Coventry University

Middle East College

Project Submission
in Partial Fulfilment of the requirements for the Degree of Master of Science in Information Technology (MSC-IT)

Predictive Analysis of Stock Market using Sentiment Analysis of Twitter

Author: Rohit Mohanan Nair

(PG17F1825)

Supervisor: Dr. Mounir Dhibi

Acknowledgment

The researcher would like to take this opportunity to thank all those who have inspired and guided him to successfully complete this project. The researcher would like to express his sincere gratitude to his project supervisor, Mr. Mounir Dhibi for his patience, intense knowledge, motivation and guidance throughout the project. The researcher is also grateful to the institution Middle East College of Applied Sciences for giving him the platform to bring forth his ideas. The researcher would also like to thank his parents for their support and cooperation. Last, but not the least, the researcher would like to thank God for his blessings upon him.

Abstract

Experienced as well as inexperienced investors have to face loss due to uncertain behavior of stock market. This uncertain behavior depends on the financial situation of the company, political situation within a country and different event that take place in a company. A stock market is a place where millions of shares trade hands every day, making shrewd and intelligent decisions can lead to enormous benefits which are extremely challenging considering that no such guidance or education is available. Our project is "Predictive Analysis of Stock Market using Sentiment Analysis of Twitter". It includes data gathering of tweets, preprocessing of tweets, improvement of data classifier algorithms and sentiment analysis of tweets for predictions of stock market.

Contents

Table of Figures

1. INTRODUCTION

1.1. Background of the Research

This project "Predictive Analysis of Stock Market using Sentiment Analysis of Twitter" aims to be a standout and accurate guidance for all kinds of investors in the country. The stock market is highly affected by political situation within a country. This is the reason the proposed methodology consists of sentiment analysis to analyze political situation within a country. Experienced investors know the rise and fall of stock market, their investment largely depends on their past experience and they get support from stock market to invest or withdraw their investments from it. Inexperienced or common investors are not aware of such techniques. By making my proposed system intelligent with sentiment analysis of tweets, we aim to provide a helpful platform for all kind of investors.

The project makes use of cutting-edge 21st century technology to ensure that whatever it claims is backed-up by correct information and error-free processing. To name them, Python and Machine Learning technologies such as Scikit-learn, Pandas and NLTK, TEXTBLOB for sentiment analysis of twitter have been used to ensure good-handling and processing of data. The major modules of the project include extracting tweets, preprocessing of tweets, sentiment analysis of tweets and classification of tweets as positive, negative and neutral using Machine Learning algorithms Naïve Bayes and SVM.

1.2. Motivation

Stock Market prediction is one of the difficult tasks to accomplish. This has been a topic of several researches (Shams and Muhammed 2005) (Deng et al. 2017) and researchers have tried to predict it by using Machine learning algorithms such as Recurrent Neural Networks (RNN), Regression Algorithms, Time Series models, Long Short Term Memory (LSTM), etc. but they wouldn't succeed in building an efficient model. That is largely due to the fact that the stock market is highly volatile, and it is affected by several factors cannot be taken into consideration for quantification. One of those factors is "tweets" that impact stock market. Its effect on stock market is something that cannot be computed so easily. Also, thing has not really been achieved so far.

1.3. Problem Statement

Experienced as well as inexperienced investors have to face loss due to uncertain behavior of stock market. This uncertain behavior depends on financial situation of the company as well as political situation within a country.

1.4. Research Objectives

In this report we intend to use social media platform for collecting the twitter feeds and conduct sentiment analysis of the feeds for predicting accurately the stock market trend.

- Identifying and comparing the machine learning algorithms best suited for stock market prediction.
- Studying the processing and implementation of human feeds from social media based on the stocks.

- Performing Sentiment analysis on social media feeds using natural language processing to understand the trends in stock market.

1.5. Research Aim

The most important research question the lecturer is seeking answer for is:

- Is it possible to improve the accuracy of algorithms used in data classifier model for increasing accuracy of predicting stock market prices?

1.6. Proposed Solution

Five features i.e. followers, clusters, polarity, sentiment confidence and Difference (difference between closing and opening price of stocks) are given as input to Support Vector Regression and LSTM (Long Short-Term Memory). These Machine Learning algorithms then give predictions about stock market.

1.7. Target Users

The aim of our project is to serve all types of investors i.e. experienced, intermediately experienced or beginners. It is good to have knowledge of the behavior and ups and downs of stock market, but this is not required because our system will give predicted decisions to its users. Our main target users are the following:

- Common investors: The common investors refer to those who have lot of financial assets and they are financially strong, but they don't have right direction for investment. Through the assistance of our project, we are hopeful that they will be able to invest in stocks.

- Experienced investors: Although experienced investors have very deep knowledge of stock market, but they also take assistance from different sources to get verification for their information. They can use our system because our project will provide assistance to them in predicting stock market prices.
- Common man: By common man, we mean those people who are not financially strong and who don't have any assets, but they want to increase their wealth and what they have. By taking guidance from our systems, these people can achieve what they want.

1.8. Stock Market

Stock Market prediction is one of the difficult tasks to accomplish. This has been a topic of several researches and researchers have tried to predict it by using Machine learning algorithms such as Recurrent Neural Networks (RNN), Regression Algorithms, Time Series models, Long Short Term Memory (LSTM), etc. but they wouldn't succeed in building an efficient model. That is largely due to the fact that the stock market is highly volatile, and it is affected by several factors among which political situation within a country is the most significant one. To handle the effects caused by the mentioned factor, the researcher is using tweets for the analysis of political situation within a country. The effect of tweets on stock market is something that cannot be computed so easily. Also, thing has not really been achieved so far. There is no tool available online that analyzes stock market by doing sentimental analysis of tweets. All trading websites and online tools only analyze historical data and give predictions on the basis of numerical data only.

It may take years of experience and knowledge to understand and analyze the stock market trends and make wise investments. Ordinary people may use Stock markets to make fortune while financial analysts may use them to determine a country's economy and analyze the growth pattern and impact of government policies and economic situation prevailing in the country. Consider the table below.

Date	Open	High	Low	Close
02-01-2015	111.39	111.44	107.35	109.33
05-01-2015	108.29	108.65	105.41	106.25
06-01-2015	106.54	107.43	104.63	106.26
07-01-2015	107.2	108.2	106.7	107.75
08-01-2015	109.23	112.15	108.7	111.89
09-01-2015	112.67	113.25	110.21	112.01
12-01-2015	112.6	112.63	108.8	109.25
13-01-2015	111.43	112.8	108.91	110.22
14-01-2015	109.04	110.49	108.5	109.8
15-01-2015	110	110.06	106.66	106.82
16-01-2015	107.03	107.58	105.2	105.99
20-01-2015	107.84	108.97	106.5	108.72
21-01-2015	108.95	111.06	108.27	109.55
22-01-2015	110.26	112.47	109.72	112.4
23-01-2015	112.3	113.75	111.53	112.98
26-01-2015	113.74	114.36	112.8	113.1
27-01-2015	112.42	112.48	109.03	109.14
28-01-2015	117.63	118.12	115.31	115.31
29-01-2015	116.32	119.19	115.56	118.9
30-01-2015	118.4	120	116.85	117.16

Table 1

AAPL stock data collected from Yahoo! Finance starting from Jan 1 2015 to Jan 30 2015

The table shows the stock details for Apple (AAPL) for a period of one month. We could notice that the stock started trading at 109.33 at the beginning of the month and ended up in 117.16 at the end of the month. On furthermore analysis the fluctuations could we visible at each trading day. For a common person this could be just random numbers, but this are a result of various factors that have taken place in the financial, economic, political and geographical actors. By taking a proper account of all these factors and with experience one could make predictions regarding the trading of a particular stock.

2. LITERATURE REVIEW

Our aim was to build a tool that would provide assistance to investors of all types so that they may get maximum profit out of their investments by investing in the stock market of USA.

During the entire cycle of FYP, we came across several challenges; the biggest among them was to do sentiment analysis of tweets. But this has to be done because tweets have potential impact on stock market. As twitter is new platform for politicians and social media users to tweet about their opinions so we decided to analyze tweets to have an idea that how much political situation within a country affects stock market. In order to find effect of tweets on stock market we followed a paper entitled "Predicting Stock market trend from twitter feed and Building a framework for Bangladesh" that directs us to a right direction, and we succeed to attain our goal to some extent. (Karim, Abdullah, and Tayaba 2018)

This paper proposes a solution that consists of several steps i.e. data collection, data pre-processing, generating dictionaries and then applying various algorithms on data including Support Vector Machine, Gradient Boosting Decision Tree and Fuzzy K-Nearest Neighbor to classify tweets as positive, negative and neutral tweets. (Karim, Abdullah, and Tayaba 2018)

After doing some literature review, it became obvious to us that project is complex. Some papers relate the prediction of Stock Market to "Arc of Noah" i.e. its complex task and nearly impossible to predict stock market completely with high accuracy due to its volatile nature. We didn't take it as much complex as it started becoming complex to us as we proceeded in our project. Our knowledge increased with time and we found that predicting the stock market is really not an easy matter.

Paper titled "Using sentiment analysis for stock exchange prediction" uses natural language processing algorithms and then applies SVM (L.Lima, Milson, P. Nascimento n.d.) to

extract patterns which are required for prediction. In the first stages of our tests, two algorithms i.e. Support Vector Machines (SVM) and Naïve Bayes were used for getting best results in terms of accuracy. In the first step, SVM give better results than Naïve Bayes so in second step it was given attribute "sentiment". This attribute classifies tweets in two classes i.e. positive and negative.

This paper "Stock Prediction Using Twitter Sentiment Analysis" (Goel and Mittal 2012) proposes a solution based on two techniques i.e. Machine Learning Techniques and Sentimental Analysis of Twitter. Tweets are used in prediction of public mood and Dow Jones Industrial Average (DJIA) past days' values is used in the prediction of stock market movement. A new cross-validation method was introduced in this paper for financial data. Self-Organizing Fuzzy Neural Networks (SOFNN) was applied on tweets along with DJIA values. Using predicted values, intelligent decisions of BUY and SELL were generated using portfolio management. (Goel and Mittal 2012)

The paper entitled "Predictive sentiment analysis of tweets: A stock market application." (Smailovic et al. 2014) proposes positive sentiment probability for prediction of sentiment analysis in finance. Stock price movements can be indicated by sentiment analysis i.e. positive and negative sentiments using Granger Causality Test. It is not necessary that tweets exhibit sentiments. To handle this, neutral zone is introduced in this paper using sentiment classification to classify tweets as neutral.

This paper (Brown 2012) attempts to find out correlation between stock market and twitter by doing sentimental analysis and some attributes of stock market such as stock volume and price movement. It also considers the effects of twitter user's reputation on stock market and sentiment analysis of tweets.

The paper titled “Prediction of Stock Market performance by using Machine Learning Techniques” explicitly mentioned that “The stock market is a complex system and often covered in mystery, it is therefore, very difficult to analyze all the impacting factors before making a decision” which is a testament to the challenge presented by this project. (Kamran 2019)

One of the other important concepts we took heed from was from the research paper titled “Deep Direct Reinforcement Learning for Financial Signal Representation and Trading” about High Frequency Trading (HFT). HFT represents a concept where an automated trading system remain vigilant round the stock market working time and immediately carries out a transaction as soon as it sees a stock loss or gain price. (Deng et al. 2017)

HFT was an interesting concept which works in seconds to ensure minimal losses and maximum profits but what it does is that it waits for the stock to lose price before selling it. That causes a loss on the stock which in our opinion isn’t the best thing out there. What we’d like would be a way to educate investors that the stock will potentially lose price on the future so that they could trade it out before the actual loss actually happened.

With the increasing number of social websites, people are getting more options to communicate with each other. The “statuses” or “comments” which people share on their profiles are considered as a source of their opinions. Whenever we want to analyze a product or topic, these opinions play a significant role. These opinions have some attributes or features associated with them. These features can be used by classifiers for classifying the opinion as negative, positive or neutral.(‘Ijsetr.Org’ 2019)

The paper titled “A hybrid stock trading framework integrating technical analysis with machine learning techniques” proposes a solution that uses historical stocks data and some renowned technical indicators to generate trading decisions using some trading rule and

analyzing trends of market. This paper was very close to our problem statement and our approach. But this paper focused only on using historical stocks data and tweets. (Dash and Dash 2016)

(Dash and Dash 2016) consists of several steps which includes gathering data of stocks, extracting technical indicators, normalizing data, analyzing trends, trading signal generation, model training, trend determination from trading signals and at last generating trading decisions. It generates trend by comparing closing value with MA i.e. if closing value is greater than MA value and MA value is increasing for past five days then trend is uptrend otherwise trend is downward. They generate trading signals for all three trends i.e. uptrend, downtrend and no trend using some formulas. After trading signal generation again, trends are determined by comparing output obtained from model that is trained by historical data and technical indicators. If predicted value (output obtained from model) is greater than mean of trading signals, then trend is uptrend otherwise trend is downtrend.

Another article titled "Machine Learning Techniques for Stock Prediction" by (Shah 2007) discusses the application of various Machine learning Algorithms along with their advantages and disadvantages. Machine Learning Algorithms used in this paper are Support Vector Machines, Linear Regression, predicting values using expert weighting and some other techniques. Some existing solutions such as Trade decision provide charts that can be analyzed and doesn't provide services to public. Their services are only for professional traders. Bar chart provides only dashboard to users. No predictions are available for users. Wallet Investor has no authentication system for users. There is no dashboard available for users. It only provides predictions of stock prices on the basis of historical stocks data. There are many other systems

but none of them is working with sentiment analysis to improve accuracy of predictions. Also, they charge heavily for providing their services.

In this paper (V.S et al. 2016) the main topic is based in the field of SENTIMENT ANAYLSIS OF TWITTER. In this paper, they have done the sentiment analysis of twitter using web scrapping and Dow Jones Industrial Average (DJIA). Over 2 lakh tweets of Microsoft from a period of August 31st, 2015 to August 25th,2016 are extracted from twitter. They have used java application called Twitter 4J that is used to scrap tweets from twitter. The tweets are correlated using Twitter 4J and filtering is done by using keywords like % Microsoft, %Apple etc. They have collected a total of 3,216 humans analyzed tweets from the total tweets. They are classified as 1 for Positive, 0 or Neutral and 2 for Negative. They have also done the training of the machine learning model for classifying the nonhuman annotated tweets and human annotated tweets. They have taken care that the keyword filtering process they are doing is done with complete care so that the emotions that are extracted from twitter represent the emotions of the people of Microsoft. They have collected the 2,50,000 tweets of Microsoft from a period of August 31st, 2015 to August 25th,2016 are scrapped from yahoo finance. All the news and tweets regarding the products and services of Microsoft were also included. There are many sentiment analysis algorithms that scraps movie reviews, financial reviews etc. on which there are lots of studies going on, where it's a very much specific field which are also open source. One of the main disadvantages of their algorithm is that they have trained their kit with different corpus. The twitter tweets are annotated as positive, negative and neutral based on their sentiment analysis algorithm. In this paper, they have also found that the mood states of people (Happy, calm, Anxiety) found from twitter feeds are compared to the well-known Dow Jones Industrial Index value. Dow Jones Industrial Average (DJIA) values are used for prediction of

movements of the future prices of the stock and have used their predicted analysis results in their algorithm management. They have also shown that the annotated dataset of human stats in their analysis is also exhausted. They also say that there exists a strong bond between twitter sentiments and the prices of the stock in average stock prices tab. They have considered the tweets and opening and closing prices of the Microsoft stock of one year. They have done the predictions using a Fuzzy neural network. Their computations show that mood states of the public in twitter are collectively compared with Dow Jones Industrial Index value. They have investigated that the investment plans are recognized by observing and classifying the twitter feeds. They have also shown that they have scrapped the twitter tweets and have predicted the stock market prices based on the various types of industries like IT, Finance etc. They have also derived that out a highly negative connection between mood states like happy, sad and neutral in tweets with the Dow Jones Average Index value. Recently, with increases and decrease in stock prices they have found that the connection between sentiments of people and prices using Pearson correlation coefficient for stocks. In this paper, they have tried to predict the increase and decrease in stock prices from predictive analysis of twitter feeds

On year 2011 (Bollen, Mao, and Zeng 2011) has presented a paper that got huge media attention recently. They have predicted the values of stocks and derived a naive strategy for keeping a portfolio that is profitable. By studying and observing the emotions of public in twitter they have predict the future stock market prices. To predict the Dow Jones Industrial Index (DJIA) values they have used various machine learning techniques using their sentiment analysis results. To obtain processed values they have fed the raw DJIA values into pre-processor. Also, the tweets are fed to the sentiment analysis algorithm which classifies the emotions to different categories. In this paper, they have used the Opinion Finder and Google Profile of Mood States

(GPOMS) algorithm to scrap the tweets of all twitter users and classify them according to 6 categories which are Calm, Alert, Sure, Vital, Kind and Happy. In this paper, they have briefly described the dataset they have adopted and the measures of data processing techniques. By cross validating the presidential elections and Thanksgiving Day in 2008 by correlating its capability to detect the response of the public in twitter. To identify the hype created by public emotions in twitter which are identified by using Opinion Finder and GPOMS mood time series which as predicted as changes in DJIA values. In this paper, they have used Self Organizing Fuzzy Neural Networks to predict DJIA values using past closing values. In predicting the increase and decrease in closing Dow Jones Industrial Index (DJIA) values there's great precision of 87% which is a remarkable figure.

This paper that was published in the year 2009 (M, S, and W 2009) applies machine learning techniques and sentiment techniques to derive the connection between stock market emotions and twitter emotions. In this paper, they have used the data that is scraped from twitter for the prediction of emotions that are previously determined. They have used Dow Jones Industrial Index (DJIA) values for the prediction of stock market. They have proposed that they have obtained remarkable accuracy of 75.56% by applying the Self Organizing Fuzzy Neural Networks (SOFNN) on the feeds that are scraped from twitter and Dow Jones Industrial Index (DJIA) values between the period June 2009 to December 2009. They have implemented naïve portfolio management strategy with respect to the public emotions. In their work, they have implemented two important data sets: - They have taken the Dow Jones Industrial Average (DJIA) values from June 2009 to December 2009. It has collected the results from Yahoo! Finance and includes the opening, closing, max and min values of a particular day.

The available data that is collected from twitter contains more than 476 million tweets from 17 million users between the period June 2009 to December 2009. The included data are timestamp, username and tweeted text. They have split their data that is scrapped from twitter according to daily or weekly basis and classified them according to their time stamps info based on their predictive analysis and sentiment analysis.

In (Uc and Yue 2019) , a significant prediction of the stock market has been made through the analysis of yahoo finance and google finance. Efficient market hypothesis (ERM) is one of the previously used financial standard of stock market prediction. Efficient market hypothesis states that the market may respond to any news articles that is being portrayed in financial sector regarding a particular stock in the long run. But the degree of efficiency always seems to have a problem and is said one can beat the market in a short run. According to this research paper the yahoo finance data for a particular period is recorded which contains open, high, low, close and Adjusted close prices of a particular stock for example AAPL (Apple Inc.) is collected. The key events form Yahoo Finance Events are recorded for that particular stock during the specified duration. Along with the Yahoo Finance events the data from Reuters a famous financial news website for AAPL is collected. For this particular research, the data is collected from the period starting from Aug 2007 to Aug 2012. The google trend data for AAPL is also collected along with this. The google trend data is collected one month earlier this enables us to make an efficient comparison between the news articles and the stock data. The google trend data collected shows us the impact that a news articles makes on the stock seeing how many people check a particular stock when a news article is published. The impact on the stock with a particular news is seen as an exponential function having a parameter 1/7. The exponential function algorithm that is being used in this paper rates the news articles each day with values -1,

0, 1. The news will have exponential decay depending on the +number of days the news is being circulated. Value of a news article on a particular day is calculated. After the value of news articles have been calculates the Google trend values are obtained. Google Trend data provides the magnitude of the news article. This value is multiplied together to give the final value of the news article. Stock market trend studied from the Yahoo finance is matched with the effective news article scores. News scores are divided into categories and a representation is created. This comparison shows an effective relationship between the news articles and the Stock market. Whenever news articles are portrayed there will be an equivalent effect in the stock market values depending on the sectors affected.

In another paper published in the year 2007, (Seghal and Song 2007) have shown that number of counts of twitter feed causes the volumes of analysis of stock market previous work on stock market predictions. They have used specific algorithms for analyzing and identifying the news feeds and allows the users to trade on them successfully. They have also done simulations with their systems so as to test the accuracy of their systems by stimulating the real-time market. They have also tried the extraction of public motions on stock from the messages that are posted in webpages. They have also trained a machine learning model for classifying the extracted messages as “good”, “bad” or “neutral” based on the emotions.

Literature review usually discusses the published information in a specific subject area and thus sometimes the information in the particular subject area within the specific time span in the research. The literature review can be stated as just the simple summary of many sources, but it is mainly having the organizational pattern and thus it combines the summary as well as the synthesis. The summary is said to be the overall overview of the information which is important and thus it is the synthesis of the re-organization or by the reshuffling of the information that is

provided. It has been stated that it might be the trace of the intellectual progression related to the field which includes the major debates of the topic. While depending upon the situation, it has been seen that the literature review might evaluate the main sources as well as advise the reader regarding the relevant and the most pertinent information.

3. METHODOLOGY

3.1. Introduction

Research Methodology includes the procedures that include data gathering, data analyzing and processing of data according to requirements. Data gathering includes various methods which are interviews, surveys, questionnaires, etc. This can also include various techniques that may help us in gathering past or present data according to our requirements.

3.2. Research Methods

There are different methods of research which includes surveys, interviews, questionnaires, experiments, case studies, etc. Survey can be of different types e.g. it can be in the form of questionnaires or interviews to collect information from a large group of people. Interview is also a method of getting information but in this method only one person can be interviewed face to face. This method should provide the interviewee privacy because he/she may want to keep his/her answers private and secret. Interview can be conducted online using latest technology or in person. Questionnaire is a method of collecting data from a handsome amount of people. This method doesn't ask people to spare some time especially for it. People can fill it out any time in their free time. They don't have to take out special time like interviews and experiments where it involves people in different groups and the researcher asks these people to complete several tests. The comparison between results of several groups is performed. Case study includes the study of a person or a group in detail. This is usually done by observing and interviewing people.

3.2.1. Methodology

3.2.1.1. Unstructured Interviews

The researcher has used method “interviews” to collect useful information and opinion of people related to my project. It is the best methodology because people will answer the questions in a separate room and their answers will be kept secret and analyzed afterwards for further processing. Qualitative research methods are discussed along with data collection strategies in (Hoepfl 1997) that include interviews and observations. Qualitative and Quantitative research methods for business and technology are discussed in (Huarng, Rey-Mart, and Miquel-Romero 2018). Also which research method is suitable for business and which method is suitable for technology, justification for each field is provided in (Huarng, Rey-Mart, and Miquel-Romero 2018). In (Wenger 1991), it is about evaluation using qualitative methods. The methods used for data gathering are interviews and observations which will further help in generating results.

3.2.1.2. Study of Stock Market prediction using historical data of stocks and Twitter data

Prediction of stock market prices has been an interesting topic for researchers since many years. Some researchers have done research on stocks using historical and numerical data of stocks. With the advancement in technology and digital media, researchers have started analyzing data of financial websites and Twitter for sentiment analysis. In (Dash and Dash 2016) researchers have generated decisions related to stocks using Machine Learning techniques with technical analysis on historical data of stocks. In (Bollen, Mao, and Zeng 2011) keeping in view the trends of modern era and advancement of technology, researchers have used Twitter data to analyze stock market movements and to predict stock market trends. In (Porshnev, Redkin, and

Shevchenko 2013), stock market indicators are predicted using historical data of stocks and data of Twitter. By reviewing different research papers some of which are mentioned in this passage, the researcher deducted various factors which influence stock market movements and those factors are discussed in interview with different participants. To effectively analyze the factors, the following interview questions are designed to be asked from interviewees in an unstructured and less formal interview.

Interview Question	Interviewee1	Interviewee2	Interviewee3
Q1. Are you interested in stock market?	Yes I have great interest in stock market as I belong to Economics background	No, I don't have interest in stock market as I find it difficult to understand.	Right now, I don't have any interest but may be in future stock market takes my interest.
Q2. How much do you know about stock market?	I know very little about stock market. I have just heard some things about it from my fellows.	I have enough knowledge to understand some important things of stock market.	I have not very technical knowledge about stocks, but I know some things about stock market.
Q3. How much do you know about stock market trends?	I don't know how trends of stock market get affected.	I have little information that trends of stock market get affected by	I don't know how much trends of stock market get affected by which thing.

		political situation within a country.	
Q4. What factors affect the stock market trends?	Political affairs affect the stock market trends	Financial condition of company also affects stock market trends	I think that political situation of a country and financial condition of a company both affects stock market trends.
Q5. Which resource is best for stock market news?	According to my opinion Twitter is the best resource for stock market news because in this era people are using social media to express their opinion about a specific topic.	I think Financial News on various authentic websites can be proved a great resource for stock market news.	I think Twitter is the best resource for stock market news because this era is the era of digital media and people use social media more than other traditional resources.
Q6. Would you like to invest in stock market?	No, I would not like to invest in stock market because I am not familiar with trends of stock market and I don't have any technical knowledge	Yes, I would like to invest in stock market as I have plenty of knowledge about stocks. Also, I am eager to invest some money in stocks of a	I am not sure right now that either I want to invest in stock market or not but may be in future, I got some confidence and enough money to trust

	related to stocks.	well reputed company.	on stock market and invest in it.
Q7. If an app is provided to you related to stocks. How would you like to use that app?	I would like to have an android app installed on my mobile phone that I can use whenever I want to use it.	I would like to use stocks app like a web app because it provides you a wider view and you can navigate through web app on a larger screen.	I think app either in the form of web app or android app will be useful, but user interface of app should be friendly, attractive and interactive.
Q8. How would you like to buy shares of a specific company?	I did not buy any stocks ever till now so I have no idea how a person can buy shares of accompany.	I have heard from fellows of mine that stockbrokers help people in buying shares so I may opt that option.	I would like to rely on new technologies rather than manual methods. I would choose a well rated stock market app and buy shares with the help of that app.
Q9. Do you consider investment in stock market a reliable one?	No, I don't consider investment in stock market a reliable one because the behavior of stock market is	I think sometimes investment in stock market can prove a valuable one to investors, but it may	As per my information, investment in stock market is not reliable because there are lots

	very unpredictable.	get affected due to uncertain reasons and investors have to face minor or huge loss.	of ups and downs in its behavior. Also, it gets affected by various factors i.e. political affairs, financial condition of company.
Q10.If you invest in stock market then which level of investor you are?	I am an intermediate level of investor and I know about stock market but not completely.	I am a beginner level of investor. I am investing just to get some information that how stock market works.	I am an experienced investor and I invest bulk of money to buy shares of various companies. I know about the behavior of stock market to a great extent.

3.3. Data Analysis

Firstly, interviewer asked interviewees about their interest in stock market. Each interviewee answered according to his interest. First interviewee said that he is interested in stock market because he belongs to Economics background. Students of Economics and professionals who work in the field of economics take interest in Stock Market and want to know more about it. Second interviewee said that he has no interest in stock market. There can be several reasons of this statement. People who belong to pure sciences background or who don't

want to take risk in case of investment are not interested in stock market. Third interviewee said that currently he is not interested in stock market. This may be due to the reason that a person has some other difficult tasks to do and he can't take interest in such a complex and unpredictable market. Secondly interviewer asked interviewees about their knowledge in stock market. First interviewee does not know much about stock market as he said. This can be due to several reasons that interviewee has not participated in any webinars/seminars on stock market or he has not researched on it to get some information or he may be not interested to get well informed about its updates and trends. Second interviewee knows about important technical indicators e.g. Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence and Divergence (MACD) that are used to track the stock market trends and he also knows about some important factors that affect the behavior of stock market. Third interviewee doesn't have technical knowledge of indicators or other calculations, but he knows some things about stocks. Thirdly interviewer asked that how much interviewee knows about stock market trends. Two of the interviewees know nothing about stock market trends and how these trends are predicted and what technical measures are taken to track these trends. One of the interviewees said that according to him political situation of the country affects the trends of stocks. On fourth number it was asked from people who participated in interview that what are the factors that affect stock trends. First person said that political situation of a country affects the stock trends. Political situation can include various political affairs that are disturbing the normal functioning of state. These political affairs may include protests, terrorism, strikes, etc. These things heavily disturb the economy and business of entire state and ultimately the stock of companies. So, his answer is very much on point. Surely it is one of the factors that affect stock market trends. Second person said that financial condition of company affects the stock trends.

This condition can be tracked by using some important technical indicators such as Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence and Divergence (MACD). These indicators tell about the rise and fall of stocks. Third person says that both factors i.e. financial condition of company and political situation within a country affect the stock market trends. According to my research and opinion both factors contribute towards the movement of stock trends. Then it was asked from people in interview that which news resource is considered best for stocks? Two of the people said that Twitter is the best resource for stock news, and one said that Financial News website is the best one. But according to my research and this era trends, Twitter is the best resource for every kind of news. People now-a-days use social media to express their opinions and views related to a specific issue, product or company. Renowned celebrities and investors tweet about stocks and performance of various companies. So, it can be considered the best resource for stock and financial news. Then it is asked from persons in interview if they would like to invest in stock market? One said that he would not invest in stock market due to uncertain behavior of stock market. Second one said that he would like to invest because he knows about the trends and ups and downs of market so he can buy and sell stocks at perfect time. Third one is not sure about investment right now because he may not want to take risk of investment without having knowledge about such a complex thing. In future he will get some knowledge and will be able to understand stocks in a better way. Then interviewees are asked about app for stocks. Two of the persons suggested mobile app and one asked for web app but the ultimate thing which the researcher concluded from the answers was friendly, attractive and interactive user interface. This interface will help investors to easily see past trends of stocks of various companies and the technical things which are implemented in app will enable people to buy stocks and get profit from their investments.

How people will like to buy stocks? One person did not know about any of the methods that how to buy stocks. The second person would go with traditional method i.e. he would like to go to stockbrokers who with the help of their technical knowledge and past experience will tell you that which stocks one should buy. Third person said that he will search various apps and the app which has maximum rating and good feedback will be chosen for buying stocks. Second last question which is asked in interview is "Is investment in stock market considered reliable". By analyzing three of the interview answers, it is concluded that people are reluctant to invest in stock market and considering it a reliable one. Many of the researchers have tried to understand the behavior of stock market but they did not succeed completely due to the unpredictable behavior of stock market. Then persons who participated in interview are asked about their level and experience of investment. Persons in every level invest in stock market. Experienced investors are the ones who have been investing in stocks for a long time and they have invested a handsome amount of money in various companies.

4. ANALYSIS AND DISCUSSION

Stock market contains hundreds of companies and these vary on the basis of their size. The researcher chose one big company of US stock market namely Apple. Proposed system will give predictions of this company.

Proposed solution works on various modules and major ones of them are automated scrapping of tweets from twitter, sentiment analysis of twitter, data preprocessing from raw data to such form of data that can be used for the system, extracting features from data, feeding processed data to model for training and making predictions. The first phase of the system is to gather tweets related to a specific company (Apple). Tweets were collected by doing web scraping from Twitter. After collecting data, second phase includes preprocessing of data that means to clean data and make it according to the requirements of our system. The third phase includes the classification of tweets as positive, negative or neutral and sentiment analysis of tweets. The fourth phase includes applying models on data and feeding to algorithms that takes input feature such as sentiment of twitter for processing. The algorithms used in model training are Support Vector Machines (SVM) and Naïve Bayes. To improve the accuracy obtained from these algorithms, the researcher tried different approaches and different methods that include "Bag of Words", "TF-IDF", "N-grams" and "Word counts". These algorithms classify tweets as positive, negative and neutral. The fifth phase includes merging sentiment analysis of Twitter with historical data of stocks to predict stock market using sentiment analysis of Twitter. Number of followers for each tweet, Clusters, polarity, sentiment confidence and difference

(difference of stock closing and opening price) are given as input features to SVR and LSTM for predictions of stock market.

The selection of technologies and analytical language was quite tough. It required research to find out that what would be the best language for backend that will work efficiently with our system. For this reason, the researcher chose Python for backend because it works quite efficiently with Machine learning techniques, data analysis, web scrapping, sentiment analysis of twitter and visualizing results in the form of graphs. As mentioned earlier, proposed solution comprises of some modules that include automated scrapping of tweets from twitter, sentiment analysis of twitter, data preprocessing from raw data to such form of data that can be used for our system, extracting features from data, feeding processed data to model for training and then making predictions.

4.1. System Architecture

The flow diagram of my project is as follows:

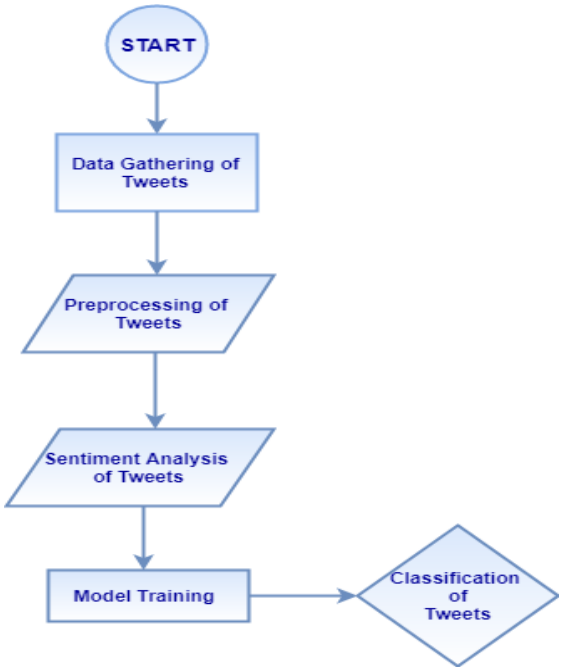


Figure 1 Flow Diagram of Proposed Solution

4.2. Automated Web-scraping for Tweets gathering

The data that the researcher was interested in fetching from the internet was live and authentic news. For that the researcher decided to download such data from twitter by using different tags which the researcher decided by analyzing different channels on the twitter. Initially for this purpose the researcher decides to use twitter API's to extract the data. But soon it came to know that twitter API's does not give access to all the tweets or it is not fulfilling our requirements. In order to overcome this problem, the researcher decided to develop our own data crawler which will crawl the website and scrap the data of the tags which were supposed to use. Data can be extracted from the dynamic web pages without any hustle. It is like creating as sitemap that tells the crawler hoe to extract the data and export them into an CSV file. So, the researcher developed my own data crawler and deployed it. It ran for days to get the tweets from the past till 2014 which scrapped the tweets around 10-12k for every tag.

The researcher faced some problems when it came to tweets scraper. The most important aspect of it was that it should fetch the relevant tweets. As if the tweets are not relevant it will affect the predictions. In order to overcome this problem, the researcher first analyzed tweets for every company and tried to find the best tags for every company. For example, for apple the researcher used AAPL and apple.

- $AAPL Daily: #**AAPL** broke out from magenta resistance as well and is chasing Upper BB. The only horizontal resistance I can make out is 215 at the moment. For now no bearish indicators, should continue to follow Upper BB #APPLE #Tech #Technology

- Corporations like Microsoft and Apple may continue to accumulate huge amounts of cash for the day humans ruin the planet and have to move to Mars. #**AAPL** #MSFT #PlanetEarth
- CrewAi short term swings been killing it, here are two recent plays, $AAPL & $**AMZN** , no opinions, just science, for more info and plays like these go to https://tradecrew.com/crewai , don't postpone profits, get CrewAi.

By taking that approach, the researcher became able to overcome the problem of getting irrelevant news. As the researcher knew the best tags, for Apple's stock news feed on the twitter. By using tags to get data has reduced the irrelevant news. Like in 10 tweets there is hardly 1 irrelevant tweet. Basically, the idea of getting news from the twitter was built on the rationale that it allows us to filter the irrelevant news and help us in getting the relevant tweets.

4.3. Natural Language Processing

Just like humans converse with each other, we can converse with computer as we do with Cortana, Google Assistant using Natural language processing. Being a part of the Artificial Intelligence the NLP has applications such as Sentiment analysis, Content modeling and categorization, Extraction etc. They can be used for spelling and error identification, text classification, chatbots. Natural language can be used in python by using NLTK, TextBlob Library.

4.4. Data preprocessing

This step includes processing of data we're receiving and formatting it into our desired shape. The pre-processing of tweets includes tokenization, removing stop-words, lemmatization, and removal of all non-alphabetical words. In the pre-processing part of it all the tweets files

were generated in a way that every day's tweet was under one file. Here comes the most interesting part of our project. It is also the second major module of the project that processing the data we're receiving and formatting it into our desired shape.

When it came to news data, it really has to be processed as it was not in numerical form. And most importantly it did not include only English sentences. To ensure that the researcher could deal with the problem of not being type-safe, the researcher used default Unicode Transformation Format 8-bit Encoding (UTF-8). Then, the news was parsed, lower-cased, its punctuation removed and lemmatized to ensure cleanliness and non-brevity. So, the researcher used NLTK library for this purpose of removing irrelevant sentences and words which were of no use to us. This includes removal of the stop words, removal of punctuations, and any other operator or word then the alphabets. Also, we did lemmatization of the words using the same library.

Lemmatization was an important step as this step reduces text to its stem and then checks whether this stem is present in the English dictionary or not. This is a very resource intensive step, but we had to do it to make our news cleaner and then use it for net effect calculation. Before lemmatization, the researcher was performing stemming on the news which proved to be bad choice. It works very similarly to lemmatization by converting a word to its stem. Though it was resource friendly, it didn't used to check whether the reduced stem was in the English dictionary or not. This was a bad choice because it used to remove lots of words in the tweets that were fundamental to the financial tweet's sentiment calculation effect. After lemmatization the tweets were broken into the tokens by using the NLTK library. All the preprocessing steps are explained in detail:

- **Removing special characters, numbers and punctuation**

In this step, special characters such as @, #, numbers and punctuation such as !, “”, etc are removed from tweets. For example:

5G reportedly coming to premium iPhones in 2020, all models in 2021 http://dlvr.it/R6mGC3 - @TechCrunch #Apple #iPhone

After removing special characters, numbers and punctuation from above sentence new sentence becomes:

reportedly coming to premium iPhones in all models in http://dlvr.it/R6mGC3 Techcrunch Apple iPhone

- **Removing URLs**

In this step URLs from tweets are removed to clean data and for further processing of tweets. For example:

reportedly coming to premium iPhones in all models in http://dlvr.it/R6mGC3 Techcrunch Apple iPhone

After removing URL above tweet becomes:

reportedly coming to premium iPhones in all models in Techcrunch Apple iPhone

- **Converting all alphabets to lowercase**

This step performs further processing on tweets. In this step all upper case alphabets are converted to lowercase. This is illustrated by following example:

reportedly coming to premium iPhones in all models in Techcrunch Apple iPhone

After converting alphabets to lower case the above tweet becomes:

reportedly coming to premium iphones in all models in techcrunch apple iphone

- **Removing stop words**

This step includes removal of stop words such as a, an, the, from, what, etc. It is shown by following example:

reportedly coming to premium iphones in all models in techcrunch apple iphone

After removing stop words i.e. to, in from above tweet it becomes:

reportedly coming premium iphones models techcrunch apple iphone

- **Tokenization**

Tokenization splits a line of text into smaller parts that are known as tokens. This step is further illustrated by following example:

reportedly coming premium iphones all models techcrunch apple iphone

After tokenization above line of text becomes:

['reportedly' , 'coming' , 'premium' , 'iphones' , 'models' , 'techcrunch' , 'apple' , 'iphone']

- **Stemming**

Different variants of a word are produced morphologically in stemming procedure. E.g. words computing, computed stems to word "compute". This procedure is also illustrated by following example:

['reportedly' , 'coming' , 'premium' , 'iphones' , 'models' , 'techcrunch' , 'apple' , 'iphone']

After stemming word "coming" becomes "come"

['reportedly' , 'come' , 'premium' , 'iphones' , 'models' , 'techcrunch' , 'apple' , 'iphone']

- **Lemmatization**

Lemmatization returns words to lemma which is dictionary form of word by doing analysis according to vocabulary and morphology of words. For example:

['reportedly' , 'come' , 'premium' , 'iphones' , 'models' , 'techcrunch' , 'apple' , 'iphone']

After lemmatization it becomes:

['reportedli' , 'come' , 'premium' , 'iphon' , 'model' , 'techcrunch' , 'appl' , 'iphon']

- **Labeling of Tweets**

For labeling of tweets, I have used library "TextBlob". TextBlob is used for processing of textual data and natural language processing which includes sentiment analysis. TextBlob performs various preprocessing operations on tweets data which include tokenization, removing of stopwords. It has a sentiment classifier which takes the tokens as input and returns the polarity for each tweet from -1 to 1.

4.5. Classification of Tweets

The biggest challenge before classification was to label data because classification algorithms take labeled data as input. So, the researcher did label of data using TextBlob library. Sentiment of each tweet was written against each tweet. This labeled was saved in a .csv file that has to be use in future for classification and prediction purposes.

The researcher used two classification algorithms i.e. Naïve Bayes and Support Vector Machines (SVM) for classification of tweets. The researcher tried different approaches with these algorithms to achieve maximum accuracy. Classifiers take input in the form of numerical data. In order to make these algorithms work with textual data, data is first converted into

numeric form. This can be done using different approaches e.g. Bag of Words, TF-IDF and word-count.

4.5.1. Naïve Bayes

It is a grouping method dependent on Bayes' Theorem with a presumption of independence among indicators. In basic terms, a Naive Bayes classifier expects that the nearness of a specific component in a class is irrelevant to the nearness of some other element. For instance, a natural product might be viewed as an apple on the off chance that it is red, round, and around 3 creeps in distance across. Regardless of whether these features rely upon one another or upon the presence of different features, these properties autonomously add to the likelihood that this natural product is an apple and that is the reason it is known as 'Naive'. It's called naive because it assumes that all of the predictors are independent from one another. Naive Bayes is mostly used for binary or multiclass classification. They provide us a way of calculating the probability of posterior. Naïve Bayes is based on Bayes Theorem. Bayes Theorem works on “conditional probability”. This probability says that if something has already occurred then something will happen. The formula of conditional probability is as follows:

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

4.5.2. Support Vector Machine (SVM)

SVM is a Machine Learning Algorithm that does analysis on textual data to classify it and regression analysis of numerical data. It works on labeled data as it is a supervised learning algorithm and it classifies data separately into separate classes. Applications of SVM include classification of text, classification of images, document classification, etc. SVM is defined by a

hyperplane which separates different classes. It gives hyperplane as an output which differentiates different classes and puts each class into a separate category. This hyperplane lies in a two-dimensional space which separates the plane into two parts and each class lay on any one of the either side. For example, based on the height and weight of an individual each and every point or feature is mapped onto an n dimensional space and we are trying to classify them into two different classes by using a hyperplane. Most importantly what you need to understand here is that we are trying to build this hyperplane so that the two classes that are separated as wide as possible. SVM can only be used on data that is linearly separable (i.e. a hyper-plane can be drawn between the two groups). There are established ways to do it, they are called Kernels. By using a combination of these Kernels, and tweaking their parameters, you'll most likely achieve better results than making up your own way. The advantage of SVMs are that you can use them, in case of features, when compared, you can use very little data as in each of your data points has. SVM uses a set of functions of mathematics that is known as kernel. Kernel takes data in the form of input and then performs different transformations on data to get the required form of data. Different forms of kernel are linear, non-linear, radial basis function (RBF), etc.

Classifiers take input in the form of numerical data. In order to make these algorithms work with textual data, data is first converted into numeric form. This can be done using different approaches e.g. **Bag of Words, TF-IDF, N-grams** and **Word-count**.

Bag of Words

For example, following are three tweets:

- Tweet1: Apple is going to launch some new features in MAC Operating System.
- Tweet2: Samsung has revealed its new mobile model, but apple has not yet.

All unique words are gathered in a vocabulary in Bag of Words approach. For the above given example Bag of Words will be:

- Vocabulary= [Apple, is, going, to, launch, some, new, features, in, MAC, Operating, System, Samsung, has, revealed, its, mobile, model, but, not, yet]

In the next step each tweet is converted to a feature vector. In feature vector, if a word is in vocabulary and it is also found in tweet then that word is assigned number "1" in feature vector and if word is in vocabulary but it is not present in tweet then number "0" is assigned to that word in feature vector. Feature vector for Tweet2 is:

- Feature Vector: [1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]

TF-IDF

TF-IDF approach gives more weightage to rare (specific) terms and less weightage to uncommon and irrelevant terms.TF stands for "Term Frequency" and IDF stands for "Inverse Document Frequency". Term Frequency is calculated by following formula:

$$Term\ Frequency\ (TF) = \frac{Frequency\ of\ a\ word\ in\ the\ document}{Total\ Number\ of\ words\ in\ the\ document}$$

Inverse Document Frequency is calculated by following formula:

$$Inverse\ Document\ Frequency\ (IDF) = \log\left(\frac{(Total\ number\ of\ documents)}{(Total\ Number\ of\ documents\ containing\ the\ word)}\right)$$

N-Grams

When we use only one-word feature for our model, we are using unigrams or 1-grams as a feature. But when we use more than one words or word sequence of two or three we are actually improving the predictive power of classifier. For example, if a sentence is "Don't like

chocolate" this sentence has word "like" which contributes towards positive sentiment but if we take all three words then this sentence has negative sentiment overall.

Word-Counts

This approach says that how many times a word appears in a sentence contributes more towards the overall sentiment of sentence. If a sentence has more positive words, then sentence has positive sentiment. For example, if a sentence has words "good", "top", "high-quality" occurring repeatedly in a sentence then that sentence has positive sentiment. This is how word-count increases the predictive power of classifier.

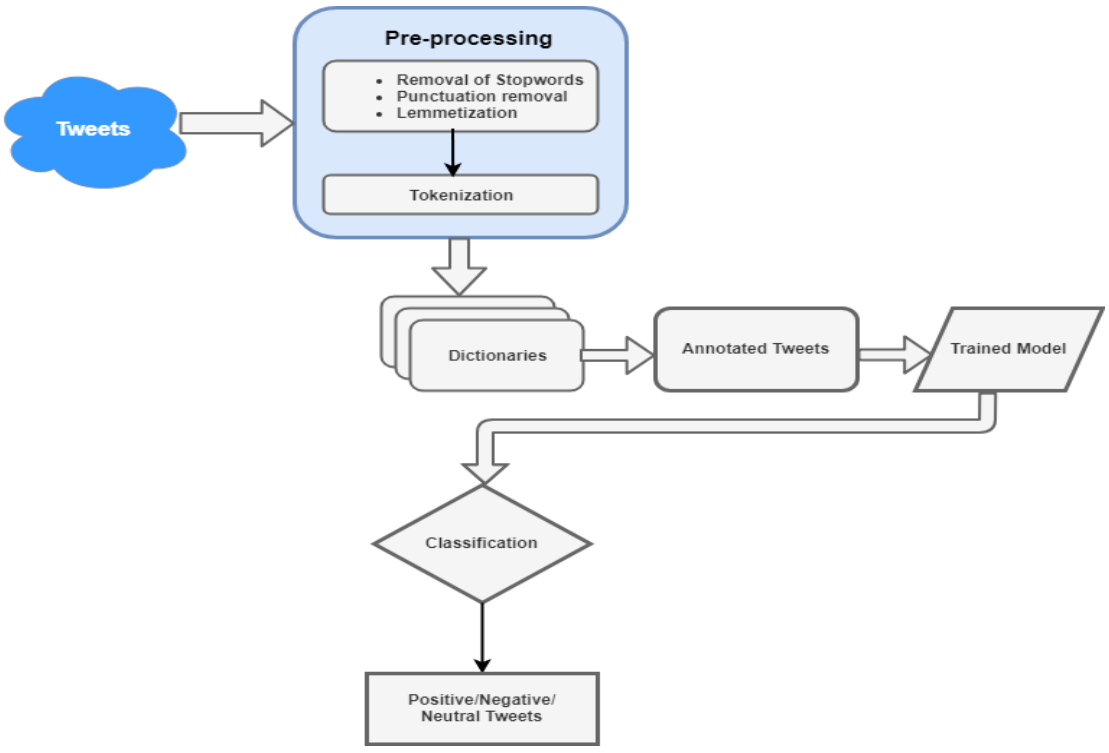


Figure 2 High level diagram of sentiment analysis

4.6. Model Training using Historical data of Stocks and Sentiment Analysis of Twitter

After the preprocessing of tweets, sentiment analysis of tweets is done using TextBlob library of Python. TextBlob returns polarity and sentiment of tweets. Then different clusters are formed of tweets depending upon the polarity and sentiment value of tweets using k-means algorithm. Historical data of stocks is downloaded from Yahoo Finance which contains the opening and closing price of stocks. Difference of opening and closing price of stock is calculated to give as input to Machine Learning models. Five features i.e. followers, polarity, sentiment confidence, clusters and difference (difference of opening and closing price of stock) are given as input to SVR and LSTM. Two algorithms Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) are used to give predictions for stock Market.

4.6.1. Long Short-Term Memory (LSTM)

Long Short-Term Memory is a special kind of Recurrent Neural Network that is efficient enough to learn long-term events. RNN work similar to LSTM but they cannot work for long-term dependencies. For example, if we have a long sentence of English and we want to predict the last word of sentence then RNN would not be able to remember the long-term information and may not give right prediction. LSTM is designed to overcome that problem. E.g. If we have a sentence “I live in Canada and my nationality is Canadian” LSTM will successfully predict the last word of sentence by remembering the entire information but RNN can only store recent information. The activation function used with LSTM in my implementation is “ReLu (Rectified Linear Unit)”. It gives zero output when input is equal to or less than zero. Otherwise it gives the same output as input. Long short-term memory networks or LSTM’s are designed for

applications where the input is an ordered sequence where information from earlier in the sequence may be important. LSTM's are a type of recurrent network which are networks that reuse the output from a previous step as an input for the next step. Like all neural networks the node performs a calculation using the inputs and returns an output value in a recurrent Network. This output is then used along with the next element as the inputs for the next step and so on. In an LSTM the nodes are recurrent, but they also have an internal state the node uses an internal state as a working memory space which means information can be stored and retrieved over many time steps. The input value previous output and the internal state are all use in the node's calculations. The results of the calculations are used not only to provide an output value but also to update the state. Like any neural network LSTM nodes have parameters that determine how the inputs are used in the calculations. So, LSTM nodes are certainly more complicated than regular recurrent nodes, but this makes them better at learning the complex interdependencies in sequences of data and ultimately, they're still just a node with a bunch of parameters and these parameters are learned during training just like with any other neural network.

4.6.2. Support Vector Regression (SVR)

Support Vector Regression (SVR) is very similar to Support Vector Machine (SVM) but it works on continuous data. The researcher chose SVR because he need to train it on numerical data of stocks and Twitter. Kernel of SVR is a function that changes the dimension of data from low to high. Hyperplane in SVR will help us in the prediction of continuous value. The data points which have minimum distance from boundary are known as support vectors. The motive of Support vector algorithm is without limiting or minimizing the size of violations on the margins between the two classes, they insist on involving or including the maximum data points

or instances between the margins while the margin the is minimized. The linear regression is performed by Support vector regression in high dimensional space. In support vector regression, in the training data set side of the hyperplane each of the instance or the data points represent their own dimension. In support vector regression, all the data point in the training data set of the hyperplane are evaluated and the higher dimensional sided all the test points have given the representation of ‘k’. The main aim of support vector regression or SVM is to limit the errors to min, where the errors are limited by maximizing the margins, coherency and without affecting the hyperplane. The advantage of Support vector regression over support vector machine that we are able to extend it to nonlinear data points where normal linear SVM cannot be applied.

5. IMPLEMENTATION

Our Project aims to assist investors about taking decisions for stocks of a specific company. By using Machine learning algorithms with sentiment analysis using Python as main programming language, our system ensures proper assistance for its users.

The backend of project operates on tweets of companies that were scrapped from twitter. The tweets are first preprocessed for sentimental analysis and classification of tweets using Machine Learning algorithms. The project focuses on only one big company of US Stock Market namely Apple (AAPL) by predicting their stock prices using sentiment analysis of tweets. Firstly, Machine Learning algorithms Support Vector Machines (SVM) and Naïve Bayes are used for classification of tweets and then accuracy of these algorithms is increased by using different approaches. Stock Market prediction is done using sentiment analysis of twitter and using historical data of stocks. In this implementation the researcher will lay a set of requirements and provide instructions on how to create and manage the virtual machines.

5.1. Software Requirements

- Backend Development

The entire backend is implemented using programming language “Python”. The version which is used in implementation is Python 3.6.

- PYTHON Libraries

Pandas:

It is an open source Python library that helps programmers a lot in manipulating data-structures. It also provides tools for data analysis in Python.

Re:

This is a module provided by Python for supporting regular expressions. It is used to find out a string or set of strings that match the sequence of characters in the pattern of regular expression.

Sklearn:

This library is used to get function “CountVectorizer”. This function extracts bag of words from tweets.

Numpy:

This library in Python is used to manipulate arrays and to perform operations on arrays. These operations include mathematical and many other operations.

Nltk:

This is used for pre-processing of tweets. For example, it provides functions to remove stop words from tweets and for performing stemming on tokenized words.

Matplotlib:

This is a Python library which is used to plot different types of graphs such as bar charts, pie graphs, scatter plot, etc, One can set plot size, figure size, labels and legends using this library in Python.

Tensorflow:

This library is used in the project to import very famous Deep Learning Algorithm Long Short-Term Memory LSTM. This algorithm is used in training both types of data i.e. sentiment data of twitter and historical data of stocks.

5.2.Hardware Requirements

OS	Windows 7, Windows 8 or Windows 10
RAM	4 GB
Free Disk Space	1 GB
PC	Laptop or Desktop Computer with Internet Connection

Other Requirements

Along with mentioned software and hardware requirements, other requirements include internet connection, web browsers e.g. Google Chrome, Mozilla Firefox, Internet Explorer, etc.

5.3. Script for extracting tweets from Twitter

In the following part, the researcher is only including supporting libraries that will scrap tweets from twitter:

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import csv
```

In the following lines, the researcher is opening Google Chrome browser through Python script and giving this browser URL of Twitter for a specific tag:

```
browser = webdriver.Chrome()

# write url here
# browser.get("https://twitter.com/hashtag/apple?f=news&vertical=news&src=rela")
browser.get("https://twitter.com/search?f=news&vertical=default&q=%23appl&src=typd")
```

In the following lines of code, Loop is running as many times as one wants to load the page of browser and get required number of tweets:

```
#Run this loop as many times as you want to load the page
elm = browser.find_element_by_tag_name("html")
for x in range (3000):
    print(x)
    elm.send_keys(Keys.END)
    time.sleep(10)
    elm.send_keys(Keys.HOME)
time.sleep(10)
```

In the following piece of code, the researcher is scrapping Tweets and Timestamp using CSS selector and saving tweets and time in “tweet_element” and “date_element” respectively.

```
# scrape the content by CSS SELECTOR
tweet_element = browser.find_elements(By.CSS_SELECTOR,'p[class="TweetTextSize  js-tweet-text tweet-text"]')
date_element = browser.find_elements(By.CSS_SELECTOR,'a[class="tweet-timestamp js-permalink js-nav js-tooltip"]')
```

Following chunk of code writes tweets along with timestamp in a csv file using function writerow():

```
# writing data to csv file
with open('Tweets_data_2.csv', 'w', newline='', encoding='utf-8') as csvfile:
    autowriter = csv.writer(csvfile)
    autowriter.writerow(['Time','Tweets'])
    for i in range(0,len(tweet_element)-1):
        tweet_text = tweet_element[i].text.encode("utf-8")
        autowriter.writerow([date_element[i].get_attribute("title"), tweet_text])
```

5.4.Script for Cleaning/ Pre-processing of Tweets

This piece of code only imports libraries and reads dataframe from .csv file.

```
import numpy as np
from nltk.stem.porter import *
stemmer = PorterStemmer()
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
from nltk.corpus import stopwords
# df=pd.read_csv('data_dup.csv')
df=pd.read_csv('appletweets.csv')
df=df.dropna(axis=0,how='any')
stop_words = stopwords.words('english')
```

Following code performs different cleaning operations on Twitter data which includes removal of stopwords, punctuation, special characters, tokenization, stemming,etc:

```
class Twitter():
    def clean_tweet(self):
        def remove_pattern(input, pattern):
            r = re.findall(pattern, input)
            for i in r:
                input = re.sub(i, '', input)
            return input
```

```
# remove twitter handles (@user)
        df['cleaned_tweet'] = np.vectorize(remove_pattern)(df['Tweets'], "@[\w]*")
        # remove special characters
        df['cleaned_tweet'] = df['cleaned_tweet'].str.replace("[^a-zA-Z#]", " ")
        # remove words less than length 3
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: ' '.join([word for word in i.split() if
len(word) > 3]))
        # remove URLs
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: re.split('https:\/\/.*', str(i))[0])
        df['cleaned_tweet'] = df['cleaned_tweet'].replace(r'[^A-Za-z0-9 ]+', '', regex=True)
        # remove numbers
        df['cleaned_tweet'] = df['cleaned_tweet'].str.replace(r'\d+', '')
        # remove special character hashtag"#"
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: i.replace('#', ' '))
        # convert all uppercase letters to lowercase
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: i.lower())
        # Tokenization
        tokens = df['cleaned_tweet'].apply(lambda x: x.split())
        tokens.head()
        tokens = tokens.apply(lambda y: [stemmer.stem(i) for i in y])  # stemming
        # df['tokens']=tokenized_tweet
        # df.to_csv('Tokens.csv')
        for i in range(len(tokens)):
            tokens[i] = ' '.join(tokens[i])
        df['cleaned_tweet'] = tokens
        cleaned_tweets = tokens
        # writing cleaned tweets to .csv file
        df.to_csv('cleaning2.csv')
        return cleaned_tweets

if __name__ == '__main__':
   tw=Twitter()
   tw.clean_tweet()
```

5.5. Script for Labeling Tweets

Following piece of code finds out the sentiments of tweets using TextBlob library. It checks if the polarity of a tweet is less than zero then tweet is negative, if polarity is greater than zero then tweet is positive and if it is equal to zero then tweet is neutral.

```
from textblob import TextBlob
import pandas as pd
import csv
df=pd.read_csv('cleaning2.csv')
cleaned_tweets=df['cleaned_tweet']
class Labelling():
    def get_sentiment(self,cleaned_tweets):

        # sentiment analysis of tweets using TextBlob
        analysis = TextBlob(cleaned_tweets)

        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'
```

Following function “get_tweets()” takes cleaned tweets and store text of tweets and sentiments in separate columns of dataframe:

```
def get_tweets(self):

    #empty list to store tweets
    list_of_tweets = []
    # iterating through tweets
    for tweet in cleaned_tweets:
        # empty dictionary to store tweets' text and sentiment
        tweet_dic = {}
        # storing text part of tweet
        tweet_dic['Tweets'] = tweet
        # storing sentiment of tweet
        tweet_dic['Sentiment'] = self.get_sentiment(tweet)
```

In main function def main (), positive tweets are labeled as “positive”, negative tweets are labeled as “negative” and neutral tweets as “neutral.” Then all tweets are written along-with their labels in csv file.

```
# appending parsed tweet to tweets list
            if tweet_dic not in list_of_tweets:
                list_of_tweets.append(tweet_dic)

                # return list of tweets
        return list_of_tweets

def main():

    # creating object of Labelling Class
    lab =Labelling()

    # calling function to get tweets
    tweets_data = lab.get_tweets()

    # selecting pos tweets from all gathered tweets
    ptweets_text=[tw for tw in tweets_data if tw['Sentiment'] == 'positive']

    # percentage of positive tweets
    print("Percentage of Positive tweets : {} %".format(100 * len(ptweets_text) / len(tweets_data)))

    # picking negative tweets from tweets
    ntweets_text=[tw for tw in tweets_data if tw['Sentiment'] == 'negative']

    # percentage of negative tweets
    print("Percentage of Negative tweets : {} %".format(100 * len(ntweets_text) / len(tweets_data)))

    # percentage of neutral tweets
    neutral_text=[tweet for tweet in tweets_data if tweet['Sentiment'] == 'neutral']
    tweet_length = len(tweets_data)
    nlength = len(ntweets_text)
    plength = len(ptweets_text)
    print("Percentage of Neutral tweets : {} % ".format(100 * (tweet_length - nlength - plength) /
tweet_length))

    csv_columns=['Tweets','Sentiment']
```

```
        with open('Labelled_tweets.csv', 'a') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames=csv_columns)
    writer.writeheader()
    for data in neutral_text:
        writer.writerow(data)

if __name__ == '__main__':
    main()
```

5.6. Model Training Using SVM

Following piece of code imports various libraries that are necessary for model training and reading dataframes from .csv files.

```
import pandas as pd
from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from nltk.stem.porter import *
stemmer = PorterStemmer()
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")
df1=pd.read_csv('final_train.csv')
df2=pd.read_csv('final_test.csv')
df1=df1.dropna(axis=0,how='any')
df2=df2.dropna(axis=0,how='any')
df=pd.read_csv('output_data.csv')
```

Following lines create a feature matrix for “Bag of Words” using “CountVectorizer” function:

```
bow= CountVectorizer(max_df=0.90, min_df=2, max_features=3000,
stop_words='english')
# Extracting features using bag of words approach
bag_of_words = bow.fit_transform(df['Tweets'])
print(bag_of_words)
```

These lines train SVM using n-gram approach. N-grams increase the predictive power of classifiers because these find out the overall sentiment of word range i.e. two to three words.

```
ngram= CountVectorizer(binary=True, ngram_range=(1, 2))
ngram.fit(df['Tweets'])
X = ngram.transform(df['Tweets'])
X_test = ngram.transform(df2['Tweets'])
output=df['Sentiment']
X_train, X_val, y_train, y_val = train_test_split(
    X, output, train_size=0.75
)

c=1.0    #85.76
svm = LinearSVC(C=c)
svm.fit(X_train, y_train)
print("Accuracy with SVM(ngrams)  for C=%s: %s"
      % (c, accuracy_score(y_val, svm.predict(X_val))))
```

Following piece of code train SVM using TF-IDF approach. TF-IDF increases accuracy because it gives more weightage to specific and relevant terms than irrelevant and common terms:

```
tfidf = TfidfVectorizer()
tfidf.fit(df['Tweets'])
X = tfidf.transform(df['Tweets'])
X_test = tfidf.transform(df2['Tweets'])
target=df['Sentiment']
X_train, X_val, y_train, y_val = train_test_split(
    X, target, train_size=0.75
)

c=6.0 #86.1%
svm = LinearSVC(C=c)
svm.fit(X_train, y_train)
print("Accuracy with SVM(Tfidf)  for C=%s: %s"
      % (c, accuracy_score(y_val, svm.predict(X_val))))
```

5.7. Model Training using Naïve Bayes

Following lines of code just import supporting libraries for model training:

```
import pandas as pd
import numpy as np
from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from nltk.stem.porter import *
stemmer = PorterStemmer()
import re
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")
df=pd.read_csv('output_data.csv')
```

In following lines, Naïve Bayes is classifying tweets using TF-IDF approach:

```
vect = CountVectorizer()
vect_count = vect.fit_transform(df['Tweets'])
tf_idf = TfidfTransformer().fit(vect_count)
vect_count = tf_idf.transform(vect_count)
print(vect_count)
X_train, X_test, y_train, y_test = train_test_split(vect_count, df['Sentiment'], test_size=0.25)
model_NB = MultinomialNB().fit(X_train, y_train)
predictions = model_NB.predict(X_test)
print("Accuracy with Naive Bayes (Tfidf):  %s"
      % ( accuracy_score(y_test, predictions)))
```

Using word-counts approach, Naïve Bayes is classifying tweets in following code:

```
word_count = CountVectorizer(binary=False)
word_counts = word_count.fit_transform(df['Tweets'])
X_train, X_test, y_train, y_test = train_test_split(word_counts, df['Sentiment'], test_size=0.25)
model_NB = MultinomialNB().fit(X_train, y_train)
predictions = model_NB.predict(X_test)
print("Accuracy with Naive Bayes (word_count):  %s"
% ( accuracy_score(y_test, predictions)))
```

Using n-grams approach, Naïve Bayes is classifying tweets in following code:

```
ngram= CountVectorizer(binary=True, ngram_range=(2, 3))
ngram.fit(df['Tweets'])
X = ngram.transform(df['Tweets'])
output=df['Sentiment']
X_train, X_test, y_train, y_test = train_test_split(
    X, output, train_size=0.75
)
model_NB = MultinomialNB().fit(X_train, y_train)
predictions = model_NB.predict(X_test)
print("Accuracy with Naive Bayes (N-grams):  %s"
      % ( accuracy_score(y_test, predictions)))
```

5.8. Stock Prediction using Support Vector Regression

```
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
import glob
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")

files = glob.glob("Final_Data/*.csv")
frame = pd.DataFrame()
list_of_files = []
for file in files:
    df = pd.read_csv(file,index_col=0,encoding='latin-1')
    df = df.drop('text', 1)
    df = df.drop('date', 1)
    list_of_files.append(df)
frame_of_files = pd.concat(list_of_files)
frame_of_files.dropna(axis=1,how='any',inplace=True)
inputs = frame_of_files.drop('difference',1)
X= np.array(inputs)
Y= np.array(frame_of_files['difference'].values)
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=0)
svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)
y_rbf = svr_rbf.fit(X_train, y_train).predict(X_test)
print('Root Mean Squared Error for SVR(rbf) :', np.sqrt(metrics.mean_squared_error(y_test,
y_rbf)))
```

5.9. Stock Prediction using LSTM

```
from numpy import array
from numpy import hstack
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
import pandas as pd
from sklearn import metrics
import numpy as np
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")
df_inputs=pd.read_csv('inputs.csv')
df_outputs=pd.read_csv('output.csv')
df = df_inputs.drop('text', 1)
df = df_inputs.drop('date', 1)
foll=[i for i in df['followers']]
pol=[i for i in df['polarity']]
senti=[i for i in df['sentiment_confidence']]
clus=[i for i in df['clusters']]
output=[i for i in df_outputs['difference']]

# Splitting of multivariate sequence data into number of samples
def split_sequences(seq, no_of_steps):
   X, y = list(), list()
   length_seq=len(seq)
   for z in range(length_seq):
      # find the end of this pattern
      end_data = z + no_of_steps
      # checking if we are not out of the bounds of dataset
      if end_data > len(seq):
         break

      # gather input and output parts of the pattern
      sequence_x, sequence_y = seq[z:end_data, :-1], seq[end_data-1, -1]
      X.append(sequence_x)
      y.append(sequence_y)
   return array(X), array(y)
```

```
# defining inputs for model
input_sequence1= array(foll[0:-20])
input_sequence2= array(pol[0:-20])
input_sequence3= array(senti[0:-20])
input_sequence4= array(clus[0:-20])
output_sequence= array(output[0:-20])
# convert to [rows, columns] structure
input_sequence1 = input_sequence1.reshape((len(input_sequence1), 1))
input_sequence2 = input_sequence2.reshape((len(input_sequence2), 1))
input_sequence3 = input_sequence3.reshape((len(input_sequence3), 1))
input_sequence4 = input_sequence4.reshape((len(input_sequence4), 1))
output_sequence = output_sequence.reshape((len(output_sequence), 1))
```

```
# Stacking columns horizonatlly
data = hstack((input_sequence1, input_sequence2, input_sequence3, input_sequence4,
output_sequence))

# setting no. of time steps
no_of_steps = 1

# converting sequences into input and output
X, y = split_sequences(data, no_of_steps)
print(X)
print(X.shape[2])

# the dataset knows the number of features
no_of_features = X.shape[2]

# defining model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(no_of_steps, no_of_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

# fitting model
model.fit(X, y, epochs=100, verbose=1)
```

```
# demonstration of prediction
yhat=[]
output=([[foll[-20+x], pol[-20+x], senti[-20+x], clus[-20+x]] for x in range(0,20)])
for i in output:
    input = array(i)
    input = input.reshape((1, no_of_steps, no_of_features))
    yhat.append((model.predict(input, verbose=0)))
results=[list(i)[0][0] for i in yhat]
orig=output[-20:]
print(input)
print(results)
for i in range(len(orig)):
    print(100*((orig[i]-results[i])/orig[i]))

print('Root Mean Squared Error for LSTM :', np.sqrt(metrics.mean_squared_error(input,
results)))
```

5.10. Merging Twitter and Stock Data

```
import pandas as pd
df = pd.read_csv("Cluster_Data/"+'apple'+"cluster.csv",index_col=0,encoding='latin-1')
def get_stock_value(diff, foll, total_foll, polar,senti, curr_price):
    stock_value = diff*(foll/total_foll)*polar*senti/(curr_price)*10000
    print(stock_value)
    return stock_value
apple=pd.read_csv('AAPL(2).csv')
current_price = apple['Open'][0]
difference = abs(apple['Open'][0] - apple['Close'][0])
total_followers = df['followers'].sum()
for i,row in df.iterrows():
    foll = df['followers'][i]
    pol = df['polarity'][i]
    confi = df['sentiment_confidence'][i]
    df.at[i,'difference'] = get_stock_value(difference, foll, total_followers, pol, confi,
current_price)
df.to_csv('Final_Data/'+'apple_stocks'+'.csv')
print(df.head())
```

6. RESULTS AND DISCUSSION

6.1. Data Gathering of Tweets

Tweets for company Apple are gathered using two tags “#apple” and “#appl”. Ten thousand tweets were scraped from Twitter by running my own written script for days. These tweets were then saved in a csv file along with their timestamps as it is shown in following figure:

Time	Tweets
########	@petenajarian Pete we r on the same page my friend,AAPL & holding for long term,I think AAPL is cheap now,+add the 8 upgrade=Giddy UP
########	Just-in-time delivery is good for supply chains, bad for your signature smartphone's OS. $AAPL
########	$AAPL
########	Analyst tells other pundits to 'quit worrying' about $AAPL's possible #iPhone8 delays http://appleinsider.com/articles/17/07/18/analyst-tells-other-pundits-to-quit-worrying-about-apples-possible-iphor
########	SOLID SWEEPER ACTION EARLY, ALOT OF IT REPEAT BUYING AT HIGHER LEVELS >> $BABA $AMBA $NUE $AAOI $YNDX $NTNX $AAPL $MNRO $MBI $AES
########	beat me to it; NFLX has same problem as AAPL: its primary products are loss leaders for other larger peers provided for ~free to consumer
########	Possible option on $AAPL for intraday (today/tomorrow) >150.05.
########	.@UBS sees $900 #iPhone8 growing $AAPL sales by 15% http://appleinsider.com/articles/17/07/18/ubs-sees-900-iphone-8-growing-apples-sales-by-15 â€¦
########	#Apple: acquisitions since 2010- $aapl #tech #wearables #ai #ecommerce #siri #iot #4ir #fintech #machinelearning #lattice @BourseetTrading
########	Couldn't help myself, took some $AAPL 152.5C for next week @ .90. price >150.05 for 151.13 area, possible rotate to 151.99
########	$AAPL Back over the 50D, like this one to play catch up now. http://schrts.co/GzHjgV
########	$AAPL this plan was correct on the breakout and on the buy the dip advice a double win to those that $STUDY it
########	https://www.youtube.com/watch?v=2qLuerYx2IA â€¦ FAST FORWARD TO 5min20sec and tell me stove jobs didn't talk about @Twitter in 1985? @Apple $aapl $twtr
########	Eying $AAPL 155c+157.50's + $BABA 155's+157.50's for next week JUL28. $AAPL needs to hold that 50MA - will look to enter around there

Figure 3 Sample of gathered tweets

6.2. Cleaning of Tweets

After collecting tweets, gathered tweets were cleaned by removing stopwords, URLs, punctuation, tokenization and stemming. Natural Language Toolkit (NLTK) was used in cleaning and pre-processing of tweets. Cleaned tweets were then saved in a csv file as shown in following figure:

Time	Tweets	cleaned_tweet
########	@petenaj	pete same page friend aapl hold long term think aapl cheap upgrad giddi
########	Just-in-tin	just time deliveri good suppli chain your signatur smartphon aapl
########	$AAPL	aapl chang http wallstjesu market updat
########	Analyst te	analyst tell other pundit quit worri about aapl possibl iphon delay http appleinsid articl analyst tell other pundit quit worri about appl possibl iphon delay
########	SOLID SW	solid sweeper action earli alot repeat buy higher level baba amba aaoi yndx ntnx aapl mnro
########	beat me t	beat nflx same problem aapl primari product loss leader other larger peer provid free consum
########	Possible c	possibl option aapl intraday today tomorrow
########	.@UBS se	see iphon grow aapl sale http appleinsid articl see iphon grow appl sale
########	#Apple: a	appl acquisit sinc aapl tech wearabl ecommerc siri iot fintech machinelearn lattic
########	Couldn't h	couldn help myself took some aapl next week price area possibl rotat
########	$AAPL Ba	aapl back over like thi play catch http schrt gzhjgv
########	$AAPL thi	aapl thi plan correct breakout advic doubl those that studi
########	https://w	http youtub watch qlueryx fast forward tell stove job didn talk about aapl twtr
########	Eying $AA	eye aapl baba next week aapl need hold that will look enter around there
########	2017	total return tsla nflx amzn aapl googl
########	$QCOM (-	qcom appl aapl iphon manufactur join legal counter against qualcomm
########	LIVE: $AA	live aapl face competit china from huawei other http yhoo ucut
########	Join @Rot	join both share stock like aapl free make sure link
########	$QQQ $A	aapl semiconductor softwar which industri lead tech sector http amigobul news semiconductor softwar which industri lead tech sector
########	Strong su	strong summer aapl seen mute financi effect late iphon launch http appleinsid articl strong summer appl seen mute financi effect late iphon launch

Figure 4 Cleaned Tweets

6.3. Labeling of Tweets

After cleaning data, now comes the step of labeling tweets by doing sentiment analysis. This is done using TextBlob library. It does sentiment analysis of tweets and label tweets according to the polarity of Tweets' sentiments. Tweets with polarity greater than zero are labeled as "positive", polarity with less than zero are labeled as "negative" and polarity with equal to zero are labeled as "neutral". Labeling of Tweets is shown in following figure:

Tweets	Sentiment
chart mas	positive
tech aapl	positive
nutshel p	neutral
happen a	neutral
hiph lalpi	positive
thi case d	neutral
bank ame	positive
imagin to	positive
these bab	neutral
australia s	positive
appl scien	neutral
said thi p	positive
time burn	neutral
held level	positive

Figure 5 Sample of Labeled Tweets

6.4. Results of Classification using SVM

Firstly, the researcher applied SVM using N-grams approach that is discussed in detail in previous chapters. N-grams approach says that using more than one gram or one word feature for finding out the sentiment of entire sentence. The researcher used range of one to two grams means one to two word features for determining the sentiment of entire sentence.

Regularization parameter was adjusted by trying different values of "C". With N-grams SVM gave accuracy of 91.2% at "C=1.0". Following screenshot shows the accuracy of SVM with n-grams:

```
Accuracy with SVM(ngrams)  for C=1.0: 0.9121171770972037
```

Figure 6 Accuracy of SVM with N-Grams

Then the researcher applied SVM using TF-IDF approach that is also described in detail in previous chapters. TF-IDF approach gives more weight to rare and specific terms than common and irrelevant terms. This is the reason that SVM gives more accuracy with TF-IDF. Following screenshot shows the accuracy of SVM with TF-IDF:

```
Accuracy with SVM(Tfidf)  for C=6.0: 0.9325343985796716
```

Figure 7 Improved Accuracy of SVM with TF-IDF

6.5. Results of Classification using Naïve Bayes

In Naïve Bayes, the researcher has used two approaches that are N-grams and Word-counts. N-grams approach is explained in previous paragraph. the researcher will explain about word-counts approach. Word counts approach says that if a positive word occurs many times in a sentence it means this sentence has positive sentiment overall. The researcher got 73.7%

accuracy using N-grams approach. Following figure shows the accuracy of Naïve Bayes with N-grams:

```
Accuracy with Naive Bayes (N-grams):  0.7376830892143809
```

Figure 8 Accuracy of Naive Bayes (N-grams)

Then the researcher applied Word Counts approach. It gives more accuracy than N-grams because in word counts approach if a sentence contains more positive words, it means sentence has positive sentiment overall. Following is the screenshot that shows the improved accuracy of Naïve Bayes with word counts:

```
Accuracy with Naive Bayes (word_count):  0.7833999112294718
```
k

Figure 9 Improved Accuracy of Naive Bayes

6.6. Results of Support Vector Regression

SVR is applied on historical data of stocks and Twitter data and root mean squared error which the researcher obtained from SVR is as follows:

```
Root Mean Squared Error for SVR(rbf) : 0.0857704723910507
```

Figure 10 Error of SVR

6.7. Results of LSTM

To get more accurate results of prediction the researcher used LSTM (Long Short Term Memory) and the researcher got the following loss which is actually mean squared error of true and predicted values.

```
loss: 2.1490e-05
```

Figure 11 Mean Squared Error of LSTM

6.8. Graph showing actual and predicted prices of stock

Following graph shows the actual and predicted prices of “Apple” stock. Red curve shows the predicted prices and blue curve shows the actual prices.

Figure 12 Graph of actual and predicted prices of Apple stock

6.9. Graph showing percentages of positive, negative and neutral tweets

Following graph is generated from labeled data using Python library "matplotlib". As it is shown in graph, green color shows positive tweets, yellow color shows neutral tweets and red color shows negative tweets.

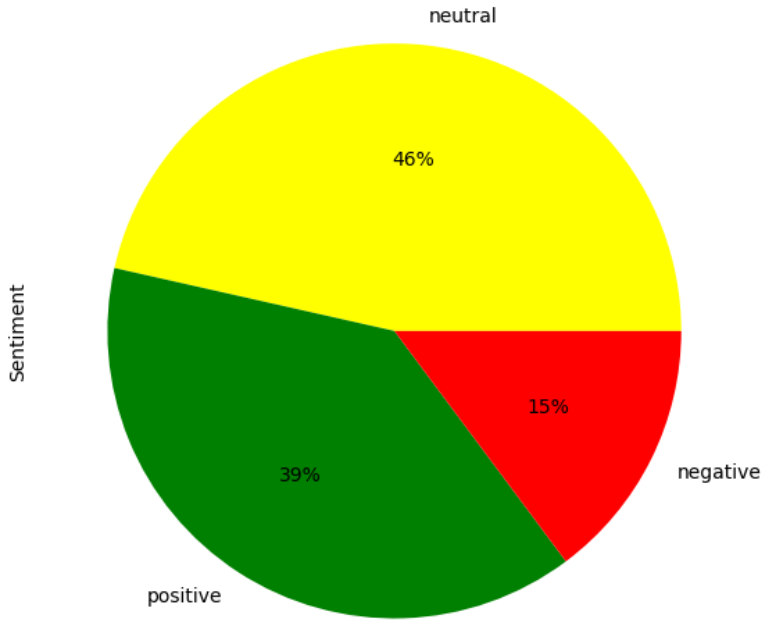


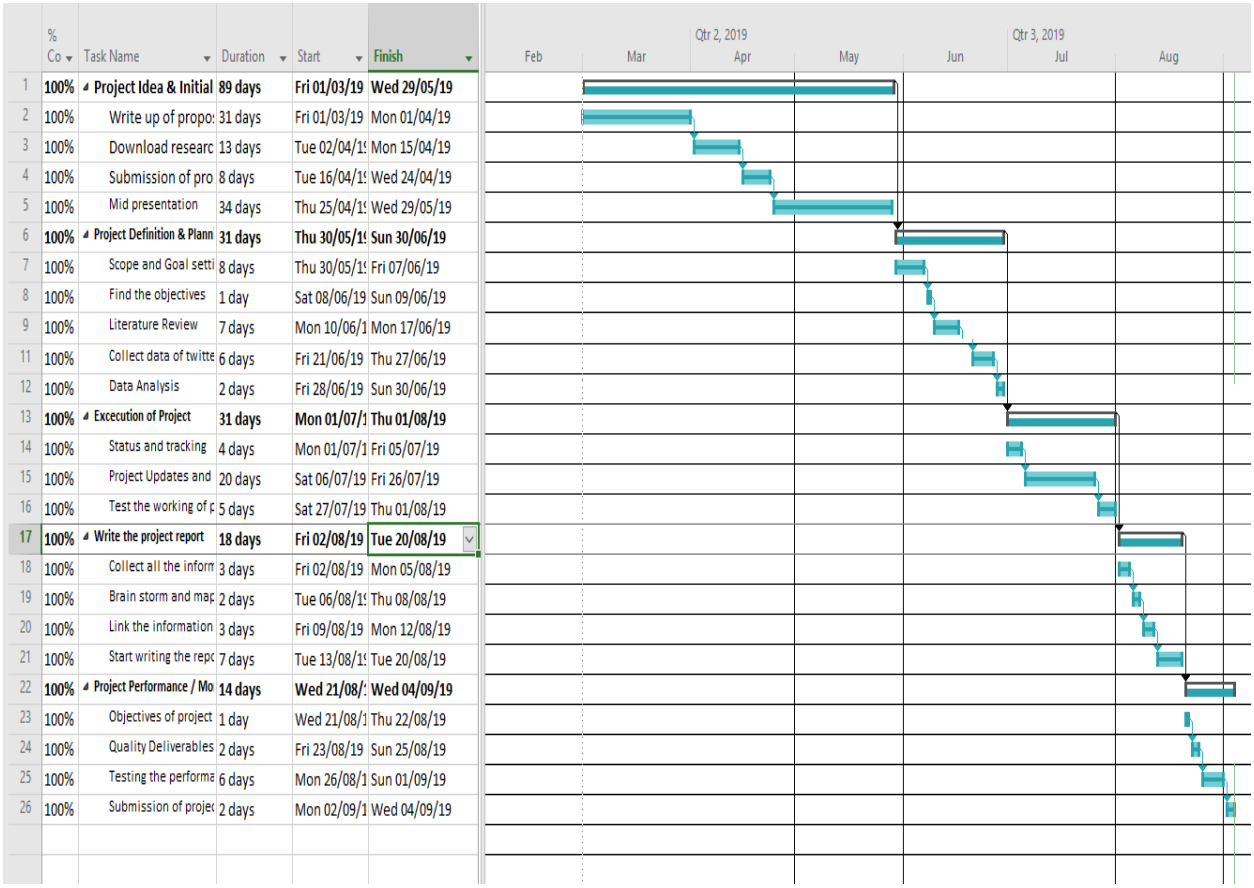
Figure 13 Percentages of Tweets w.r.t. sentiments

7. PROJECT MANAGEMENT

The entire project is completed by following a proper project plan which is shown below. In the beginning phases of project first of all the researcher study a lot about project, stock market and Natural Language Processing for sentiment analysis of Twitter. Then the researcher did literature review related to stock market prediction using sentiment analysis of Tweets. After doing proper research on the project the researcher started implementation of project. Also, the researcher deliver project proposal and presentations on time.

WBS NUMBER	TASK TITLE	START DATE	DUE DATE
1	**Project Idea & Initial Planning**		
1.1	Write up of proposal of project	01.03.2019	01.04.2019
1.2	Download research papers related to project	01.04.2019	08.04.2019
1.3	Submission of proposal	24.04.2019	24.04.2019
1.4	Mid presentation	29.05.2019	29.05.2019
2	**Project Definition & Planning**		
2.1	Scope & Goal Setting	05.06.2019	07.06.2019
2.2	find the objectives	08.06.2019	09.06.2019
2.3	literature review related to the research	10.06.2019	15.06.2019
2.4	research methodology	16.06.2019	18.06.2019
2.5	Collect data of twitter and stocks	23.06.2019	26.06.2019
2.6	Data analysis	27.06.2019	30.06.2019
3	**Execution of Project**		

3.1	Status & Tracking	02.07.2019	05.07.2019
3.2	Project Updates and check	22.07.2019	29.07.2019
3.3	Test the working of project	30.07.2019	01.08.2019
4	**Write the project report**		
4.1	Collect all Information	02.08.2019	5.8.2019
4.2	Brain Storm & Mapping Structure	06.08.2019	8.8.2019
4.3	Link the Information with literature Review	09.08.2019	12.8.2019
4.4	Start Writing the report	13.08.2019	20.8.2019
5	**Project Performance / Monitoring**		
5.1	Objectives of project	21.08.2019	23.08.2019
5.2	Quality Deliverables	24.08.2019	25.08.2019
5.3	Testing the performance of project	26.08.2019	01.09.2019
5.4	Submission of project	02.09.2019	04.09.2019



	% Co	Task Name	Duration	Start	Finish
1	100%	**Project Idea & Initial**	**89 days**	**Fri 01/03/19**	**Wed 29/05/19**
2	100%	Write up of propo:	31 days	Fri 01/03/19	Mon 01/04/19
3	100%	Download researc	13 days	Tue 02/04/19	Mon 15/04/19
4	100%	Submission of pro	8 days	Tue 16/04/19	Wed 24/04/19
5	100%	Mid presentation	34 days	Thu 25/04/19	Wed 29/05/19
6	100%	**Project Definition & Plann**	**31 days**	**Thu 30/05/19**	**Sun 30/06/19**
7	100%	Scope and Goal setti	8 days	Thu 30/05/19	Fri 07/06/19
8	100%	Find the objectives	1 day	Sat 08/06/19	Sun 09/06/19
9	100%	Literature Review	7 days	Mon 10/06/1	Mon 17/06/19
11	100%	Collect data of twitte	6 days	Fri 21/06/19	Thu 27/06/19
12	100%	Data Analysis	2 days	Fri 28/06/19	Sun 30/06/19
13	100%	**Excecution of Project**	**31 days**	**Mon 01/07/1**	**Thu 01/08/19**
14	100%	Status and tracking	4 days	Mon 01/07/1	Fri 05/07/19
15	100%	Project Updates and	20 days	Sat 06/07/19	Fri 26/07/19
16	100%	Test the working of p	5 days	Sat 27/07/19	Thu 01/08/19
17	100%	**Write the project report**	**18 days**	**Fri 02/08/19**	**Tue 20/08/19**
18	100%	Collect all the inform	3 days	Fri 02/08/19	Mon 05/08/19
19	100%	Brain storm and map	2 days	Tue 06/08/19	Thu 08/08/19
20	100%	Link the information	3 days	Fri 09/08/19	Mon 12/08/19
21	100%	Start writing the repo	7 days	Tue 13/08/19	Tue 20/08/19
22	100%	**Project Performance / Mo**	**14 days**	**Wed 21/08/**	**Wed 04/09/19**
23	100%	Objectives of project	1 day	Wed 21/08/1	Thu 22/08/19
24	100%	Quality Deliverables	2 days	Fri 23/08/19	Sun 25/08/19
25	100%	Testing the performa	6 days	Mon 26/08/1	Sun 01/09/19
26	100%	Submission of projec	2 days	Mon 02/09/1	Wed 04/09/19

8. CONCLUSION

The proposed solution is an excellent guidance to investors of all kinds. It is a fully functional, intelligent system that predicts stock market of "Apple" in the US Stock Market with an excellent accuracy. The proposed solution uses hybrid approach to predict stock market. Hybrid approach means to combine two different approaches. The proposed system combines sentiment analysis of Tweets as an extra feature along with historical or numerical data of the company. Five features which include followers (followers of a tweet), polarity (polarity of a tweet), sentiment confidence, clusters (clusters of tweets according to polarity and sentiment confidence) and difference (difference of open and close price of stock) are given as input to Machine Learning models SVR and LSTM.

The main challenge in the project is collecting tweets from social network website Twitter for this the researcher first used Twitter API's, but the problem with Twitter API's are that they give a very smaller number of tweets for free. After that the researcher changed methodology and the researcher scrapped twitter webpages in order to gather more data, when he researcher get enough data the researcher preprocess that data, remove time stamps, and remove stop words form the text and feed the cleaned text into SVM and Naïve Bayes.

8.1. Challenges

Predicting the stock market in itself is one of the biggest mysteries of this world. The first and foremost challenge was the project itself because it was a culmination of different modules that were huge challenges in themselves.

In light of that, the biggest challenges we faced include:

- **Coming-up with an accurate prediction model**: At first, predictions from the LSTM were not that good due to the fact that that the subsequent values were different to the ones on which the LSTM was previously trained on. To counter that, the researcher came up with retraining the LSTM on daily basis so that it becomes acquainted with what's happening currently so that it can predict better.
- **Building prediction algorithm**: Building prediction algorithm was also a challenging task primarily due to the fact that the stock market was an unknown quantity for the researcher.
- **Learning new technologies**: Another mighty challenge the researcher came across was learning different Machine Learning related technologies.

8.2. Recommendations

Ordinary people may use Stock markets to make fortune while financial analysts may use them to determine a country's economy and analyze the growth pattern and impact of government policies and economic situation prevailing in the country. With the use of sentimental analysis along with web scrapping more accurate results on stock prediction can be made. Using Neural Networks and artificial intelligence embedded into such a system it will be possible for us to implement automatic training and analysis of data. A self-learning system can be built which trains itself from the data obtained and use that for further prediction. Once human behavior patterns are formulated this data could be used for finding out the best possible time to make investments and this data could be also used by companies for taking financial decisions that could make an impact on companies' profit margins. A more personalized system can also be made which helps them to make right investments with the funds available and then predicting the rights stocks for investment. Such a system could also predict financial crisis that

could occur. Creating such a system will require huge processing and computational power since huge chunks of data needs to be processed and evaluated. With the use of AI and Neural Nets, occurrence of a particular scenario is understood and conditions that causes it are also taken into account so that when such conditions occur in future it could trace back and warn the user about the situation that could occur. As the world, today is moving towards more automated artificial intelligence-based technologies, such a system implanted in the financial field could be a great boost to the economy.

The researcher has scheduled numerous improvements to the project as the researcher aim to turn it into a fully functioning product. These are as follows:

- Collect news data from different news channels and combine news sentiments with our existing sentiments to have more authentic sentiments of the current day as well as previous days.
- Generate data from tweets for US Stock market and we will combine it the historical data of US Stock market in order to predict US stocks.
- Using a multivariate LSTM to predict future stock prices by incorporating more factors that influence the stock market. However, since not all factors can be quantified, this is a big challenge in itself.
- At the moment, proposed system predicts stock prices for one stock namely APPLE. The researcher aim to predict future stock prices for all stocks listed in the US stock market.
- Using a multi-node HADOOP cluster to store data in a fault-tolerant manner and provide even more scalability to the project during its post-product life cycle.

- Deploying our backend on a powerful Python supported server for even quicker data processing and results calculation.

8.3. Critical Appraisal

When the researcher first set out to do this project, it was hard for the researcher to fathom how we would be able to get high prediction accuracy, particularly for daily closing prediction. This is highly attributed to the fact that the stock market shows irregular patterns, something that even outmuscles Artificial Intelligence many a times. Time series analysis was an impossible attempt to predict expected future stock prices because of the irregular nature of the data points we collected and not having many features to work with further added to the uncertainty of making the project a success. However, through astute decision making and a good system architecture design, the researcher was able to pull-off excellent results, particularly for our hourly price prediction. The researcher can further bring accuracy to predictions through reinforcement learning or a multivariate LSTM while the daily closing predictions can also be improved by further research.

8.4. Student Reflection

In this research work we have managed the proposed solution to show how the stock market prediction can be made using sentimental analysis can be carried for efficient prediction of stocks. The first step starts off with scrapping the tweets related to Apple products. The next steps involved are preprocessing and cleaning the tweets scrapped. The third step carried out is classifying the tweets as positive, negative and neutral. In the next steps, the data is fed into models, which are in turn trained by Naïve Bayes and SVM algorithm. These algorithms are then

improved by changing or altering the parameters according in order to increase the accuracy. The next step includes the twitter sentiment analysis is then merged with the historical data to predict the stock market.

This project “Predictive Analysis of Stock Market using Sentiment Analysis of Twitter” aims to be a standout and accurate guidance for all kinds of investors in the country. The stock market is highly affected by political situation within a country. This is the reason the researcher proposed methodology that consists of sentiment analysis to analyze political situation within a country. The researcher also has managed to learn technically python coding, the algorithms for model training such as Naïve Bayes and SVM algorithm, the modules and packages used for web scrapping, preprocessing of tweets, classifying and labeling of tweets, prediction of stock market by comparing it with historical stock data.

9. REFERENCES

Bollen, J., Mao, H., and Zeng, X. (2011) 'Twitter Mood Predicts the Stock Market'. *Journal of Computational Science* 2 (1), 1–8

Brown, E.D. (2012) 'Will Twitter Make You a Better Investor? A Look at Sentiment, User Reputation and Their Effect on the Stock Market'. *Proc. of SAIS,* 36–42

Dash, R. and Dash, P.K. (2016) 'A Hybrid Stock Trading Framework Integrating Technical Analysis with Machine Learning Techniques'. *The Journal of Finance and Data Science* 2 (1), 42–57

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017) 'Deep Direct Reinforcement Learning for Financial Signal Representation and Trading'. *IEEE Transactions on Neural Networks and Learning Systems* 28 (3), 653–664

Goel, A. and Mittal, A. (2012) *Stock Prediction Using Twitter Sentiment Analysis. Standford University, CS229*. [online] available from <http://cs229.>

Hoepfl, M.C. (1997) 'Choosing Qualitative Research: A Primer for Technology Education Researchers'. *Journal of Technology Education* 9 (1)

Huarng, K.-H., Rey-Mart[illegible], A., and Miquel-Romero, M.-J. (2018) 'Quantitative and Qualitative Comparative Analysis in Business'. *Journal of Business Research* 89, 171–174

'Ijsetr.Org' (2019) in *Ijsetr.Org* [online] available from <http://ijsetr.org/wp-content/uploads/2018/09/IJSETR-VOL-7-ISSUE-9-664-668.pdf>

Kamran, R. (2019) *Prediction of Stock Market Performance by Using Machine Learning Techniques*.

Karim, S., Abdullah, T., and Tayaba, U. (2018) 'Predicting Stock Market Trend from Twitter

Feed and Building a Framework for Bangladesh'. *Doctoral Dissertation, BRAC University*

L.Lima, Milson, P. Nascimento, T. (n.d.) 'Using Sentiment Analysis for Stock Exchange Prediction'. *International Journal of Artificial Intelligence & Applications* 7 (1), 59–67

M, M., S, S., and W, L. (2009) *Stock Prediction Using Twitter Sentiment Analysis*. 1

Porshnev, A., Redkin, I., and Shevchenko, A. (2013) 'Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Twitter Sentiment Analysis'. . . *In 2013 IEkEE 13th International Conference on Data Mining Workshops (Pp. 440-444). IEEE.*

Seghal, V. and Song, C. (2007) 'SOPS: Stock Prediction Using Web Sentiment'. *Proc. - IEEE Int. Conf. Data Mining, ICDM* 21–26

Shah, V.H. (2007) 'Machine Learning Techniques for Stock Prediction'. *Foundations of Machine Learning| Spring, 1(1),* 6–12

Shams, N.Z. and Muhammed, Z. (2005) 'Stock Price Prediction Using Artificial Neural Networks: Case Study'. *Journal of Independent Studies and Research (JISR)* 3 (2)

Smailovic, J., Grcar, M., Lavrac, N., and ﷺnidar ﷺic, M. (2014) 'Predictive Sentiment Analysis of Tweets: A Stock Market Application. In International Workshop on Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data (Pp. 77-88). Springer, Berlin, Heidelberg.' *Information Sciences*

Uc, X. and Yue, S. (2019) *Stock Price Forecasting Using Information from Yahoo Finance and Google Trend*.

V.S, P., K.N.R, C., G, P., and B, M. (2016) 'Sentiment Analysis of Twitter Data for Predicting Stock Market Movements'. *Scopes* 6

Wenger, Z. (1991) 'Qualitative Evaluation and Research Methods (2nd Ed.). By Michael Quinn

Patton. 532 Pp. Newbury Park, CA'. *Qualitative Evaluation and Research Methods (2nd Ed.). By Michael Quinn Patton. 532 Pp. Newbury Park, CA*

Appendix A: Full Script for extracting tweets from Twitter

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import csv

browser = webdriver.Chrome()
# write url here
browser.get("https://twitter.com/search?f=news&vertical=default&q=%23appl&src=typd")
#Run this loop as many times as you want to load the page
elm = browser.find_element_by_tag_name("html")
for x in range (3000):
    print(x)
    elm.send_keys(Keys.END)
    time.sleep(10)
    elm.send_keys(Keys.HOME)
time.sleep(10)
```

```
# scrape the content by CSS SELECTOR
tweet_element = browser.find_elements(By.CSS_SELECTOR,'p[class="TweetTextSize  js-
tweet-text tweet-text"]')
date_element = browser.find_elements(By.CSS_SELECTOR,'a[class="tweet-timestamp js-
permalink js-nav js-tooltip"]')

# writing data to csv file
with open('Tweets_data_2.csv', 'w', newline='', encoding='utf-8') as csvfile:
    autowriter = csv.writer(csvfile)
    autowriter.writerow(['Time','Tweets'])
    for i in range(0,len(tweet_element)-1):
        tweet_text = tweet_element[i].text.encode("utf-8")
        autowriter.writerow([date_element[i].get_attribute("title"), tweet_text])
```

Appendix B: Full Script and Pseudocode for Cleaning/ Pre-processing of Tweets

```
import re
import pandas as pd
import numpy as np
from nltk.stem.porter import *
stemmer = PorterStemmer()
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
from nltk.corpus import stopwords
# df=pd.read_csv('data_dup.csv')
df=pd.read_csv('appletweets.csv')
df=df.dropna(axis=0,how='any')
stop_words = stopwords.words('english')

class Twitter():
    def clean_tweet(self):
        def remove_pattern(input, pattern):
            r = re.findall(pattern, input)
            for i in r:
                input = re.sub(i, '', input)
            return input
        # remove twitter handles (@user)
        df['cleaned_tweet'] = np.vectorize(remove_pattern)(df['Tweets'], "@[\w]*")
        # remove special characters
        df['cleaned_tweet'] = df['cleaned_tweet'].str.replace("[^a-zA-Z#]", " ")
        # remove words less than length 3
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: ' '.join([word for word in i.split() if len(word) >
3]))
        # remove URLs
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: re.split('https:\/\/.*', str(i))[0])
        df['cleaned_tweet'] = df['cleaned_tweet'].replace(r'[^A-Za-z0-9 ]+', '', regex=True)
        # remove numbers
        df['cleaned_tweet'] = df['cleaned_tweet'].str.replace(r'\d+', '')
        # remove special character hashtag"#"
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: i.replace('#', ' '))
        # convert all uppercase letters to lowercase
        df['cleaned_tweet'] = df['cleaned_tweet'].apply(lambda i: i.lower())
```

```
# Tokenization
        tokens = df['cleaned_tweet'].apply(lambda x: x.split())
        tokens.head()
        tokens = tokens.apply(lambda y: [stemmer.stem(i) for i in y])  # stemming
        # df['tokens']=tokenized_tweet
        # df.to_csv('Tokens.csv')
        for i in range(len(tokens)):
            tokens[i] = ' '.join(tokens[i])
        df['cleaned_tweet'] = tokens
        cleaned_tweets = tokens
        # writing cleaned tweets to .csv file
        df.to_csv('cleaning2.csv')
        return cleaned_tweets

if __name__ == '__main__':
   tw=Twitter()
   tw.clean_tweet()
```

Pseudocode for Cleaning the tweets

```
Twitter Class
        Function Clean Tweets
                Remove twitter handles "@"
                Remove special characters
                Remove URLs
                Remove numerics
                Conversion of uppercase letters to lowercase
                Make tokens of text
                Perform stemming on tokens
                return cleaned tweets
        End function

End Twitter Class

Main function
        Create object of Twitter Class
        Call Clean tweets function
```

Appendix C: Full Script and Pseudocode for Labeling Tweets

```
from textblob import TextBlob
import pandas as pd
import csv
df=pd.read_csv('cleaning2.csv')
cleaned_tweets=df['cleaned_tweet']
class Labelling():
    def get_sentiment(self,cleaned_tweets):

        # sentiment analysis of tweets using TextBlob
        analysis = TextBlob(cleaned_tweets)

        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'

    def get_tweets(self):

        #empty list to store tweets
        list_of_tweets = []

        # iterating through tweets
        for tweet in cleaned_tweets:

            # empty dictionary to store tweets' text and sentiment
            tweet_dic = {}

            # storing text part of tweet
            tweet_dic['Tweets'] = tweet
            # storing sentiment of tweet
            tweet_dic['Sentiment'] = self.get_sentiment(tweet)

            # appending parsed tweet to tweets list


            if tweet_dic not in list_of_tweets:
                list_of_tweets.append(tweet_dic)
                # return list of tweets
```

Pseudocode for Labelling the tweets

```
Create Labelling Class
        Function Get Sentiment of Tweets
                Sentiment Analysis of Tweets using TextBlob
                If analysis.sentiment.polarity > 0 Then
                        return 'positive'
                 elif analysis.sentiment.polarity == 0 Then
                        return 'neutral'
                else Then
                        return 'negative'
         End Function
        Function Get Tweets
                Create empty list to store Tweets
                Iterate through Tweets using For loop
                Create an empty dictionary to store text and sentiments of tweets
                Store text of tweet and its sentiment in a dictionary
                Append all the tweets in a list
        Return list of all tweets
        End Function
  End Class
Main Function
  Create object of Labelling Class
  Call Get Tweets function
  Select positive tweets from all tweets
  Calculate percentage of positive tweets
  Select negative tweets from all tweets

  Select neutral tweets from all tweets
  Calculate percentage of neutral tweets
  Write all tweets with their sentiments in a csv file
```

```
  Select negative tweets from all tweets
  Calculate percentage of negative tweets
  Select neutral tweets from all tweets
  Calculate percentage of neutral tweets
  Write all tweets with their sentiments in a csv file
End Main Function
```

Appendix D: Full Script and Pseudocode for Model Training Using SVM

```
import pandas as pd
from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from nltk.stem.porter import *
stemmer = PorterStemmer()
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")
df1=pd.read_csv('final_train.csv')
df2=pd.read_csv('final_test.csv')
df1=df1.dropna(axis=0,how='any')
df2=df2.dropna(axis=0,how='any')
df=pd.read_csv('output_data.csv')
bow= CountVectorizer(max_df=0.90, min_df=2, max_features=3000,
stop_words='english')
# Extracting features using bag of words approach
bag_of_words = bow.fit_transform(df['Tweets'])
print(bag_of_words)
ngram= CountVectorizer(binary=True, ngram_range=(1, 2))
ngram.fit(df['Tweets'])
X = ngram.transform(df['Tweets'])
X_test = ngram.transform(df2['Tweets'])
output=df['Sentiment']
X_train, X_val, y_train, y_val = train_test_split(
    X, output, train_size=0.75
)
```

```
c=1.0    #85.76
svm = LinearSVC(C=c)
svm.fit(X_train, y_train)
print("Accuracy with SVM(ngrams)  for C=%s: %s"
      % (c, accuracy_score(y_val, svm.predict(X_val))))

tfidf_vectorizer = TfidfVectorizer()
tfidf_vectorizer.fit(df['Tweets'])
X = tfidf_vectorizer.transform(df['Tweets'])
X_test = tfidf_vectorizer.transform(df2['Tweets'])
target=df['Sentiment']
X_train, X_val, y_train, y_val = train_test_split(
    X, target, train_size=0.75
)
```

Pseudocode for Model Training Using SVM

```
Reading dataset from csv file
Create Bag of Words Feature Matrix
Extract features using N-grams approach
Set inputs and output for SVM model
Split dataset into training and test datasets
Set regularization parameter
Train SVM model
Extract features using TF-IDF approach
Set inputs and output for SVM model
Split dataset into training and test datasets
Set regularization parameter
Train SVM model
```

Appendix E: Full Script and Pseudocode for Model Training Using Naïve Bayes

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from nltk.stem.porter import *
stemmer = PorterStemmer()
import sys
if not sys.warnoptions:
   import warnings
   warnings.simplefilter("ignore")

df=pd.read_csv('ready_data.csv')
ngram= CountVectorizer(binary=True, ngram_range=(2, 3))
ngram.fit(df['Tweets'])
X = ngram.transform(df['Tweets'])
output=df['Sentiment']
X_train, X_test, y_train, y_test = train_test_split(
   X, output, train_size=0.75
)
model_NB = MultinomialNB().fit(X_train, y_train)
predictions = model_NB.predict(X_test)
print("Accuracy with Naive Bayes (N-grams):  %s"
      % ( accuracy_score(y_test, predictions)))

vect = CountVectorizer()
vect_count = vect.fit_transform(df['Tweets'])
tf_idf = TfidfTransformer().fit(vect_count)
vect_count = tf_idf.transform(vect_count)
print(vect_count)
X_train, X_test, y_train, y_test = train_test_split(vect_count, df['Sentiment'],
test_size=0.25)
model_NB = MultinomialNB().fit(X_train, y_train)
predictions = model_NB.predict(X_test)
print("Accuracy with Naive Bayes (Tfidf):  %s"
      % ( accuracy_score(y_test, predictions)))
```

Pseudocode for Model Training Using Naïve Bayes

```
Reading dataset from csv file
Extract features using N-grams approach
Set inputs and output for Naive Bayes model
Split dataset into training and test datasets
Train Naive Bayes model

Extract features using TF-IDF approach
Set inputs and output for Naive Bayes model
Split dataset into training and test datasets
Train Naive Bayes model

Extract features using word counts approach
Set inputs and output for Naive Bayes model
Split dataset into training and test datasets
Train Naive Bayes model
```

Appendix F: Full Script and Pseudocode for Stock Prediction using Support Vector Regression

```
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
import glob
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")

files = glob.glob("Final_Data/*.csv")
frame = pd.DataFrame()
list_of_files = []
for file in files:
    df = pd.read_csv(file,index_col=0,encoding='latin-1')
    df = df.drop('text', 1)
    df = df.drop('date', 1)
    list_of_files.append(df)
frame_of_files = pd.concat(list_of_files)
frame_of_files.dropna(axis=1,how='any',inplace=True)
inputs = frame_of_files.drop('difference',1)
X= np.array(inputs)
Y= np.array(frame_of_files['difference'].values)
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=0)
svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)
y_rbf = svr_rbf.fit(X_train, y_train).predict(X_test)
print('Root Mean Squared Error for SVR(rbf) :', np.sqrt(metrics.mean_squared_error(y_test,
y_rbf)))
```

Pseudocode for Stock Prediction using Support Vector Regression

```
Reading dataset from csv files
Drop column 'text'
Drop column 'date'
Take all columns of csv file as inputs except 'difference'
X=array of inputs
Y=array of 'difference' values
Split dataset into training and test dataset
Train SVR model using kernel 'rbf'
```

Appendix G: Full Script and Pseudocode for Stock Prediction using LSTM

```
from numpy import array
from numpy import hstack
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
import pandas as pd
from sklearn import metrics
import numpy as np
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")
df_inputs=pd.read_csv('inputs.csv')
df_outputs=pd.read_csv('output.csv')
df = df_inputs.drop('text', 1)
df = df_inputs.drop('date', 1)
foll=[i for i in df['followers']]
pol=[i for i in df['polarity']]
senti=[i for i in df['sentiment_confidence']]
clus=[i for i in df['clusters']]
output=[i for i in df_outputs['difference']]

# Splitting of multivariate sequence data into number of samples
def split_sequences(seq, no_of_steps):
  X, y = list(), list()
  length_seq=len(seq)
  for z in range(length_seq):
    # find the end of this pattern
    end_data = z + no_of_steps
    # checking if we are not out of the bounds of dataset
    if end_data > len(seq):
      break

    # gather input and output parts of the pattern
    sequence_x, sequence_y = seq[z:end_data, :-1], seq[end_data-1, -1]
    X.append(sequence_x)
    y.append(sequence_y)
  return array(X), array(y)
```

```
# defining inputs for model
input_sequence1= array(foll[0:-20])
input_sequence2= array(pol[0:-20])
input_sequence3= array(senti[0:-20])
input_sequence4= array(clus[0:-20])
output_sequence= array(output[0:-20])
# convert to [rows, columns] structure
input_sequence1 = input_sequence1.reshape((len(input_sequence1), 1))
input_sequence2 = input_sequence2.reshape((len(input_sequence2), 1))
input_sequence3 = input_sequence3.reshape((len(input_sequence3), 1))
input_sequence4 = input_sequence4.reshape((len(input_sequence4), 1))
output_sequence = output_sequence.reshape((len(output_sequence), 1))
```

```
# Stacking columns horizonatlly
data = hstack((input_sequence1, input_sequence2, input_sequence3, input_sequence4,
output_sequence))

# setting no. of time steps
no_of_steps = 1

# converting sequences into input and output
X, y = split_sequences(data, no_of_steps)
print(X)
print(X.shape[2])

# the dataset knows the number of features
no_of_features = X.shape[2]

# defining model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(no_of_steps, no_of_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

# fitting model
model.fit(X, y, epochs=100, verbose=1)
```

```
# demonstration of prediction
yhat=[]
output=([[foll[-20+x], pol[-20+x], senti[-20+x], clus[-20+x]] for x in range(0,20)])
for i in output:
    input = array(i)
    input = input.reshape((1, no_of_steps, no_of_features))
    yhat.append((model.predict(input, verbose=0)))
results=[list(i)[0][0] for i in yhat]
orig=output[-20:]
print(input)
print(results)
for i in range(len(orig)):
    print(100*((orig[i]-results[i])/orig[i]))

print('Root Mean Squared Error for LSTM :', np.sqrt(metrics.mean_squared_error(input,
results)))
```

Pseudocode for Stock prediction using LSTM

Reading dataset from csv file
Set input parameters for LSTM model
Split multivariate sequence into number of samples
Define inputs for model
Set number of time steps for LSTM model
Convert sequence into input and output
Define LSTM model
Train LSTM model

Appendix H: Full Script and Pseudocode for Merging Twitter and Stock Data

```
import pandas as pd
df = pd.read_csv("Cluster_Data/"+'apple'+"cluster.csv",index_col=0,encoding='latin-1')
def get_stock_value(diff, foll, total_foll, polar,senti, curr_price):
    stock_value = diff*(foll/total_foll)*polar*senti/(curr_price)*10000
    print(stock_value)
    return stock_value
apple=pd.read_csv('AAPL(2).csv')
current_price = apple['Open'][0]
difference = abs(apple['Open'][0] - apple['Close'][0])
total_followers = df['followers'].sum()
for i,row in df.iterrows():
    foll = df['followers'][i]
    pol = df['polarity'][i]
    confi = df['sentiment_confidence'][i]
    df.at[i,'difference'] = get_stock_value(difference, foll, total_followers, pol, confi,
current_price)
df.to_csv('Final_Data/'+'apple_stocks'+'.csv')
print(df.head())
```

Pseudocode for Merging Twitter and Stock Data

```
Read cluster dataset from csv file
Function Get Stock Value
        Value_of_stock=Difference*(followers/Total_followers)*polarity*sentiment_confide
nce/
        (current_price)*10000
        return Value_of_stock
Read Historical data of stocks
Current_Price=Open Price
Difference=Open Price - Close Price
Total_followers=Sum of column 'followers'
Iterate throught stocks dataset using 'FOR' loop
        Take inputs from dataset
        Call function 'Get Stock Value'
        Store returned values from function in dataframe
Write dataframe to a separate csv file
```