Predicting Stock Market Volatility using Sentiment Analysis of Twitter Data

Odemuno Ogelohwohor
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
oogelohw@cs.cmu.edu

Aishwarya Agrawal
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
aishwara@cs.cmu.edu

Diwen Deng
Tepper School of Business
Carnegie Mellon University
New York, NY, 10004
diwend@andrew.cmu.edu

Moulya Sudhir
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
msudhir@cs.cmu.edu

Abstract

Predicting the movement of stock markets, as well as stock market volatility, has always been a challenging task that requires a deep understanding of the factors that affect market behavior. In recent years, social media has emerged as a valuable alternative source of information for predicting market movements. This proposal aims to explore the effectiveness of using sentiment analysis techniques on tweets data to predict stock market volatility by investigating the relationship between the sentiment score analyzed from Twitter and volatility indices.

1 Introduction

Many investors and traders are interested in stock trends and volatility as they can adjust their trading and pricing strategies to make profits by predicting stock markets' volatility. While there has been extensive research analyzing and predicting stock returns using historical stock data and alternative data, many studies have neglected the importance of analyzing social media data. The recent GameStop discussion on Reddit highlighted the significant impact of social media on stock market behavior, leading to a surge in GameStop's stock prices and causing major losses for short sellers, creating a global frenzy around the stock market [1]. As such, this study aims to explore whether social media platforms, particularly Twitter, can serve as valuable sources of information for predicting stock market volatility.

The hypotheses to be tested include whether sentiment analysis can identify stock market events quicker and more accurately than traditional models, the correlation between the volatility of large-cap and small-cap stocks like AAPL and ACMR respectively, the influence of one stock's movement on other stocks, and the relevance of specific keywords or tickers in predicting stock market movement. The study aims to improve upon previous research exploring the use of alternative keywords and tickers beyond those previously identified and applying sentiment analysis on these but not limited to those data.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

2 Literature Review

Long Short-Term Memory (LSTM), a type of recurrent neural network is a popular choice for modeling financial time series data and predicting stock prices. Ho et al. [2] proposed an improved LSTM model to predict stock prices using Twitter data. Their model outperformed the traditional LSTM model and other existing methods. Ali Derakhshan [3] compared different feature selection methods to predict stock market movements using sentiment analysis and news articles. They found that a combination of sentiment analysis and news articles improved the prediction accuracy of stock market movements. R. Ray Chen [4] proposed a sentiment analysis approach to predict stock market movement using Twitter feeds. They found that sentiment analysis of Twitter feeds could predict stock market movement with an accuracy of 87.6%. However, all these articles do not investigate the correlation between sentiment analysis scores and stock market volatility indexes.

Similarly, Alexander Porshnev [5] used a combination of sentiment analysis and machine learning algorithms to predict stock market volatility. The study found that incorporating sentiment analysis improved the accuracy of the model. Jasmina Smailović [6] use Twitter sentiment analysis to predict stock market trends. The study showed that Twitter sentiment analysis is a useful tool for predicting stock market trends.

This study did not specifically examine stock market volatility. David Valle-Cruz et al. [7] explored the use of deep learning algorithms for predicting stock market trends based on sentiment analysis of tweets. The authors found that deep learning algorithms can improve the accuracy of stock market predictions. Salah Bouktif [8] used word embeddings and neural networks to predict stock market trends from news articles. The study found that their approach outperformed traditional machine learning methods. Arezoo Hatefi Ghahfarrokhi [9] investigated the correlation between social media sentiment and stock market trends in China. The study found that social media sentiment is a leading indicator for stock market trends. However, this study focuses on the Chinese market, while our work aims to focus on the global market.

3 Dataset

3.1 Sentiment140

Our baseline implemented sentiment analusis dataset is available on Kaggle [10] and it contains 1.6 million tweets extracted with 6 data columns: (1) the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive), (2) the id of the tweet, (3) the date of the tweet, (4) the query, if there is no query, then this value is NO_QUERY, (5) the user that tweeted, (6) the text of the tweet.

To create the cleaned dataset, we employed the same data cleaning methods done by Go et al. [11] were: lowercasing, replacing URL, replacing emojis, replacing usernames, removing non-alphanumeric, removing consecutive letters, removing short words (words with a length of less than two), removing stopwords, removing duplicated tweets, lemmatizing. The download size of the csv file is 77.59MB.

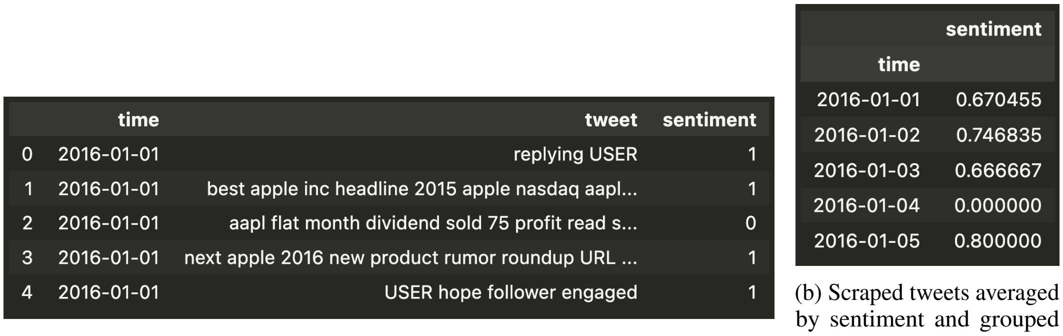
Table 1: Tweet Cleaning Process

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

3.2 GetOldTweets / Scweet

After training the best model on train data, the next step is to obtain old tweets from January 01, 2016 to December 31, 2016. The baseline paper made use of the GetOldTweets python library which was deprecated due to an update on Twitter guidelines. Hence, Scweet, an alernative Twitter scraping tool was made use of to extract the tweets. 1846 Twitter entries were extracted beginning from January

01, 2016 containing the keyword"AAPL". The columns that were extracted were "Embedded Text" and "Timestamp". The scraped tweets were then pre-processed the same way as the training dataset and the sentiment for each tweet were predicted. Figure 1a shows the structure of the scraped tweets data frame along with predicted sentiment. Figure 1b shows the average sentiment for the first few days of the scraped tweets.



	time	tweet	sentiment
0	2016-01-01	replying USER	1
1	2016-01-01	best apple inc headline 2015 apple nasdaq aapl...	1
2	2016-01-01	aapl flat month dividend sold 75 profit read s...	0
3	2016-01-01	next apple 2016 new product rumor roundup URL ...	1
4	2016-01-01	USER hope follower engaged	1

(a) Pre-processed Scraped tweets summary

	sentiment
time	
2016-01-01	0.670455
2016-01-02	0.746835
2016-01-03	0.666667
2016-01-04	0.000000
2016-01-05	0.800000

(b) Scraped tweets averaged by sentiment and grouped by day

Figure 1: Old tweets obtained from scraping Twitter data in 2016

3.3 Yahoo Finance

The baseline paper extracted stock historical price data from Yahoo Finance for the same time period between January to December 2016, in CSV file format with seven features, including Date and Close price. Only these two features were used in this study. The stock close price data for DJIA and Apple Inc. were collected only on days when the stock market was open.

4 Model Description

4.1 Sentiment

After pre-processing the Sentiment140 dataset, TD-IDF was used to vectorize the dataframe. The dataframe was then split into train and test dataset, where test data was 5% of the entire dataset. 3 models were trained on this dataset, namely : Support Vector Machine (SVM), Logistic Regression (LR), and Decision Tree Classifier (DT).

The LR model was trained on the Sentiment140 dataset consisting of 1599538 rows and two columns: sentiment and tweet. The sentiment column contains binary labels, where 0 represents negative sentiment and 1 represents positive sentiment. Our goal was to predict the sentiment of tweets using their textual content.

The LR model was implemented using Scikit-Learn's LogisticRegression class with the following parameters: C = 0.1 and max_iter = 1000. The regularization parameter C controls the strength of the regularization, with smaller values resulting in stronger regularization. The maximum number of iterations, max_iter, defines the maximum number of iterations for the solver to converge.

Given the training data, the LR model was fit to learn the optimal coefficients for each feature. The LR model was then evaluated using a hold-out test set.

Mathematically, the LR model can be defined as follows:

Given a tweet x, the LR model computes the probability that the sentiment is positive as:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

where w is the weight vector and b is the bias term.

The LR model predicts the sentiment of the tweet as follows:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

The LR model is trained by minimizing the following loss function:

$$\text{minimize}_{w,b} \; \frac{1}{n} \sum_{i=1}^{n} -y_i \log(P(y_i = 1|x_i)) - (1 - y_i) \log(1 - P(y_i = 1|x_i)) + \frac{1}{2C} ||w||_2^2$$

where n is the number of training samples, x_i is the i-th tweet, y_i is its corresponding sentiment label, and $||w||_2^2$ is the L2 regularization term.

4.2 Stock Price Movement

After getting the average sentiment value of tweets of the present day, this paper tends to predict how much the stock market will rise or fall the next day. This paper extracts historical AAPL and DJIA data from January to December 2016, taking the difference in 'Close' price as the value to the predictor, and labeling it as "PriceDiff". Boosted Regressor Tree and MLP models are then trained on data from January to August 2016 and tested on stock-related data from September to December 2016.

4.2.1 Boosted Tree Regressor

The Gradient Boosted Model was trained on the data from January to August 2016 consisting 170 rows and eight columns. The average sentiment score is the numeric number that represents the average sentiment score of the tweets on that day. Higher average sentiment score represents higher market sentiment. Our goal was to predict the next day's price difference using average sentiment scores and lag price difference.

The Gradient Boosted Model Model was implemented using Scikit-Learn's Ensemble class with the default parameters (loss:"squared error", learning rate: 0.1, n_estimator: 100). Given the training data, the Boosted Tree Regressor model was fit to optimize the follwing loss function (squared error between the true value and the predicted value):

$$\text{minimize}_{\hat{f}(x)} \; \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x))^2$$

In each stage a regression tree is fit on the negative gradient of the given loss function. The Booeted Tree Regressor was then evaluated using a test set from September to December 2016.

The Boosting for Regression Trees algorithm can be defined as follows:

Algorithm 1 Boosting for Regression Trees

```
Set $f_0(x) = argmin_\delta \sum_i^n L(y_i, \delta)$
for $b = 1, \ldots B$ do
    For $i = 1, \ldots n$ compute:
        $r_{ib} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{b-1}}$
    Fit a tree $\delta_b$ to the response $r_{ib}$
    Update $f_b(x) = f_{b-1}(x) + \eta\delta_b$
end for
Output the boosted model:
    $\hat{f}(x) = f_B(x)$
```

4.2.2 Multilayer Perceptron Neural Network

The Multilayer Perceptron Neural Network (MLP) model was also employed on the same train and test data as Booted Tree Regressor Model. The MLP moodel was implemented using Scikit-Learn's Neural Network class with the following parameters: hidden_layer_sizes = 5, and max_iter = 2000. The hidden layer size represents the number of neurons in the hidden layer. The maximum number of iterations, max_iter, defines the maximum number of iterations for the solver to converge.

Given a set of features and a target, it can learn a non-linear function approximator for regression. Given the training data, the MLP model was fit to learn the best weight and bias for each feature to minimize the loss function (squared error) using stochastic gradient descent:

$$\text{minimize}_{W,b}\,\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{f}(x;W,b))^2$$

The Multilayer Perceptron Neural Network procedure with backpropagation is in the following:

Algorithm 2 Multilayer Perceptron

```
Input: hidden layer sizes $m$
Set weights and biases randomly: $W_j, b_j \sim$ Uniform$(-\sqrt{m}, \sqrt{m})$
Set the learning rate, the number of epochs, and the error threshold.
For each epoch :
    i. Propagate the input forward through the network (Feedforward)
        $y = \sum_{j=1}^{m} w_j h(x; W_j, b_j) = \sum_{j=1}^{m} w_j \max\{0, W_j^T x + b_j\}$
    ii. Compute the output error $C = d - y$
    iii. Propagate the error backward through the network.
    iv. Update the weights and biases using the backpropagation algorithm.
Compute the total error for the epoch.
If the error is below the threshold, stop training.
Output the MLP model:
    $\hat{f}(x; W, b) = \hat{f}(x; W^*, b^*)$
```

The multilayer perceptron has a linear activation function in all neurons, the rectifier or ReLU (rectified linear unit), that maps the weighted inputs to the output of each neuron, and is defined as the follows:

$$h(g(x)) = h(W^T x + b) = (W^T x + b)^+ = \max\{0, W^T x + b\} = \begin{cases} W^T x + b & \text{if } W^T x + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h' = \begin{cases} 1 & \text{if } W^T x + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

The loss function measures the difference between the predicted output (y) and the true output (d), and the backpropagation algorithm updates the weights (W) and biases (b) to minimize this difference. The weights and biases of the MLP are learned through backpropagation, a gradient descent algorithm that adjusts the weights and biases in the opposite direction of the gradient of the loss function with respect to the network parameters.

In the backpropagation, the gradient is calcualated using chain rule:

$$\frac{\partial C}{\partial W_{jk}} = \frac{\partial C}{\partial h}\frac{\partial h}{\partial g_j}\frac{\partial g_j}{\partial W_{jk}}$$

$$\frac{\partial C}{\partial b_j} = \frac{\partial C}{\partial h}\frac{\partial h}{\partial g_j}\frac{\partial g_j}{\partial b_j}$$

The gradients allow us to optimize the model's parameters:

$$W := W - \eta \frac{\partial C}{\partial W}$$

$$b := b - \eta \frac{\partial C}{\partial b}$$

where η is the constant learning rate to avdid overfitting.

4.2.3 Evaluation Metrics

After fitting both gradient boosted tree regression model(GBM) and multilayer perceptron neural network model(MLP), both models are evaluated on the test set which include stock price difference between September and December 2016. Metrics including the Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 are used to evaluate the models performances. Those metrics can be defined as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}|$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y_i})^2$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y_i})^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y_i})^2}{\sum_{i=1}^{n} (y_i - \bar{y_i})^2}$$

5 Experiments

5.1 Baseline Selection

This paper by Kolasani et al. [12] aims to improve previous research that utilized social media and historical data to predict stock market trends and prices by implementing a Multilayer Perceptron Neural Network (MLP) model. The study will use a sentiment tagged Twitter dataset of 1.6 million tweets collected from Sentiment 140 for sentiment classification. The research will compare the effectiveness of the MLP model with the Boosted Regression Tree model to predict the next day's stock movement with the present day's tweets containing the "stock market", "StockTwits", "AAPL". The benefits of this paper are to analyze the effectiveness of using Twitter data to predict stock market trends and prices and determine if neural networks are more effective than traditional models.

The study used SVM for sentimental analysis of stock-related tweets with keywords "stock market," "Stocktwits," and "AAPL" since it worked best on Sentiment 140 test data. The tweets with these keywords were trained to predict DJIA and Apple Inc. closing difference values and were preprocessed.

They improved regression modeling by using a Multilayer Perceptron Neural Network model to predict stock closing difference. Training was done on stock-related data from January to August 2016, and testing on data from September to December 2016. The model predicted stock price differences the next day using average sentiment values from tweets containing "stock market," "stocktwits" for DJIA and "AAPL" for Apple Inc., obtained through SVM classification.

From this study we concluded that tweets or any other social media information can play a significant role in predicting stock market movement and neural networks perform better than the Boosted Regression Tree model. The Multilayer Perceptron Neural Network model has a lower MAE and RMSE than the Boosted Regression Tree model for all three sets of data with the keywords: "stock market", "stocktwits", "AAPL".

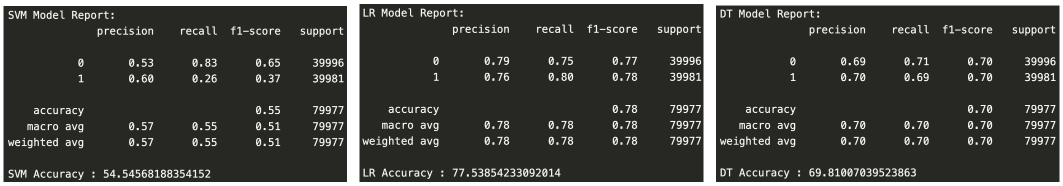
The selection of this research work as a baseline is well-suited because past studies on stock prediction mainly focused on historical data and trained different models on it. While these studies employed mathematical and machine learning techniques, they did not take into account external factors such as social media data. However, it is important to consider social media data as it can influence stock prices and trends since human behavior, which is reflected in social media, impacts prices. Therefore, integrating social media data into stock prediction models is crucial, and this paper marks a significant advancement in this area.

5.2 Baseline Implementation

The link to the complete code is provided in [13].

5.2.1 Sentiment Prediction

Figure 2 shows all sentiment model metrics comparison. As observed from the figures, the LR Model had the best accuracy of 77%, and corresponding higher precision and f1-scores. The baseline paper reported SVM being the best model, but due to no data available from the paper regarding model parameters and machine requirements, LR model received the best accuracy in the implementation.



```
SVM Model Report:
              precision    recall  f1-score   support

           0       0.53      0.83      0.65     39996
           1       0.60      0.26      0.37     39981

    accuracy                           0.55     79977
   macro avg       0.57      0.55      0.51     79977
weighted avg       0.57      0.55      0.51     79977

SVM Accuracy : 54.54568188354152
```

(a) SVM model metrics

```
LR Model Report:
              precision    recall  f1-score   support

           0       0.79      0.75      0.77     39996
           1       0.76      0.80      0.78     39981

    accuracy                           0.78     79977
   macro avg       0.78      0.78      0.78     79977
weighted avg       0.78      0.78      0.78     79977

LR Accuracy : 77.53854233092014
```

(b) LR model metrics

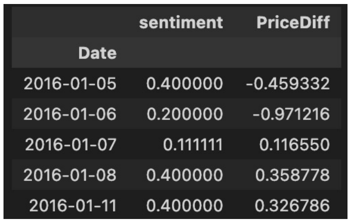
```
DT Model Report:
              precision    recall  f1-score   support

           0       0.69      0.71      0.70     39996
           1       0.70      0.69      0.70     39981

    accuracy                           0.70     79977
   macro avg       0.70      0.70      0.70     79977
weighted avg       0.70      0.70      0.70     79977

DT Accuracy : 69.81007039523863
```

(c) DT model metrics

Figure 2: Sentiment model metrics comparison

5.2.2 Stock Price Movement Prediction

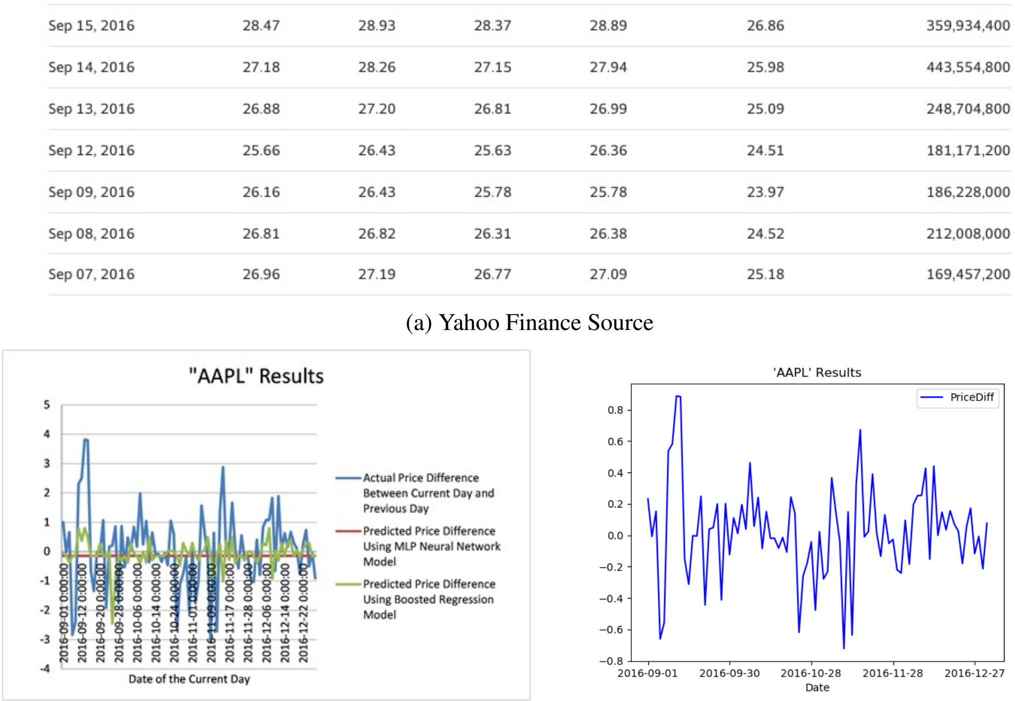
The LR model was chosen as the most effective predictor of sentiment, and it was used to analyze data from old tweets. The model identified keywords related to "AAPL" and assigned sentiment scores to these tweets. These scores were then grouped by day to predict the daily price difference of AAPL stock, which can be shown as Figure 3.



	sentiment	PriceDiff
Date		
2016-01-05	0.400000	-0.459332
2016-01-06	0.200000	-0.971216
2016-01-07	0.111111	0.116550
2016-01-08	0.400000	0.358778
2016-01-11	0.400000	0.326786

Figure 3: Avg. Sentiment Score vs. Price Difference

The numerical value presented may not be entirely accurate as the paper did not provide information on the model's parameters and machine requirements. Furthermore, the price difference obtained in our analysis differs from the baseline, possibly due to differences in data sources. Specifically, when using Yahoo Finance data, we observed a maximum difference of around 1, whereas the baseline reported a difference of around 4, although the shapes are the same shown as Figure 4.



Sep 15, 2016	28.47	28.93	28.37	28.89	26.86	359,934,400
Sep 14, 2016	27.18	28.26	27.15	27.94	25.98	443,554,800
Sep 13, 2016	26.88	27.20	26.81	26.99	25.09	248,704,800
Sep 12, 2016	25.66	26.43	25.63	26.36	24.51	181,171,200
Sep 09, 2016	26.16	26.43	25.78	25.78	23.97	186,228,000
Sep 08, 2016	26.81	26.82	26.31	26.38	24.52	212,008,000
Sep 07, 2016	26.96	27.19	26.77	27.09	25.18	169,457,200

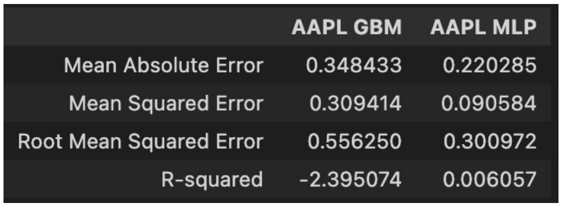
(a) Yahoo Finance Source

(b) baseline Stock Price Difference

(c) Stock Price Difference from Yahoo Finance

Figure 4: AAPL Stock Price Difference from Baseline vs. Yahoo Finance

Figure 5 shows gradient boosted tree model and multilayer perceptron model metrics comparison. As observed from the figures, the MLP Model had a lower MAE of 0.22 as well as a lower RMSE of 0.30, and corresponding higher R^2. The baseline paper reported MLP neural network is in fact on average better than the boosted regression tree model at predicting the Price difference of stocks. This paper shows a similar result since MLP has a better performance in terms of all the metrics. The baseline paper also mentions that the boosted regression tree model tended to overpredict the values whereas, the MLP neural network tended to underpredict the price difference values, where such phenomenon can also be reflected in the implementation result in the figure 6. It is noticeable that the MLP model's predictions are more consistent and have a smaller range centered around 0. On the other hand, the GBM models' predictions may deviate significantly from the true values.



	AAPL GBM	AAPL MLP
Mean Absolute Error	0.348433	0.220285
Mean Squared Error	0.309414	0.090584
Root Mean Squared Error	0.556250	0.300972
R-squared	-2.395074	0.006057

Figure 5: GBM vs. MLP model metrics

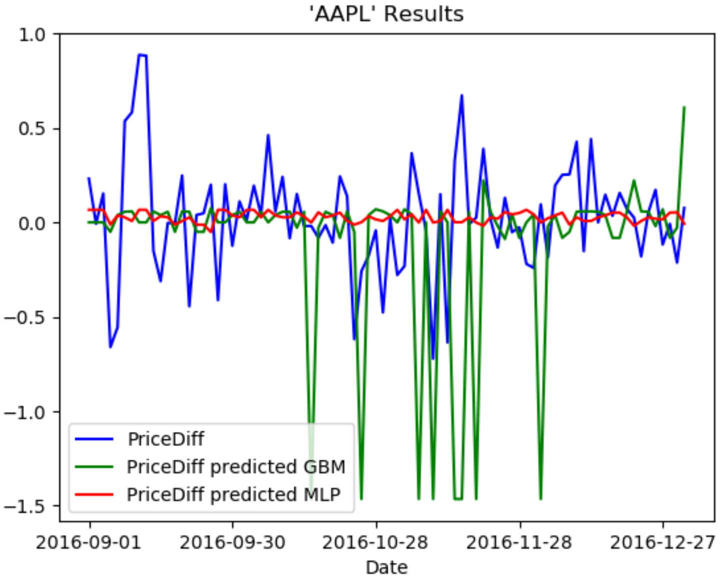
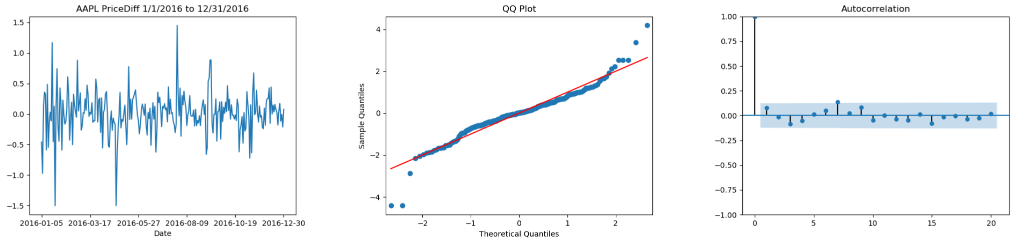


Figure 6: AAPL stock prediction

Moreover, this paper also performed time series analysis to examine the price difference value and conduct feature engineering. The study discovered that the price difference values exhibit significant autocovariance at lag 7. As a result, the price difference values for lag 1 through 7 were added as features to predict the price difference once more. The outcomes can be shown as the Figure 7.



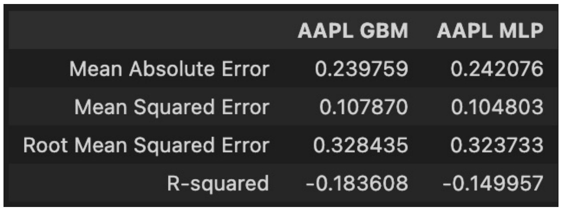
(a) AAPL PriceDiff Time Series

(b) QQ Plot

(c) AutoCorrelation

Figure 7: AAPL Time Series

Figure 8 shows gradient boosted tree model and multilayer perceptron model metrics comparison after adding new features. As observed from the figures, adding new features boosted GBM model performance, by lowering the MAE to 0.24 and RMSE of 0.31. Whereas adding new features did not boost MLP performance. It is evident that the predictions of the MLP model exhibit greater variability this time and are not centered around 0. In contrast, the GBM models' predictions tend to avoid overprediction by adding new features, which can be shown as the figure 9.



	AAPL GBM	AAPL MLP
Mean Absolute Error	0.239759	0.242076
Mean Squared Error	0.107870	0.104803
Root Mean Squared Error	0.328435	0.323733
R-squared	-0.183608	-0.149957

Figure 8: GBM vs. MLP model metrics (Feature Engineered)

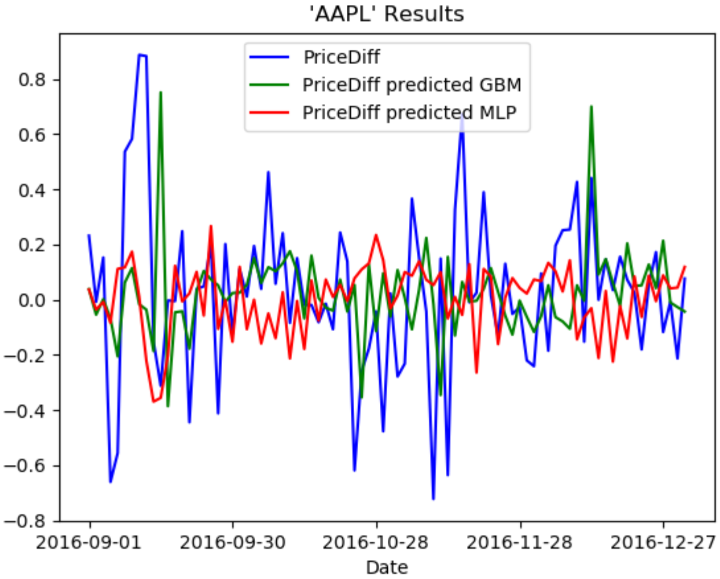


Figure 9: AAPL stock prediction (Feature Engineered)

6 Proposed Extensions

To broaden the applicability of this study, further research could involve deploying the models across a range of global stock markets and analyzing how these forecasts may vary across distinct stock categories. For instance, one possibility could be distinguishing between companies with small-cap and large-cap stocks, with the hypothesis that large-cap stocks may exhibit lower volatility and risk, while small-cap stocks may demonstrate greater volatility and necessitate more frequent computation of stock movements.

Another direction to advance the scope of this study, involves expanding the data range beyond a year to achieve more accurate outcomes. Additionally, incorporating elements such as sarcasm in the language of tweets, hashtags, taking data from more platforms and not just Twitter; for example Reddit, may improve the model's effectiveness. Due to the potential limitations of using tweets for a short duration, further research may require the integration of an attention-based model which can be seen in the works of Zhishuo et al. [14] or a transformer model with attention mechanism like Qiuyue et al. [15] used, on a more comprehensive text corpus for extended periods.

Additionally, the team intends to increase the frequency of stock predictions by training a model capable of producing hourly stock predictions for various stock groups. Achieving this will necessitate developing more accurate deep learning models, which is our team's critical future objective. To achieve this, our team plan to use advanced deep learning models like recurrent neural networks (RNNs) or bidirectional LSTMs as presented by Shengting et al. in [16] that are capable of comprehending intricate patterns and connections within the data, resulting in precise predictions. By utilizing these models, it becomes feasible to make more frequent forecasts. Furthermore, these models can be fine-tuned and trained to manage real-time data and generate predictions on an hourly basis.

Furthermore, a qualitative assessment of the models under varying economic circumstances, such as recession or prosperity, would aid in a more comprehensive evaluation of their efficacy.

References

[1] Popper, N. (2021). How Amateur Investors Took Wall Street by Storm. The New York Times. https://www.nytimes.com/2021/01/27/business/gamestop-wall-street-bets.html.

[2] Ho, T.-T.; Huang, Y. Stock Price Movement Prediction Using Sentiment Analysis and CandleStick Chart Representation. Sensors 2021, 21, 7957. https://doi.org/10.3390/s21237957

[3] Derakhshan, A., amp; Beigy, H. (2019). Sentiment analysis on stock social media for stock price movement prediction. Engineering Applications of Artificial Intelligence, 85, 569–578. https://doi.org/10.1016/j.engappai.2019.07.002

[4] Chen, R., amp; Lazer, M. (n.d.). Sentiment analysis of Twitter feeds for the prediction of stock market ... Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement. Retrieved February 28, 2023, from http://cs229.stanford.edu/proj2011/ChenLazer-SentimentAnalysisOfTwitterFeedsForThePredictionOfStockMarketMovement.pdf

[5] A. Porshnev, I. Redkin and A. Shevchenko, "Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Twitter Sentiment Analysis," 2013 IEEE 13th International Conference on Data Mining Workshops, Dallas, TX, USA, 2013, pp. 440-444, doi: 10.1109/ICDMW.2013.111.

[6] Smailovic, J., Grcar, M., amp; Znidarsic, M. (1970, January 1). Table 2 from sentiment analysis on tweets in a financial domain: Semantic scholar. Retrieved February 28, 2023, from https://www.semanticscholar.org/paper/Sentiment-analysis-on-tweets-in-a-financial-domain-Smailovic-Grcar/ba53a72840a5e9dd5787235007a873984d3a4f3d/figure/2.

[7] Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A. et al. Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods. Cogn Comput 14, 372–387 (2022). https://doi.org/10.1007/s12559-021-09819-8

[8] S. Bouktif, A. Fiaz and M. Awad, "Stock Market Movement Prediction using Disparate Text Features with Machine Learning," 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 2019, pp. 1-6, doi: 10.1109/ICDS47004.2019.8942303.

[9] Hatefi Ghahfarrokhi, A., amp; Shamsfard, M. (2020). Tehran stock exchange prediction using sentiment analysis of online textual opinions. Intelligent Systems in Accounting, Finance and Management, 27(1), 22–37. https://doi.org/10.1002/isaf.1465

[10] Kaz Anova, M.M. (2017) Sentiment140 Dataset with 1.6 Million Tweets. https://www.kaggle.com/kazanova/sentiment140

[11] Go, A., Bhayani, R., Huang, L. (2009) Twitter Sentiment Classification using Distant Supervision. http://help.sentiment140.com/home

[12] Kolasani, S. and Assaf, R. (2020) Predicting Stock Movement Using Sentiment Analysis of Twitter Feed with Neural Networks. Journal of Data Analysis and Information Processing, 8, 309-319. doi: 10.4236/jdaip.2020.84018.

[13] https://github.com/odemuno/IDL-finance-project.git

[14] Zhishuo Zhang Manting Luo Ziyu Luo Huayong Niu, 2022. "The International City Image of Beijing: A Quantitative Analysis Based on Twitter Texts from 2017–2021," Sustainability, MDPI, vol. 14(17), pages 1-21, August.

[15] Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, Peide Liu, Transformer-based attention network for stock movement prediction, Expert Systems with Applications.

[16] Shengting Wu, Yuling Liu, Ziran Zou Tien-Hsiung Weng (2022) S I LSTM: stock price prediction based on multiple data sources and sentiment analysis, Connection Science, 34:1, 44-62, DOI: 10.1080/09540091.2021.1940101