Predicting Stock Prices Based on Reddit Post Sentiment

by

Havan Patel

Dr. Piotr Szczurek, Advisor

Masters project submitted in partial fulfillment of the

requirements for the Master of Science in Data Science degree

in the College of Aviation, Science, and Technology of

Lewis University

***Abstract***

**This paper proposes a novel approach for predicting stock prices based on sentiment analysis of Reddit posts. Financial news has been proven to be a valuable source of information for the evaluation of stock market volatility. Leveraging data available on Reddit and the dynamic nature of social media sentiment, we aim to enhance traditional stock price prediction methods. The project integrates historical stock market data and sentiment analysis of Reddit posts mentioning specific stocks. By utilizing natural language processing techniques, we extract sentiment features from Reddit posts and incorporate them into machine learning models for stock price prediction. In this context, this research aims to examine the influence of financial news within the stock price prediction problem, by using the VADER sentiment analysis model to process the comments and get the sentiments as a feature then we use the LSTM model for stock prediction for future days. Through this approach, we aim to provide valuable insights for people who are investors and traders by capturing the collective sentiment expressed on social media platforms. Though these are just predictions of stocks for learning and educational purposes and should not be used for actual trading of stocks.**

***Index Terms*— LSTM, Roberta, SARIMAX, Vader, sentiment analysis, stock price prediction**

# INTRODUCTION

Nowadays, the age of the Internet has changed the way people express their views, and opinions. All types of apps are being created to allow people to trade stocks instantly and on a device that fits in their pocket. This has led to the number of people talking about the stock market rising, especially on social media. Millions of people are using social network sites like Facebook, Twitter, Google, etc. to express their emotions, opinions and share views about their daily lives. For example, GameStop trading frenzy, mostly driven by discussion on the subreddit’s wallstreetbets, which highlighted the potential impact of social media sentiment on the stock prices. [1] Through online communities, we get an interactive media where consumers inform and influence others through forums. Social media is generating a large volume of sentiment rich data in the form of tweets, status updates, blog posts, comments, reviews, etc. Whenever someone wants to buy a product or any service, then they will look up the reviews online before purchasing. The objective of this project is to develop a predictive model for weekly stock prices using sentiment analysis of Reddit posts. By analyzing the sentiment of discussions related to specific stocks on Reddit, we aim to identify patterns and correlations that can assist in predicting future stock prices. We also aim to propose a model that could potentially offer insights into stock price movements based on public perception and discussion trends. This paper will contain three primary parts: implementation of reddit post sentiment analysis by using a model like Vader, stock price prediction using LSTM model, and evaluation and analysis of the model and predicted outputs.[2]

The motivation behind this project stems from the increasing influence of social media on financial markets. Social media platforms like reddit, twitter, and others have become a huge factor of forums for discussion on various topics, and one of them includes stock market and investments. This project seeks to explore this relationship further by systematically analyzing Reddit posts to predict stock price movements. By quantifying the sentiment of these discussions and combining it with traditional stock market data, we can potentially improve our ability to predict stock prices accurately. This project aims to leverage the power of sentiment analysis and machine learning algorithms to enhance stock price predictions. If you are using such methods to trade stocks, then this is highly risky and should only be used for learning and educational purposes.

# Discussion Of Related Work

## Sentiment Analysis Model: Vader

## Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources through Natural Language Processing (NLP). Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative" or "neutral". It's also referred to as subjectivity analysis, opinion mining, and appraisal extraction [3]. There are two main ways to automate the sentiment analysis process: dictionary-based methods and machine learning application-based methods. Using sentiment analysis, it is possible to perceive the intentions of companies and investors in real time, something extremely important for stock exchange decision-making. There are many different sentiment analysis models and transformers, but we will be using the VADER and Roberta model for text sentiment analysis. However, we are not going in depth for Roberta model as we are only using it to compare the results with the Vader model. VADER sentiment analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text [3]. For example, words like ‘joy’, ‘happy’, ‘exciting’ all convey a positive sentiment. We can also utilize VADER for determining the context of words in a sentence, such as ‘I don’t like it’ as a negative sentiment. It uses a list of lexical features (e.g. words) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment. Vader sentiment returns the probability of a given input sentence to be Positive, negative, and neutral. In fig 1 we can see examples of comments where a comment has 4 types of scores given to a comment. The first comment has more of a neutral or negative score given. We can see from reading the comment that the person wanted to buy the stock, but he missed his chance. That to us seems more of a neutral sentence than a positive one. VADER uses a sentiment lexicon, which is a list of words and their associated sentiment scores. The lexicon contains over 7,500 words, each with a score ranging from -4 (extremely negative) to +4 (extremely positive). In addition to the lexicon, VADER employs a set of rules that consider the grammatical and syntactical patterns in text to determine the overall sentiment intensity. [4]

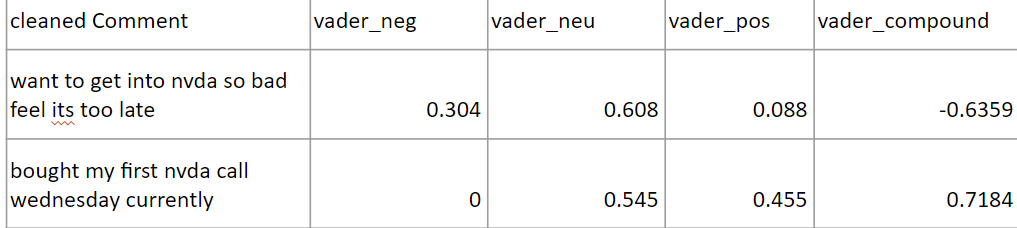


Fig.1 Vader scores of a comment

## LSTM

There are many cutting-edge models, most of which only require you to write only a few lines of code to train and predict your data. Linear Regression, Random Forest, or Moving Average could be potential options, but we will be utilizing the LSTM model. Long Short-Term Memory (LSTM): This is a type of recurrent neural network (RNN) that is particularly useful for time series prediction.[5] We have also used the SARIMAX model as well, but we will not go into details here. We will use it to compare it to the LSTM model. LSTM is a more advanced model that can learn long-term dependencies in the data and is particularly useful for making predictions over longer time horizons. Reason for using it because firstly LSTM is one of many types of Recurrent Neural Network RNN, it’s also capable of catching data from past stages and using it for future predictions [6]. RNN can’t store long time memory, so the use of the LSTM based on memory line proved to be very useful in forecasting cases with long time data. This information is carefully regulated by three structures called gates, which will control what information will be thrown away from the cell, what information will be inserted on the cell and what will be the cell output. With this complex structure of memory cells, the LSTMs can assimilate the structure of the data dynamically over a time span with high prediction ability [7]. In a LSTM the memorization of earlier stages can be performed through gates with a long memory line incorporated. An LSTM has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward. The differences are the operations within the LSTM’s cells [8]. LSTM stands out from other recurrent neural networks because of its ability to learn long-term dependencies. This is accomplished by using a memory cell that allows information to be stored and retrieved selectively over a long period of time. In fig. 2 the memory cell is controlled by three gates: the input gate, forget gate, and output gate, which are used to control the flow of information into and out of the memory cell. The input gate determines how much new information is added to the memory cell, while the forget gate determines how much of the previous memory should be forgotten. The output gate determines how much of the current memory state should be used as the output to the next layer in the network [9].

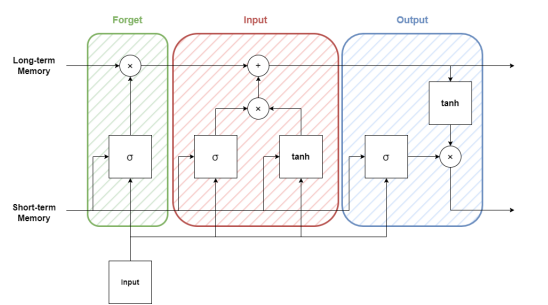


Fig. 2 LSTM Architecture

The approach presented in this work separates the model into two stages: sentiment analysis and price prediction. Both stages have different requirements and will end up working together. The sentiment analysis stage aims to process the posts of the companies studied and define the position of investors during the period studied, based on the sentiment obtained. Sentiment analysis tasks generally require some kind of data pre-processing in order to accomplish good results. The prediction of prices stage, which is supported by the result of the previous stage and by the historical series of the prices of assets in the period studied, has the objective of developing a model that is intelligent enough to generalize the prediction to all studied assets, which will represent a portion of the universe of assets on a stock exchange. The price prediction task requires a robust model to be developed, which has the necessity of validation with data and parameter tuning.

# Methodology

For this section we are going to break the methodology into two parts: first will be the methodology with the sentiment analysis and then with the LSTM model. Then we will take a look at the results for Vader and LSTM in a different section. All the code written for this project is in python utilizing the VsCode editor. ***VADER***

*1).* *Data collection*

So, we have used the praw library to fetch the reddit post. allows for simple access to Reddit's API. The problem we ran into is that using the pushshift library we were getting an authentication issue, and we contacted the reddit support to give access to use that endpoint, but we did not hear back, so we continued with the project using praw library. The pushshift.io Reddit API was designed to help provide enhanced functionality and search capabilities for searching Reddit comments and submissions. This library has been deprecated meaning that it is not supported much. With praw you can really customize the data extraction based on an upvote of comments, number of likes, post\_fliars but that can restrict the amount of data you can fetch therefore we have not used it. So, we fetched the data from reddit on the following subreddits wallstreetbets, stocks, StockMarket, Investing, pennystocks, RobinHood and we keep track of how many stocks are being mentioned in each post so we can have a tracker of most mentioned stocks. We then convert this data into a dataframe so we can save it into a csv file so we can work with that for reusability purposes. As seen in table 1, we have features like ticker, comment, author, date, subreddit in the dataframe.

Table 1. Vader Dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Example** | **Description** |
| subreddit | String | Stocks | Which subreddit the comment came from |
| ticker | String | NVDA | stock ticker |
| comment | String | Buy NVDA stocks now | comment by user |
| date | Date | 2024-01-01 | Date of comment made |

*2).* *Pre-processing Data*

For this section we have cleaned up our data and mostly on the comments. For doing sentiment analysis and getting accurate sentiment we need to clean the comments and remove unnecessary words. We are doing the following to clean the texts: removing URLs, punctuation though it can make or break with sentiment, removing numbers, removing stop words that are not helpful. We are doing this data cleaning by tokenizing the text into words. Then we are applying the entire text to Vader and Roberta models to give us sentiments by generating extra features like positive, negative, neutral and compound scores as seen in fig 1. There are two ways to go about this to either do sentiment analysis on the entire text or word by word of the text. Both had their advantages and disadvantages, but we went with the entire text because we wanted to capture the sentiment/emotion of the entire text.

***LSTM***

In Fig. 3 we have shown the pipeline that we will try to implement for LSTM. Some sections we will describe in the later sections.

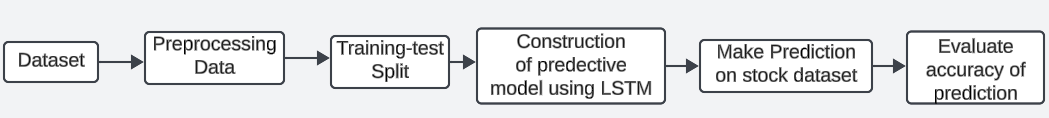


Fig. 3 LSTM Pipeline

*1). Data collection*

In order to get the stock data, we utilized yahoo finance specifically using the pandas\_datareader library. Pandas\_datareader library is what allows us to access yahoo API to fetch stock prices. We picked one of the most mentioned stocks and we tried to get the data for it. We are getting data from the year 2010 to the current date. We used NVDA stock where we would get a shape of (3593, 6). This is 14 years' worth of data, and it is plenty of data for testing and training our model. Using that library we are getting the following features ‘Open, High, Low, Close, Adj Close, Volume, Date’ as seen in Fig 4. From this we mostly care for the closing price and date feature.

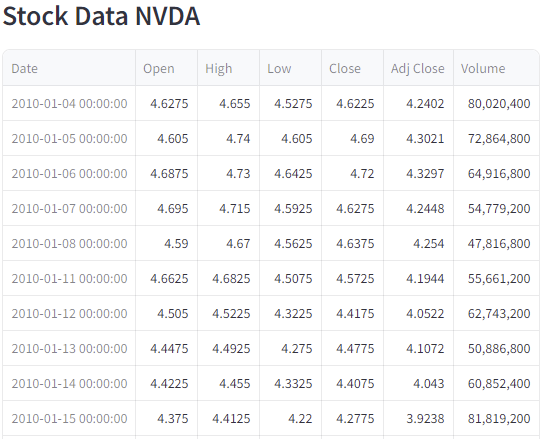
~~~~

Fig. 4 Stock information pulled from Yahoo Finance.

*2). PROCESSING DATA AND APPLYING THE MODELS*

After getting the data we are going to extract the closing price data from the dataframe and then we scale the data between -1 to 1. We applied feature scaling using the MinMaxScaler to ensure that all data falls within a specific range, which is essential for LSTM models to converge efficiently. Since the stock price’s numeric values range from 0 to 800+, this can be very difficult for our model to be accurate. Then we split out the dataset into 85% training and 15% testing. We split the dataset into 2, based on the time step we take. The first dataset dataX takes the values as its input and the second dataset dataY takes the values as output. Basically, it creates a dataset matrix from the above dataset mentioned in fig. 4. The input sequences will consist of historical stock prices, and the corresponding target values will be the stock prices at the next time step. By doing so, we enable the LSTM to learn the temporal patterns in the data and make predictions accordingly. Then we reshape the training and testing data to add the sample dimension, ensuring that the input data has the expected 3-dimensional shape as it is required by the LSTM model.

*3). TRAINING ARCHITECTURE OF THE LSTM MODEL*

The LSTM model architecture plays a crucial role in capturing and learning temporal patterns from sequential data. We’ll design a stacked LSTM model with dropout regularization to prevent overfitting. Stacking multiple LSTM layers enables the model to learn complex relationships in the data. As shown in table 2, we used 4 layers of LSTM with 1 as input layer, 1 as hidden layer and 2 as output layer as Dense. In the first 2 layers we took 256 neurons and 128 for the second. Also 25 and 1 for output layers. We also added a dropout layer to prevent overfitting. Once the architecture is defined, we need to compile the model by specifying the optimizer and the loss function. For time-series forecasting tasks like stock price prediction, we’ll use the mean squared error (RMSE) loss, which measures the difference between predicted and actual stock prices. We’ll also use the ‘Adam’ optimizer, which is an efficient optimization algorithm widely used in deep learning.

Table 2. LSTM Model Summary

|  |  |  |
| --- | --- | --- |
| **Layer** | **Output Shape** | **Param #** |
| lstm (LSTM) | (None, 60, 256) | 264,192 |
| dropout (Dropout) | (None, 60, 256) | 0 |
| lstm (LSTM) | (None, 128) | 197,120 |
| dropout (Dropout) | (None, 128) | 0 |
| dense (Dense) | (None, 25) | 3,225 |
| dense (Dense) | (None, 1) | 26 |

# Results

The result section is divided into two parts. The first part presents a summary of the sentiment analysis results. The second part describes the evaluation and results of the LSTM model. We also want to mention that we also utilized Roberta model for sentiment analysis to compare VADER and SARIMAX model for stock price prediction for comparing LSTM model.

*1). Vader*

While scraping the posts from reddit, we have kept a tracker of a stock ticker to help us identify for which stock this comment is for. In fig. 5 we can see the most mentioned stock in the square plot with the number of posts. We can see NVDA is the most mentioned stock. This makes sense because NVDA was one of the hottest stocks this year. We can see that we have NVDA mentioned in 457 and rightfully so because we saw in the news that they showed off their earnings and new products which gave a boost to their stock prices.

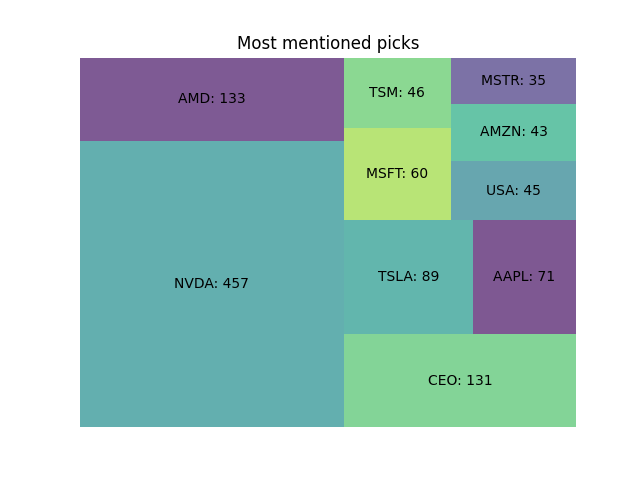


Fig. 5 Most mentioned stocks in the posts

As mentioned in the previous section that sentiment we are given scores between values of -1 to 1. The negative score means that the post received a bad reaction, a score of 0 represents that the post is neutral about the comment, and a score of positive means the post about the stock is optimistic. As seen in fig 6. We have the scores of the post's sentiments about a particular stock. So here the red line depicts a bear market meaning people are not buying the stocks. The light green represents neutrality of stock where people didn’t really influence the stock price. The dark green line represents the bullish where the stock prices are growing. So, let's take a look at MSFT stock as an example. We see on the left plot for the vader model that it is analyzing this stock is mostly a bullish stock because the neutrality and positive score is high compared to the negative score. Comparing MSFT stock with the roberta model it is detected that this stock is mostly a bearish stock where it is not recommended to buy the stock. The negative score in Vader and Roberta model is almost the opposite. Since our data is not labeled, it is hard to decide which model is accurate and which gives us the best results. One way to compare the accuracy of the model is to look at the correlation of both models and see what difference we can find. This will give us a basic idea of how well our model did.

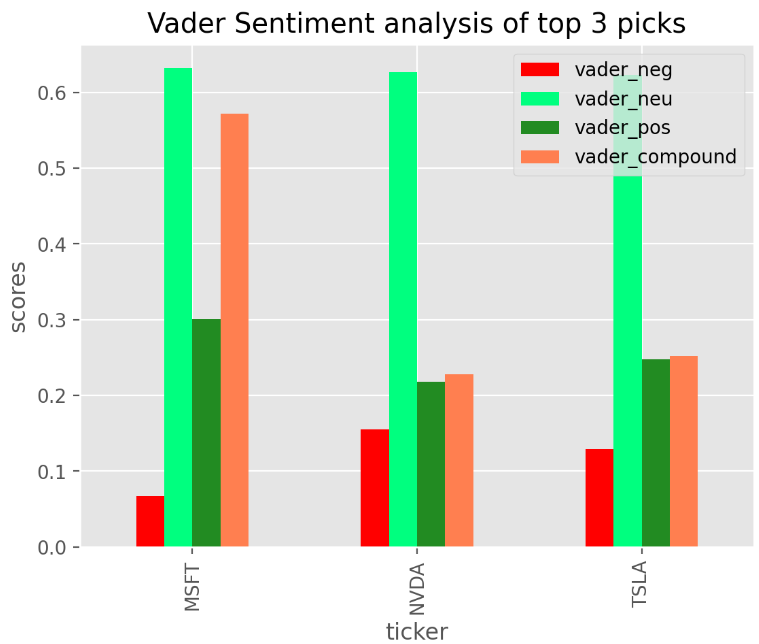
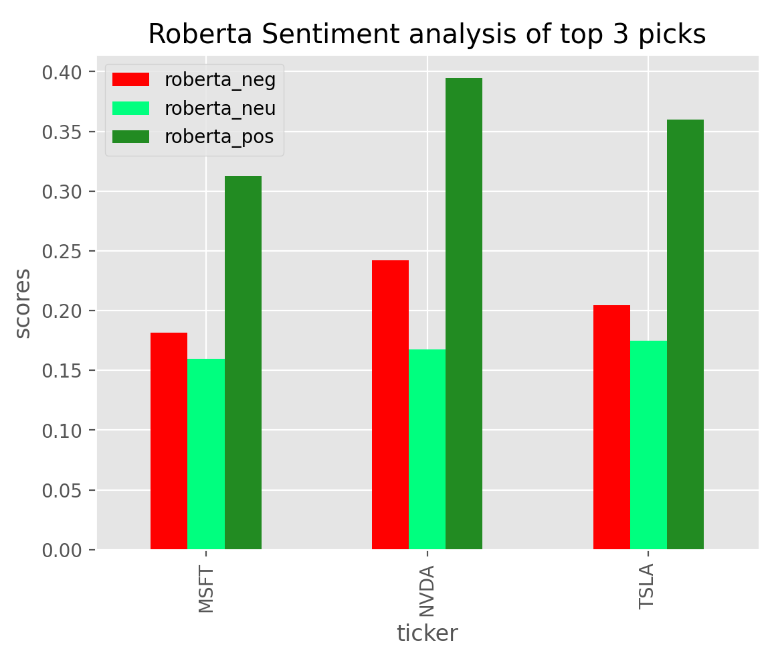


Fig. 6 Left plot is of Vader sentiment scores and on the right, we have Roberta sentiment scores of the stocks.

Let’s compare both models' accuracy. In fig 7. we see that the negative sentiment score for vader on the top left on the first row is to the left compared to roberta on the fourth row. When manually looking at the scores of the posts this makes sense in the vader model to shift to left then what the scores were generated with Roberta. For a neutral score on the second row in vader it is in the middle unlike roberta on the fifth row. For the positive score on the third row, it is really hard to determine in Vader because it is in between negative and neutral but roberta model seems to have better positive score than vader model on row six. Looking at this, we think vader model seems to be more accurate and upon looking at the scores of the text personally vader scores did make more sense than the scores that were generated by Roberta model.

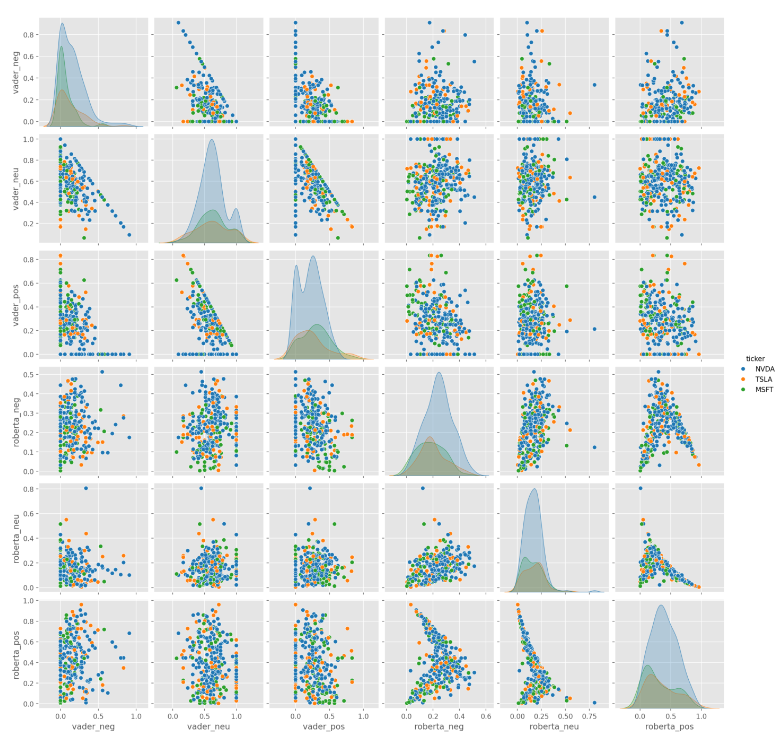
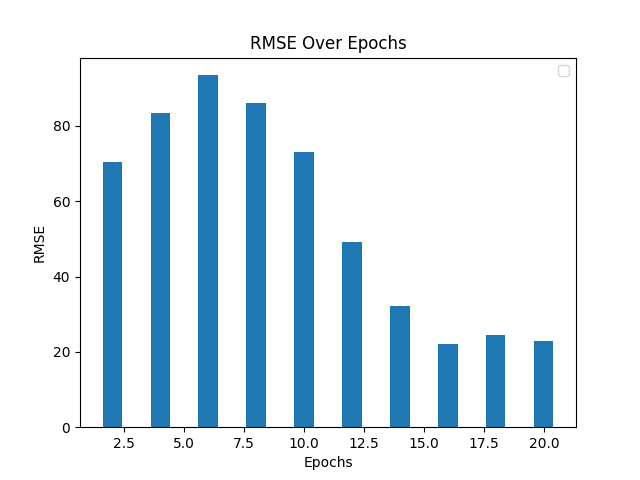
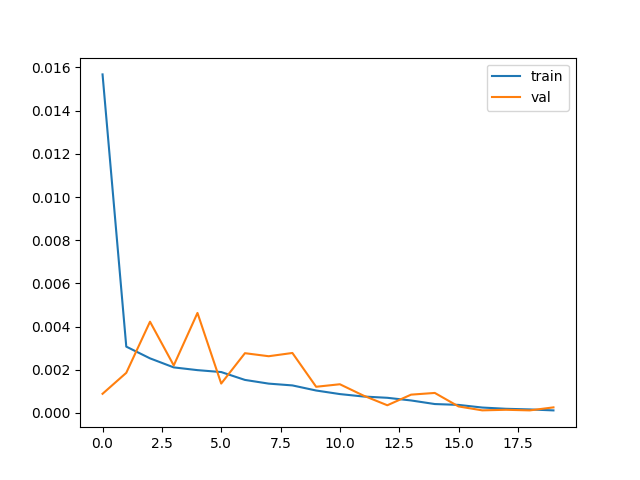


Fig 7. Both models accuracy determination

*1). LSTM*

The training process is carefully monitored, and the training history is stored for further analysis, such as plotting the loss during training. With the training and validation loss data available, we can now plot the losses over the epochs. This will give us insights into how well the model is learning from the training data and whether there is any indication of overfitting or underfitting. We can see in fig 8. that the model does seem to learn up to 15 epochs then the train and validation data converge at around 17 epochs. We can conclude the model is performing well and we can use around 17 epochs for our learning rate. Let’s also look at the RMSE which is a commonly used metric to evaluate the performance of LSTM models for stock price prediction tasks. It provides a measure of the average magnitude of the errors between the predicted stock prices and the actual stock prices. The RMSE is calculated by taking the square root of the mean squared differences between the predicted and actual values. It gives more weight to larger errors, making it sensitive to outliers or large deviations in the predictions. We can see in fig. 9 the model continues to learn and that the error for the training data is decreasing, which also indicates that the model is doing well. We are calculating the RMSE every 2 epochs to evaluate the model as seen in the plot below.

 Fig 8. Validation and Training Loss Fig 9. RMSE over epochs for NVDA

For time series we need a lookback variable and that variable in context of time series forecasting with LSTM models is used to define the number of previous time steps or lags to use as input to the model for predicting the next time step. In other words, the lookback variable determines how much of the past data or sequence the LSTM model should consider when making a prediction for the next time step. In this case we chose to look back for 60 days (about 2 months)' worth of data to predict the next day's closing price. To determine what the lookback should be; we utilized the moving average (MA) of the stock data for every 60 days. In fi. 10 we can see 3 different moving averages for 60 days, 100 days (about 3 and a half months) and 250 days (about 8 months). To calculate the moving average, we need to take the meaning of the closing price for the particular day you want to get the moving average. We can see that the MA for 60 days in the red line is very close to the actual data compared to other MA days. Instead of using the raw stock prices directly, we can calculate moving averages of different window sizes and use these moving average values as the input sequences for the lookback variable

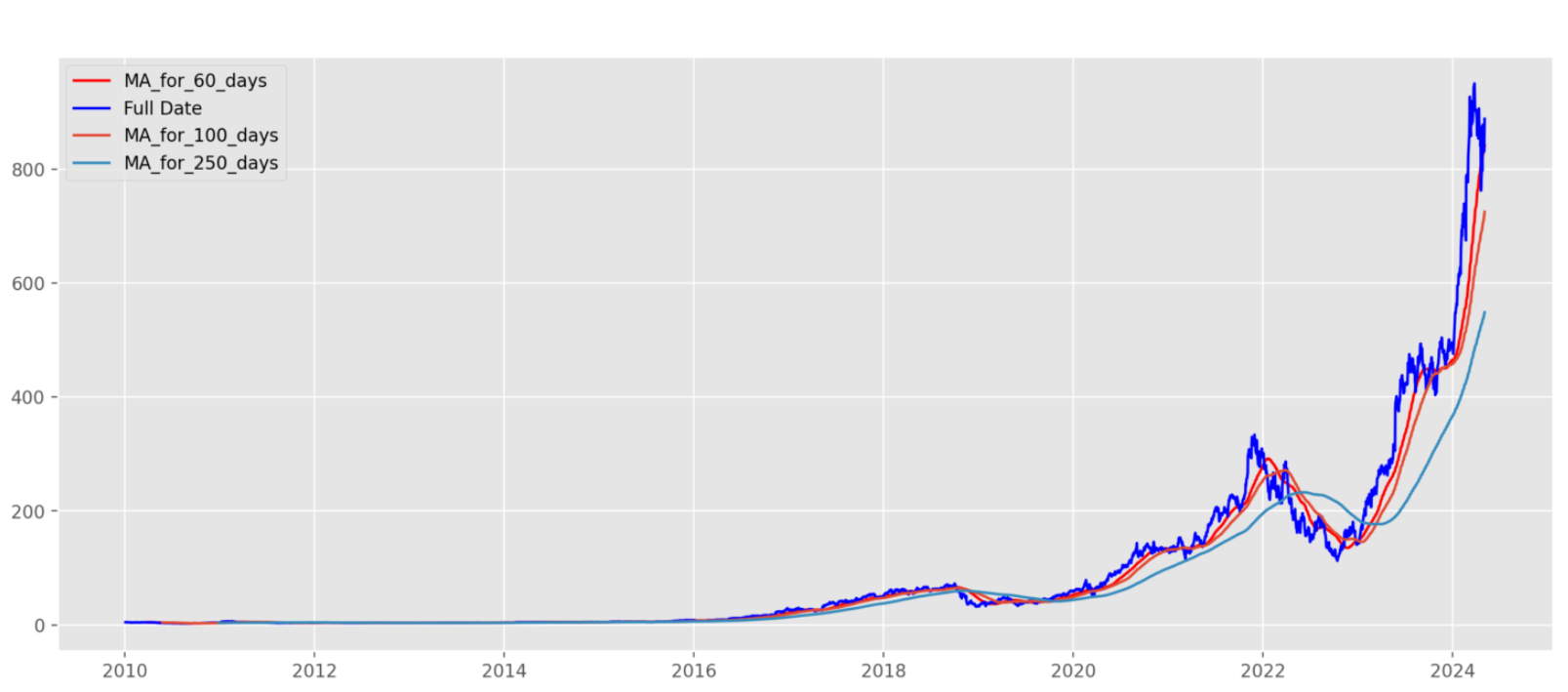


Fig 10. NVDA Stock’s moving Average for 60, 100, 250 days

Now let’s look at how our model predicted the stock's prices. In fig. 11 we can see the entire history of the NVDA stock in the bottom of the graph but for visibility we have zoomed in the graph to see the predictions closely. In blue we have the data the model was using for training. Then the red line is the actual stock data used for testing and the line in green is our actual prediction which is very close to the red line. We can see it is close to the actual price and it is performing very well.

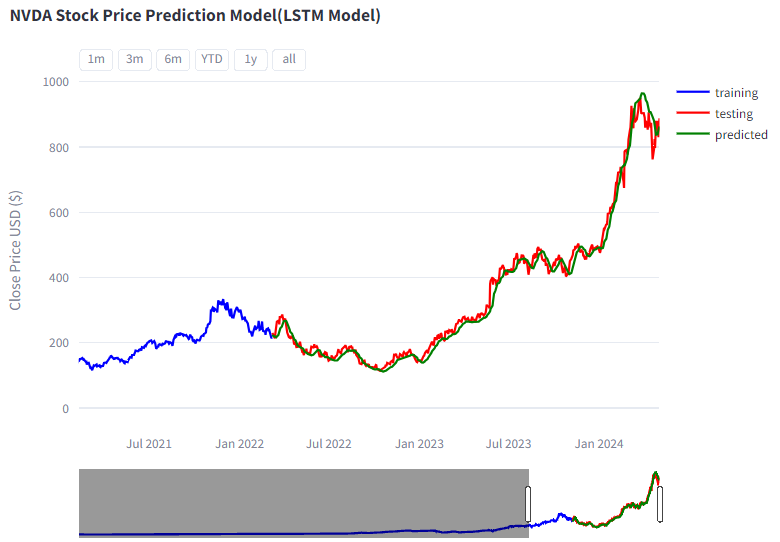


Fig 11. LSTM Prediction

Here in fig. 12, we see the prediction of the price for the next 5 days based on the test data learning. It seems to be accurate based on the trend of the price. Of Course, there is no way to certainly tell if the price will be the same as predicted when the stock market closes. But based on the past data provided, this seems to be accurate.

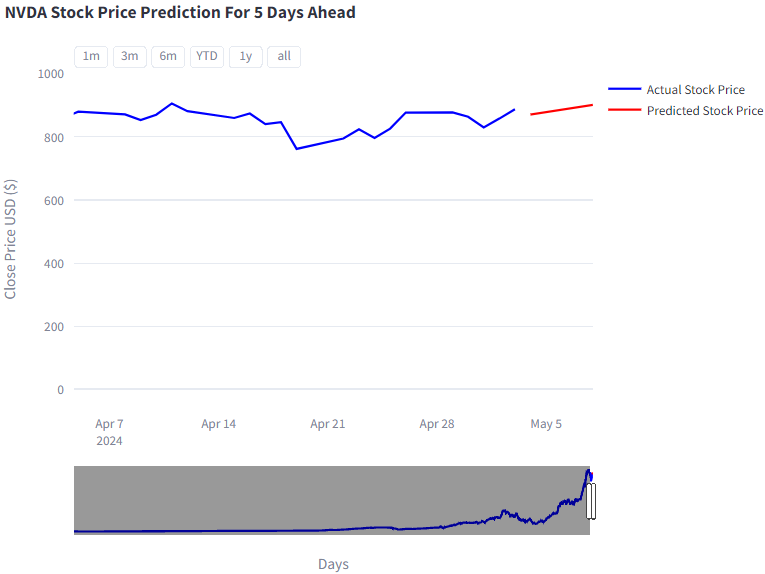


Fig 12. LSTM Prediction for future 5 days

Then we used the SARIMAX model to predict the future prices. We are going to briefly explain this model to compare it with the LSTM model. There are a couple of exceptions to using the SARIMAX model. For example, you need to capture if your data is stationary or not. If it is not, then you need to make your data stationery and there are a couple of different ways to check that. To check if a time series is stationary, you can use the Augmented Dickey-Fuller (ADF) test or the Kwiatkowski-Phillips-Schmidt- Shin (KPSS) test.[11] These tests are commonly used to determine if the time series is stationary or not. [10] We utilized ADF, and we printed the results, including the test statistics, p-values, and critical values. By analyzing the results of these tests, you can determine if the time series is stationary or not. If the p-value is less than 0.05 for the ADF test you can conclude that the time series is stationary. If the p-value is greater than or equal to 0.05 for the ADF test the time series is considered non-stationary. You can see in fig. 13 that data is not stationary based on the p value on the original data, since it's bigger than 0.05.

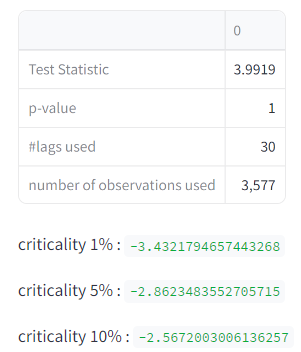
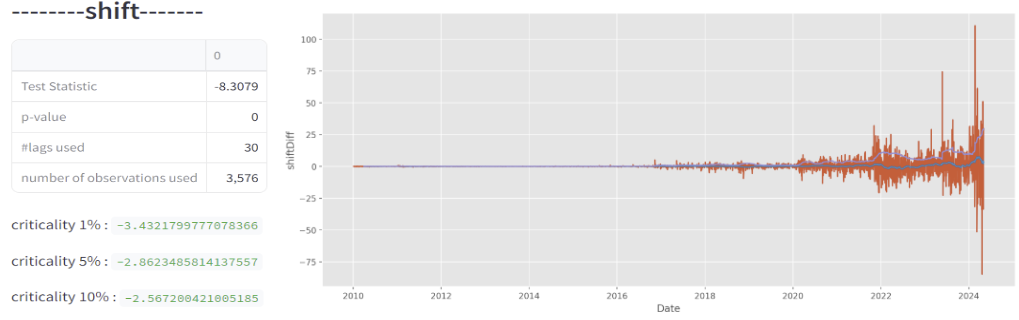


Fig 13. Stationary check of actual data

To make the data stationery you can use a couple of different methods like shifting data then you take difference of the original and shifted data. Differencing involves computing the differences between consecutive observations. The order of differencing (d) in ARIMA refers to the number of times the data needs to be different to achieve stationarity. Another is Log transformation where if the time series exhibits exponential growth or decay, taking the natural logarithm of the data can help stabilize the variance and make the data stationary. This transformation is often applied to financial or economic time series data, where the variance tends to increase over time.[12] There are more methods which we have displayed in the web app. From there you can use the methods that give you the best less p-score less than 0.05. In fig. 14 we see that the p-value is 0 after shifting the data so we can use ARIMA/SARIMAX models because the data is stationary now.

 Fig. 14 Making data stationary by using the shift method

Then comes that hard part to determine the p, d, q values which are the hyper parameters for the model. There are a couple ways to do this. You can use the Auto Arima model to set the min and max values of p, d, q values to give you the best hyper parameter values to use to train your data. Another way which I preferred is to make your own method to give you the best p, d, q values. I used the ARIMA model and made predictions on the entire data set with different combos between (0-8) for p, d, q. Then I found the mean squared error for all the combos and the one that gave me least RMSE; is the p, d, q value used for my SARIMAX model. Third method is to use the Partial Autocorrelation (PACF) plot and Autocorrelation (ACF) to find the p and q values. However, this method is very difficult to rely on and we were not getting good results, so we settled for the second method we suggested, which is to write our own method to give you the best hyper parameters.

After going through the whole ordeal, we can use the model for prediction, and based on the hyper parameters we selected we get the following predicted values which are shown in green line in fig. 15. Then the line in orange is the future 60 days of prediction which seems to be pretty good given the trend of the stock price. There is no way to know for sure whether the future prices will be that high or low. It could lose all its value, or it could skyrocket, or it might not fluctuate much from previous days. If we compare fig. 15 with LSTM fig. 11 and 12 then SARIMAX seems to predict the prices very close to the actual price. Also, the future prediction is very different in that SARIMAX prediction seems to go down then goes up since it is capturing the seasonality of the previous data. As we see that based on the hyper parameters, we select our model and will learn based on that and make predictions.

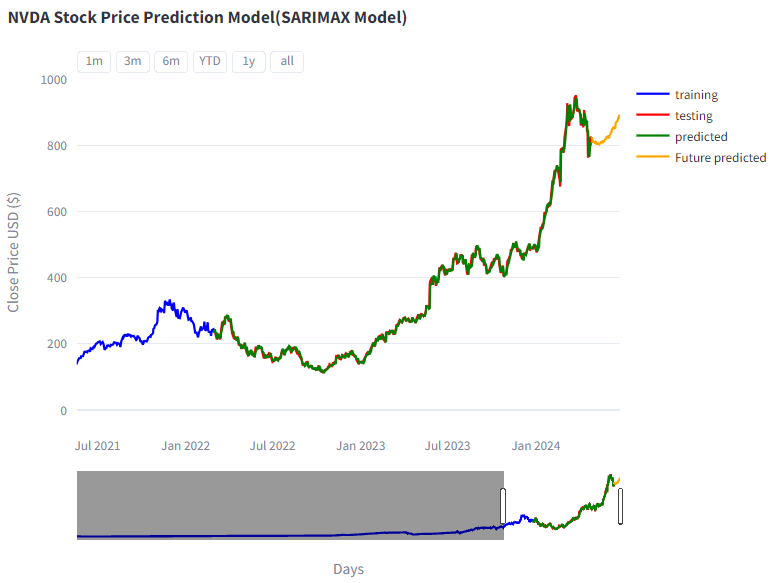


Fig. 15 SARIMAX prediction of NVDA stock

# Problems

Data availability is a significant problem for stock price prediction because financial data is often difficult to obtain, and there are limitations on how much data can be accessed. The availability of data can affect the accuracy and robustness of the models used for stock price prediction. [13] This is true for sentiment analysis because we already face a lot of challenges getting historical data as Reddit has deprecated the precious libraries where you can get past information. We tried using twitter to get information about stock related tweets, but we were not able to due to security reasons and simply they have also changed a lot of things after twitter changes to X. Another issue we faced is cleaning the posts related to comments because this step is very essential to getting the sentiment scores. If your data is not cleaned properly then your sentiment scores can be skewed and end up giving incorrect output. This step needs to be highly customized for the project you are working on. You need to make sure that we don’t end up removing words from text that are important for analyzing the sentiment score. Doing measures or getting the accuracy of the data is very challenging for Vader as our data is not labeled with the polarity scores so we can't generate any metrics for it.

# Ethical Considerations

Models like VADER and LSTM are trained on historical data, which may contain inherent biases or reflect past discriminatory practices. If these biases are not addressed, the models could perpetuate or amplify unfair treatment based on different factors. Specifically, sentiment analysis to do your prediction is not always the best model to use for applications like stock prediction. Analyzing the sentiment of someone’s post is highly flawed because that’s their belief and we don’t know their true intentions. People can lie in their posts so relying on this is very wrong. This could lead to unequal opportunities or unfair advantages for certain groups in the stock market. Additionally, the use of alternative data sources like social media posts or news articles for sentiment analysis could introduce biases or privacy concerns if not handled responsibly. One key ethical consideration is the use of insider information or non-public data. Accessing or utilizing such information for stock market predictions can lead to unfair advantages for certain individuals or organizations.[14]

# Conclusion

Predicting the stock market is a very ambitious and complicated task, mostly because of the many different factors that affect a stock price that simply cannot be predicted with high accuracy. This paper presents a strategy to aggregate one of these major factors: the post associated with the company. We proposed the sentiment analysis of posts collected from Reddit using the VADER framework and used the results as one of the features, along with the historical data of the stocks, of a stock price prediction model based on the LSTM architecture. We also used Roberta mode to compare it with Vader in sentiment analysis and SARIMAX to compare it with the LSTM model. Different models can give you different results as discussed in the paper. Which method you pick really depends on your use case, how your data is and which model gives you best accuracy. These models are very valid models to use if you are trying to build a service that predicts stock prices. [12] Of Course this can use a lot of improvements like using more data for sentiment analysis and refining the data which can even further improve the accuracy and efficiency of your model. We could also focus on different trading strategies like day trading, in which the investor gives more value to short-term predictions. Using a more realistic scenario would also open the possibility of unbiased comparison of the model with state-of-the-art technology.

References

1. Carter, Caymen. “Stock Price Prediction Using Sentiment Analysis and LSTM.” SUNY Open Access Repository (SOAR), 1 May 2022, [soar.suny.edu/handle/20.500.12648/7334](http://soar.suny.edu/handle/20.500.12648/7334).
2. Burgess, Michael; Javed, Faizan; Okpara, Nnenna; and Robinson, Chance (2022) "Stock Forecasts with LSTM and Web Sentiment," SMU Data Science Review: Vol. 6: No. 2, Article 10.
3. Kharde, Vishal. A., and Prof. Sheetal. Sonawane. “Sentiment Analysis of Twitter Data: A Survey of Techniques.” arXiv.Org, 22 Apr. 2016, [arxiv.org/abs/1601.06971](http://arxiv.org/abs/1601.06971).
4. Hutto, C., and E. Gilbert. “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text”. Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, May 2014, pp. 216-25, doi:10.1609/icwsm.v8i1.14550.
5. Patterson J., 2017. Deep Learning: A Practitioner’s Approach, O’Reilly Media.
6. Lamberti, Alessandro. “Sentiment Analysis with Vader and Python.” Medium, Artificialis, 14 Mar. 2022, medium.com/artificialis/sentiment-analysis-with-vader-and-python-5b7ac4f3b13b#:~:text=VADER %20relies%20on%20a%20dictionary,all%20convey%20a%20positive%20sentiment.
7. Heiden, Alexandre, and Rafael Stubs Parpinelli. Applying LSTM for Stock Price Prediction with Sentiment Analysis, sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC\_2021\_paper\_45.pdf. Accessed 31 Mar. 2024.
8. Phi, Michael. “Illustrated Guide to LSTM’s and GRU’s: A Step by Step Explanation.” Medium, Towards Data Science, 28 June 2020, towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a- step-by-step-explanation-44e9eb85bf21.
9. Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: Neural computation 9.8 (1997), pp. 1735–1780.
10. Artley, Brendan. “Time Series Forecasting with Arima , Sarima and SARIMAX.” Medium, Towards Data Science, 27 June 2022, [towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6](http://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6)
11. Xiao, Ruochen, et al. “Predict Stock Prices with Arima and LSTM.” arXiv.Org, 31 Aug. 2022, arxiv.org/abs/2209.02407.
12. Shweta. “Introduction to Time Series Forecasting - Part 2 (Arima Models).” *Medium*, Towards Data Science, 30 July 2021, [towardsdatascience.com/introduction-to-time-series-forecasting-part-2-arima-models-9f47bf0f476b](http://towardsdatascience.com/introduction-to-time-series-forecasting-part-2-arima-models-9f47bf0f476b).
13. Avci, Rasim. “Challenges with Stock Price Prediction.” Medium, Medium, 27 Mar. 2023, [medium.com/@rasim.avci/challenges-with-stock-price-prediction-17e151bd79a7#:~:text=Data%20availability%20is%20a%20significant,used%20for%20stock%20price%20prediction](mailto:medium.com/@rasim.avci/challenges-with-stock-price-prediction-17e151bd79a7#:~:text=Data%20availability%20is%20a%20significant,used%20for%20stock%20price%20prediction).
14. Analytics, Emerging India. “CAN Data Science Predict the Stock Market.” Medium, Medium, 13 Feb. 2024, medium.com/@analyticsemergingindia/can-data-science-predict-the-stock-market-de3748a86026#:~:text=One%20key%20ethical%20consideration%20is,for%20certain%20individuals%20or%20organizations.