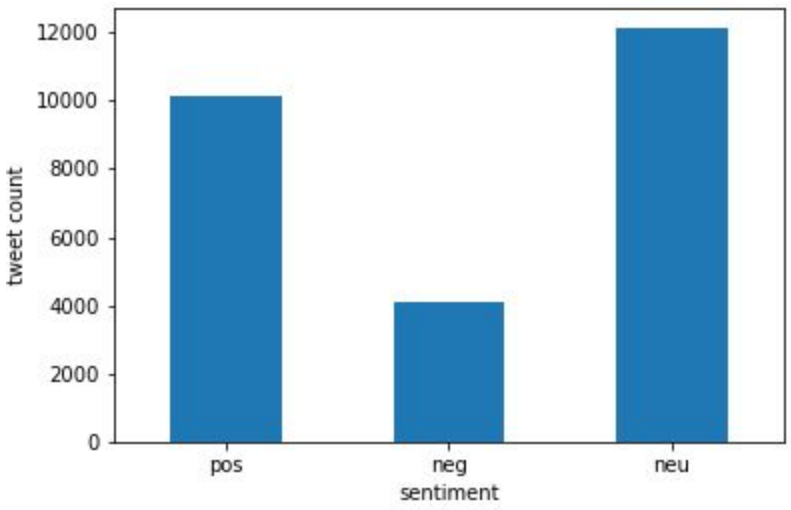
XBIT

Method

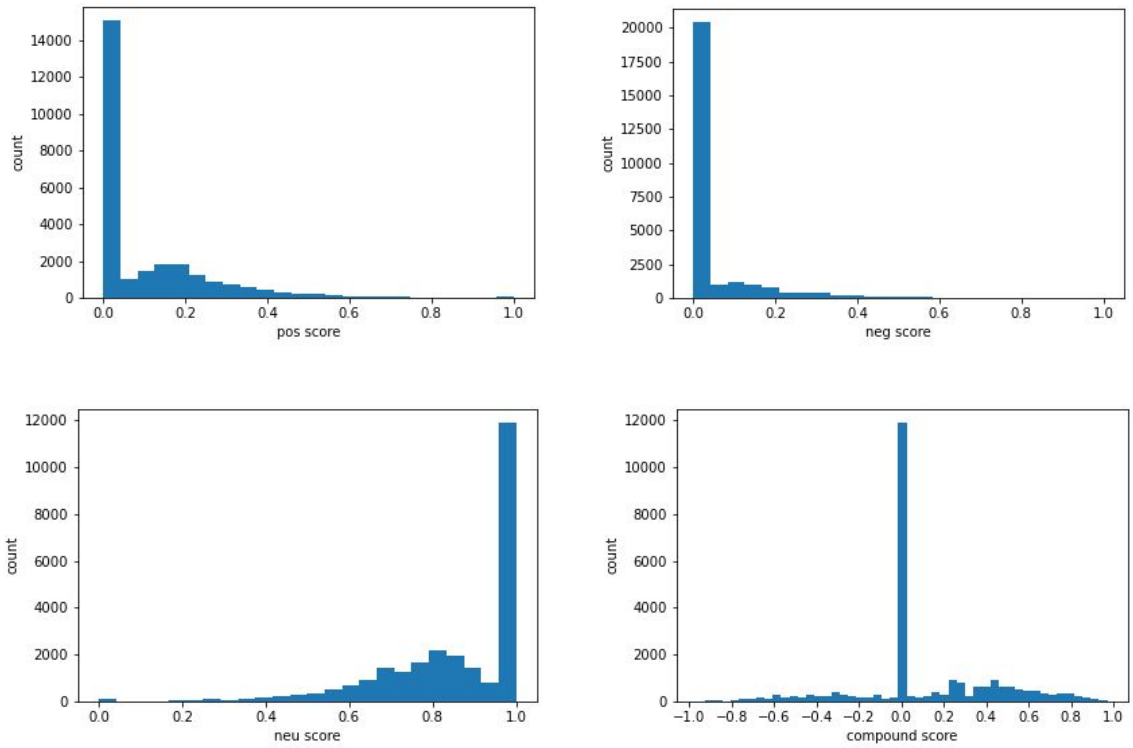
First, we attempted to use Tweepy to gather tweet data. However, we encountered issues when we realized that Twitter limits developers to 5,000 historical tweets per month using their full archive API. It is possible to increase this limit to 1.25 million tweets per month by buying Premium API access from Twitter, but this is very costly.

We instead decided to scrape data using the Selenium library for Python. For each company and each day, we queried Twitter for up to 100 of the most recent tweets made on that day using that company's cashtag. We filtered out replies and retweets from our queries because those sometimes resulted in tweets that did not actually contain the cashtag. We also excluded tweets that were truncated by Twitter because they were too long to fit on the search results page. Finally, we excluded tweets that contained multiple cashtags, since their sentiment (if any) would not necessarily reflect their opinion on a specific company but the market as a whole. In addition to textual content, we were also able to extract other metadata from tweets such as number of comments, number of retweets, and number of likes.

We used VADER to perform sentiment analysis for each tweet. Before sending tweets to VADER, we cleaned them by removing URLs, mentions, and cashtags, since these features do not help with sentiment analysis. We saved the positive, negative, and neutral scores returned by VADER for use as inputs to our machine learning models. The scores represent ratios for proportions of text that fall in each category.



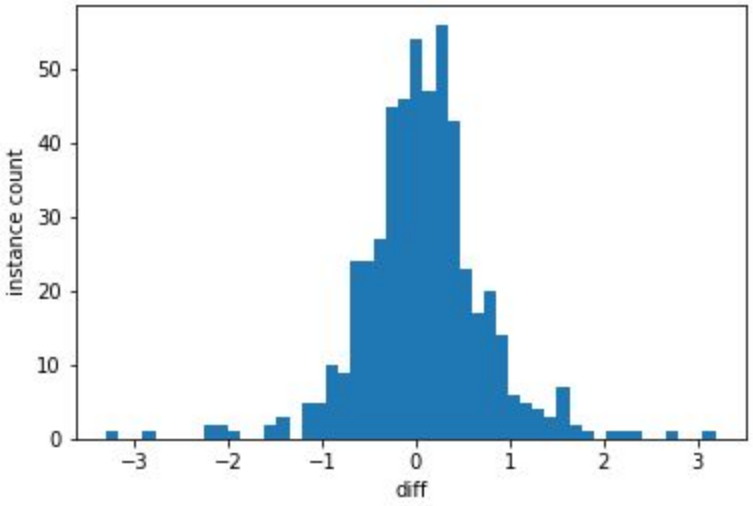
Number of tweets for each sentiment label



Distribution of pos/neg/neu/compound scores returned by VADER

From here we identified two general approaches for feeding our inputs into the models which were differentiated by whether we counted tweets individually or in aggregate for a given day. The first regression models (using PyTorch) took as input a set of 22,384 entries. Each entry contained a company's opening stock price for a particular day and information about tweets mentioning the company during the corresponding time period, including the VADER sentiment scores and the number of likes, comments, and retweets for each tweet. The next method (using scikit-learn) yielded what we considered a more logical approach to the problem, as we predicted that it might be more reasonable to take the general public sentiment of all the tweets for a specific company on a day in order to make our predictions.

For both methods, we made a train/validation/test split of 70%/15%/15%, stratified on the company each instance represented. There were 514 instances in total, resulting in a split of 359/77/78 instances. The classification task was to predict the sign of the change in stock price based on these features; the regression task was to predict the value of the change using these features and the opening price for that day. The opening price was also fed to the regression model with the reasoning that the change will likely be sensitive to the scale of the stock price.



Distribution of stock price changes across training instances

In PyTorch, we used a GRU to process the input tweet data since the number of tweets could vary from day-to-day. The outputs of the GRU at each timestep were taken to be "weights" that represented the tweet's overall contribution to the final prediction; we were hoping that the network would learn to ie. place more emphasis on the sentiments of tweets with more likes/comments/retweets. The weights were passed through a fully-connected layer to convert them to scalars, summed, and passed through a sigmoid function to get a prediction probability for the classification. The architecture for the regression model was exactly the same, except the opening price was concatenated to the input for each layer of the GRU and the final output was not passed through a sigmoid function.

For classification, we measured the performance of our model with cross-entropy loss, while for regression we used MAE loss. We tried other losses including MSE loss and mean absolute percentage error (MAPE) loss for regression, but they seemed to either encourage the model to predict numbers very close to 0 (with regularization) or blow up the loss function due to very incorrect predictions (without regularization). After some trial-and-error, we decided to use a learning rate of 10^-3 and a weight decay (aka regularization hyperparameter) of 10^-3. The GRU had 2 layers with 16 hidden units in each layer. We ran the training process for 10 epochs. For both tasks, we saved the model that showed the best performance on the validation set and evaluated it on the test set.

In the scikit-learn model, we pulled out all of the tweets for a single stock on a single day and computed a weighted average of the VADER sentiment scores where the total score of a tweet was calculated by the number of likes, comments, and retweets all multiplied by the scores. In terms of emulating general public opinion, this system works well in ensuring that outlying opinions with no votes don’t count as much as highly upvoted posts that clearly more people agree with. When it was all averaged together, we had a set of stocks for each day which was a single vector of size 4 containing the 3 average VADER scores and the open price of the stock with a corresponding output of the change in price that day. This data was then fed into a linear regression model and

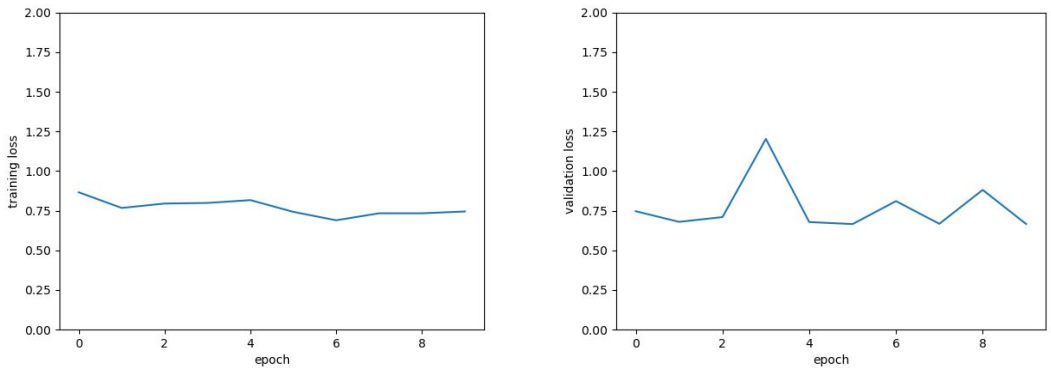
logistic regression model with C = (1 / λ) = 1.0 where the “accuracy” of the linear regression model was calculated by comparing the signs of the predicted and actual outputs.

Results

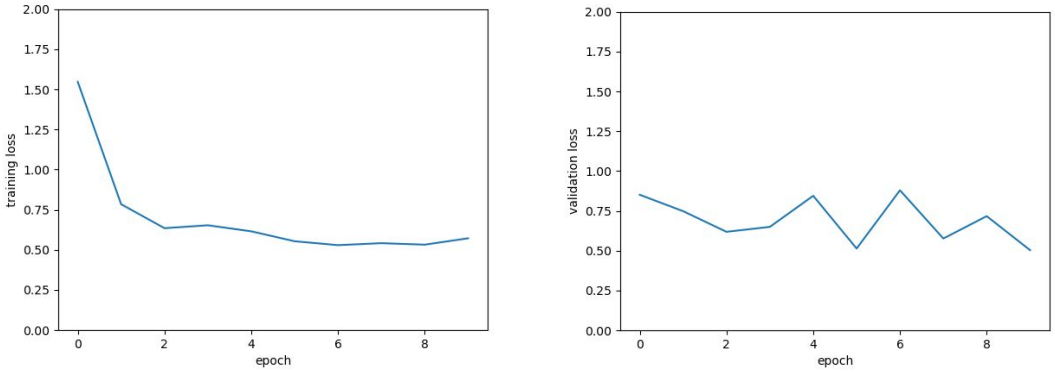
From all the 26,321 tweets, VADER produced the following sentiment scores: POS: 38.42%, NEG: 15.49%, NEU: 46.09%. This says that the set of financial tweets we looked at are generally neutral in sentiment and only about half display actual emotion. This is due to the use of cashtags which are typically used by serious traders and financial usage only, which on initial inspection seems to contain mostly unopinionated recommendations and financial terminology by twitter trading accounts. While VADER might be good for accurately finding the sentiment of regular English tweets, it seems that financial tweets form a unique subset of these tweets and a different classifier will be necessary to get a better division of POS, NEG, NEU and a better understanding of what sentiment means in relation to finance.

Additionally, in analyzing the highest-performing predicted stocks on a day-by-day basis, the closing price for AMD and AAPL was greater than the opening price 62.5% and 64.58% of the days of the study respectively. Since these two stocks comprise the bulk of the dataset, in order for our study to be deemed successful, we would want our accuracy to be better than a model that just guessed an increase all the time.

Unfortunately, we did not achieve great results with the PyTorch models. For the classification task, the minimum validation loss we achieved was 0.666 after training the model for 5 epochs. Evaluating the model on the test set, we got a loss of 0.774, an accuracy of 49%, a precision of 50%, a recall of 95% and an F1 score of 66%. This indicates that our predictions are little better than random guessing. The regression model achieved a minimum validation loss of 0.504 after 9 epochs and a test loss of 0.492.



Training and validation curves for PyTorch classifier



Training and validation curves for PyTorch regressor

The entire dataset was then fed into the tweet-by-tweet models where we output the percent error and accuracy for the linear regression model and the accuracy for the logistic regression model.

Overall Tweet by Tweet				
# Tweets (Total / Good)	**Train-Test Split**	**Linear Regression %Error**	**Linear Regression "Accuracy"**	**Logistic Regression Accuracy**
26321 / 22384	0.75 : 0.25	211.946	60.597	60.508
	0.80 : 0.20	202.099	60.576	60.465

The highest performing split was the 80% : 20% split with bearish/bullish prediction accuracy of 60.576% and a percent error of that prediction of 202.099%. The split was done chronologically as if to take historical data and then test on "modern values" to simulate how a real test of the model world work in attempting to predict the future of the stock price. The logistic regression performed approximately the same, so then we move on to the day by day models where we notice much better percent error on the full dataset, but worse accuracy.

Overall Day by Day				
# Tweets (Total / Good)	**Train-Test Split**	**Linear Regression %Error**	**Linear Regression "Accuracy"**	**Logistic Regression Accuracy**
26321 / 22384	0.75 : 0.25	139.949	50.0	50.735
	0.80 : 0.20	112.369	50.459	53.211

This dataset features the combination of all the tweets for a stock on a given day and merges their sentiment scores. With the best performing split in both cases being the 80% : 20% split, we calculated the confusion matrices below and found that the logistic regression model seems to have learned to predict an increase in nearly every case for the 20% test data.

Tweet by Tweet Overall Confusion Matrix		
	Predicted Negative	**Predicted Positive**
Actual Negative	0 (0%)	1769 (39.51%)
Actual Positive	1 (~0%)	2707 (60.46%)

Day by Day Overall Confusion Matrix		
	Predicted Negative	**Predicted Positive**
Actual Negative	5 (4.59%)	49 (44.95%)
Actual Positive	2 (1.47%)	53 (48.62%)

However, the data becomes far more interesting when analyzing the results split stock by stock.

Company Tweet by Tweet				
Company + # Tweets (Total / Good)	**Train-Test Split**	**Linear Regression %Error**	**Linear Regression Accuracy**	**Logistic Regression Accuracy**
AMD 12608 / 11498	0.75 : 0.25	142.856	69.948	69.913
	0.80 : 0.20	158.409	68.696	68.696
AAPL 11774 / 9381	0.75 : 0.25	247.635	53.623	53.879
	0.80 : 0.20	225.717	53.277	53.436
CHGG 819 / 650	0.75 : 0.25	97.668	44.172	55.828
	0.80 : 0.20	108.772	36.154	35.385
ARW 282 / 219	0.75 : 0.25	97.118	40.0	45.455
	0.80 : 0.20	107.858	38.636	45.455
XBIT 526 / 391	0.75 : 0.25	1673.725	58.163	59.184
	0.80 : 0.20	1871.008	49.367	49.367
UIS 312 / 245	0.75 : 0.25	84.678	50.0	51.613
	0.80 : 0.20	93.893	38.776	38.776

When comparing high cap vs mid and low cap stocks, it is evident that high cap stocks are easier to predict in terms of increase or decrease in price, but based on an individual tweet, it is not very easy to predict exactly what that increased price will be.

We noticed an outlier in this data, XBIT, that was giving a very high average percent error of 1600-1800%. This is likely due to the fact that XBIT trades between 8 to 10 dollars per stock, and on a day to day basis the price generally fluctuates between 1-3% change. However around December 29, 2019, Xbiotech negotiated a sale of one of their drugs for $750 million and their stock price surged 30% in a single day. A similar surge occurred a few months later, but given that our predictions are made on the chronologically last 20% of the data, it seems that these combined factors with such little data gave us such a high percent error.

Alternatively, the collected day by day data shows no noticeable trend in high to low cap stocks as AMD has very low percent error, but AAPL is nearly double and CHGG is in between. The accuracies also fluctuate and we can see that it seems to be a lot easier to predict AMD and AAPL price increases over CHGG, ARW, and UIS. Large capitalization companies like APPL and AMD are more popularly traded and thus tweeted about on a day-to-day basis. While lower capitalization companies are often less discussed, less established, and thus more subject to speculation and volatile price action. XBIT is a good example here with huge spikes in price occurring from news.

Company Day by Day				
Company + # Tweets (Total / Good)	**Train-Test Split**	**Linear Regression %Error**	**Linear Regression Accuracy**	**Logistic Regression Accuracy**
AMD 12608 / 11498	0.75 : 0.25	88.516	58.333	66.667
	0.80 : 0.20	94.892	63.158	68.421
AAPL 11774 / 9381	0.75 : 0.25	217.522	58.333	54.167
	0.80 : 0.20	204.778	55.0	55.0
CHGG 819 / 650	0.75 : 0.25	151.129	54.167	45.833
	0.80 : 0.20	175.972	42.105	57.895
ARW 282 / 219	0.75 : 0.25	136.51	38.889	50.0
	0.80 : 0.20	136.324	50.0	57.143
XBIT 526 / 391	0.75 : 0.25	243.761	62.5	62.5
	0.80 : 0.20	183.682	63.158	63.158
UIS 312 / 245	0.75 : 0.25	147.574	52.174	60.869
	0.80 : 0.20	153.218	52.632	57.895

Discussion and Future Work

There is evidently a lot of room for improvement in how we handled this task. We could have collected more metadata about each tweet, such as how many followers its author has and how many times it has been viewed to better approximate the “authority” of the opinion expressed as this might better correlate to price changes. Further, it seems that VADER did not produce the expected divisions of sentiment as it seems that over 45% of the tweets were neutral. For the future it would be worthwhile to repeat the experiment with an expanded set of hashtags that might better capture the sentiment towards a specific company at a specific time like collecting hashtags mentioning #AAPL, #apple, #iphone, although this might include a lot of data that is unrelated to how the company is performing and more about their product. Alternatively, instead of just using VADER for sentiment analysis, we could have labeled a subset of the tweet data, and used that to bootstrap machine learning models such as Naive Bayes or a BERT-based method to generate the rest of the sentiment labels. This might allow us to better tailor a classifier towards financial tweets which might not necessarily contain classic english sentiment keywords, but do contain other keywords like “fall, soar, bullish..” which indicate a sort of “financial sentiment” that VADER is unable to capture. We could have used approaches proposed in other papers like the

Nisar, Yeung paper of manually cleaning the data as well as it seems that occasionally the dataset contains tweets that don’t seem to have anything to do with the stock or company, and mistakenly tag it. Finally, we could have fed more input data to the model. For these inputs we could have used various technical indicators and/or other market data as features in addition to the sentiment. For example, we could have used a 50 period moving average (average of last 50 close prices) as a feature which is usually used to track trends. Another feature we could have included is order book information. Does sentiment align with orders in the order book, i.e. if sentiment is bullish, is there buying pressure present on the given exchange. However, for this paper we felt that this would be "cheating" since we wouldn't know if the model was actually using the Twitter data or ignoring it and using the stock data alone. We wanted to measure how well it could predict price changes using Twitter data alone and in the end found that just based on traditional sentiment, the models do not perform very well.

Works Cited

1. Musk, Elon (@elonmusk). “Tesla stock price is too high imo” May 1, 2020, 11:11 AM. Tweet. https://twitter.com/elonmusk/status/1256239815256797184?lang=en.
2. McAfee, John (@officialmcafee). “In fact, I own more SAFEX than Bitcoin. So. Trust me, I was not trying to dis SAFEX. An excellent coin. It just did not come to mind as I searched my memory for the older and more established privacy coins. Or maybe I just didn't want anyone horning in on my secrets:)” Dec 14, 2017, 11:34 PM. Tweet. https://twitter.com/officialmcafee/status/941526965781229569?lang=en.
3. Pearson, Jordan. “John McAfee Appears to Move Cryptocurrency Markets With a Single Tweet.” *VICE*, VICE Media Group, 10 Jan. 2018, www.vice.com/en/article/9knnpz/john-mcafee-twitter-coin-of-the-day-cryptocurrency-markets.
4. Two Sigma. “Two Sigma: Using News to Predict Stock Movements.” 2018. https://www.kaggle.com/c/two-sigma-financial-news.
5. B Amaral. “A simple model - using the market and news data” Sep 27, 2018. https://www.kaggle.com/bguberfain/a-simple-model-using-the-market-and-news-data
6. A Patel. “Bird Eye view of Two Sigma + NN Approach” October 26, 2018. https://www.kaggle.com/ashishpatel26/bird-eye-view-of-two-sigma-nn-approach
7. T. M. Nisar and M. Yeungi, "Twitter as a tool for forecasting stock market movements: A short-window event study," The Journal of Finance and Data Science, Volume 4, Issue 2, June 2019, pp. 101-119, doi: 10.1016/j.jfds.2017.11.002. https://www.sciencedirect.com/science/article/pii/S2405918817300247.