Using Sentiment Analysis to Measure the Influence of Stock-related Tweets in Forecasting Technology Sector Stock Prices

Felicity Bui

Columbia University
Department of Applied Analytics

18 April 2023

Introduction and Research Question

The stock market is one of the most widely studied subjects with numerous participants in research from top financial firms to academia aiming to understand not only how to make accurate predictions but to understand its behavior. Its movements are influenced by various internal and external factors such as political and legal events, economic cycles, and even social media trends. With the rise of social media platforms providing greater transparency, velocity, and exchange of information, it is now easier than ever for financial market participants to follow and analyze the market and individual stock. In this project, I applied sentiment analysis using a statistical machine learning model and random forest in an attempt to capture the correlation between the tweets extracted from Twitter and stock's price market movements. My exploration sought to answer the following research question:

How would daily stock prices behave in response to a positive, neutral, or negative sentiment scoring of tweets related to the respective stock?

Sentiment analysis is a technique used to determine the polarity of a text. It can be used to analyze the sentiment of any text data; here I looked at tweets related to technology companies. The sentiment analysis was performed on daily Twitter data for AAPL, AMZN, GOOG (Google's short-term class-C stock), MSFT, and TSLA. Extending the sentiment analysis, random forest and support vector regressions can be used upon the incorporation of stock price data to evaluate my hypothesis that, regardless of the sentiment, tweets referring to a particular stock can provide prediction capabilities on the respective stock price. The models are evaluated and compared using root mean squared error (RMSE). The results obtained from this study could provide valuable insights to financial market participants in making informed stock trading decisions based on shared text information beyond traditional finance media.

Data Sources

Tweets about the Top Companies from 2015 to 2020 via Kaggle
This dataset is utilized for extracting tweets mentioning Amazon, Apple, Google, Microsoft, and Tesla, by employing their relevant share tickers.

Yahoo Finance
The data of stock prices for Apple, Google, Amazon, Tesla, and Microsoft was obtained for the time period of 2015-2020.

Discussion of the Data

For the project, I used two main sets of data, one is the tweets data about Apple, Google, Amazon, Tesla, and Microsoft from 2015 to 2020 and the other is stock price data of those companies also during the period of 2015 to 2020. Instead of using all the data, I only used 1000-day-period data from April 2014 to 2020.

In the Twitter dataset Tweet.csv , I have the the following features:

- Tweet_id: Tweet's ID given by Twitter
- Writer: Account name of the tweet's author
- Post_date: post date in form seconds since epoch
- Body: text of tweet
- Comment_num: number of comments on tweet
- Retweet_num: number of retweet
- Like_num: number of like on tweet

In the Company's tweet dataset Company_Tweet.csv, I have the following features:

- Tweet_id: unique tweet's ID given by Twitter
- Ticker_symbol: company's stock ticker

In the Company's tweet dataset Company_Tweet.csv, I have the following features:

- Ticker_symbol: company's stock ticker
- Company_name: company's official name

All of the features listed above were extracted and merged to produce a dataset of tweets matching with each company. After compiling, Apple, Google, Amazon, Tesla, and Microsoft have [1,048,575]; [720,517]; [718,867]; [1,096,903]; and [375,938], respectively from April 2017 to December 2020.

In the Stock price dataset, I have the following features:

- Date: Date
- Close value: Stock price closing value
- Adjusted close value: Stock price adjusted closing value
- Open value: Stock price open value
- High value: Stock price high value
- Low value: Stock price low value
- Volume: Trading volume of stock

Similar to the Tweet's datasets, I used daily stock price data of 1000-day-period from April 2017 to December 2020. Out of the stock prices, I decided to use adjusted close price to run prediction on because adjusted close prices take into account any corporate actions, such as stock splits or dividends, which can significantly impact a stock's price. Ignoring these factors may lead to an incomplete and potentially misleading analysis of the stock's performance. Moreover, adjusted close prices are a widely accepted benchmark used by financial analysts and traders. According to a study by the CFA Institute, using adjusted prices can increase the accuracy of stock price predictions by as much as 7% compared to using only the regular closing prices. Furthermore, using adjusted close prices helps to ensure consistency in historical data analysis, as it allows for meaningful comparisons across time periods. This is especially important when evaluating long-term trends and making investment decisions based on past performance.

Data Preparation and Preprocessing and Reasoning

In terms of the analytical methods used in this assignment, I had a few important decisions to make and plan the whole process. To begin with, the first step involved pre-processing the tweets file for the company of my choice to clean up the data. This means removing all special meaningless special characters, hyper links, ASCII characters, stopwords, punctuations, replacing some particular symbols , removing ads and lowercasing all the text to maintain consistency. Finally, the text was not stemmed to retain the meaning of the words after removal of stopwords intact. Initially, I considered not removing the stop words as the individual tweets were often short in length; however, as I are concatenating all tweets for any given day together as one block of text, removing stopwords now would make sense. Therefore, as I conduct the sentiment analysis and subsequently use that sentiment data with the date for training in supervised learning models, I need to make sure the sentiment is as accurate as possible. For this, removing common occurring stopwords which may introduce noise in my results and increase the sentiment inaccuracy. Ultimately, key words and small text size were kept for quicker processing.

After cleaning the tweets, I grouped tweets together by day over the last few years available in the dataset; the purpose is to match the adjusted closing price for each date with the sentiment. Moreover, rather than finding the sentiment of each individual tweet, which is not very accurate or relevant to my greater objective, I find the overall sentiment of all the tweets made on a particular day of the year. The output of this is a file with the tweets grouped together by date and a subset from this, filtering for the tweets made after April 1st, 2017 to maintain consistency in the dataset and fix the time range for all the predictions.

In the next step now I need to summarize my tweets for each date. This is the core backbone of my sentiment analysis. Note that directly computing the sentiment of all the tweets is costly, with an exhaustive run time–around 10-15 minutes just for one day of tweets at a time. Summarizing the tweets avoids this problem by extracting the most important ones and performing sentiment analysis on those. Theoretically speaking this method is less accurate than running sentiment analysis on all the tweets as a lot of the information is excluded from the sentiment analysis and also there is a higher chance of misleading or false tweets influencing the results. Initially, the Lex Ranker Algorithm was explored, which works similar to Google's page rank algorithm to choose the most important tweets; however, the same slow runtime and lack of memory issue appeared. Moreover, this problem gets worse as I approach later dates as the number of tweets for any given day increases, making the file even larger and slower to process. This is expected as it makes more sense that more tweets were sent out about any given stock in the later months and years as the number of internet users increased in combination with increase in awareness of stocks. Accounting for such, I opted for another technique to summarize the tweets, in which I compiled the data into a tibble of sentences or lines and then tried to extract the 5 most relevant lines or sentences from this list. This method is significantly quicker and does not have not much of a negative impact on accuracy. This is because I extract concatenated tweets which are long so this way I manage to capture a lot of the information and manage to reduce the risk of sentiment inaccuracy due to false tweets. Following this logic I managed to reduce massive text blocks of tweets, some of which could be greater than a 1000 words, to only about 100 more relevant words quickly and efficiently. I also have dropped the old tweets body to reduce the file size.

Lastly, for the sentiment analysis, the vader library (Valence Aware Dictionary and sEntiment Reasoner) was used to perform the sentiment analysis on the tweets. This library is very accurate and is known to perform well on summarizing unlabelled text data, like my collection of tweets. My output generates sentiment in 4 categories–neutral, positive, negative and compound values. This way I can have a good understanding of the sentiment from different perspectives, and this would also help in the supervised learning analysis. With the summarized tweets, the file runs a lot quicker and I obtain a csv file at the end with the dates and the sentiments for those dates for the company. Note that I again removed the summarized tweets column to keep the file size small. Now this file would be used in the prediction model for training and testing. *More details on the Neural Network and Support Vector Machine available in Appendix F.*

Limitations and Opportunity for Further Exploration

Before deciding to pursue this subject of exploration, the limitations were how to accurately interpret the results of any study done with the relationship between tweets and stock price. Particularly looking at the relationship between tweets and stock behavior, the chicken or the egg problem emerges, whether the tweets cause stock behavior, or an external event indirectly fluctuated the frequency of tweets about a stock for which its behavior is immediately reflecting. For simplicity of my analysis, I must assume that the two events of a tweet being posted and a stock-related event are simultaneous occurrences, without assuming causation between the two. From there, a time inconsistency problem and scale problem in regards to the length of time that an outlier sentiment changing event influences a stock's price, as well as how much reach individual tweets have. This has large implications on the analysis of my data especially when considering that tweets are notoriously unreliable when posted from an account other than those traditionally trusted sources like news agencies, such as the Wall Street Journal, and company affiliates. For this reason, a later study can factor tweet reposting and like volumes into the analysis in addition to simple tweet frequency; additional isolation of tweets made by traditionally trusted sources could also be explored in comparison to unverified accounts. Another limitation is the limited computational power available; recall that I could not use vader to directly calculate the sentiment of all the tweets for any day but instead had to summarize the data first. Moreover, even the summarization technique used could not be very advanced such as a neural network or page rank algorithm again due to cpu limitations and slow processing speeds.

Discussion of Findings

Among my findings, specifically, as I explored the sentiment analysis conducted, there tends to be a more significant proportion of neutral tweets about stocks. Considering the tweet volume shared among my focus group of stocks, this could suggest that the nature of discussion surrounding a stock on Twitter is mainly informational as opposed to persuasive or reciprocal of any external events that may influence sentiment. While I extrapolate the aggregate of specific content in tweets posted during the observation period–as to whether they are factual–this information exchange provides the basis for influence on daily trading activities of financial market participants. *See Appendix B*. During the observation period, the average proportion of neutral tweets ranged between 79% and 86%; however, in general, tweets that do express sentiment tend to sway more positively for my focus group of stocks. *See Appendix C.* In understanding trading activity, the relationship between price and sentiment becomes clearer.

Since my prediction model only factors in the proportions of sentiments for testing my hypothesis, stock price predictions heavily reflect the trends presented in the sentiment analysis. Without any data manipulation, I found that the sentiment alone cannot predict stock prices perfectly. As evident in *Appendix D* and *Appendix E*, the predicted stock prices remain relatively stable compared to their actual adjusted close. Improving from this model, the aim would be to minimize the influence of the neutral proportion of tweets by measuring tweet frequency and either normalizing positive, neutral, and negative tweet sentiments; or removing the neutral sentiment altogether. From here, even with the current model I developed for this study, forecasting could be incorporated to anticipate stock prices in the future, outside of the observation period.

Models Performance

Random Forest

One of the models that I implemented to predict the stock price was Random Forest, one of the popular machine learning algorithms for predicting stock price. For this project, I used the Ranger algorithm, an optimized version of Random Forest, due to its faster computational speed and flexibility in split criterion on continuous variables. Overall, the Ranger model showed competitive accuracy levels and was evaluated using RMSE. This choice of algorithm allowed us to achieve promising results in predicting stock prices, demonstrating the effectiveness of the Ranger model in such tasks. The model's performance on Apple, Google, Amazon, Tesla, and Microsoft are as follows below.

Figure 1. Ranger RMSE Values.

Ticker	**Random Ranger RSME**	**Tuned Ranger RMSE**
AAPL	8.2358	8.0083
GOOG	6.022543	5.71428
AMZN	17.80423	17.39054
TSLA	3.205662	3.008078
MSFT	25.89311	24.549

Out of all the stocks, the model performed the best on TSLA while performing the worst on MSFT in both the random Ranger model and the tuned Ranger model. The tuned Ranger model achieved 3.008078 on TSLA while only 24.549 on MSFT. After inspecting the dataset, I concluded that the model performance difference was due to two main factors. Firstly, the number of tweets available for each company is different. Even though I used the same time interval, from April 2017 to December 2020, the number of tweets for companies is different. For example, TSLA has over 1 million tweets, while MSFT has less than 400 thousand. Secondly, in addition to the frequency of tweets, TSLA’s superior performance relative to other stocks can be attributed to the company's unique nature. One of the distinguishing factors of Tesla is the prolific tweeting activity of its CEO, Elon Musk. This behavior has a disproportionate impact on the stock performance of the company due to the influence and reputation of Musk in the financial markets. Elon Musk's tweets are closely followed by investors and are often seen as an important signal of the company's performance and prospects. Therefore, the market perceives his tweets as significant signals that can drive investor sentiment and affect the stock price.

Support Vector Machine

The RMSE is a commonly used statistical measure of the difference between predicted and actual values of a target variable. In the context of stock market prediction, the RMSE represents the accuracy of a predictive model in forecasting the future prices of a stock. A lower RMSE value indicates that the model is more accurate in its predictions, while a higher RMSE value indicates a lower accuracy. In the analysis using SVM to predict stock prices, the data was split into a 80-20 ratio for training and testing the model, and the time series data was also split accordingly to avoid data leakage.

In addition to the impact of social media trends on stock prices, the RMSE values presented in the table can also be affected by various other factors. For example, the data used for training the models may not accurately represent the future behavior of the stock prices, leading to inaccurate predictions. Additionally, the choice of features used in the model can also affect its accuracy, as certain features may be more relevant to predicting stock prices than others.

Figure 2. Support Vector Machine RMSE Values.

Ticker	**SVM RMSE**
AAPL	23.9932
GOOG	27.06885
AMZN	76.9797
TSLA	7.332471
MSFT	59.6568

Neural Network Model

Neural networks are trained to learn patterns and relationships between tweets and stock prices, potentially providing more accurate predictions. However, neural networks are also computationally intensive and require considerable training data. Overall, neural networks have shown promising results in sentiment analysis and stock price prediction.

TSLA has the lowest RMSE value of 3.016923, indicating that the neural network model used for TSLA is the most accurate among the five companies. GOOG follows closely with an RMSE value of 5.501641, suggesting that the neural network model used for GOOD is also highly accurate. The RMSE values for AAPL, AMZN, and MSFT range between 8.111595 and 23.34197, suggesting that the neural network models used for these companies are less accurate than TSLA and GOOG, but still provide useful predictions.

Overall, the RMSE values can provide valuable insights for investors in evaluating the performance of neural network models for stock price prediction. It is important to note that other factors, such as the availability and quality of data, market conditions, and economic indicators, can also impact the accuracy of predictive models. Therefore, investors should use the RMSE values along with other information to make informed investment decisions.

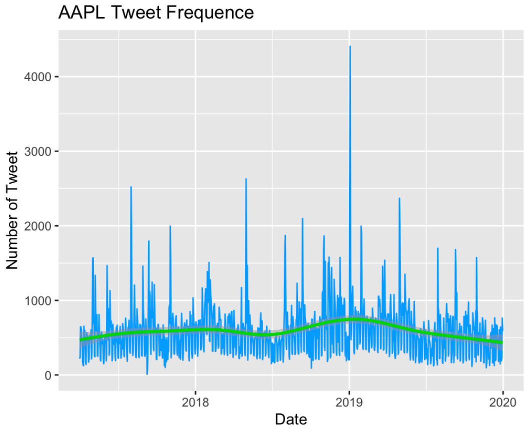
Figure 3. Neural Network RMSE Values.

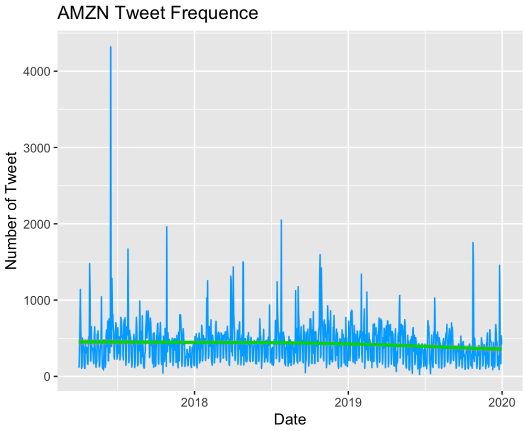
Ticker	**Neural Network RSME**
AAPL	8.111595
GOOG	5.501641
AMZN	16.62926
TSLA	3.016923
MSFT	23.34197

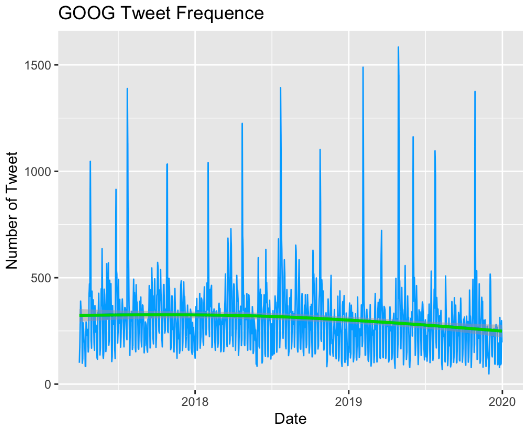
Conclusion

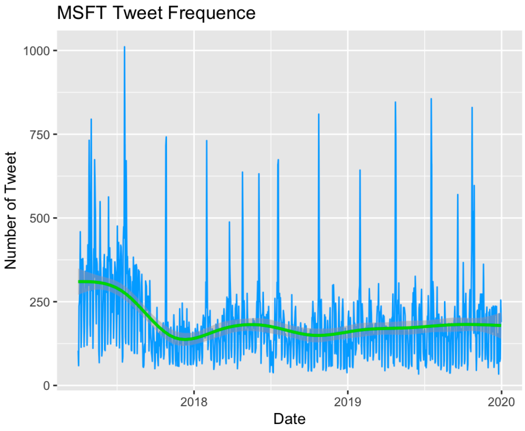
The report discussed the performance of three different machine learning algorithms in predicting stock prices: Random Forest, Support Vector Machine (SVM), and Neural Network Model. The Range Algorithm was used for random forest due to its faster computational speed and flexibility in split criterion on continuous variables. The Ranger model was evaluated using RMSE, and the results showed competitive accuracy levels. The highest-performing stock for Random Forest was TSLA, while the lowest-performing stock was MSFT. For SVM, the RMSE values for five stocks (AAPL, GOOG, AMZN, TSLA, and MSFT) were provided, with the lowest RMSE for TSLA and the highest for AMZN. Finally, TSLA had the lowest RMSE value for the Neural Network Model, indicating the model's high accuracy, while AAPL, AMZN, and MSFT had higher RMSE values. While RMSE values can provide valuable insights for investors, other factors can also impact the accuracy of predictive models. This study provides a solid foundation for further exploration of Twitter's potential as a stock prediction tool.

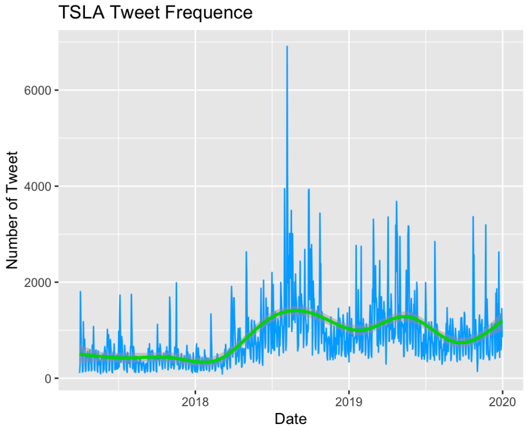
Appendix A. Tweet Frequency of Focus Group of Stocks.



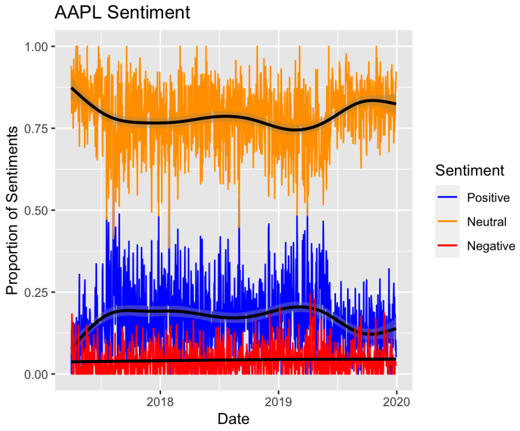


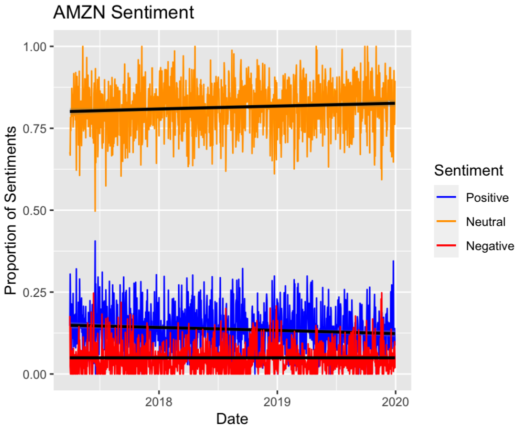


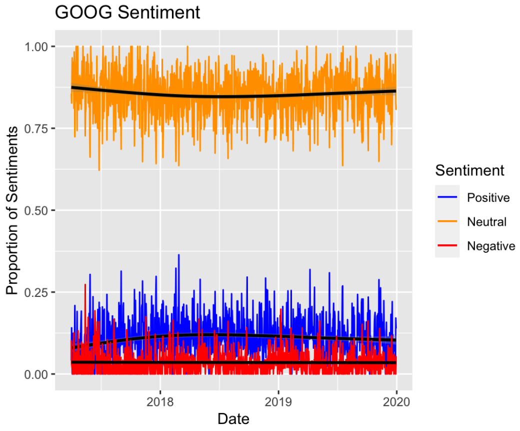


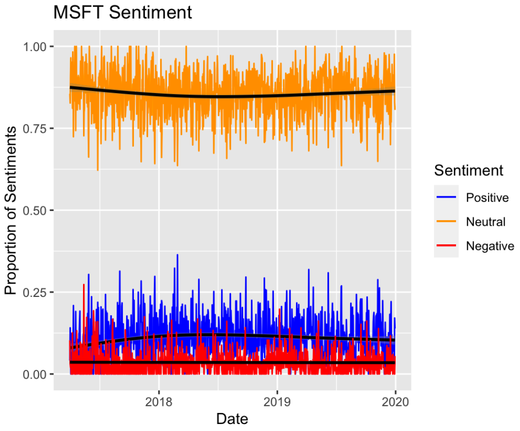


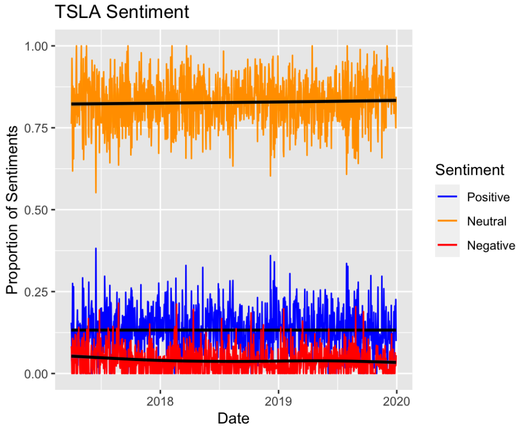
Appendix B. Sentiment Proportions Over Time.



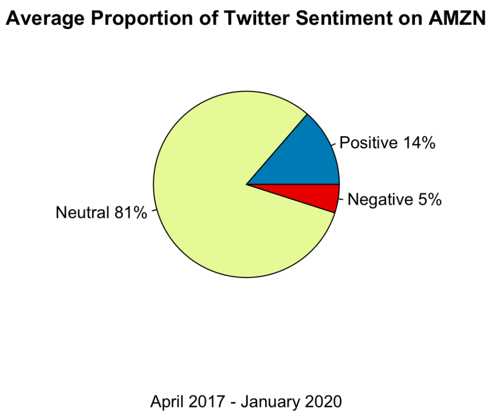




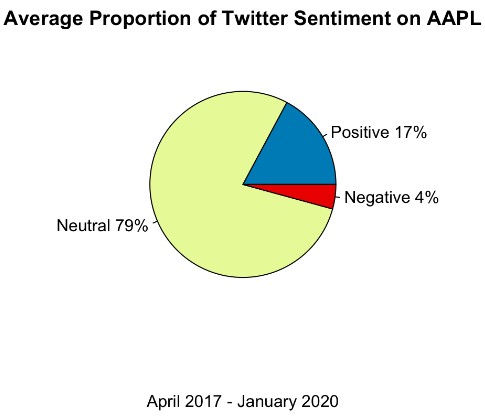




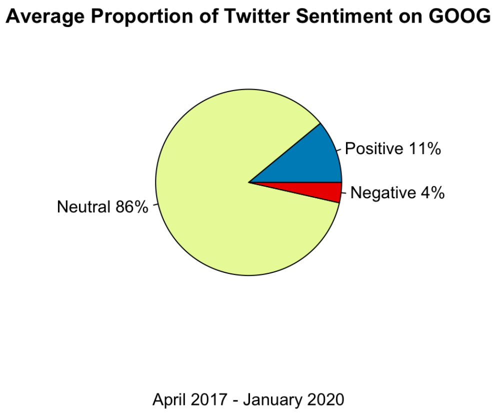
Appendix C. Average Proportions of Sentiment Across the Observation Period.


Average Proportion of Twitter Sentiment on AMZN

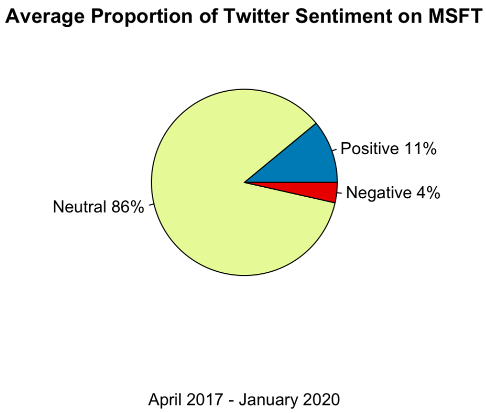
April 2017 - January 2020


Average Proportion of Twitter Sentiment on AAPL

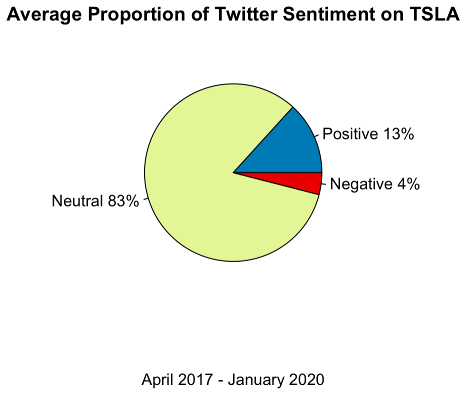
April 2017 - January 2020


Average Proportion of Twitter Sentiment on GOOG

April 2017 - January 2020

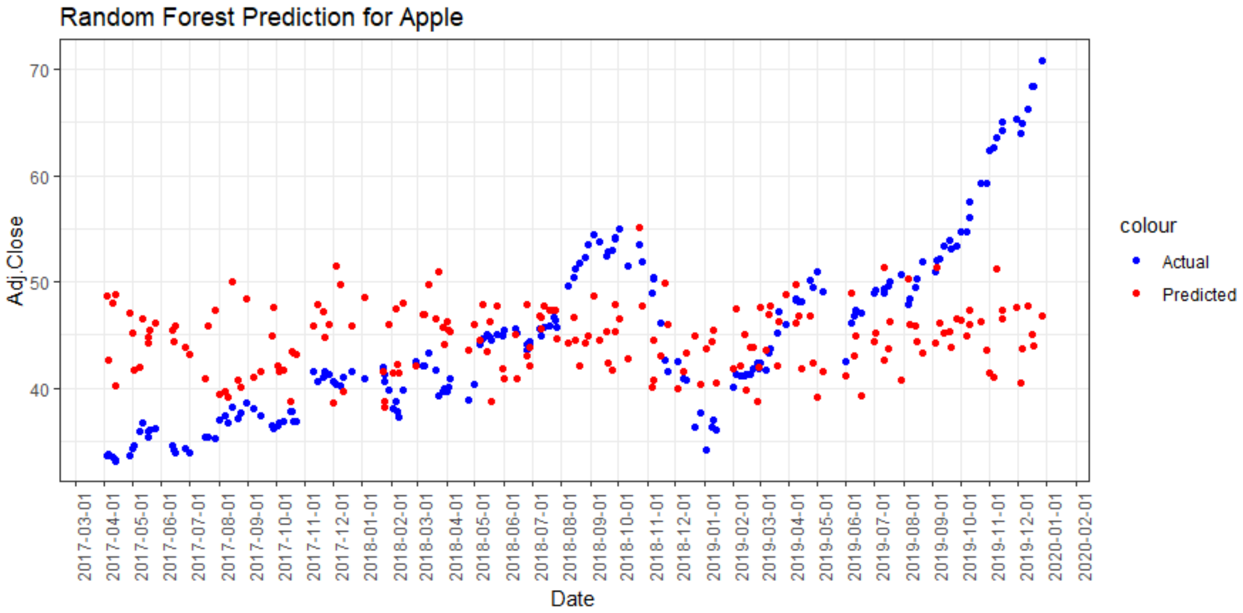

Average Proportion of Twitter Sentiment on MSFT

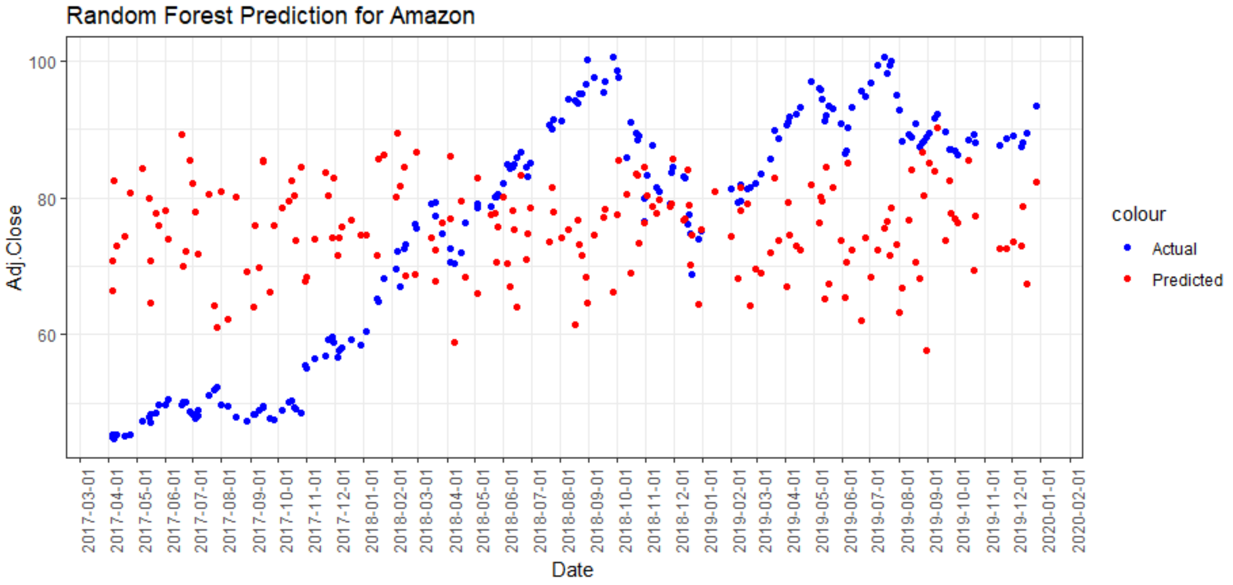
April 2017 - January 2020

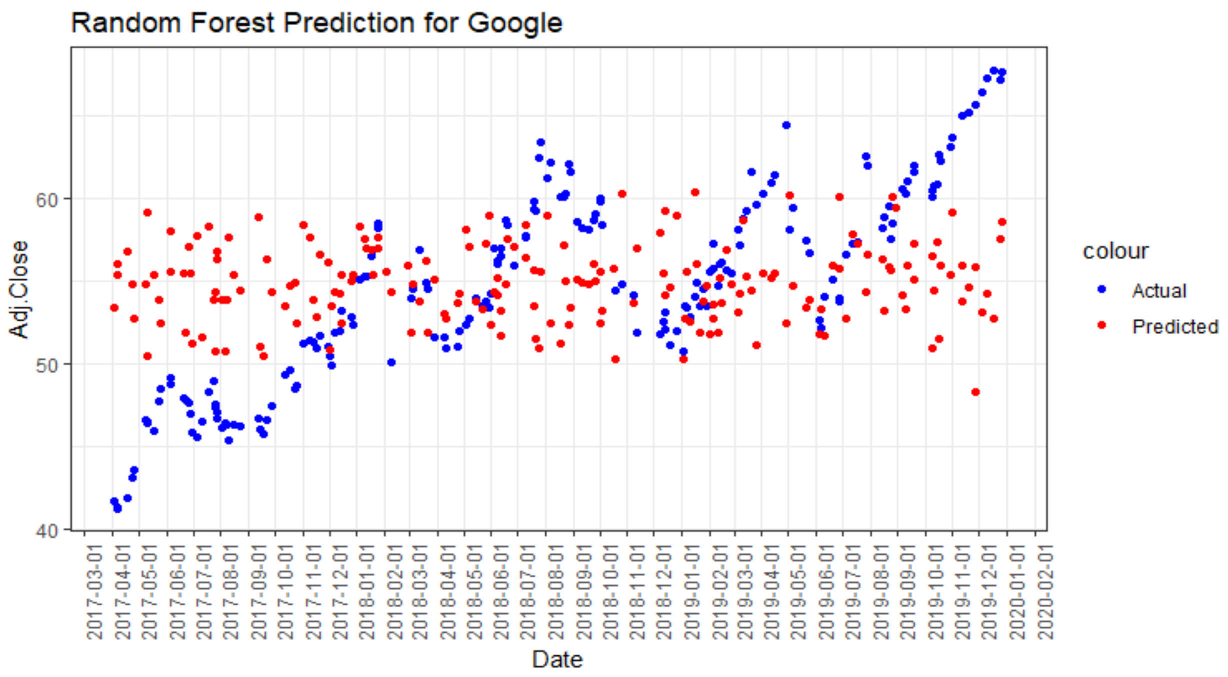

Average Proportion of Twitter Sentiment on TSLA

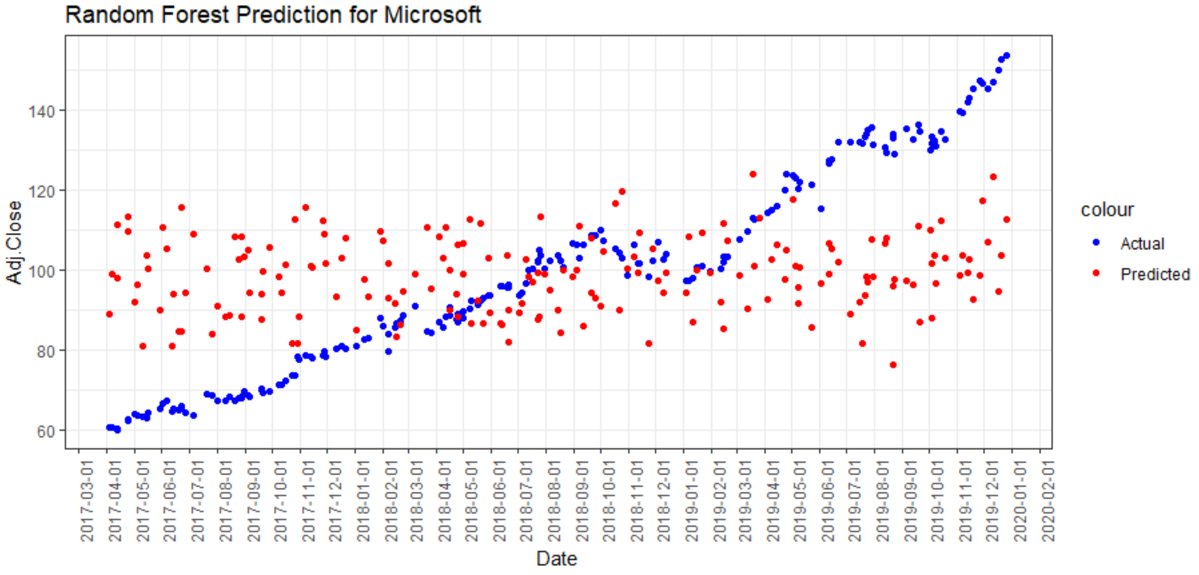
April 2017 - January 2020

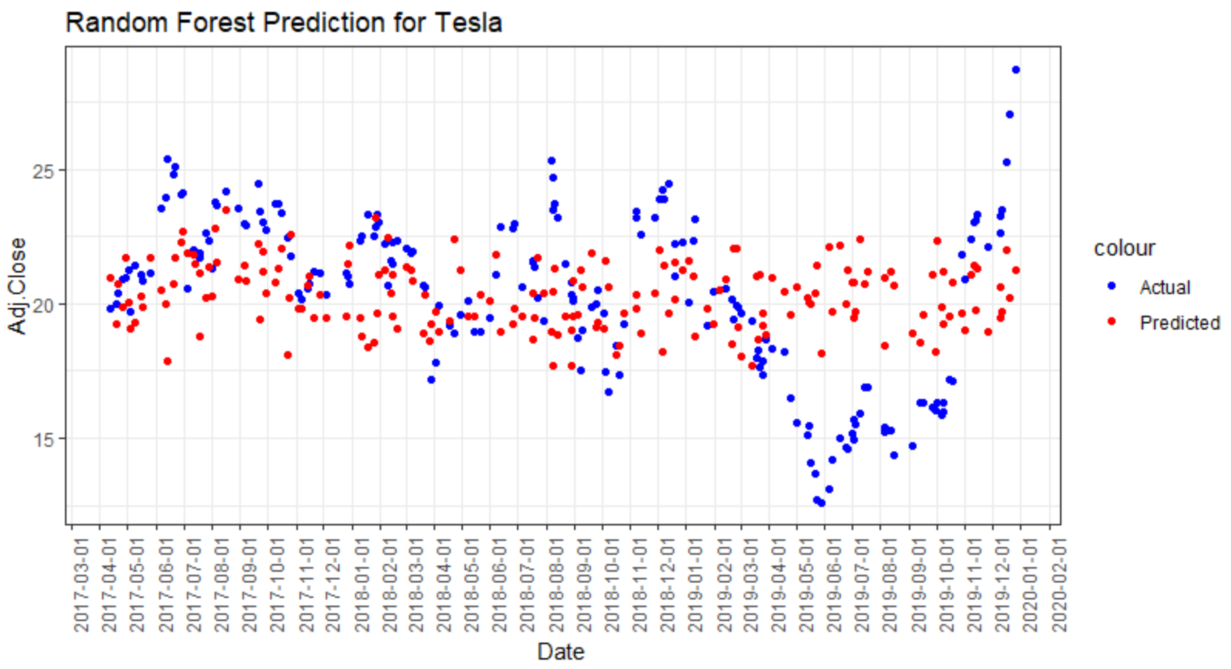
Appendix D. Comparison of Prediction and Actual Price. Ranger Random Forest.



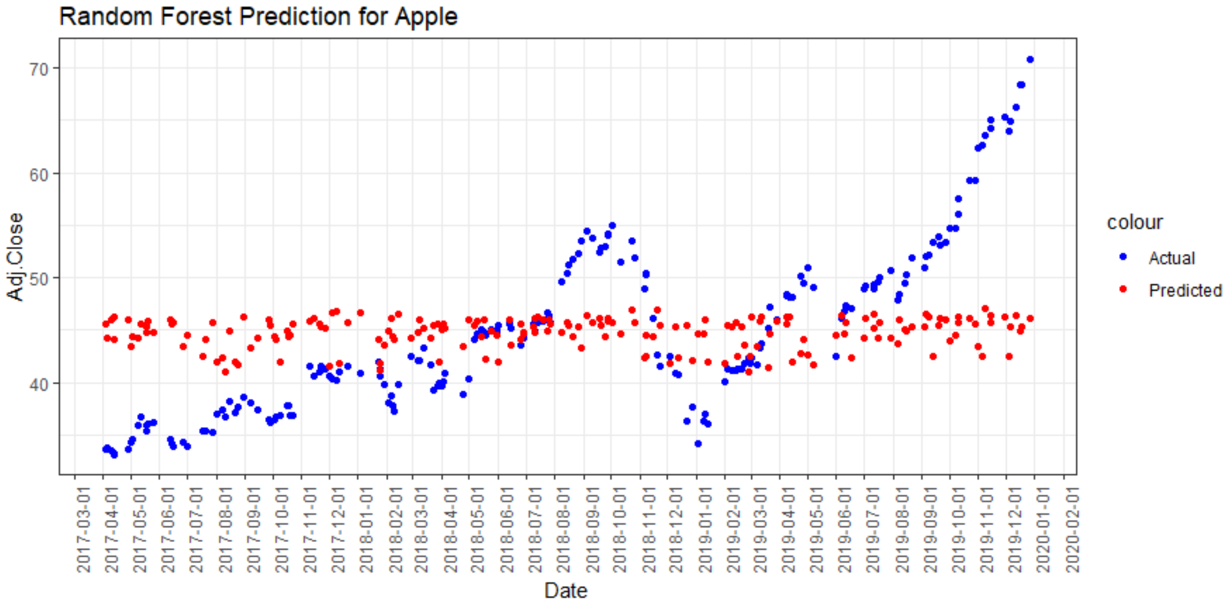


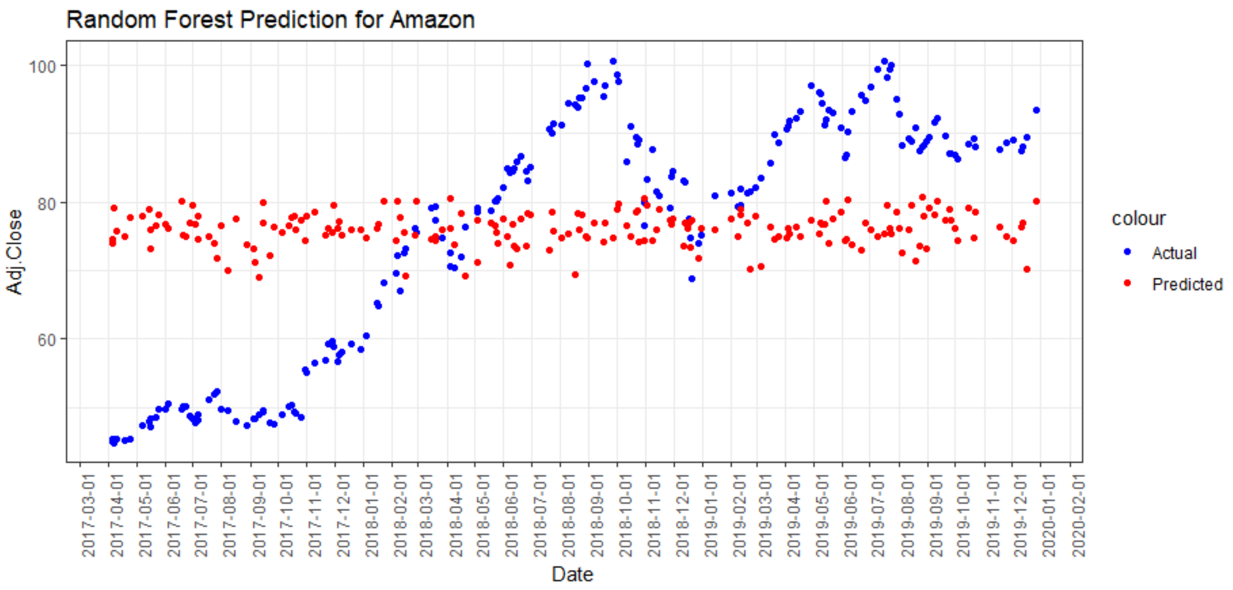

Random Forest Prediction for Google
Adj.Close
40
50
60
Date
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01
2018-01-01
2018-02-01
2018-03-01
2018-04-01
2018-05-01
2018-06-01
2018-07-01
2018-08-01
2018-09-01
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
2019-06-01
2019-07-01
2019-08-01
2019-09-01
2019-10-01
2019-11-01
2019-12-01
2020-01-01
2020-02-01
colour
Actual
Predicted

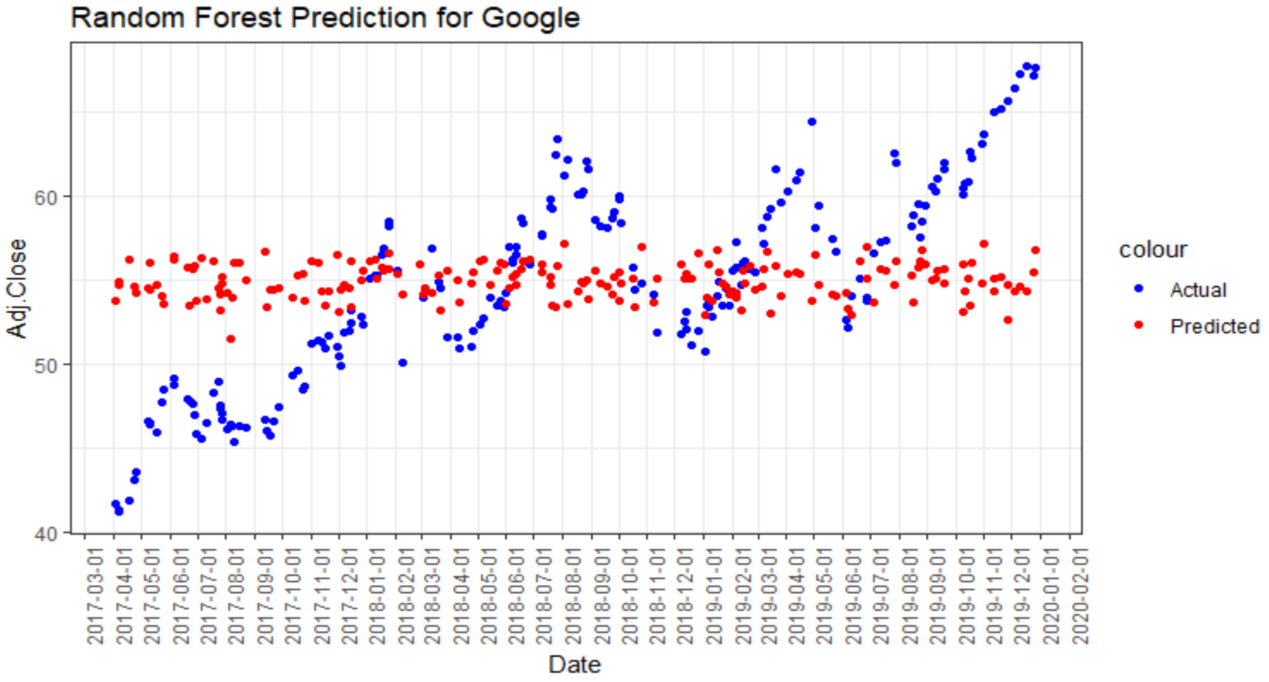

Random Forest Prediction for Microsoft
Adj.Close
60
80
100
120
140
Date
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01
2018-01-01
2018-02-01
2018-03-01
2018-04-01
2018-05-01
2018-06-01
2018-07-01
2018-08-01
2018-09-01
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
2019-06-01
2019-07-01
2019-08-01
2019-09-01
2019-10-01
2019-11-01
2019-12-01
2020-01-01
2020-02-01
colour
Actual
Predicted

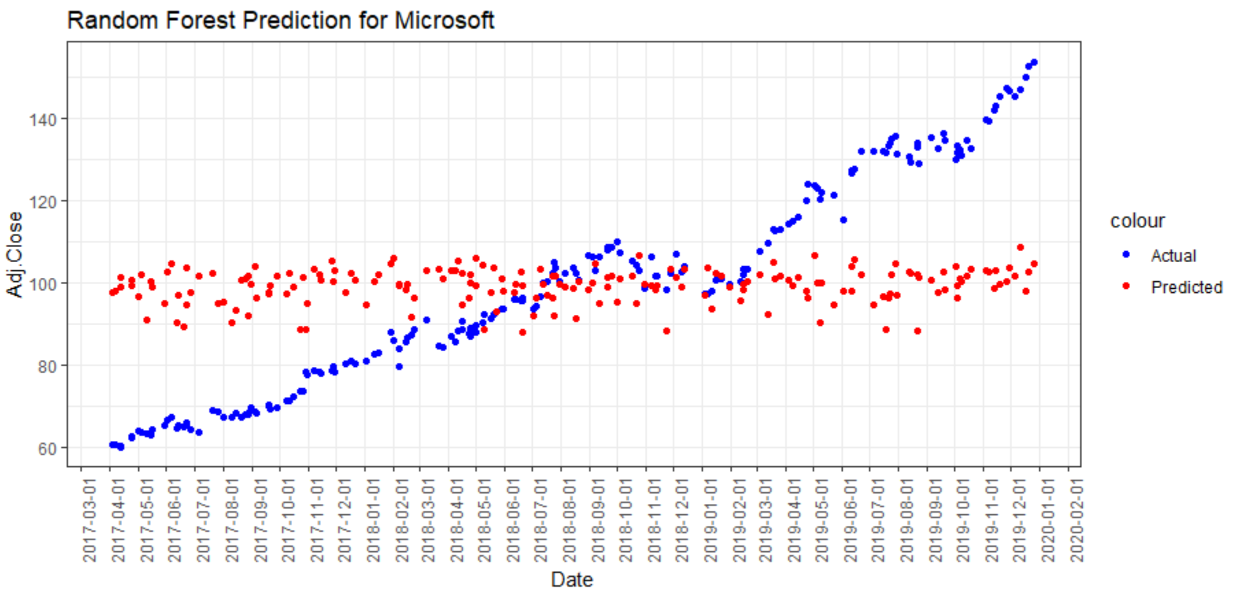


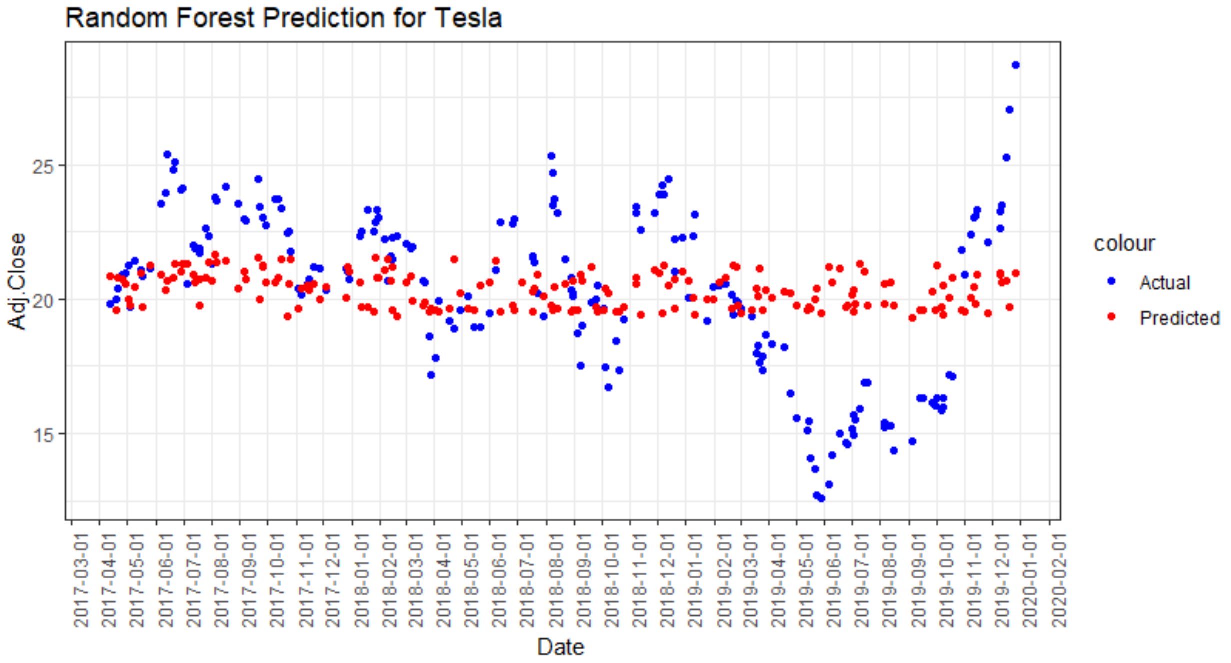
Appendix E. **Comparison of Prediction and Actual Price. Tuned Ranger Random Forest.**




Random Forest Prediction for Amazon
Adj.Close
100
80
60
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01
2018-01-01
2018-02-01
2018-03-01
2018-04-01
2018-05-01
2018-06-01
2018-07-01
2018-08-01
2018-09-01
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
2019-06-01
2019-07-01
2019-08-01
2019-09-01
2019-10-01
2019-11-01
2019-12-01
2020-01-01
2020-02-01
Date
colour
● Actual
● Predicted


Random Forest Prediction for Google
Adj.Close
60
50
40
2017-03-01
2017-04-01
2017-05-01
2017-06-01
2017-07-01
2017-08-01
2017-09-01
2017-10-01
2017-11-01
2017-12-01
2018-01-01
2018-02-01
2018-03-01
2018-04-01
2018-05-01
2018-06-01
2018-07-01
2018-08-01
2018-09-01
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
2019-06-01
2019-07-01
2019-08-01
2019-09-01
2019-10-01
2019-11-01
2019-12-01
2020-01-01
2020-02-01
Date
colour
● Actual
● Predicted





Appendix F. Neural Network Deep Learning and SVM.

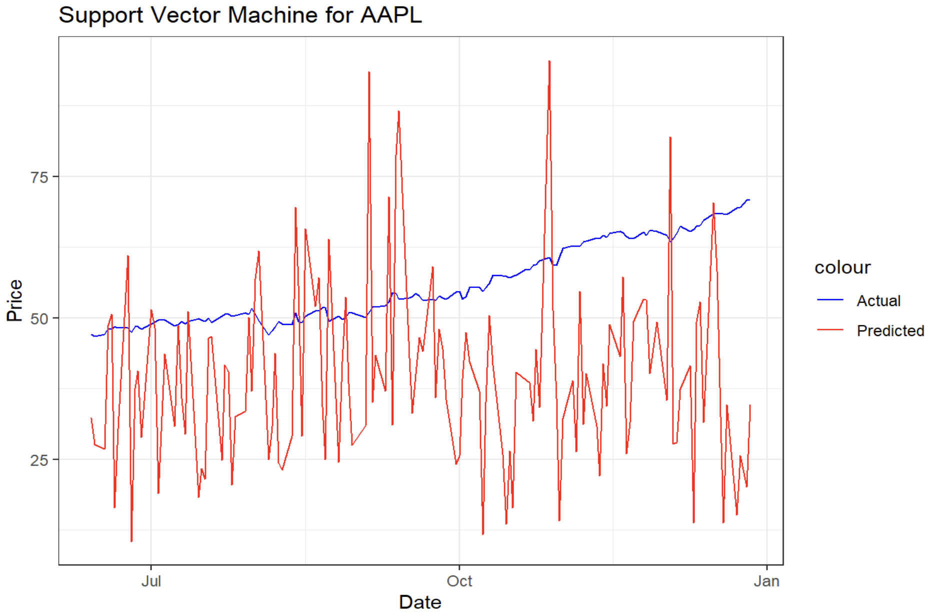
In addition to the random forest and support vector regression models, I also implemented a deep learning model using H2O.ai's deep learning library to further improve my prediction accuracy. The deep learning model was trained using a neural network architecture, which is a type of machine learning algorithm that is loosely inspired by the structure and function of the human brain.

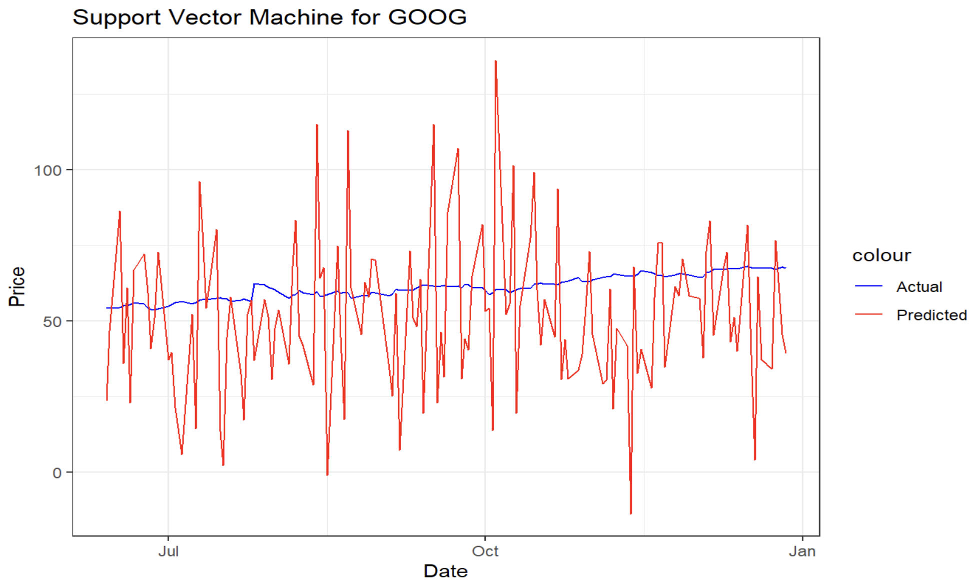
My neural network model was tuned using grid search, which is a hyperparameter optimization technique that involves searching over a range of hyperparameters to find the optimal combination that yields the best performance. I specified various hyperparameters such as activation functions, number of hidden layers and neurons, L1 and L2 regularization values to prevent overfitting, and other model-related parameters.

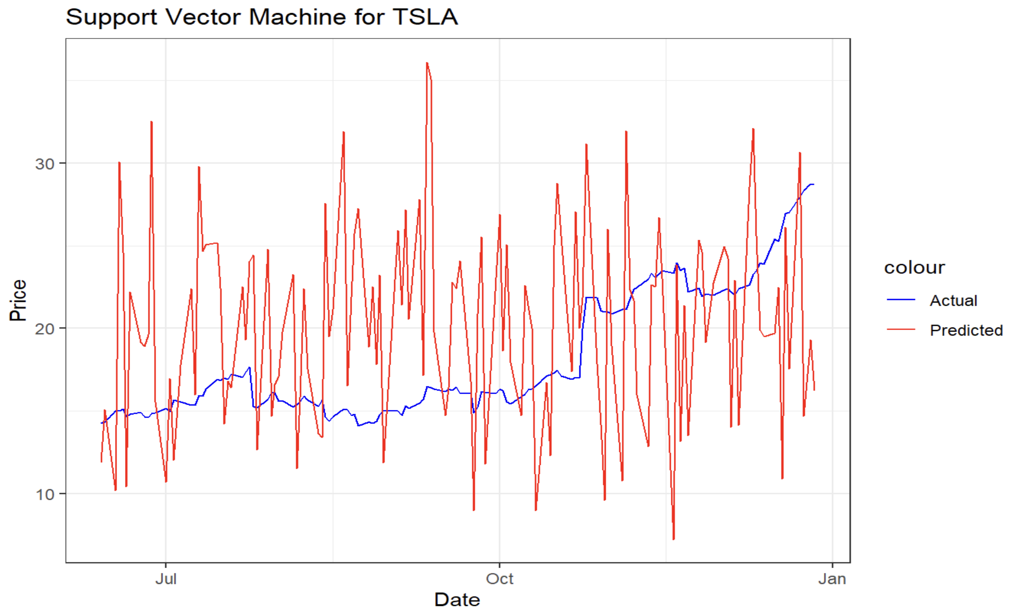
The best neural network model was selected based on its RMSE value, which is a commonly used metric for evaluating regression models. The deep learning model achieved the best performance among all models tested, demonstrating the potential of neural networks in predicting stock market prices using sentiment analysis data. Overall, my findings suggest that deep learning models can provide a more accurate and efficient way to predict stock market prices, particularly when analyzing large amounts of complex data such as social media sentiment.

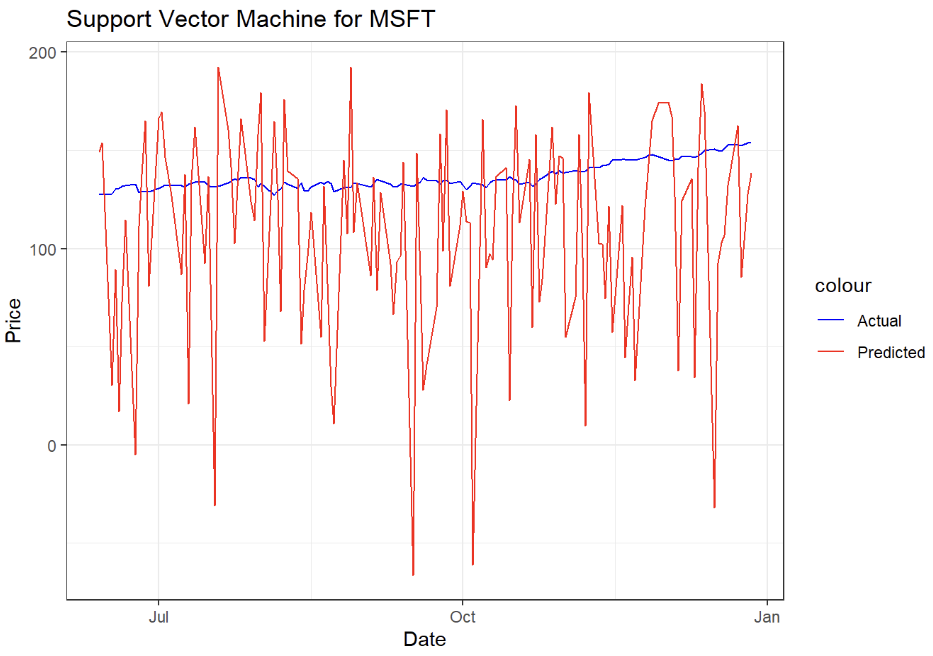
Appendix G. Comparison of Prediction and Actual Price. Support Vector Machine.

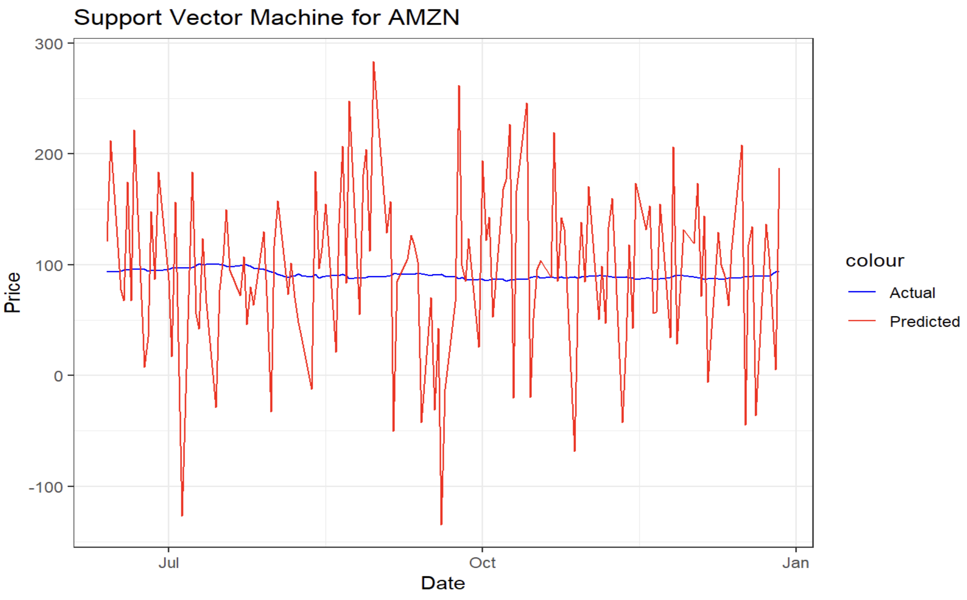
The graphs below are produced from values of the test-train split applying the model of the training dataset to the test model, hence the time frame covers 2019.




Support Vector Machine for GOOG
Price
100
50
0
Jul
Oct
Jan
Date
colour
Actual
Predicted


Support Vector Machine for TSLA
Price
30
20
10
Jul
Oct
Jan
Date
colour
Actual
Predicted


Support Vector Machine for MSFT
200
100
0
Price
Jul
Oct
Jan
Date
colour
Actual
Predicted


Support Vector Machine for AMZN
300
200
100
0
-100
Price
Jul
Oct
Jan
Date
colour
Actual
Predicted