Analysis of Reddit Sentiment as a Stock Market Predictor

Andrew Hill, Dunovan Rodgers, Jeremiah Kincaid, and Alex von Jena

ISA 414 | Section A

Dr. Arthur Carvalho

May 6, 2022

Table of Contents

I. Abstract.....2

II. Project Overview.....3
- A. Business Background
- B. Business Problem Definition

III. Data Collection and Preparation.....4
- A. Reddit Sentiment Data Collection Process
- B. AlphaVantage Data Collection Process
- C. Data Preprocessing and Cleaning
- D. Sentiment Analysis Process

IV. Data Analysis and Evaluation.....8
- A. Pearson Correlation Analysis
- B. Simple Linear Regression Model
- C. Regression Model Assumptions
- D. Regression Model Evaluation

V. Conclusion 13
- A. Hypothesis Confirmation
- B. Pros and Cons of the Solution
- C. Limitations and Areas for Improvement

VI. References 14

Abstract
Financial forums on Reddit are a popular, yet polarizing setting to discuss the stock market. In 2021, r/WallStreetBets in particular gained notoriety for performing a short squeeze on GameStop (GME), causing a spike in its value, freezing trade of the stock multiple times, and bringing attention from the media and government alike. With Reddit making the stock market increasingly accessible to the average person, and in response to the short squeeze, we sought to determine whether sentiment of various stock tickers on stock market oriented subreddits would be a good predictor of their short term change in price. In the end, we determined that the change in stock price corresponding to the NLTK sentiment analysis had a p-value of 0.852, meaning the sentiment of a stock price on Reddit is not a significant predictor of its performance at any reasonable level of significance. For this reason, we advise against using popular opinion of a stock on Reddit as a factor in predicting stock performance in the short run. However, further analysis could be conducted by testing trends over a longer period of time or comparing the general sentiment of a comment against its karma.

KEYWORDS: Reddit, Stock Market, Sentiment Analysis

Introduction

Business Background

The stock market as a whole is as complex as it is difficult to predict. There are thousands of different variables that could be used to evaluate and predict stock price, ranging from demand of a product to the political affiliation of the company owner. People have been trying to predict future stock prices using a variety of different methods since the dawn of the market itself. Most have been disappointed in their efforts.

The goal for this project is to examine if stock performance can be predicted by Reddit users. Many experts offer predictions attempting to gauge the stock market but most are ultimately inconclusive due to the variability of their accuracy. We wanted to pull data from a source that anyone, at any experience level, could access and convey their own opinions about stocks with other users. We decided to use Reddit to predict stock performance because Reddit is a free online source where anyone can comment. The structure of Reddit communities, known as subreddits, also makes it easy to find topical information. This will help to indicate the validity of using this social media platform as a starting point for investment evaluation.

Business Problem Definition

Our business problem for this project is to try and see if the comments and posts on the social media platform Reddit have any merit when it comes to predicting the price movement of certain stocks. In order to achieve this we must utilize text based sentiment analysis for the Reddit comments so that each comment has its own positive or negative score. Due to the financial subject matter of these comments we have opted to run sentiment analysis using not only a standard lexicon, but also one geared specifically towards analyzing financial documents in hopes to obtain a more accurate sentiment score. After getting the scores of all of the comments we will need to group them by stock in order to evaluate them, while doing so we also need to average all of the separate sentiment scores for the comments under each stock, by day.

Following the collection and creation of the average sentiment score and stock ticker data table, we will need to get the open and close prices of all the days in which we collected comments for each separate stock. We decided that we would collect the open and close prices off of alphavantage.co, a website that gives a free API key to access a variety of information pertaining to stocks. After pulling the open and close prices of all of our stocks of interest we will then combine the two data tables using Date and Ticker as the primary keys. Once the final data table is clean we will create an aggregate column that calculates the percent change of each stock for each day that we have comments for. As soon as the final data table is prepared and the percent change column added we will begin to see if our predictions are correct. We will be using regression to evaluate whether the average daily sentiment scores of the Reddit comments are a viable predictor for the percent change of price for that day.

We came up with three hypotheses for our analysis, the first being that the sentiment data has a positive correlation with stock prices and is an accurate predictor of those prices, or the opposite where the sentiment data counteracts the real price data of the stocks. Our final hypothesis, and the one that is the most likely outcome, is that the sentiment analysis of Reddit comments and posts will have no correlation in predicting whether or not that stock's price increases or decreases. We could see a world where discussions of the stock market on Reddit proved to be a self-fulfilling prophecy (or at least be emblematic of the wider phenomenon), or alternatively, the general consensus of Reddit proved to be particularly unreliable against the market as a whole. Ultimately, however, we felt no particularly positive or negative influence would prevail, general sentiment would coincide with the market as a whole, and that there would be too much noise to find a definitive answer. Although we found it most likely that there would be no strong correlation between Reddit sentiment and stock performance, testing this would still be worthwhile in confirming the idea not to use this information as an indicator for investments.

Data Collection and Preparation Process

Reddit Comment Collection

The data collection and preparation process was one of the most difficult and technical aspects of this project. The first step in this process is collecting Reddit comments. We chose to gather insights from three of the most popular stock investment subreddits, r/WallStreetBets, r/Stocks, and r/Investing. Although this analysis could be done on any subreddit where investment securities are discussed, we felt it was important to choose subreddits that provided a high volume of data, and were more likely to be representative of Reddit as a whole. Additionally, while we considered r/WallStreetBets too important of a subreddit to exclude from our analysis, we felt its unique culture demanded we collect data from other subreddits to account for potential variation in language and opinion.

We used the Reddit API to request all top level comments from a defined number of posts in each subreddit. This is a simple process but takes significant time due to the Reddit API's ratelimit and the amount of data needed to run proper analyses on. Before processing the comments, comments made by known bots such as "AutoModerator," which comment on every post automatically, were removed as they contain no user insight. From there, we utilized regular expressions to identify anything in each comment that could be in reference to a stock ticker. This included any string of 2-4 capital letters in sequence and any string of 2-4 letters immediately following a dollar sign (e.g., AAPL, $Amd). Tickers that were found were compared against a blacklist of common strings that would frequently appear in an investing forum but are not stock tickers and those found to be erroneous were removed. Everyday language, such as "LOL," business and economic jargon, such as "CEO," and slang specific to Reddit, such as "OP," were all accounted for in building this blacklist. Of course this could never be a complete list of capital strings that may be found, but removing these comments now saves processing time later.

```
hits = re.finditer(("([A-Z]{2,4})|(\$[A-z]{1,4})"),
top_level_comment.body) #Finds text in the comment that
appears to be a stock ticker
tickers = [m.group(0) for m in hits]
tickers = [re.sub("\$", "", i) for i in tickers]
#Removes the '$' from in front of some tickers
tickers = [x.upper() for x in tickers] #Makes all tickers
uppercase
```

Sentiment Analysis

Comments that were found to have a potential stock ticker in them are run through two different sentiment analysis programs. The first utilizes the Loughran and McDonald Sentiment Master Dictionary, a lexicon developed to analyze financial documents and gauge text sentiment given a business context where specific words may have alternative meanings than what would be expected given a different context. Given the somewhat financial nature of our data, this seemed an appropriate use of this dictionary. To perform this analysis we used a program developed by Dr. Kai Chen, an accounting professor at Wilfrid Laurier University. This program receives the given text, in this case a reddit comment, and performs a count of positive and negative tone words and a negation check which we used to calculate an aggregate sentiment score.

To further substantiate our data, we added an additional sentiment analysis using a different method and dictionary. Using the Natural Language Toolkit, we ran the comment through the Sentiment Intensity Analyzer which utilizes the Vader lexicon to return an aggregate sentiment score for each comment. One of the primary purposes of including this additional analysis is that the Loughran-McDonald dictionary works best with longer documents that provide more word matches than most Reddit comments could. The Vader dictionary is more likely to provide some measure of sentiment than the previous analysis.

Preprocessing

After comments have been collected and sentiment analyses have been performed for all three subreddits, the data is put into a dataframe containing the comment ID, comment text, karma score (net upvotes or downvotes of other reddit users), Date and time comment was made, Loughran-McDonald sentiment score, NLTK Vader sentiment score, stock tickers found in the comment, and the subreddit it was collected from. Because many comments contain multiple tickers, that column then must be “exploded” to make a separate row for each one, the rest of the data remaining the same. One limitation present in this method is that it is not possible for us to differentiate between sentiment of different tickers present in one comment, only comment sentiment as a whole. If a Reddit user were to comment, “I like $GME. I dislike $IBM.”, the net sentiment would be zero and therefore the sentiment for both of those stocks, despite being

oppositely positive or negative, would both be zero. Although this may reduce the richness of our data, it is ultimately not problematic due to the sentiments canceling each other out. Regardless, separating the tickers in each comment is necessary to match the ticker with its daily pricing.

After separating the tickers in each comment and determining sentiment, the comments are grouped together by subreddit, date, and ticker, then aggregated by sentiment score average and frequency of ticker appearance by the grouped together values. That dataframe is cleaned and then filtered to only include comments made in the last week. The date filtration was performed just for the purpose of this analysis but could be easily excluded or changed to include a different length of time. From that data, a subset was made to include the ticker and appearance frequency within the last week so that we can narrow our analysis to a set number of the most frequently discussed stocks.

Stock Price Collection

The next step in our data collection is obtaining stock price information for the most popular distinct stocks present in our data. There are a variety of APIs publicly available to obtain this information, but unfortunately most are either expensive to utilize or have a free version that is incredibly limited in capability. We only required very basic financial data such as open and close prices for a given stock for each day and essentially any stock data API should be able to accomplish this. Therefore, instead of searching for an API based on capabilities of rich data, our decision would instead be influenced by ratelimits allowed free of charge. On that front, there are no particularly favorable options. One of the most popular APIs of its kind right now, the Alpha Vantage API, was chosen for its ease of use. For every stock ticker in our list of the most popular securities, a request was made using that symbol which automatically obtained the last 100 days of price data and returned the daily open, close, high, low, and volume.

In order to avoid manual entry, a for loop was used to put each of the most occuring tickers in the designated section of the API key. Each iteration of the loop returned a JSON file, which was then appended to the list. The ratelimit of 5 requests per minute and 500 requests per day made this a time consuming and delicate process. We had to be very particular as to how we went about testing our code so as to not hit the daily rate limit. This meant a 13 second rest was taken between each iteration of the loop.

Sometimes suspected stock tickers that are not present in the blacklist are popular enough to make it into the list for top stocks of the week. When there is no actual ticker associated with these terms, the API returns an error (or in cases of a delisted company, only data predating our analysis) which had to be removed from the list of data. Using the list of tickers obtained earlier, the list of JSON dictionaries is searched through to obtain the date, open price, close price, and to calculate the daily percentage change before the data is appended to a list. A dataframe is formed

with these columns and is once again filtered to only include data from the previous week as opposed to the last 100 days. This information is then merged with the Reddit Comment data using an inner join on Tickers and Date and is then exported as a csv file for analysis.

```
x = 0
stock_week = []
while x < len(stock_data): #Access the json returned by the API
to format stock info
   tick = stock_data[x]["Meta Data"]["2. Symbol"] #Gets stock
ticker
   for key, value in stock_data[x]["Time Series
(Daily)"].items():
         key_item = key
         open_item = float(value["1. open"]) #Gets open price
         close_item = float(value["4. close"]) #Gets close price
         percent_change = (close_item-open_item)/open_item
#Calculates percent change
         temp_week = [tick, key_item,open_item, close_item,
percent_change]
         stock_week.append(temp_week) #Adds all data to a list
   x=x+1
```

Data Analysis and Evaluation

Pearson Correlation Analysis

The goal of the data analysis is to examine the relationship between a stock's sentiment score and their percent change. The relationship between the two variables are observed using the Pearson correlation. The Pearson correlation coefficient is a correlation measure used to identify the strength of two quantitative variables' linear relationship. It normalizes the covariance of the two quantitative variables, which is done by dividing it by the standard deviation. Pictured below is the Python code and output of the Pearson correlation coefficient of the NLTK sentiment versus the actual stock percent change. Based on the output, the correlation between the two variables is -0.013 which means there is an extremely weak, negative relationship between the NLTK sentiment and the stock percent change. This relationship will be visualized as a scatter plot later in the analysis when testing the linearity assumption for the OLS regression model.

	NLTK_Sentiment_mean	Percent_Change
NLTK_Sentiment_mean	1.000000	-0.013202
Percent_Change	-0.013202	1.000000

```
df1 = df[['NLTK_Sentiment_mean','Percent_Change']] # subset
the data
pearsCorr = df1.corr() # calculate Pearson correlation
print(pearsCorr)
```

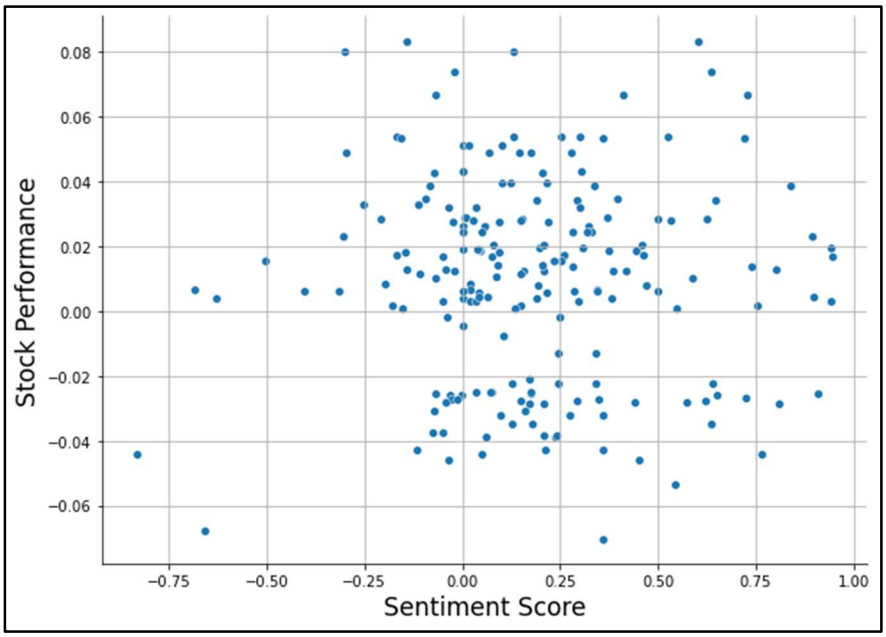
Simple Linear Regression Model

The final analysis is a simple linear regression (SLR) model predicting percent change of a stock's percent change using the Natural Language Toolkit (NLTK) sentiment scores for each stock. For the SLR regression model, NLTK sentiment was used as the predictor instead of the Loughran and McDonald (LM) sentiment analysis. When the stock percent change was regressed on the LM sentiment score, the model did not pass any of the model assumptions. This could lead to misleading or incorrect results. Before evaluating the model's R-squared value, coefficient value, and statistical significance, the four assumptions of SLR are tested. The model is an ordinary least squares (OLS) regression and was constructed using the statsmodels library in python shown in the code chunk below.

```
Y = df['Percent_Change']
X = df['NLTK_Sentiment_mean']
X = sm.add_constant(X) # add constant column
model = sm.OLS(Y, X, missing='drop') # run the
model
model_result = model.fit()
```

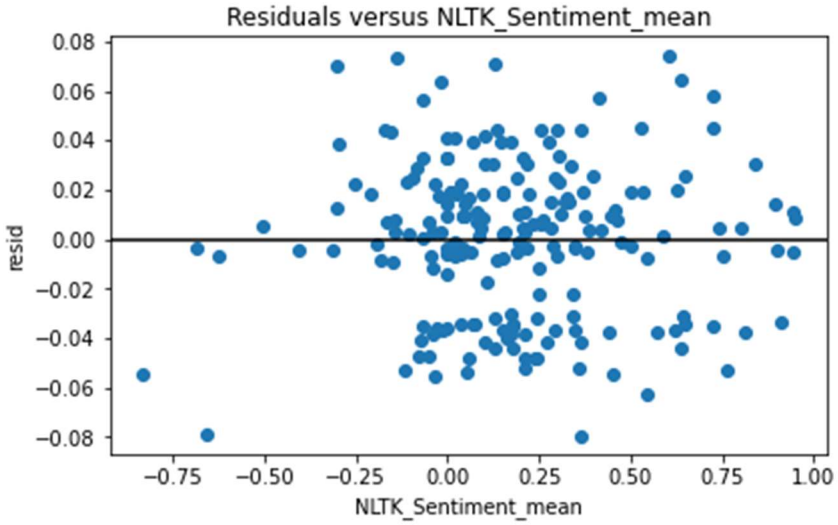
Regression Model Assumptions

The first assumption, linearity, assumes the relationship between actual stock percent change (percent change) and NLTK sentiment scores are linear. Linearity is evaluated by observing the scatterplot which is pictured below. The scatter plot confirms the value of the Pearson correlation coefficient from earlier. There is evidence that there is a very weak relationship or no relationship at all.



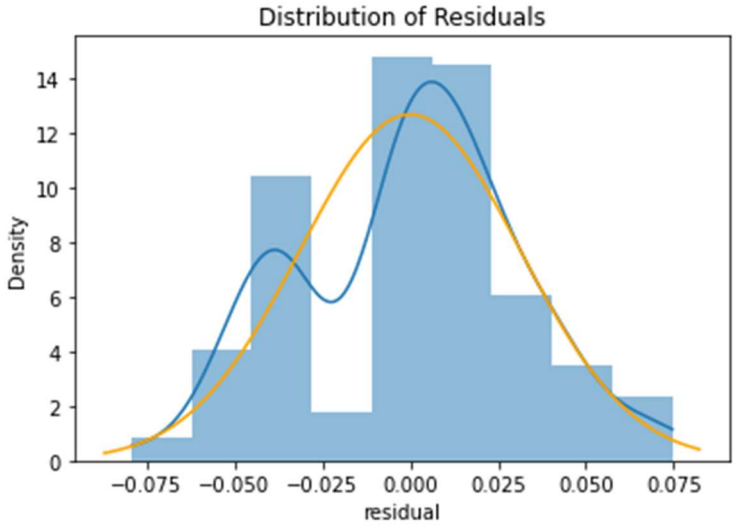

```
# test the linearity assumption
scatter = sns.relplot(data=df, x='NLTK_Sentiment_mean',
y='Percent_Change', kind='scatter', height=6, aspect=1.4)
scatter.set_axis_labels(x_var="Sentiment Score", y_var="Stock
Performance", fontsize=17)
plt.grid()
```

Next, the assumption of independence and equal variance are tested by examining the scatter plot of the model residuals and the fitted values of stock percent change. In order for the model to be independent of errors, there should not be a visible relationship between the target variable and the residuals. For equal variance to be met, the points on the scatter plot should be consistent along the x-axis and exhibit no pattern. Based on the plot below, the points are somewhat consistent along the x-axis. Additionally, the points do not form any shape or pattern. Therefore it can be concluded that the relationship between the residuals and the target variable are independent of each other. Additionally, there is evidence of equal variance among all values on the x-axis.



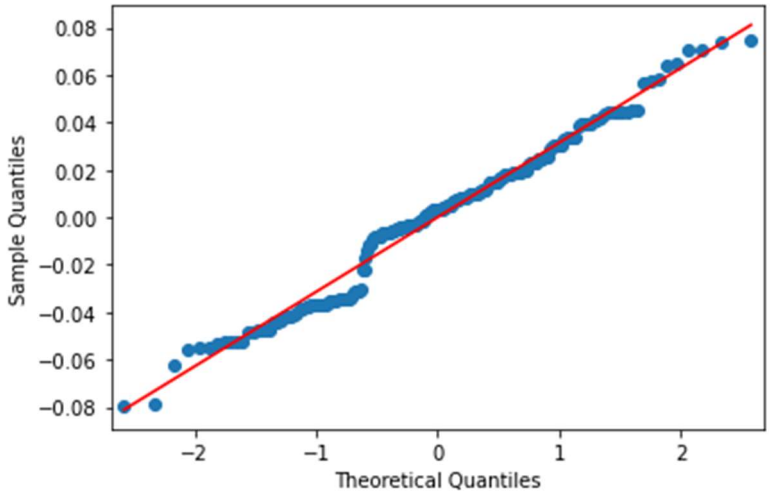

```
fig = plt.figure(figsize=(12,8)) # define the figure size
fig = sm.graphics.plot_regress_exog(model_result,
'NLTK_Sentiment_mean', fig=fig)
```

The final assumption ensures that the residuals are approximately normally distributed. Normality can be examined by observing the distribution of the residuals. The plot below is a histogram of the residuals overlaid with a normal curve which uses the mean and standard deviation of the residuals. The Python code below uses stats.norm.fit() function to retrieve the two aforementioned summary statistics of the best-fit normal distribution. The residuals, in addition to the normal curve, are now plotted as a kernel density plot. By observing the histogram and kernel density plot below, the residuals follow a normal distribution.



```
mu, std = stats.norm.fit(model_result.resid)
fig, ax = plt.subplots()
sns.histplot(x=model_result.resid, ax=ax, stat='density', linewidth=0,
kde=True)
ax.set(title='Distribution of Residuals', xlabel='residual')
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = stats.norm.pdf(x, mu, std)
sns.lineplot(x=x, y=p, color='orange', ax=ax)
plt.show()
```

Another way to test for normality is through a quantile-quantile (Q-Q) plot. The Q-Q plot displays the data in ascending order against quantiles of a normal distribution. The normality assumption is checked if the points on the Q-Q plot are on, or close, to the plot's line. The distribution on the Q-Q plot pictured below matches up with the line. Both ends of the plot do not tail off away from the line meaning that the distribution is not skewed.




```
sm.qqplot(model_result.resid, line='s')
```

Regression Model Evaluation

Now that all of the simple linear regression assumptions are checked, the next step in evaluating the simple linear regression model is to analyze the OLS regression results. All of the necessary statistics and code are shown in the model summary output below.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Percent_Change   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.005
Method:                 Least Squares   F-statistic:                   0.03469
Date:                Fri, 06 May 2022   Prob (F-statistic):              0.852
Time:                        13:47:44   Log-Likelihood:                 410.34
No. Observations:                 201   AIC:                            -816.7
Df Residuals:                     199   BIC:                            -810.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0096      0.003      3.705      0.000       0.004       0.015
NLTK_Sentiment_mean    -0.0014      0.007     -0.186      0.852      -0.016       0.013
==============================================================================
Omnibus:                        1.951   Durbin-Watson:                   1.207
Prob(Omnibus):                  0.377   Jarque-Bera (JB):                1.653
Skew:                          -0.080   Prob(JB):                        0.438
Kurtosis:                       2.586   Cond. No.                         3.38
==============================================================================
```

```
print(model_result.summary())
```

The intercept coefficient for the NLTK sentiment score is -0.0014. Each unit increase in the NLTK sentiment score is associated with a 0.0014 decrease in the percent change of a stock's percent change. The R-squared value for this model is less than 0.001, meaning that less than 0.1% of the observed variation in the stocks' percent change can be explained by the NLTK sentiment score. Since the R-squared value is so low, the NLTK sentiment score does not explain the changes in the stock percent change. The NLTK sentiment score predictor should be tested for statistical significance. Statistical significance is evaluated using the p-value, which is the probability that there is no relationship between the response and predictor(s). The p-value for the regression model is 0.852 which is above the significance level of 0.05. Therefore, the null hypothesis of statistical significance is not rejected. The NLTK sentiment score is not a significant predictor of a stock's percent change.

Conclusion

Hypothesis Confirmation

Out of the three hypotheses developed at the beginning of the report, we conclude that there is no relationship between the two variables in predicting the performance of a stock. Observing the

results from an analytical perspective, the model produced several values that indicated there was no relationship between the percent change and the sentiment score. The R-squared value, measuring the model's goodness of fit, told us that the sentiment score does not explain any changes in the percent change. Furthermore, the intercept coefficient shows us how small the change would be for every unit increase in the percent change. Even if the sentiment score could explain the changes in the percent change, the p-value is too high for the predictor to be statistically significant. Observing the results from a non-analytical lens, it makes sense that the two variables are not related. As stated earlier, Reddit attracts people from different backgrounds and various levels of stock market knowledge. This brings an influx of different opinions and viewpoints across many different stocks. Additionally, it is the nature of the people involved in the stock market to keep their findings and opinions to themselves because they have invested their own money into these stocks.

Pros and Cons of the Solution

The main advantage of our analysis is the ability to examine day-to-day variations in sentiment and performance. This allows for a high degree of precision in determining correlation and testing for causation. In case of market volatility over the course of the study, the change in sentiment is likely to be observed. This level of granularity is ideal for time-series analysis, especially if scaled up over a longer period of time. Additionally, as the APIs and methods we used allow the volume of trades and discussion to be gathered, we would be able to see if increases in discussion regarding a stock on reddit corresponds to higher levels of trade, or other analysis involving one or both components.

There are also moderate concerns regarding the replicability of our analysis. The API keys for Reddit and Alpha Vantage were both helpful to us in their ease of use at no charge. Furthermore, at the time of writing, Alpha Vantage allows for the collection of 20+ years of historical data at no charge. However, while Reddit's API is generous with historical data, it takes a long time to run, and will eventually hit a limit. This means that, at some point, anyone seeking to recreate our analysis would either have to pay a premium, or analyze over a different span of time.

Some drawbacks of this methodology are based in the difficulties inherent in processing comments on social media and other similarly unstructured data. Given the volume of comments required to actively monitor general sentiments the only realistic ways to evaluate these comments are those very similar to what we did. One of the specific issues that was mentioned in the data collection process is that the sentiment analysis is based on the entire comment. This is somewhat accounted for by increasing the volume of data collected to average out the sentiments for each ticker listed, but is not ideal. What is truly the much larger issue in terms of this project and of sentiment analysis as a whole is a lack of context. Essentially all of the comments collected are some sort of direct response to the post they replied to; no comment can stand alone. The text of these therefore lacks not only the context of the comment itself, but also the text that it was referencing when posted. This manifests itself in many ways in the data and

analysis results, but the most significant is that there is no way of knowing whether the commenter's sentiment is based on the long or short term. The Reddit user could think a stock will go up tomorrow, or ten years from now. Because of this, if you as a user trust in this sentiment, it should be taken at face value; this stock will be a good investment or a bad one.

Limitations and Areas for Improvement

One of the main limitations of this analysis was the short-run nature of our observations - primarily due to the limitations of Reddit's API in gathering histering historical data. If analysis were run over a longer period of time, be it through a premium API or dedicated a prolonged time period for observation, a larger volume of data could be collected, and it would be possible to examine long-run correlation between reddit sentiment and overall performance. Our analysis also did not incorporate karma into the results, meaning a comment widely endorsed by the Reddit community would hold the same weight as one only seen by two people.

Due to the nature of our data collection methods, many comments that would be meaningful if analyzed by a human were left out, as they would be difficult to interpret by a machine. For example, comments where the stock ticker in question was all lowercase (without a dollar sign), the name of the company was spelled out, or the ticker in question was implied (usually being a response to the original post) were excluded. We decided that collecting data for these comments would be impractical for the scope of our project, and that the data we would collect would be sufficient in volume and comparable in general sentiment, though having not collected the data, this cannot be guaranteed. On a similar note, posts not discussing any of the tickers we analyzed could still be useful for understanding the sentiment of the market as a whole, though we did not make said analysis, primarily due to the short time span covered.

One other limitation, however minor, is that the NYSE is closed on weekends and holidays, while the subreddits we studied are still active. This leads to a decision between discarding the data on days the stock exchange is closed or having abnormal data for Mondays and the occasional day following a holiday. We opted to exclude comments posted on the weekend, but this means a lot of data which could otherwise be useful was excluded from our analysis.

One area with potential for further exploration is to run a similar analysis for subreddits such as r/pennystocks. We opted against doing so for our research, due to concerns about getting a sufficient volume of data over the short period of time we would be running the project, and the stocks popular on this subreddit varying too much from those discussed on the most popular forums (which generally stick to larger companies). With that being said, the more volatile nature of companies trading at lower prices may be impacted more directly by activity on Reddit, and thus, this domain has potential for further analysis.

References

Brownlee, J. (2018, April 27). *How to Calculate Correlation Between Variables in Python*. Machine Learning Mastery. https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

Brydon, M. (n.d.). *Simple Linear Regression*. sfu.ca. https://www.sfu.ca/~mjbrydon/tutorials/BAinPy/09_regression.html#q-q-plot

Chen, K. (2015, September 7). *Use python to calculate the tone of financial articles*. Kai Chen. Retrieved May 6, 2022, from http://kaichen.work/?p=399

Lopatto, E. (2021, January 27). *HOW R/WALLSTREETBETS GAMED THE STOCK OF GAMESTOP*. The Verge. https://www.theverge.com/22251427/reddit-gamestop-stock-short-wallstreetbets-robinhood-wall-street

Moradian, M. (2020, August 17). *The History of Reddit*. Honor Society. https://www.honorsociety.org/articles/history-reddit

Ng, R (n.d.). *Evaluating a Linear Regression Model*. ritchieng.com. https://www.ritchieng.com/machine-learning-evaluate-linear-regression-model/