Leveraging News and Twitter Sentiments for Stock Predictions

Atharvan Dogra
101903783
Thapar Institute of Engineering and Technology
adogra_be19@thapar.edu

Subject: Data Science
Submitted to: Dr. Sahil Sharma
May, 2022

Contents

Leveraging News and Twitter Sentiments for Stock Predictions

Atharvan Dogra
Thapar Institute of Engineering and Technology
adogra_be19@thapar.edu

Abstract

The project leverages news and twitter sentiments to find a pattern in people's sentiments for a company and the changes in the stock prices. A weekly correlation is calculated between the sentiments and changes in prices. Pearson Correlation is used for that. The popular tweets along with corresponding sentiment scores are displayed alongside.

1 Introduction

Most of the works done till now on stock market prediction are based on historical stock prices. Later on the idea of predicting the stock market based on historical data alone was debunked by various studies as it was seen that various other factors are affecting the stock market movements and the prices are largely fluctuating due to them.

According to the efficient market hypothesis (EMH), the financial market movements are dependent on news, current events, and product releases and all of these factors have a significant impact on a company's stock value (Qian and Rasheed, 2007). Because of the lying unpredictability in news and current events, stock market prices follow a random walk pattern and cannot be predicted with more than 50% accuracy (Fama, 1965).

As the popularity of social media is on growth, information on public opinions and feelings has become abundant as people like to post a lot of their ideas online. Social media is transforming into a perfect platform to share public emotions and ideas about any topic and has a significant impact on overall public opinion. Twitter, Facebook, and some other online news platforms like yahoo, have received a lot of attention from researchers in recent times. Twitter is a micro-blogging application that allows users to follow and comment on other users' thoughts or share their opinions in real-time (Leskovec et al., 2007). More than a million users post over 140 million tweets every day. This situation makes Twitter like a corpus with valuable data for researchers (Jansen et al., 2009). Similarly, posts on other social media platforms and news headlines include information that can be exploited for making very useful predictions (Pak and Paroubek, 2010). The financial news has a significant impact on shaping investors' perceptions and assessments of companies and hence influences the stock market.

This project follows the Fundamental analysis technique to discover the future trends of stock by considering news articles and tweets about a company as prime information and tries to classify news as good (positive) and bad (negative). If the news sentiment is positive, there are more chances that the stock price will go up and if the news sentiment is negative, then the stock price may go down.

2 Motivation for the Project

There are many factors that influence stock market prices. One of those factors is investors' reaction to financial news and day-to-day events. Nowadays, news availability has increased dramatically. It is hard for investors to decide the trend of stock prices based on the huge amount of news. So the idea of an automated system to predict future stock prices sounds viable for investors. An automated system that can gather financial news related to the companies of interest in real-time and can execute a machine learning model on the data, along with historical stock price information, to predict price.

3 Related Work

A well-known publication on this idea is by Bollen (Bollen et al., 2010). They investigated whether the collective mood states of the public (Happy, calm, Anxiety) derived from Twitter feeds are correlated to the value of the Dow Jones Industrial Index. They used a Fuzzy neural network for their prediction. Their research shows that public moods

Date	Open	High	Low	Close	Adj Close	Volume	News
2006-12-01	13.114285	13.19	12.871428	91.32	13.045714	198769900	WHAT'S ON TONIGHT : 8 P.M. (TLC) ASHLEY
2006-12-04	13.125714	13.15	12.928572	91.120003	13.017143	177384200	More on Housing Prices : The broadest governr
2006-12-05	13.092857	13.19	12.981428	91.269997	13.038571	165709600	
2006-12-06	12.948571	13.055715	12.81	89.830002	12.832857	159546100	Honoring R.W. Apple in Words and Food : Abou
2006-12-07	12.861428	12.928572	12.414286	87.040001	12.434286	251206900	Homebuilders, and Worries Over Jobs, Lead a [
2006-12-08	12.461429	12.77	12.428572	88.259995	12.608571	196069300	Homebuilders, and Worries Over Jobs, Lead a [
2006-12-11	12.7	12.757143	12.578571	88.75	12.678572	124945100	Sales of iPods and iTunes Not Much in Sync : T
2006-12-12	12.658571	12.691428	12.218572	86.139999	12.305715	256655000	
2006-12-13	12.564285	12.724286	12.45	89.050003	12.721429	214263000	Op-Art; iDentities : Op-Art by Seymour Chwast [
2006-12-14	12.721429	12.857142	12.608571	88.549995	12.65	208082700	Adam, She's Ms. Madam : Kristin Chenoweth u:
2006-12-15	12.717143	12.745714	12.475715	87.720001	12.531428	184984800	Adam, She's Ms. Madam : Kristin Chenoweth u:
2006-12-18	12.518572	12.571428	12.084286	85.470001	12.21	180394200	Apple Delays Earnings Statement Due to Optior
2006-12-19	12.104285	12.382857	11.945714	86.309998	12.33	227851400	Maple's Got Some New Tricks : Maple has new
2006-12-20	12.352858	12.381429	12.105714	84.759995	12.108571	141922900	Recent Openings : New shows in New York inc
2006-12-21	12.1	12.211429	11.742857	82.900002	11.842857	225899800	
2006-12-22	11.922857	12.005714	11.657143	82.199997	11.742857	153325900	'Right on the Money' : Kristin Che
2006-12-26	11.735714	11.795714	11.555715	81.510002	11.644286	122672200	Waiting to Take You Away on a Fact-Filled Tour

Figure 1: Apple Dataset Example.

on Twitter are strongly correlated with Dow Jones Industrial Index. Chen and Lazer (Chen and Lazer, 2011) derived investment strategies by the observation and classification of the Twitter feeds. Bing et al. (Bing et al., 2014) studied the tweets and concluded the predictability of stock prices based on the type of industry like Finance, IT, etc.

Recently, Brian et al. (Pagolu et al., 2016) investigated the correlation of sentiments of the public with stock increases and decreases using the Pearson correlation coefficient for stocks.

Also, Zhihan and Liqian (Zhou et al., 2021) have proposed the approach of domain adaptation for a better understanding of the sentiments. They introduce an event-driven trading strategy that predicts stock movements by detecting corporate events from news articles, unlike all other approaches that just utilize textual features (e.g., bag-of-words) and sentiments to make the predictions directly. Their approach focuses particularly on corporate events.

In this project, I've adopted the approach of predicting the rise and fall in stock prices based on the sentiments extracted from Twitter and News to find the correlation and also found the correlation between the rise and fall of stock prices and sentiments using Pearson correlation.

4 Dataset Collection and Preprocessing

This section describes the dataset available before and the preprocessing and additions that had to be done on it.

4.1 Dataset Description

The dataset has been formed by collecting data of 2 companies' stocks i.e. Microsoft and Apple and 2 indexes i.e. Dow Jones and Nifty. The datasets available to me were in various forms.

- **Nifty:** Data with cleaned news sentences of 18 years [2003-2020] was available along with the *negative*, *neutral*, *positive* and *compound* sentiment scores available with each sentence.

Figure 2: Word Cloud of NIFTY News Positive Keywords

- **Dow Jones:** Data with top 25 news for nine years [2008-2016] was available for this index. Similar to Nifty, the news data available was with *negative*, *neutral*, *positive* and *compound* sentiment scores for each of the 25 sentences for a day, in uncleaned form.

- **Apple:** Important news data for 11 years [2006-2016] was available in the uncleaned form, with news missing for some dates. Also, in this case, the sentiment scores weren't available, and hence the sentences needed to be cleaned and sentiment scores calculated.

- **Microsoft:** Similar to apple, the data with news for 11 years was available in an un-

cleaned form for this stock with news for several dates missing.

Along with the dataset for news and sentiments, the data for stock prices containing prices for the respective stock for 11 years was taken from yahoo finance and open and close prices were used in the final dataset.

4.2 Data preprocessing

Data of stock prices collected is not complete because of missing prices on weekends and public holidays when the stock market does not function. The missing data is approximated using a simple technique by (Mittal and Goel). Stock data usually follows a concave function. So, if the stock value on a day is a and the next value present is b with some missing in between. The first missing value is approximated to be (b+a)/2 and the same method is followed to fill all the gaps.

News data for several dates was missing and hence had to be scraped for the particular dates so as to get consistent data for the complete range of dates the data was being used for.

Tweets consists of many acronyms, emoticons and unnecessary data like pictures and URL's. So tweets are preprocessed to represent correct emotions of public. For preprocessing of tweets we employed three stages of filtering: Tokenization, Stopwords removal and regex matching for removing special characters.

1. **Tokenization:** Tweets are split into individual words based on the space and irrelevant symbols like emoticons are removed. We form a list of individual words for each tweet.

2. **Stopword Removal:** Words that do not express any emotion are called Stopwords. After splitting a tweet, words like a,is, the, with etc. are removed from the list of words.

3. **Regex Matching for special character Removal:** Regex matching in Python is performed to match URLs and are replaced by the term URL. Often tweets consists of hashtags() and @ addressing other users. They are also replaced suitably.

   For example, #Microsoft is replaced with Microsoft and @Billgates is replaced with USER. Prolonged word showing intense emotions like cooooooooool! is replaced with cool! After these stages the tweets are ready for sentiment classification.

```
def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will
        not", phrase)
    phrase = re.sub(r"can\'t", "can
        not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not",
        phrase)
    phrase = re.sub(r"\'re", " are",
        phrase)
    phrase = re.sub(r"\'s", " is",
        phrase)
    phrase = re.sub(r"\'d", "
        would", phrase)
    phrase = re.sub(r"\'ll", "
        will", phrase)
    phrase = re.sub(r"\'t", " not",
        phrase)
    phrase = re.sub(r"\'ve", "
        have", phrase)
    phrase = re.sub(r"\'m", " am",
        phrase)
    return phrase
```

Listing 1: Code snippet of the regex used for decontraction.

4.3 Vader Sentiments

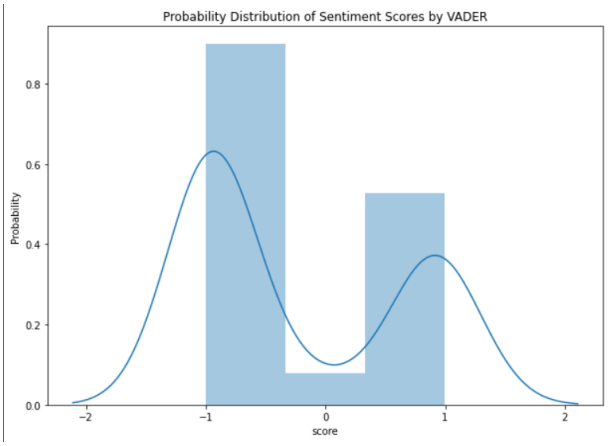

Figure 3: Probability Distribution of Vader Sentiments

VADER (Valence Aware Dictionary and sEntiment Reasoner) is an open-source, lexicon, and rule-based sentiment analysis tool that is specifically designed for sentiment recognition in the social media context.

VADER is a kind of sentiment analysis that depends on the lexicons of sentiment-related words. In this methodology, for every word in the vocabulary it is estimated whether it is positive or negative, and, how +ve or -ve.

For our dataset, VADER sentiments were calculated for the Microsoft and Apple stock news

as that is the score that was already available for the NIFTY and Dow Jones data. The compound scores were appended as a column in the data for the individual stocks.

5 Methodology

The preprocessed data is turned into a pandas dataframe for simpler functioning. With the data of opening and closing prices of stocks, the open-close difference is also calculated and appended as a column in the dataframe. Various model

Various different methods and models were experimented with for the prediction of stock prices, like, ARIMA, SARIMAX, Facebook Prophet, and LSTM, the best RMSE came out to be from LSTM and hence it was used for developing the final model.

5.1 Methods Experimented

5.1.1 ARIMA:

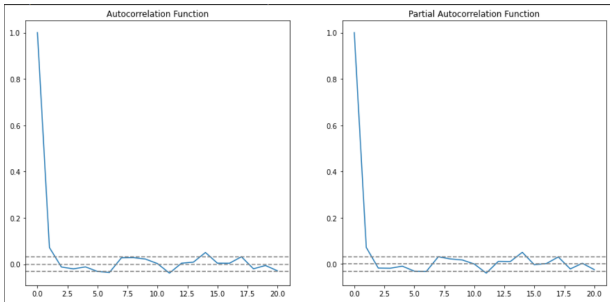

Figure 4: Autocorrelation and Partial Autocorrelation functions of ARIMA

ARIMA, acronym for AutoRegressive Integrated Moving Average, is a statistical model used for analysis and forecasting of time series data.

The name captures the key aspects of the model:

- AR: Autoregression. The dependent relationship between observation and number of lagged observations is used in this model.
- I: Integrated. The time series is made stationary using differencing of an observation from the observation at the previous time step.
- MA: Moving average. The dependency between obeservation and residual error from moving average applied to lagged observations is used.

5.1.2 SARIMAX

In ARIMA only the trend information in the data was considered and the seasonal variation was ignored. In this variation of ARIMA, the seasonal variation in the data is also considered.

5.1.3 Facebook Prophet:

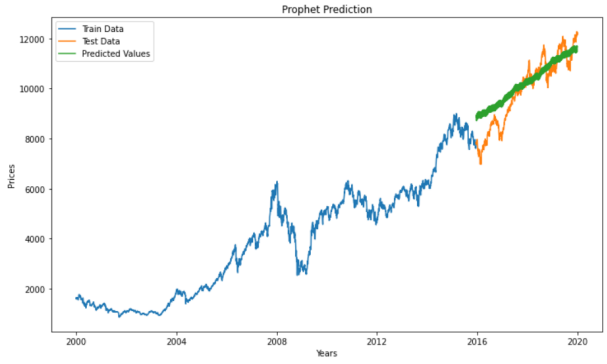

Figure 5: Prediction plot for facebook prophet

Prophet is a library based on decomposable (trend+seasonality+holidays) models. Using this time series predictions can be made with good accuracy using simple intuitive parameters. It also supports the inclusion of the impact of custom seasonality and holidays.

5.1.4 LSTM Model

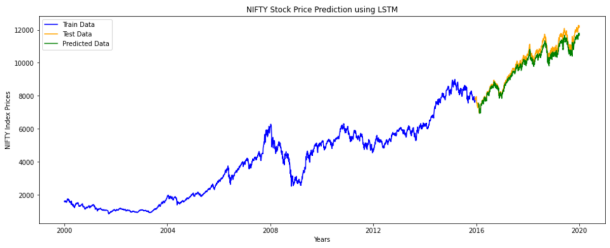

Figure 6: Predictions plot for LSTM

As the stock prices are do not guarantee stationarity and are highly nonlinear and sometimes even stock prices can seem completely random, above methods do not work very effectively on them.

The above problem is countered using Neural Networks (sequential models like LSTM, GRU, etc.), which do not require any stationarity. Neural networks are extremely efficient in finding relationships between data and using it to predict new data.

5.2 LSTM Model with News Polarity

After results from all above models, LSTM is decided to be the final model for the application and

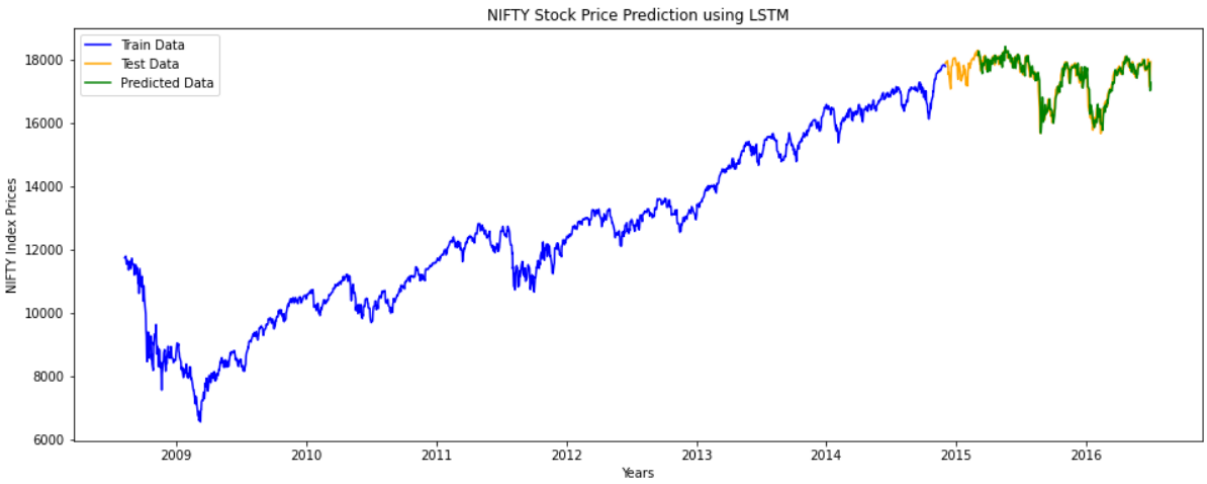

Figure 7: Prediction plot for LSTM with Sentiments

the factor of news sentiment is added. For this another column for sentiment scores for all the dates in the dataframe is added. The stock prices and sentiment scores are inputted in the model for the prediction of stock price for the next day.

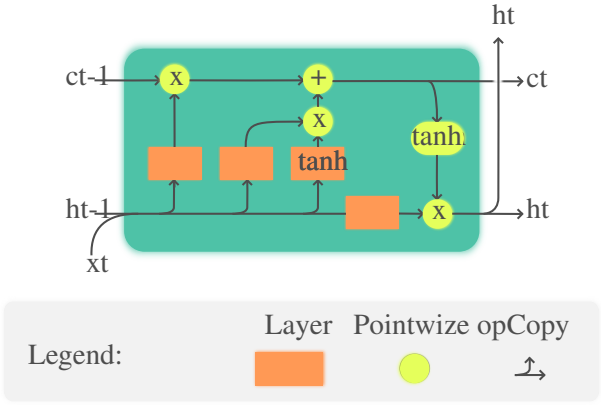

Figure 8: LSTM cell representation

5.3 Pearson Correlation

Along with the prediction of a gain or fall in the stock price, the application also tries to capture the correlation between public sentiments and changes in stock price over the last 7 days. For this, the difference in the open-close price of a stock in the last 7 days is used along with the sentiment scores for the last 7 days and the person correlation is calculated between them given by (1)

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \tag{1}$$

and the estimate is given by (2)

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}} \tag{2}$$

Latest Tweets and the latest news are obtained from the Twitter API and Polygon API respectively, regarding the particular companies.

The sentiment scores for this latest news and tweets are calculated using Flair (Akbik et al., 2019) sentiments, which utilizes DistilBERT (Sanh et al., 2019) as the language model for the task.

5.3.1 Flair

Flair is a powerful NLP library that allows us to apply our state-of-the-art natural language processing models to your text, for tasks such as named entity recognition (NER), part-of-speech tagging (PoS), special support for biomedical data, sense disambiguation and classification, with support for a rapidly growing number of languages

5.3.2 DistilBERT

It is a small, fast, cheap and light Transformer model trained by distilling BERT base. The process to make it small and light included reducing its parameters by 40% than bert-base-uncase, it runs 60% faster while preserving more than 95% of BERT's performance measured on the GLUE language understanding benchmark.

5.4 Generating models for application

After a final model was decided and trained and tuned for hyperparameters, it was saved as 4 different files for 4 different prediction models, i.e., Microsoft, Apple, Nifty, and Dow Jones.

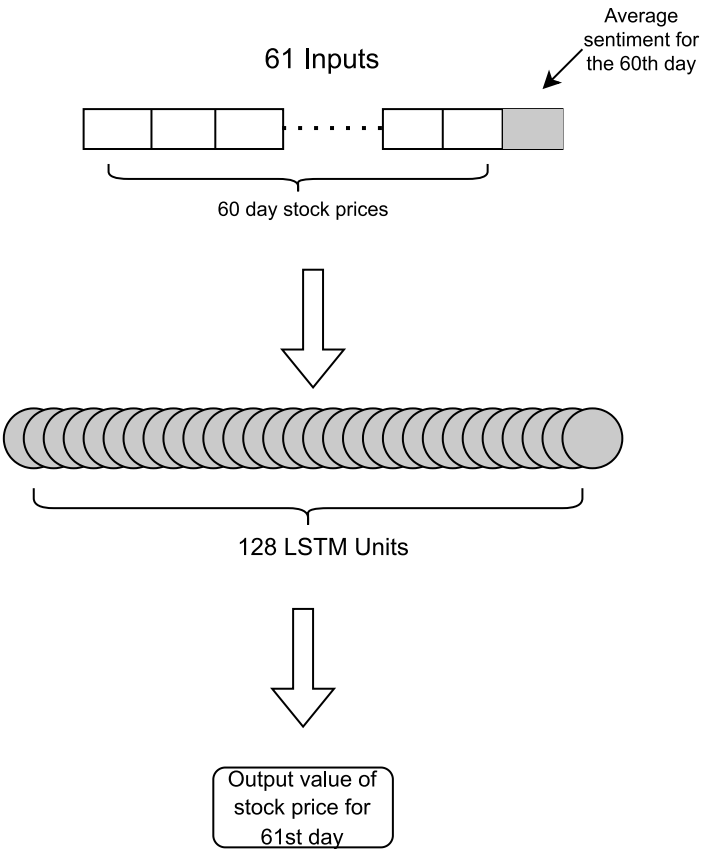

Figure 9: Neural Architecture for stock price prediction

Along with the models, the min-max scalers were also saved and used in the application as the final predictions on the live data needed to be scaled back to normal values for the detection of a rise or fall in price in comparison to the previous price.

5.5 Prediction Method

One important thing that was noticed through the prediction graph was that the consecutive predictions for a few days were shifted up or down compared to the actual prices of the test set but they were still following the same rise or fall pattern.

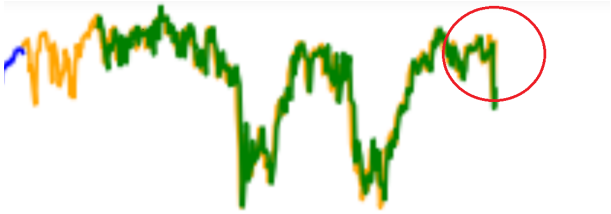

Figure 10: Representation of shifted prediction

So in order to avoid an error in the prediction of gain or fall, the output was given based on the difference in the previous day's prediction and the current day's prediction.

5.6 Deployment to application

Finally, the whole application is deployed to an application developed using React. The ML model were saved using the default function provided in tensorflow and the live prediction are being made through an API function developed using FastAPI, which is being used to display the output on the frontend.

6 Experimental Setup

Structuring of data is done using the sliding window concept, which is the standard for using LSTM models. The window is being used with a look back of 60 days as this is proven to be the best time frame for accurate prediction of stock prices, through several researches.

Hence, the total number of features being inputted in the LSTM layer is 61 [60 days of stock prices and 1 unit for news sentiment of 60th day].

Theses 61 features are are then inputted into a LSTM layer having 128 $units$ and activation of $hyperbolic\ tangent$ function. The neural architecture is compiled with this LSTM layer, a dense layer with 1 $unit$ (as this is a regression problem) and the $Adam$ optimizer with a learning rate of

Hyperparameters	Value
LSTM Layer	1
LSTM Size	128 units
Activation	$tanh$
Optimizer	Adam
Learning Rate	0.001
Dense	1
Batch Size	64
Epochs	50

Table 1: Major Hyperparameters of the Model

0.001. This model gives the prediction for the next day's price.

7 Results

Experimenting with several kind of models like ARIMA, Facebook Prophet, and LSTM gave a fair idea of the difference in the performance of each of these models. LSTM came out to be the strongest performer of among all.

Model	RMSE on Test Data
ARIMA	1690.05
SARIMAX	964.5
FB Prophet	709.71
LSTM	260.66
LSTM with News Sentiments	180.5

Table 2: Results comparison of various methods

And later adding sentiment values to the LSTM input further improved the performance to new heights.

7.1 Result Analysis

Hence, from the above results, it can be concluded that neural networks are able to detect better patterns even in very random and nonlinear time-series data such as that of stock prices.

It was also noticed that the introduction of a single input of sentiment score to the LSTM mechanism improved the RMSE by a drastic 30% hence supporting that public sentiments have a major role in the movement of stock markets and the LSTM mechanism is able to detect its pattern very well.

8 Future Scope of Development

The next steps in improving the accuracy of prediction of stock movement based on public sentiments would be to develop a better and more complex neural architecture using the modern method of neural architecture search (NAS) (Zoph and Le, 2017).

Also, an important method in stock sentiment analysis is to capture news of the particular domain that affects the stock. (Zhou et al., 2021) proposed that corporate news is the ones that affect the stocks most and hence they should be focussed upon. They also used the method of domain adaptation.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Bing, Keith C.C. Chan, and Carol Ou. 2014. Public sentiment analysis in twitter data for prediction of a company's stock price movements. In *2014 IEEE 11th International Conference on e-Business Engineering*, pages 232–239.

Johan Bollen, Huina Mao, and Xiao-Jun Zeng. 2010. Twitter mood predicts the stock market. *CoRR*, abs/1010.3003.

Ray Chen and Marius Lazer. 2011. Analysis of twitter feeds for the prediction of stock market movement.

Eugene F. Fama. 1965. The behavior of stock-market prices. *The Journal of Business*, 38(1):34–105.

Jim Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *JASIST*, 60:2169–2188.

Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5–es.

Anshul Mittal and Arpit Goel. Stock prediction using twitter sentiment analysis.

Venkata Sasank Pagolu, Kamal Nayan Reddy Challa, Ganapati Panda, and Babita Majhi. 2016. Sentiment analysis of twitter data for predicting stock market movements. *CoRR*, abs/1610.09225.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Bo Qian and Khaled Rasheed. 2007. Stock market prediction with multiple classifiers. *Applied Intelligence*, 26(1):25–33.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Zhihan Zhou, Liqian Ma, and Han Liu. 2021. Trade the event: Corporate events detection for news-based event-driven trading. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2114–2124, Online. Association for Computational Linguistics.

Barret Zoph and Quoc V. Le. 2017. Neural architecture search with reinforcement learning.

A Graphs for Losses

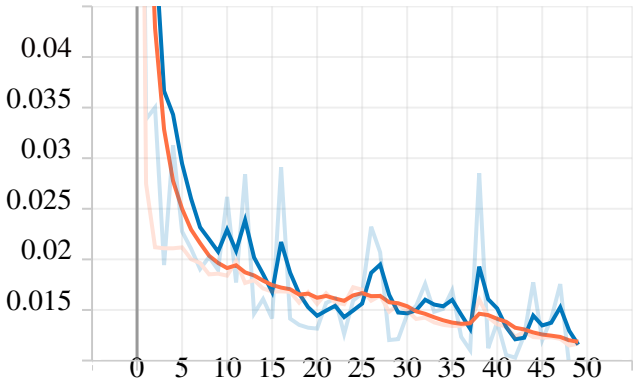

Figure 11: Epochs vs MAE Loss

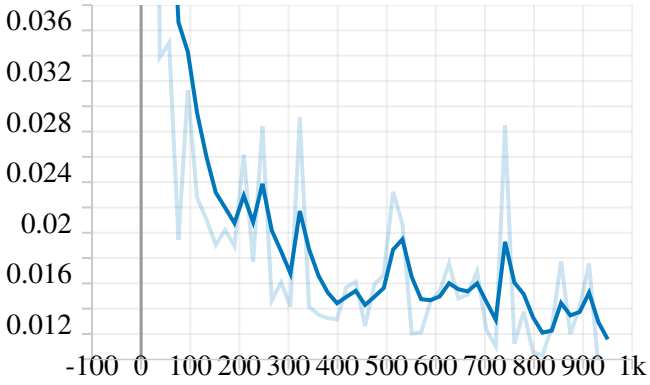

Figure 12: Evaluation MAE Loss vs Iterations

B Application developed

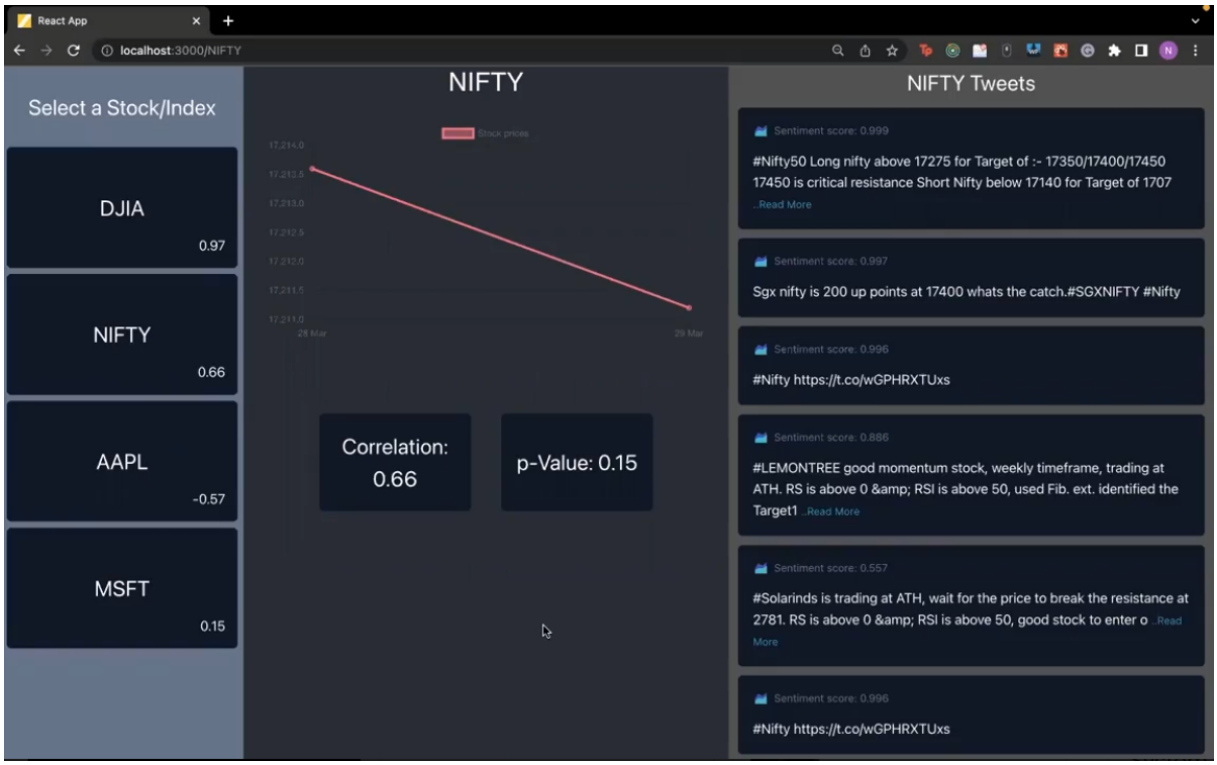

Figure 13: Display of application developed

The application couldn't be hosted as the dependencies which included tensorflow and flair, were taking over 1.5 GB and no free hosting service allows that amount of storage.

Attaching a GitHub link which consists the code for the backend, front and the ML code in an IPYNB notebook:
https://github.com/AtharvanDogra/ionathon