Stock Prediction Using Twitter Sentiment Analysis

*Note: Sub-titles are not captured in Xplore and should not be used

Nithin Teja Reddy Gottam
Deparment of Information Science
Univeristy of North Texas
Denton, USA
Nithintejareddygott@my.unt.edu

Rushitha Kondreddy
Department of information science
Univeristy of North Texas
Denton, USA
rushithakondreddy@my.unt.edu

Sai Kishore Addala
Department of information science
Univeristy of North Texas
Denton, USA
Saikishoreaddala@unt.edu

Haihua Chen
Department of information science
Univeristy of North Texas
Denton, USA
Haihua.Chen@unt.ed

Abstract— Stock market prediction has been an active area of research for a long time. The Efficient Market Hypothesis (EMH) states that stock market prices are largely driven by new information and follow a random walk pattern. Though this hypothesis is widely accepted by the research community as a central paradigm governing the markets in general, several people have attempted to extract patterns in the way stock markets behave and respond to external stimuli. In this paper we are predicting stock value using twitter data. For prediction first we are extracting tweets from twitter for the particular stock and then we are doing sentiment analysis on this data using transformer architecture. At the same time, we are extracting the last 10 days price using yahoo finance. After getting all the data we are finding the standard deviation of data and keeping one weightage for the sentiment to predict using previous open price, standard deviation and weightage of sentiment and its value.

GitHub link: https://github.com/nithin24011/Secion002-Group1

Keywords: Stock Market, Twitter, Yahoo Finance, Sentiment Analysis, Standard Deviation.

I. INTRODUCTION

Stock exchange is a subject that is highly affected by economic, social, and political factors. There are several factors e.g., external factors or internal factors which can affect and move the stock market. Various Data mining techniques are frequently involved to solve this problem. But technique using machine learning will give more accurate, precise and simple way to solve such issues related to stock and market prices. “Stock Price Prediction Using Twitter Sentiment Analysis” a method for predicting stock prices is developed using news articles. The changes in stock prices of a company, the rises and falls, are correlated with the public opinions being expressed in tweets about that company. Understanding author’s opinion from a piece of text is the objective of sentiment analysis.
Sentiment analysis (or opinion mining) is a natural language processing technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback and understand customer needs.

Sentiment analysis is the process of detecting positive or negative sentiment in text. It is often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers. Since customers express their thoughts and feelings more openly than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment. Automatically analyzing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their

customers' needs.

For example, using sentiment analysis to automatically analyze 4,000+ reviews about your product could help you discover if customers are happy about your pricing plans and customer service. Maybe you want to gauge brand sentiment on social media, in real time and over time, so you can detect disgruntled customers immediately and respond as soon as possible. The applications of sentiment analysis are endless. Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), urgency (urgent, not urgent) and even intentions (interested v. not interested). Depending on how you want to interpret customer feedback and queries, you can define and tailor your categories to meet your sentiment analysis needs.

II. LITERATURE REVIEW

The applications are endless and some of them are mentioned below:

A. *Emotion detection:*

This type of sentiment analysis aims to detect emotions, like happiness, frustration, anger, sadness, and so on. Many emotion detection systems use lexicons (i.e., lists of words and the emotions they convey) or complex machine learning algorithms. One of the downsides of using lexicons is that people express emotions in different ways. Some words that typically express anger, like bad or kill (e.g., your product is so bad, or your customer support is killing me) might also express happiness (e.g. this is bad ass or you are killing it)

B. Aspect-based Sentiment Analysis:

Aspect-based sentiment analysis (ABSA) is a text analysis technique that categorizes data by aspect and identifies the sentiment attributed to each one. Aspect-based sentiment analysis can be used to analyze customer feedback by associating specific sentiments with different aspects of a product or service.

When we talk about aspects, we mean the attributes or components of a product or service e.g., "the user experience of a new product," "the response time for a query or complaint," or "the ease of integration of new software."

Here is a breakdown of what aspect-based sentiment analysis can extract:

- *Sentiments***:** positive or negative opinions about a particular aspect
- *Aspects***:** the category, feature, or topic that is being talked about

Usually, when analyzing sentiments of texts, let us say product reviews, you'll want to know which particular aspects or features people are mentioning in a positive, neutral, or negative way. That is where aspect-based sentiment analysis can help, for example in this text: "The battery life of this camera is too short", an aspect-based classifier would be able to determine that the sentence expresses a negative opinion about the feature battery life.

C. Multilingual sentiment analysis:

Multilingual Sentiment Analysis: An RNN-Based Framework for Limited Data. ... For this purpose, we train a sentiment analysis model using recurrent neural networks with reviews in English. We then translate reviews in other languages and reuse this model to evaluate the sentiments.

Multilingual sentiment analysis can be difficult. It involves a lot of preprocessing and resources. Most of these resources are available online (e.g., sentiment lexicons), while others need to be created (e.g., translated corpora or noise detection algorithms), but you'll need to know how to code to use them.

Background-

Hardware requirements:

1. I5 Windows System
2. 4 core cpu machine

Software requirements:

1. Python 3.7
2. Anaconda
3. Pycharm
4. Tweepy developer Keys
5. Transformer Library
6. yfinance Library

Ideology:

Sentiment analysis algorithms fall into one of three buckets:
Rule-based: these systems automatically perform sentiment analysis based on a set of manually crafted rules.
Automatic: systems rely on machine learning techniques to learn from data.
Hybrid systems combine both rule-based and automatic approaches.

Rule-based Approaches:

Rule-based approaches are the oldest approaches to NLP. Why are they still used, you might ask? It's because they are tried and true and have been proven to work well. Rules applied to text can offer a lot of insight: think of what you can learn about arbitrary text by finding what words are nouns, or what verbs end in -ing, or whether a pattern recognizable as Python code can be identified. Regular expressions and Conent Free Grammars are textbook examples of rule-based approaches to NLP.
Rule-based approaches:

- tend to focus on pattern-matching or parsing.
- can often be thought of as "fill in the blanks" methods.
- are low precision, high recall, meaning they can have high performance in specific use cases, but often suffer performance degradation when generalized.

Usually, a rule-based system uses a set of human-crafted rules to help identify subjectivity, polarity, or the subject of an opinion. These rules may include various NLP techniques developed in computational linguistics, such as: Stemming, tokenization, part-of-speech tagging and parsing. Lexicons (i.e., lists of words and expressions). Here is a basic example of how a rule-based system works: Defines two lists of polarized words (e.g., negative words such as bad, worst, ugly, etc and positive words such as good, best, beautiful, etc). Counts the number of positive and negative words that appear in each text.
If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral sentiment.
Rule-based systems are very naive since they don't take into account how words are combined in a sequence. Of course, more advanced processing techniques can be used, and new rules added to support new expressions and vocabulary. However, adding new rules may affect previous results, and the whole system can get very complex. Since rule-based systems often require fine-tuning and maintenance, they will also need regular investments.

Automatic Approaches:

Automatic methods, contrary to rule-based systems, don't rely on manually crafted rules, but on machine learning techniques. A sentiment analysis task is usually modeled as a classification problem, whereby a classifier is fed a text and returns a category, e.g., positive, negative, or neutral. Here's how a machine learning classifier can be implemented:

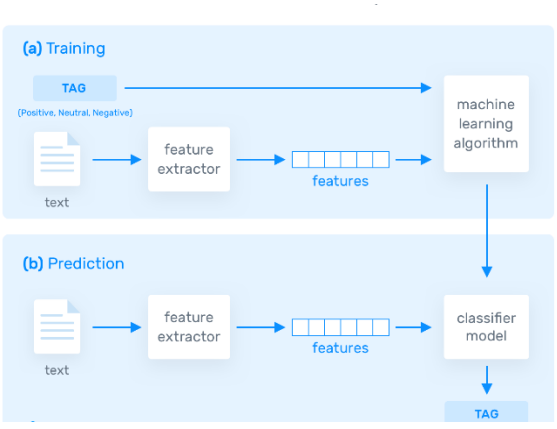

Fig. Implementation of Machine Learning Classifier

In the training process (a), our model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the test samples used for training. The feature extractor transfers the

text input into a feature vector. Pairs of feature vectors and tags (e.g. positive, negative, or neutral) are fed into the machine learning algorithm to generate a model. In the prediction process (b), the feature extractor is used to transform unseen text inputs into feature vectors. These feature vectors are then fed into the model, which generates predicted tags (again, positive, negative, or neutral).

III. METHODOLOGY:

We also used transformer architecture to predict sentiment. which is as follows:
Deep learning models are introduced at an increasing rate and sometimes it's hard to keep track of all the novelties. That said, one particular neural network model has proven to be especially effective for common natural language processing tasks. The model is called a Transformer. The paper 'Attention Is All You Need' introduces a novel architecture called Transformer. As the title indicates, it uses the attention-mechanism we saw earlier. Like LSTM, Transformer is an architecture for transforming one sequence into another one with the help of two parts (Encoder and Decoder), but it differs from the previously described/existing sequence-to-sequence models because it does not imply any Recurrent Networks (GRU, LSTM, etc.).

Implementation:

IV. DATA COLLECTION & PREPROCESSING

we extracted the tweets from Twitter API using the Tweepy library. For using the Twitter API you need to have a developer access Twitter account. Request for the same it might take 2–3 hours to get an approval. Once, you're done with the set up create an app, in it, you will get Keys and tokens, which will help us retrieve data from Twitter. They act as login credentials. Now, we will get on to code to get the tweets. First, you need to import all the packages required and initialize the token and key variables. we define a variable called Auth, which essentially allows the user to give another website/service a limited access authentication token for authorization to additional resources.

Recurrent Networks were, until now, one of the best ways to capture the timely dependencies in sequences. However, the team presenting the paper proved that an architecture with only attention-mechanisms without any RNN (Recurrent Neural Networks) can improve on the results in translation task and other tasks! One improvement on Natural Language Tasks is presented by a team introducing BERT: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

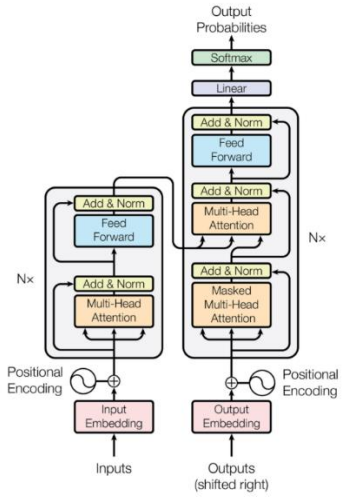

Figure 1: The Transformer - model architecture.

```
consumer_key    = '3jmA1BqasLHfItBXj3KnAIGFB'
consumer_secret = 'imyEeVTctFZuK62QHmL1I0AUAMudg5HKJDfkx0oR7oFbFinbvA'

access_token  = '265857263-pF1DRxgIcxUbxEEFtLwLODPzD3aMl6d4zOKlMnme'
access_token_secret = 'uUFoOOGeNJfOYD3atlcmPtaxxniXxQzAU4ESJLopA1lbC'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)
```

A dataset consisting of specific tweets from the past 10 days, pertaining to specific stocks was chosen to perform the sentiment analysis. First, a web scraper was developed to handle the format of the tweet links that needed to be scrapped. Next, the tweets were fetched tweepy API. The entire text of the articles was fetched from the specific URL's using in Python. The scraper was designed in such a way that only

articles from the previous 10 days were stored for the stock.

```
fetch_tweets=tweepy.Cursor(api.search, q="#unitedAIRLINES",count=100, lang ="en",since="2021-4-13", tweet_mode="extended").items()
data=pd.DataFrame(data=[[tweet_info.created_at.date(),tweet_info.full_text]for tweet_info in fetch_tweets],columns=['Date','Tweets'])
```

Also, retweets having keywords like 'rt','@rt' etc. were removed from the respective stocks. Since there are high chances of finding duplicate news tweets, only unique news tweets were selected to form the news corpus. The text was then preprocessed to handle punctuation, special characters, and white spaces by simply removing them from the text.

```
cdata['Tweets']=cdata['Tweets'].apply(lambda x:re.sub(r'[^\w]',' ',x))
cdata['Tweets']=cdata['Tweets'].apply(lambda x:re.sub(r'[^a-zA-Z]',' ',x))
cdata['token']=cdata['Tweets'].apply(lambda x:x.split())
cdata['token']=cdata['token'].apply(lambda x : [i.lower() for i in x ] )
cdata['token']=cdata['token'].apply(lambda x : [i for i in x if not i.startswith('x')] )
redundant_words=['rt','nrt']
cdata['token']=cdata['token'].apply(lambda x : [i for i in x if i not in redundant_words]
cdata['token']=cdata['token'].apply(lambda x : [i for i in x if len(i)>3])
cdata['Tweets']=cdata['token'].apply(lambda x:' '.join(x))
cdata=cdata.drop(columns='token')
```

After getting access to Twitter data, we'll now create a csv file to save all the extracted tweets in it.

```
data.to_csv("Tweets.csv")
cdata=data
```

The figure below shows the csv file created which contains the tweets extracted which was grouped by date, that particular tweet was posted on.

	Date	Tweets
0	2021-04-27	another flight books flightsim avgeek msfs pla...
1	2021-04-26	someone explain purpose social distancing airp...
2	2021-04-25	well unitedairlines unitedmileageprogram used ...
3	2021-04-24	afbranco branco cartoon woke skies https czanu...
4	2021-04-23	ohhh good news sissy going back first love hah...
5	2021-04-22	have recently launched podcast called wingtips...
6	2021-04-21	https kmryre learn this flight expansion angel...
7	2021-04-20	chrisli united rhapsody blue flying into after...
8	2021-04-19	mclovinhanson wiseguy peppermintmom prayerful ...

Fig. Extracted Tweets Data

Data Transformation and Model Configuration

A Python library was used for transforming the text corpus into numerical vectors. For each tweet, the main text was extracted and transformed into n-grams. Initially unigrams (where the number of word tokens is one) were used for the sentiment analysis. However, the unigrams did not capture the context of the text. For instance, the word 'decline' is generally a negative word, however, if the phrase is 'costs declined' then it is positive for the company. Eventually, bigrams (where the number of word tokens is two) and trigrams (where the number of word tokens is three) were created from the text corpus. These helped to understand the polarity of individual words and the surrounding phrases as well, which allowed capturing the context in a much better way. Stemming i.e., reducing words to their base forms such as 'declining' to 'decline' is preferred for a phrase and was applied to processing bigrams and trigrams.

Due to the unavailability of dictionaries specific to the financial markets, we manually created a dictionary that contains words and phrases which carried specific meaning pertaining to stock. The dictionary was created by leveraging domain expertise and thorough analysis of news articles over the days. The dictionary consisted of 100 words which cover a broad spectrum of topics specific to the stock. Each entry in the dictionary was tagged with positive, negative, or neutral class of sentiment. The generated n-grams were then compared against the dictionary, and if a match was found then each matching word was assigned a positive or negative polarity. The positive and negative words were then counted, and the sentiment scores were generated based on their frequency. For instance, if there are three positive words then the sentiment would be +3. Words with neutral sentiment do not affect the score. For all the news articles, the scores were cross validated against the stock prices to understand their effects.

	Date	Tweets	Open	label
0	2021-05-02	unitedairlines inexplicably cancelling flights...	53	POSITIVE
1	2021-05-01	delta delta seriously americanairlines regular...	53	NEGATIVE
2	2021-04-30	elizabetb tried over from kosovo through vienn...	53	NEGATIVE
3	2021-04-29	librarian spring united airlines timetable pos...	54	NEGATIVE
4	2021-04-28	phillipsa please support companies that fight ...	53	POSITIVE
5	2021-04-27	believe these overhead luggage space rendered ...	53	NEGATIVE
6	2021-04-26	someone explain purpose social distancing airp...	54	NEGATIVE
7	2021-04-25	well unitedairlines unitedmileageprogram used ...	53	NEGATIVE
8	2021-04-24	afbranco branco cartoon woke skies https czanu...	53	NEGATIVE

Fig. Emotion Detection

V. DATA ANALYSIS

Transformers is an opinionated library built for NLP researchers seeking to use/study/extend large-scale transformers models. The library was designed with a strong goal, that's to be as easy and fast to use as possible. we used transformers library to implement sentiment analysis quickly and effectively. We used pretrained transformers rather than fine-tuning our own.

We have installed transformers library first and Instead of importing the entire library.

After we have successfully installed Transformers to our local environment, we created a new Python script and imported the Transformers library. Instead of importing the entire library, we introduced the pipeline module within the library that provides a simple API to perform various NLP tasks and hides all code complexity behind its abstraction layer.

Now after you import the pipeline module, we can start building the sentiment analysis model and tokenizer using the module. To build it, we did:

sentiment_analysis = pipeline("sentiment_analysis")

This will create a pipeline suited for the sentiment analysis task. Wait, but what model and tokenizer are used here, you might ask. Well, by default, the Transformers library uses DistilBERT model that was fine-tuned on the Stanford Sentiment Treebank(V2) task from the GLUE dataset.

If you want to use another model or tokenizer, you can pass them as the model and tokenizer parameters when you instantiate the pipeline.

We have built up our pipeline, now it is time for us to input the text we want to test for its sentiment. We have declared two variables for two sentences, one positive and one negative.

VI. RESULTS & DISCUSSION:

We have implemented application in number of phases. We would like to describe our results and discuss about results in following manner.

In first phase of task, we have collected stock tweets using twitter API. It responses tweets in JSON format. It is mandatory to keep configuration file as an authentication to download tweet data live.

The indispensable configuration parameters are:

{

"consumer_key":"",
"consumer_secret":"",
"access_token":"",
"access_token_secret":""

}

After fetching twitter corpus, we have store it in text file. In parallel we process our tweets for further analysis purpose. To process tweet, we have followed some basic steps:

1. Tokenized tweets
2. Remove extra keyword from tweets
3. Remove stop words from tweet 4. Replace URLs and meaningful notation with meaning full keywords.

To prepare training set we have used AFINN library. It computes tweet keywords positive and negative scores and result tweet sentiment with positive, negative or neutral score. We have map tweet score with positive, negative or neutral keyword. If total tweet score is greater than 0 then it considers as positive keyword. If total tweet score is less than 0 then it considers as negative. If total tweet score is 0 then it is considered as neutral. For example, the processed tweet is "Oracle, #Google fail to settle Android lawsuit before retrial". The total sentiment score of this tweet -4. Hence it is considered as negative tweet. The generated tweet corpus csv file contains tweet date, tweet text, tweet sentiment keyword and total tweet score. We have prepared dataset using generated tweet corpus csv file.

Our next step is to predict tweet sentiment using transformer architecture which is explained above. We kept a weight for the sentiment and predicted next value using the previous value with weight and standard deviation.

The formula used was:
xnext=xprev+std*weight*value

The final output gives map value of only 5 percent and the graph for actual vs pred is shown below. MAPE indicates the Mean Absolute Percentage Error between our predicted values and the actual normalized stock values. So, the error was only 5 that means we predicted it 95% right.

VII. CONCLUSION

After analyzing tweet sentiment, we have predicted stock market movement with 95% accuracy. The implemented model works much better for tweet sentiment analysis. Performing whole task, I come to conclude that we should work more on twitter and yahoo dataset which help us to improve our model.

The same can be integrated with other web app systems to automatically predict the future values. This study is a niche application of sentiment analysis in gauging the effects of tweets on stocks for all sectors. One major contribution of this work is a sentiment analysis dictionary. The sentiment scores obtained from the analysis of the tweets is a powerful indicator of stock movements and can be used to effectively leverage the prediction of short-term trends. We believe that the reason the model can achieve an accuracy of 95% with approach is that statistical analysis was leveraged.

VIII. LIMITATIONS AND FUTURE WORK

In this paper, we investigated whether public sentiment, as measured from tweets, is correlated or even predictive of stock values and specifically for UNITED AIRLINES according to Yahoo! Finance. Our results show that changes in the public sentiment can affect the stock market. That means that we can indeed predict the stock market with high chances. Furthermore, it is worth mentioning that our analysis does not take into consideration many factors. First, our dataset does not really extract the real public sentiment, it only considers the twitter using, English speaking people.
Secondly, the bigger the dataset is, the better the prediction but at the same time the problem gets more complicated.
There are many areas in which this work could be expanded in the future. With a longer period of time and more resources, there is much potential in the field. If it is possible, we would want to collect data over the course of a few years, both from Twitter and the stock market. In addition, we could investigate intraday stock changes in order to make our prediction more accurate. Finally, in the future we could create a stock lexicon based on the most common words used.
A word weighting approach can be implemented in the sentiment analysis pipeline to determine which word is more important and then assign the sentiment score accordingly. Since domain specific sentiment lexicon cannot be easily scaled manually, expansion of the dictionary leveraging well established lexicons would make our model scalable for other sectors as well. Lastly, creating a hybrid model using traditional statistical classification models such as Auto Regressive Integrated Moving Average (ARIMA) and machine learning models such as the long short-term memory (LSTM) neural network model, using stock prices and sentiment analysis model based on social media and news data, may provide more accurate predictions of long and short-term stock price movements.

AUTHOR CONTRIBUTION

Nithin Teja Reddy Gottam worked on Data cleaning and Sentiment Analysis, Rushitha Kondreddy wrote the Reports and worked on Data Visualization and Sai Kishore worked on Data Extraction he wrote the First Derivable.

REFERENCES

[1] L. Zhang, “Sentiment analysis on twitter with stock price and significant keyword correlation,” *Utexas.edu*. [Online]. Available: https://apps.cs.utexas.edu/tech_reports/reports/tr/TR-2124.pdf. [Accessed: 02-May-2021].

[2] A. Mittal and A. Goel, “Stock Prediction Using Twitter Sentiment Analysis,” *Stanford.edu*. [Online]. Available: http://cs229.stanford.edu/proj2011/GoelMittal StockMarketPredictionUsingTwitterSentimentAnalysis.pdf. [Accessed: 02-May-2021].

[3] M. Blondel, A. Fujino, and N. Ueda, “Large-scale multiclass support vector machine training via euclidean projection onto the simplex,” in *2014 22nd International Conference on Pattern Recognition*, 2014.

[4] G. Rizzo and R. Troncy, : “Nerd: Evaluating named entity recognition tools in the web of data,” vol. 21, 2011

[5] A. E. Stefano Baccianella and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In LREC. LREC.

[6] A. Lapedes and R. Farber. Nonlinear signal processing using neural network: Prediction and

system modeling. In Los Alamos National Lab Technical Report.

[7] G. P. Gang Leng and T. M. McGinnity. An on-line algorithm for creating self-organizing fuzzy neural networks. Neural Networks, 17(10):1477–1493

[8] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

[9] J. Bollen and H. Mao. Twitter mood as a stock market predictor. IEEE Computer, 44(10):91–94.