Predict Stock Prices Based on Tweets and News

Yaniv Steinberg
yaniv.steinberg@post.idc.ac.il

Guy Schneider
guy.schneider@post.idc.ac.il

Submitted as final project report for the NLP course, IDC, 2022

1 Introduction

The stock market is known to be one of the most intriguing aspects of our world, at least for some. It holds the promise to be a part of the global economy. It is the pinnacle of capitalism, every one of us can participate and enjoy the feast. Nonetheless, it is widely believed that for the common investor, the stock market is non other than a disguised casino. If only one could tell if a given stock will increase its value tomorrow, how great would that be? Stock price predictions based on historical performance were proven to be inaccurate. So does predictions based on tweets alone. In this paper we propose to make a prediction using multiple resources instead of just one. The resources are historical price data, tweets sentiment and news sentiment, regarding a specific stock. The company we chose to focus on in our project is Apple. We chose Apple because we believe that this tech giant provides a stable stock history with enough news and tweets data coverage.

1.1 Related Works

Since the stock market is so popular, many people tried to solve this problem before us. Some notable mentions include an interesting work by Michael Jermann et al. who has tried to predict stock movement through executive tweets [1]. In his work he selected tweets exclusively from users that are related to the company that its' stock he tries to predict. He tested several tokenizations methods such as an adaptation of Glove and Twokenize but the best tokenizer was NLTK. For the price prediction he used logistic regression which achieved a poor fit on the training data. Another such work is from Kevin Hu, Daniella Grimberg, Eziz Durdyev [2] which have used a pretrained BERT model to get the sentiment of the tweets. Another notable work is from Thormann et al. [3] which used TextBlob for the sentiment analysis of the tweets. In his work, TextBlob has been trained on movie reviews corpus which gives less weight to emojis as can be seen in [4]. Since tweets contain many emojis, this can be problematic. In the majority of the literature we came across, sentiment analysis is used to extract a signal from the tweets, often classified to 'Positive' or 'Negative' and not the actual polarity value which we plan to use.

2 Solution

2.1 General approach

As in previous works, we approached the problem by using a sentiment analysis model followed by second model to predict the price.

Sentiment analysis
In the first stage of our project, the goal is to extract the sentiment out of tweets and news related to our chosen stock. We wanted to experiment with several sentiment analysis models in order to choose the best one and have decided to investigate Bert, Vader and a LSTM model. In addition, we used a 4th ”model” as a baseline reference, in which all sentiments were set to neutral. For the training we wanted to use a dataset that contains financial tweets / news articles and their sentiment which we couldn't find so we have decided to use the dataset Sentiment140 [5] which contains 1.6 million tweets and their sentiment, but is not financially specific. We have created and trained both a LSTM model and a Bert model, and used a pretrained Vader model. Many of the previous works used a binary value (Negative or Positive) for the sentiment output. We wanted to try and use a more detailed sentiment value as we think that the more expressive the sentiment is the more accurate the result will be. Therefore, our sentiment output is a number between -1 and 1 where -1 translates to Negative and 1 to Positive.

Price prediction
In the second stage of our project, the goal is to build a predictor of stock price based on several features including the sentiment analysis. Twitter's API has many limitations such as restrictions on older tweets. Since we needed old data, we decided to use a scrapped Tweets dataset. We chose the dataset ”Tweets about the Top Companies from 2015 to 2020” [6] which consists of more than 5 million tweets, out of those 1,384,359 are ”AAPL” tweets, and the dataset ”Historical financial news archive” [7] which consists of more than 221,513 articles, out of those 20,231 are ”AAPL” news articles separated to news and opinions. The dataset we used for the historical stock data was downloaded from Yahoo Finance [8]. These 3 datasets where preprocessed and combined to create one dataset that consists of 5 features and one label. The features were tweets sentiment score, news sentiment score, stock volume, percentage of change in the stock price compared to the previous day and the stock price. The label was the stock's price of the next day.
As for the prediction model, we saw a few models trained to predict the future price of a stock, some of them have used a simple linear regression which in our opinion does not make much sense as a stock price is highly impacted by the historical performance of the prior days, so we need a model that can utilize data from the previous days. Therefore, we chose a LSTM model.

2.2 Design

We chose Bert and LSTM as our models to train for sentiment analysis. Technical information regarding the LSTM model for sentiment analysis: We used a keras tokenizer, the vocabulary size was 290,575. The model itself is built from a convolution layer that has 64 filters and a kernel size of 5, followed by a LSTM layer and after that two fully connected layers. The activation function between the layers was ReLu, and Sigmoid was used at the end. In addition, different dropout layers were used in order to add regularization throughout the model. Training this model for 10 epochs on NVIDIA Tesla K80 GPU took about 26 minutes.

Technical information regarding the Bert model for sentiment analysis: For the tokenizer we used bert's tokenizer. We used weight decay and Adam as an optimizer. For the model itself we took the pretrained 'bert-base-uncased' model and added on top of it a dropout layer and a fully connected layer to output a single result. Training this model for 5 epochs on the same GPU as above took about 130 minutes.

The prediction model was straight forward. A LSTM model with a hidden layer of size 150, followed by a fully connected layer of a single output. That is based on previous work that was done on this task and because it makes sense considering the data is a temporal sequence (days). The loss function was MSE because we wanted to calculate how far we were from the actual price. The optimizer was Adam. Because of the small amount of data, training this model for 15 epochs on the same GPU as above took us 3 minutes.

During the training of these models we faced some technical challenges. For example, the output of the Bert and LSTM sentiment analysis models was a value between 0 and 1 where 0 means absolute negative and 1 absolute positive. The output of the Vader model was a number between -1 and 1, so we needed to scale them to the same range. We used the library MinMaxScaler for the job. Furthermore, MinMaxScaler was used to scale the features stock price and volume to the range of -1 to 1 as well. In addition, we came to the conclusion that not all tweets have an equal impact. Tweets with higher "engagement" (number of followers, likes, retweets and comments) are more likely to have an impact on the stock price. Therefore, we articulated the above metrics to a single number, the tweet's engagement score, and discarded of tweets with low engagement score from the dataset.

3 Experimental results

We have trained 3 models and will explain the results of each one of them:

Data split		
Dataset	Train	Test
Sentiment140	1,280,000	320,000
Stock Price - YFinance	1,003	250
Tweets about Top Companies	1,107,487 (1,003)	276,871 (250)
Historical Financial News Archive	16,184 (1,003)	4,046 (250)

Table 1: **Train - Test Dataset Split**

The Yahoo! Finance Stock Price dataset consists of the adjusted daily close price (we used the adjusted since some stocks had a split a few years ago - AAPL for example). The dataset is small as there were 1253 trading days in 2015-2019. For the tweets and news datasets we have aggregated the tweets' and news' sentiment according to the relevant trading days (counted in parenthesis).

Training Results			
Dataset	Model	Metric	Result
Sentiment140	LSTM	F1	0.78
Fine-tune Sentiment140	Bert	F1	0.83
Pre-trained	Vader	F1	0.96
Sentiment140	None	F1	0.5

Table 2: **Training Results - Sentiment Analysis**

The first row is the LSTM sentiment analysis model trained on Sentiment140 dataset (the best results we achieved as seen in the table are using lr=1e-3, batch-size=1024, and by reducing the learning rate when reaching a plateau by a factor of 0.1). The second row is the Bert sentiment analysis model which we fine tuned using Sentiment140 dataset (the best results we achieved as seen in the table are using lr=3e-5, batch-size=6, and weight-decay=0). The third row is the Vader sentiment analysis model which we have downloaded and used the pre-trained model. The fourth row is meant to illustrate the neutral sentiment ”model” we used as the baseline. As it classifies every tweet as neutral, it achieves an F1 score of 0.5 trivially.

Training Results			
Price Prediction Model	Sentiment Analysis Model	Metric	Result
LSTM	LSTM	MSE	0.114
LSTM	Bert	MSE	0.132
LSTM	Vader	MSE	0.088
LSTM	None	MSE	0.12

Table 3: **Training Results - Price Prediction**

This table represents the price prediction model trained on the combined preprocessed dataset. These table results were achieved by training our models on Apple's (AAPL) historical stock data. The model was trained several times using different datasets, each acquired by using a different sentiment analysis model that we have trained before hand as can be seen in table 2. We have also compared the models with a basic model without sentiment analysis at all by setting the sentiment to be 0 (neutral) to all tweets and news. The best stock predictions results we achieved as seen in the table are using lr=0.001, batch-size=1 as the dataset is quite small, and using a train window of 4 days for the LSTM neural network layer.

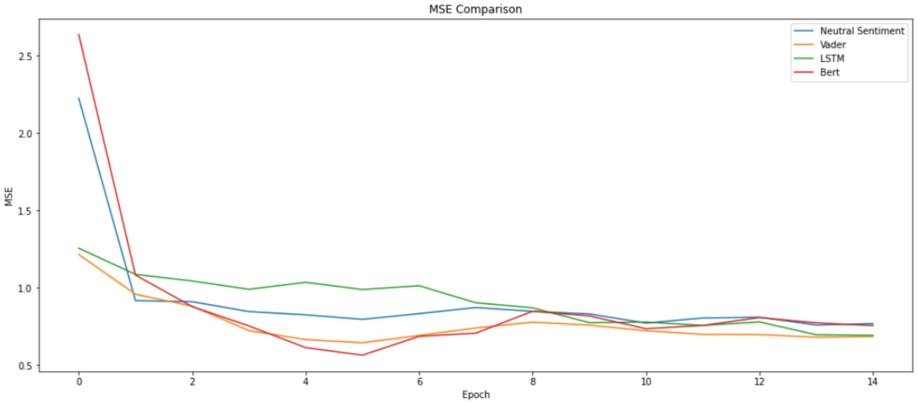

Figure 1: MSE Score comparison on validation during training

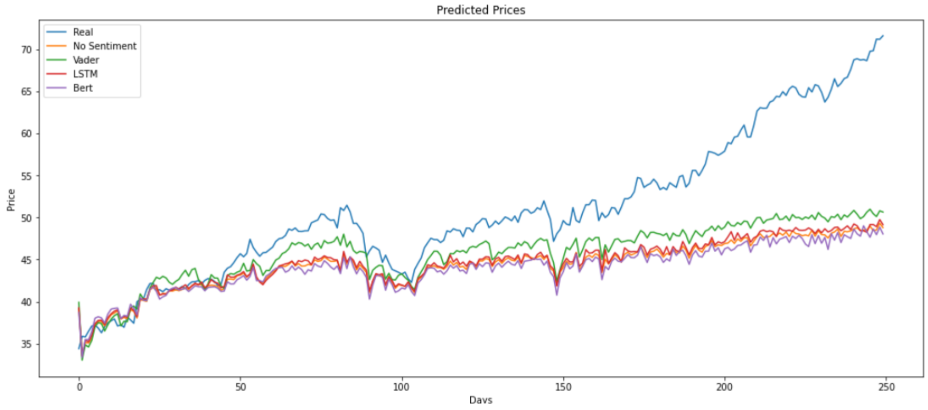

Figure 2: Price Prediction Comparison
The gap between the actual and the predicted price from day 170 and forward can be explained by the fact that Apple's stock prices have risen sharply compared to the 4.5 years prior to that time.

4 Discussion

In this project we explored sentiment analysis in the hope of producing a useful model for stock price prediction. The main features we focused on were the sentiment for a specific stock carved out of related tweets and from related news articles. In addition, after much experimentation, we also added the trading volume of the stock, the percentage of change in the stock price compared to the previous day and the stock price as features. We tried to tweak the prediction model in several different ways, at times considering changing our output from a price to a binary output (Did the stock price increase or decrease?), but previous works showed us that regression is more suited to the task than classification. Regarding the sentiment analysis models we trained, we were able to achieve relatively good results for both (Bert - 83, LSTM - 78), by training them on the sentiment140 dataset. Comparing them to the results of the Vader Sentiment model and the neutral-sentiment model, which we used as a baseline, provided some interesting insights.

An important point to note regarding the sentiment analysis models is that they did not perform as well as we expected at predicting the sentiment of the financial tweets and articles. After examining the issue, we have come to the conclusion that this is quite reasonable as the dataset that these models were trained on was a dataset that contained general tweets and these tend to be written in a different manner than financial related tweets.

Last but foremost, it is crucial to keep in mind that stock prices are affected by many factors, and not just by the social media and news sentiments. Furthermore, it is often the case that the market fluctuates irrationally and deceives even the most experienced of analysts. Considering the above and the fact that this problem is inherently difficult, we were not successful in predicting the stock price with good accuracy. Despite that, we found that one of the models (Vader) performed better than the baseline - the neutral sentiment model. Thus, we conclude that sentiment analysis of financial social media and news articles can provide additional value towards predicting stock prices and should be taken into account in future pursuits after this elusive task.

5 Code

https://github.com/yaniv-steinberg/Predict-Stock-Prices-Based-on-Tweets-and-News

References

[1] Michael et al.”Predicting Stock Movement through Executive Tweets” URL: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2743946.pdf

[2] Kevin Hu et al.”Twitter Sentiment Analysis for Predicting Stock Price Movements” URL: http://cs230.stanford.edu/projectsfall2021/reports/103158402.pdf

[3] Thormann et al. ”Stock Price Predictions with LSTM Neural Networks and Twitter Sentiment” URL: http://www.iapress.org/index.php/soic/article/view/1202/758

[4] TextBlob vs. VADER for Sentiment Analysis URL:https://pub.towardsai.net/textblob-vs-vader-for-sentiment-analysis-using-python-76883d40f9ae

[5] Sentiment140 dataset URL: https://www.kaggle.com/datasets/kazanova/sentiment140

[6] Tweets about the Top Companies URL:https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020

[7] Historical financial news archive URL: https://www.kaggle.com/datasets/gennadiyr/us-equities-news-data

[8] Y-Finance AAPL historical data URL: https://finance.yahoo.com/quote/AAPL/history?p=AAPL