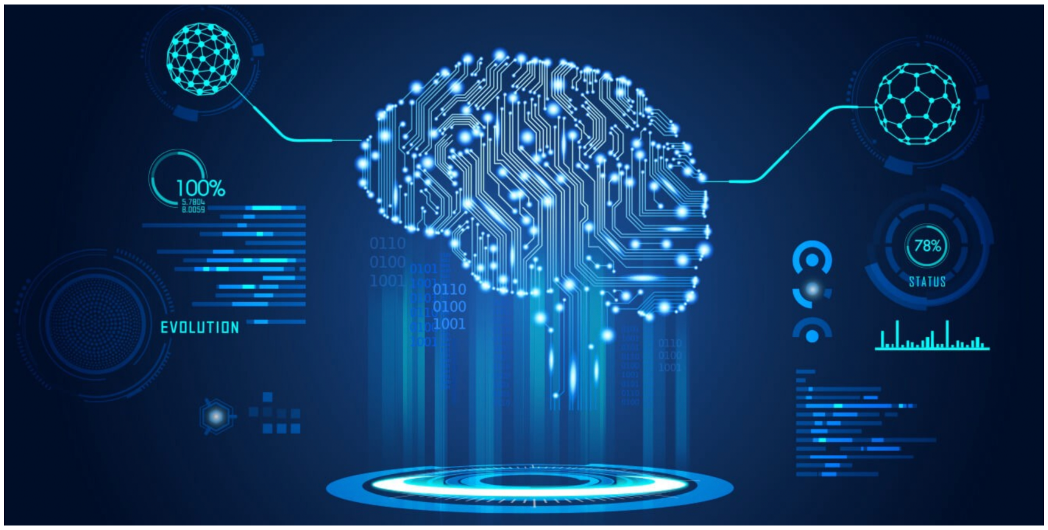
Stock Market Sentiment Predictor

Yawar Ashraf

Abdulrahman Shahzad

Harishguna Satgunarajah



Sentiment Analysis Model [1]

ECE324: Machine Intelligence, Software, and Neural Networks

Table of Contents

1. Introduction

Link to Github Repo : https://github.com/ashrafya/ECE324_StockMarketPrediction

The problem that we attempted to address is predicting if the stock market is bullish, bearish, or static for the upcoming day. Hence this information can be used by traders, investors, and the general public as a quick and efficient method to get a summarized prediction of how the market will be tomorrow to enable them to intelligently select their investing decisions. Hence we are hoping to answer the fundamental question that every trader and investor may have on a daily basis, "How will the market perform tomorrow?" or "Is today a good or bad day to invest in the market ". To answer these questions manually is a very tedious task that many traders go through every day as they have to look at factors such as the investor greed index, manually perform sentiment analysis on multiple news and tweets relating to the stock market, analyze multiple daily percentage changes of stocks and ETFs, look at technical analytical factors such as RSI, MACD, ADX and then having to compare all of this data with past historical trends. Hence daily traders have to perform this labor-intensive task to give their prediction on how the market will perform tomorrow and hence act accordingly.

Stock Market: Bullish or Bearish [2]

Therefore our mission was to formulate a machine learning model which learns trends from past historical data and at the end of every stock market day, provides a categorization of “Bullish”, “Bearish”, and “Static” summarizing its prediction of the market for the upcoming day. Our model will consist of an RNN model to predict the S&P value for the next day using technical analysis factors, and percentage changes in stock/ETF prices. Furthermore, there will be an NLP model that will perform sentiment analysis on relevant tweets to output a value for investor sentiment. In the end, we use an AI model to combine these two values to accurately predict the market behavior for the upcoming day.

2. Background & Related Work

2.1 - Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach

In the Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach paper [3], the researchers utilized convolutional neural networks to classify stocks and ETFs as “Buy”, and “Sell”, or “Hold” at a particular point in time. The team decided to represent a particular stock for a certain time window as a matrix or an “image”. Each image was a 15x15 2D array, with 15 technical indicators on one axis and 15 days of data on the other axis. Convolutional neural networks are generally used to detect small localized details as well bigger picture information. By using a CNN on these matrices, the model is able to detect correlations between indicators and time and how they interact to determine the performance of a stock. The paper compares the model to many other similar models from academia and shows that the average performance of this model is the best with an average annual return of 12.59% from 2007 to 2017 for DOW30 stocks. Another interesting contribution of the paper is its algorithm for labeling stocks as buy/hold/sell.

2.2 - Sentiment Analysis Based on Deep Learning: A Comparative Study

This paper studies the performance of different sentiment analysis methods. The paper compares different preprocessing methods such as word embedding and Term Frequency — Inverse Document Frequency [4]. Furthermore, the paper studies various deep learning approaches to analyze the data. This includes dense neural networks, convolutional neural networks, and recurrent neural networks. The researchers discovered that word embedding was a superior preprocessing method to TF-IDF when it came to sentiment analysis using deep learning approaches. Moreover, the researchers found that convolutional neural networks were the best balance between computational time and performance. These CNNs took the embedding matrix as input, and output a positive or negative class based on the sentiment of the input text. Additionally, the paper discusses alternate methods of sentiment analysis, such as lexicon-based methods and traditional machine learning.

3. Data Collection

3.1 - Web Scraping for Investor Sentiment Analysis

Our team scraped tweets using the free-to-use Twitter intelligence tool twint [3]. The tweets were scraped using a simple query that returns all tweets that include the specific keyword(s) chosen by the team. For example, the selected words may include “S&P500” or “stocks". Then these tweets were saved in a CSV file for future use.

Twitter logo [5]

3.2 - Data Acquisition for the time series RNN model

The goal was to train an LSTM model to predict the closing price of the S&P 500 for a given day based on historical data of technical factors. In order to do this, a sufficient dataset containing all the relevant factors for a sizable time period was necessary. The recency of the dataset was not relevant, since the model is expected to perform the same regardless of when it is applied to. Thus, instead of manually scraping a brand new dataset for the purposes of this project, an existing dataset used by academia was utilized instead. The dataset is called “Stock Market Index Dataset”, and it is from the paper “Stock Market Index Data and indicators for Day Trading as a Binary Classification problem”. It includes a variety of technical factors such as closing price, percent variance, return, momentum, moving average convergence divergence, exponential moving average, return on investment, and relative strength index. There is a data point for each day from April 20th, 2010 to July 12th, 2016 (1601 data points) [6].

4. Implementation and Training

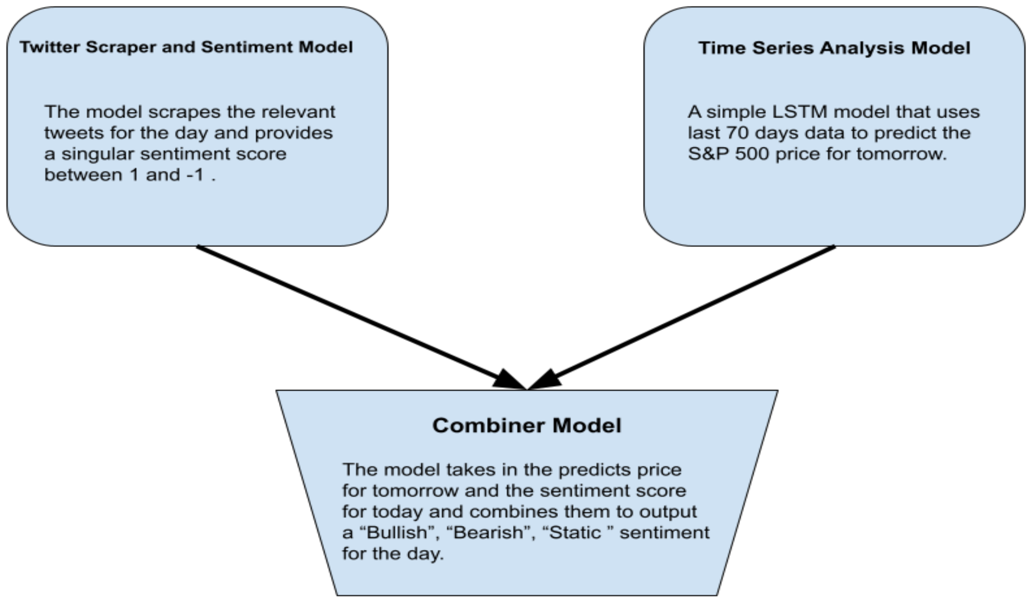

Twitter Scraper and Sentiment Model
The model scrapes the relevant tweets for the day and provides a singular sentiment score between 1 and -1 .
Time Series Analysis Model
A simple LSTM model that uses last 70 days data to predict the S&P 500 price for tomorrow.
Combiner Model
The model takes in the predicts price for tomorrow and the sentiment score for today and combines them to output a "Bullish", "Bearish", "Static " sentiment for the day.

Figure 1: Overall hierarchy of the project

4.1 Implementing the Twitter Sentiment analyzer

As mentioned before, the model will scrape twitter for any given day using the Twitter intelligence tool *twint* [7]. Furthermore, for the current model, the API only scrapes the tweets that include the words *stock* or *Nasdaq* in them, using the variable query "stock OR Nasdaq". Moreover, when all the relevant tweets have been scraped, they are loaded into a .csv file. Once the tweets are loaded successfully into the .csv file they are analyzed using the analyzer.polarity_scores from the VADER sentiment analyzer [8]. The analysis outputs a score between -1 and 1, -1 indicates a highly negative sentiment and 1 corresponds to highly positive sentiment. Finally, the final sentiment score for the day is computed by averaging all sentiment values for the day.

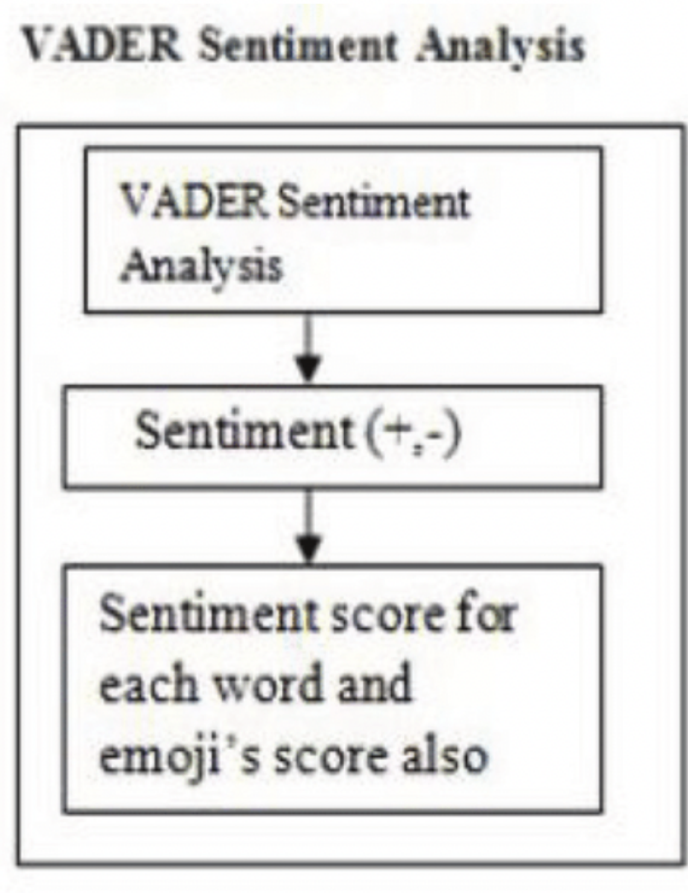


Figure 2: Vader Model [9]

The VADER (Valence Aware Dictionary for Sentiment Reasoning) sentiment analyzer utilizes a lexical mapping of words labeled according to their respective semantic orientation of either being positive or negative. Thus the sentiment score of a text can be obtained by summing up the scores for each word in the tweet. For example, a tweet consisting of words like, “Excited”, “Great”, and “happy” all convey a positive sentiment, furthermore VADER takes into account context words, and grammar such as “Not Great” will have negative sentiment, not a positive one.

4.2 Implementing and training LSTM

One major part of the full model is the price regression LSTM. The implementation of this submodel was kept as basic as possible since the purpose of the research was to explore the implications of combining the two submodels rather than further developing the individual submodels.

The overall model consists of a 4 layer LSTM PyTorch module chained with a fully connected Linear PyTorch module. Every timestep has 26 input dimensions (as referenced in the data section), while there are 40 hidden dimensions and the output dimension is 1 (the closing price). The 40 hidden dimensions were hyperparameter tuned by running multiple trials and comparing various factors such as time to convergence, test set graph, and mean squared error. Also, the time window chosen for the LSTM was 70 days. Similar to the number of hidden parameters, this is a hyperparameter that was chosen through testing multiple values.

For training the model, the Adam optimizer was used, with a learning rate of 0.01. The loss was computed using the PyTorch mean squared error function. Also, the LSTM was trained on the S&P 500 dataset mentioned previously in the report. The full training process took anywhere from 70-90 epochs.

Initially, the model was somewhat overfitting onto the dataset. The closing price prediction for any given day was highly correlated with the closing price of the previous day. The model was most likely ignoring most other values and simply picking a price close to yesterday's every time, since the majority of the time the prices would be close. However, this was not desirable for our use case, since sudden peaks and troughs can be valuable or dangerous and are important to predict.

To combat this issue, the data was preprocessed so that any input time series based on price would be replaced with the first-order difference of that time series. This way, the model does not know the absolute value of any given day and instead tries to predict the change in price.

5. Final Analysis and Results

As mentioned earlier, our project hierarchy is dependent on two upstream models. Firstly, one model is used to scrape numerical data about key stock market indicators which is used to train our LSTM model. And we use a second sentiment analysis model to augment our LSTM model in an attempt to harmonize strictly numerical data with data that is more qualitative in nature. Therefore, we need a third model that can harmonize these two models in the best possible way. One of the key hurdles in combining these models was to first determine how much each of the models should be weighted. For example, the sentiment analysis model initially outputs a result between -1 and 1, this final result needs to be mapped to a better range that relates better to the scale at which the numerical model is functioning at. In order to determine the mapping, a lot of hyper parameter tuning was done, and the final decision was made to map the sentiment analysis output to a range of -50 to 50 rather than -1 to 1. Moving on, another issue came up, where we did not want to use the exact sentiment output of the day to come up with the daily sentiment score. This was done to minimize any unwarranted noise in the sentiment analysis model and also to emulate the real world model that investor sentiment on day *x* is not independent of investor sentiment from the past.

And finally, came the task of choosing what coefficient should be used to map the sentiment analysis data properly with the numerical analysis data. After iteratively tuning these three hyperparameters, the final precision vs actual stock market data curve was found, which is shown in Figure 3 below.

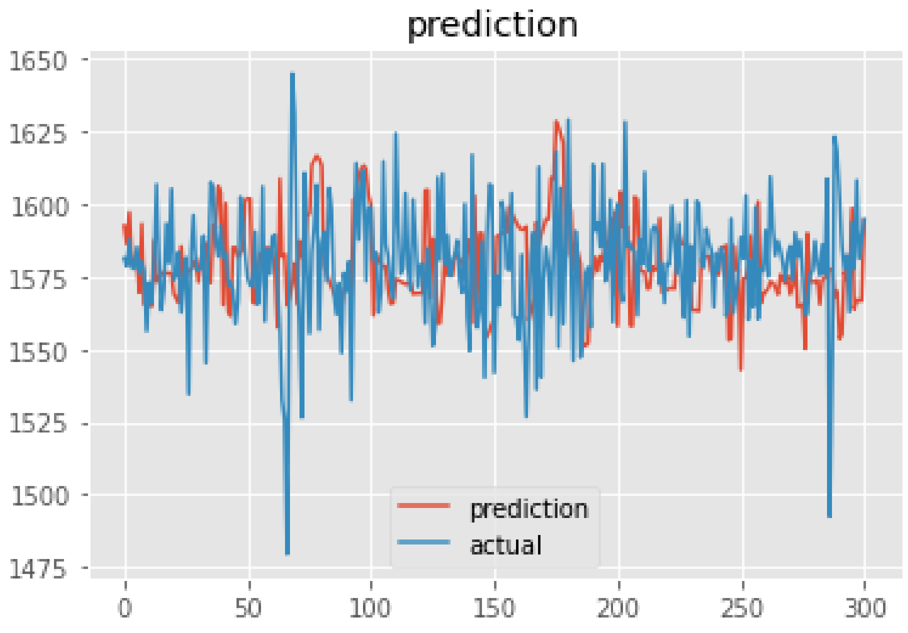


Figure 3: Initial combined model

The model accuracy was found to be 60.79 percent which is a valuable result, considering the scope of our project. The goal of this project was to create something that investors can use to gauge how the market will perform on a higher, classification based granularity as opposed to framing this as a regression problem.

In order to measure the model's accuracy, it was divided into 5 different classes. Namely *very bullish, bullish, static, bearish and very bearish*. The bounds for these classes were chosen as shown in Table 1 below. The percentage bounds are first used to divide up the actual stock market curve into the 5 classes as mentioned and then these preprocessed classes are used to determine whether the predicted outcome fell into this window or not to calculate the accuracy of the model. Given the extremely complex nature of the stock market, the model was able to perform well in term of the scope defined for its use case.

Class Name	Bound
Very Bearish	Less than -2%
Bearish	Between -2% and -0.5%
Static	Between -0.5% and 0.5%
Bullish	Between 2% and 0.5%
Very Bullish	Greater than 2%

Table 1: showing the classification bounds in terms of the percentage change of the stock market

6. Our broader impact and ethical implications statement

Before analyzing the impact and ethical implications of the performed research, it is important to note that most of the work done for this project is highly applied. Thus, though uncertainties still remain, the majority of the impact and implications of the research are clear and comprehensible.

Even in the present day, there are various software and techniques available to assist investors to make profitable purchasing and selling decisions on the stock market. There is some software that is highly inaccessible for retail investors and is only feasible for corporations such as trading firms. One example of such software would be the Bloomberg Terminal, which costs more than

$20,000 USD a year. On the other hand, highly accessible internet applications such as Wealthsimple and Robinhood have become prevalent, leading to a surge in retail investors in recent times. In order to understand the impact and ethical implications of the research, it's important to understand how such an application would impact the different types of investors.

It is crucial to consider the impact of the research on retail investors. Retail investors are often middle-class citizens investing a portion of whatever savings they have in hopes of financial stability. This research could have a major impact on retail investors since all information utilized by the model is easily available to the general populace. Using Twitter data as well as common technical factors means that the application could be used with little to no cost barrier. This could be beneficial for retail investors, as it could become an efficient yet accessible strategy for investing. Though the possible financial benefits are clear, it is equally clear that poor performance of the model could lead to losses for the investor. Therefore, there are two major initiatives that must be taken for the sake of retail investors. First, the model must be further validated and tuned to ensure minimal losses for the retail investors, as significant losses could have a major impact on the lives of said retail investors. Next, it is important to attach sufficient information and warning to the application so that potential investors could educate themselves regarding the risk implicit in using the model.

Though to a lesser extent than when it comes to retail investors, trading firms could be affected by this research as well. Value investing firms are not taken into consideration here as those would be unlikely to use such tools. However, quantitative firms could potentially use tools based on this research. Since these companies often perform high-frequency trading, this model which can provide predictions/classifications at quick time intervals can be fairly valuable. Thus, it is possible for this model to be impactful for some trading firms. However, it is important to note that finance is generally considered a zero-sum game. This means that if such HFT firms make profits utilizing this research, most likely someone else is losing money. Furthermore, since such firms often have the ability to trade quicker than others due to geographical location and high-end technology, they will be able to take advantage of this research more than others (such as retail investors). Thus, this research has the potential to make the titans of finance even stronger than they already are, thus negatively impacting the weak.

7. Conclusion

Overall, it is clear to see that the results of this research project are sufficiently significant. The created model tackles the popular problem of stock market prediction for investment purposes. Though various existing projects with unique takes were explored, this project takes a novel approach by combining two different types of machine learning approaches, with two completely different types of data. By utilizing more classical data such as technical factors, as well as more modern sources of data such as Tweet sentiment analysis, the created model is able to take into account a greater amount of information than most other models. Furthermore, the more open-source nature of the data used for this model means that this technique is highly accessible to retail investors and trading firms alike. With our decently high accuracy rate, it is clear to see that this research can be impactful. Of course, more work can and should be done. Some initial ideas for the next steps include incorporating sentiment scores into LSTM input and up-weighting training examples during peaks or troughs. Combining other news and sentiment sources such as Bloomberg articles, and news websites.

8. References

[1]

“Linear Regression | Implementing Linear Regression from Scratch,” Analytics Vidhya, Jun. 16, 2021. https://www.analyticsvidhya.com/blog/2021/06/getting-started-with-machine-learning%E2%80%8A-%E2%80%8Aimplementing-linear-regression-from-scratch/ (accessed Apr. 14, 2022).

[2]

S. Stock, “concept of stock market exchange or financial technology, polygon...,” iStock, 2020. https://www.istockphoto.com/vector/stockmarketconcept-gm1262967772-369598299 (accessed Apr. 14, 2022).

[3]

O. B. Sezer and A. M. Ozbayoglu, “Algorithmic financial trading with deep convolutional neural networks: Time Series to Image Conversion Approach,” Applied Soft Computing, vol. 70, pp. 525–538, 2018.

[4]

N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “Sentiment analysis based on Deep Learning: A Comparative Study,” Electronics, vol. 9, no. 3, p. 483, 2020.

[5]
R. Bruni, “Stock Market Index Data and indicators for Day Trading as a Binary Classification problem,” Data in Brief, vol. 10, pp. 569–575, Feb. 2017, DOI: 10.1016/j.dib.2016.12.044.

[6]

SaskiaEpr, “Twitter tightens rules on the spread of misinformation,” IT Security Guru, Mar. 02, 2021. https://www.itsecurityguru.org/2021/03/02/twitter-tightens-rules-on-the-spread-of-misinformation/ (accessed Apr. 14, 2022).

[7]
twintproject, “twintproject/twint: An advanced Twitter scraping & OSINT tool written in Python that doesn’t use Twitter’s API, allowing you to scrape a user’s followers, following, Tweets and

*more while evading most API limitations.," GitHub, Mar. 02, 2021.
https://github.com/twintproject/twint (accessed Mar. 16, 2022).*

*[8]
cjhutto, "cj hutto/vaderSentiment: VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media and works well on texts from other domains.," GitHub, Mar. 15, 2021.*

*[9]
U. Soni, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Medium, May 30, 2019.
https://medium.com/@urvisoni/vader-a-parsimonious-rule-based-model-for-sentiment-analysis-of-social-media-text-34380204b96f (accessed Apr. 14, 2022).*

*[10]
https://github.com/cjhutto/vaderSentiment#:~:text=Notifications-,VADER%20Sentiment%20Analysis.,on%20 texts%20from%20other%20 domains. (accessed Mar. 16, 2022).*