# Stock Price Prediction Using Sentiment Analysis

Nir Nandu[1], Taibaz Pathan[2], Rahul Shukla[3], Kanhaiya Singh[4], Prof. Payal Varangaonkar[5]
[1,2,3,4] *Student Rajiv Gandhi Institute of Technology, Mumbai*
[5]*Faculty Rajiv Gandhi Institute of Technology, Mumbai*

**Machine learning is THE future. As our hardware systems inch ever so forward so does our capability in deploying ever increasingly more complex machine learning algorithms. With the failure of the efficient market hypothesis and more advent of more powerful processors stock market prediction has become a viable solution. By taking current news sentiments and combining them with technical analysis of price factors, this research aims to predict trends in future prices on an intraday basis. It is human nature to be more focused on the negative parts of a dialogue. So, the sentiment analyzer was made to be more sensitive towards negative news. The predicted values showed a certain trend in which they were averse to rising too quickly. We think this is because the model was made to be more prone to negative things. This coupled with the more negative focused sentiment analyzer made it such that the model took a rather conservative or even pessimistic approach in its prediction. It was found that considering the news polarity drastically reduces the MSE values to below 1. This just shows us how much the general public relies on the news headlines to navigate the stock market and that just technical or fundamental analysis isn't enough for accurate predictions.**

## I. INTRODUCTION

The stock market is a treacherous minefield to navigate at the best of days and can rise or fall at a moment's notice. With a past record of following no apparent trend and being influenced by myriad of factors that are impossible to predict, the stock market is fatal for those who are looking to earn a free quick buck. Using machine learning for stock market prediction greatly reduces the workload and risks associated with this. The program catches onto the general public consensus regarding the stock and uses this along with real time stock data to predict the future.

The stock market prediction is of two types – fundamental and technical. Both have their own use cases and are at odds with the efficient – market hypothesis which states that the market prices are unpredictable. This research goes on as a mix of both areas by combing the technical data and public sentiment and using them in certain mathematical algorithms. Ever since the advent of neural networks it has become possible for us to do even more complex calculations allowing us to identify trends easily. Even the various indicators of stock market analysis have been combined with the models to give an even more accurate result. This project employs L1 and L2 standardization as a benchmark and goes on towards more complex algorithms. Each and every parameter and feature has been fine tuned to suit their specific models which are then tested on train and test datasets.

News headlines are fundamentally catchy and attractive to listen to and often times the general public places their bets based on them. We have taken these headlines and utilized neural networks along with LSTMs in order to predict their sentiment. Since the data is small and very quirky, we didn't want to make use of pre-defined models. It was found that these models were very clunky and failed to provide a good result. The news data was fed through several neural networks till we settled on LSTM's as they seemed to provide the most accurate sentiment. We trained the model using twitter tweets and focused more on the negative aspects of the dialogue.

This research aims to build a model that can predict price fluctuations by evaluating them with the news sentiments. We are using RNN's and LSTM's to retain and identify news polarities and regression models to get the stock price. The latest data of the past few months was scraped from finviz website using alpha vantage.

## II. LITERATURE SURVEY

This work was referenced to gain an insight into the numerous ways in which we can deploy neural networks. We took particular notice of how the pre-trained sklearn models were unable to satisfactorily

predict the direction of prices. The work pays special attention to the inner workings of regularization techniques and we built upon these by deploying them with bagging. [1]

This showcased the inner workings of regression models and how to hyper-parameterize them. We gained a lot of knowledge from the assumptions and graphs plotted. We saw how we must go about while building our neural network and what things to avoid and prioritize in the process. [2]

This was another paper that we used to better understand Long Short Term Memory neural networks. Since this work paid special attention to the sophisticated inner workings of the network instead of just treating it as a black box we gained crucial insight on how to modify the model to better suit our data. [3]

This work sought to utilize the Naïve Bayes algorithm to analyse the sentiments given in text format. Even though the algorithm used was not what we ended up using, through this paper we gained a glimpse of how text pre-processing and classification looked like. We also went through the algorithm's working to better understand the data and its intricacies. [4]

This paper delves into classifying sentiment using ensemble and clustering algorithms like KNN. We gained important knowledge on the relationship between the text and their mathematical counterparts. Vectorization, silhouette and instance learning algorithms were a few topics that we touched in this paper.[5]

This paper was essential for our work as it helped us understand on how to actually get the data required for prediction. It helped us understand concepts that would later on become invaluable assets for our project. The various methods of data scraping and summarizing the model's prediction were two of the most notable aspects of the research paper [6].

This paper was sought by us when we faced overfitting problems and couldn't understand just how different intra-day and long-term stock price prediction truly were. This paper was especially useful when we couldn't find a balance between the model's complexity and generalization. The insights of the author helped us immensely during the latter part of the project. [7]

## III. METHODOLOGY

3.1.1. News Collection-

By scraping the finviz website with lovely soup and alpha vantage, we were able to acquire data for two months. The information is separated into two parts: stock market data (open, close, high, low, date, and volume) and news headlines for that specific stock.

We take a ticker as input that corresponds to the stock that we want to predict.

3.1.2. Pre-Processing-

Text data is unstructured data. As a result, we are unable to submit raw test data to the classifier as an input. To act on the word level, we must first tokenize the document into words. Text data has a higher number of noisy words that don't help with categorization. As a result, such words must be dropped. Text data may also include numerals, additional white spaces, tabs, and other symbols.

punctuation characters, stop words etc. We also need to clean data by removing all those words. For this purpose, we created own stop-word list which specifically contains stop words related to finance world and also general English stop words. We built this using reference from [16]. This stops words list contains general words including Generic, names, Date and numbers, Geographic, Currencies.

Stemming is also essential for reducing word repetition. All words are replaced with the original version of the term using the stemming method. The terms 'developed,' 'development,' and 'developing,' for example, are reduced to the stem word 'develop.

3.1.3. Sentiment Detection Algorithm-

We are using a Dictionary-based strategy that employs the Bag of Word technique for text mining to identify automated sentiment in news articles.

This technique is based on J. Bean's study on the use of Twitter sentiment analysis for airline businesses. To build the polarity dictionary, we need two types of words collection, i.e., positive words and negative words. Then we can compare the article's words to both of these word lists, count the number of terms that exist in both dictionaries, and determine the document's score. Using generic terms with positive and negative polarity, we constructed the polarity words dictionary. In addition, utilizing McDonald's study [16], we employed Finance-specific phrases

with their polarity. We found 2360 positive words and 7383 negative terms in this lexicon.
For the news story, we're looking at the string that simply includes the headline. The algorithm for calculating a document's sentiment score is shown below.

Algorithm:
1. Tokenize the document into word vector and pad the sequences.
2. Prepare the dictionary which contains words with its sentiment (positive or negative) (-1 to 1).
3. Check against each word weather it matches with one of the words from positive word dictionary or negative words dictionary.
4. Count the number of positive and negative polarity terms.
5. Determine the document's score: count (pos.matches) – count (neg.matches).
6. If the Score is 0 or above, the document is considered positive; otherwise, it is considered negative. We are considering one assumption in our implementation: if the document's score is 0, we identify it as positive because we are investigating a two-class problem. As a consequence, we have a news collection with a sentiment score and positive or negative polarity.

3.1.4. Scraping Stock Market Data-
We import the necessary libraries like NumPy, pandas, request, etc. Taking a ticker or list of tickers as input we scrap the finviz website for stock market data or news related to that company. Using alpha-vantage, time series and API we try to get the daily adjusted (intra-day) data and plot it as a graph.

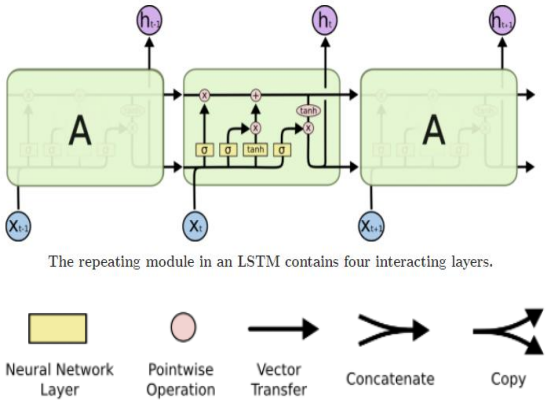
3.1.5. Building Regression Model-
Starting from linear regression we went onwards to more complex algorithms like naïve bayes and LSTM (long short-term memory). Since we had only taken in the latest information, we encountered a case of low bias and high variance while prediction. To circumvent this, we used L1 and L2 standardization techniques as a benchmark. This essentially aids us in establishing a penalty for the less relevant traits while also improving accuracy. For prediction, we used certain deep learning algorithms like Long Short-Term Memory. The LSTM model was created with care, employing numerous dense, maxpooling, and dropout layers and a soft max activation. We intended to avoid using the soft max activation function because of the expanding gradient problem, but we couldn't since we needed the loss to be categorical – entropy.

- Ridge Regression:

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{p}w_j\times x_{ij}\right)^2+\lambda\sum_{j=0}^{p}w_j^2$$

Cost function for ridge regression

- LSTM(Long Short Term Memory):



The repeating module in an LSTM contains four interacting layers.

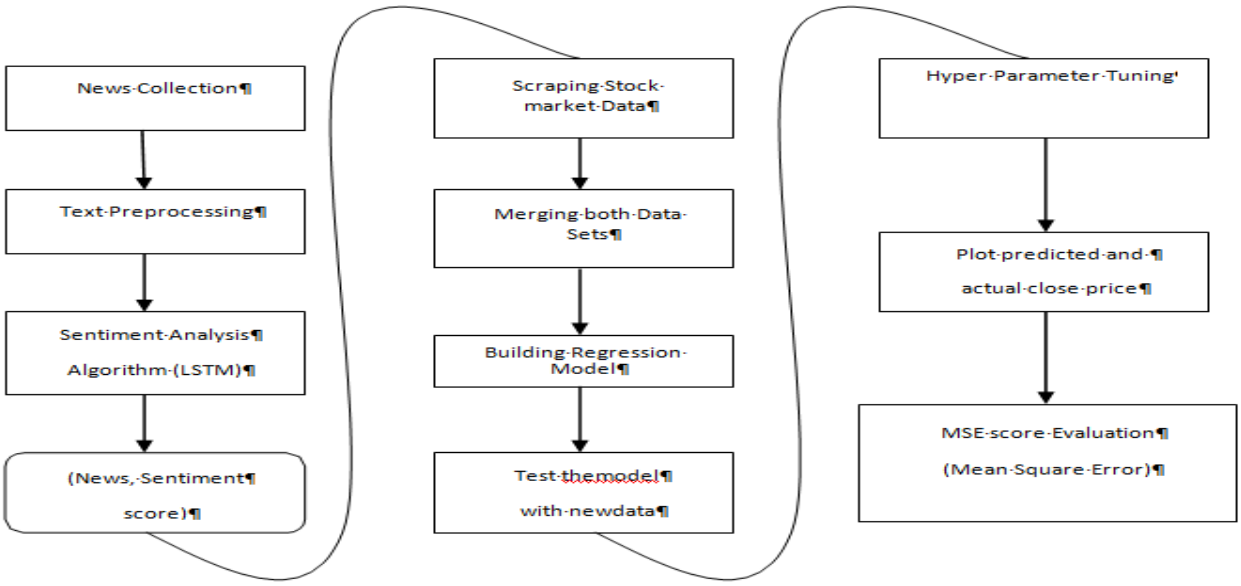
3.1.6. System Evaluation-
We divided the data into train and test set. Also, we created unknown data set for classifier to check accuracy of classifier against new data. We evaluated all three classifiers performance by checking each one's accuracy, precision, recall and F1 score. The results are as given in the next section.

3.1.7. Testing with new Data-
Hundred news articles as an average were considered for any one stock. When comparing the sentiment results it was found that LSTMs were able to predict with a greater accuracy as compared to predefined models or RNN's.

3.1.8. Plotting the values-
After prediction of data, we plotted the predicted and actual values using pyplot and seaborn. Since the output was in a continuous format we cannot find metrics like accuracy, recall or f-score. Instead, we focused on getting the mean squared error for the output. All models underwent hyper-parameter tuning in order to ensure that the MSE didn't go over 1. The end result was expressed in the form of a

graph in which the red lines denote the predicted output along with the blue lines that are the actual output.

## IV.FLOW CHART




## V. EXPERIMENTAL OUTPUT AND RESULTS.

1.Data of stock market

| | date | 1. open | 2. high | 3. low | 4. close | 5. volume |
|---|---|---|---|---|---|---|
| 0 | 2022-03-18 16:15:00 | 28.59 | 28.59 | 28.5900 | 28.59 | 5155.0 |
| 1 | 2022-03-18 16:00:00 | 28.54 | 28.60 | 28.5200 | 28.60 | 104190.0 |
| 2 | 2022-03-18 15:45:00 | 28.46 | 28.54 | 28.4400 | 28.54 | 36869.0 |
| 3 | 2022-03-18 15:30:00 | 28.41 | 28.48 | 28.4077 | 28.47 | 18763.0 |
| 4 | 2022-03-18 15:15:00 | 28.38 | 28.43 | 28.3750 | 28.42 | 13334.0 |

2.Stock News Data

| | ticker | date | time | title | compound |
|---|---|---|---|---|---|
| 0 | TTM | 2022-03-01 | 07:33PM | Jaguar Land Rover suspends sales to Russia | 0.0000 |
| 1 | TTM | 2022-03-01 | 11:01AM | Carmakers hit the brakes on Russia as sanction... | 0.0000 |
| 2 | TTM | 2022-03-01 | 07:34AM | UPDATE 2-Jaguar, Aston Martin pause Russian de... | 0.0000 |
| 3 | TTM | 2022-02-25 | 06:00AM | 5 Autonomous Vehicles Stocks to Buy as Nvidia-... | 0.0000 |
| 4 | TTM | 2022-02-25 | 03:59AM | UK car production plummets to 13-year low in J... | -0.2732 |

3.Feature Correlation

| | 1. open | 2. high | 3. low | 4. close | 5. volume | compound |
|---|---|---|---|---|---|---|
| 1. open | 1.000000 | 0.958973 | 0.868303 | 0.824986 | -0.134752 | 0.277911 |
| 2. high | 0.958973 | 1.000000 | 0.852027 | 0.840132 | 0.004456 | 0.280950 |
| 3. low | 0.868303 | 0.852027 | 1.000000 | 0.984479 | -0.105003 | 0.178618 |
| 4. close | 0.824986 | 0.840132 | 0.984479 | 1.000000 | -0.007876 | 0.152300 |
| 5. volume | -0.134752 | 0.004456 | -0.105003 | -0.007876 | 1.000000 | -0.100600 |
| compound | 0.277911 | 0.280950 | 0.178618 | 0.152300 | -0.100600 | 1.000000 |

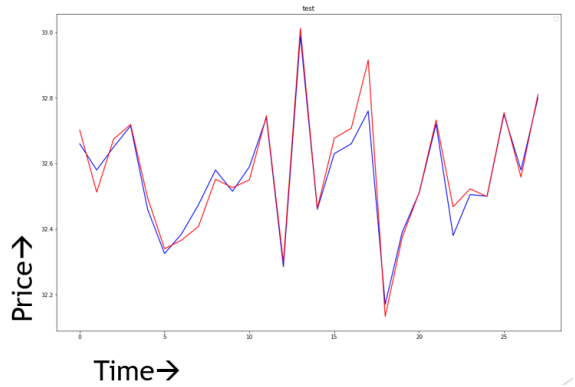
4.LSTM Base Model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 28, 128)           256000
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 28, 128)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 196)               254800
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 394
=================================================================
Total params: 511,194
Trainable params: 511,194
Non-trainable params: 0
_________________________________________________________________
None
```

5.Validating model on train dataset

```
Epoch 1/7
 - 20s - loss: 0.4382 - acc: 0.8168
Epoch 2/7
 - 19s - loss: 0.3214 - acc: 0.8648
Epoch 3/7
 - 19s - loss: 0.2836 - acc: 0.8808
Epoch 4/7
 - 19s - loss: 0.2578 - acc: 0.8958
Epoch 5/7
 - 19s - loss: 0.2288 - acc: 0.9036
Epoch 6/7
 - 19s - loss: 0.2103 - acc: 0.9158
Epoch 7/7
 - 19s - loss: 0.1959 - acc: 0.9239
```

6.Output Graph




Red line –Predicted stock price
Blue line – Actual stock price

## VI. CONCLUSION

A This paper first uses NLP technical tools to analyze and quantify public emotion and proposes a multilayers LSTM model to predict stock price, then compares the prediction effect between the model with both news sentiment score and historical stock technical indicators as input and the model based only on historical stock technical indicators. The experimental results based on three large cap company show that compare to model only considers stock time-series data, model used both sentiment score from news article effectively improve the stock prediction accuracy with smaller MSE and MAE value. In addition, from the MSE value for both models, the LSTM predictor shows a good prediction result, which indicates LSTM model is efficient in time-series prediction, such as stock price and stock return. Thus, this paper verifies the significance of public sentiment from news for the stock market forecasting and demonstrates the importance and feasibility of investigating public emotion when conducting stock market prediction and research.

## VII.ACKNOWLEDGMENT

I would like to appreciate Professor Payal Varangaonkar for her valuable suggestions and persistent guidance.

## REFERENCES

[1] (ICIEECT), Karachi, 2017, pp. 1-1. [3] H. Gunduz, Z. Cataltepe and Y. Yaslan, "Stock market direction prediction using deep neural networks," 2017 25th Signal Processing and Communications Applications Conference (SIU), Antalya, 2017, pp. 1-4

[2] M. Billah, S. Waheed and A. Hanifa, "Stock market prediction using an improved training algorithm of neural network," 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi, 2016, pp. 1-4

[3] 2787-2790. [8] T. Gao, Y. Chai and Y. Liu, "Applying long short term memory neural networks for predicting stock closing price," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2017, pp. 575-578

[4] International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-8 June, 2019, Sentiment Analysis Using Naïve Bayes Classifier Sentiment Analysis Using Naïve Bayes Classifier

[5] Huy Tien Nguyen, Minh Le Nguyen, “An ensemble method with sentiment features and clustering support”, Neuro-computing”,vol. 370, pp. 155-165, 22 December 2019.

[6] Mondher Bouazizi and TomoakiOhtsuki, “Multi-Class Sentiment Analysis on Twitter: Classification Performance and Challenges, Big data mining and analytics,” ISSN 2096-0654 03/05 pp181–194, vol. 2, no. 3, September 2019.

[7] K. A. Althelaya, E. M. El-Alfy and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock market prediction," 2018 9th International Conference on Information and Communication Systems (ICICS), Irbid, 2018, pp. 151-156