An Analysis of Tweets Concerning Top NASDAQ Companies

James Mick

Department of Computer Science, Texas A&M University-San Antonio

CSCI 4391: Senior Project

Dr. Izzat Alsmadi

May 5, 2021

**Abstract**

The goal of this paper is to determine if there is any prediction can be made between tweets concerning the following companies: Apple, Amazon, Google, Microsoft, and Tesla. The analysis was performed by James Mick who is an undergraduate student at Texas A&M University-San Antonio. The analysis was performed on a set of tweets which contained the company’s ticker symbols collected over the period of January 1, 2015 through December 31, 2019. Previous analysis on this dataset has been performed by several researchers and can be found on the dataset’s Kaggle page. Python was used to consolidate and modify the data before using classifiers to determine if the stock prices were predicted by several different meta-features. The analysis shows that it is likely that interaction level and sentiment likely have some predictive qualities based on the meta-features added. The price showed some predictive qualities, but new meta-features are required to find better predictive values.

The stock market is fickle and can be influenced by both good and bad news. Many people use Twitter to maintain continual news about their interests. Investors are no different. As such, it is essential to determine if tweets that investors are seeing everyday influence the prices of stocks.

**Data collection and organization.**

The data for this project was collected by Kaggle user Ömer Metin and can be retrieved from <https://www.kaggle.com/omermetinn/tweets-about-the-top-companies-from-2015-to-2020>. Ömer collected tweets containing the following tags: AAPL (apple), GOOG (Google Inc), GOOGL (Google Inc), AMZN (Amazon.com), TSLA (Tesla Inc), and MSFT (Microsoft). The tweets were collected beginning January 1, 2015 and ending December 31, 2019. The data is contained in 3 comma separated values documents called Company.csv, Company\_Tweet.csv, and Tweet.csv.

Company.csv contains two columns which are: ticker\_symbol and company\_name. Company\_Tweet.csv contains two columns which are: tweet\_id and ticker\_symbol. Tweet.csv contains seven columns which are: tweet\_id, writer, post\_date, body, comment\_num, retweet\_num, and like\_num. This analysis will merge the Tweet and Company\_Tweet datasets on the column tweet\_id. Company.csv is not useful for this analysis, therefore, it will not be used.

**Research questions.**

This research will attempt to learn if tweets about companies influence the companies stock value, whether user interactions (likes, retweets, comments) influence the companies stock value, and whether a tweet’s sentiment influences the company’s stock value.

**Kernels, discussion, and relevant literature.**

Other researchers have used these datasets to attempt to answer various questions concerning the stock market and twitter. The research performed attempts to connect the volume of tweets about a company to the volume of stocks traded on that same company, whether sentiment correlated to changes in stock price, and whether twitter bots could affect a stock’s price.

Kaggle user Darrenljw was interested to find out if the quantity of tweets about a company on a given day influenced the number of trades made for that company on the same day. He used stock data available from Yahoo Finance to chart the volume for each day of the study. He then used Pandas to count the number of tweets for each company on each day of the study. Those values were graphed along with the information from Yahoo Finance. Darrenljw was able to show a small level of correlation between the two data, but the correlation was weak.

Darrenljw also attempted to determine if the sentiment of tweets about a company led to fluctuations in stock prices. He used a package called Afinn to determine the sentiment of tweets. Next, he graphed average sentiment for each day along with the stocks closing price. This analysis showed a much better correlation than the volume analysis. Tesla showed to be a bit of an outlier compared to the other stocks as it did not correlate as well in regard to tweet sentiment.

Another Kaggle user, Alex Kozlov, performed research to root out bots and determine if the bot’s tweets influenced the stock. He looked for bots by determining how many unique tweeters were in the set, and then using Pandas to see which tweeters were tweeting the most and the fastest. He compared that data to which tweeters were regurgitating other user’s tweets. He assumed that those who tweeted the most frequently, the most volume, or the least unique tweets were likely bots. He evaluated the bot’s tweets with Vader from the Natural Language Tool Kit to assign a sentiment level. He then calculated the correlation coefficient between the company’s closing price to the sentiment of the bot’s tweets. The analysis showed a positive correlation between the tweet sentiment and the stock price in a range between 0.4 and 0.56. This likely means that as more tweets show positive sentiment, the price of the stock is likely to also rise and as more tweets show negative sentiment, the price of the stock is likely to fall.

**Data Processing.**

The dataset for this analysis is quite large and is contained in three separate files. To use the dataset, it must be modified. The data will need to include the stock ticker symbol for the company mentioned in the tweet, the change in price for each given day, a rating of the level of interaction that the tweet has received, and a rating of the polarity of the tweet. In addition to these additions, the date needs to be modified from a relative date to an absolute date.

Company\_Tweet.csv contains a list of tweet\_id’s as well as the company that is mentioned in the tweet. Tweet.csv contains a tweet\_id column as well as the other relevant data concerning the tweets. Pandas can be used to merge the two datasets where they share the same tweet\_id. Pandas can also be used to convert the post\_date column to a new column called date which uses absolute dates instead of relative dates.

The yahoo finance module (yfinance) was used to obtain the stocks adjusted close values over the range of the dataset. The dates in the stock’s dataset were used to only monitor tweets that occurred on days where the stock market was open. Next, the stock data was used to calculate the change in price for each day by subtracting the previous day’s adjusted close from the next day’s adjusted close for each company and each day. A column called ‘price\_diff’ was added and populated with either increase, decrease, or same depending on the change in the stock prices. There were less than 2000 entries where the stock price showed no change, so those rows were deleted.

The interactions included in the dataset included likes, retweets, and comments. This data was manipulated to determine the average number of each interaction received given the condition that the tweet had at least one like, one retweet, or one comment. This analysis showed that likes were roughly seven times more frequent than comments and three times more frequent than retweets. Comments and retweets were corrected so that they weighed evenly with likes, and all interactions were summed together. The interaction level was determined to be none if the tweet received no interactions, low if the tweet received between one and the average of the combined interactions minus its standard deviation, medium if the tweet received the mean plus or minus one standard deviation, and high if the tweet received more than the mean plus one standard deviation.

Polarity can be determined using various modules in Python. This analysis uses the WordBlob module to determine each tweet’s polarity. This polarity rates each tweet between negative one and one. Sentiment is assigned such that polarity less than negative zero point five is negative, negative zero point five to zero point five is neutral, and greater than zero point five is positive.

Each stock can be considered a variable by its own right. As such, the tweets were grouped by the company that each tweet concerned. This will help the researcher to determine if one stock has better predictive abilities than the others.

**Feature selection.**

Various meta-features are added to the dataset to determine what attributes of a tweet are likely to predict certain outcomes. New columns include: wordcount (number of words in the tweet), unique word count (number of non-recurring words in the tweet), character count (number of characters in the tweet), punctuation count, upper-case word count, lower-case word count, average word length, special character count, number character count, upper-case letter count, lower-case letter count, swear word count, typo count, and polarity (sentiment). Additionally, a bag of words was added that include a column for each of the 100 most common words in the tweets.

**Test.**

Each training data set, 18 total, was run through the following classifier Python modules: DecisionTreeClassifier, DecisionTreeRegressor, and RandomForestClassifer. For each run, the y and y\_train variables are set to the column that is to be tested by each classifier. The nominal columns are removed from the X\_train and X\_test set. Additionally, any column that is used to calculate y is removed.

The DecisionTreeClassifier is run in entropy mode and precision and accuracy are recorded. The DecisionTreeRegressor is run in mae mode, but there is no output for this method regarding precision or accuracy. As such, the DecisionTreeRegressor will not be used in the analysis. Finally, the RandomForestClassifier is run with the following selections: oob\_score = True, n\_jobs = -1, n\_estimators = 1000, max\_features=100, and random\_state=0. The results from the RandomForestClassifier are recorded and compared to the results from the DecisionTreeRegressor.

The precision and accuracy are a result of how the meta-features selected relate to how the target column has been rated. The data must be deeply analyzed, and new meta-features added to improve these scores. The researcher was unable to find meta-features that provided better results. The trees and forests that are output by the modules can be pruned to find the meta-features that have the greatest impact on the target even though the precision and accuracy are not improved. Pruning is a time intensive operation. As such, only one company from each target was pruned for this analysis.

**Analysis.**

The following table shows results obtained by running the training data sets through the classifiers.



The data shows that the targets chosen are not predicted very well based on the meta-features selected.

***Interaction.***

**Decision tree vs. random forest.** The difference between the precision determined by the decision tree classifier and the random forest classifier suggests that there is correlation between the meta-features and interactions. It is possible that interaction has correlation with what is being tweeted and how the tweet is constructed, but who posted the tweet may have a large effect on whether the tweet receives user interactions. The data that was put through the classifiers had no column that classified the tweet by writer. With better meta-features or a better way of calculating user interaction.

**AAPL interaction pruned tree and optimized forest.**  The tree created by the decision tree classifier suggests that posts that share owler alerts are more likely to have a high amount of interactions. Of the remaining tweets that do not mention owler, robinhood can be used to classify stocks by interactions. Stocks that mention robinhood and have words of 6 letters or more on average or 6 or more words that appear in title case should rate as high interaction. There are many more classifiers as even the pruned tree has 30+ branches to make its decision. The random forest begins classifying the data using the feature mean\_word\_len.

Like the decision tree, tweets with words that average 6 letters, or more, are more likely to rate as high interaction, whereas less than 6 letters lead to more medium, low or no interactions. The random forest tree that is output by the code uses 29 branches to make its classification decision. None of the features that determine the interaction level involve the price change occurring on the day of the tweet.

***Price.***

**Decision tree vs. random forest.** The price change from one day to the next has higher precision and accuracy for training to matching the meta-features selected than the interaction level showed. There was little/no difference between the decision tree and the random forest which may suggest that the meta-features that have been selected have some correlation with the price change, but there are other factors which affect stock prices that are not accounted for in the dataset.

**MSFT price change pruned tree and optimized random forest.** The tree created by the decision tree classifier shows that the biggest decider for whether the price will increase, or decrease is whether the tweet has more than five uppercase letters. Those with more than five are more likely to see a decrease in stock price, while the opposite is true for those with five or less. Polarity (the value used to determine sentiment) is found in this tree on the third branch. The tree shows that when a tweets sentiment is greater than 0.362 it is likely that the price of the stock is increasing on that day. This is logical because people are likely to tweet positive things when their stock is gaining value. Farther down in the tree, retweet\_num can be found. The classifier suggests that when a tweet is retweeted less than five times then there is a greater chance for the stock price to increase. This seems counterintuitive to the researcher and he cannot think of a valid reason for this result. Sentiment and retweets do show some promise into whether a stock price will rise or fall, but this classifier is only 50 percent accurate when making its decision.

The individual tree that is shown from the random forest classifier shows that the length of the tweet is the main classifier to use to determine a price change. Longer tweets, over 67 characters, are more likely to predict a price increase. No classifiers in this tree are concerned with sentiment or interaction.

***Sentiment.***

**Decision tree vs. random forest.** Both the decision tree and random forest show a marked increase in correlation between sentiment and the meta-features. There is a sharp increase in precision when switching from a random tree to a random forest. This is good indication that there is either some missing meta-feature which will increase the precision, or some combination of meta-features which is likely to improve the model performance. The higher correlation makes logical sense as sentiment is created by measuring the polarity of each word in a tweet. There may be a higher correlation between the sentiment and the dataset if the bag of words were structured to include more positive or negative words and less neutral words.

**GOOG sentiment pruned tree and optimized random forest.** The key decider in the decision tree for whether sentiment will be positive, negative, or neutral is whether the word more appears in the tweet. Tweets containing the word more are increasingly likely to be categorized as positive than tweets that do not include the word more. Price\_diff appears at least three times in this tree. The occurrences of price\_diff seem to suggest that when the price increases by more than $6.38, and the word more appears in a tweet that the tweet will have a positive sentiment. When the word more is not included in the tweet, a price difference of $11.32 is required to sway the sentiment to positive. Interestingly, if the price difference decreases by more than $37.47, then the sentiment becomes neutral and negative sentiment is shown between negative $37.47 and $11.32. There are many branches between when a tweet contains the word more and the price difference appears, so it is very possible to have tweets in the above ranges that do not match the sentiment as described above. The researcher is not sure if tweet sentiment is affecting the stock price or if stock price is affecting sentiment or if this is a feedback loop.

The random forest shows that the key decider is whether the tweet contains more than 12 special characters. Tweets that have 12 or less special characters skew positive whereas those with 13 or more skew negative. The random forest does not compare sentiment to interaction or price.

**Conclusion.**

The goal of the project was to show correlation between tweets about a company and that company’s stock price. Specifically, the goal was to determine if the level at which tweets were liked, commented, or retweeted affected the stock price, or if the sentiment of the tweets about a company would affect the stock price. The data was processed so that classifiers could be used on interaction level, price change, and sentiment. The classifiers showed limited usefulness in predicting outcomes based on the meta-features submitted. Price change did show some correlation with sentiment and the quantity of retweets, but the precision was only about 50 percent. Sentiment showed some correlation with stock prices, but again the precision rarely exceeded 70 percent. Overall, this project showed a very limited connection between tweet interaction and price, and a moderate connection between price and sentiment. Unfortunately, it failed to determine a link that can accurately predict price changes.

References

Metin, Ö. (2020, November 26). Tweets about the top companies from 2015 to 2020. Retrieved March 17, 2021, from https://www.kaggle.com/omermetinn/tweets-about-the-top-companies-from-2015-to-2020