Using Twitter Sentiment Analysis to Predict Stock Market Simple Moving Averages

Leonard Box

Texas A&M University – San Antonio  
1 University Way  
San Antonio, TX 78224  
(210) 784-2302

lbox01@jaguar.tamu.edu

**ABSTRACT**

We are seeking to explore the relationship between stock market-related sentiment and movements within the US stock market, we collected a sample of 28,000 tweets using the key hashtag, #stockmarket during an 11-day period in July 2018. Since the advent of the Internet and near-real time information to the layman, an industry of day traders have cropped up as they use this and new technology tools in an attempt to stay one-step ahead in the game. We intend to determine if Twitter is one of these research tools for these days traders and if so, what influence does it have on the market itself. Simple Moving Averages have been used for years by investors to determine trends in the stock market, is there a correlation to the overall mood on Twitter to the mood on Wall Street? After processing our data, we intend to match this information up with the actual movements of the market. Taking the five FAANG (Facebook, Apple, Amazon, Netflix, and Google) stocks along with the tweets in our data set that mention them to try determine if there is a cause and effect where the tweet potentially moved the stock price up or down or was the tweet in response to a previous market movement.

# INTRODUCTION

Since its founding, speculators have attempted to make accurate stock market predictions. Researchers from all disciplines have also attempted to apply their own discipline’s techniques to find an advantage. According to the researcher Eugene F. Fama, that many people regard as the foremost academic on the market, the Random Walk Hypothesis governs stock prices and as a result, they cannot be predicted any better than the results of a coin toss. [1] This project attempts to uses tweets to predict next day stock prices.

# RELATED WORK

Stock market trends have been influenced for years by the big money moving of private equity funds and other major players, however with the advent of the Internet and ability for anyone to easily access market research and trends, the concept of the day trader began to flourish. This day trader uses various tools to conduct their research. Given the vast amount of information at their fingertips, including Twitter feeds from brokerage firms as well as cable financial news talking heads are filled with information that may have important information that may influence the day trader or even large firms positions in certain companies. Given the vast amount of information that investors can find on companies, what influence do these mainstream Twitter actors have on day trader investors.

Twitter, currently attracts an estimated average of 271million users every month. [2] And depending on how events or topics are perceived by Twitter users can affect how events and topics are perceived by the general public. Over the years Twitter has been the demise of many a multi-million-dollar blockbuster for example. One of the recent examples would be the Broadway show turned movie, Cats. Cats was not only panned by critics, but Twitter users equally panned the movie which cost over $100 million to make and made $6 million at the box-office. The popularity of microblogging and especially Twitter, can be explained by its distinctive features such as convenience and accessibility, which allows users to instantly respond and disseminate information with limited or no restrictions on content. [2] This of course can lead to a sort of ‘Wild West’ feeling which is sometimes freeing but recently has become increasingly hostile.

However, the Twitter effect has been shown to be particularly relevant to experiential media products (e.g., movies, music, and electronic games); these are generally the products for which ‘instant’ success is essential. [3, 4] This then leads to the assumption that if these products can of course benefit from the Twitter effect, can other products or services also enjoy this success? Stocks would surely fall into this category. If the Efficient Market Hypothesis (EMH), the assumption that stock rise and fall of stock prices follow a random pattern and are driven by new information, surely a Twitter effect on a particular stock could either drive the stock price up or down depending on its sentiment. Behavioral finance has provided further proof that financial decisions are significantly driven by emotion and mood. [5] It is therefore reasonable to assume that the public mood and sentiment can drive stock market values as much as news.[6]

The paper is organized as follows: The second section discusses the analysis process that is being conducted with the current dataset and discusses in detail how the data is perceived and how it is likely to affect trader sentiment and move market prices and holdings. In the third section, the processes used to clean the dataset are discussed. As well as what preprocessing was accomplished to ensure that usable data was extracted from the original dataset which will then be used to test this paper’s hypothesis.

# DATASET DESCRIPTION

This project utilized David Wallach’s StockerBot tweets that were extracted from Twitter between July 9th and 19th of 2018. He posted the results of the pull on Kaggle. Since, another user included the historic stock prices of all of the S&P 500 companies in order to match these tweets to actual stock prices. This data can be found at: <https://www.kaggle.com/davidwallach/financial-tweets>. David Wallach’s StockerBot python module, found at: <https://github.com/dwallach1/StockerBot>. This original tweet data set contains 28,436 rows and eight columns: *id*, *text*, *timestamp*, *source*, *symbols*, *company\_names*, *url*, and *verified*. hyperlinks. The historic stock price data set contains 579 companies each with their own .csv file. Each .csv has the number of rows equal to the trading days throughout the company’s history ending on November 1, 2018 and seven columns: *date*, *volume*, *open*, *close*, *high*, *low*, and *adjclose*.

# METHODS AND PROPOSED APPROACH

## Cleaning

### Initial Evaluation of the Data Set

When StockerBot extracted the data from Twitter it failed to retrieve the interactions with each tweet. That is, the comments, likes, and retweet values were not returned. This could have given this data more practical uses as those data points could have yielded the popularity of the tweet thus allowing the tweet to be further categorized.

There was an assumption made that *source* was the Twitter user who posted the tweet. There is no indication within the data set that this is true or false. Manually sampling from the *source* column did in fact verify that these are indeed valid Twitter handles.

Finally, the tweet data set’s one binary column, verified was so unbalanced: 99% to 1% unverified to verified, that it was unusable. So, if this data set was going to be able to yield any useful predictive data it was going to have to come from the text which would mean cleaning and preprocessing would have to focus on this column. An idea was actually borne from the *symbols*’ column. Although this column only contained one stocker ticker symbol from the tweet and most tweet had multiple symbols, even if they were not properly extracted, perhaps categorizing the tweet’s stock ticker symbols with their respective stock prices on the day of the tweet and that this information may lead to a prediction. However with 567 unique stock ticker symbols identified in *symbols* it was decided to focus on only a small group, so the FAANG (Facebook [FB], Apple [AAPL], Amazon [AMZN], Netflix [NFLX], and Google [GOOGL]) stocks were chosen.

### Manual Cleaning

Initially, it was determined that *symbols*, only 1 of perhaps up to 10 in a particular tweet; *company\_names*, the company name attributed to the stock ticker symbol found in *symbols*; and *url*, the url of the tweet would likely not provide any useful information. Thus, they were removed from the data set. The remaining columns included: *id*, *text*, *timestamp*, *source*, and *verified*.

Next, *text* was search for the five FAANG stock ticker symbols. If the symbol was found in *text*, a new column (one for each company) was given a value of ‘1’. And if *text* did include one of these symbols, another new binary column was created to indicate it contained a FAANG symbol.

A problem was also identified with *timestamp*. The format that was returned from StockerBot was not a format recognized by Python’s datetime module. So, this column was initially divided into a *date* and *time* column where the contents were reformatted into a proper date and time form.

At the end of the initial cleaning stage, the data set contained 16 columns. Of these 16, 4 were original to the data set. The columns that now encompassed the data set included: *id*, *text*, *date*, *time*, *source*, *verified* (binary), *facebook* (binary), *apple* (binary), *amazon* (binary), *netflix* (binary), *google* (binary), *faang* (boolean), *faang\_total*, *faang\_mean*, *above\_mean* (boolean), and *beats\_mean* (binary).

### Automated Cleaning

With the data set imported into pandas, the data frame was split along *beats\_mean*. This resulted in 1684 rows of FAANG related tweets as well as an exact amount of non-FAANG related tweets. A word cloud, using the wordcloud module, was created from *text*.

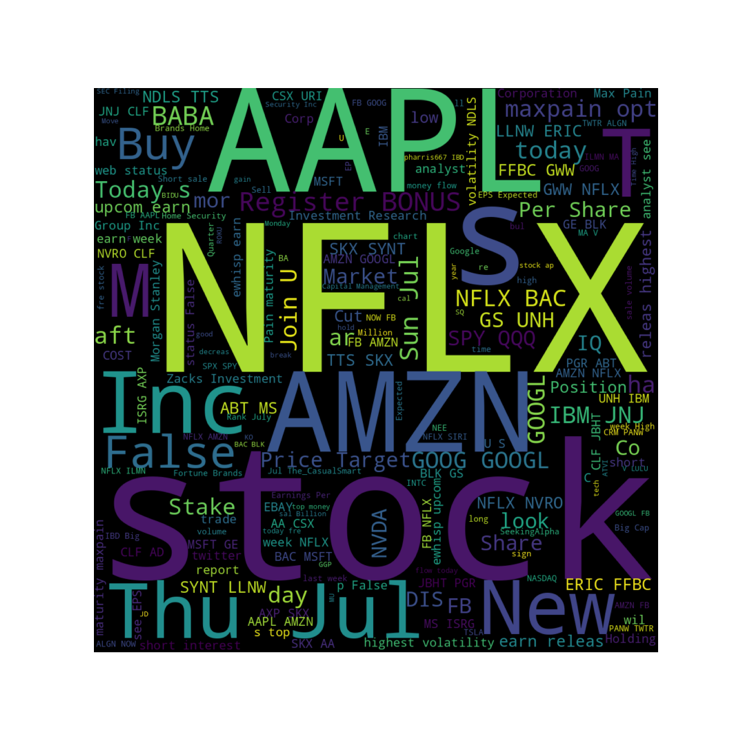
Figure . Initial word cloud produced from data set



As shown in Fig. 1, *text* has an inordinate amount of urls embedded within the tweets as well at some indications of retweeting based on the prevalence of ‘RT’.

Tokenization of text was the next step followed by categorizing each token as a Part of Speech using NLTK’s pos\_tag module. This procedure was followed up with stemming using the Lancaster Stemmer and lemmatizing using the Word Net Lemmanizter. Stop words were then removed using a combination of both the stopwords module and from the dictionary created by the first word cloud.

Figure 2. Word cloud produced from cleaned data set



A new column in the data frame was created that held the cleaned version of *text*. A new word cloud was produced from the cleaned tweets as shown in Fig. 2. Although the word cloud has less potential stop words than the original, there are still some additional words that could be added to the dictionary in the future.

## Preprocessing

### Initial Preprocessing

This step involved creating numeric from *text*. Several functions were run against *text* resulting in a number of new columns. These columns were *num\_words*, the number of words in the tweet; *num\_unique\_words*, unique words; *num\_words\_upper*, upper case words; *num\_words\_title*, words starting with a capital letter; *num\_chars*, number of characters in the tweet; *num\_punctuations*; *num\_special\_char*; *num\_numerics*, the number of numbers; *num\_uppercase*, *num\_lowercase*; and *mean\_word\_len*, mean word length. This was done as feature creation to be used during processing.

### Sentiment Evaluation

This was thought to be the most important step during the preprocessing steps. Without the ability to calculate popularity of a tweet based on comments, likes, and retweets, another method would be needed to evaluate each tweet. It was decided that determining the sentiment of the tweet would allow for the matching of tweets to selling and buying calls and to ultimately determine if they marked or created their own 2- or 5-day simple moving average when compared against stock prices.

To ensure that a true sentiment was being extracted from each tweet, a combination of the NLTK and textblob sentiment modules was used. Since NLTK’s VADER sentiment returns four scores, *neg*, negative; *neu*, neutral; *pos*, positive; and *compound*, the sum of the previous three scores and textblob only one score, both scores were broken into three columns where the first, max\_score would be the highest positive which could be a combination of *pos* and *neu* or a positive textblob score, the second would be the minimum score based this time a possible combination of neg and neu or negative textblob, and finally the compound score or textblob score. These sentiment scores were kept separately to help to also determine if one was more accurate than the other in predicting stock movement.

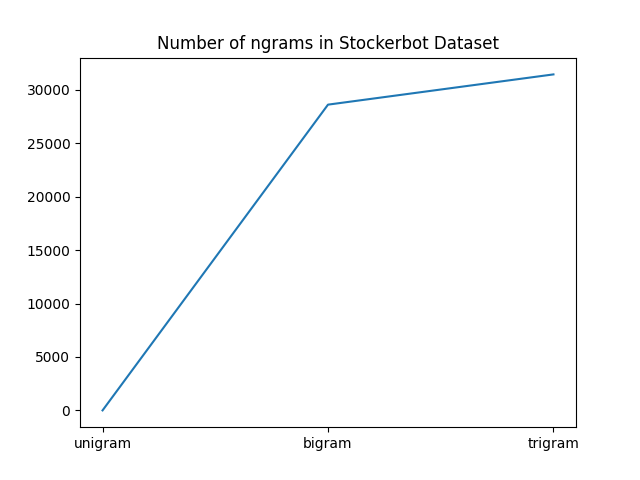
### Bag of Words Production

Using the SKLearn’s Count Vectorizer, a bag of words was produced for each of the tweets. Then, the Tfidf Transformer was applied. The initial sizes were set to 50, 100, and 500. This showed some early promise during initial processing. But ultimately it did not enhance any predictive algorithms on this data set. In fact, it made decisions much more oblique and non-sensical. Even after testing a dynamic addition of columns to the smaller sub sets, it still did not produce anything close to accurate predictions. Thus, the data set was not run through this preprocessor for final processing.

### Ngram Production

The Count Vectorizer was also used to create various ngram types and lengths. The first test created character type ngrams at lengths 2, 3, and 4. These were not effective in aiding any processing. The next test involved creating word length ngrams in combinations of 2 and 3. There were a significant number of each as show in fig. 3.

**Figure 3. Number of word length ngrams produced from data set**

Where there was at least common usage of the character ngrams, there was no common word ngrams between the tweets. And just as the bag of words Count Vectorizer was aided by the Tfidf Transformer, so too was the ngram version. Once again however, it was determined that the inclusion of either type of ngram within the data set and would have been just as troublesome as the bag of words version.

### Simple Moving Average Calculations

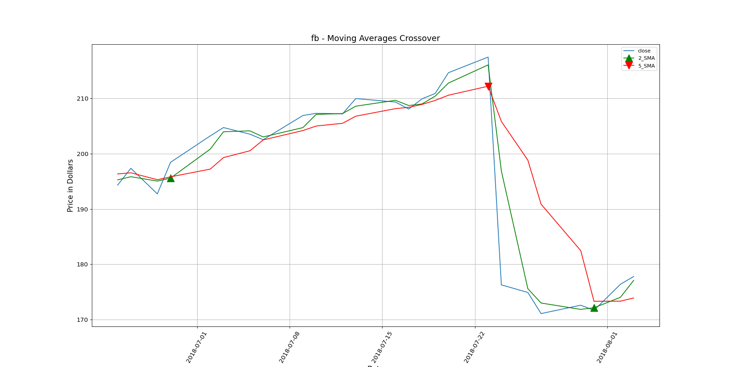
Utilizing historic stock data from the Yahoo! Finance API, each of the five FAANG stocks had their 2-Day Simple Moving Average (SMA), 5-Day SMA, and 7-Day Estimated Moving Average from June 25 – August 3, 2018. SMA which is calculated as the sum of the prices over a time period and then divided by said time period, typically this calculation has been used as a technical indicator that can assist investors in determining whether or not a stock will continue or reverse its trends.

**Figure 4. SMA calculation utilizing the pandas module**

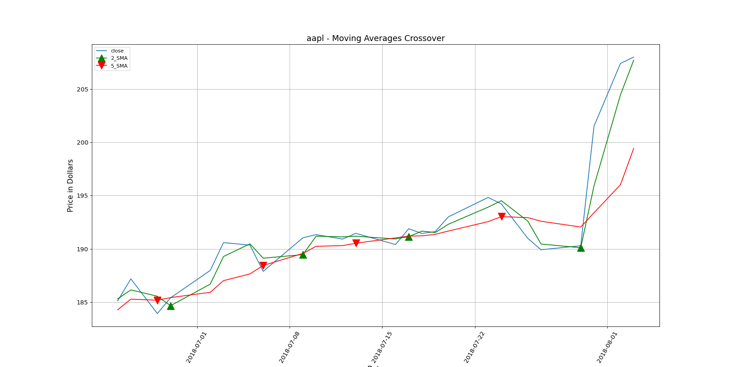


As the data set’s tweets were collected between July 9 – 19, 2018, it was necessary to get previous stock prices prior to July 9th to ensure there was sufficient data to run the tweet sentiments against starting with the first tweet and ending with the last tweet. Additional dates after the 19th were included for testing validation purposes.

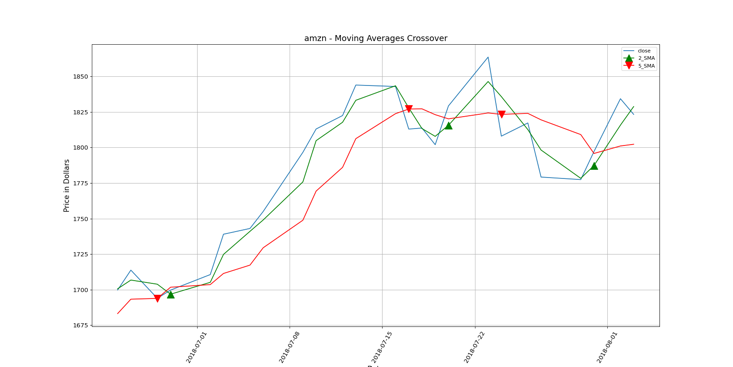
**Figure 5. Facebook (FB) stock 2- and 5-Day SMA**



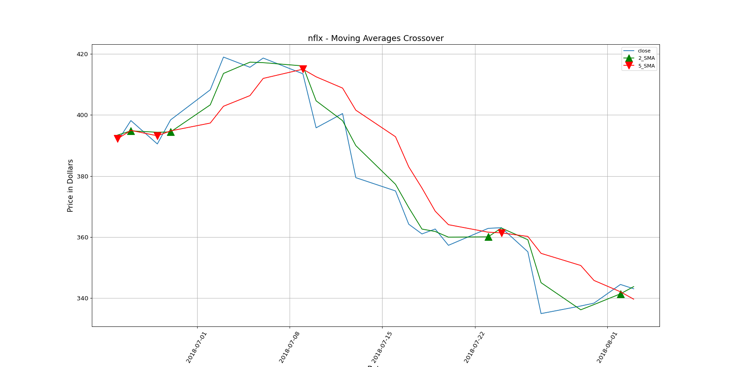
**Figure 6. Apple (AAPL) stock 2- and 5-Day SMA**



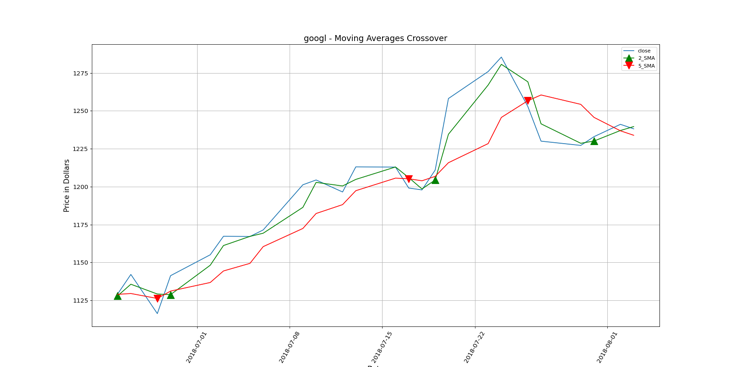
**Figure 7. Amazon (AMZN) stock 2- and 5-Day SMA**

****

**Figure 8. Netflix (NFLX) stock 2- and 5-Day SMA**

****

**Figure 9. Google (GOOGL) stock 2- and 5-Day SMA**

****

### Training and Testing Set Creation

Training and testing set creation was challenging while attempting to test the data set against various classifiers and regressors. Because of this, the formation of the training set needed to be continuously modified. Some algorithms require a one-dimensional array where others a two-dimensional. Additionally, binary arrays were required.

In the case of this experiment, all efforts were made to focus on the sentiment score as the ‘y’ variable. Because of the need of a binary array, a second column was created indicating a positive or negative score. This binary array was needed when classifying the data set and the actual sentiment score was used against regressors.

On all training and testing sets, the ‘x’ was the 2-day and 5-day SMA as well as the 7-day EMA calculations. The theory behind this decision was that if this experiment was actually attempting to test if the tweets from this data set’s sentiment does have an effect on predicting SMA then the sentiment should be tested against SMA without the background noise of additional features that might disrupt this correlation. However, when training and testing the Decision Tree Classifier, the ‘x’ was expanded to include all the basic features produced during the initial preprocessing stage (See paragragh. 4.2.1 for more details).

A total of 10 training and testing sets were utilized during this experiment. Although at times small, they were divided by company as well as which sentiment calculation was being used, VADER or TextBlob. The took some extra time to run through the classifiers and regressors, but hopefully gives a better insight not only on how these specific companies stocks match up to the trend lines during this period of time but also if there is any discernable difference in the two sentiment algorithm’s overall results.

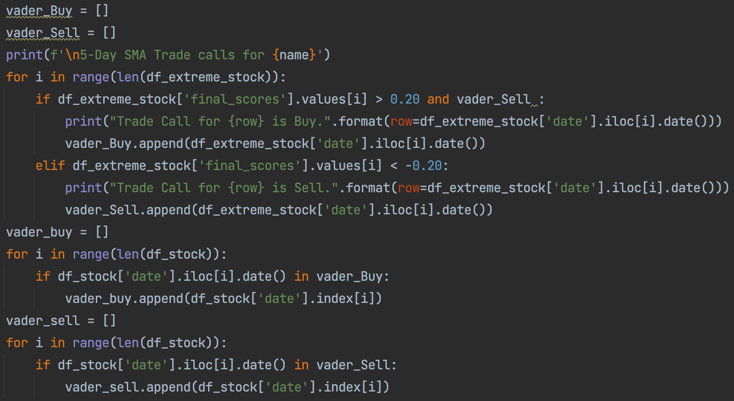
## Processing

Because of this experiment’s parameters and its data set, there were a number of processing algorithms utilized to determine whether the data set could answer any of the research questions. Besides using traditional classifiers and regressors to help with analysis, testing the tweet sentiment directly against the SMA was also useful. It was hoped that using a wide variety of tools would ensure better results while answering more questions.

### Simple Moving Average matching

### To determine if the tweet’s sentiments were at least inline with SMA calculation, a simple algorithm was created to test if any of the previously calculated SMA’s lined up with SMA’s based on the sentiment or if SMA’s based on the sentiment lined up with the SMA’s trend lines.

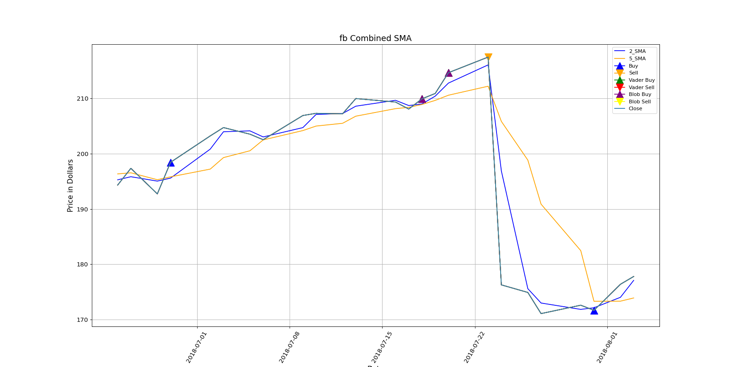
### **Figure 10. Algorithm to test sentiments against SMA**



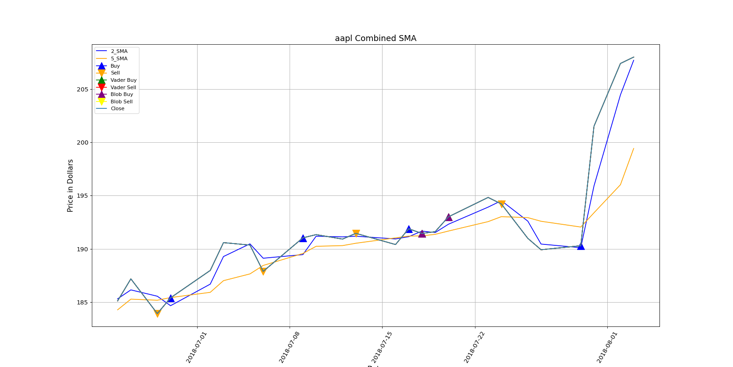
As seen in fig. 10, not all tweet sentiments precipitated a buy or sell call. Sentiment scores greater than 0.20 and less than -0.20 was needed. This gives more weight to tweets that in fact could be either positive or negative will trying to ensure any neutral tweets were excluded.

As you can see in the following figures, this test was a success when testing both the VADER and TextBlob sentiment scores:

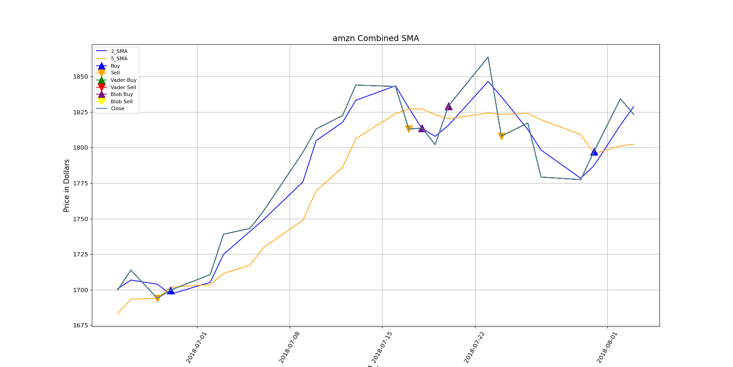
**Figure 11. Facebook (FB) VADER, TextBlob, and SMA**

****

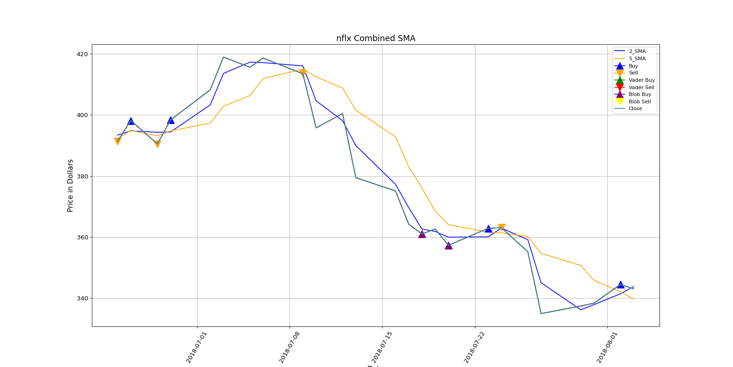
**Figure 12. Apple (AAPL) VADER, TextBlob, and SMA**

****

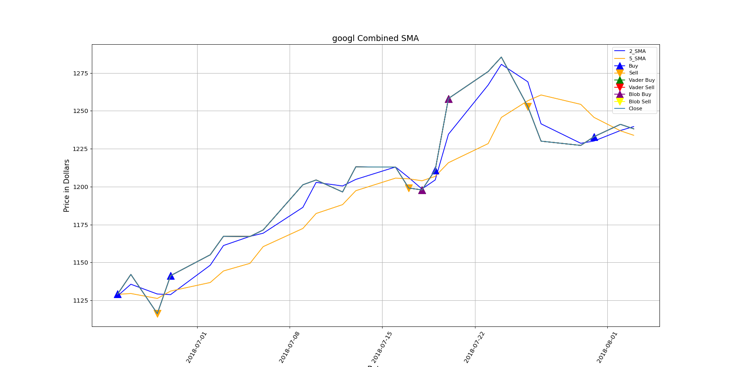
**Figure 13. Amazon (AMZN) VADER, TextBlob, and SMA**

****

**Figure 14. Netflix (NFLX) VADER, TextBlob, and SMA**

**

**Figure 15. Google (GOOGL) VADER, TextBlob, and SMA**

****

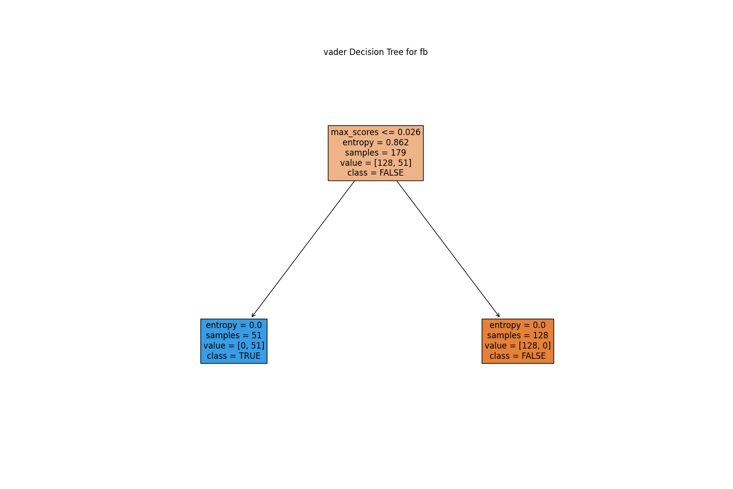
### Decision Tree Classifier

While utilizing the Decision Tree Classifier a larger training and testing set (additional features) as well as the binary sentiment score. It is unclear how reliable the results were from this algorithm due to the small size of the training and testing sets, it did however yield consistent results throughout the 10 separate training and testing sets.

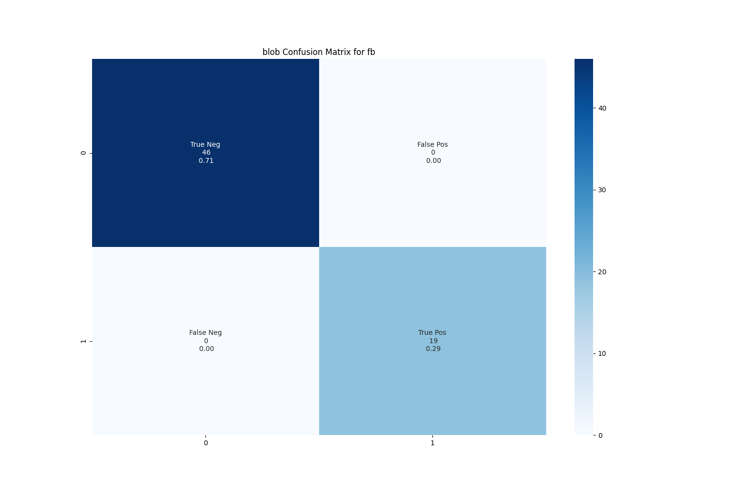
**Figure 16. DecisionTreeClassifier Attributes Used**



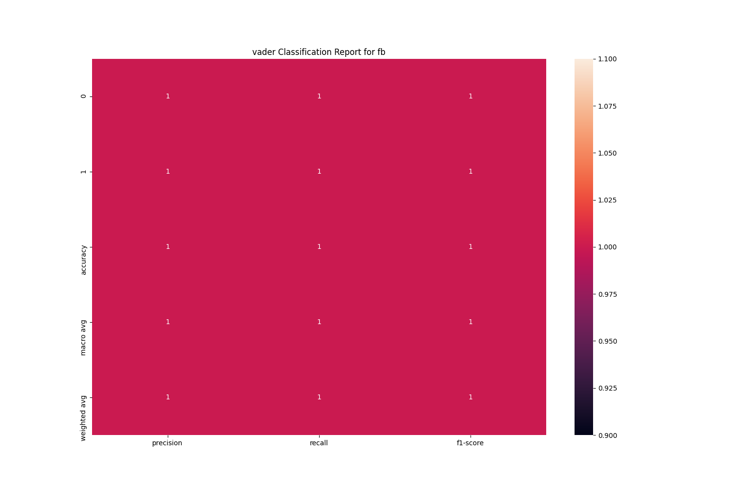
**Figure 17. Facebook (FB) Decision Tree (VADER)**

****

**Figure 18. Facebook (FB) Confusion Matrix (VADER)**

****

**Figure 19. Facebook (FB) Classification Report (VADER)**

****

The high scores and small trees where not predicted since this training and setting set was expanded. It did however provide one decision per set and it was based on the overall sentiment score. Table 1 depicts not only the one decision and the score require to be true or false.

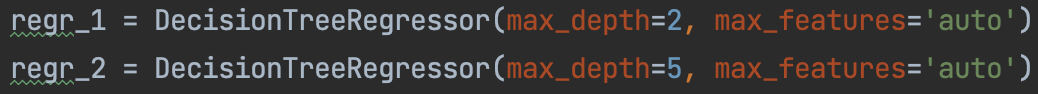
**Table 1. Overall Decision Tree Classifier Scores**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test/Trng Set** | **Sentiment Result** | **Precision** | **Accuracy** |
| VADER FB | <= 0.026 | 100 | 100 |
| VADER AAPL | <= 0.039 | 100 | 100 |
| VADER AMZN | <= 0.01 | 100 | 100 |
| VADER NFLX | <= 0.013 | 100 | 100 |
| VADER GOOGL | <= 0.07 | 100 | 100 |
| TextBlob FB | <= 0.01 | 100 | 100 |
| TextBlob AAPL | <= 0.03 | 100 | 100 |
| TextBlob AMZN | <= 0.02 | 100 | 100 |
| TextBlob NFLX | <= 0.02 | 100 | 100 |
| TextBlob GOOGL | <= 0.02 | 100 | 100 |

### Decision Tree Regression

Two Decision Tree Regressors were used during this experiment. The only change was the max depth attribute:

**Figure 20. Decision Tree Regressor Attributes Used**



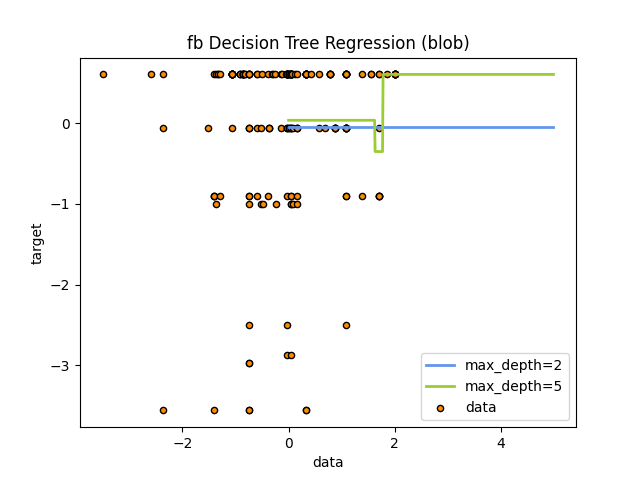
### The purpose of running two versions of the algorithm was to gain insight as to how the data set responded to each depth to get a better overall result.

**Figure 21. Facebook (FB) Decision Tree Regression (VADER)**

### 

The results shown in Fig. 19 depict the tweet sentiment score (data) against the 2-Day SMA score which shows a slight predicted bump in the SMA for max depth = 5 and even for 2. The TextBlob results are similar to the VADER results, but with an approximate quarter point increase in SMA with max depth=5. The resulting SMA are consistent by company with their respective sentiment algorithms. However, it was noted that typically, the VADER algorithm moved the SMA within the data points more significantly than the TextBlob version.

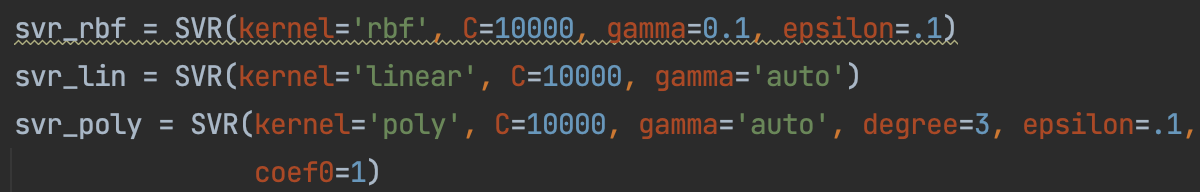
**Figure 22. Facebook (FB) Decision Tree Regression (TextBlob)**



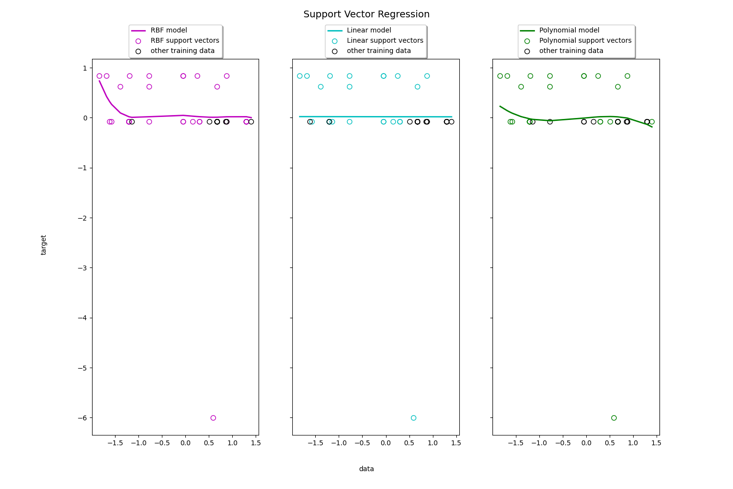
### Support Vector Regression

A second regressor type was used to ensure accuracy overall regressor accuracy. In this portion of the experiment, the SVR (Support Vector Regressor) was utilized and tested against itself using three of its kernel attributes. Linear, Polynomial and Radial Basis Function (RBF) were tested against the data sets.

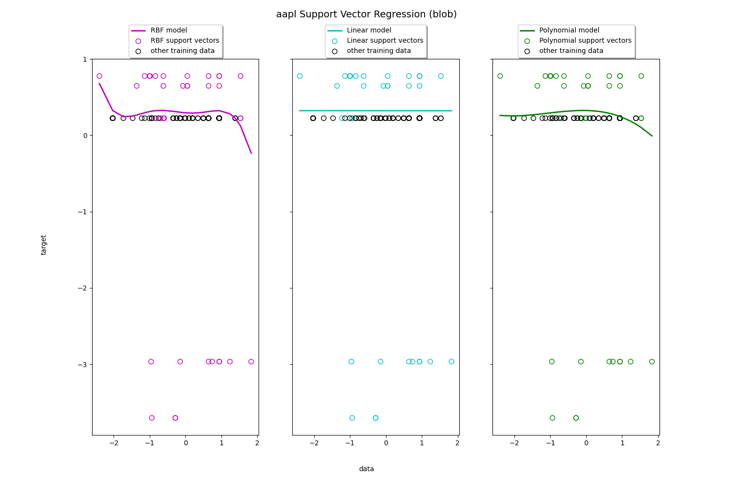
**Figure 23. Support Vector Regressor Attributes Used**



**Figure 24. Apple (AAPL) Support Vector Regression (VADER)**



**Figure 25. Apple (AAPL) Support Vector Regression (TextBlob)**

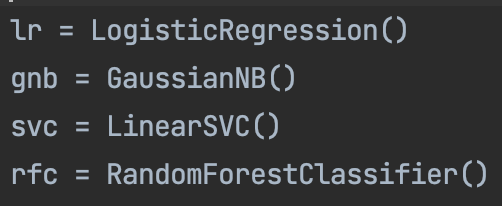


Although the Apple data set is not as large as the Facebook data set, it did yield results typical of all of the data sets. This model yielded a different result than its Decision Tree Regression model. In the other model there was a projected increase in SMA, here in both fig. 21 and 22 there appears to be a downward trend that is even more pronounced in the TextBlob data set.

### Comparison of Calibration of Classifiers

The final processing that was accomplished was to test the validity of the train and test sets. This test utilized a grouping of classifiers and regressors. Logistic Regressor, Gaussian Naïve Bayes, Support Vector Classifier and Random Forest Classifier tested the test set specifically against a standard deviation value to check calibration of the test sets and a column graph to measure the mean values. Unlike the other tests performed, this time the testing set was changed from 30% of the overall data set, to 70%.

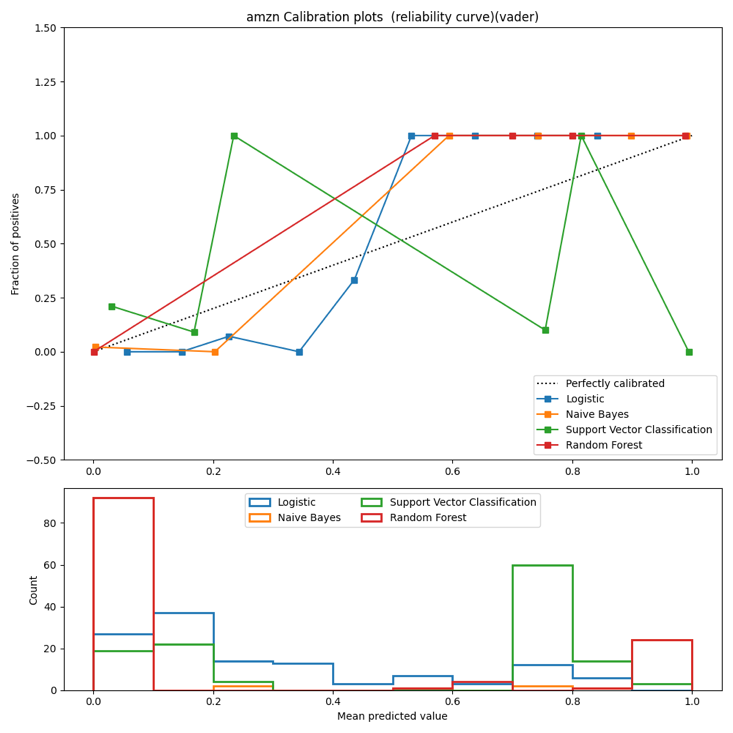
**Figure 26. Support Vector Regressor Attributes Used**



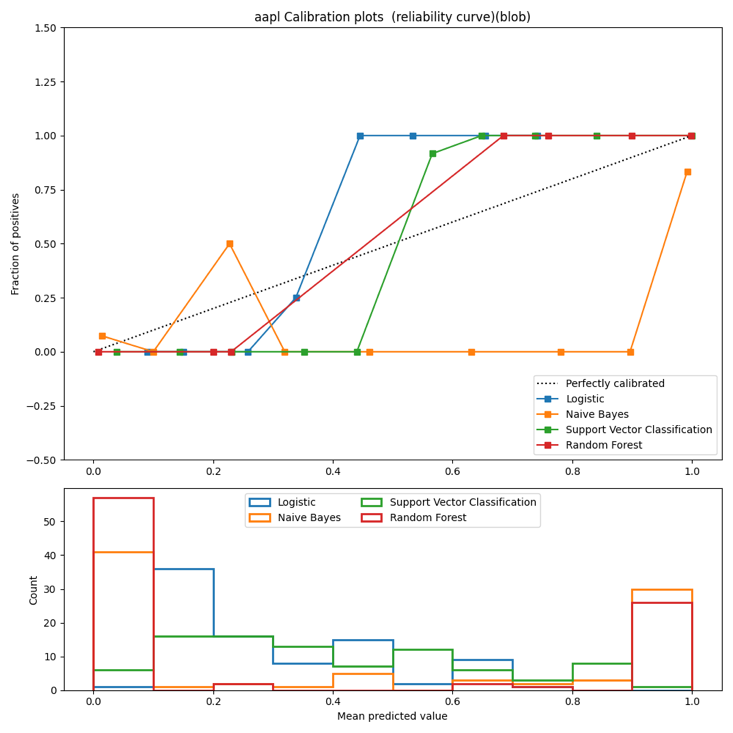
One issue that was unable to be corrected was that the Linear Regressor did not converge on any data set. The kernel nor the number of iterations did not affect this outcome. Additionally, because of the small data sets it appears the Gaussian Naïve Bayes and Support Vector Classifier also performed poorly. The Random Forest was typically closest to being perfectly calibrated although typically off-center.

These challenges are likely due to not fine tuning these classifiers, but in order to truly test the test sets it was decided to leave them in their default modes instead of potentially poisoning the test set validation results.

**Figure 27. Apple (AAPL) Calibration of Classifiers (VADER)**



**Figure 28. Apple (AAPL) Calibration of Classifiers (TextBlob)**



# RESULTS

Looking back at the research questions, it is hard to determine if this experiment was able to answer any of these. It could be argued that two of the three research questions originally asked:

* How much, if at all, does Twitter influence of stock market trading?
* Of the myriad of tweets that used the hashtag: #stockmarket, which tweets may have influenced stock market trading by mirroring simple moving averages?
* Do tweet sentiment trends correlate with stock prices being bullish or bearish?

were at least partially answered.

If Twitter does have an influence, a far larger data set and intra-day prices would be required to honestly answer that. A week and one-half worth of tweets is not enough to establish this even in the short-term.

It was determined at least that the tweets in this data set did have a very close relation to 2-Day Simple Moving Averages. With this small of a data set, again it is difficult to discern if this was coincidence because the data set skewed heavily positive and most stocks saw gains during this time period. In figs. 11 – 15 calculated and sentiment determined SMAs matched up. So, at least in the short term we can conclude that there may in fact be some correlation.

The last research question is again difficult to ascertain. In the short term, July 9 – 19, 2018, yes it seems to. This small data set leaned positive and there were positive movements in most stock prices. Perhaps another maybe another question answered.

The experiment was not a total loss, there was some compelling research that if can be replicated at scale may yield definitive results. If at scale with a much larger data set within a longer time period these same results could be achieved while matching the SMA data. It may be a perfect opportunity to attempt to predict the Twitter-verse sentiment trends to determine if they can be matched to stock SMA trends.

# CONCLUSION

Although these results are not going to produce instant market millionaires, some promising work seems to have been accomplished. If given better data on the onset of this project potentially results could have been better. However, if this research is to continue, this simple experiment could easily be the starting point to expand this exponentially.

# ACKNOWLEDGMENTS

Our thanks to the members of the Computing and Cyber Security Department, Texas A&M University – San Antonio especially for the guidance provided by Dr. Izzat Alsmadi.

# REFERENCES

1. Eugene Fama., 1965. Random walks in stock-market prices. *Financial Analysts Journal, 21.* (Nov. 1965), 55-59.
2. Farzindar Atefeh and Wael Khreich. 2015. A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.* 31, 1 (February 2015), 132–164. DOI: <https://doi.org/10.1111/coin.12017>
3. Yubo Chen and Jinhong Xie. (2008). Online Consumer Review: Word-of-Mouth as a New Element of Marketing Communication Mix. Management Science. 54. 477-491. 10.1287/mnsc.1070.0810. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
4. Joel Comm, Dave Taylor, and Guy Kawaski. (2015). Twitter Power 3.0: How to Dominate Your Market One Tweet at a Time. *Wiley.* (Mar, 2015).
5. J. Bollen and H. Mao. (2011) Twitter Mood as a Stock Market Predictor. *Computer, vol. 44, no. 10*. pp. 91-94, 2011. DOI: 10.1109/MC.2011.323
6. John R. Nofsinger. (2005) Social Mood and Financial Economics. *The Journal of Behavioral Finance, 6:3*, 144-160. DOI: 10.1207/s15427579jpfm0603\_4