Sentiment Analysis on Financial Tweets using LSTM

Drumil Shah 202018009@daiict.ac.in
Kanakamedala Ragaja 202018012@daiict.ac.in
Shubhaditya Roy 202018021@daiict.ac.in
Himani Desai 202018030@daiict.ac.in

Bhavya Dubey 202018049@daiict.ac.in
Mansi Thakkar 202018053@daiict.ac.in
Dhaval Vachheta 202018058@daiict.ac.in

Department - MSc. Data Science
College - DAIICT, Gandhinagar, Gujarat.

ABSTRACT

Analyzing the big textual information manually is tougher and time-consuming. Sentiment analysis is an automated process that uses computing to spot positive, negative, or neutral opinions from the text. Sentiment analysis is widely used for getting insights from social media such as Twitter, and merchandise reviews to create data-driven decisions. Sentiment analysis systems are accustomed to add up to the unstructured text by automating business processes and saving hours of manual processing. Twitter is an online social networking and microblogging service with over 200m monthly active users. Given this massive user base researchers have tried to mine the derived vast source of data for different purposes. In this work, we investigate the relationship between the market indicators for various companies and the volume of tweets mentioning their names or stock symbols. In recent years, Deep Learning (DL) has garnered increasing attention within the industry and academic world for its high performance in various domains. This work is going to predict the stock price of different companies based on the sentiments of the news and tweets related to a particular company. We do sentiment analysis on text reviews by using Long Short-Term Memory (LSTM).

Keywords – sentiment, sentiment analysis, twitter, finance, stock market, LSTM, RNN.

1. INTRODUCTION

Sentiment analysis is that the computerized process of text processing that higher cognitive process to an opinion a couple of given subjects from a transcription. In an exceedingly present generation, we create quite 1.5 quintillion bytes of information daily, sentiment analysis has become a key tool for creating a sense of that data. It absolutely was utilized by the businesses to induce key insights and automate every kind of process for their business development. Sentiment Analysis [2] is also called opinion mining. Sentiment analysis isn't only sentiment mining but also contextual mining of text which identifies and extracts subjective information in source material and helps a business to know the social sentiment of their service, brand, or product while monitoring online conversations. Sentiment Analysis is that the most used text classification tool that analyses an incoming message and tells whether the essential opinion is positive or negative. Sentiment analysis will be applied at different levels of scope like Document-level sentiment analysis obtains the sentiment of an entire document or paragraph. Sentence level sentiment analysis obtains the results of one sentence. Sub-sentence level sentiment analysis obtains the results of sub-expressions within a sentence. Recently, LSTM is most popular to deal with sentiment classification. LSTM was proposed by Hoch Reiter and Schmid Huber in 1997 and was refined and popularized by many people in the following work. They work tremendously well on large different types of problems and are now widely used. LSTMs are explicitly designed to ignore the long-term dependency problem [3]. Remembering information for a long time is practically their default behavior, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of neural-networks.

The project workflow we have established is shown in figure 1. Initially, we extract the tweets from twitter using the tweepy package of python. Then we preprocess those tweets. At this step all the unnecessary words are removed from tweets such as stop words, urls and other symbols. As the next step, we build the LSTM model to predict the sentiment of each tweet. We use a kaggle dataset to analyze the sentiment of each English tweet related to the selected companies. Once the model was built, we fed real time tweets for testing purposes. The model gives the predicted sentiment of each tweet. After getting the sentiment of each tweet we are trying to correlate the normalized net sentiment value with the net percentage changes in stock price of a particular company.

2. DATASET

The **training** dataset used in this project is taken from Kaggle[1] (Sentiment Analysis on Financial Tweets - tweet_sentiment.csv) which has cleaned tweets and a corresponding sentiment value(3 labels) assigned to those tweets. There are a total of 21,995 unique tweets in this dataset and 28,439 valid tweets. These tweets were originally taken from David Wallach's Kaggle account and have all the publicly traded companies (tickers and company names). The influencers whose tweets were monitored were: *'MarketWatch', 'Business','YahooFinance','TechCrunch', 'WSJ',* *'Forbes', 'FT', 'TheEconomist', 'nytimes', 'Reuters', 'GerberKawasaki', 'jimcramer', 'TheStreet', 'TheStalwart', 'TruthGundlach', 'CarlCIcahn', 'ReformedBroker', 'benbernanke', 'bespokeinvest', 'BespokeCrypto', 'stlouisfed', 'federalreserve', 'GoldmanSachs', 'ianbremmer', 'MorganStanley', 'AswathDamodaran', 'mcuban', 'muddywatersre', 'StockTwits', 'SeanaNSmith'.*

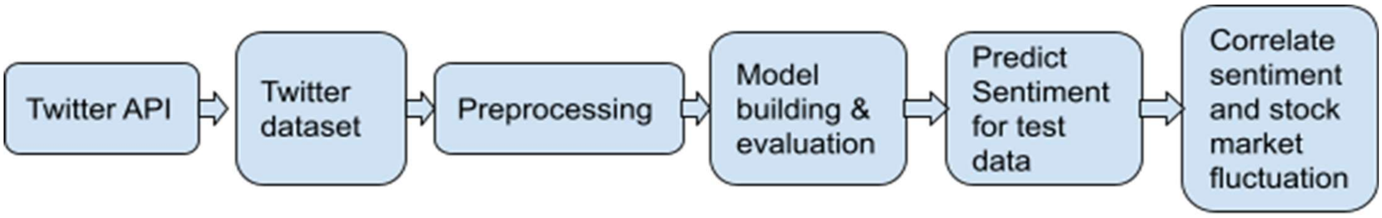

Figure 1. Project workflow

For **testing**, the dataset is processed by us and the data for this dataset is streamed directly from the twitter streaming API which is connected with the Python package called Tweepy which downloads tweets in real-time and is useful for obtaining a high volume of tweets. To stream the data a Twitter Developer Account is required. Twitter grants authentication credentials to applications, not accounts so an application was registered by giving the application name, application description and the use of the application, to be able to make API calls. Thereafter authentication credentials were created by generating keys, tokens, and secrets. The influencers whose tweets were monitored were:

NIFTY 50 Handles:

'INDMarketsLIVE', 'Nifty50trade', 'forum_stock', 'Nifty50Striker', 'livemint', 'moneycontrolcom', 'MSChawla555'

NIFTY 50 Companies:

'Anandmahindra', Bajaj_Finserv', 'Bajaj_Finance', 'sanjivrbajaj', 'BajajAutoFin', 'TataMotors_Cars', ' TataMotors', 'TataCompanies', 'RNTata2000', 'Maruti_Corp', 'NexaExperience', 'MSArenaOfficial', 'reliancegroup', 'reliancejio', 'RelianceCapital'

3. METHODOLOGY

3.1 Data Preprocessing

In data preprocessing we focus on making the textual tweets dataset and its corresponding labels to be appropriate for model training. So we clean the tweets and perform tokenization and many more techniques to convert each word representable in numeric and to be given it to the model. Similar techniques like label encoder and one hot encoder are applied to the labels.

Some Important Terminologies

1. Lemmatizer

"Lemmatization" is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word.

2. NLTK

The "Natural Language Toolkit", or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.

NLTK is intended to support research and teaching in NLP or closely related areas, including Empirical Linguistics, Cognitive Science, Artificial Intelligence, Information Retrieval, and Machine Learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

3. Stop Words

The process of converting data to something a computer can understand is referred to as **"Pre-Processing"**. One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words (data), are referred to as **"Stop Words"**.

Stop Words: A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

We would not want these words to take up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to stop words. NLTK (Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages.

4. Label Encoder

In machine learning, we usually deal with datasets that contain multiple labels in one or more than one column. These labels can be in the form of words or numbers. To make the data understandable or in human-readable form, the training data is often labeled in words.

"Label Encoding" refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important Preprocessing step for the structured dataset in supervised learning.

5. Tokenizer

In python, Tokenization basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language.

6. Pad Sequences

NLP Sequencing is the sequence of numbers that we will generate from a large corpus or body of statements by training a neural network. We will take a set of sentences and assign them numeric tokens based on the training set sentences.

7. One Hot Encoder

"One Hot Encoding" refers to splitting the column which contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains "0" or "1" corresponding to which column it has been placed

3.2 Block Diagram of Training Data and Testing Data

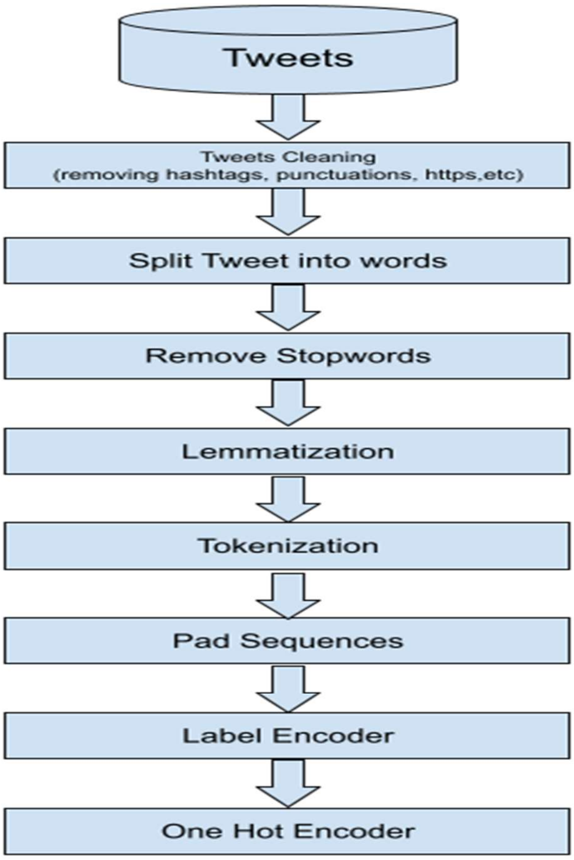

Figure 2. Training Data Preprocessing flowchart

Here we perform dataset preprocessing of training dataset:

1. Cleaning tweets is done on the tweets by which we get plain sentences.
2. Splitting the tweets into words from the cleaned sentence.
3. Removing the stopwords.
4. Doing lemmatization of each word
5. Tokenization of tweets
6. Pad sequence makes numeric representation of tokens.
7. Label Encoder is applied on the target labels.
8. One hot encoding is then done by the label encoded targets.

Then in testing dataset as it is an unlabeled data to which model gives appropriate labels by predicting their labels so till pad sequences the same approach as training dataset is done but before that company name is extracted from the tweets for analysis purpose. Upon prediction of this data we need to perform label encoding on the predicted labels in order to compare it with actual labels and compute various metrics.

3.3 Model

- **What is LSTM?**

The Long Short-Term Memory network, or LSTM for short, is a type of recurrent neural network that achieves state-of-the-art results on challenging prediction problems. Long Short-Term Memory is a kind of recurrent neural network. In RNN

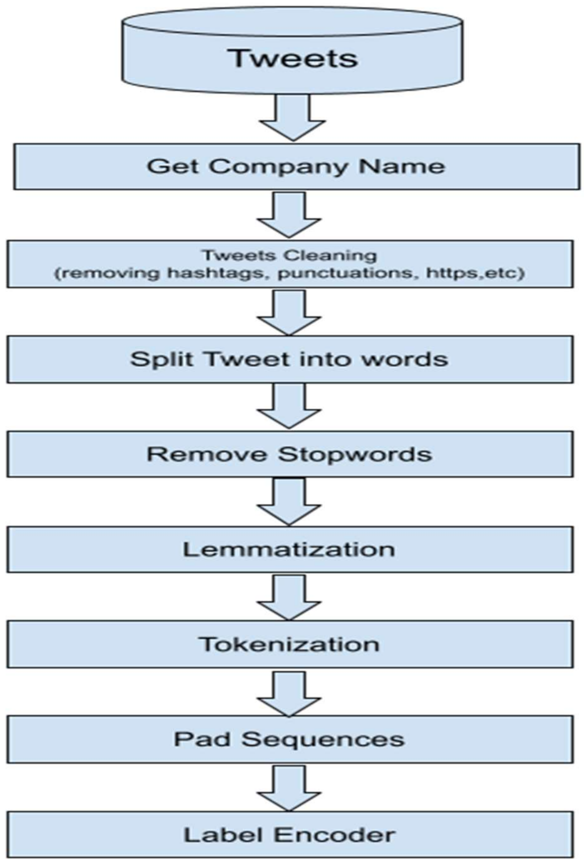

Figure 3. Testing Data Preprocessing flowchart

output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN.

Cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give efficient performance. LSTM can by default retain the information for a long period of time. It is used for processing, predicting and classifying on the basis of time series data.

- ## Layers of LSTM model

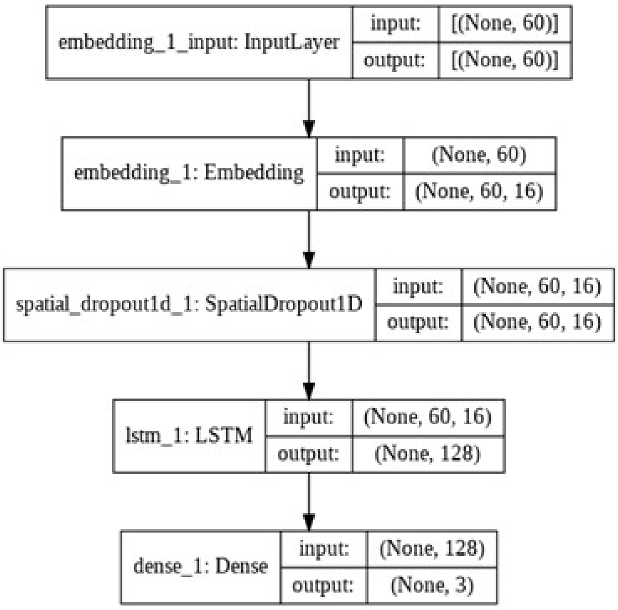

Figure 4. Layers of LSTM model

- ## Embedding

The "Embedding" layer encodes the input sequence into a sequence of dense vectors of dimension embedding.

- ## Spatial Dropout

This version performs the same function as Dropout, however, it drops entire 1D feature maps instead of individual elements. If adjacent frames within feature maps are strongly correlated (as is normally the case in early convolution layers) then regular dropout will not regularize the activations and will otherwise just result in an effective learning rate decrease. In this case, SpatialDropout1D will help promote independence between feature maps and should be used instead.

- ## LSTM

The LSTM transforms the vector sequence into a single vector of size lstm_out, containing information about the entire sequence.

- ## Dense Layer

Dense layer is the regular deeply connected neural network layer. It is the most common and frequently used layer. Dense layer does the below operation on the input and returns the output.

output = activation (dot(input, kernel) + bias)

- input represent the input data
- kernel represent the weight data
- dot represent numpy dot product of all input and its corresponding weights
- bias represent a biased value used in machine learning to optimize the model
- Activation represents the activation function.

- ## Why LSTM?

LSTM outperforms the other models when we want our model to learn from long term dependencies. LSTM's ability to forget, remember and update the information pushes it one step ahead of RNNs.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

4. Evaluation Metrics

For evaluation of the sentiment labels obtained from Neural Network based sentiment analyzer, the same test data was labelled using standard sentiment analysis tool. The performance of Neural Network model built for sentiment analysis on streaming tweets is estimated using a set of evaluation metrics such as Final Accuracy, Precision Score, Recall Score and F1 Score.

Final Accuracy is a measure of correct classifications, which can be calculated as, the number of correct classifications made by the classifier, divide by the total number of test samples. Precision Score summarizes the fraction of samples assigned the positive class that actually belong to the positive class. Recall Score summarizes how well the positive class was predicted and is the also referred as sensitivity score. F1-score is calculated by taking the harmonic mean of precision score and recall score. It is a metric used to find an equal balance between precision and recall, which is very useful in scenarios when we are working with imbalanced datasets [6]. These values of evaluation metrics can be obtained for each individual class and also for the entire test data. The values obtained considering the proportion of each class in the test data are termed as weighted average scores and those obtained without considering proportion of each class are termed as Macro average scores [7].

Results

The variation of accuracy observed and loss observed with epochs during the training phase of building the Neural Network based sentiment analyzer is in the graphs below.

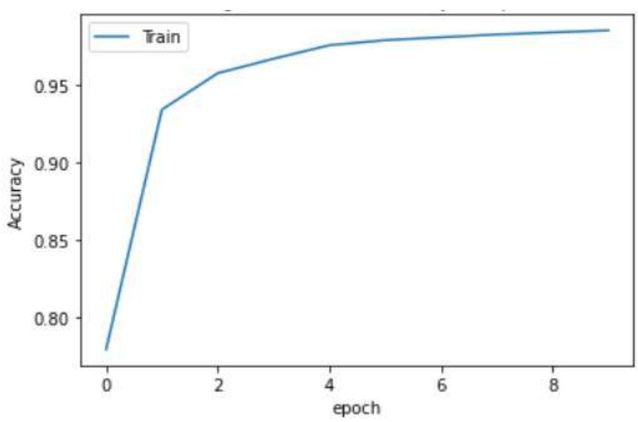

Figure 5. The plot of Model accuracy vs epochs during Training phase

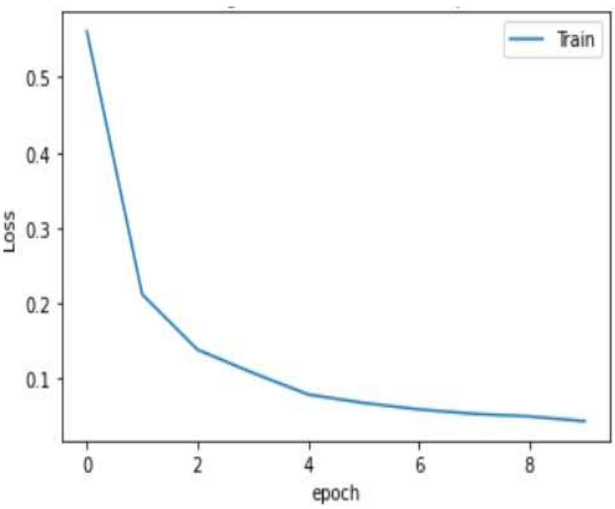

Figure 6. The plot of Model loss vs epochs during Training phase

Later this sentiment analyzer built was tested on the stock related tweets extracted from twitter, and overall an accuracy of around 63.6248% was observed on this testing data. The overall a weighted average F1 score of 0.64 is obtained on this testing data.

The sentiment classification by the model built is visualized using the Confusion matrix and the Table 1 is representing the corresponding evaluation metrics obtained is also shown below.

Few other graphs are used to visualize the predicted sentiment labels obtained for five days which include- The bar plot in figure 7 represents the number of tweets of Nifty50 stock obtained per each day categorized by the predicted sentiment label value. It can be observed that the majority of tweets obtained are classified as Neutral sentiment and the minimum number of tweets belong to Negative sentiment class.

Evaluation Metric	Class 0 (Negative sentiment)	Class 1 (Neutral sentiment)	Class 2 (Positive sentiment)	Macro average	Weighted average
F1 Score	0.50	0.60	0.72	0.61	0.64
Precision score	0.81	0.44	0.87	0.71	0.75
Recall Score	0.36	0.95	0.62	0.65	0.64

Accuracy: 63.6248 %

Table 1. Table showing the evaluation metric scores obtained on testing data

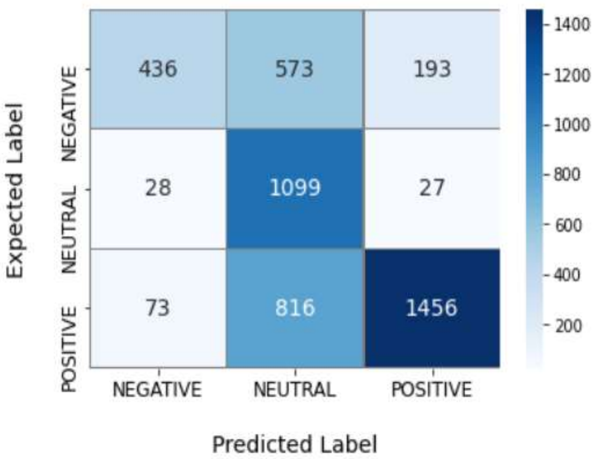

Figure 7. Confusion matrix

The plot in figure 8 shows the variation of Price change percentage of Nifty 50 stock and the corresponding overall Positive sentiment ratio obtained in the period of five days. The overall positive sentiment ratio is calculated as total positive sentiment tweets subtracted from the negative sentiment tweets divided by the total count of positive or negative sentiment tweets.

The bar plot in figure 9 represents the number of tweets of Mahindra & Mahindra Limited Company stock obtained per each day categorized by the predicted sentiment label value. It can be observed that the majority of tweets obtained are classified as Neutral sentiment and the minimum number of tweets belong to Negative sentiment class.

The plot in figure 10 shows the variation of Price change percentage of Mahindra & Mahindra Limited Company stock and the corresponding overall Positive sentiment ratio for five days period. From the graph it can be seen the sentiment of tweets not highly correlated to the stock price change percentage.

Correlation of tweets with stock market

This section describes how the tweets are correlated with the stock market price. For that we have considered live streaming tweets from twitter and stock prices from 'investing.com'. Our implemented LSTM model predicts the sentiment of each tweet. For each day we sum up the same sentiment label tweets count and create a dataset for total counts of each sentiment per day. Then we find the difference between total positive tweets count and total negative tweets count from all tweets for each day. This will give how many positive tweets are occurring more than the occurrence of negative tweets. So, this equation will give a value in the range of 0 to 1. The change percentage of stock price is considered here to correlate it with the positive sentiment ratio. Here, change percentage is the percentage of change in previous stock price from the current price. Based on these two percentage values we try to correlate stock price changes and sentiments for particular company for each day.

5. CONCLUSION AND FUTURE SCOPE

Sentiment analysis emerges as a challenging field with lots of obstacles as it involves natural language processing. It has a wide variety of applications that could benefit from its results, such as news analytics, marketing, question answering, readers do. Getting important insights from opinions expressed on the internet especially from social media blogs is vital for many companies and institutions, whether it is in terms of product feedback, public mood, or investor's opinions.

In this paper we investigated the sentiment of online tweets. Manual analysis of large amounts of such data is very difficult, so a reasonable need for their computer processing has emerged. We used a combination of different pre-processing methods to reduce the noise in the text. We have used LSTM model with 3 inner layers. We have reported extensive experimental results, showing that, if the values of parameters which are used in LSTM model and the model training purpose. An accuracy of 63.62% was achieved and overall a weighted average F1 score of 0.64 is obtained.

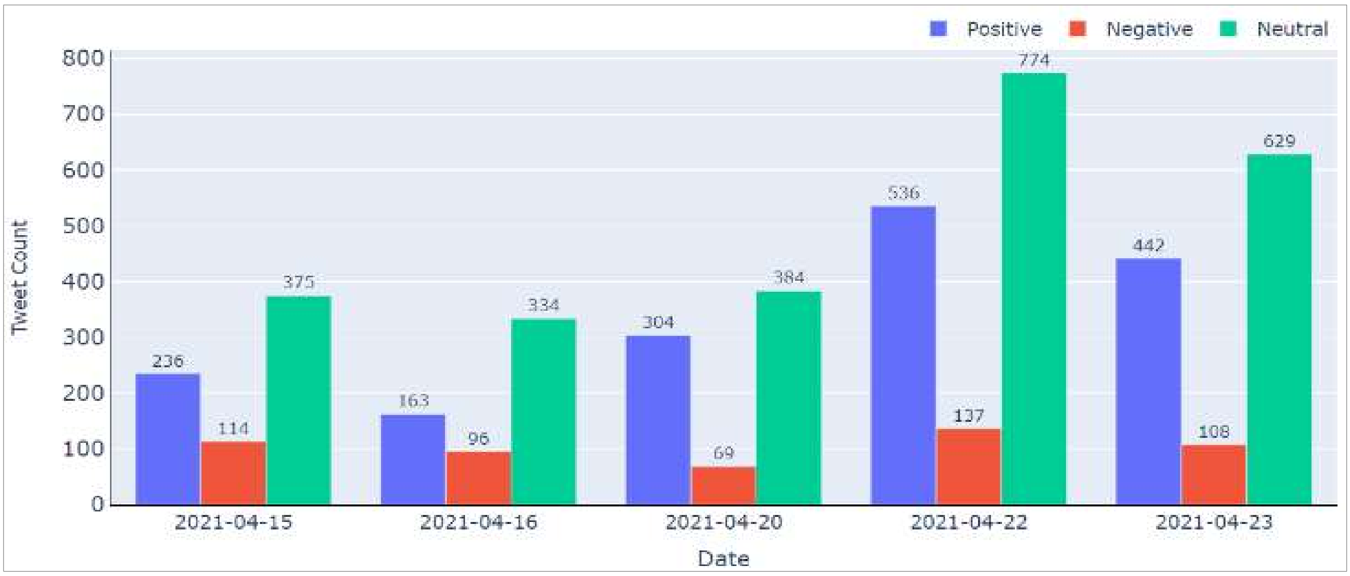

Figure 8. Count of Positive, Negative, Neutral tweets of Nifty 50 stock observed for five days

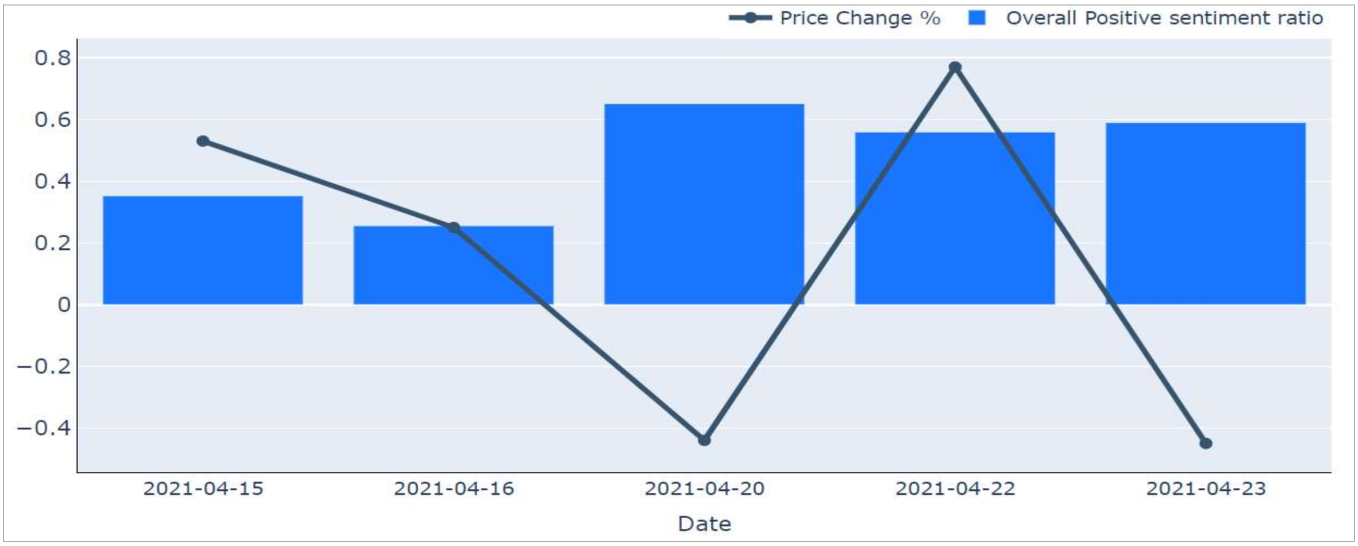

Figure 9. Plot of Price change percentage of Nifty 50 stock and the corresponding overall Positive sentiment ratio

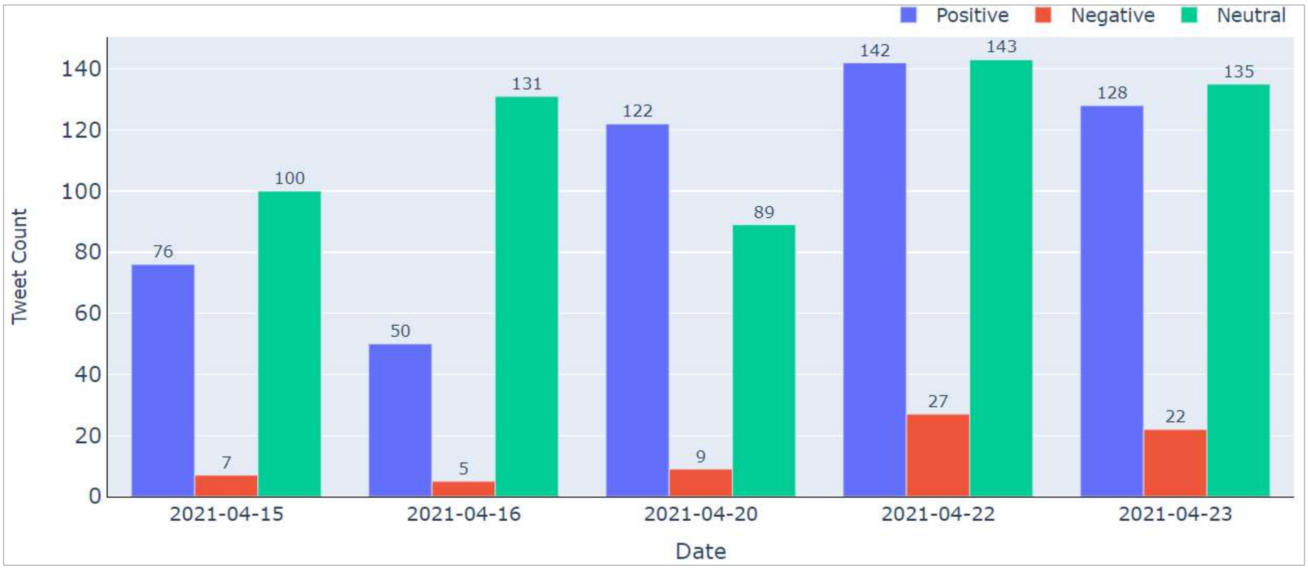

Figure 10. Count of Positive, Negative, Neutral tweets of Mahindra & Mahindra Limited Company observed for five days

Figure 11. Plot of Price change percentage of Mahindra & Mahindra Limited Company and the corresponding overall Positive sentiment ratio

Based on the output graphs shown below we can conclude that sentiments in tweets is somewhat related to the of the company. The future scope include extension of current work to prediction of stock market price from the sentiment of real time tweets can be implemented using LSTM model.

6. REFERENCES

[1] Link of training dataset from kaggle https://www.kaggle.com/vivekrathi055/sentiment-analysis-on-financial-tweets

[2] Hoch Reiter S, Schmidhuber, J. (1997). Long short-term memory. In: Neural Computation 9(8): 17351780. https://doi.org/10.1162/neco.1997.9.8.1735

[3] Ain QT, Ali M, Riazy A, Noureenz A, Kamranz M, Hayat B, Rehman (2017). Sentiment analysis using deep learning techniques: A review. In: International Journal of Advanced Computer Science and Applications (IJACSA) 8(6): https://doi.org/10.14569/IJACSA.2017.080657

[4] Choosing the right Hyperparameters for a simple LSTM using Keras Karsten Eckhard. https://towardsdatascience.com/choosing-the-right-

hyperparameters-for-a-simple-lstm-using-keras-f8e9ed76f046

[5] Reading between the layers (LSTM Network) Using PyTorch framework for Deep Learning Samarth Agrawal. https://towardsdatascience.com/reading-between-the-layers-lstm-network-7956ad192e58

[6] Hossin, M. and Sulaiman, M.N, March 2015, A Review on Evaluation Metrics for Data Classification Evaluations, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2

[7] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, 2008 Introduction to Information Retrieval- Evaluation of text classification, Cambridge University Press.