Stock Trading Agent That Utilizes Number of Tweets

Anthony Chen
Jeremy Hamning

May 2018

1 Introduction

1.1 Problem Description

Our problem was to tackle the ever changing stock market. Our goal was to investigate and produce a useful stock market trading agent built by combining data from Twitter with historical stock price data. Essentially the agent needs to be able to predict the stock price each week by finding an equation that uses number of tweets that company's name popped up each day throughout the week and the past few stock prices each day that week. Specifically, given past stock price and number of tweets data as evidence/independent variables, the agent needs to predict the stock price at the end of the week and decide based on this new predicted new stock price how many shares to buy or sell to best maximize the agent's net worth from an initial budget and shares.

The prediction can stock price should follow a sort of linear equation:

$SP_6 = W_1 * T_1 + W_2 * T_2 + W_3 * T_3 + W_4 * T_4 + W_5 * T_5 + W_6 * T_6 + W_7 * SP_1 + W_8 * SP_2 + W_9 * SP_3 + W_{10} * SP_4 + W_{11} * SP_5 + B$

Where SP_i represents the stock price that day, T_i represents the number of tweets that day, and W_i and B are the coefficient variables and intercept respectively. Depending on the machine learning algorithm, our AI agent's formula will not look exactly like the equation above depending on other functions (activation functions, multiple layers, etc.) used, but the overall representation on how the AI predicts the stock price is through the equation above.

From using 6 years worth of Twitter Data and Stock Prices, we tackled the problem with a Neural Network by training the network with the 11 independent variables over a course of 5 years and then testing on the 6th year.

1.2 Importance

This problem has significant intellectual, social, and economic importance.

First, this project is of intellectual importance because it helps to investigate how social media affects stock prices. Obviously, in cases where social media is simply used as a faster means to distribute news, we would only expect stock price movement related to the news to occur earlier. However, the relationship between the stock price movement and the amount the company is discussed on Twitter is an interesting connection to research. There may not even be a cause and effect relationship but even a correlation relationship would be interesting. Tweets could be a very important indicator of a company's value as it is a means of communication between big corporations to the consumers. This is especially true the larger the market audience is for a company and how much the company relies on it's users and consumers. A recent example is Facebook and how the user data misuse with Cambridge Analytica resulted in a DeleteFacebook trend to rise in Twitter and subsequently the stock price to drop 3.3% [9]

This project has some social merit because the influence and correlation of tweets to stock price and reputation of companies can help create social changes to the professional and corporate world. Now-a-days, social media has proven to be an accurate representation of public sentiment and opinion of organizations, people, and events. If social media plays a strong influence on companies and their net worth, it can be used as tool to drive social movements and hold companies accountable to current social issues. Just in the

past year, the MeToo was a social movement to fight sexual harassment and assault was primarily held on Twitter had caused many top executives and companies in industries such as Finance, Politics, and more to rethink their policies and behaviors. The influence of social media platforms like Twitter on company branding and reputation along with stock price can affect how we communicate or choose to use social media to drive social issues and raise awareness.

This project has economic importance because modern social media could exacerbate or (if manipulated correctly) play a huge role in the future of our economy and be a key component in financial prosperity or crisis. Social Media is becoming more and more important in our time and is the main way to connect with the newer generations which represent a large percentage of the market. Nowadays, social media is one of the main ways companies market their products/services to their audience. If an influence exists and social media is used in a certain way, social media trends/movements can have a part to play in a national economic crisis. Therefore, understanding how social media affects stock market prices could help to understand the influence that social media has not only as a means of communication but also as a tool to affect the economy.

It is obvious that this kind of problem is very important and interesting. Trying to predict and quantify the sentiment of a company has been a challenge for investors for decades. And trying to predict it using social media which has only become popular within the past decade is an even larger challenge with the various forms it can take and trying to interpret modern texts and images. To even attempt at extracting the entire social media sentiment towards companies requires strong natural language processing, image recognition, and a strong word bank that maps words to a quantity representing the sentiment. The key sub-problem we are tackling from this larger challenging sub-problem will only have to deal with the number of times a company is mentioned in tweets on Twitter.

1.3 Related Literature

It has long been established that contrary to the efficient market hypothesis, stock prices are not derived solely from the fundamental aspects of the associated company, but are instead a combination of psychological and fundamental factors [5]. Since fundamental factors are easy to calculate by all parties, the focus has generally been on quantifying the sentiment of all market participants. Historically, it was slower and more difficult to do this because people had no place to openly voice their opinions for all to see. Luckily for us, social media allows the opinions of a large number of market participants to be viewed with relative ease. However, quantifying these opinions into actionable information to guide market decisions is still difficult. We will now review literature that tries to obtain actionable information about stocks using Twitter.

1.3.1 Twitter sentiment around the Earnings Announcement Events

In this paper, the authors investigate the usefulness of Twitter sentiment data when trading around quarterly reports for all companies in the Dow Jones Industrial Average [6]. They first gathered all tweets related to these companies between the period of June 1st, 2013 to June 3rd, 2016. These tweets were manually annotated by financial experts and separated into training and test sets for cross validation. The training's labeled set was used to train and tune the classifier. The classifier was a two-plane support vector machine with one classifier separating the negative tweets from neutral-positive class and another separates the negative-neutrals from the positive class. In the end, the SVM classifies sentiment among tweets as negative, neutral, or positive. Their first result was that both the number of tweets and the level of trading activity increased in the day before and after a quarterly report is released. Companies that released their results in the afternoon rather than the morning had a larger increase in number of tweets. Finally based on the SVM classifier, it was found that the Twitter sentiment on the day of the earnings announcements was accurate in predicting the stock movement. However, trying to predict the stock movement a day before the earnings announcement from the Twitter sentiment proved to have inaccurate results. This paper proved to show the importance of day alignment with the Twitter data and the change in stock price when the earnings announcements are released as a lot of sentiment in Twitter happens after-hours right before the announcements.

1.3.2 Twitter Sentiment Analysis Applied to Finance: A Case Study in the Retail Industry

Another paper relevant to this topic was written by Souza et al. in which the authors investigate whether Twitter sentiment analysis can help predict the stock prices of five retail companies better than traditional news sources [10]. Their period of investigation was November 01, 2013 to September 30, 2014. The Twitter and traditional news datasets they obtained already had sentiment and relevance analysis performed on them. They performed a hypothesis test using Granger causality and a vector auto-regressive framework to test whether positive sentiment for either sentiment source led to out-performing the S&P 500 and vice versa. The twitter sentiment analysis found a significant relationship between beating the index and positive sentiment on Twitter. In fact, they found this relationship to be stronger than the one found using traditional news sources. However, they did not find any relationship between negative sentiment tweets and stock performance.

1.3.3 Public Sentiment Analysis in Twitter Data for Prediction of a Company's Stock Price Movements

Another paper that is relevant to our research is one that was lead by Bing et al [7]. In this paper the authors collected tweets from October 2011 to March 2012. They then used association rule mining to find associations between tweets and 30 stocks from diverse sectors of the United States stock markets. For example, their mining algorithm found that Macbook and iPhone are associated with Apple. Then, sentiment analysis was performed on those tweets associated with these companies to map them into a five-level scale ranging from extremely negative to extremely positive. The result is a dataset where the opinion of Twitter users about the company and notable products and services associated with the company is captured. The net sentiment was then calculated in order to predict the stock will move based on five categories that are similar to the sentiment categories, with positive sentiment being associated with an increase in price and vice versa. They then tested this algorithm with real price data and obtained an accuracy rate 52.94%-76.12%, with the main difference in accuracy rate being the sector the stock is from. They also found this strategy obtained a higher accuracy than SVM, C4.5, and Naive Bayes, but it's unclear if the strategy is accurate enough for trading. Their research especially showed that the proposed algorithms had best prediction accuracy in industries such as IT and media. It was also found that three day intervals performed best in predicting stocks up to 3 days later. This is significant because it shows that twitter sentiment towards a company can be used as an indicator for stock price change for up to three days. This study shows that although the accuracy may not be high enough for realistic trading, it does indicate that the usage of Twitter as evidence data to predict the change in stock price contains some merit depending on the industry and time.

1.3.4 Sentiment Analysis of Twitter Data for Predicting Stock Market Movements

Finally, this research group at India Institute of Technology's goal was to observe how strong or weak the correlation between the public opinions about a company expressed in tweets was to the rise or fall of stock prices [8]. The group developed their own sentiment analyzer by extracting textual representations of the tweets in an N-gram Representation (every word sequence of length N is parsed) and Word2vec Representation (mapping of words to a 300 dimensional vector). These features were each then trained on the Random Forest, Logistic Regression, and SMO machine learning classifiers. Then using the best sentiment analyzer (Random Forest with Word2vec representation), Logistic Regression and LibSVM classifiers were made to use the sentiment values of three day periods from the sentiment analyzer and classify whether the stock price would increase or decrease. The results from both classifier showed around 70% accuracy value. From these results, it shows that there is a significant correlation with stock market movement and sentiment on Twitter. This paper further shows proof of a correlation between Twitter sentiment and change in stock price. Also, the fact that the use of a three day period for predicting stock change gave such good correlated results gives more evidence that intervals of three days is a good amount for representing the Twitter sentiment opinion of a company.

1.4 Solution Strategy

The overall solution strategy for devising an agent that buys/sells stocks based on predicted stock price from previous stock prices and number of tweets involves (1) Collecting stock price data and number of tweets data, (2) Building and training the prediction stock classifier by building a neural network that represents our problem equation and training it over five years worth of training data, and (3) use our machine learning stock predictor to predict stock prices throughout a year and buy/sell shares depending on the difference between the predicted stock price and previous stock price.

For our project we focused on testing our idea on five stocks: Amazon (AMZN), General Electric (GE), Exxon Mobil (XOM), Michael Kors (KORS), and American Airlines (AAL). We selected these five stocks because of their name recognition, size, number of tweets, and differing performance over the past five years. The data is to be stored in a CSV file for the learning agent to eventually parse when constructing the neural network. The CSVs created would have three columns, one for date, one for number of tweets, and one for the stock price.

For simplicity we are assuming no transaction costs and a budget of $1,000 and an initial 2000 shares for each company tested. We are also assuming the at the overall net worth or the balance or the number of shares can go below 0 at any point. However, if such a thing occurs, we would only analyze the time before it hits 0 and disregard any regrowth to positive net worth after the hit to 0. We will also be normalizing the stock price data, so the net worth will be calculated with this normalized price and not the actual stock price.

1.5 The Data

For our project we focused on testing our idea on five stocks: Amazon (AMZN), General Electric (GE), Exxon Mobil (XOM), Michael Kors (KORS), and American Airlines (AAL). We selected these five stocks because of their name recognition, size, number of tweets, and differing performance over the past five years.

We collected the number of times the five stock symbols were mentioned on Twitter each day between the period of March 23th 2012 and March 23th 2017. In addition, we collected the daily adjusted close price for these stocks using the Quandl API [1]. The adjusted close is the closing stock price on that day, but adjusted to account for dividends and splits. [2] Finally, the data was normalized and used to train the neural network. We made sure to choose stock prices with varying trends in stock price and/or number of tweets.

General Electric (GE) is a highly diversified multinational corporation that operates in the health-care, aviation, transportation, consumer goods markets, and others. It is a good choice for testing because it has a large number of tweets about it (daily mean = 6,412, total = 8,733,164) and slow and steady upward price movement. The number of tweets tend to have a slight constant decrease in occurrence of tweets as the stock price of GE increases.

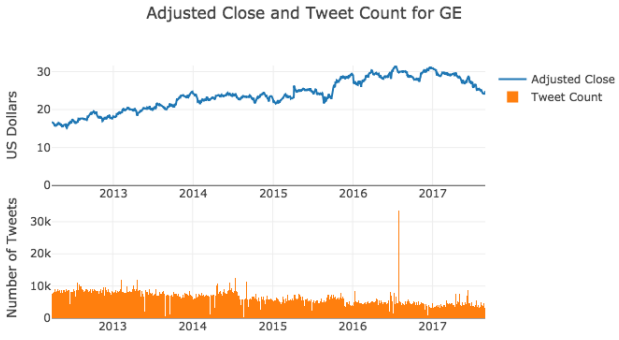

Figure 1: General Electric Tweets and Adjusted Close

American Airlines (AAL) is one of the largest airlines in the world. The overall volatility of this stock, along with the large sudden drop at the end of 2013 made it a good research subject. In addition, it has

relatively few tweets about it (daily mean = 272, total = 410,834). This is most likely because most people do not tweet about airlines unless there is an issue. In this company, it appears that AAL is fairly sporadic in both number of tweets and stock price. The stock price does appear to rise and drop three times over the past 6 years.

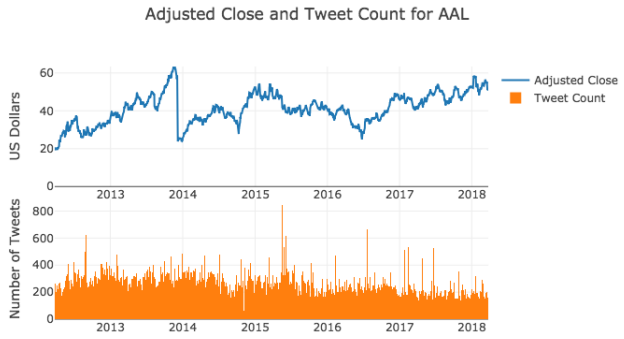

Figure 2: American Airlines Tweets and Adjusted Close

Amazon (AMZN) is the largest online retailer in the world when measured by total revenue. Besides pure name recognition, this is a good research candidate because of its long term upward movement and relatively low volatility. In addition, there are a decent number of tweets about it every day (daily mean = 1,158, total = 1,746,821). Note that the outlier in figure 3 is not a data collection error. On that day, June 16th 2017, there were 17,310 tweets about Amazon because it was announced they had bought Whole Foods. The number of overall tweets these past year also appear to increase along with the stock price.

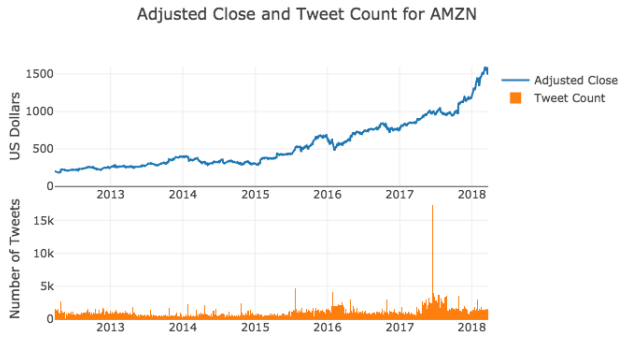

Figure 3: Amazon Tweets and Adjusted Close

Michael Kors (KORS) is a fashion design and retail corporation. It is a good research subject because of its relatively low volatility and unsteady price movement over time. In addition, it has good brand recognition and in turn a decent number of tweets each day (daily mean = 1,861, total = 2,759,536). Here the stock price is high the first few years and then drops and starts to rise again mid 2017. The number of tweets trends reflect a similarity with the stock price trend as in the first few years has a large number of tweets but then dies off in later years.

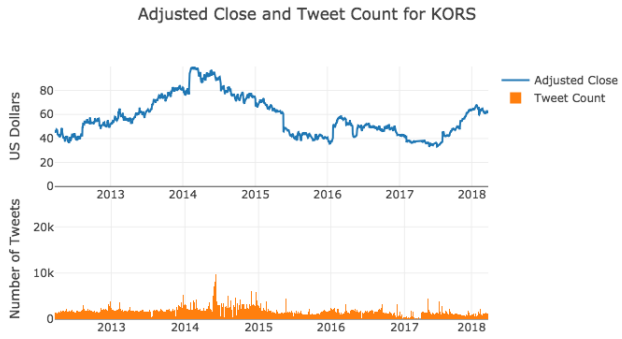

Figure 4: Michael Kors Tweets and Adjusted Close

Exxon Mobil (XOM) is a multinational company that deals with oil and gas from extraction to selling to consumers. It is a good research subject because of its steady stock price over the research period along with a relatively small number of tweets (daily mean = 209, total = 315,388). The stock price trend is that it remains around the same price throughout the years, but the number of tweets increases significantly over time even though there are still a relatively low number of tweets.

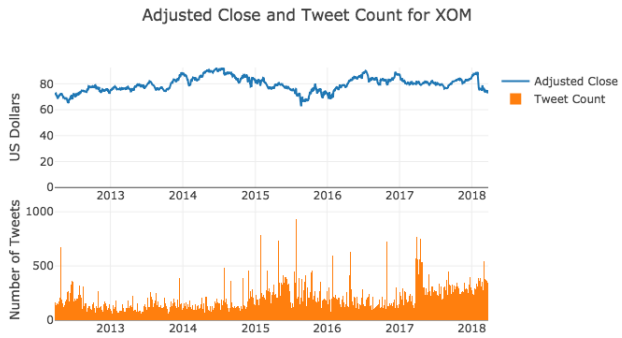

Figure 5: Exxon Mobil Tweets and Adjusted Close

2 Methods and Algorithms

2.1 Data Collection

We wrote code which uses the open source library Twint [11] to collect all of the tweets in English from March 27th 2012 to March 27th 2018 for these five companies. We used this library instead of the Twitter API directly because the Twitter API rate limits applications to a max of 1800 tweets an hour. In total, we collected 13,965,743 tweets, so even with 6 API keys, it would have taken around 1,300 hours to collect the same amount of data. Instead, the Twint library cleverly creates a search URL and uses web scraping to get the results from that page. This allowed us to collect all the tweets we did over the course of 120 hours when using six lab computers remotely. Since the only option with the Twint library is to download the tweets and not count or summarize the results, our code iterated over all the files generated to obtain tweet counts and merge the counts with the stock price data obtained from Quandl. Quandl is platform that has a library allowing us to access financial and economic data from hundreds of publishers through its API. This is the main method we used to access adjusted close stock price data in our functions. The resulting data had a tweet count and adjusted close value for each day the United States stock markets were open during the period of our research (tweets on days the markets were closed were disregarded).

Our code creates a folder that has a CSV file of each company's data sorted with date, number of tweets, and stock price. The code to do this is mainly done with five functions:

get_ data_ for_stock: Uses the Quandl API to modify the CSV column representing the company the agent will predict. It uses the company's stock market symbol to access its stock price over a set start and end date.

get_tweets: Uses the TWINT open source library to find all tweets on Twitter from a set start date and end date that contains mentions of the company's stock symbol and places it in a separate special CSV file for tweets.

get_tweet_counts_from_files: This function goes through the tweet CSV file and gets the size aka number of tweets each day for that company's stock symbol. It then places each number of each day in the CSV file that will be used by the agent in a column for number tweets next to the column with the stock price.

summarize_results: This function normalizes the two columns of stock price data and number of tweets. This is because the two numbers will be in very different ranges and in order to use both kinds of variables as input variables, they both need to be normalized relative to their mean for the neural network to accurately train and calculate accurate weights for each variable.

make_graph: This function is meant to simply create a graph of the actual number of tweets and stock price over the course of the six years. This graph is meant to be used in section 1.5 to describe the chosen data and why we chose those companies as data.

2.2 Neural Network

The method we chose to use to predict the next stock price was an artificial neural network. We chose this as our machine learning prediction classifier because as stated in the problem description in section 1.1, the goal is to devise a linear regression type equation that can predict the stock price from a given set of independent variables. A neural network works well in generating the linear regression equation because of the tuning of weight variables (coefficient of the input variables) and the intercept through optimization.

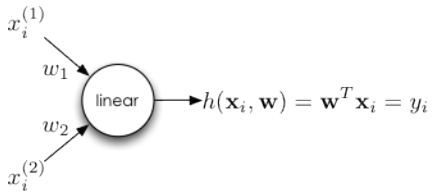

Figure 6: Linear Regression Perceptron [4]

A neural network was inspired by biological neurons. How it works is that for every input variable x_i, it is given a weight variable to be multiplied. Then each product is summed together (through matrix multiplication) and an intercept constant is added in the neuron and then leaves the neuron as a prediction. The input variables x_i are stored in a vector and the weight variables w_i are either stored in a matrix or a vector depending on the layering of the neural network.

The neural network can perform this kind of operation multiple times through layers. Instead of having every input variable be sent to a single perception and create the predicted value in the output layer, hidden layers that reduce the number of inputs to a smaller number is sometimes used to produce better predictions. There is no rule for how many hidden layers should be used, so it is best for us to try multiple numbers of layers.

Within these hidden layers, activation functions are used sometimes to convert input signals of nodes to be a certain kind of output for the next layer. When an activation function is used, the neural network is no longer a simple linear function neural network. However, many linear regression type problems still use an activation function called Rectified Linear Unit (ReLU). This activation function is defined as $R(z) = max(0, z)$ and essentially makes sure that the output of a hidden layer is always positive for the input of the next layer.

The prediction that is eventually produced after the output layer is then tested through a loss function. The loss function we used was the Mean-Squared Error loss function which has commonly be proven successful in linear regression problems. The MSE equation is: $\frac{1}{n}\sum_{t=1}^{n}(y_i - y)^2$. This equation finds the difference

between the predicted and the actual stock price and then squares for each instance and then finds the average among all the instances.

Once the loss is calculated by the loss function, optimization is done through gradient descent. The gradient descent is meant to minimize the overall error (calculated by loss function aka MSE) over the training data by tuning or changing the weight and intercept constants at each step. Initially the weight and intercept of the neural network is randomized. But upon each training step, gradient descent tries to find the local minimum of a function after modifying the weights and intercepts. This is the basis of how the neural network is trained and learns how to find the linear regression equation.

And so this cycle of calculating the prediction, calculating the loss function, and then modifying the weights/intercepts through gradient descent is done continuously through the training data. Then at the end, the model is tested and evaluated on the test data that is separate from the training data. This kind of training and testing is known as cross-validation and is used to find the best neural network model to use.

2.3 Decision Engine

Now assuming a stock prediction neural network is built, the only task left is for the agent to perform actions that utilizes the prediction to make the best choice.

Because the agent starts out with 2000 shares, it has a lot of shares that it can sell. After looking over the training data and the range of normalized stock prices, the difference from a current stock price to the next stock price are usually around a max of 0.1 but most of the time 10x less (in normalized price form). And since we do not want the agent to lose all of its shares, it would be best that it sells at most 100 shares at every transaction.

So each time a prediction is made in the year that the agent is making trading decisions, the agent will find the difference from the predicted stock price and the previous stock price and multiply that by 1000 to represent the change in number of shares. This is perfect because the change in number of shares can be either positive or negative and the magnitude of it is correlated with the degree of rise or fall of the stock price.

So when the change in number of shares is positive, that means we want the agent to buy more shares now because the stock price is going to rise. So the number of shares the agent has increases by the change in number of shares. The balance decreases because you are spending more money to buy more shares. So the balance decreases by the absolute value of the change in shares times the previous stock price. It is the previous stock price because this is occurring before the predicted day had arrived. Similarly, if the change in number of shares is negative, then we want the agent to sell shares. That means the agent needs to decrease the number of shares by the magnitude of the change in shares. The balance would then increase by the magnitude of the change in shares multiplied by that previous stock price that day.

Once the trading action had been made, the overall new net worth would then be: net_worth = balance + number_of_shares * actual_new_stock_price. Now the day in which we predicted the stock price had arrived. So we need to use the actual stock price, that our prediction aimed to be, to see what the actual net worth after the agent made the trade would be.

3 Experimental Setup

The neural network we use to create stock price predictions is a feed forward neural network created with TensorFlow [3]. TensorFlow is an open-source library for programming machine learning algorithms such as neural networks.

However, there are lots of different ways to build a neural network and each one may perform differently from the rest. So it is important to try various neural networks with different variables/parameters.

We tried six different neural networks with two independent variables that identify what makes each neural network different. One of the variables being modified is the number of hidden layers. Hidden layers can help better fit the data. Too many layers can cause over-fitting while too few layers can cause under-fitting. The one with one hidden layer reduced the input from 11 to 6 before going through the output layer. The one with two hidden layers had the first hidden layer reduce from 11 to 6 and then the second one reduce from 6 to 3.

The second variable modified was the activation function. In half the neural networks, an ReLU activation function will be added and in the other half it won't be added and instead it would just be a linear regression neural network. So for non-ReLU networks, the perceptron equation would be $Y_i = matrix\_mult(W, X) + B$ while the ReLU networks would have the perceptron equation would have $Y_i = max(0, matrix\_mult(W, X) + B)$.

What remains constant among all neural networks is that there will be 11 input nodes/variables. Six of which will be the number of tweets each day and five of which will be the stock prices each day. The goal is to use the previous five days of stock prices and the six days of number of tweets to predict the sixth day's stock price. The reason six days of number of tweets was chosen was because of the findings of the related works in section 1.3. In "Twitter sentiment around the Earnings Announcement Events", it was shown that twitter sentiment proved more useful in predicting the day of the earnings announcement and stock price. That is why the 6th day of number of tweets is included in predicting the 6th day's stock price. Similarly, both papers in section 1.3.3 and section 1.3.4 have found three days of twitter sentiment data to work best in representing the overall sentiment at that time of the company. Although we do not use three days of sentiment data, we use a multiple of it which is six. The reason we use six rather than three is because the papers that used three had more representative data because their sentiment was measured through their own analyzer/classifier whereas ours is simply the number of tweets. So we decided that twice as many days would be needed to get a better representation of the prediction.

The neural network is performed as how the method in section 2.2 is described with MSE as the loss function and gradient descent as the optimization algorithm for tuning the weight and intercept parameters of the neural network.

Each of the six different neural networks were compared by testing average loss or MSE across the aal data training data and testing data. The one with the best results (low MSE in both training and testing data) would be used for all of the other companies.

The training and testing data was read and separated from a pandas dataframe that read the CSV file containing the company's data on number of tweets and stock price across six years. The training data will contain the first five years with each training set separated in sets of six days. This means that the testing data will contain about a year's worth which we decided was 252 days (due to stock market only being open on business days). The test data was set to every week so we ended with 51 test sets. Each week, the neural network would need to predict the stock price at the beginning of the next week. The training data would have 51*5 = 255 sets.

Once the best neural network had been decided, that neural network was trained on each of the five companies. After testing, the trained neural network would then predict the stock price for the next week for every week (every 5 days) of the March 23 2017 - March 23 2018 year. The prediction's magnitude was multiplied by 40 to attempt to offset the remaining error in prediction. Then by following the decision engine described in section 2.3, the agent will buy or sell stocks with an initial balance of $1000 and 2000 shares for each company.

Along with

4 Results

Below are the results from the various neural networks trained and tested with their calculated mean-squared errors for the training and testing sets.

Add ReLU to Hidden	Number of Hidden Layers	Average Training Error	Average Testing Error
Yes	0	0.346	0.321
	1	0.231	0.089
	2	0.187	0.315
No	0	0.342	0.335
	1	0.435	0.246
	2	0.234	0.347

The zero hidden layers neural network performed the same with or without ReLU obviously because there was no hidden layer for an ReLU activation function to even be added. The neural network that performed

the best was the neural net with the ReLU activation function added and only having one hidden layer.

The reason adding the ReLU activation function produced lower error might be because it ensure that no negative outputs can be calculated. If a negative output is calculated by one of the hidden layers, it messes up how the weights and intercepts gets optimized by gradient descent at one of the layers. A negative prediction should not even be possible since all stock prices can only be positive.

It appeared that having zero hidden layers caused a higher mean-squared error. This is most likely because the lack hidden layers caused the neural net to under-fit the data. This is supported by the fact that both the training and test errors are high compared to the other neural networks. Having two hidden layers did create a lower mean-squared error compared the one hidden layer neural network in some cases. However, the fact that the test error is high even though the training error is low indicates that there may be an over-fitting issue with the two hidden layer model.

In the end, the single hidden layer model with the ReLU activation function included performed the best. Below are the results of using the single hidden layer model on each of the five companies. Note that all of these are with normalized stock price.

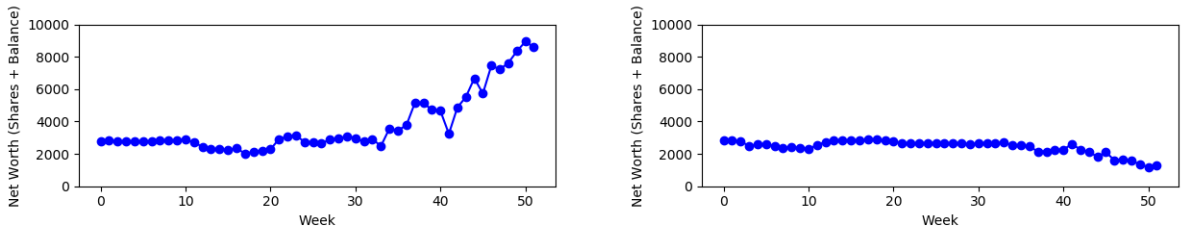

Figure 7: Trading Agent vs Random Net Worth General Electric

From the graph, it shows that agent was able to perform considerably well and was able to raise the net worth considerably after 40 weeks. Around 35 weeks it appeared to be gaining a lot but was offset by the small loss at around the 41st week. This agent performed remarkably well compared to if no trades were made: \$1000 + (2000 shares * 0.5929) = \$2185.8. The trading agent was able to make the net worth about four times as much. One of the reasons the trading agent was able to perform so well was because of how smooth the trends for the stock price and tweets were. If you recall to section 1.5, the stock price for General Electric had a nice constant increase while the number of tweets had a nice constant decrease. This strong correlation was probably a reason for the neural network/agent to be able to perform as well as it did.

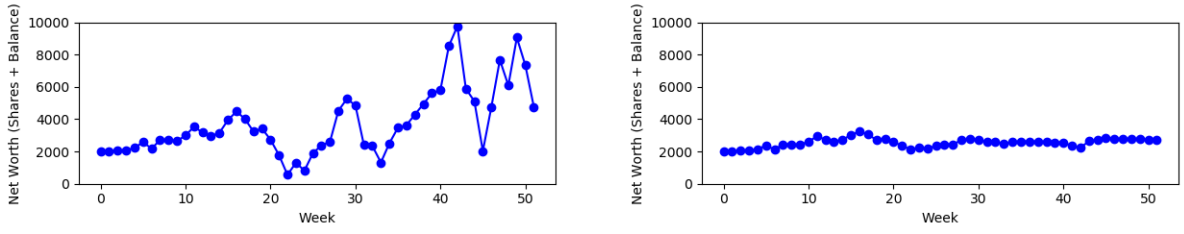

Figure 8: Trading Agent vs Random Net Worth for American Airlines

In this graph, the net worth was far more sporadic and inconsistent. It appeared that throughout the year the agent made large gains as well as large losses. A good reason for why this had happened is because the stock price and number of tweets for American Airlines is also very sporadic. In section 1.5, the stock price is very volatile and the number of tweets are even more volatile with no clear pattern/trend at all. This kind of volatility in the data most likely causes a strain on the neural network as it tries too hard to try to fit the inputs into some kind of linear function as it appears there is almost no pattern. In the end, the agent was still able to make a large profit from it's trading than if it had not traded at all. If no trades were made: \$1000 + (2000 shares * 0.7158) = 2431.6. The trading agent in the end was able to make the net worth almost twice that amount. However, if time were to continue, the agent might have lost it all due to the volatility of the agent's losses and gains.

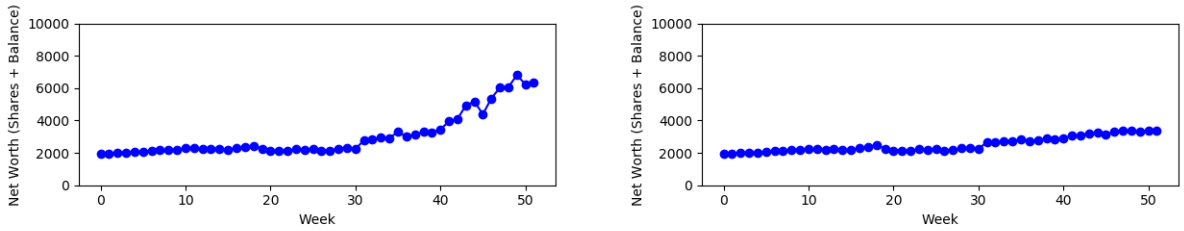

Figure 9: Trading Agent vs Random Net Worth for Amazon

In these results, the agent was also able to make a profit from its trading, but in a much more smooth and curved manner. Even the random trading was able to have a smooth linear increase in net worth. A reason for this strong smooth increase in net worth is most likely due to Amazon's strong increase in price over the past 6 years with very little drops in price. This strong increase in price makes it easy for the agent to predict the next price without even looking at the number of tweets as a sentiment reference. Although the number of tweets also has a gradual increase over the years, it is definitely not as strong or consistent as the stock price increase. If there was no trades made: \$1000 + (2000 shares * 0.9283) = \$2856.6 while the agent was able to make the net worth to be around twice that amount.

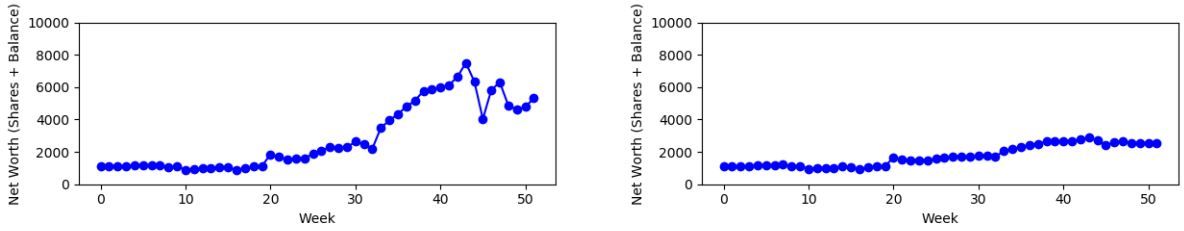

Figure 10: Trading Agent vs Random Net Worth for Michael Kors

It appears from the graph that the agent was able to make significant increase in net worth much earlier than the other companies. Although it appeared that it started to fluctuate a little towards the end. The stock price for Michael Kors is alot less consistent than Amazon's stock price as seen in section 1.5. Yet, the agent was able to perform just as well in Michael Kors as it did with Amazon. The reason for this is most likely because even though it lacks a consistent stock price, it has a higher correlation with it's number of tweets data. In section 1.5, the stock price for Michael Kors is high when the number of tweets for Michael Kors is also high. This suggests a strong correlation for this company with stock price and number of tweets. And so where Amazon had a consistent trend in stock price and less correlated number of tweets, Michael Kors was the opposite in that it had an inconsistent trend in stock price, but made up for it with its strong correlation with stock price and number of tweets. If there was no trades made: \$1000 + (2000 shares * 0.4261) = \$1852.2 whereas the trading agent was able to make the net worth to be three times as much.

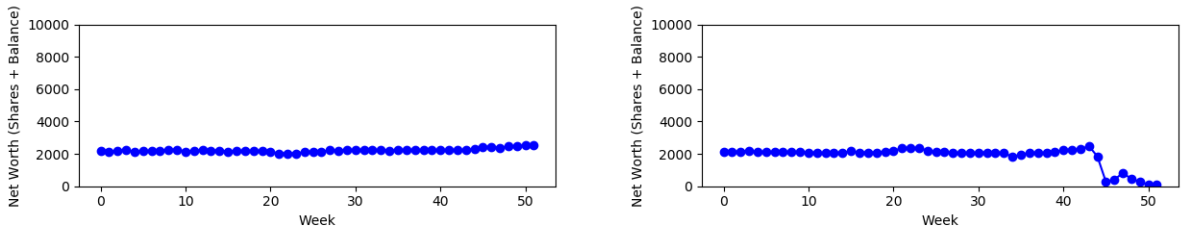

Figure 11: Trading Agent vs Random Net Worth for Exxon Mobil

The agent was not able to produce very large change of net worth for Exxon Mobil. The net worth had almost stayed the same the entire time. This is most likely because the Exxon Mobile stock price did not change very little throughout the six years. Because the stock price was almost constant throughout, the neural network predicted around the same stock price each week resulting in very little need for the agent to buy or sell shares. If there was no trades made: \$1000 + (2000 shares * 0.3541) = \$1708.4. The agent had barely made the net worth a little more than what it would have if there were no trades.

In all cases our agent was able to perform significantly better than if a random trading agent would. It is important to note that the stock prices were all normalized, so our agent could still have performed less than not trading at all depending on how the distribution curve for the actual stock price looks.

5 Conclusions

In the end, we were able to successfully create an agent that was able to take in past stock prices as well as number of tweets to predict the next stock price and trade shares based on this newly predicted stock price with the goal of maximizing net worth. We believe the results could have been improved in future next steps by implementing more data cleaning procedures such as detrending the stock price data, performing a more rigorous Twitter term search, and filtering out spam tweets. In addition, adding short selling, the ability to create a portfolio, and transaction costs would make it more useful and realistic. Finally, extracting the sentiment from the tweets instead of using a count would have likely improved the predictive capability. Another future modification might be to include along with number of tweets, a sentiment tweet analyzer that can give each tweet a positive or negative value of the opinions of each company. Perhaps instead of just a binary or trinary representation of the sentiment of a tweet as other related literature has tried, maybe look into trying to design a sentiment analyzer that can give a more continuous value instead of a discrete one. Maybe look into natural language processing options in trying to fully understand and read a tweet. An perhaps maybe even explore other social media platforms such as Reddit or Facebook.

References

[1] Quandl api. https://www.quandl.com/tools/python.

[2] The comprehensive guide to stock price calculation. https://blog.quandl.com/guide-to-stock-price-calculation, Jan 2017.

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[4] Brian Dolhansky. Artificial neural networks: Linear regression (part 1), Jul 2013.

[5] Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383, 1970.

[6] Peter Gabrovsek, Darko Aleksovski, Igor Mozetic, and Miha Grcar. Twitter sentiment around the earnings announcement events.(research article). *PLoS ONE*, 12(2), February 2017.

[7] Keith C. C. Li Bing, Carol Chan, and Carol Ou. Public sentiment analysis in twitter data for prediction of a company's stock price movements. pages 232–239. IEEE, November 2014.

[8] Venkata Sasank Pagolu, Kamal Nayan Reddy, Ganapati Panda, and Babita Majhi. Sentiment analysis of twitter data for predicting stock market movements. *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016.

[9] Lucinda Shen. Here's who is winning after mark zuckerberg's response to the facebook data breach, Mar 2018.

[10] Thársis Tuani Pinto Souza, Olga Kolchyna, Philip C. Treleaven, and Tomaso Aste. Twitter sentiment analysis applied to finance: A case study in the retail industry. July 2015.

[11] Cody Zacharias. Twint - twitter intelligence tool. https://github.com/haccer/twint, 2017.