*CIS 400 Project Report*

*John-Paul Besong, Dounglan Cheung, Mario Garcia, Jeremy Gavrilov, Zachary Pinter*

*5.2.2020*

*Table of Contents*

## Approach..…………………………………………………………… 2

Research ……………………………………………………………..3

Packages ……………………………………………………………. 4

Data ………………………………………………………………….. 9

Other Information …………………………………………………...15

**Approach**

Motivation:

Given the impact the COVID-19 pandemic has had on businesses, our team decided to use sentiment analysis to predict fluctuations in stock prices in various industrial fields. Given our goal, we decided to stream tweets about specified companies, give each of these tweets a positive or negative polarity value based on the word usage in the tweet, and look at the correlation of our results with stocks prices. The overall trend these stocks had with one another is that they all had a significant rise or fall due to the spread of COVID-19.

Method:

We chose to focus on four specific stocks that we saw were heavily affected by the crisis. To retrieve the most Twitter data, we decided to observe the companies that had been the most trending. First, we looked into the various airlines which are available in the stock market. Airlines have been deeply impacted worldwide due to strict travel bans many countries are putting into place. All American based airlines had a similar impact with stocks going down, so we decided to work with American Airlines ($AAL). Another market that had a heavy negative impact was the oil and automotive industry. Due to a recent decline of the automotive industry, as well as the dropping of oil prices, car sales have plummeted. Tesla ($TSLA) is one of the most popular automotive companies one Twitter, so it seemed like the obvious choice for this category. With two stocks that had been doing poorly, we decided to balance our data and look at stocks that had been going up, the obvious route was to look at media outlets, Netflix ($NFLX) and Zoom ($ZM) have been used heavily. We looked into how the two would be affected in the stock market because of the pandemic. The pandemic put the world in a state of quarantine and thanks to it, people at home would use these media outlets as a way to operate life. For Netflix, many people would use their streaming service as a way to pass time and cope with what is happening in the world. Therefore people would renew their subscriptions so they can get their entertainment during these times. For Zoom, many academic institutions are using it to operate classes, lectures, discussions and exams. Because of the frequent use of these two media sources, the stocks in the stock market for each would increase.

Research:

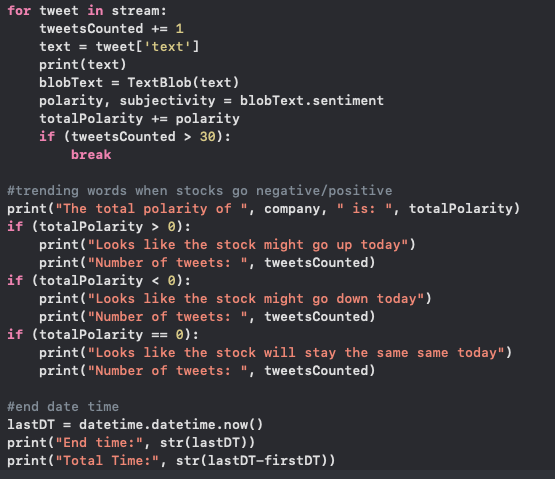
Before choosing our approach, we spent some time researching past examples of how people have used data and text mining techniques to make predictions about the stock market. Many of the papers we found focused on analyzing stock trends from the past, to make a prediction as to when a stock would be rising or falling. It does this using decision trees. This seemed like a good approach but we were afraid the versatility of the stock market would make old trends somewhat unreliable to predict whether a stock was about to go up or down. The stock market is affected by the economy of people; when a stock is doing well, it is being heavily invested in, and the opposite when it is not faring so well. To some degree, people seem to be in control of how a stock might be doing, so we thought what better way to get the opinion of people than through mining what they might say about the stock on social media. As stated before, we chose twitter as our social media because it seemed the platform most appropriate to search for opinion. Once finding these opinions, we are able to find a sentiment score for all the tweets about a certain company. Using the sentient score we find, we thought we would be able to make a prediction about the stock from there. This seems like a much more reliable approach than mining data from the past, and looking at trends of the stock to make a prediction of a near future event. We also researched a few more methods, but this seemed to be our favourite for the task at hand.

Packages:

From previous projects, we used Python to utilize the Twitter API, it seemed to be the best pairing to use. The packages we chose to use are Numpy, TextBlob, Panda, Alpha\_Vantage, DateTime, and MatPlotLib. The Twitter API was used to collect tweets pertaining to the stocks we had chosen to examine. Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. TextBlob is used primarily to output the polarity value by using the textual data that every tweet possesses given the stock industry abbreviation. Alpha\_Vantage pulls the stock data which we work with and Panda manipulates that data. DateTime allows Python to manipulate date and time. For our purpose, we use it to record the starting, ending, and total runtime of the program. MatPlotLib allows us to take the twitter data and show it via a graph. Each package was utilized in the following ways:

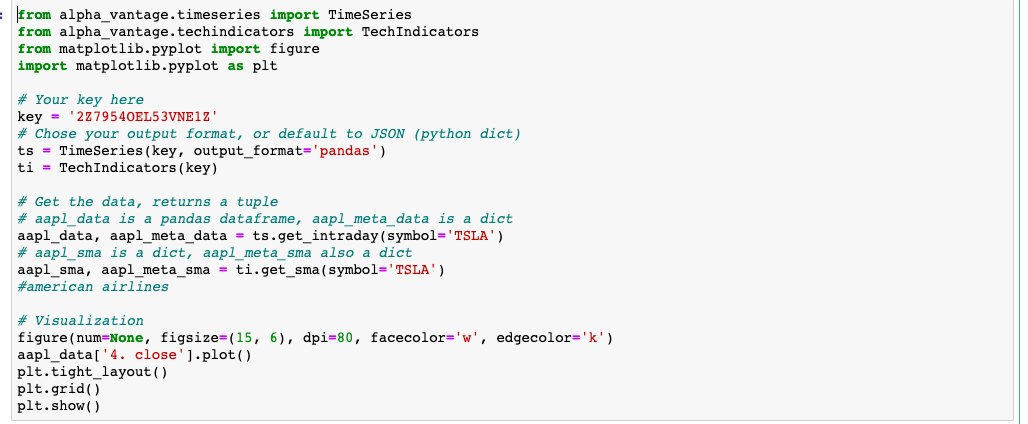
**TextBlob**

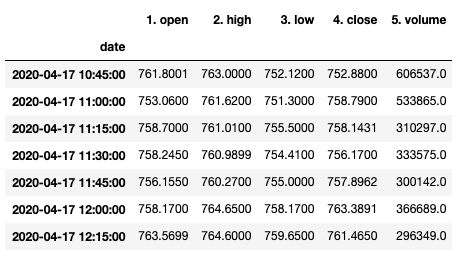
TextBlob is used to perform sentiment analysis on each tweet. For the purposes of our project we focused on only the polarity of each tweet. For trial reasons the total number of tweets counted was reduced significantly in order to reduce run time. Tweets posted directly after the pre-market ends were taken into account. If the summation of all tweets counted was positive, our prediction would be that the stock would go up by the end of the day. If the summation was negative, our prediction would be that the stock would go down by the end of the day. Lastly, on the off chance the summation of the polarities was zero, the stock would be predicted as it would not change.



**Alpha\_Vantage and Pandas**

Alpha\_Vantage. is a leading provider of real time and historical stock APIs as well as forex (FX) and digital/crypto currency data feeds. Alpha\_Vantage was used to collect the stock information of our carefully selected companies. We chose Alpha Vantage as a group due to its robustness as an API. The API is able to collect the open & closing prices of individual stocks as well as trading volumes, more so it was able to display this information for selected intervals(every 5 minutes, or 15 minutes) throughout the day. This robustness was important because if we were to extend the project even further, we would be able to see how certain types of tweets affect a stock at different times of the day; as well as what time of the day is best to tweet to affect the price of a stock.

Pandas is used within Alpha\_Vantage. Pandas is a software library written for Python for data manipulation and analysis. We use pandas when handling some of the returned output from the stock API. The stock API returns a dataframe, and we use pandas to manipulate and analyse the returned table. 



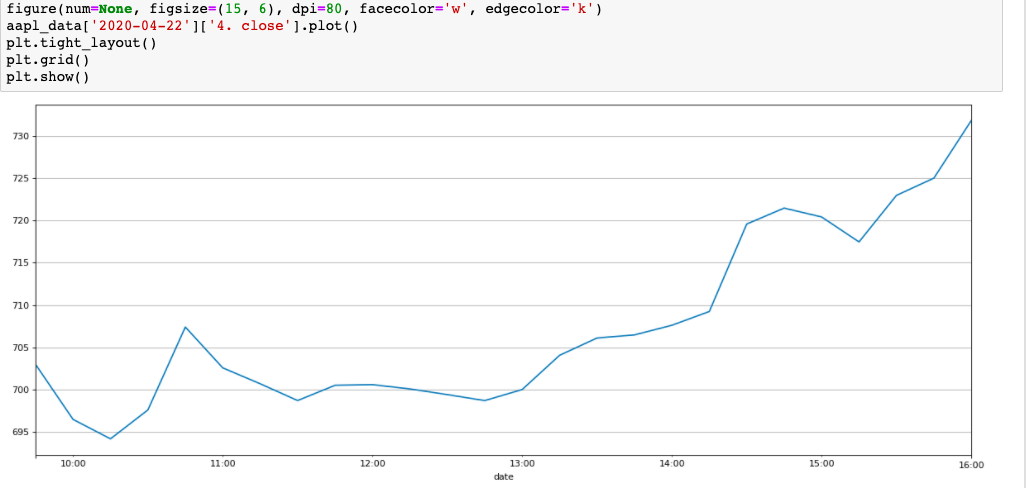
**DateTime**

DateTime is used to indicate the time started and ended for our program. It would then calculate the total running time by taking the difference of the end time with the start time. For our purposes, we used this to calculate how much time would pass as the program collects twitter data for the sentiment analysis.



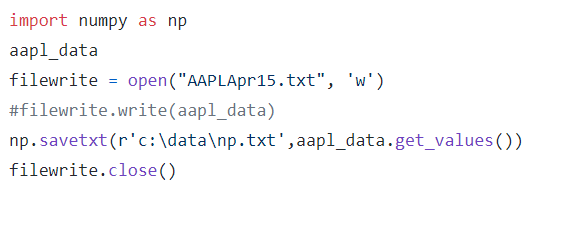
**MatPlotLib**

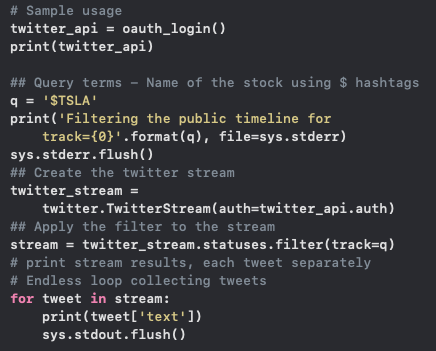
Matplotlib is a plotting library for Python. Once we have received the stock information in the form of a table from Alpha\_Vantage, we use the Matploblib library to display the data points received. Matploblib uses several general purpose GUI’s to display the data.

****

**Numpy**

Numpy is MatPlotLib’s numerical counterpart. It can be used to manipulate and visualize large datasets.

**Twitter API**

Since we used Twitter as the medium for our data collection, the Twitter API was useful. The base of the project involved collecting tweets in order of being sent. The streaming API was used for this reason. The basics of the streaming API were used to just gather the text of each tweet, and print it out. The query terms only included stock names when they appeared with their “cashtag” ahead of the stock. We decided that the tweets that contained the cashtags were sent specifically talking about the stock itself, rather than the company. If we just queried “Tesla” we might have gotten a lot of tweets about the company, or the cars that weren’t related to how the company is doing financially. But most of the accounts that send tweets using the cashtag were more credible sources like news outlets or people that study the stock market. 

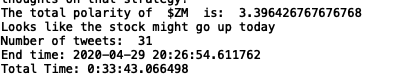
**Data**

As described in our APIs section, we used Alpha Vantage to collect information on particular stocks. Alpha vantage returned the stock information as a table. This returned table usually returned the stock information for a specific company over the last seven days, and each point was usually an interval apart(either 5 or 15 minutes, determined by a search parameter). Having this short history of the stock returned, instead of purely live stock information, allows us to see the trend of the stock and whether it has risen or fallen; this history also allows us to give the system a sense of this trend. With this history, at the end of the day, the system can relay to use if its predictions were correct or not.

When selecting which stocks to run our experiment on, we needed to select the stocks most affected by the current pandemic. We chose this selecting approach because the stocks most affected would have the most day to day movement, and thus yield measurable results. More so, these were going to be the companies/stocks most spoken about, so it would be easy to generate tweets for sentiment analysis, whereas less affected and thus less relevant stocks would be a bit harder to visualize what was going on. We selected Netflix, because the streaming giants skyrocketed after almost the whole world has been put into quarantine, and this is because people do not have much more to do, so everyone is on netflix. Almost everyone at the moment, has an opinion on Netflix. This popularity of opinions makes it easy to judge sentiment on the company. Next, we chose American Airlines because of how poorly the airline industry is doing. Noone can travel around the world for the foreseeable future, so it was interesting to see how they would bounce back as things progressed. Zoom has become the new super power in video communications, so everyone is now aware of and uses zoom. This rise in popularity is reflected by the stock's huge growth over the last few months. Lastly, we chose to look at Tesla, another stock that would be in the news and on social media a lot during this pandemic, so they are an easy target to analyse and visualize all the changes.

Some of the sentiment analysis took hours to collect, sometimes it took 5 minutes to arrive at a sentiment verdict for each company. This came down to a few factors. When collecting sentiment analysis from the stocks, we realised that certain companies are more popular than others. Some companies had traded large volumes everyday, while some companies did not change as much; this reflected in the popularity of the stock. Popular companies were traded and tweeted about more often, so when performing sentiment analysis on the company, the program will not take long to run because many tweets are being live collected at the same time; whereas less popular companies took long to perform the analysis because it would take long to live tweet our set goal of tweets.

Picking a certain number of tweets was another factor in determining how long it would take to run the sentiment analysis. We experimented with various tweet targets to find a number that provided the most accuracy. Sometimes we collected five tweets; collecting five tweets never took too long but we were sometimes uncertain as to if this provided an accurate sentiment score, seeing as five tweets is not much information collected. Sometimes we also receive neutral tweets or tweets with little to no meaning regarding the stock. Sometimes we experimented with up to 60 tweets; which took exponentially longer than five minutes, yet usually provided much more accurate sentiment scores, which provides a much more accurate prediction as to whether the stock will rise or fall.

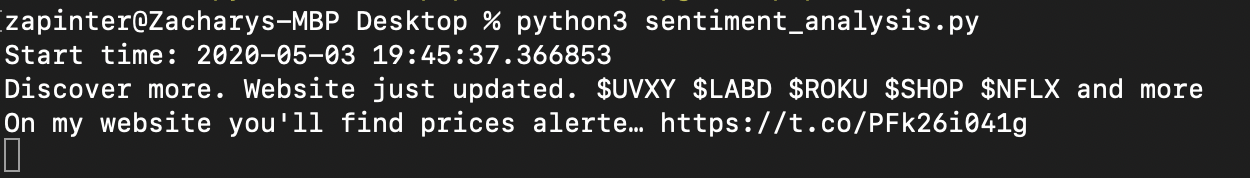


Usually we perform sentiment analysis before the markets open, which is usually the morning in the USA. Doing this usually gave us a good idea of how people are feeling about a particular stock prior to the market opening, which should provide an indication to what their actions might be once the market opens. We can also perform sentiment analysis during the day to show how the general opinion toward a particular stock might change during the day, and how this is reflected in trades made for that stock.

Actual stock information from alpha\_vantage is collected at the end of the day, which shows all the prices of the day. We chose to use this end of day API because it is easiest to track the past information of the stock from there. Using live stock APIs would have been more efficient but also less cost effective because we could not find any free access to these APIs. Once the information is collected at the end of the day once the market has closed, we are able to see if the analysis of the stock was correct. The only part of this that can be improved is the timing of when we do everything. It is possible for sentiment about a certain stock to change during the day, like a story may suddenly break in the news, this could change public opinions about the stock and thus make the stock do something different than what our premarket analysis predicted. We think we might get more accurate predictions of particular periods of the day, rather than doing an analysis and prediction for the whole day.

**Experimental Results and Analysis**

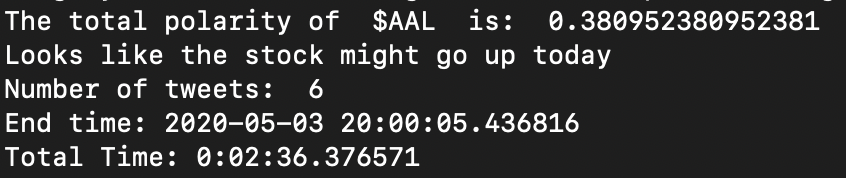
Results from the Twitter stream look like this:

The start time will be printed, and the incoming tweets’ text will be printed.

We decided to focus on both direct tweets and retweets. Direct tweets talking about stocks more times than less are from stock brokers talking about the stock, which are the most accurate source. Retweets of those tweets are also good to consider too, while they are most likely being retweeted by regular people, they are still reaching a wider pool of individuals than a regular tweet would.

In the code, there was a line that decides how many tweets to filter for sentiment analysis. 

In most of our studies we used about 30 tweets for one sentiment analysis result. More tweets being collected would give better results for how that stock might perform because more people would be giving their opinion on the stock. The best time we figured collecting the tweets would be in the morning, right before the stock market opens, because the sentiment analysis predicts how the stock should perform that day. We used 30 tweets as a filter because that was a pretty easy number to get results after a few minutes. It would take a bit too long when there are more than about 100 tweets, unless there is something crazy going on within the stock and people were talking about it with the cashtag.

At the end of the stream, the total polarity is calculated and printed out. The results look like this: 

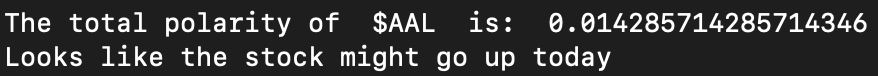
Each tweet was given its own polarity score, and at the end of the program, the total polarity was calculated and printed. If it was positive, we predicted the stock price would go up. If it was negative we predicted it would go down that day. That is why our best results were when we ran the program early in the day before the market opened and compared it to how the stock performed later in the day. Because the sentiment analysis was a running total, there was a large variance on the polarity that we got for a given run of the program. Sometimes we had some near 1, or 0, or -1, which would show a fairly neutral but slight preference one way or the other. Other tests gave us polarities over 4.0 so there were several high polarity tweets that were overwhelming the results.

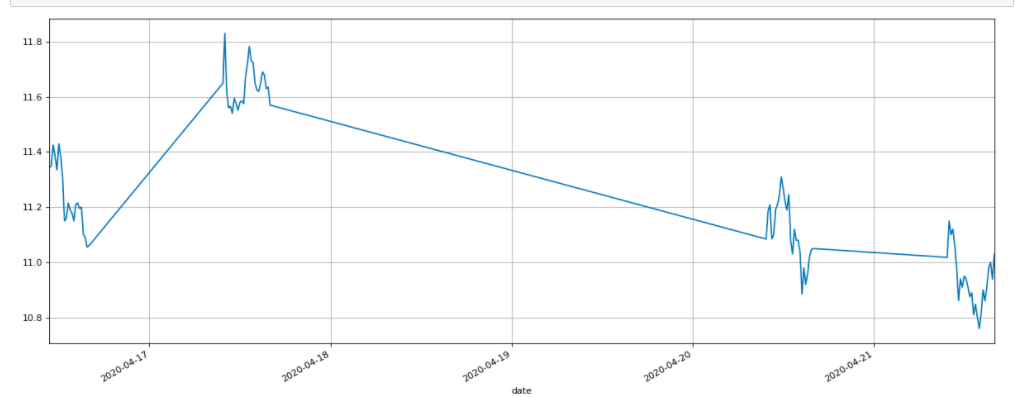
**Company Specific Results**

The following results were from one day of testing. The program was run in the morning, and then after the stock market closed we could see the graph data, and compare the results.

**American Airlines**

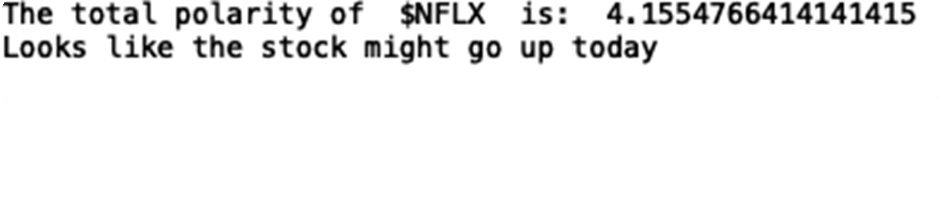
Prediction

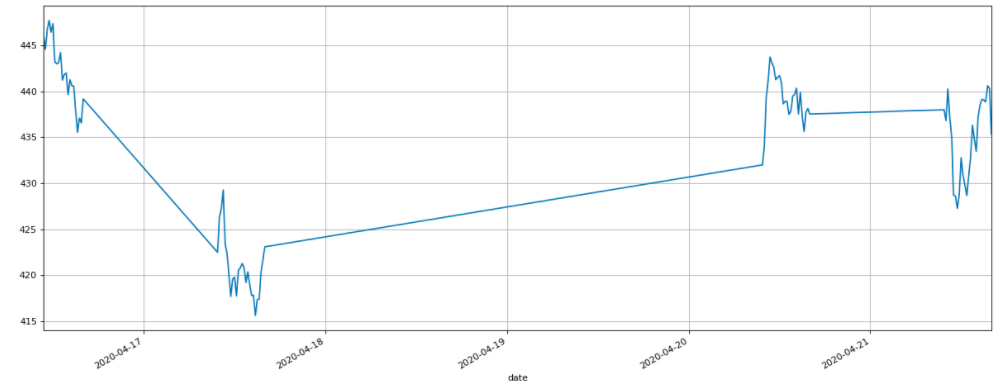
Graph



**Netflix**

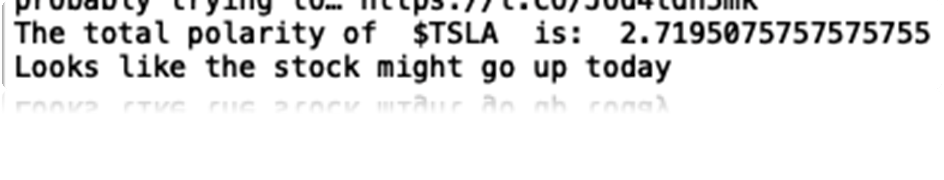
Prediction

 Graph

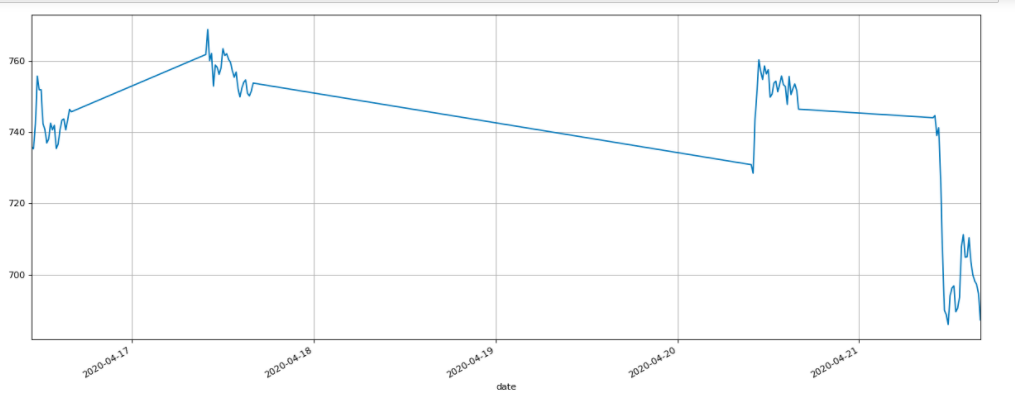


**Tesla**

Prediction

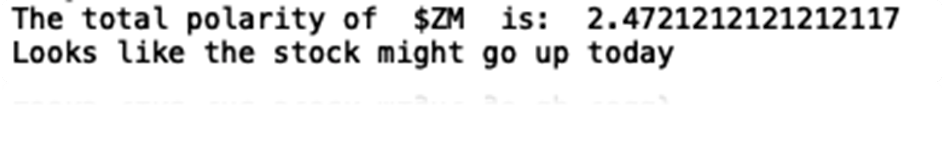


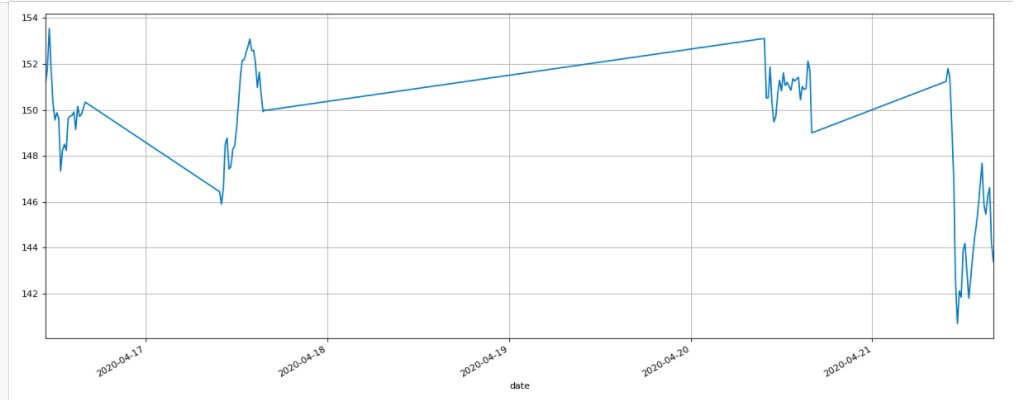
Graph



**Zoom**

Prediction



Graph 

**Inconsistencies with Data**

With any experiment, there are bound to be inconsistencies with the gathered data. The main inconsistency would be if the prediction our program assembled was correct. This did happen from time to time, but the percentage of error varied. Due to the high complexity of human decision the change in a stock may have occurred mid day, as it did with Zoom ($ZM) on 2020-4-21. Perhaps a way to get around this is to run the program for a longer period of time rather than till a set number of tweets are collected.

**Other Relevant Information**

There were challenges that we came across during the project. Initially the quarantine made it so we couldn't meet in person, since we formed the group close to spring break, so we had to do the entire project over zoom meetings and group messages. It wasn’t very hard to communicate what tasks people wanted to do, and focus on. We had a good idea of how we wanted to collect the information, and where we were going to do it. We all felt comfortable with using the Twitter API, and were familiar with how to work with tweets, since we did other projects using Twitter in the class.

Obtaining live stock information, would have given our system a more efficient and effective way to automatically detect if its predictions were correct. It would do this by comparing the current stock price to previous ones(maybe in the last hour or so). Obtaining only live/current stock prices would not have had the same effect. This was why we did not simply collect stock information from websites as xaml, because it would not because this would not have given us a clear history of the stock to check the validity of the predictions. If we were able to receive free live stock information or receive the information at a reduced rate, we would be able to extend the testability of our system. From the live stock information, we will be able to have the system running for several hours without human interruption, while it checks its own accuracy rates and how often it makes accurate predictions. Currently, because we do not have live stock information, this is not a fully automated system. We have to keep running the sentiment analysis on each company individually, then manually check the trend of the stock to see if the predictions made are correct. This slows down the speed at which we can test results as well as reducing the number of results which we can obtain; with little results, it is hard to tell how well the algorithm is doing.

Another problem we ran into was finding a group of keywords that most affected different types of stock. By group of keywords, we were looking for specific words that possibly could skew the results of the prediction routine, to either being very positive so the stock would go up or very negative so the stock would fall. We had a few challenges with finding these words because we could not quite devise an approach that would produce the list of words that actually had some sort of impact. Initially we thought of using the trending words module we learned in the twitter cookbook. This module collects the list of words that have been most frequent in a certain period of time. This module does a good job collecting actual trends but we did not think it would fit our needs because this module might simply just display common words from the tweets we have streamed.

We also wanted to implement some sort of influence model to give certain tweets collected from the stream more weight than others. This meant creating some sort of hierarchy or popularity score for each tweet. This probably would have been a further application of the Twitter API to lookup the profile information for every tweet that was received. The limitations with this would be the cap on profiles searched per 15 minutes.