

Stock Market Analysis using Twitter (Sentiment Analysis)

A PROJECT REPORT

Submitted

in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

ABHINAV REDDY AYYADAPU (19B81A05Q5)

ABHISHEK NAMALA (19B81A05Q4)

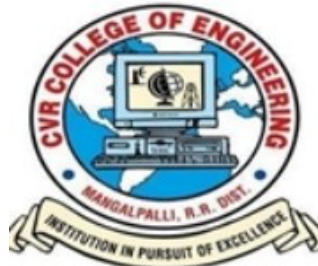
SAI KUMAR AKKINI (19B81A05U2)

Under the guidance of

Dr. K. MADHUSUDHANA

Dr. R. K. SELVAKUMAR

Associate Professor, CSE Department



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

*(An Autonomous institution, NBA, NAAC Accredited and Affiliated to
JNTUH, Hyderabad)*

Vastunagar, Mangalpalli (V), Ibrahimpatnam
(M), Rangareddy (D), Telangana- 501 510

October 2022

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey our profound sense of gratitude to our principal **Dr.K. Ramamohan Reddy**, Vice-Principal **Prof. L. C. Siva Reddy**, **Dr. A. Vani Vathsala**, Head of CSE Department, CVR College Of Engineering, for having been kind enough to arrange necessary facilities for executing the project in the college.

We deem it a pleasure to acknowledge our sense of gratitude to our project guide **Dr. K.Madhusudhana and Dr. R. K. Selvakumar** under whom we have carried out the project work. His incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish a deep sense of gratitude and heartfelt thanks to the management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

ABHINAV REDDY AYYADAPU (19B81A05Q5)

ABHISHEK NAMALA (19B81A05Q4)

SAI AKKINI (19B81A05U2)

DECLARATION

We the undersigned solemnly declare that the project report titled ‘STOCK MARKET ANALYSIS USING TWITTER’ is based on our own work carried out during the course of our study under the supervision of Dr. K.Madhusudhana and Dr. R. K. Selvakumar, CSE Dept., CVR College of Engineering.

We assert the statements made and conclusions are drawn are an outcome of our research work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or in the any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.

CERTIFICATE

This is to certify that the project titled ‘**Stock market analysis using Twitter**’ is being submitted by **Abhinav Reddy Ayyadapu (19B81A05Q5)**, **Abhishek Namala (19B81A05Q4)**, **Sai Kumar Akkini (19B81A05U2)** in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering to the CVR College of Engineering, is a record bonafide work carried out by them under my guidance and supervision during the year 2022-2023. The results embodied in this project work have not been submitted to any other University or Institution for the award of any degree or diploma.

Signature of the project guide,

Dr. R. K. Selvakumar

Dr. K. Madhusudhana

Associate Professor

CSE Department

Signature of the HOD,

Dr. A. Vani Vathsala

Head of Department (CSE)

CVR College of Engineering

ABSTRACT

In a study, it was investigated relationship among stock market movement and Twitter feed content. We are expecting to see if there is connection among sentiment information extracted from the Tweets using a Vader in predicting movements of stock prices.

Predicting stock market movements is a well-known problem of interest. Now-a-days social media is perfectly representing the public sentiment and opinion about current events. Especially, Twitter has attracted a lot of attention from researchers for studying the public sentiments.

So, we apply sentiment analysis and machine learning principles to find the correlation between “public sentiment” and “market sentiment”.

Previous studies have concluded that the aggregate public mood collected from Twitter may well be correlated with Dow Jones Industrial Average Index (DJIA).

Such that, we use twitter data to predict public mood and use the predicted mood and previous days' DJIA values to predict the stock market movements. In an elaborate way, positive news, tweets in social media about a company would definitely encourage people to invest in the stocks of that company and as a result the stock price of that company would increase. At the end of this project, it is shown that a strong correlation exists between the rise and falls in stock prices with the public sentiments in tweets.

TABLE OF CONTENTS

Table of Contents		Page No.
	List of Figures	vii
	List of Abbreviations	ix
1	Introduction	1
1.1	Introduction	1
1.2	Motivation	2
1.3	Problem statement	2
1.4	Project Objectives	2
1.5	Project Report Organization	3
2	Literature Survey	4
2.1	Literature Survey	4
3	Software & Hardware specifications	8
3.1	Functional/Non-Functional Requirements	8
3.1.1	Functional Requirements	8
3.1.2	Non-Functional Requirements	8
3.2	Software requirements	9
3.3	Hardware requirements	9
4	Design	10
4.1	Use case Diagram	10
4.2	Activity Diagram	11
4.3	ER Diagram	12
5	Implementation & Testing	13
5.1	Implementation	13
5.2	Testing	25
6	Conclusion & Future Enhancement	28
6.1	Conclusion	28
6.1	Future Enhancement	28
7	References	29

List of Figures

4.1 Use Case Diagram.....	7
4.2 Activity Diagram	12
4.3 ER Diagram.....	15
5.1 Twitter API.....	18
5.2 Fetching the tweets.....	18
5.3 Tweets converted into Dataset	19
5.4 Clubbed all tweets into respective date.....	20
5.5 Adding prices column in ccdata.....	20
5.6 Making Dataset to perform Sentiment Analysis	21
5.7 Downloading VADAR.....	21
5.8 VADAR performed.....	22
5.9 Dataframe for testing.....	22
5.10 Testing and Training	23
5.11 Random Forest Regression	23
5.12 Final Output Dataset(small)	24
5.13 Large Dataset	24
5.14 Sentiment Analysis.....	25
5.15 Pie chart.....	26
5.16 Random Forest Regression for Large Dataset.....	27
5.17 Final Stage.....	28

List of Abbreviations

CSE	Computer Science Engineering
VADER	Valence Aware Dictionary and sentiment Reasoner
ER	Entity Relationship
SVM	Support Vector Machines

1.INTRODUCTION

1.1 INTRODUCTION

With development of social media, public opinion becomes abundant. Social media is excellent platform for sharing emotions publicly about any subject and as platform has important effect on public opinion. In recent years twitter as a social media become interesting for researchers. As real time information, connects users and inform them about subjects that are interested in. Users need to follow others to receive constant information and updates. It is a great source of data since user's everyday post more than 200 million tweets and maximum size of tweet is 140 characters. There are around 50 million users of tweets, and motives for using that social media differ from user to user: some heir users use it to stay informed, connected to other users or to increase their popularity and awareness. Since limited number of characters to be followed tweet needs to be easy to understand and concise. Single tweet may not look valuable but aggregated tweets analyzed can provide appreciated insight of sentiment and public opinion. Stock market prediction was always challenging as a study, and previous research were based on historical market prices. Well known efficient market hypothesis (EMH) find that prediction of market significantly depends on contemporary events, product releases and news. Since news and contemporary events are unpredictable was proven that market prices follow an arbitrary walk pattern with more than 50% precision. According to behavioral economics people are not rational as customers and decisions are significantly affected by emotions and other people opinion.

Getting public sentiment by retrieving online information from Twittter can be very valuable on market trading. If aggregated tweets about certain companies are used and correlated with economic indicators referring to financial market, it is expected to get interesting information. In this paper we are hoping to collect tweets related to the Microsoft Company and stock prices for the same period of time, then decide the polarity of tweets and check correlation for the tweets and stock prices.

1.2 MOTIVATION

Website is a collection of web pages. It is just a collection of documents or files that you can access through the internet and usually, they look something like, now in order to be able to look at a website people use a program called a browser. The most popular browsers are Chrome Firefox Internet Explorer and Safari. Now a browsers job is to make the code into something that the user can look at and use but where the websites come from how your browser gets the information for them. Having a department website will really help students to reach out faculty and seek their help for their academic development and keep them updated with the latest news.

1.3 PROPOSED STATEMENT

The proposed analysis is to make how stock market is depending upon the social media platform like twitter, facebook etc. Predicting stock market movements is a well-known problem of interest. Now-a-days social media is perfectly representing the public sentiment and opinion about current events. Especially, Twitter has attracted a lot of attention from researchers for studying the public sentiments. We will find out the accuracy between both the original stock prices and tweets from twitter using sentiment analysis.

1.4 PROJECT OBJECTIVES

At the end of this project, it is shown that a strong correlation exists between the rise and falls in stock prices with the public sentiments in tweets. To find the correlation between the tweets and the stock depending on it.

1.5 PROJECT REPORT ORGANIZATION

- i. This report is divided into 6 chapters after this introductory chapter.
- ii. Chapter 2 summarizes functional, non-functional requirements and system requirements along with software and hardware specifications.
- iii. Chapter 3 deals with analysis and design of the proposed model which includes use case diagram etc.
- iv. Chapter 4 encloses Implementation and testing of the proposed model and testing with different scenarios.
- v. Chapter 5 includes conclusion and future work.
- vi. Chapter 6 includes reference.

2.LITERATURE SURVEY

2.1 LITERATURE SURVEY

Sentiment analysis is the most important concept of the research area in various different fields. There are many researchers that studies and aim to identify a method to predict sentiment analysis on different area fields. Social media is popularized source of data which collect useful data such as blogs, micro-blogs, Facebook, Twitter etc.

This covers the evaluation of a system that can be used to predict future stock price based on analysis of social media data. Twitter messages are retrieved in real time using Twitter Streaming API. The large volume of data to be classified using Naive Bayes method for fast training process with a large volume of training data. The stock market prediction should be calculated by using linear regression technique.

This presents the two different textual representations, Word2vec and N-gram, for analysing the public sentiments in tweets. The author applied sentiment analysis and supervised machine learning principles (such as logistic regression, random forest, SMO) to tweets extracted from twitter and analysing the correlation between stock market movement of company and sentiments in tweets. A data can be extracted from twitter API of Tesla using keyword #Tsla, etc.

The authors created a system that predicts stock market movements on a given day, based on time series data and market sentiment analysis. They collect prices for stock from Yahoo! Finance into Excel spreadsheet. For sentiment analysis, they obtained Twitter Census stock Tweets data-set from Info-chimps, a privately held company that offers a “data marketplace”. Naive Bayes Classifier used to analyze sentiment in the tweet data set. The SVM, Logistic and Neural network techniques would be used for predicting market movement.

The sentiment analysis of a product is performed by extracting tweets about products and classifying the tweets that can be as positive and negative sentiment. This paper proposes a hybrid approach which combines unsupervised learning to cluster the tweets and then performing supervised learning methods for classification.

DATASET

A. Social Media

The following features capture useful aspect of Twitter and authors for opinion retrieval

1. URL : Most Tweets containing a link usually give the objective introduction to the links. Additionally, spam in Twitter often contain links. Hence, we use a feature indicating whether a Tweet contains a link in our ranking model.
2. Mention : In a Tweet, people usually use “@” preceding a user name to reply to other users. The text of this Tweet is more likely to be „personal content“.
3. Hashtag : A hashtag refers to a word in the text of the Tweet that begins with the “#” character. It is used to indicate the topic of the Tweet.

Twitter is a social network. The more author information can also be used for the analysis of spammer detection.

1. Statuses : The number of Tweets (statuses) the author has ever written that which related to the activeness of an author. The most active authors are likely to be spammers who post very large number of Tweets. Therefore, we use the number of statuses as a feature for Tweets ranking.
2. Place : The location associated with tweet that help to identify which place the tweet should be posted. In this contains country, country code, id, name of the place.
3. Followers and Friends : In Twitter a user can choose to follow any number of other users that he finds interesting for one reason or another. If userA follows userB, all the Tweets posted by userB will be updated in the userAs private stream. We call userA a follower of userB and userB a friend of userA. The number of followers indicates the popularity of the user. The number of friends also reflects the type of the user.

4. Retweets : Twitter provide services to user to retweet tweets generated by other user. All retweets start with symbol indicated by @RT. The retweet of the most recent tweets of a user is also one of the feature in spam detection system.

5. Listed : A user can group their friends into different lists according to some criteria. If a user is listed many times, it means that his Tweets are interesting to a large user population. We use a feature that measures how many times the author of a Tweet has been listed for Tweet ranking.

B. Classification Techniques

Supervised learning is an important technique for solving classification problems in sentiment analysis. Training and testing the data make it more easier for prediction future data. The different classifiers can be used to classifies the sentiment score of each tweets that predict the emotion of the tweets. The different techniques are compared with their result and accuracy.

1. Naive Bayes :

Naive Bayes is a conditional probability model that given a problem instance to be classified. Naive Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variable.

$$P(c|d) = \frac{(P(c) \sum_{m=1}^m P(f_m|c) P(f_m|c)^{n_i(d)})}{P(d)}$$

In this formula, f_i represents a feature and $n_i(d)$ represents the count of feature f_i found in tweet d . There are a total m features. Parameters $P(c)$ and $P(f|c)$ are obtained through maximum estimates.

2. SVM :

Support Vector machines (SVM) are primarily Classifiers that can classify by constructing hyperplanes that separate cases that belong to different categories. A Support Vector Machine (SVM) is a supervised classification algorithm that recently has been applied successfully to text classification tasks.

$$c(x) = \begin{cases} 1 & w \cdot \phi(x) + b \geq k \\ -1 & w \cdot \phi(x) + b \leq -k \end{cases}$$

where, $w = \{w_1, \dots, w_n\}$ is a weight vector. $x = \{x_1, \dots, x_n\}$ is a input vector.

$\Phi(x)$ is kernel function.

Among these two classifier it is observed that SVM classifier outperforms every other classifier in predicting the sentiment of the tweets.

2. Random Forest:

Random Forest classifier is a tree-based classifier. It consists of numerous classification trees that can be used to predict the class label for a given data point based on the categorical dependent variable. The error rate of this classifier depends on the correlation among any two trees in the forest that adds the strength of definite or individual tree in the forest. In order to minimize the error rate, the trees should be strong and the degree of correlation should be as less as possible.

3.SOFTWARE & HARDWARE SPECIFICATIONS

3.1 Functional/Non-Functional Requirements

3.1.1 Functional Requirements

1. Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows.
2. System should be able to process new tweets stored in database after retrieval.
3. System should be able to analyze data and classify each tweet polarity.

3.1.2 Non-Functional Requirements

1. The performance of the system should be fast and accurate.
2. The system should be able to handle large amount of data. Thus, it should accommodate large number of data entry without any fault.
3. User friendly.
4. To perform with efficient throughput and response time.

3.2 SOFTWARE REQUIREMENT

The following few tools have been used to perform this analysis

- Jupyter Notebook/pycharm/Google colab
- Operating System: Windows OS/MACOS

3.3 HARDWARE REQUIREMENTS

PROCESSOR:

- 4 GHz minimum, multi-core processor

RAM:

- At least 4GB, preferably

HARD DISK SPACE:

- At least 10 GB.

4.DESIGN

4.1 USE CASE DIAGRAM

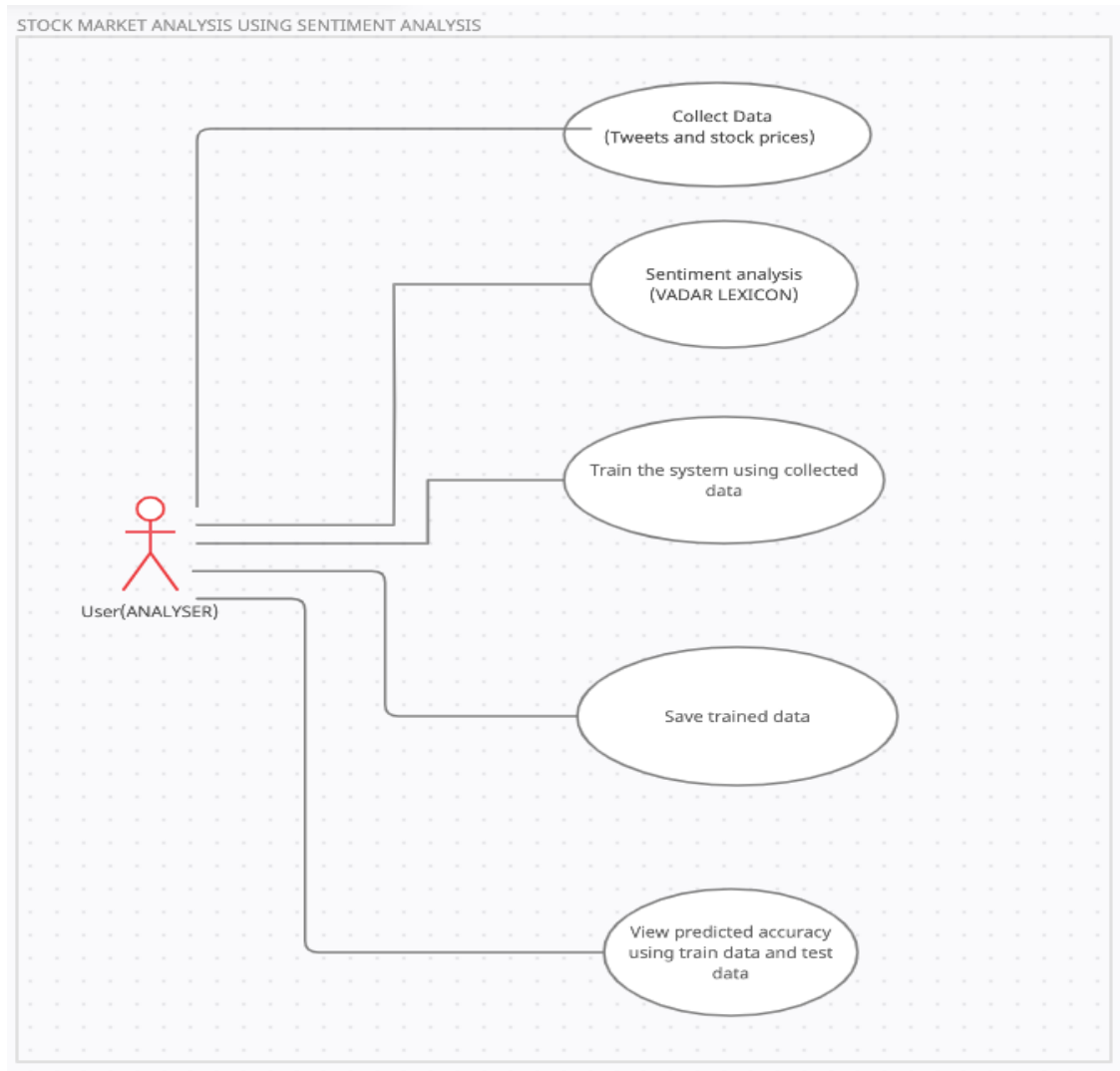


Figure 4.1: Use Case Diagram

In our Use Case Diagram, we have one actor analyzer. The beloved use case diagram shows the set of use case actor and their relationships. The actor analyzer performs the all the use cases mentioned above. The data can be extracted from the twitter and the yahoo finance. The data can be performed for training and testing data.

4.2 ACTIVITY DIAGRAM

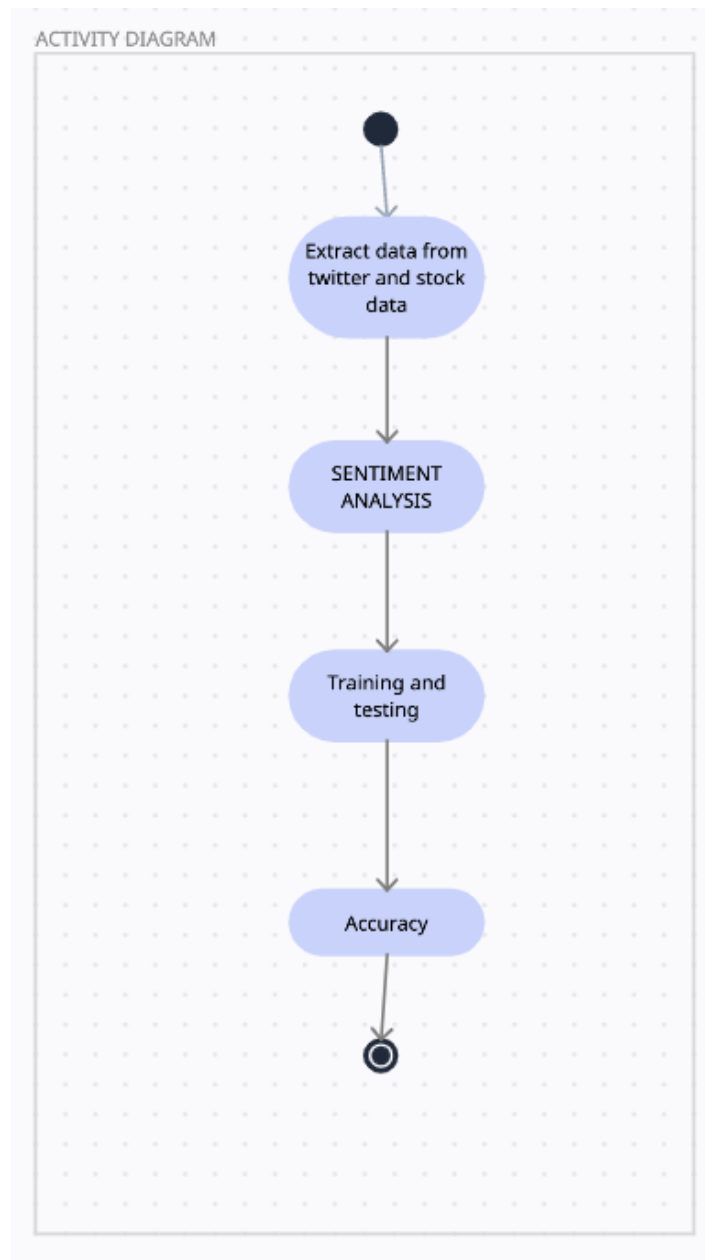


Figure 4.2: Activity Diagram for admin and faculty

The activity diagram provides a view of the classes and what is going on between several classes and use cases. First twitter tweets will be extracted, and particular stock price will be extracted from yahoo finance. After that the sentiment analysis is going to perform by using Vader lexicon. By using regression classifiers, the test and train data are performed to get more accuracy.

4.3 ER DIAGRAM

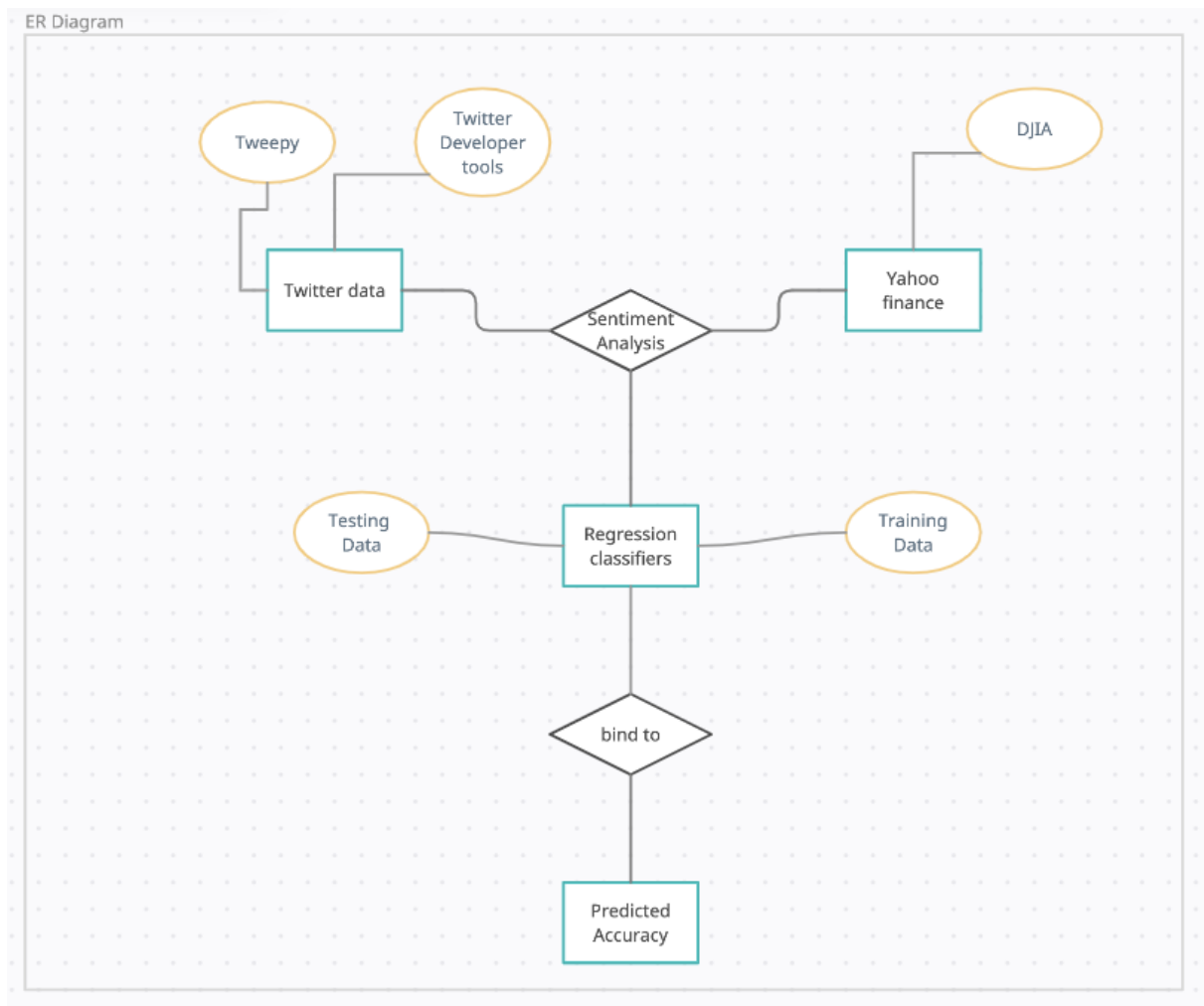


Figure 4.3: ER Diagram

An Entity Relationship illustrates how “entities” such as people, objects or concepts relate to each other within a system. In our ER diagram we have for entities namely twitter data, Yahoo finance, Regression classifiers, predicted accuracy.

5. IMPLEMENTATION AND TESTING

5.1 IMPLEMENTATION

Twitter API

The Twitter developer portal contains a set of self-serve tools that developers can use to manage their access to the Twitter API and Twitter Ads API. In the portal, you have the opportunity to: Create and manage your Twitter Projects and Apps (and the authentication keys and tokens that they provide).

Features of API tools

Aside from the endpoints, let's take a look at some of the salient features of Twitter's API:

- There are four main objects: Tweets, Entities, Places, and Users.
- There are daily restrictions: Calls and changes in the API are restricted by access tokens to protect the platform from abuse.
- It is based on HTTP (rather than SSL).
- There are specific measures to adapt the API operation to the social network including library restrictions, generated paging, and specific parameters.

How to get access to the Twitter API ?

1. Sign up for a developer account.
2. Save your App's key and tokens and keep them secure.
 - API key and Secret
 - Access Token and Secret
 - Client ID and Client Secret
 - App only Access Token
3. Make your first request.

```

▶ consumer_key    = "C2NPH57TPGXs4hEAwAzIKBQa3"
  consumer_secret = "8DGvW2qhVTi9ZiMuauvkiQ13rckmBJZRxVUYBKpraD12xnnG0c"

  access_token    = "2957279347-UUWHf2h9D0YY4IwphStuPDck8kkzIM4m61gTEV0"
  access_token_secret = "BH2hUYHXPELMqqoTxXM4jH3N0WoqkiX2lwn1dq0AZvLHK"

  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)

```

Figure 5.1: Twitter API

Tweepy

Tweepy is an open-source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints, and it transparently handles various implementation details, such as:

- Data encoding and decoding
- HTTP requests
- Results pagination
- OAuth authentication
- Rate limits
- Streams

In this project, we will be using for extracting the tweets from the twitter by using Twitter API.

```

[6] fetch_tweets=tweepy.Cursor(api.search, q="#tsla",count=100, lang="en",since_id="2010-06-29", tweet_mode="extended").ite
    data=pd.DataFrame(data=[[tweet_info.created_at.date(),tweet_info.full_text]for tweet_info in fetch_tweets],columns=['Dat

```

Figure 5.2: Fetching the tweets

In above Figure 5.2 shows how, the tweets are extracted from the Twitter.

The fetched tweets are going to be converted as csv file as followed by data set by having columns Date and Tweets. The data cleaning also done in the process the code mentioned below in **Figure 5.3**.

```
[7] data.to_csv("TweetsTESLA.csv")
cdata=pd.DataFrame(columns=['Date', 'Tweets'])
total=100
index=0
for index,row in data.iterrows():
    stre=row["Tweets"]
    my_new_string = re.sub('[^ a-zA-Z0-9]', '', stre)
    temp_df = pd.DataFrame([data["Date"].iloc[index],
                           my_new_string]), columns = ['Date', 'Tweets'])
    cdata = pd.concat([cdata, temp_df], axis = 0).reset_index(drop = True)
    # index=index+1
    #print(cdata.dtypes)
```

```
[8] cdata
```

	Date	Tweets
0	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...
1	2022-10-13	I aint know Tesla was this cheap Tsla Amzn GooG
2	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...
3	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...
4	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...
...
3531	2022-10-04	Elons letter to Twitter today tsla twtr https...
3532	2022-10-04	RT ThorYoung Went long on TSLA when the order ...
3533	2022-10-04	RT Teslanews10 Kathy Wood buys up Tesla httpst...
3534	2022-10-04	RT akiheikinen Somebody had to do it TSLA Te...
3535	2022-10-04	Looks like a lot of Executives will be on the

Figure 5.3: Tweets converted into Dataset

Now we are clubbed all tweets into single date without date duplicates. These duplicates cause data inconsistency to perform the sentiment analysis.

The original stock prices are collected from the yahoo finance for particular stock extracting in twitter. The csv will be consisting of columns Date, Open, High, Low, Close, Adj Close, Volume. The overall will be present in the csv file.

```
[10] indx=0
get_tweet=""
for i in range(0,len(cdata)-1):
    get_date=cdata.Date.iloc[i]
    next_date=cdata.Date.iloc[i+1]
    if(str(get_date)==str(next_date)):
        get_tweet=get_tweet+cdata.Tweets.iloc[i]+" "
    if(str(get_date)!=str(next_date)):
        temp_df = pd.DataFrame([[get_date,
                                get_tweet]], columns = ['Date','Tweets'])
        cdata = pd.concat([cdata, temp_df], axis = 0).reset_index(drop = True)
    get_tweet=""
```

ccdata

	Date	Tweets
0	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...
1	2022-10-12	RT FinanceLancer We discuss why Chicken Geniu...
2	2022-10-11	RT stevenmarkryan httpstcobXfcmiinc Twitter...
3	2022-10-10	RT BestTrader01 Institutes are Buying TSLA li...
4	2022-10-09	RT theeconomystic The only stock I own and ha...
5	2022-10-08	RT stevenmarkryan httpstcoiBp1y4SaOT Is Tesl...
6	2022-10-07	Open 23394Close 22307Range 22202 23457Its no...
7	2022-10-06	Its finally happening the next evolution in l...
8	2022-10-05	RT DoctorJack16 How to know you are a veteran...

Figure 5.4: Clubbed all tweets into respective date

Now we are clubbed all tweets into single date without date duplicates. These duplicates cause data inconsistency to perform the sentiment analysis.


```
[13] ccddata['Prices']=''
```

```
[14] indx=0
    for i in range (0,len(ccdata)):
        for j in range (0,len(read_stock_p)):
            get_tweet_date=ccdata.Date.iloc[i]
            get_stock_date=read_stock_p.Date.iloc[j]
            if(str(get_stock_date)==str(get_tweet_date)):
                #print(get_stock_date," ",get_tweet_date)
                # ccddata.set_value(i,'Prices',int(read_stock_p.Close[j]))
                ccddata['Prices'].iloc[i] = int(read_stock_p.Close[j])
```

```
[15] ccddata
```

	Date	Tweets	Prices
0	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...	
1	2022-10-12	RT FinanceLancer We discuss why Chicken Geniu...	217
2	2022-10-11	RT stevenmarkryan httpstcobXfcmiiinc Twitter...	216
3	2022-10-10	RT BestTrader01 Institutes are Buying TSLA li...	222
4	2022-10-09	RT theeconomystic The only stock I own and ha...	
5	2022-10-08	RT stevenmarkryan httpstcoiBp1y4SaOT Is Tesl...	
6	2022-10-07	Open 23394Close 22307Range 22202 23457Its no...	223
7	2022-10-06	Its finally happening the next evolution in l...	238
8	2022-10-05	RT DoctorJack16 How to know you are a veteran...	240

Figure 5.5: Adding prices column in ccddata

The prices of the stock on particular date taken to the ccddata dataset by adding price column. The blank values are replaced by the mean price which occurs for holidays. The overall data present in the **Figure 5.6**

```
[18] ccdata['Prices'] = ccdata['Prices'].apply(np.int64)
```

```
[19] ccdata["Comp"] = ''
ccdata["Negative"] = ''
ccdata["Neutral"] = ''
ccdata["Positive"] = ''
ccdata
```

	Date		Tweets	Prices	Comp	Negative	Neutral	Positive
0	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...		226				
1	2022-10-12	RT FinanceLancer We discuss why Chicken Geniu...		217				
2	2022-10-11	RT stevenmarkryan httpscobXfcmiinc Twitter...		216				
3	2022-10-10	RT BestTrader01 Institutes are Buying TSLA li...		222				
4	2022-10-09	RT theeconomystic The only stock I own and ha...		226				
5	2022-10-08	RT stevenmarkryan httpstcoiBp1y4SaOT Is Tesl...		226				
6	2022-10-07	Open 23394Close 22307Range 22202 23457Its no...		223				
7	2022-10-06	Its finally happening the next evolution in l...		238				
8	2022-10-05	RT DoctorJack16 How to know you are a veteran...		240				

Figure 5.6: Making dataset to perform Sentiment Analysis

Sentiment Analysis

Sentiment analysis (or opinion mining) is a natural language processing (NLP) 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 and understand customer needs.

Here comes Sentiment Analysis to get to know the emotion of the tweet. This is main moto that is used in this project.

In addition to identifying sentiment, opinion mining can extract the polarity (or the amount of positivity and negativity), subject and opinion holder within the text. Furthermore, sentiment analysis can be applied to varying scopes such as document, paragraph, sentence and sub-sentence levels.

Types of Sentiment Analysis

1. Fine-grained sentiment analysis provides a more precise level of polarity.
2. Emotion detection identifies specific emotions rather than positivity and negativity.
3. Intent-based analysis recognizes actions behind a text in addition to opinion.

4. Aspect-based analysis gathers the specific component being positively or negatively mentioned.

VADER Sentiment Analysis:

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only talks about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

Example

“The party is wonderful.” and “I hate that man.”

The first one clearly conveys positive emotion, whereas the second conveys negative emotion. Humans associate words, phrases, and sentences with emotion. The field of Text Sentiment Analysis attempts to use computational algorithms in order to decode and quantify the emotion.

Text Sentiment Analysis is a really big field with a lot of academic literature behind it. However, its tools really just boil down to two approaches: the lexical approach and the machine learning approach.

Lexical approaches aim to map words to *sentiment* by building a lexicon or a ‘dictionary of sentiment.’ We can use this dictionary to assess the sentiment of phrases and sentences, without the need of looking at anything else. Sentiment can be categorical — such as {negative, neutral, positive} — or it can be numerical — like a range of intensities or scores. Lexical approaches look at the sentiment category or score of each word in the sentence and decide what the sentiment category or score of the whole sentence is. The power of lexical approaches lies in the fact that we do not need to train a model using labelled data, since we have everything, we need to assess the sentiment of sentences in the dictionary of emotions. VADER is an example of a lexical method.

By using the VADAR the emotions of the tweets are going to defines the polarity.

```
[20] import nltk
      nltk.download('vader_lexicon')

[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
True
```

Figure 5.7: Downloading VADER lexicon from NLTK

```
[21] from nltk.sentiment.vader import SentimentIntensityAnalyzer
      from nltk.sentiment.vader import SentimentIntensityAnalyzer
      import unicodedata
      sentiment_i_a = SentimentIntensityAnalyzer()
      for indexx, row in ccdata.T.iteritems():
          try:
              sentence_i = unicodedata.normalize('NFKD', ccdata.loc[indexx, 'Tweets'])
              sentence_sentiment = sentiment_i_a.polarity_scores(sentence_i)
              ccdata['Comp'].iloc[indexx] = sentence_sentiment['compound']
              ccdata['Negative'].iloc[indexx] = sentence_sentiment['neg']
              ccdata['Neutral'].iloc[indexx] = sentence_sentiment['neu']
              ccdata['Positive'].iloc[indexx] = sentence_sentiment['compound']
              # ccdata.set_value(indexx, 'Comp', sentence_sentiment['pos'])
              # ccdata.set_value(indexx, 'Negative', sentence_sentiment['neg'])
              # ccdata.set_value(indexx, 'Neutral', sentence_sentiment['neu'])
              # ccdata.set_value(indexx, 'Positive', sentence_sentiment['pos'])
          except TypeError:
              print (stocks_dataf.loc[indexx, 'Tweets'])
              print (indexx)

/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
self._setitem_single_block(indexer, value, name)
```

[22] ccdata

	Date	Tweets	Prices	Comp	Negative	Neutral	Positive
0	2022-10-13	RT TheTw1tterX As the TSLA and TWTR holders we...	226	0.9999	0.037	0.854	0.9999
1	2022-10-12	RT FinanceLancer We discuss why Chicken Geniu...	217	0.9998	0.065	0.838	0.9998
2	2022-10-11	RT stevenmarkryan httpstcobXfomiinc Twitter...	216	1.0	0.068	0.81	1.0
3	2022-10-10	RT BestTrader01 Institutes are Buying TSLA li...	222	1.0	0.044	0.814	1.0
4	2022-10-09	RT theeconomystic The only stock I own and ha...	226	0.9998	0.07	0.817	0.9998
5	2022-10-08	RT stevenmarkryan httpstcoiBp1y4SaOT Is Tesl...	226	1.0	0.027	0.846	1.0
6	2022-10-07	Open 23394Close 22307Range 22202 23457Its no...	223	0.9999	0.066	0.829	0.9999
7	2022-10-06	Its finally happening the next evolution in l...	238	1.0	0.057	0.825	1.0
8	2022-10-05	RT DoctorJack16 How to know you are a veteran...	240	1.0	0.052	0.832	1.0


Figure 5.8: VADAR performed on Dataset

Training and Testing

Making a new data frame with necessary columns for providing machine learning.

```
[24] df_=ccdata[['Date', 'Prices', 'Comp', 'Negative', 'Neutral', 'Positive']].copy()
```

```
[25] df_
```



	Date	Prices	Comp	Negative	Neutral	Positive
0	2022-10-13	226	0.9999	0.037	0.854	0.9999
1	2022-10-12	217	0.9998	0.065	0.838	0.9998
2	2022-10-11	216	1.0	0.068	0.81	1.0
3	2022-10-10	222	1.0	0.044	0.814	1.0
4	2022-10-09	226	0.9998	0.07	0.817	0.9998
5	2022-10-08	226	1.0	0.027	0.846	1.0
6	2022-10-07	223	0.9999	0.066	0.829	0.9999
7	2022-10-06	238	1.0	0.057	0.825	1.0
8	2022-10-05	240	1.0	0.052	0.832	1.0

Figure 5.9: Data Frame for testing

```
[26] train_start_index = '0'
train_end_index = '5'
test_start_index = '6'
test_end_index = '8'
train = df_.loc[train_start_index : train_end_index,:]
test = df_.loc[test_start_index:test_end_index,:]
```

```
[27] sentiment_score_list = []
for date, row in train.T.iteritems():
    sentiment_score = np.asarray([df_.loc[date, 'Negative'],df_.loc[date, 'Positive']])
    sentiment_score_list.append(sentiment_score)
numpy_df_train = np.asarray(sentiment_score_list)
```

```
[28] print(numpy_df_train)
```

```
[[0.037  0.9999]
 [0.065  0.9998]
 [0.068  1.    ]
 [0.044  1.    ]
 [0.07   0.9998]
 [0.027  1.    ]]
```

```
[29] sentiment_score_list = []
for date, row in test.T.iteritems():
    sentiment_score = np.asarray([df_.loc[date, 'Negative'],df_.loc[date, 'Positive']])
    sentiment_score_list.append(sentiment_score)
numpy_df_test = np.asarray(sentiment_score_list)
```

```
[30] print(numpy_df_test)
```

```
[[0.066  0.9999]
 [0.057  1.    ]
 [0.052  1.    ]]
```

```
[31] y_train = pd.DataFrame(train['Prices'])
#y_train=[91,91,91,92,91,92,91]
y_test = pd.DataFrame(test['Prices'])
print(y_train)
```

```
Prices
0    226
1    217
2    216
3    222
4    226
5    226
```

Figure 5.10: Testing and Training

The training and testing values are going to use in the Random Forest Regressor to get predicted values.

Random Forest Regression

Random Forest Regression is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

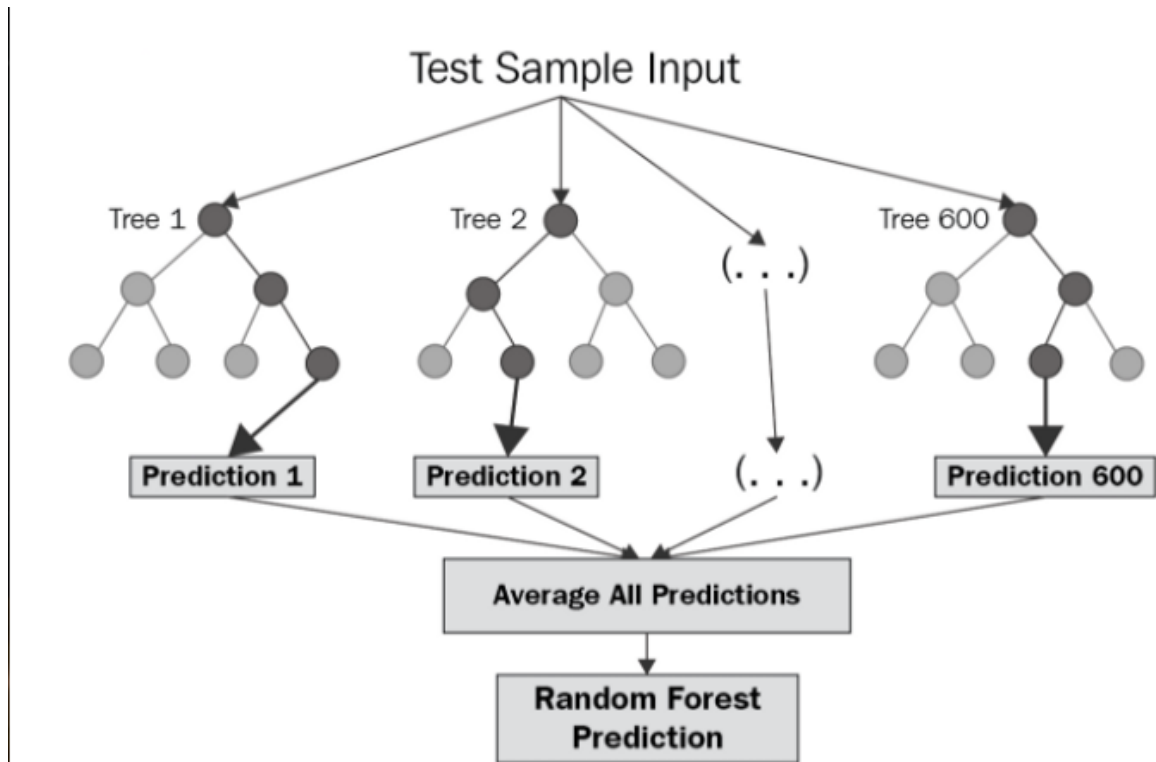


Figure 5.11: Random Forest Regression

The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

Random Forest Regression model performed the best out of all the other regression models. The other regression models like Simple linear regression, Multiple linear regression, Decision tree regression, Support vector regression, etc.

```
[32] # from treeinterpreter import treeinterpreter as ti
      from sklearn.tree import DecisionTreeRegressor
      from sklearn.ensemble import RandomForestRegressor
      from sklearn.metrics import classification_report, confusion_matrix

      rf = RandomForestRegressor()
      rf.fit(numpy_df_train, y_train)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n,) or (n, 1) before calling the fit method.
import sys
RandomForestRegressor()

[33] prediction = rf.predict(numpy_df_test)

[34] print(prediction)

[219.72 217.63 220.74]

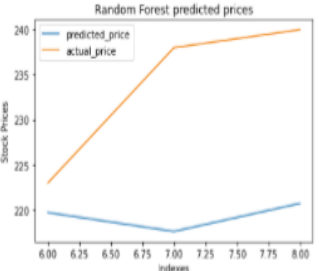
[35] import matplotlib.pyplot as plt

[36] idx=np.arange(int(test_start_index),int(test_end_index)+1)
      predictions_df_ = pd.DataFrame(data=prediction[0:], index = idx, columns=['Prices'])

[37] predictions_df_

   Prices
6  219.72
7  217.63
8  220.74

[38] ax = predictions_df_.rename(columns={"Prices": "predicted_price"}).plot(title='Random Forest predicted prices')#predicted value
      ax.set_xlabel("Indexes")
      ax.set_ylabel("Stock Prices")
      fig = y_test.rename(columns={"Prices": "actual_price"}).plot(ax = ax).get_figure()#actual value
      fig.savefig("random_forest.png")
```



Indexes	predicted_price	actual_price
6.00	219.72	222.00
6.75	217.63	235.00
7.00	217.63	238.00
8.00	220.74	240.00

Figure 5.12: Final Output Dataset(small)

The final predictions are shown in the line graph which shows minor changes at certain point where the stock prices are depending upon tweets. As it is small data set the prediction not so great.

So we are going to perform on the large dataset to get good accuracy.

5.2 TESTING LARGE DATASET

Now we are performing on the large dataset taken from the Kaggle to get clear results. It consists of Date, Closing price, Adj close price and Tweets as columns.

```
[41] stocks_dataf = pd.read_pickle('Twitter_Dataset.pkl')
      stocks_dataf.columns=['closing_price', 'adj_close_price', 'Tweets']
```

```
[42] stocks_dataf
```

	closing_price	adj_close_price	Tweets
2007-01-01	12469.971875	12469.971875	. What Sticks from '06. Somalia Orders Islamis...
2007-01-02	12472.245703	12472.245703	. Heart Health: Vitamin Does Not Prevent Death...
2007-01-03	12474.519531	12474.519531	. Google Answer to Filling Jobs Is an Algorith...
2007-01-04	12480.690430	12480.690430	. Helping Make the Shift From Combat to Commer...
2007-01-05	12398.009766	12398.009766	. Rise in Ethanol Raises Concerns About Corn a...
...
2016-12-27	19945.039062	19945.039062	. Should the U.S. Embassy Be Moved From Tel Av...
2016-12-28	19833.679688	19833.679688	. When Finding the Right Lawyer Seems Daunting...
2016-12-29	19819.779297	19819.779297	. Does Empathy Guide or Hinder Moral Action?. ...
2016-12-30	19762.599609	19762.599609	. Shielding Seized Assets From Corruption's Cl...
2016-12-31	19762.599609	19762.599609	Terrorist Attack at Nightclub in Istanbul Kill...

3653 rows x 3 columns

Figure 5.13: Large Dataset

Applying VADER Sentiment Analysis to the Dataset

```
[50] dataframe
```

	adj_close_price	Comp	Negative	Neutral	Positive
0	12469	-0.9814	0.159	0.749	-0.9814
1	12472	-0.8521	0.116	0.785	-0.8521
2	12474	-0.9993	0.198	0.737	-0.9993
3	12480	-0.9982	0.131	0.806	-0.9982
4	12398	-0.9901	0.124	0.794	-0.9901
...
3648	19945	-0.9898	0.178	0.719	-0.9898
3649	19833	-0.9844	0.177	0.704	-0.9844
3650	19819	-0.9782	0.14	0.761	-0.9782
3651	19762	-0.995	0.168	0.734	-0.995
3652	19762	-0.2869	0.173	0.665	-0.2869

3653 rows x 5 columns

Figure 5.14: Sentiment Analysis

Showing positive and negative tweets in Dataset

```
[51] posi=0
      nega=0
      for i in range (0,len(dataframe)):
          get_val=dataframe.Comp[i]
          if(float(get_val)<(-0.99)):
              nega=nega+1
          if(float(get_val)>(-0.99)):
              posi=posi+1
      posper=(posi/(len(dataframe)))*100
      negper=(nega/(len(dataframe)))*100
      print("% of positive tweets= ",posper)
      print("% of negative tweets= ",negper)
      arr=np.asarray([posper,negper], dtype=int)
      mlpt.pie(arr,labels=['positive','negative'])
      mlpt.plot()
```

```
% of positive tweets= 44.34711196277033
% of negative tweets= 55.43388995346291
[]
```

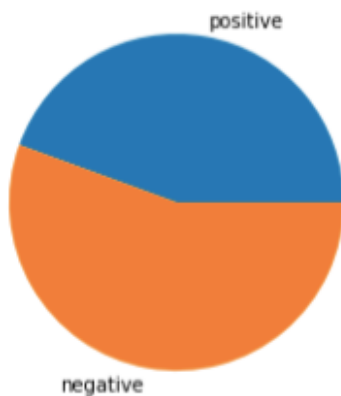


Figure 5.15: Pie Chart

Data Mining

Data mining is the process of sorting through large data sets to identify patterns and relationships that can help solve business problems through data analysis. Effective data mining aids in various aspects of planning business strategies and managing operations. That includes customer-facing functions such as marketing, advertising, sales, and customer support, plus manufacturing, supply chain management, finance and HR.

Data Mining process will be done in the process of analysis to identify the data easily.

```

from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import classification_report, confusion_matrix

rf = RandomForestRegressor()
rf.fit(numpy_dataframe_train, train['adj_close_price'])
prediction=rf.predict(numpy_dataframe_test)
import matplotlib.pyplot as plt
%matplotlib inline
idx = pd.date_range(test_data_start, test_data_end)
predictions_df = pd.DataFrame(data=prediction[0:], index = idx, columns=['adj_close_price'])
predictions_df['adj_close_price'] = predictions_df['adj_close_price'].apply(np.int64)
predictions_df['adj_close_price'] = predictions_df['adj_close_price'] + 4500
predictions_df['actual_value'] = test['adj_close_price']
predictions_df.columns = ['predicted_price', 'actual_price']
predictions_df.plot()
predictions_df['predicted_price'] = predictions_df['predicted_price'].apply(np.int64)
test['adj_close_price']=test['adj_close_price'].apply(np.int64)
#print(accuracy_score(test['adj_close_price'],predictions_df['predicted_price']))
print(rf.score(numpy_dataframe_train, train['adj_close_price']))

```

Figure 5.16: Random Forest Regression for Large Dataset

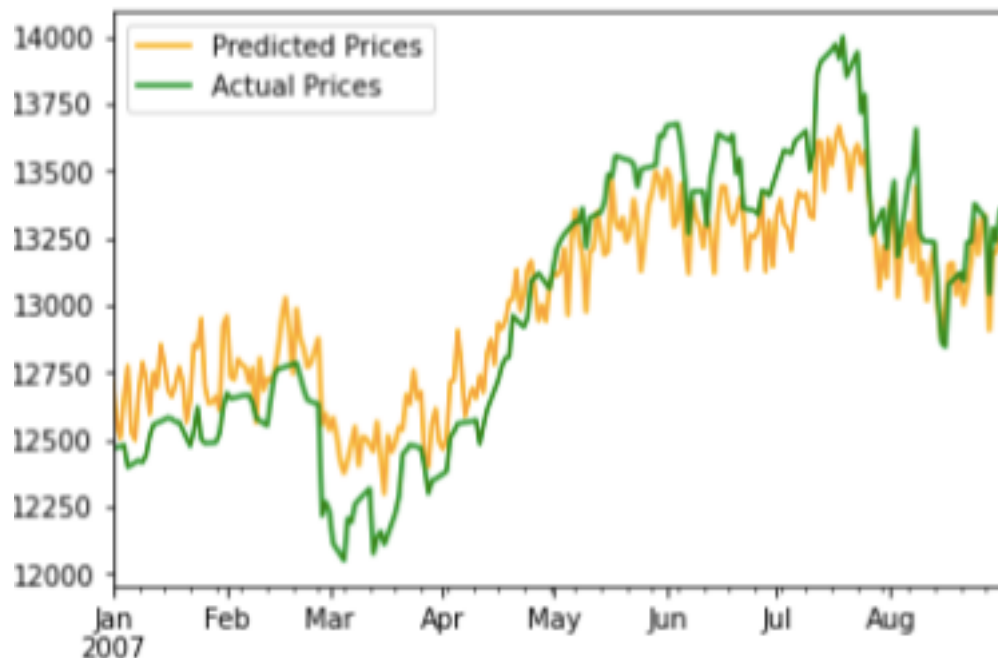


Figure 5.17: Final Stage

The above figure shows the stock market is depending upon the tweets on twitter. This is our final analysis.

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

The purpose of our project how sentiment analysis of the twitter data is correlated to the prediction of the stock market price for all the companies which are taken. The result obtained after the prediction process clearly specifies that, we have obtained the accurate value which matches with the actual stock price appropriately.

Thus, social media such as twitter can be used as a source to predict the stock market price with maximum accuracy.

6.2 FUTURE ENHANCEMENT:

We can further extend the functionality for this analysis to predefined application where all stocks should be able to predict, that depending upon tweets from Twitter, should give output as line graph.

7. REFERENCES

- [1] Michal Skuza, Andrzej Romanowski, "Sentiment Analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction", Computer Science and Information Systems pp. 1349– 1354, 2015 F230 ACSIS, Vol.5.
- [2] Venkata Sasank Pagolu, Kamal Nayan Reddy Challa, Ganapati Panda, Babita Majhi, "Sentiment Analysis of Twitter Data for Predicting Stock Market Movements", International conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), 2016.
- [3] Tina Ding, Vanessa Fang, Daniel Zuo, "Stock Market Prediction based on Time Series Data and Market Sentiment", 2012.
- [4] Phillip Tichaona Sumbureru, "Analysis of Tweets for Prediction of Indian Stock Markets", International Journal of Science and Research (IJSR), Volume 4 Issue 8, August 2015.
- [5] Rishabh Soni, K. James Mathai, "Improved Twitter Sentiment Prediction through Cluster-then-Predict Model", International Journal of Computer Science and Network, Volume 4, Issue 4, August 2015.
- [6] Linhao Zhang, "Sentiment Analysis on Twitter with Stock Price and Significant Keyword Correlation", April 16, 2013.
- [7] Jose, A.K., Bhatia, N., Krishna, S. : "Twitter sentiment analysis", Major Project Report, NIT Calicut (2010).
- [8] Dr. P. K. Sahoo, Mr. Krishna charlapally, "Stock Price Prediction Using Regression Analysis", International Journal of Scientific & Engineering Research, Volume 6, Issue 3, March-2015.