Tweet Sentiment Analysis to Predict Stock Market

Stanford CS224N Custom Project

Christian Palomo
Department of Mathematical and Computational Science
Stanford University
palomoc@stanford.edu

Abstract

In this project, we aim to develop an NLP model that can predict the stock market of certain stocks by analyzing Twitter sentiment using a transformer based neural network and show that it makes stock predictions with reasonable accuracy. Previous implementations of news-based stock market predictors have usually only focused statistical methods instead of Machine Learning and Natural Language Processing techniques. The NLP methods for sentiment classification in the context of finance are rather new, so this model aims to replicate and focus on fine-tuning the Sentiment Analysis component of the model, which is more aligned with the scope of this class.

1 Key Information to include

- Mentor: Ansh Khurana
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

The stock market is a crucial aspect of modern economies, and investors are always seeking ways to predict market trends and make informed investment decisions. This can be difficult because the stock market is notorious for being volatile, unpredictable, and fast-moving. Sentiment analysis on social media has emerged as a tool that can help investors gauge the mood of the market and predict stock prices. Twitter, in particular, has become a popular platform for investors to express their opinions on various companies and stocks. There have been many instances on Twitter where important public figures, CEOs of companies, and various financial experts make claims about the economy or certain companies, and this news gets reflected in the stock price with drastic fluctuations. Twitter distinguishes itself from its competitors by allowing you to hear news from companies and individuals directly, and instantaneously. Because of this, Twitter is a prime candidate for drawing patterns with the stock market; it moves as fast as the market does. As a result, we will be focusing solely on Twitter as the social media platform of choice from which our model analyzes text. The model that we build aims to capture this relationship between certain tweets and stock prices.

Natural Language Processing (NLP) has made significant strides in recent years, and sentiment analysis is one of the most widely studied applications of NLP. The goal of sentiment analysis is to identify the sentiment expressed in a text, which can be positive, negative, or neutral. Within the context of finance, we will be using sentiments that hold a slightly more nuanced meaning than typical positive or negative sentiments–for certain financial texts and tweets, they can be classified as "bullish" (corresponding to a rise in stock price), "bearish" (corresponding to a drop in stock price) or neutral.

The main goal of this project is to develop an NLP model that can predict the stock movement of certain stocks with reasonable accuracy by analyzing Twitter sentiment using a transformer-based

neural network. As we have seen above, there have been many previous attempts at using news and Twitter sentiment analysis to inform stock market predictions. While accurate, these models have used older machine learning techniques that are not as efficient as the Transformer based methods emphasized both in class and in industry today. Therefore, another goal of this project is to utilize these transformer-based methods to make faster, more efficient models, while still retaining the accuracy of predictions that previous models have achieved.

2.1 Problem Statement

Given historical price data and tweets for a stock s over the previous T trading days (a time window over the day range [t - T, t - 1], which we refer to as the "Lookback Window"), we define the price movement of stock s from day t - 1 to t as:

$$Y_t = \{ \begin{array}{l} 1, \textit{ if today's closing price} > \textit{yesterday's closing price} \\ 0, \textit{ if today's closing price} \leq \textit{yesterday's closing price} \end{array}$$

Given these definitions, we have two goals in mind:

1. Classify several tweets pertinent to the given stock using Sentiment Analysis with a Transformer based neural network.
2. We aim to estimate Y_t–the price of the stock today with accuracy at least as good as baseline models.

3 Related Work

The use of mathematical and statistical models on historical data to predict the stock market has been of interest to people for decades. Over time, as modeling tools get faster, more powerful, and more sophisticated, so have the models used for stock prediction. In recent years, the use of NLP techniques to predict stock prices has gained significant attention in the field of finance. The incorporation of sentiment analysis on social media platforms, particularly Twitter, has proven to be a valuable source of information for predicting stock market trends. Before the rise of Machine Learning modeling, traditional statistical models have been applied for classification, including linear regression, Support Vector Machines (SVM), Decision Tree (DT), Boosted Tree, and Random Forests Kolasani and Assaf (2020). In the research and methods conducted by (Kolasani and Assaf, 2020), each of these 5 models are used to predict the sentiment (positive or negative) of various tweets.

Table 1: Kolasani and Assaf (2020) Sentiment Performance Metrics (non-NLP)

Model	**Accuracy**	**F1 Score**	**Precision**	**Recall**
LR	0.82	0.82	0.82	0.82
SVM	0.83	0.83	0.83	0.83
DT	0.72	0.72	0.72	0.72
BT	0.70	0.70	0.70	0.70
RF	0.70	0.70	0.70	0.70

As we can see, the top-performing model (Logistic Regression) performed with an accuracy of around 0.82. This will be used as our baseline in an effort to show that NLP Transformer based neural models will outperform the traditional non-ML models for classification.

In regards to the stock prediction component of the model, there have also been many different methodologies with different complexities. One of the most common models (although not necessarily simple!) for price prediction is the Black-Scholes model. Other more complex techniques are continuously being built, such as Graph Attention methods used in the MAN-SF model to predict stocks from related sectors Sawhney et al. (2020). However, since the class revolves around Natual Language Processing, we will be focusing more on the NLP-related aspect of the model through sentiment analysis. As a result, we will keep the stock prediction component of the model relatively simple–these methods can get very complex and much outside the scope of an NLP class. Instead of focusing on the magnitude of daily price changes, we will only be focusing on the direction of the price change to simplify the model and to make it easier to understand the evaluation metrics.

In sum, our proposed methodology extends the prior work on sentiment analysis-based stock price prediction by leveraging NLP techniques within transformer-based frameworks. Additionally, instead of classifying texts or tweets as positive or negative, we will take it one step further and include a "neutral", and have a more nuanced meaning of "positive" and "negative" by classifying the tweets as being "bullish" and "bearish". By utilizing our approach aims to improve the accuracy and applicability of stock price predictions in the financial domain.

4 Approach

The model that I build contains two parts: one consisting of a sentiment classifier for tweets, the other being the stock prediction model.

4.1 Sentiment Analysis

- 3 Classifications of Twitter Sentiment:
  Bullish (buy the stock), Bearish (sell the stock), Neutral (do nothing).

  The model will output probabilities for each of the three labels; we will choose the argmax as the label of "Sentiment" for each tweet.

  Thus, when we pre-train the model, we will not use a generic sentiment classifier, but one that is appropriate for the financial context.
- Dataset: "zeroshot/twitter-financial-news-sentiment" zeroshot. This dataset is "an English-language dataset containing an annotated corpus of finance-related tweets. This dataset is used to classify finance-related tweets for their sentiment."
- Pretrained Model & Tokenizer: "ahmedrachid/FinancialBERT-Sentiment-Analysis" (Hazourli) This transformer-based model outperforms traditional BERT on finance texts. Optimized for Financial Texts, the positive label corresponds to buy and the negative label corresponds to sell, not necessarily positive and negative in the traditional sense.
- Finetuning: Using the above Dataset and pre-trained Model, the model was further finetuned with the following parameters:

  ```
  output_dir=repo_name, learning_rate=2e-5,
  per_device_train_batch_size=16, per_device_eval_batch_size=16,
  num_train_epochs=2, weight_decay=0.01.
  ```

  Using the Adam optimizer, the model took about 3 hours to finetune on Google Collab.
- Preprocessing of Tweets. Before the tweets are tokenized and evaluated, I wrote a function that preprocesses each tweet, replacing website links within the tweet with a "[URL]" token, removing the "#" symbol in front of hashtags, and removing repeated punctuations.

  For instance, the tweet: "Check out this crazy link about #TeslaMotors!!!! http://t.co " would become "Check out this crazy link about TeslaMotors! [URL]" after preprocessing.

4.2 Price Prediction

Following the description of the price movement as outlined in the problem statement, we will use the "RandomForestClassifier" model from the sklearn library to predict the price movement (increase or decrease) of today's stock for a given stock symbol. Using the TwitterAPI, the past "Lookback Window" days of tweets and historical price data are collected. We collect the "Daily Tweet Limit" amount of tweets per day of tweets containing the stock ticker symbol or the company name. After we run sentiment analysis on the tweets that are collected, we feed the following as input features into the price prediction model:

Features:

- Retweets: A way to calculate a multiplier effect for each tweet).
- Sentiment score from the previous model where (-1, 0, 1) corresponds to (Bearish, Neutral, Bullish)

- Adjusted Closing Price. The adjusted closing price takes into account stock splits and dividends to give a more accurate representation of an investor's profit if they own the stock.

Once these features are collected, the tweets are grouped by each day, and a **Weighted Sentiment Score** is calculated. This is found by taking the Normalized Retweet Count and multiplying it by the sentiment and summing this value over all the corresponding tweets for that day. This weighted sentiment aims to give a holistic view of the sentiment of all the tweets for a company per day. Scores closer to 1 indicate a more bullish outlook on that day, while scores closer to -1 indicate a bearish outlook.

5 Experiments

5.1 Data

For training and testing our Sentiment Model, we use the "Twitter-Financial News Sentiment" dataset from the HuggingFace website[citation]. There are 9,938 labeled tweets in the Training set and 2,486 in the Validation set. This dataset contains various finance-related tweets and classifies them into three categories: "bearish", "bullish" and "neutral".

For tweets, we use the Twitter API to harvest tweets within a specified time frame that contain the company's stock ticker (e.g. $TSLA) in the tweet. You could also choose to scrape for tweets that just contain the company name, or the Twitter handle of the company (e.g. Tesla, #TeslaMotors) but I chose not to because they might not talk about the company's financials in the same manner that tweets containing the stock ticker would. If your Twitter API does not get approved (Twitter stopped administering free usage of its API in Feb 2023), there is an HTML scraper method I used using `sncrape`. [citation]

For historical stock data, we pull the financial info from Yahoo Finance which is also available through the `yfinance` module in Python. For any stock within a given timeframe, we can take the Adjusted Closing Price(1. Adjusted Closing Price is used in lieu of regular Closing Price because Adjusted Closing Price accounts for stock splits and dividends, and is a more accurate number for the "value" of a stock) for the end each trading day. Non-trading days data for both stock and tweet data are ignored for consistency in matching the dates of tweets to stock data. The stock data for $TSLA from August 2022 to December 2022 is collected and stored in a CSV.

5.2 Evaluation method

To evaluate the model on both its sentiment analysis accuracy and its stock prediction accuracy, we will use accuracy and F1 score:

$$F_1 = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{1}$$

where tp = number of true positives, tn = number of true negatives, fp = number of false positives and fn = number of false negatives.

In the context of stock investing, it may be important to differentiate the difference between false positives and false negatives. Here, false positives indicate that you *lose* money on your invested stock–the stock price you bought was predicted to increase when it actually fell. False negatives indicate that you missed out on *potential* gains–the stock you chose to short (or not buy the long position) was predicted to fall but actually rose. This is the motivation for including both accuracy and F1 score as evaluation metrics. "Accuracy is used when the True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives are crucial" Huilgol (2019)

5.3 Experimental details

First, we choose to evaluate several sentiment classification models: standard BERT binary classifier, Twitter-roBERTa-base for Sentiment Analysis (Barbieri et al., 2020), and FinancialBERT-Sentiment-Analysis (Hazourli). Of these, the FinancialBERT Model has the best success, so I chose to use that as the model that I would proceed to finetune for sentiment analysis.

The pre-trained model was trained further on a random sample of 3000 tweets from the "Twitter-Financial News Sentiment" database and evaluated on 300 tweets. After playing around with hyperparameters, the ones that minimized loss after trial and error were: Training used a learning rate of 2e-5, batch size of 16, 2 epochs, and weight decay of 0.01. Training took around 3 hours to run and achieved a loss (cross-entropy) of 0.614.

Because the preprocessing function preprocesses the text in a very Twitter specific way, I wanted to evaluate the effectiveness of its improvement in a quantitative way. After finetuning the model, I evaluated it on tweets that were tokenized without the preprocessing function and calculated the difference in accuracy and F1 score.

For evaluating the stock predictions, the `'y_label'` for each stock price per day was calculated according to Y_t in the problem statement. The model was evaluated over a 5-month period from August 2022 through December 2022 on real-world stock data and tweets for $TSLA, and its accuracy and F1 score were computed both over this 5-month period and for the best-performing month (November 2022).

One of the hyperparameters of the stock model is the "Lookback Window"–how many trading days back you look at to inform the model with tweet sentiment and historical stock data. The model was evaluated over the 5-month period using Various Lookback Windows between 2-14 trading days and their accuracy is plotted and compared.

6 Results and Analysis

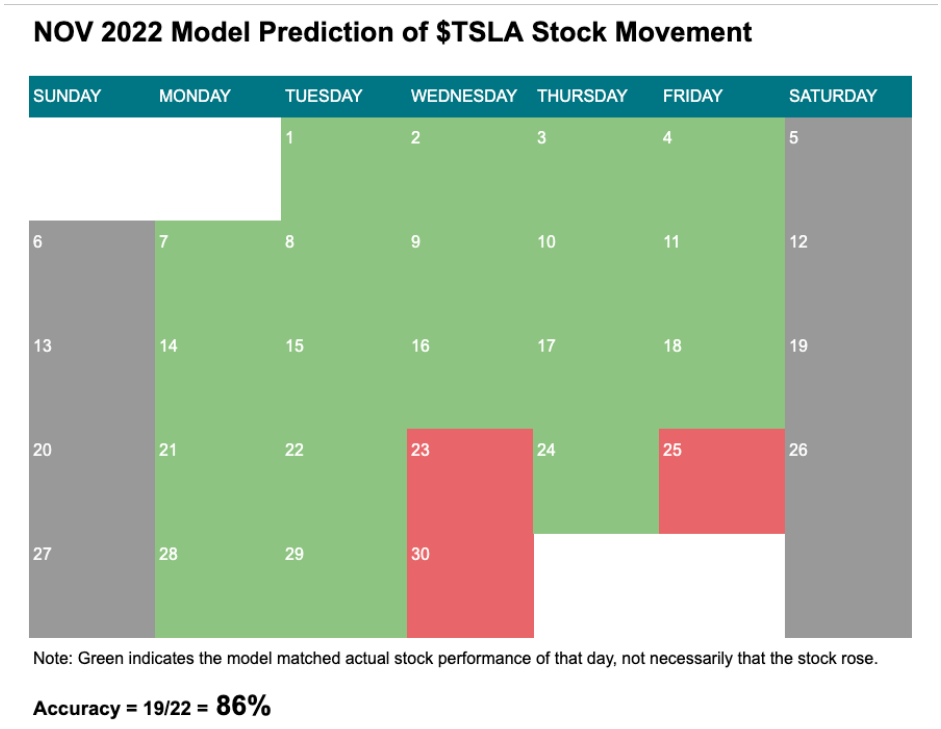

Figure 1: Stock Prediction Performance of Model for NOV 2022 for $TSLA

For each difference in the accuracy and F1 between the models, we can identify specific attributes unique to each model that provide a relative advantage or disadvantage. Let us start be analyzing the results from Table 2.

Model	Accuracy	F1
Finetuned FinancialBERT-Sentiment-Analysis	**0.843**	**0.843**
FinancialBERT-Sentiment-Analysis	0.803	0.803
FinancialBERT-Sentiment-Analysis (no preprocessing)	0.756	0.756
Twitter-roBERTa-base for Sentiment Analysis	0.711	0.711
BERT	0.653	0.653

Table 2: Sentiment Model Comparisons

6.1 Sentiment Analysis

We see that the Finetuned FinancialBERT (FinBERT) performed the best for sentiment classification out of all of the models, with an accuracy and F1 score of 0.843. We see that this Finetuned FinancialBERT model had a 4% increase in performance compared to the FinancialBERT model that was not finetuned. Thus, we can attribute the 4% increase to the benefit gained by finetuning on *Twitter specific* financial texts. We can attribute the largest delta in performance between the traditional BERT classifier and the FinBERT model–around a 10% increase. Additionally, we noticed about a 5% increase in performance from the FinBERT model without preprocessing tweets to the same model *with* preprocessing tweets. These two deltas in performance suggest that the syntax and language on Twitter are noticeably different that other texts in news articles and across the internet, within the subject of finance. This is important to keep in mind–*the platform (Twitter vs Facebook vs News Websites) from which you are drawing data from has a noticeable effect on the classification accuracy.*

From the deltas in performance we uncovered, this suggests that the room to delve deeper into sentiment classification for this project is twofold.

One is to increase the size of the finetuning dataset. While the general availability of tweets pertaining to a particular stock is high, the availability of labeled data sets with regard to the financial sentiment of tweets of low, which makes training this model on a large scale difficult. That being said, by training on larger, more diverse data, we might expect to see an increase in the delta of performance.

The second is to delve deeper into the stylistic and syntactic differences in tweets and finance jargon compared to traditional texts and across other platforms. More specifically, it might be worth looking into tokenizers that are Twitter-specific that can more accurately parse the Twitter-specific syntax–hashtags, emojis, Twitter handles, URLS–rather than just filtering these instances out altogether.

6.2 Stock Prediction

Model	Accuracy	F1
Random Forest Classifier	**0.913**	**0.897**
Random Forest Classifier (w/o Weighted Sentiment)	0.869	0.846

Table 3: Stock Prediction Model Comparisons

Turning our attention to Table 3., we see that the difference in performance attributed to weighted sentiment is a 4.4% increase in accuracy. In some way, this weighted sentiment idea parallels the idea of using attention mechanisms. Since the idea of weighting sentiment shows promise, the idea could be taken further by incorporating more attention-like mechanisms, including but not limited to temporal attention across the different days of tweets, and attention across the tweet author (as some authors have more of an effect on the stock price than others).

From the graph that compares the size of the lookback window against the accuracy and F1 score of the stock predictor model, we can see that *increasing the lookback window greatly increases model's performance* in the accuracy of stock prediction, as we can see with the following graph. In experimenting with differing lookback windows, the accuracy ranges from about 60% on the low end, before it levels out at about 85% with a lookback window of around 14 days. Intuitively, this means that tweets and previous stock information up to 14 days prior still as a noticeable effect on informing market decisions. Thus, when making investment and trading decisions, you have a lot to

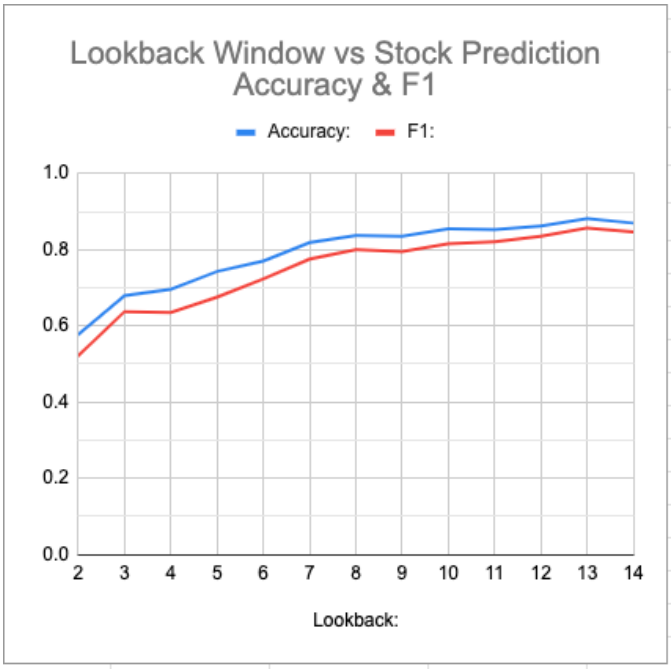

Figure 2: How Performance of Model Increases with Differing Lookback Windows

gain from looking around 2 weeks into the past. For a time over 2 weeks, the information has a more negligible effect.

7 Conclusion

In this project, we develop a model that both predicts the financial sentiment of tweets using BERT-based transformer models and uses the weighted sentiments to predict the daily rise or fall of a stock corresponding to a ticker symbol. We demonstrated that by finetuning on data that was *relevant and similar to the text and platform the model would see new data*, and implementing twitter-specific preprocessing, we improved the accuracy of the sentiment classifier to where it outperformed non-ML models. For predicting the direction of a stock price, we demonstrated that a *weighted-sentiment* approach based on the number of retweets provides better performance than equally weighted sentiments.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Ahmed Rachid Hazourli. Financialbert for sentiment analysis. `https://huggingface.co/ahmedrachid/FinancialBERT-Sentiment-Analysis`.

Purva Huilgol. 2019. Accuracy vs f1 score. `https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2`.

Sai Vikram Kolasani and Rida Assaf. 2020. Predicting stock movement using sentiment analysis of twitter feed with neural networks. `https://www.scirp.org/journal/paperinformation.aspx?paperid=104142`.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn Shah. 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8415–8426, Online. Association for Computational Linguistics.

zeroshot. Twitter financial news sentiment dataset. `https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment`.