Group 82 Final Report

Sam Wagner, Jack Lindsay, Tyler Brown, Srikanth Basavaraju, Haohui Ding

Overview

Background

The efficient market hypothesis states that the market price of a financial asset reflects all available information. All the past available information is already incorporated into the asset price and thus an investor cannot earn a risk-adjusted return. While the hypothesis is widely accepted by the research community as the central paradigm governing the stock market in general, it still fails in reality to a certain extent. This failure motivates abundant research in stock price prediction. Fundamental analysis and technical analysis are the two methods for price prediction. The former measures the intrinsic value of a security by examining factors ranging from macroeconomic and industrial conditions to micro-company level performance. In contrast, the latter approach uses past performance metrics such as price and volume to predict future price movement. Fundamental analysis is robust in predicting long-term price fluctuations but it lacks the agility to predict shorter-run prices, which is the edge of technical analysis.

Past academic research has found several significant technical factors: Stochastic Oscillator (KDJ), Moving Average Convergence Divergence (MACD), Bollinger Band (BB), and Relative Strength Index (RSI) (Khairi et al., 2019). Nevertheless, technical analysis by itself cannot handle sudden good or bad news about a particular company. Many researchers (e.g. Nagar & Hahsler, 2012) have utilized news-based data mining to extract market sentiment to complement the technical analysis. Most of the existing papers focus on extracting sentiment from a static corpus of text such as financial reports, collection of previous financial news items, and analysts' reports (Nagar & Hahsler, 2012). With the advent of live news and micro-blogging sites such as Twitter, the sentiment could happen in real-time. Inspired by Bollen and Mao (2011), we plan to extract sentiment on particular companies from recent tweets and encode it as a factor along with other technical factors supported by past research to predict stock prices in the near future. We aim to quantify the relationship between sentiment and stock prices. In addition, we will also try to comprehend whether the sentiment is a leading, simultaneous or lagging factor of stock prices.

General Approach

As outlined in the prior section, the problem we are trying to solve is predicting stock's direction for the next few days, which is a widely known and well appreciated problem in the stock market. Ability to predict the direction of a stock very well can lead to significant financial benefit with minimal risk to the investors.

It is a widely accepted hypothesis that social media reflects and to some degree drives stocks' direction. Leveraging this premise, we intend to use Twitter, which is a well known social media platform for information exchange to capture the sentiments of the public about a company. We will scan through the tweets in set date ranges and analyze the emotion expressed in the tweet. As an example - an optimistic discussion about a company showing greater satisfaction with a product launch could be a positive emotion. A highly credible CEO of the company leaving the firm could trigger negative emotion. No news or just regular tweets with little substantial positive or negative news may trigger neutral sentiment.

Tweets were analyzed for three "sentiments": positive, neutral, or negative. However, based on the analysis of tweets, and given unlimited resources, incorporating more granular "sentiments" would likely lead to better results. It is anticipated that a positive emotion would increase the stock price, negative will reduce the stock price and a neutral will not lead to significant change.

Apart from the sentiment, it is anticipated that the number of positive or negative tweets also has a significant impact on the direction and the increase/decrease in dollar value. Based on all these parameters, this data will be fed into the model to obtain a "score" and then aggregated by day to determine how positively or negatively the public is regarding a stock on that day.

Initial Hypothesis

Based on our research and recent market movement in this area, it is anticipated that a strong positive correlation exists between the general "sentiment" of the public towards a particular stock and the corresponding change in the near-future closing price of that stock. In practice, this means that as if the "sentiment" toward a stock's outlook is positive, then that stock will reflect an increase in price in the near future. While these variables are difficult to quantify, we have chosen to analyze the recent tweets about a stock to proxy for its "public sentiment" and next-day stock closing price to proxy for "near future" price change.

We have seen in recent years the ability of major news and influential people to drastically swing the price of a particular stock, or sometimes even the larger market itself. In fact, this is often seen with the new CEO of Twitter, Elon Musk, who is commonly able to shift the price of Tesla and Twitter (and Dogecoin) with a single positive or negative tweet. In its simplest form, this phenomenon is free market dynamics at work. When there is abundant (and trusted) positive sentiment regarding a particular stock, demand for that stock will rise as more people attempt to buy it. And since the supply of stock is constant, this causes the price to rise.

The plot below shows a comparison of the total positive sentiment towards Bitcoin and the price of Bitcoin. Although the price may lag the sentiment in some cases, there still exists a strong positive correlation between the two variables. Obviously, the cryptocurrency markets behave

differently than the larger stock markets, however since several stocks that have been chosen to analyze are extremely volatile (when compared to other stocks), it is anticipated that the pattern shown below will indeed appear.

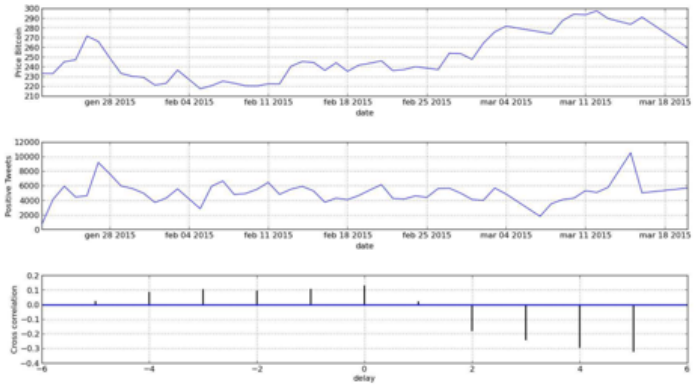

Figure 1: Comparison of Bitcoin sentiment and price

Data

Data Overview

The data used in this project originated from two points. The data that was used to determine sentiment was taken from Twitter (via an API connection), and the underlying financial data originated from yFinance.

yFinance

Luckily, given the nature of yFinance data, it can be downloaded in either daily, weekly, or monthly granularity and comes formatted in a clean way. We did have to add the ticker name to each respective CSV, and then concatenate all of the files together into one big dataframe. It is ready to be merged to our Twitter data via ticker_catergory and date.

Date	Ticker	Open	High	Low	Close	Adj Close	Volume
11/1/2021	TWTR	53.560001	55.330002	53.410099	55.110001	55.110001	14747011
11/2/2021	TWTR	55.040001	55.080002	53.775002	53.990002	53.990002	11571394
11/3/2021	TWTR	53.68	54.549999	53.099998	54.529999	54.529999	9774196
11/4/2021	TWTR	54.580002	54.84	53.18	53.68	53.68	11284746
11/5/2021	TWTR	54.110001	54.18	52.810001	53.150002	53.150002	13273810
11/8/2021	TWTR	53.450001	54.32	53.16	54.080002	54.080002	10565452
11/9/2021	TWTR	54.110001	54.93	53.154999	53.700001	53.700001	11230069
11/10/2021	TWTR	53.25	53.974998	51.75	52.330002	52.330002	18858416
11/11/2021	TWTR	52.59	52.84	51.84	51.98	51.98	13972051
11/12/2021	TWTR	52.299999	52.810001	51.919998	52.25	52.25	13287622

Figure 2: yFinance data section

Twitter

We have pulled some data from the Twitter API using cashtag queries that specifically search for given stock/ETF tickers. Within this data, we are mostly interested in the date, ticker, and full-text of the tweet. This full_text of the tweet data will serve as the input to the sentiment analysis model. The output of the sentiment analysis model will then serve as an input into the LSTM model. Below is a sample of the Twitter data:

	A	B	C	D	E	F	G	H	I
1	created_at	id	full_text	entities.u	user.id	user.screen_name	user.location	user.followers	ticker
2	Sat Oct 29 04:22:43	1.58621E+18	152% win! Congrats @Joseadyarg , 1 great trade	[]	1.37E+18	TradingITMinc	Metaverse	555	$TQQQ
3	Sat Oct 29 03:42:22	1.5862E+18	21.80 stop, buying a 5% allocation of $tqqq on M(	[]	23473235	S_AND_P	1st Earl of Shaf	749	$TQQQ
4	Sat Oct 29 03:41:39	1.5862E+18	â€ä¸»è¦ï¼“å€ãƒ¬ãƒãƒ¬ãƒƒã‚¸ã,¿ãƒ¼ãƒªã	[]	2.949E+09	asari_wiz	æ±äº¬æ¹¾	1320	$TQQQ
5	Sat Oct 29 03:25:56	1.5862E+18	RT @mimiru_usstock: ãŠãã¤ã¿ã¿ã‚‹ã,“ðŸ¥°	[]	8.641E+17	yazawahiroki125		78	$TQQQ
6	Sat Oct 29 03:14:57	1.58619E+18	What is more risky? Investing like a Venture Cap	[]	9.467E+17	TradeUVXY	Seattle, WA	407	$TQQQ
7	Sat Oct 29 02:50:39	1.58619E+18	.	[]	1.297E+18	HazelEyesHatBoy		376	$TQQQ
8	Sat Oct 29 02:21:59	1.58618E+18	Alerts before spikes and right as big news drops	[{'url': 'htt	1.565E+18	philip30964		12	$TQQQ
9	Sat Oct 29 02:07:57	1.58618E+18	Alert on $TQQQ delivered at 8:42AM CDT	[]	1.44E+18	KXTrading_		132	$TQQQ
10	Sat Oct 29 02:00:38	1.58618E+18	Bullish earnings. We were on the right side of	[{'url': 'htt	1.44E+18	MURALIE622		164	$TQQQ

Figure 3: Twitter API data section

Our key variables were the date, the overall sentiment (percentage of positive tweets out of all tweets) about the stock, and the price of the stock.

Data Cleaning

Due to the sources of our data , the data cleaning process was relatively straight-forward. The data from the Twitter API was structured and required little modification. From there, sentiment analysis was run and added as a column to the dataset with the label of the sentiment (positive, negative, or neutral). Finally, the overall sentiment was aggregated to a daily basis by looking at what percentage of the day's tweets were positive.

There was a decent amount of data formatting that had to be done in order to get the data in an LSTM friendly format. The most resource intensive was designing it to fit into a 1D array, yet maintain the value of what we perceive or assign as dates (since LSTMs have no concepts of time). Thus, once the LSTM had made its predictions based on the array, artificial dates had to be added to the predictions to represent dates in the real world. There was a bit of normality and seasonality testing done, but none of that required additional cleaning or munging (see Figure 4 below):

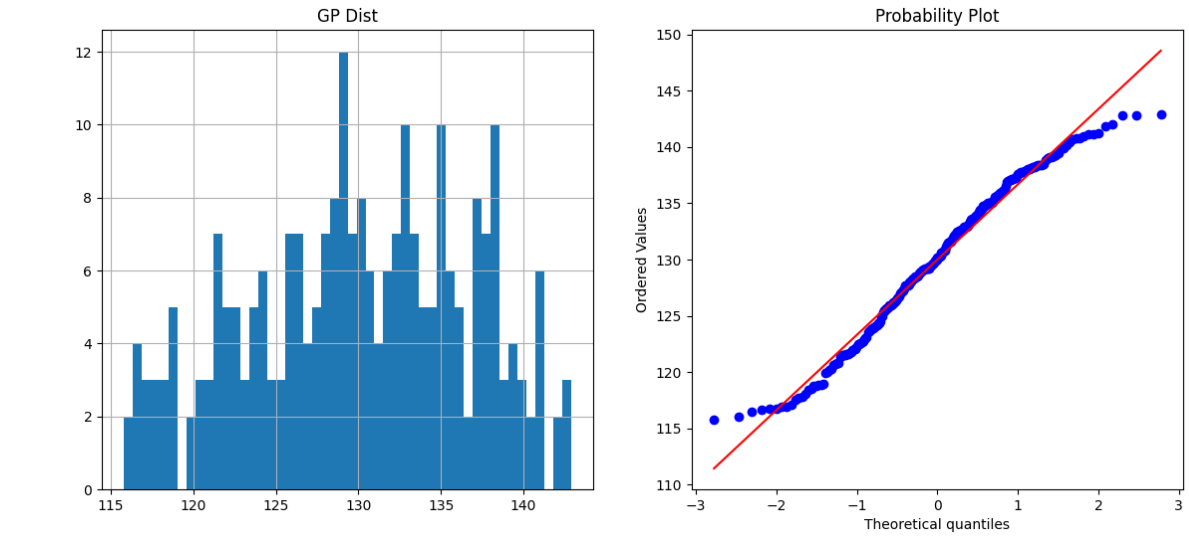

Figure 4: EDA plots

Initially, tweets were raw text, but we converted the tweets into daily overall sentiment in terms of the count of the positive tweets out of the count of all tweets. So, our feature engineering used a sentiment analysis model first to label each tweet as positive, negative, or neutral. Next, we calculated the percentage of sentiments that were positive to aggregate to a daily overall sentiment. This daily percentage positive sentiment was then used for the pricing predictive model.

Modeling

Sentiment Analysis Transformer

Transformers have become one of the most important tools in a natural language processing practitioner's toolkit. Here, we will follow suit in that trend. Transformers are essentially the evolution of LSTM and GRU (Gated Recurrent Unit) models as they relate to natural language processing. Researchers at Google discovered that the memory cells created in LSTM and GRU are not necessary if one can create a natural language processing model where the model recognizes the importance of each word in the sentence to predict the next word. For instance, a transformer reading, "My breakfast in Paris consisted of …" is going to realize that the words "breakfast" and "Paris" are very important in order to predict the upcoming words. From there, the model can predict what the next words will be. Additionally, these transformers use this attention mechanism in an array of ways, and many transformers can serve several tasks. For instance, the popular GPT-3 transformer can perform question-answering, text summarization, and machine translation. Researchers believe that part of transformers' strength is their generalization to the understanding of language. Hence, many use-cases arise due to their training.

Another benefit of transformers is that they lend themselves well to transfer learning. So, a model trained at Google or Facebook can be readily available for anyone to use because they are often quite generic and able to take on many natural language processing tasks. This is the approach we will take because training a transformer from scratch takes a lot of time and a lot of computing power… both of which are assets we do not have in abundance. Due to this, we found a transformer model trained to find the positive vs. negative vs. neutral sentiments in Tweet data, and we applied it to our tweets to discover their sentiments.

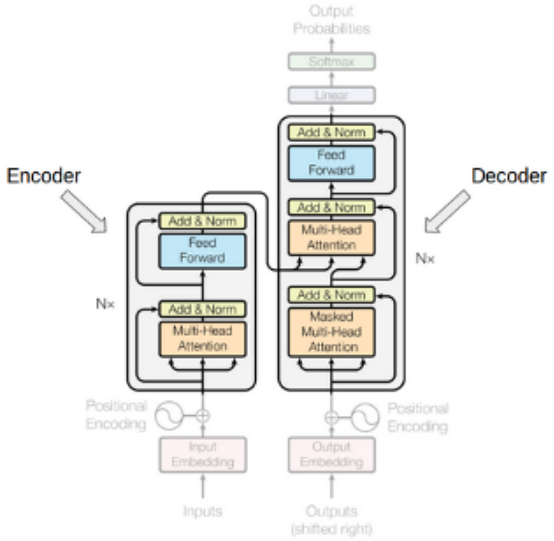

Figure 5: Sentiment Analysis Transformer infographic

After comparing available pre-trained sentiment analysis models, we selected "cardiffnlp/twitter-roberta-base-sentiment-latest" because it has already been trained on over 124M tweets. So, it was trained on a large corpus of applicable text, and the hyperparameter optimization was done previously. However, given that this is a large neural network model, we had to install a CUDA version that works with the version of PyTorch this sentiment analysis model is built with. This enabled us to make inference on several hundred thousand tweets in less than a day.

LSTM / BiLSTM

One of the methods we used in our analysis of stock trends is the LSTM (Long Short-Term Memory) model. The LSTM model is often touted as a new and improved Recurrent Neural Network (RNN). The LSTM, much to the credit of its name, is a type of RNN that remembers information over a long period of time, thus making it more capable of accurately predicting stocks that have more complex patterns than simple seasonality. LSTMs are particularly useful in

predicting data where a plethora of factors exist that affect the value of a given number, yet it would be nigh impossible to account for the aforementioned factors. In layman terms, the LSTM model sees patterns without needing the causative factors explicitly stated - though it can be helped by adding an array in parallel (such as sentiment based binary values or one-hot-encoded arrays that go in parallel to the main array (closing price)). Due to the nature of LSTM models, we had to train an individual model for each ticker we chose (HD, IBM, JNJ, TQQQ, and UNH) – resulting in 6 semi-unique LSTM models. We monitored the accuracy of our predictive model primary based on a combination of MSE (Mean Square Error) and MDA (Mean Directional Accuracy) - the latter simply because, in terms of stocks, unless we are buying/selling futures, we simply need to know if the number is going up or down, not necessarily the exact amount by which it goes up or down.

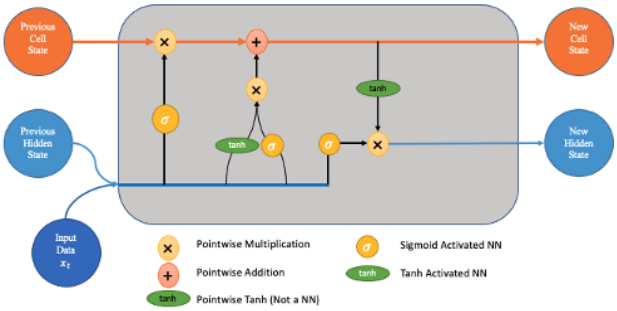

Figure 6: LTSM infographic

The LSTM alone did a decent job at predicting the stock prices:

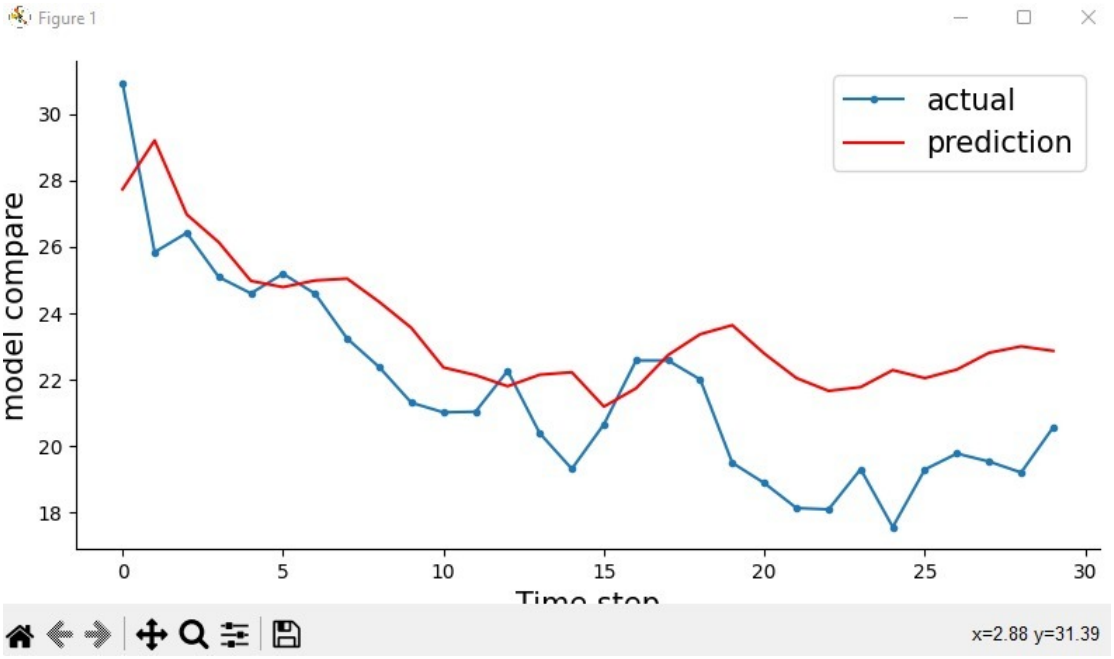

Figure 7: LSTM model prediction performance

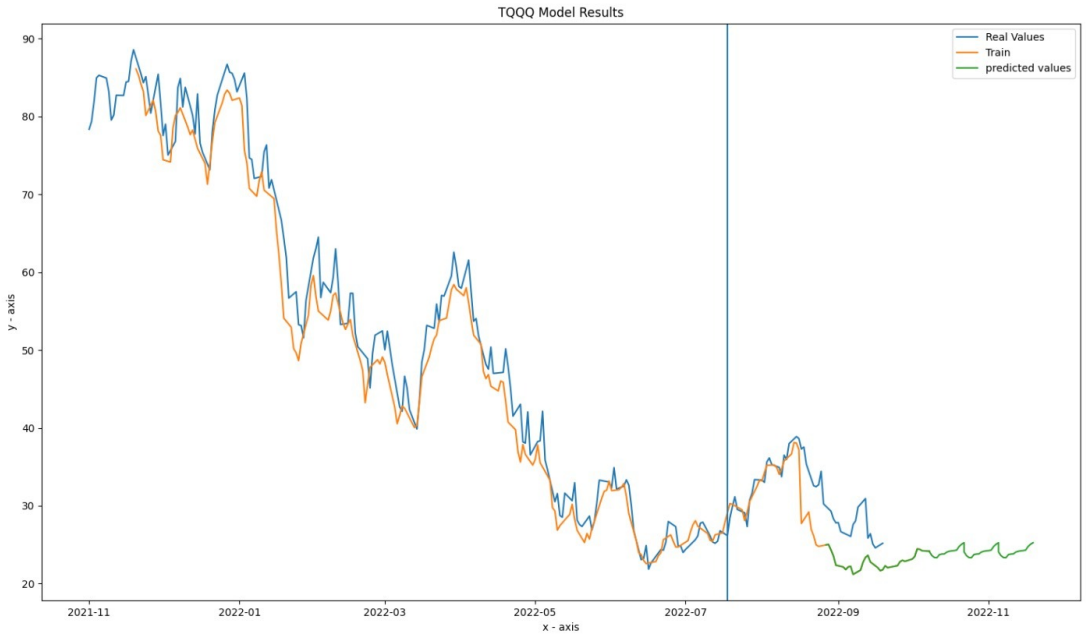

Figure 8: LSTM model prediction performance

This produced a MDA (Mean Directional Accuracy) of roughly 50%. After training an LSTM model for each ticker separately, the sentiment layer was applied - after some basic logic was applied (higher than 80% positive sentiment and directional flow of positive was detected), the model would prescribe a buy.

Date	predicted_val	model_no	Ticker	predicted_val_sent	Close	year_month_day	pct_pos_sentiment	val_higher	sent_higher	buy_function
11/2/2022	135.17564	model_0	IBM	0.88188374				FALSE	FALSE	inconclusive data
11/3/2022	129.10408	model_0	IBM	0.8847263				FALSE	TRUE	inconclusive data
11/4/2022	127.78149	model_0	IBM	0.88063693				FALSE	FALSE	inconclusive data
11/5/2022	126.91457	model_0	IBM	0.8821688				FALSE	TRUE	inconclusive data
11/6/2022	126.3173	model_0	IBM	0.8839232				FALSE	TRUE	inconclusive data
11/7/2022	127.29675	model_0	IBM	0.8844104				TRUE	TRUE	BUY!
11/8/2022	128.0398	model_0	IBM	0.8846524				TRUE	TRUE	BUY!
11/9/2022	128.60146	model_0	IBM	0.8863945				TRUE	TRUE	BUY!
11/10/2022	128.64897	model_0	IBM	0.88974327				TRUE	TRUE	BUY!
11/11/2022	126.39534	model_0	IBM	0.8940516				FALSE	TRUE	inconclusive data
11/12/2022	127.026344	model_0	IBM	0.8912129				TRUE	FALSE	inconclusive data
11/13/2022	129.70291	model_0	IBM	0.8931475				TRUE	TRUE	BUY!
11/14/2022	132.82388	model_0	IBM	0.8879515				TRUE	FALSE	inconclusive data
11/15/2022	134.38596	model_0	IBM	0.88875204				TRUE	TRUE	BUY!
11/16/2022	135.17564	model_0	IBM	0.8912137				TRUE	TRUE	BUY!

Figure 9: Sample of raw results data

As seen in Figure 9 above, the model would recommend buying 7 times over a 15 day period - 5 of which would have resulted in profit, making this method 71% effective, in an anecdotal sense.

Unfortunately, the LSTM model proved to be too difficult for us to set up with multiple inputs. The multi-dimensional LSTM model is rather controversial in the data science world, and the set up for a true multivariate LSTM model is very complex. We found that, rather than waste resources forcing a cube to fit in a circular hole, we could achieve similar or near similar results in accuracy by applying basic logic to a standard LSTM output based on sentiment analysis and directionality. Training 1 model for every ticker also resulted in unusable predictions on our test data (based on MDA), which was predicted, so we chose to train 5 independent models. This did, however, result in unpredictable outputs that, given the stochastic nature of LSTM models, had varying degrees of accuracy. This, and the fact that LSTMs are something of a black box brought us to our wits end, thus we decided to pivot to regression.

Regression

In order to address the issue outlined above regarding the difficulties in using an LSTM model and to provide a baseline for comparison, linear and logistic regression were performed in sci-kit learn.

Hyperparamter optimization was performed on the logistic regression models by utilizing a standard grid search methodology with five-fold cross-validation in sci-kit learn. The “C” and “solver” logistic regression parameters were analyzed and searched over four and five values respectively. This optimization yielded extremely positive results, increasing the accuracy of the logistic regression models by ~5% on average.

Results

The sentiment analysis model yielded intuitive results. We were able to use the percentage of positive tweets as a predictive feature for the price of a stock. Figure 10 below shows the results of the logistic and linear regressions run in sci-kit learn. The best parameters of the grid search hyperparameter optimization are also displayed as well as the resultant coefficients for each linear regression model (percent positive sentiment, previous day volume, and previous day adjusted close price respectively). The logistic regression models for all five stocks were able to achieve greater than 50% accuracy and the linear regression models for all five stocks were able to achieve an r^2 value greater than 0.9. This means that over 90% of the variance was able to be explained by the models. These results are sufficient to claim that our original hypothesis - that there exists a positive correlation between public sentiment and stock price movement - has been confirmed by this analysis.

```
IBM
Log Reg Best Params:  {'C': 8, 'solver': 'sag'}
Log Reg Best Score:  0.5599999999999999
Lin Reg R2:  0.9451781923914718
Lin Reg Coeffs:  [[3.86282200e+00 5.09467815e-08 9.49444437e-01]]

JNJ
Log Reg Best Params:  {'C': 4, 'solver': 'newton-cg'}
Log Reg Best Score:  0.52
Lin Reg R2:  0.9185531941709131
Lin Reg Coeffs:  [[2.01005825e+00 5.68080136e-08 9.46050644e-01]]

HD
Log Reg Best Params:  {'C': 4, 'solver': 'liblinear'}
Log Reg Best Score:  0.5942857142857143
Lin Reg R2:  0.9780507380324713
Lin Reg Coeffs:  [[2.41763230e+01 1.54313991e-07 9.97699260e-01]]

TQQQ
Log Reg Best Params:  {'C': 8, 'solver': 'newton-cg'}
Log Reg Best Score:  0.6285714285714286
Lin Reg R2:  0.9830235893050341
Lin Reg Coeffs:  [[1.21214297e+01 2.46337459e-09 1.00038136e+00]]

UNH
Log Reg Best Params:  {'C': 8, 'solver': 'newton-cg'}
Log Reg Best Score:  0.5771428571428572
Lin Reg R2:  0.937699701715975
Lin Reg Coeffs:  [[1.60875898e+01 5.15142724e-07 9.80763638e-01]]
```

Figure 10: Logistic and linear regression results

As discussed in the project overview, these results could potentially provide a methodology to make significant financial gains in the stock market either for an individual investor, or a fund manager. Additionally, accuracy could likely be enhanced in larger hedge funds or investment banks given their large budgets and advanced technology capacities. A central axis to these results is the ability to gauge the public sentiment via Twitter data - being able to do this at high frequency would lead to more confidence in the sentiment analysis, and therefore more confidence in buy or sell signals from the stock price prediction.

Although the noted results were very encouraging, this project was not without its speed bumps. The Twitter API had issues when pulling data for the most discussed stocks on Twitter because some of these stocks have around 5,000 tweets every day. This resulted in hitting rate limitation errors when trying to pull tweet data on these stocks. We adjusted our target stocks accordingly so that we could easily pull data from the Twitter API. Additionally, the sentiment analysis model required additional installations of a specific CUDA and PyTorch version on one of our team members' computers. This allowed us to utilize a GPU for the tweet sentiment inferences, and this GPU-based inference was 10x faster than CPU-based.

Conclusion

In this project, we used sentiment analysis to model public sentiment toward traded stocks as a percentage of positive tweets over total tweets. It served as a predictive feature for predicting stock price change direction together with other features in the linear regression and logistic regression models. The models returned a high enough prediction accuracy and explanatory power to realistically confirm our initial hypothesis. The intended multivariate LSTM model was abandoned due to the feasibility and comparative performance of simpler models (linear regression and logistic regression).

If we were to extend this project into the future and continue our analysis, we could attempt the following:

Only the direction of stock price change was considered in our models when making a purchasing decision. The amount of stock price change did not matter. To apply the models in reality, transaction costs must be taken into account. Thus, the magnitude of price change matters. A small increase in price with relatively high transaction cost could overturn our buying decision.

Another possible extension of the current implementation is to use this concept and apply it to crypto currencies, ETFs and other financial assets. Crypto currencies in particular, in recent times, have shown a lot of volatility in this area and most of it is heavily driven by sentiments. Thus, leveraging the current solution and fine tuning it towards the crypto market can help us assess whether a specific crypto currency is going to move up or down in the next 24 hours window and even potentially look at how much change we may anticipate.

Works Cited

Khairi, T. W., Zaki, R. M., & Mahmood, W. A. (2019, March). Stock price prediction using technical, fundamental and news based approach. In *2019 2Nd scientific conference of computer sciences (SCCS)* (pp. 177-181). IEEE.

Matta, Martina, Ilaria Lunesu, and Michele Marchesi. "Bitcoin Spread Prediction Using Social and Web Search Media." *UMAP workshops*. 2015.

Nagar, A., & Hahsler, M. (2012). Using text and data mining techniques to extract stock market sentiment from live news streams. In *2012 International Conference on Computer Technology and Science* (Vol. 47, pp. 91-95).

Bollen, J., & Mao, H. (2011). Twitter mood as a stock market predictor. *Computer*, *44*(10), 91-94.