SMART INVESTMENTS AND FINANCIAL TRADING SYSTEMS

A project report submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the

Award of the Degree

of

Bachelor of Technology

in

Computer and Communication Engineering

by

S.Sai Aravind — **Reg. No. 150953188**

Nagam Lahari Teja — **Reg. No. 150953218**

Under the guidance of

Mr. Arjun CV
Assistant Professor
Department of I & CT
Manipal Institute of Technology
Manipal, India

JULY 2020

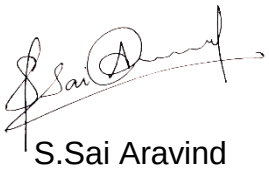
I dedicate my thesis to my friends and family.

DECLARATION

I hereby declare that this project work entitled **Smart Investments and Financial Trading Systems** is original and has been carried out by me in the Department of Information and Communication Technology of Manipal Institute of Technology, Manipal, under the guidance of **Mr.Arjun CV**, Assistant Professor, Department of Information and Communication Technology, M. I. T., Manipal. No part of this work has been submitted for the award of a degree or diploma either to this University or to any other Universities.

Place: Manipal

Date : 15/07/2020



S.Sai Aravind

Nagam Lahari Teja

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

CERTIFICATE

This is to certify that this project entitled **Smart Investments and Financial Trading Systems** is a bonafide project work done by **Mr.S.Sai Aravind** (Reg.No.150953188) and **Mr.Nagam Lahari Teja** (Reg.No.150953218) at Manipal Institute of Technology, Manipal, independently under my guidance and supervision for the award of the Degree of Bachelor of Technology in Computers and Communication Engineering.

Mr.Arjun CV
Assistant Professor
Department of I & CT
Manipal Institute of Technology
Manipal, India

Dr.Balachandra
Professor & Head
Department of I & CT
Manipal Institute of Technology
Manipal, India

ACKNOWLEDGEMENTS

We express profound gratitude to our Project Guide **Mr.Arjun CV**, Assistant Professor, Department of Information and Communication Technology, for his guidance has been of high esteem, and the faith that he has shown in us during the Project work has been the cradle of illumination for us. His technical expertise and approach of tackling problems have been of great help to me. Besides, his encouragement, personal guidance and understanding helped me to carry out the project with great zeal and enthusiasm.

We would also like to thank and express our gratitude to **Dr.Srikanth Rao**, Director, Manipal Institute Of Technology, Manipal, **Dr.Balachandra**, Head of the Department of Information and Communication Technology, **Dr.Sanjay Singh**, Project Coordinator - Practice School (CCE), Department of Information & Communication Technology, for giving us such an opportunity, support, kind cooperation and guidance regarding the process of carry out the project work to enrich the knowledge and development of skills in such a premier and reputed institute.

We would also like to thank all our department faculty members for their valuable suggestion and support, during the course of study and in our current project work.

We warmly thank our parents for their never ending support and all our classmates and friends at Manipal Academy of Higher Education, who have helped us to be what we are today.

ABSTRACT

A Stock Market is an aggregation or a collection of people or entities who participate in buying, selling and issuing of stocks, which represents shares in a company. A stock exchange is a place where individuals and organizations can trade stocks.

This is a great area of interest for traders as one can generate profit by buying or selling stocks of a company, at the right moment. Determining the right price and the right time for trading a stock is a challenging problem for traders. Techniques like fundamental analysis, technical analysis, predicting stock prices have been used and they are effective than randomly buying or selling.

Trading in a stock market basically involves the buying, selling or holding of stocks, corporate bonds, equities and a few other securities. These transactions are decided and based upon a lot of factors which we are going to discuss in this project.

The project aims to deal with providing a smart decision system for buying and selling of stocks based on technical indicators and oscillators and sentiments of the company that it is associated with. We also try to predict the price and other technical indicators of the stock for a future time period.

Investors look for trends, fundamental data, finance of company, history of the company, technical charts etc. and then decide whether to invest in it or not. We are creating a smart investing system that uses various tools, indicators and oscillators and predict the best stock options for the investor.

Sentiment analysis has also been done for providing a boost to the technical analysis. Sentiment analysis can be technically defined as the process which consists of identifying and extracting different textual data computationally, and determining which emotions the people or entities in general displays. This type of analysis is used in different capabilities such as to predict a customer's opinion of a product, customer service and predict the brand value of a company.

We are making an automated system such that it can inform you immediately the best time to buy and sell stocks through emails so that the investor can have more profits on an active livestock market according to the trading strategies chosen by the investor.

[Theory of computation]: Theory and algorithms for application domains — Structured Prediction, Machine Learning Theory;

[Information systems]: Information retrieval — Retrieval tasks and goals, Sentiment Analysis, Business Intelligence;

Contents

List of Tables:

List of Figures:

ABBREVIATIONS

MA	:	Moving Average
LS-SVM	:	Least Square Support Vector Machine
RBF	:	Radial Basis Function
NSE	:	National Stock Exchange
SMA	:	Simple Moving Average
EMA	:	Exponential Moving Average
MACD	:	Moving Average Convergence Divergence
RSI	:	Relative Strength Index
VWMA	:	Volume Weighted Moving Average
MLP	:	Multi-Layer Perceptron
API	:	Application Programming Interface
AWS	:	Amazon Web Services

List of Equations:

Chapter 1

1.1 Introduction and Trading Mechanisms

A Stock Market is an aggregation or a collection of people or entities who participate in buying, selling and issuing of stocks which represents a share in a company. A stock exchange is a place where individuals and organizations can trade stocks.

This is a great area of interest for traders, as one can generate profit by buying or selling stocks of a company at the right moment. Determining the right price and the right time for trading stock is a challenging problem for traders. Techniques like fundamental analysis, technical analysis, predicting stock prices have been used and they are effective than randomly buying or selling.

Trading in a stock market basically involves the buying, selling or holding of stocks, corporate bonds, equities and a few other securities. These transactions are decided and based upon a lot of factors which we are going to discuss in this project.

Problem Statement:

Stock market trading for long has been one of the most important forms of trading carried out by most of the people in India. It is generally carried out by common people who look of a quick source of income, on a daily basis and others who look for a long term investments. The price of the stock is volatile and changes frequently due to a lot of external factors. Most of the cases people invest and trade stocks based on their intuitions. There is a need of system which could provide proper factual support to their intuitions.

Objectives:

- The project aims to deal with providing a smart decision system for making all the mentioned transactions of stocks based on technical indicators and oscillators and sentiments of the company that it is associated with.

- To predict the price and other technical indicators such as SMA, EMA, MACD, RSI and Bollinger Bands of the stock for a future time period.

- To automate the system such that it can inform you immediately the best time to buy and sell stocks through emails so that the investor can have more profits on an active livestock market according to the trading strategies chosen by the investor.

- To analyze the current sentiments of the people regarding the stock and provide a support to the technical analysis report.

The stock exchange or equity market offers chance to investors to extend their financial gain while not putting them in the high risk of stepping into their own businesses with high overheads and startup prices.

Smart Investors look for trends, fundamental data, finance of company, history of the company, technical charts etc. and then decide whether to invest or not. Trading strategies are generally classified into two different categories:

(i) Fundamental analysis
(ii) Technical analysis.

(i) Fundamental Analysis :

The study of financial statements and determining the true potential or value of the stock and then taking decision as to buy or sell, is called fundamental analysis.

(ii) Technical Analysis :

The study of charts and determining the trend based on past stock data to look for patterns and by making a decision to as to sell or buy the stock, is called technical analytics.

There are 3 principles that are followed in technical analysis:

- The investor's perception and understanding of the stock and how it behaves forms the core ideology of the technical analysis.
- Stock Prices follow trends.
- History repeats itself.

Various technical indicators like SMA, EMA, RSI, MACD, Bollinger Bands etc. have been in use for a long time. They are used by technical analysts. This project has also made an attempt of making a smart decision based on indicators, oscillators. An attempt to predict stock price has also made. Statistical Regression is used for determining relationship between index and stock price.

Chapter 2

Literature Survey

As per the reports of "Global Algorithmic Trading Market 2020-2024" published by the research and markets on January 10,2020, the global algorithmic or computational trading market is expected to grow at a CAGR of 11% during the period 2020-2024.Due to the current Covid-19 pandemic, the current Compounded Annual Growth Rate has fallen down to 9%.

Almost 30-35% of the trade done in India are Algorithmic, while the global percentage is around 75%.Some of the already existing trading platforms are Omnesys NEST, ALGOTrader, and FLEXTRADE and there are many online platforms that are dealing with the technical analysis. Technical analysis is a method of making buy and sell decision used extensively by the technical analyst. Technical analysis is based upon assumptions that are against the widely accepted Efficient Market Hypothesis [1].

Artificial Neural Network in one of the technique that is useful in predicting stock prices [2]. They have tried to predict the share/stock price of companies under LIX15 of NSE (National Stock Exchange). The data of LIX15's companies for past 36 months was taken as input (01-01-2011 – 01-01-2014).MLP was implemented and trained and tested over this data. It gave satisfactory results. RBF Neural Network is preferred over MLP Network for stock market prediction [3].

Chapter 3

Software Requirements

The supported Operating Systems are as follows:

- Windows 7, 8, 10(32 or 64 bit)
- Mac OS X 10.5.8
- Linux (Ubuntu Linux)

The Supported Development Environments are as follows:

- Safari Browser
- Google Chrome
- Internet Explorer

Any Integrated Development Environment (IDE) that provides comprehensive facilities to build the code can be used. It should basically contain the facilities of a source code editor, an automated tool builder and a debugger and others that can be used to develop the project.

3.1 Client Side Technologies:

IDE

We have used Anaconda, which is a free and an open-source platform for running different languages like Python and R programming languages for scientific computing and development. It aims to simplify the process by having several package deployments and easy management.

The IDE is suitable for Windows, Linux, and Mac-OS and its distribution includes packages for Data Science as well.

The IDE which we have used is the Jupyter Notebook. It is a Web based application and is also open sources, and a part of the Anaconda platform. It allows the users to to create and share documents and execute live code, visualizations, equations and textual narration.

Python 3.8

Python, today is considered to be one of the most valuable programming languages which is currently being used in a wide variety of applications, purposes and employments. It is an easy to understand and implement language for beginners, considering its understandability and other predefined libraries which pretty much makes it straight forward. Currently all the major interesting projects and applications in the world are developed using python.

Python is an interpreted or a deciphered language, meaning that the code written in Python executes and runs at real time, making it simple to write and test snippets of code and also making the process of moving the code between two stages simpler. Since Python is a general language found in a wide range of utilizations, most of the operating systems support it and make use of it for development of applications.

Pandas

Pandas is a library offered by Python Library which is open sourced, giving elite control over data and it uses its amazing data structures as an examination utility device to examine the snippets of code. “Panel Data” is the source from which the name called Pandas was derived.

Before Pandas was introduced, Python was significantly used for data wrangling, i.e. converting raw data from one form to the other and planning the data. The idea of data investigation was not even given due importance. The introduction of Pandas helped to solve this issue. By using Pandas, we can implement the 5 main goals in the handling and examination of data, i.e. “load, get ready, control, model, and dissect” whilst paying less attention to the source of the data.

Python along with the use of Pandas, has a wide variety of applications in the fields of academic and scholarly articles and different area of business including all financial securities such as money, bonds, analytics and statistics etc.

Some of the main features of Pandas library are:

- Quick and productive Data Frame object with default and tweaked ordering.
- Devices for stacking data into in-memory data objects from various document designs.
- Data arrangement and coordinated treatment of missing data.
- Reshaping and conversion of date sets.
- Superior combining and joining of data.
- Different columns from a data structure can be erased or embedded.
- Time Series functionality.

Numpy

NumPy is a python library package which is open source and is the short name for Numerical Python. It consists of multidimensional array objects and an accumulation of schedules for handling of the arrays.

Jim Hugunin was the creator of Numpy. He created another library package called numeric as its predecessor. He also created one more bundle called the “Numarray” which consists of some extra functionalities which were missing in both of the previous packages.

Travis Oliphant, in 2005, made NumPy bundle by extracting and implementing all the important features of the Numarray into the Numeric bundle. A lot of people worldwide use the Numpy package for various applications.

A few functionalities of the Numpy package are:

- Arrays have the capability to do numerical and logical tasks.
- Shape control can be done using Fourier transforms and schedules.
- NumPy has the built-in ability for various mathematical functions such as direct polynomial math and arbitrary number creation and different tasks are identified with straight variable based math.

Sckit-learn

We have also used one more python module called Sckit-learn or sklearn. The most exciting feature of this module is that it integrates the most commonly used machine learning algorithms into the world of different scientific Python packages such as numpy, scipy and matplotlib.

The most important feature of sklearn is that it provides an efficient and a simple solution for the learning problems that are accessible to every person and which are reusable in various contexts in the different fields of Machine Learning and as a versatile and a useful tool for science and engineering. Scikit-learn has a consistent interface in python which provides a range of supervised and unsupervised learning and classification algorithms.

The sckit-learn library is built upon the SciPy library which is a prerequisite for the sklearn and other libraries. The SciPy library stack consists of:

- NumPy – It is a base n-dimensional array package.
- SciPy – It is a fundamental library used for scientific computing.
- Matplotlib – It is used for comprehensive 2D/3D plotting.
- IPython: It is an enhanced interactive console for python.
- SymPy: It is used for representing symbolic mathematics.
- Pandas: It is used for data structures and analysis of codes.

Scikit-learn is a module that provides learning algorithms. It is an extension of Scikit. Generally all the extension modules for SciPy care are named SciKits.

The Sckit-learn library was developed on the vision to provide a level of wholesomeness and the support that is required for extended usage in different systems for production and provide the maximum support on different important functions such as the quality of the code, the ease of usage, its documentation and its performance.

Server Side Technologies

AWS Lambda

AWS Lambda is a product of the Amazon Web Services that offers a server less processing framework which helps to run code automatically and consequently deal with the fundamental figuring of assets. An amazing feature of the AWS Lambda is that, we can run code without the provision of a managing server and it also executes the inputted code only when it is triggered or when it is needed. The code is also scaled automatically based on the requests which range from a few requests per day to a few thousand requests per second [4].

AWS Lambda runs the inputted code on a high-availability compute infrastructure which simplifies all the preceding processes and performs all of the required administration of the basic compute resources which included the server and operating system, its maintenance, its capacity provisioning, the code monitoring and logging and the automatic scaling of code as per the incoming requests.

An additional feature of the AWS Lambda is that it allows an individual to naturally run code without any problems and automatically solves the issues that are raised in numerous sorts of occasions, such as, HTTP demands from Amazon API portal, table updates in Amazon Dynamo Database and advances of the state. It additionally empowers the users to reach out to different AWS administrations with custom functionalities, and even make its very own backend administrations.

The administration works by running the code on a high-accessibility PC foundation. It at that point plays out, all the authoritative obligations of that registered asset, as:

- Maintaining upkeep on the server and working of the framework.
- Naturally scaling and dealing with the individual's ability of arrangements.
- Dealing with security fix arrangement
- Code observation
- Logging

SendGrid

SendGrid is an online tool used for marketing and communication. It is generally used to send bulk emails. It generally runs on a cloud-based SMTP server and also tracks and manages all the technical information including the ISP outreach and infrastructure scaling. It also monitors the reputation of the brand and makes real time analytics while also catering to the whitelist services at the same time.

There are two ways of sending an email through SendGrid. One of the method is via an SMTP relay and the other method is through the Web API. SendGrid provides client libraries in many programming languages which is the most preferred and the easiest way to integrate our project with SendGrid. If we choose to use SendGrid without a client library, then the Web API is the most recommended as it is faster and provides some benefit with encoding, and makes it easier for the user to use [5].

Tweepy

Tweepy is a Python library for accessing the Twitter API. It is great for simple automation and creating twitter bots. Tweepy has many features. It is an open-sourced tool, hosted on GitHub and enables Python to communicate with Twitter platform and use its API.

TextBlob

TextBlob is a python library and offers a simple API to access its methods and performs basic Natural Language Processing tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. A good thing about TextBlob is that they are just like python strings.

3.2 Data Collection and Management

Quandl

Quandl is an API that is used to collect millions of financial data and economic datasets of different companies from various publishers into the Python interface.

Quandl offers a simple and an easy to use API for downloading stock market data. Quandl's has a daily data feed feature that that delivers important features like end of the day prices, historically important fundamental stock data including harmonized fundamentals, financial ratios, investor sentiment, indexes, earnings , ratings and whole lot of other options in the financial sector that helps us in the prediction of a few parameters in the technical analysis report.

It collects data from various sources and integrates into one single platform. Quandl also provide API's so that we can make use of that data. We have made use of this API to collect all the stock value and other data of a company

Quandl API was used to fetch the stock quote of any company under NSE. Stock Quote consist of Close Price, High, Low, Volume, Open etc. as a data-frame.

A data-frame can be defined as a two dimensional labelled data structure that stores data of different types in different columns. A data-frame consists of columns and a single column represent a series.

Series of Close price is extracted and is used as data source for every indicator, oscillator and neural network. Date is the index of the series and is also used as data for indicators and oscillators. This series represents the past closing prices of a company under NSE.

API Key was generated: dPyyCmjxE9TZP6Jfdztp

Chapter 4

Back Testing Module

In this module, we deal with technical indicators and trading strategies. We aim to develop a program using which we can apply a combination of Technical Indicators, thereby creating and testing a strategy. This is applied on the different stock price data of the companies we have derived from the Quandl API.

Then the results of back testing are visualized using Matplotlib to gain valuable insights. We also apply a support and resistance analysis on the data that the program executes to predict a value for the stock for the next day.

4.1 Implementation of Indicators and Oscillators

A technical indicator offers an alternate point of view from which to investigate the share price activity. A few, for example, moving averages, are gotten from the basic formulas and are moderately straightforward.

Oscillators are defined as those indicators that are to be used when the viewing charts of the stock are ranging i.e. from trending to non-trending and are used to determine the condition of the stock as an overbought stock or an oversold stock [6].

We will be implementing the following indicators and oscillators:

- SMA - Simple Moving Average.
- EMA - Exponential Moving Average.
- MACD - Moving Average Convergence Divergence.
- RSI - Relative Strength Index.
- Bollinger Bands.
- Statistical Regression.

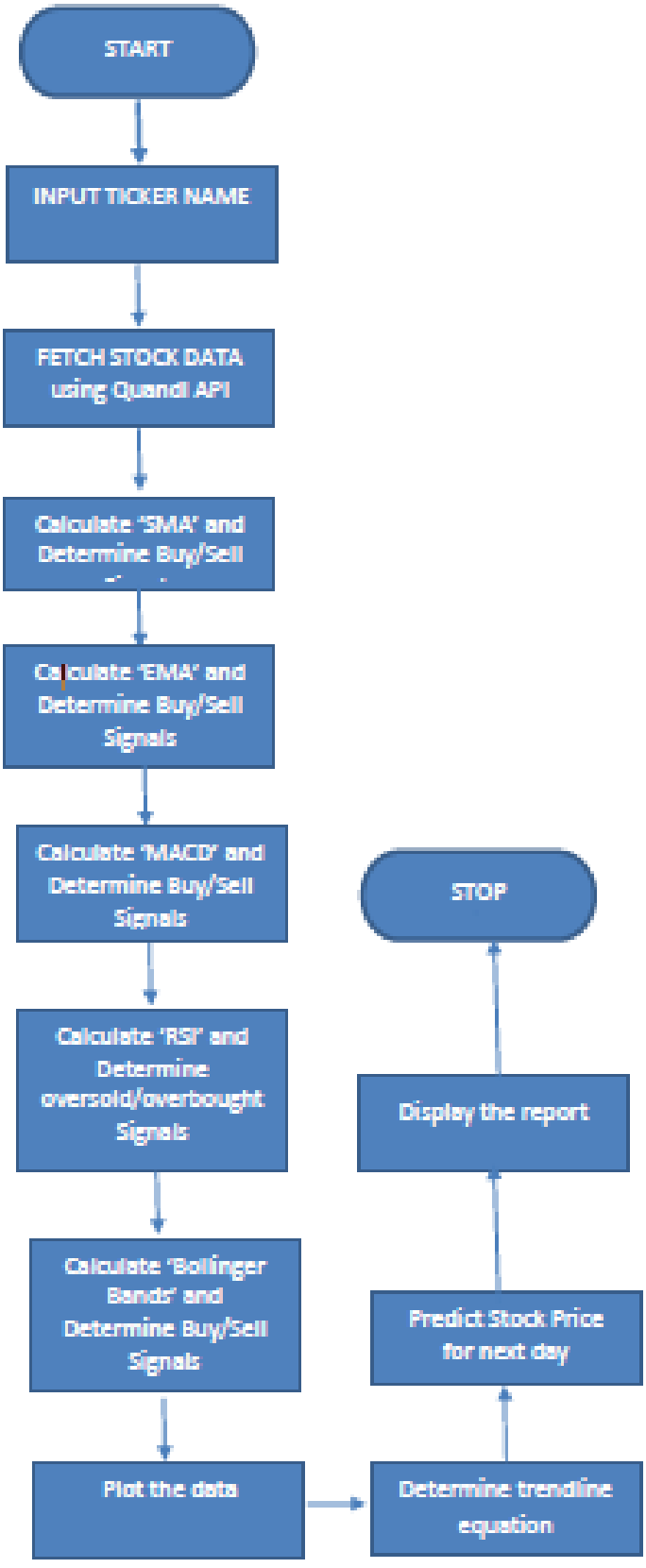


Figure 4.1 – Flowchart of the Project

4.1.1 SMA – Simple Moving Average :

The Simple Moving Average or SMA in sort is an indicator that is calculated by adding the price of a stock or an entity over a few time periods and are then calculated by, dividing the sum by the total number of time periods [7].

The average price of a stock or an entity over a time period, with equal weightage given to the price of the stock in each period is how the Simple Moving Average is calculated.

It is a moving average in which mean is calculated over the price of n-periods. Using it, we get a series of averages of different subsets of a full dataset.

SMA = (pm + pm-1 + pm-2 + …….. + pm-(n-1))/n (Equation 4.1.1)
n = 20 but it can have any value.

Where numerator is the summation of previous n-periods prices and n is the length of period.

The data that has been used should be centered close to the mean as shown in the Figure 4.1.1. Failing to do so, will make the average SMA to lag behind the last entered data point by around half of the width of the input sample.

The old data points that are being dropping out or new data points that are coming in can influence a Simple Moving Average very easily. One main and an important feature of a SMA is that even if the data has a certain periodic fluctuation, applying an SMA of that particular period will eliminate the variation that has been caused.

A perfectly regular cycle will be a rare occurrence.

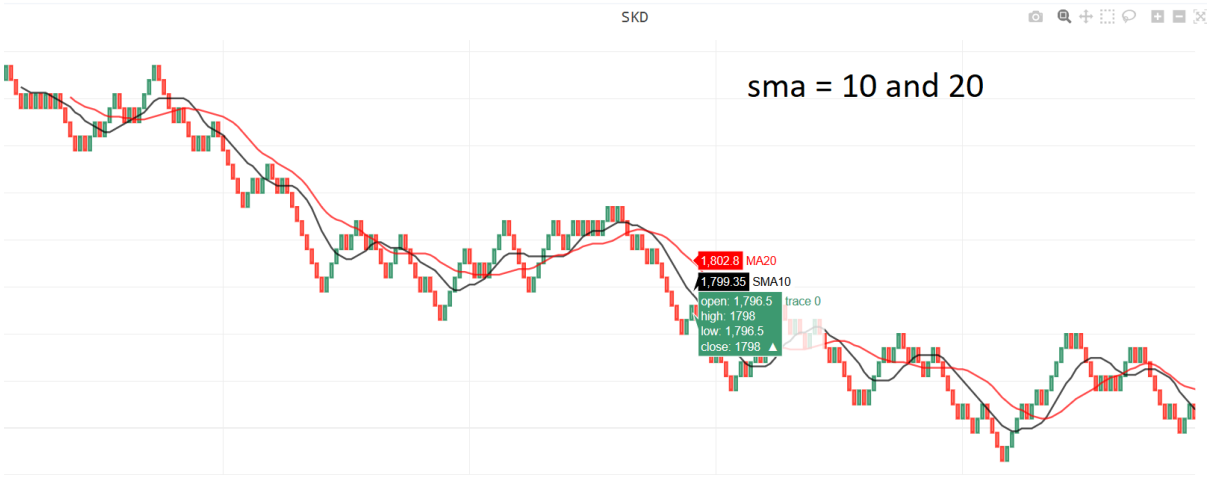


Figure 4.1.1 – SMA Curve

4.1.2 EMA – Exponential Moving Average :

The Exponential Moving Average or simply the EMA is a type of moving average which is almost similar the simple moving average but it also has a few exceptions. In the Exponential Moving Average, typically more weight is assigned to the latest data thus making the EMA react faster to the recent or latest changes in price, as when compared to the SMA [8].

Calculating the EMA involves three main steps as mentioned below:

- Calculation of the Simple Moving Average.
- Calculation of the multiplier.
- Calculation of the current Exponential Moving Average.

Multiplier = 2 / (n+1),
EMA = (Closing Price – EMA (previous day))*multiplier + EMA (previous day))

(Equation 4.1.2)

We make use of the Exponential Moving Average to determine something called the trend direction, and the trade in that trend direction. When the EMA rises, it suggests us to consider buying the stock when prices start to dip or if they fall just below the EMA. In turn, when the EMA falls or goes low, it suggests us to consider selling i.e. when the stock prices go towards positive side or sometimes just above the EMA.

The support and resistance areas of the stock are also calculated with the help of the Moving Averages. A rising EMA tends to support the price action, while a falling EMA tends to provide resistance to price action as shown in the Figure 4.1.2. This brings strength to the strategy to buy the stock when the price is near the rising EMA and to sell the stock when the price is near the falling EMA.

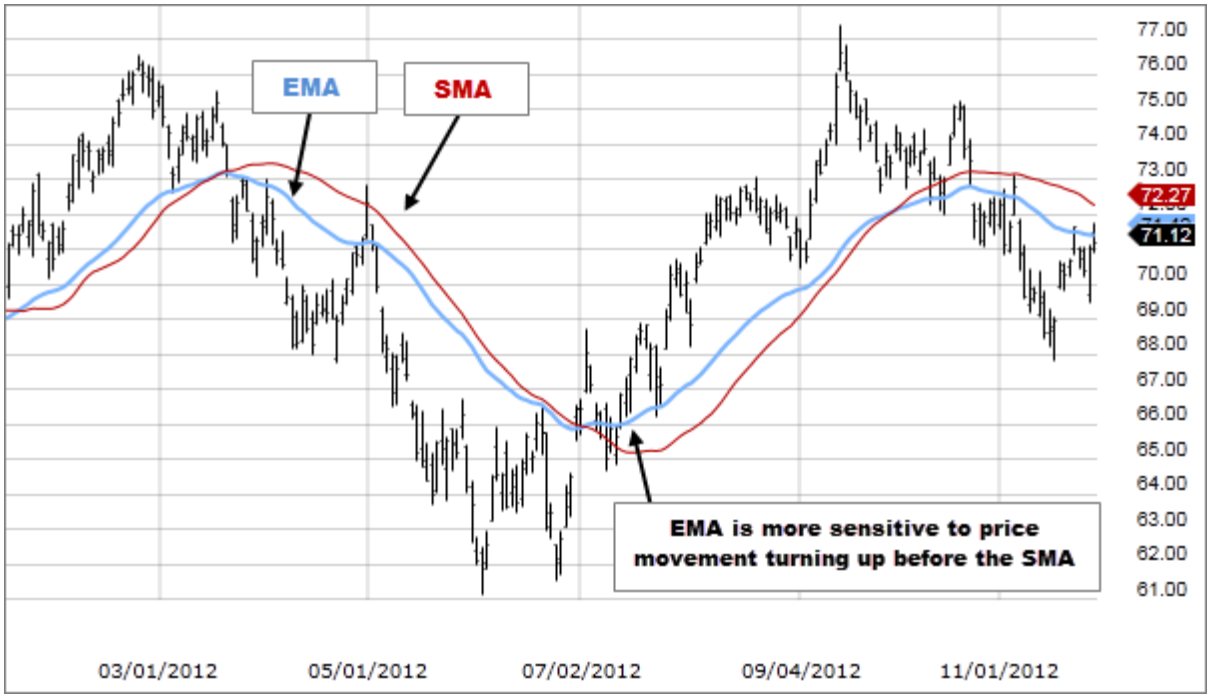


Figure 4.1.2 – EMA Curve

4.1.3 MACD – Moving Average Convergence Divergence :

Gerald Appel was the man behind the development of this indicator called the Moving Average Convergence Divergence and MACD in short. It is generally a line that is based on the differences between the two moving averages consisting different lengths, i.e. a slow moving average and a fast moving average. It also shows the relationship between the two moving averages on how they change with respect to their prices and it always follows the same trend.

Another second line, generally called the “Signal line” which is plotted as the moving average of the MACD. There is also one more third line, called a “histogram” which is optional. It is generally plotted as the difference between the Signal line and the MACD Line [9].

The MACD is generally calculated by subtracting the 26 day Estimated Moving Average from the 12 day Estimated Moving Average. A 9 day Estimated Moving Average of the MACD, called the "signal line", is then plotted on top of the MACD as shown in the Figure 4.1.3. It acts as a point of reference to trigger the selling and buying of signals.

MACD = Fast MA – Slow MA
MACD = ((12-day EMA) – (26-day EMA)) (Equation 4.1.3)

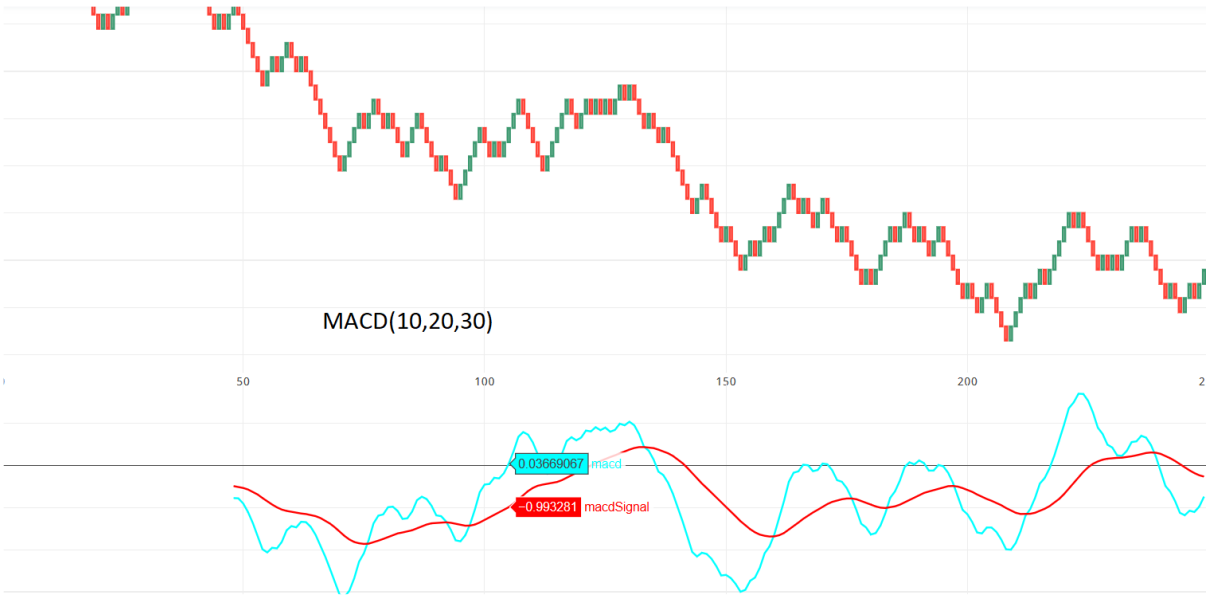


Figure 4.1.3 MACD Graph

4.1.4 RSI - Relative Strength Index:

J. Welles Wilder came up with the concept of The Relative Strength Index or RS, in short. Normalization of the current price is presented as a percentage ranging from 0 to 100. The name of this oscillator is deluding in the light of the fact that it does not quite match up to the instrument relative to another instrument or set of instruments, but rather speaks to the current price compared to other recent pieces within the selected lookback window length [10].

The RSI considers how big the recent gains and losses are, over a given certain time period to gauge or measure the speed and change of price developments of a security. It is fundamentally used to recognize the overbought and oversold condition in the trading of a security or asset.

RSI = 100 - 100 / (1 + RS) (Equation 4.1.4)

Where, RS = Average Gain of stock during the up periods of a specific time frame divided by the Average loss of stock during the down periods of a specific time frame.

The RSI has values ranging from 0 to 100. The default time span differentiating up periods to down periods is 14 trading days. Customary understanding and use of the RSI is that RSI estimations of 70 or above show that a security is turning out to be overbought or in other words overvalued, and thus may be primed for a trend reversal or corrective pullback in price, as shown in the Figure 4.1.4.

On an alternative note, an RSI reading of under 30 is generally interpreted as an indication overbought or oversold conditions that may flag a trend change or corrective price reversal to the upside.

Figure 4.1.4 – The RSI Graph

4.1.5 Bollinger Bands

John Bollinger was the man behind the conception and formulation of the Bollinger Bands. It works by plotting extremities of something called the “Envelope Bands” around the price of the stock or the instrument. The width between the bands is decided by the standard deviation of the closing prices from both the moving averages of the stock price.

The Bollinger Band is plotted as two standard deviations away from a simple moving average as shown in Figure 4.1.5. Lot of traders concur that closer the prices move towards the upper band, the stock is overbought in the market, and the more closer the prices move towards the lower band, the more stock is oversold in the market [11].

John Bollinger has come up with a set of 22 rules that are to be followed while the bands are being used as a trading system [12].

Upper-Band= SMA-20 + 2 * standard deviation of 20 day closing

Lower-Band= SMA-20 + 2 * standard deviation of 20 day closing

%B = ((price – lower band) / (upper band – lower band))*100

(Equation 4.1.5)

%B represents the price of a security or stock when compared to the lower and the upper Bollinger bands. The basic fundamental relationship levels are:

- %B is equal to 1, when stock price is at the upper band.
- %B is equal to 0, when stock price is at the lower band.
- %B is above 1, when stock price is above the upper band.
- %B is below 0, when stock price is below the lower band.

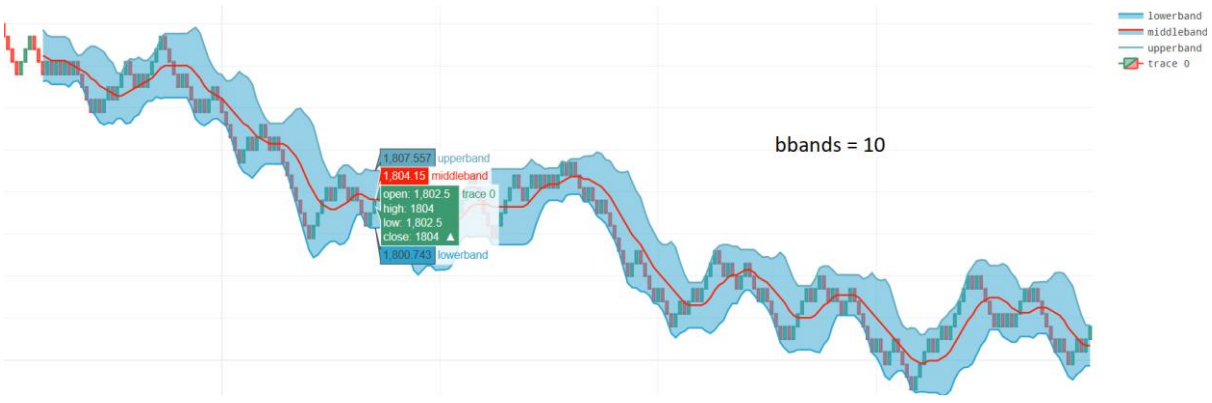


Figure 4.1.5 Bollinger Band Graph

4.1.6 Statistical Regression

Statistical Regression is generally a machine learning technique that is used for determining, how one or more independent variables in the set, bring a change or affect the interest of another variable, or a dependent variable.

The most important work that Statistical Regression does is that, it tries to draw a line that is closest to all of the dots on our graph, representing the dataset. This is generally called the “Line of Best fit”, and it gives us the idea, by how much the value of Y changes with every change in the value of X.

When the Statistical Regression is paired up with different assumptions in the form of a statistical model, the regression technique can be useful in different applications like prediction which includes forecasting of different time-series data, data inference, and testing of hypothesis

It is a way of determining a relation between variables. There are 2 independent variables here.
(i) Index
(ii) Stock Price

y = mx +c (trend line equation)
Where y = closing price, x = nth number, m =slope

m = (sum(x*y)-sum(y)*sum(x))/sum((x-mean(x))^2) (Equation 4.1.6)

c = y – mx

Buy & Sell Signals:

When prices reach the level of the upper band, it is considered as a sell signal and when the prices reach the lower band, it is considered as a buy signal.

%B (Percentage bandwidth):

- If (%B > 1 or %B <100%), it means that the prices are above upper band.
- If (%B == 1 or %B == 100%), it means that the price is at the upper band.
- If (%B above .50 or 50%), it means that the price is above the middle line.
- If (%B below .50 or 50%), it means that the price is below the middle line.
- If (%B == 0 or 0%), it means that the price is at lower band.
- If (%B<0 or 0%), it means that the price is below the lower band.

4.2 Determining Buy/Sell Signals

From all the indicators and oscillators we have seen above, we can come to a conclusion as to buy or sell a security or a stock or a signal with the help of the following rules [13].

- If previous SMA is lower than the previous Close Price and current SMA is greater than the current Close Price, then the Buy Signal is generated.
- If previous SMA is greater than the previous Close Price and current SMA is lower than the current Close Price, then the Sell Signal is generated.
- If previous EMA-20 is lower than the previous EMA-50 and current EMA-20 is greater than the current EMA-50, then the Buy Signal is generated.
- If previous EMA-20 is greater than the previous EMA-50 and current EMA-20 is lower than the current EMA-50, then the Sell Signal is generated.
- If previous MACD value is less than previous EMA-9 and current MACD value is greater than the current EMA9, then the Buy Signal is generated.
- If previous MACD value is greater than previous EMA-9 and current MACD value is lower than the current EMA-9, then the Sell Signal is generated.
- If (Close Price - lower band) < (Upper Band - Close Price) and both the differences are greater than 0, then the buy Signal is generated.
- If (Upper band – Close Price) < (Close Price – Lower Band) and both the differences are greater than 0, then the Sell signal is generated.

Chapter 5

Real Time Model

RBF Neural Network

Because of the very unstable nature of financial markets, it is usually acknowledged that stock value expectation is an assignment loaded with challenge. However, to make profits or comprehend the essence of equity market, various market participants or specialists attempt to figure out stock value utilizing different statistical, econometric or even neural network models.

Financial market is a complex, dynamic, chaotic and nonlinear system. However, it does not follow an arbitrary walk process [14]. There are numerous elements that might cause a fluctuation in the movement of the functioning of a financial market. The primary components include economic condition, political situation, traders' expectations, calamities and other unforeseen circumstances.

So, forecasting the stock market price and its direction are not quite easy. Considering the fact that the stock markets are reliant on each other, one should take into consideration, how this dependency could impact of process of making precise conjectures. This interdependency has a particularly enormous effect on stock market indices in the developing nations [15].

In this project, a novel and a particular predictive statistical method called the "Radial Basis Function Neural Network (RBF–NN)" has been implemented as a forecasting model and a predictive modelling technique to study daily movement of different stock indices.

RBF-NN is appropriate for demonstrating complex non-linear connections and conditions between different stock indices. The RBF-NN model is an extraordinary feed-forward neural network system with one input, one hidden, and one linear output layers. In the RBF-NN model, the number of parameters is less than the Multilayer Neural Networks (ML-NN), because inputs are associated with the hidden layer, without completely utilizing the weights [16].

Different parameters of the RBF-NN model, such as width, weight and the center can be adjusted and changed accordingly. In addition to this, the center and width parameters belong to the hidden layer of the RBF-NN model. On the other hand, the weight parameters are the association of weights between the hidden and output layers.

Generally, the conventional RBF-NN learning algorithms are used to determine the best parameters by making use of different iterative strategies, for example, the gradient descent and the forward selection procedures. RBF-NNs are a sort of general linear model where the input data are moved to a highlighted space, by different non-linear transformations that are utilizing the different Radial Basis functions.

The RBF-NN model, in the above sense, can be written as a general linear model, with the dependent variable y and independent variables x1, x2... xm in the following form:

Y=Σ*wmj*=1j hj = w1h1+ w2h2+……+ wmhm (Equation 5.1)

Where the regressors, {hj(x)}, are called the fixed radial basis functions of the input variables, and {wj} are the unknown adaptable coefficients or weights.

For ex., for index n, output is X. where n can be anything between 1 – 30
For index 31st, stock price is predicted which is the next day stock price.

We have used the RBF Neural network for predicting a few technical analysis report parameters for the next day so that it becomes easier for an investor or trader to decide to either sell or buy or exchange the stocks. People invest and sell stocks on a daily basis looking or forecasting how the company is faring that particular day.

We have chosen the data from Quandl to be the data of Indian Companies where the previous 30 day stock price is used as training data.

The entire program runs in an environment of AWS Lamda which is a server less processing framework that runs code and consequently deals with the fundamental figuring assets.

Cron Job Scheduler

A Cron Job Scheduler is initialized which runs the code at a particular time of the day, generally after the closing of the stock exchange so that we can make an informed decision or suggest others to buy, sell or exchange a stock. It is a tool that enables us to scheduled task and run that particular task regularly on our server. Most of the system administrators would use cron job to run regularly to scheduled backups on timely basis on the different servers. In a regular Windows server environment, we would have to set up a “Scheduled Task” program instead of the cron job.

A cron daemon is a long running daemon procedure that executes the given instructions at the dates and time to witch it were assigned. In order plan a one-time undertaking with cron, we have to use “use at” or clump.

For directions that have be processed more than once (i.e. hourly, day by day or week by week), use have to use crontab, which has the accompanying choices:

- crontab filename to add a file to crontab
- crontab -e to edit the crontab file
- crontab -l to list the files in crontab
- crontab -r to remove a crontab file
- MAILTO=user@domain.com to email the output

The crontab order, makes a crontab document which contains different directions about how regularly the cron has to execute them. Every passage in the crontab document comprises of six fields,as indicated in the accompanying request :
"minute(s) hour(s) day(s) month(s) weekday(s) command(s)"

Chapter 6

Sentiment Analysis and Implementation

Social media has created a new method for people to voice their thoughts and opinions. Sentiment analysis is the process of extracting textual data and determining which emotions the author displays. This type of analysis is used in various ways: customer's opinion of a product, customer service and brand can be predicted; Analysis of brand's marketing strategy on any social media site, analysis of citizen's sentiment on changes in policy, government officials, campaigns, etc. For improving results in all these kinds of scenarios sentiments of statements created by people are determined in numerical terms.

This project will also conduct sentiment analysis on actual tweets extracted from Twitter website. Through its APIs, twitter lets developers and programmers have access to its data (or tweets).

The polarity of people's sentiments can be of a wide range and there are many human sentiments but for simplicity, this evaluation is limited to three categories of polarities namely:

- Positive
- Neutral
- Negative

Even though this evaluation is based on three polarities, it is generally a tough task to determine or predict a person's statement because it is very subjective in nature. Some percent of the people might disagree on the classification of a sentence with others. For example:

"I love to eat curd!"

"I hate to eat curd!"

The two statements can be very clearly determined weather positive or negative.

Some statements like "*the curd is pretty much more or less the same to me!"*

This statement can be said to be categorized as close to neutral but some may argue saying this statement indicates negativity. So this statement's classification is not as clear as previous statement's. Another example:

"The stock price of Reliance went up"

This statement might be positive to the organization and its investors but can be negative for its market competitors. But this statement is pretty dull in emotional nature as it is represented as a fact. Another example:

“I would love to have that curry, but it is too spicy for me.”

Here the person who is saying this starts off positively, but ends it in a negative way. So it is not clear whether the statement is overall positive or negative. The purpose of these examples is to show that sentiment analysis is not going to be perfect all the time and not all statements are clear to distinguish.

Tweepy is one of the Python library that can be used for accessing the Twitter API. It is actually a great tool for simple automation of various twitter entities and also for creating chat bots for twitter. Tweepy has a lot of unique features. It is also an open source library tool, which is hosted on GitHub generally and it enables Python to communicate with the Twitter platform and also make use of its API [17].

The Tweepy makes use of a method called OAuth to provide access to twitter. Generally twitter makes sure that Tweets are made in such a way that it can store a string which stores the details such as the app which was used. The best part is that it does not have to reveal any password, which makes it even more secure. The permissions can be managed easily by generating tokens and keys that allows only read only function to work on the timeline so that even if someone gets their hand on those credentials, they will not be able to login and send messages or make tweets.

The OAuth credentials used in this project are as follows:

ConsumerKey = 'nxB3JxRDmv2gyEzIDCb8nojzj'
ConsumerSecret = 'SsaYBiF7zxJGaRp2CmSHV4jh9nfsZ5lzfOSKlBYioxKaBuZeKl'
AccessToken = '209146458-rmjf0yo1uDa8KfCTALifvC88nEZMlDxwKxc0TSoS'
AccessTokenSecret = 'jQ7FYJDSiMYMqBIz9yWz8LtlkDpRlXiXcsIxPdw1dzdtk'

```
self.tweets = tweepy.Cursor(api.search,q=Company_Name,lang="lang").items(NoOfTweets)
```

The sentence mentioned above is used to search through the twitter database and search the term which has been assigned to the variable “searchTerm” and retrieve a fixed amount of tweets from the database, as mentioned in “NoOfTerms”.

The retrieved data is then appended and written to a csv file and encoded into a UTF-8 Format as it is the most compatible with the contents from the World Wide Web and also so that no special characters are deemed unwanted.

The tweets are cleaned and then calculated for their polarity from textblob’s inbuilt polarity calculator.

The average polarity is calculated by **polarity+=polarity/no of tweets**

We have considered 7 sentiments for categorizing the polarities so that it is easier for us to classify and conclude at one option. They have been categorized into the following: “Neutral, weak positive, positive, strong positive, weak negative, negative and strong negative.”

- If the sentiment polarity is equal to 0 then neutral += 1.
- If sentiment polarity > 0 and <= 0.3, then weak positive += 1
- If sentiment polarity > 0.3 and <= 0.6, then positive += 1
- If sentiment polarity > 0.6 and <= 1, then strong positive += 1
- If sentiment polarity > -0.3 and <= 0, then weak negative += 1
- If sentiment polarity > -0.6 and <= -0.3, then negative += 1
- If sentiment polarity > -1 and <= -0.6, then strong negative += 1

The values are then converted into a pie chart for analyzing different emotions expressed by people over a period of time.

The huge number of social media posts that are being posted by people on a daily basis can create a short term effect that would influence the stock prices and other parameters of a certain entity or a product. They also contribute and play a major role in determining the future of a particular brand or a stock.

Sentiment analysis, is one such tool that considers the general public's opinion and brings in a meaning to the relationship between the market and the social media.

Chapter 6
Screenshots

Sample Company Name: Reliance Industries Limited
Date of Report: 15/07/2020
Stock Value: Rs.1587.490

```
Enter Ticker:Reliance
Name: Reliance Industries Limited
Technical Analysis Report:
SMA says Downtrend
Support is Rs. 1089.15
Resistance is Rs. 1136.8
```

Figure 6.1 Predicted Value

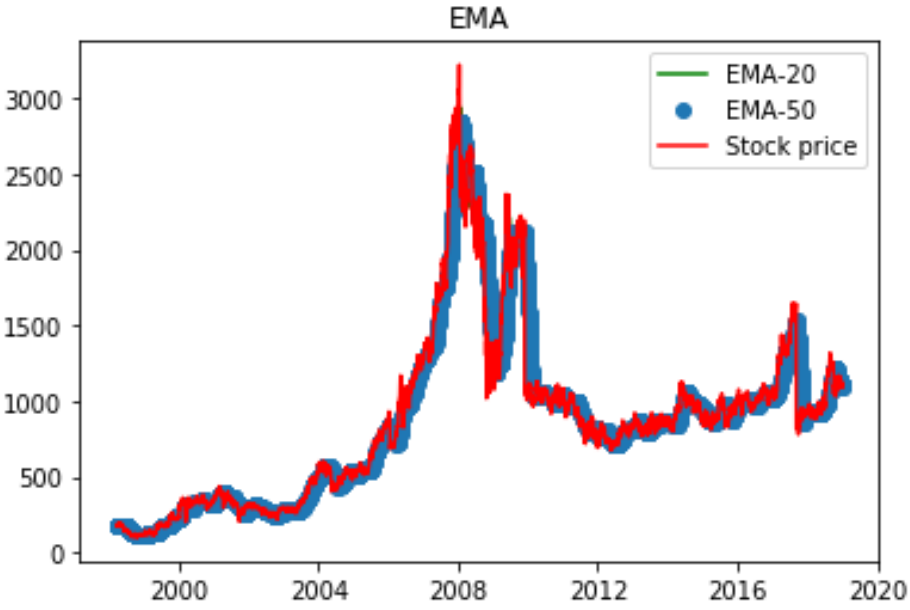


Figure 6.2 SMA Graph for Reliance

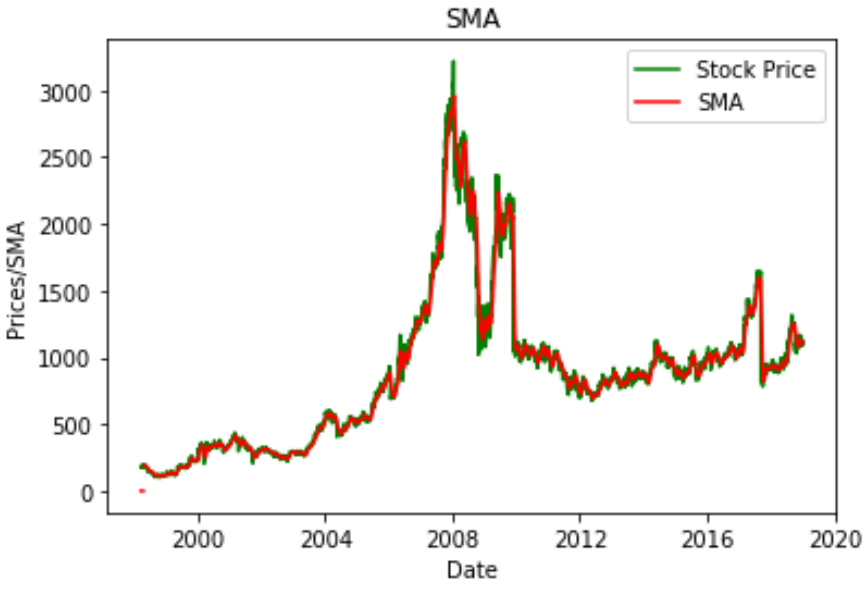


Figure 6.3 EMA Graph for Reliance

```
MACD: -13.6 and Signal Line: -13.29
```

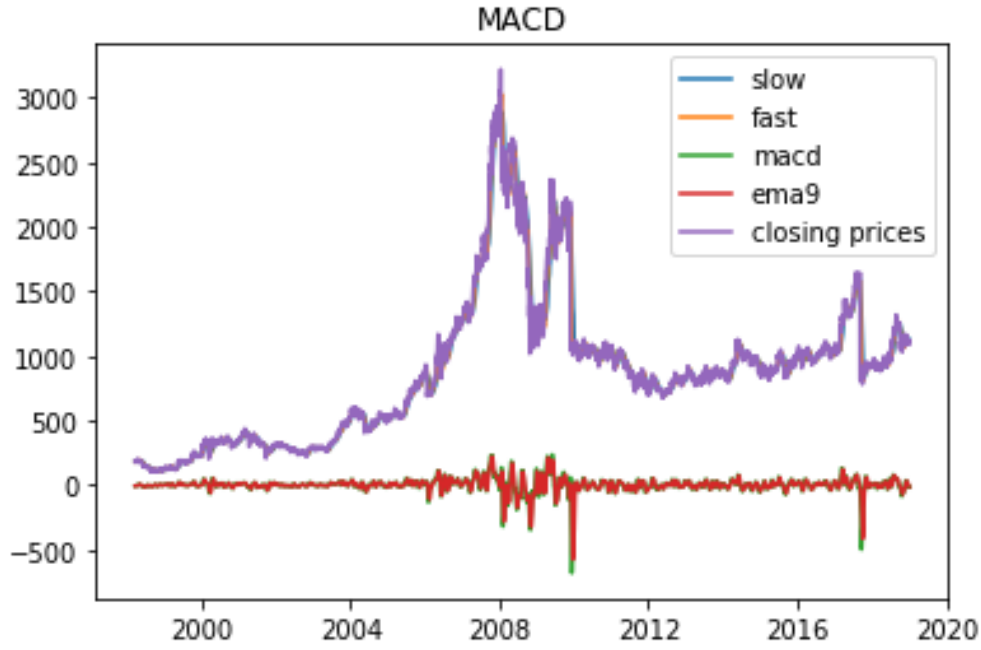
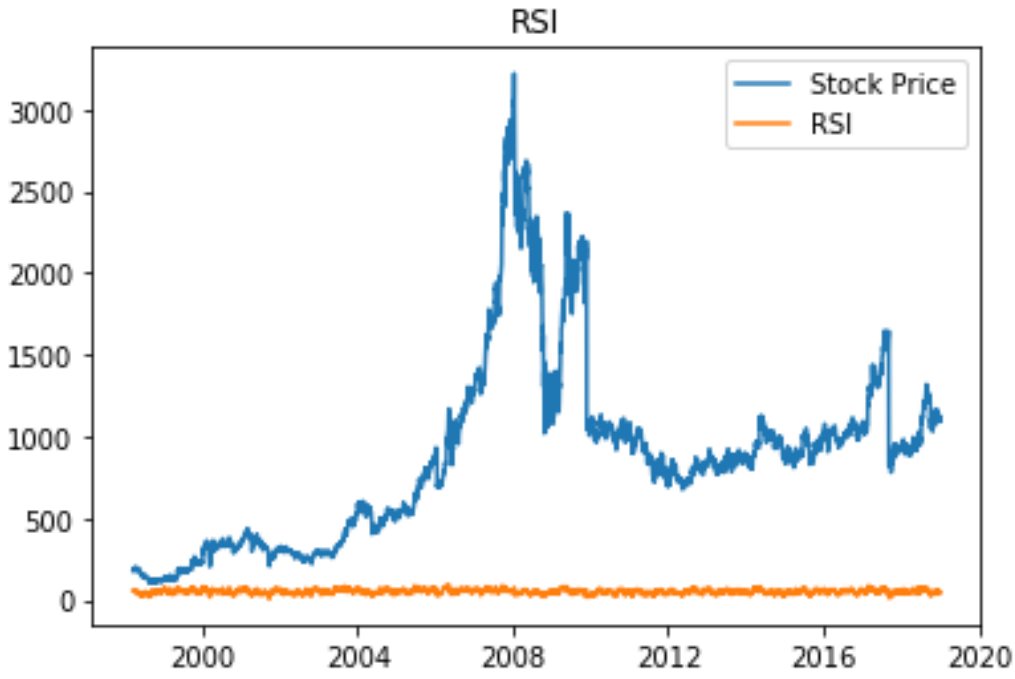


Figure 6.4 MACD Graph for Reliance




```
RSI is: 44.51
%B is: 27.75 %
```

Figure 6.5 RSI Graph for Reliance

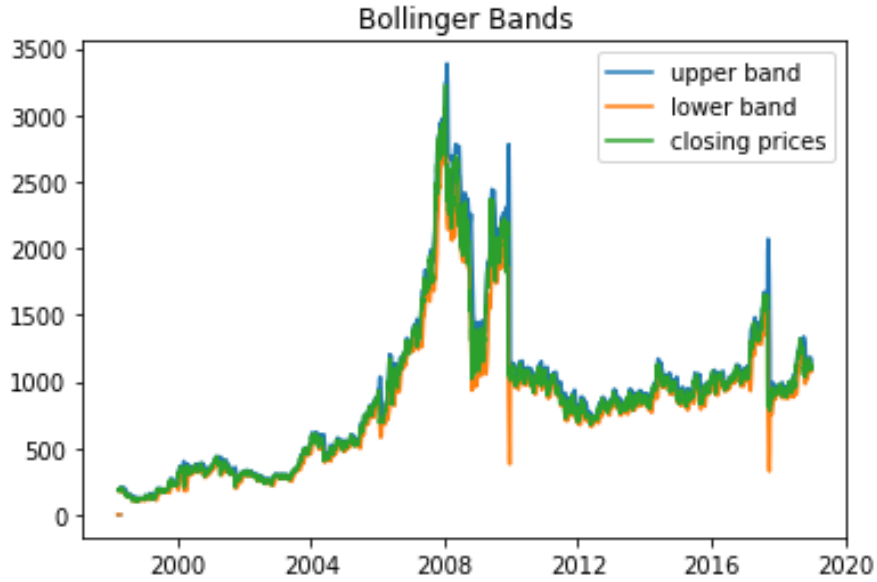


Figure 6.6 Bollinger Band Graph for Reliance

```
Enter how many tweets to search for Sentimental Analysis: 500
How people are reacting on Reliance by analyzing 500 tweets.

General Report:
Weakly Positive

Detailed Report:
12.80% people thought it was positive
24.00% people thought it was weakly positive
9.40% people thought it was strongly positive
5.40% people thought it was negative
12.40% people thought it was weakly negative
0.20% people thought it was strongly negative
35.40% people thought it was neutral
```

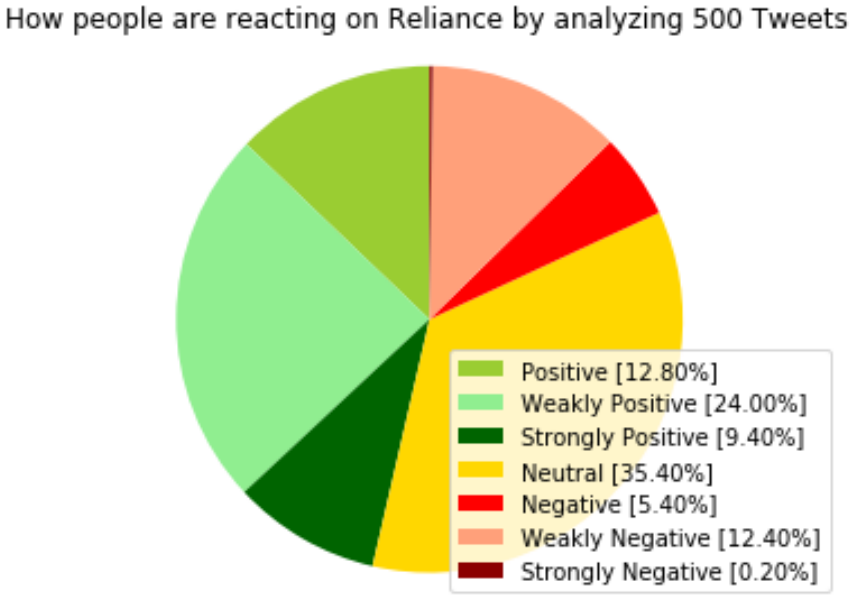


Figure 6.7 Sentiment Analysis Graph for Reliance

Chapter 7

Conclusion and Future Work

Conclusion

Technical Analysis of stock market is a method of evaluating the securities by analyzing the different statistics or graphs that are generated by the daily market activity. Chartists uses these data to determine whether they are going to invest in that stock. They look for the buy and sell signals.

This project has also performed technical analysis over stock prices by using some of the mostly used indicators and oscillators. We have also tried to predict the value of different technical analysis securities such as stock price, SMA, EMA, RSI, MACD and Bollinger Bands value for the next day.

These securities are influenced by many factors including internal information, news, Speeches of ministers etc. So this does not give accurate results all the time. But it is better than blindly investing in it.

Hence, it can be concluded that a smart investing decision can be made if one uses Technical indicators and oscillators and other sorts of information like news etc. before making an investment.

Some of the other conclusions are:

- Executing Trades is not an easy job.
- There is no perfect strategy which works all the time.
- Hit and Trail is the best method to find out working strategies.
- There is both profit and loss but losses tend to be minimized as compared to the conventional trading.

Future Work

Some future works in this area will be to

- Increase the accuracy of prediction.
- Add more Indicators and oscillators.
- Make a Dashboard with all the necessary indicators and have a backend to integrate buy and sell options affiliating directly to the NSE.
- Develop a web application using the dashboard and act as a point of reference for stock brokers.

Appendices

Appendix A

Code

```
import quandl
import requests
import matplotlib.pyplot as plt
import pandas
from pandas import DataFrame
import datetime
import numpy as np
import matplotlib.patches as mpatches
from math import sqrt
from sklearn.svm import SVR
import sys,tweepy,csv,re
from textblob import TextBlob

quandl.ApiConfig.api_key = 'dPyyCmjxE9TZP6Jfdztp'

def email(txt):

sg=sendgrid.SendGridAPIClient(apikey='SG.uj_DUPBYSrKiEf2qq1Fpjg.az_jnV7Ngq_fTOGQZLI
i0vwHIWwXN-zvHITar8E1SuY')
    from_email = Email("aravind.krishnan.mit@gmail.com")
    to_email = Email("saiaravind033@gmail.com")
    subject = "Technical Analysis Report - Reliance"
    content = Content("text/plain",txt)
    mail = Mail(from_email, subject, to_email, content)
    response = sg.client.mail.send.post(request_body=mail.get())
    print(response.status_code)
    print(response.body)
    print(response.headers)

 def drawChartSMA(close, date, sma):
     y = close
     x = date
     plt.plot(x, y, 'g', label='Stock Price')
     plt.plot(x, sma, 'r', label='SMA')
     plt.xlabel('Date')
     plt.ylabel('Prices/SMA')
     plt.title('SMA')
     plt.legend()
     temp_length = len(close)
     k = temp_length - 20
     min = close[k]
     max = close[k]
     for t in range(k, temp_length):
         if close[t] < min:
             min = close[t]
         if close[t] > max:
             max = close[t]

     if sma[len(sma) - 1] < close[len(close) - 1]:
         print('SMA says Uptrend')
         print('Support is Rs.', min)
         print('Resistance is Rs.', max)
     elif sma[len(sma) - 1] > close[len(close) - 1]:
         print('SMA says Downtrend')
         print('Support is Rs.', min)
         print('Resistance is Rs.', max)
     else:
         print('Trend Reversal may occur')
```

```
print('Support is Rs.', min)
        print('Resistance is Rs.', max)
    plt.show()


def movingaverage(values, window):
    weights = np.repeat(1., window) / window
    smas = np.convolve(values, weights, 'valid').tolist()
    listt = list()
    for c in range(0, window - 1):
        listt.append(0)
    smas = listt + smas
    for m in range(0, len(smas)):
        smas[m] = float('{0:.2f}'.format(smas[m]))
    return smas


def ExpMovingAverage(values, window):
    weights = np.exp(np.linspace(-1., 0., window))
    weights /= weights.sum()
    a = np.convolve(values, weights)[:len(values)]
    a[:window] = a[window]
    return a


def computeMACD(x, slow=26, fast=12):
    emaslow = ExpMovingAverage(x, slow)
    emafast = ExpMovingAverage(x, fast)
    return (emaslow, emafast, emafast - emaslow)

def rsiFunc(close, n, date):
    deltas = np.diff(close)
    seed = deltas[:n + 1]
    up = seed[seed >= 0].sum() / n
    down = -seed[seed < 0].sum() / n
    rs = up / down
    rsi = np.zeros_like(close)
    rsi[:n] = 100. - 100. / (1 + rs)
    for i in range(n, len(close)):
        delta = deltas[i - 1]
        if delta > 0:
            upval = delta
            downval = 0.
        else:
            upval = 0.
            downval = -delta
        up = (up * (n - 1) + upval) / n
        down = (down * (n - 1) + downval) / n
        rs = up / down
        rsi[i] = 100. - 100. / (1. + rs)
        rsi[i] = float('{0:.2f}'.format(rsi[i]))
    plt.plot(date, close.tolist(), label='Stock Price')
    plt.plot(date, rsi.tolist(), label='RSI')
    plt.legend()
    plt.title('RSI')
    plt.show()
    return rsi


def bollinger_band(close, window):
    rolling_mean = close.rolling(window=20).mean()
    rolling_std = close.rolling(window=20).std()
    upper_band = rolling_mean + rolling_std * 2
    lower_band = rolling_mean - rolling_std * 2
    cl = close.tolist()
    lo = lower_band.tolist()
    up = upper_band.tolist()
    b = (cl[len(cl) - 1] - lo[len(lo) - 1]) / (up[len(up) - 1]
                                        - lo[len(lo) - 1])
```

```
    print('%B is:', float('{0:.2f}'.format(b * 100)), '%')
    return (rolling_mean, upper_band, lower_band)

def regression(price):
    price = price.tolist()
    lis = list()
    lis2 = list()
    x = len(price)
    a = x - 20
    X = cal_X_mean(x)
    Y = cal_Y_mean(price, x)

    # y-Y

    for i in range(0, 20):

        lis.append(price[a] - Y)
        a = a + 1

    # x-X

    a = 0
    for i in range(x - 20, x):
        lis2.append(i - X)
        a = a + 1
    lis3 = []
    for i in range(0, 20):
        lis3.append(lis[i] * lis2[i])
    lis4 = []
    for i in range(0, 20):
        lis4.append(lis2[i] * lis2[i])
    sum1 = 0
    sum2 = 0
    for i in range(0, 20):
        sum1 = sum1 + lis3[i]
        sum2 = sum2 + lis4[i]
    m = sum1 / sum2
    m = float('{0:.2f}'.format(m))
    b = Y - m * X
    b = float('{0:.2f}'.format(b))
    print('TrendLine equation is : y = ', m, '* x +(', b, ')')

    return 0


def cal_X_mean(x):

    sum = 0
    for i in range(x - 20, x):
        sum = sum + i
    return sum / 20


def cal_Y_mean(close, x):

    sum = 0
    for i in range(x - 20, x):
        sum = sum + close[i]
    return sum / 20


def plotMACD(
    close,
    ema9,
    emaslow,
    emafast,
    macd,
    date,
):
    plt.plot(date, emaslow.tolist(), label='slow')
    plt.plot(date, emafast.tolist(), label='fast')
    plt.plot(date, macd.tolist(), label='macd')
    plt.plot(date, ema9.tolist(), label='ema9')
```

```
    plt.plot(date, close, label='closing prices')


plt.title('MACD')
    plt.legend()
    plt.show()


def plotB_band(
    b,
    c,
    date,
    close,
):
    plt.title('Bollinger Band')
    plt.plot(date, b, label='upper band')
    plt.plot(date, c, label='lower band')
    plt.plot(date, close, label='closing prices')
    plt.title('Bollinger Bands')
    plt.legend()
    plt.show()


def plotEMA(
    ema50,
    ema20,
    date,
    close,
):
    plt.plot(date.tolist(), ema20, 'g', label='EMA-20')
    plt.plot(date.tolist(), ema50, 'o', label='EMA-50')
    plt.plot(date.tolist(), close, 'r', label='Stock price')
    plt.legend()
    plt.title('EMA')
    plt.show()


def predict_price(date, close, x):
    close = close.tolist()
    x = 31
    y = len(close) - x
    closed = []

    date[0] = 0

    for i in range(0, 30):
        if y < len(close):
            closed.append(close[y])
            y = y + 1

    date = np.array(date).reshape((len(date), 1))

    svr_lin = SVR(kernel='linear', C=1e3)

    svr_poly = SVR(kernel='poly', C=1e3, degree=2)
    svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)

    svr_lin.fit(date, closed)
    svr_poly.fit(date, closed)
    svr_rbf.fit(date, closed)

    return (svr_rbf.predict(x)[0], svr_lin.predict(x)[0],
            svr_poly.predict(x)[0])


try:
    name = input('Enter Ticker:')
    url = 'https://www.quandl.com/api/v3/datasets/NSE/' + name \
        + '/metadata.json'
    try:
        meta_data = requests.get(url)
        parsed_meta_data = meta_data.json()
        print('Name: {}'.format(parsed_meta_data['dataset']['name']))
```

```
  print('Technical Analysis Report:', ".format(parsed_meta_data['dataset']['name']))
searched_data = quandl.get('NSE/' + name)
      close = searched_data.Close
      n = 20
      sma = movingaverage(close.tolist(), n)
      ema50 = ExpMovingAverage(close, 50)
      ema20 = ExpMovingAverage(close, 20)
      ema50 = ema50.tolist()
      ema20 = ema20.tolist()
      for x in range(0, len(ema50)):
         ema50[x] = float('{0:.2f}'.format(ema50[x]))
         ema20[x] = float('{0:.2f}'.format(ema20[x]))

      date = pandas.to_datetime(searched_data.index)
      drawChartSMA(close.tolist(), date.tolist(), sma)
      plotEMA(ema50, ema20, date, close)
      nema = 9
      (emaslow, emafast, macd) = computeMACD(close)
      ema9 = ExpMovingAverage(macd, nema)
      print('MACD:', float('{0:.2f}'.format(macd[len(macd) - 1])),
          'and Signal Line:',
          float('{0:.2f}'.format(ema9[len(ema9) - 1])))
      plotMACD(
         close,
         ema9,
         emaslow,
         emafast,
         macd,
         date.tolist(),
      )
      rsi = rsiFunc(close, 14, date.tolist())
      print('RSI is:', rsi[len(rsi) - 1])
      if rsi[len(rsi) - 1] > 70:
         print('Stock may be Overbought')
      if rsi[len(rsi) - 1] < 30:
         print('Stock may be Oversold')
      (a, b, c) = bollinger_band(close, 20)
      a[:19] = 0
      b[:19] = 0
      c[:19] = 0
      plotB_band(b.tolist(), c.tolist(), date.tolist(),
              close.tolist())
      dates = []
      for x in range(0, 30):
         dates.append(x)

      (e, f, g) = predict_price(dates, close, 30)
      print(float('{0:.2f}'.format(f)))
      regression(close)
   except Exception:

      print('May be No. Of Attempts for today by the key is finished')
except Exception:
   print('Check your Internet Connection')

 class SentimentAnalysis:
 def __init__(self):
      self.tweets = []
      self.tweetText = []

   def DownloadData(self):
      consumerKey = 'nxB3JxRDmv2gyEzIDCb8nojzj'
      consumerSecret = 'SsaYBiF7zxJGaRp2CmSHV4jh9nfsZ5lzfOSKlBYioxKaBuZeKl'
      accessToken = '209146458-rmjf0yo1uDa8KfCTALifvC88nEZMlDxwKxc0TSoS'
      accessTokenSecret = 'jQ7FYJDSiMYMqBIz9yWz8LtlkDpRlXiXcsIxPdw1dzdtk'
      auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
      auth.set_access_token(accessToken, accessTokenSecret)
      api = tweepy.API(auth)
```

```
searchTerm = input("Enter Keyword/Tag to search about: ")
  NoOfTerms = int(input("Enter how many tweets to search: "))


  self.tweets = tweepy.Cursor(api.search, q=searchTerm, lang = "en").items(NoOfTerms)
  csvFile = open('result.csv', 'a')
  csvWriter = csv.writer(csvFile)

  polarity = 0
  positive = 0
  wpositive = 0
  spositive = 0
  negative = 0
  wnegative = 0
  snegative = 0
  neutral = 0

  for tweet in self.tweets:

    self.tweetText.append(self.cleanTweet(tweet.text).encode('utf-8'))

    analysis = TextBlob(tweet.text)

    polarity += analysis.sentiment.polarity

    if (analysis.sentiment.polarity == 0):
      neutral += 1
    elif (analysis.sentiment.polarity > 0 and analysis.sentiment.polarity <= 0.3):
      wpositive += 1
    elif (analysis.sentiment.polarity > 0.3 and analysis.sentiment.polarity <= 0.6):
      positive += 1
    elif (analysis.sentiment.polarity > 0.6 and analysis.sentiment.polarity <= 1):
     positive += 1
   elif (analysis.sentiment.polarity > 0.6 and analysis.sentiment.polarity <= 1):
     spositive += 1
   elif (analysis.sentiment.polarity > -0.3 and analysis.sentiment.polarity <= 0):
     wnegative += 1
   elif (analysis.sentiment.polarity > -0.6 and analysis.sentiment.polarity <= -0.3):
     negative += 1
   elif (analysis.sentiment.polarity > -1 and analysis.sentiment.polarity <= -0.6):
     snegative += 1


csvWriter.writerow(self.tweetText)
csvFile.close()


positive = self.percentage(positive, NoOfTerms)
wpositive = self.percentage(wpositive, NoOfTerms)
spositive = self.percentage(spositive, NoOfTerms)
negative = self.percentage(negative, NoOfTerms)
wnegative = self.percentage(wnegative, NoOfTerms)
snegative = self.percentage(snegative, NoOfTerms)
neutral = self.percentage(neutral, NoOfTerms)


polarity = polarity / NoOfTerms


print("How people are reacting on " + searchTerm + " by analyzing " + str(NoOfTerms) + " tweets.")
print()
print("General Report: ")
```

```
  if (polarity == 0):
        print("Neutral")
     elif (polarity > 0 and polarity <= 0.3):
        print("Weakly Positive")
     elif (polarity > 0.3 and polarity <= 0.6):
        print("Positive")
     elif (polarity > 0.6 and polarity <= 1):
        print("Strongly Positive")
     elif (polarity > -0.3 and polarity <= 0):
        print("Weakly Negative")
     elif (polarity > -0.6 and polarity <= -0.3):
        print("Negative")
     elif (polarity > -1 and polarity <= -0.6):
        print("Strongly Negative")

     print()
     print("Detailed Report: ")
     print(str(positive) + "% people thought it was positive")
     print(str(wpositive) + "% people thought it was weakly positive")
     print(str(spositive) + "% people thought it was strongly positive")
     print(str(negative) + "% people thought it was negative")
     print(str(wnegative) + "% people thought it was weakly negative")
     print(str(snegative) + "% people thought it was strongly negative")
     print(str(neutral) + "% people thought it was neutral")

     self.plotPieChart(positive, wpositive, spositive, negative, wnegative, snegative, neutral, searchTerm,
NoOfTerms)


  def cleanTweet(self, tweet):

     return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t]) | (\w +:\ / \ / \S +)", " ", tweet).split())

  def percentage(self, part, whole):
     temp = 100 * float(part) / float(whole)
     return format(temp, '.2f')

  def plotPieChart(self, positive, wpositive, spositive, negative, wnegative, snegative, neutral, searchTerm,
noOfSearchTerms):
     labels = ['Positive [' + str(positive) + '%]', 'Weakly Positive [' + str(wpositive) + '%]','Strongly Positive [' +
str(spositive) + '%]', 'Neutral [' + str(neutral) + '%]',
           'Negative [' + str(negative) + '%]', 'Weakly Negative [' + str(wnegative) + '%]', 'Strongly Negative ['
+ str(snegative) + '%]']
     sizes = [positive, wpositive, spositive, neutral, negative, wnegative, snegative]
     colors = ['yellowgreen','lightgreen','darkgreen', 'gold', 'red','lightsalmon','darkred']
     patches, texts = plt.pie(sizes, colors=colors, startangle=90)
     plt.legend(patches, labels, loc="best")
     plt.title('How people are reacting on ' + searchTerm + ' by analyzing ' + str(noOfSearchTerms) + ' Tweets.')
     plt.axis('equal')
     plt.tight_layout()
     plt.show()



if __name__ == "__main__":
  sa = SentimentAnalysis()
  sa.DownloadData()
```

References:

[1] E. F. Fama, Efficient market hypothesis: "A review of theory and empirical work", Journal of Finance, 25(2), 28-30, (1970).

[2] M. &. Y. S. Patel, "Stock Price Prediction Using Artificial Neural Network", 2014.

[3] D. C. C. I. &. K. H. Komo, "Stock market index prediction using neural networks. In Applications of Artificial Neural Networks V", *International Society for Optics and Photonics.,* pp. (Vol. 2243, pp. 516-526), (1994, March).

[4] R. W. Hendrix, “Lambda”, 1983. [Online]. Available: https://docs.aws.amazon.com/lambda/index.html..

[5] Sendgrid, “Getting Started with the SendGrid API”, [Online]. Available: https://sendgrid.com/docs/for-developers/sending-email/api-getting-started/.

[6] N. M. Shah, "Trading strategies for autonomous agents in financial markets", (2013).

[7] A. Hayes, "Simple Moving Average (SMA) Definition", Investopedia, 26 4 2019. [Online]. Available: https://www.investopedia.com/terms/s/sma.asp. [Accessed 1 3 2020].

[8] A. Hayes, "Exponential Moving Average (EMA)", Investopedia, 26 4 2019. [Online]. Available: https://www.investopedia.com/terms/s/ema.asp. [Accessed 1 3 2020].

[9] A. Hayes, "Moving Average Convergence Divergence – MACD Definition", Investopedia, 19 04 2019. [Online]. Available: https://www.investopedia.com/terms/m/macd.asp [Accessed 01 03 2020].

[10] J. Chen, "Relative Strength Index (RSI)", Investopedia, 26 04 2019. [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp [Accessed 01 03 2020].

[11] C. Mitchell, "Using Bollinger Bands to Gauge Trends", Investopedia, 26 04 2019. [Online]. Available: https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends. [Accessed 02 03 2020].

[12] J. Bollinger,"Bollinger on Bollinger Bands", McGraw Hill Professional, 2002.

[13] J. M. I. &. S. M. Stanković, "Investment strategy optimization using technical analysis and predictive modeling in emerging markets", *Procedia Economics and Finance,* pp. 51-62, 2015.

[14] A. W. &. M. A. C. Lo, "Stock market prices do not follow random walks: Evidence from a simple specification test", *The review of financial studies,* vol. 41, no. 66, 1988.

[15] M. B. &. Y. S. R. Patel, "Stock price prediction using artificial neural networks", *International Journal of Innovative Research in Science, Engineering and Technology,* vol. 3(6), no. 13755-13762, 2014.

[16] P. Kumar, "*STOCK MARKET FORECASTING USING RBF NEURAL NETWORKS",* 10.13140/RG.2.1.4141.5525, 2015.

[17] A. Guha, "Twitter Sentimental Analysis", Medium, 31 05 2019. [Online]. Available: https://medium.com/@abhishekguha/twitter-sentimental-analysis-17142f4ee13d. [Accessed 28 05 2020].

Table B.1: Project Details

Student Details

Student Name	Sai Aravind Singaravelu		
Registration Number	150953188	Section/Roll No. CCE A	A/01
Email Address	aravind.krishnan.mit@gmail.com	Phone No.(M)	8050297127
Student Name	Nagam Lahari Teja		
Registration Number	150953218	Section/Roll No. CCE A	A/01
Email Address	nagamtej@gmail.com	Phone No.(M)	9848566009

Project Details

Project Title	Smart Investment and Financial Trading System		
Project Duration	4-6 Months	Date of Reporting	13-01-2020

Organization Details

Organization Name	Manipal Institute Of Technology
Full Postal Address	Eshwar Nagar , Manipal , Karnataka 576104
Website Address	www.manipal.edu

Supervisor Details

Supervisor Full Name	N/A		
Designation	N/A		
Full Contact Address with PIN Code	N/A		
Email Address	N/A	Phone No.(M)	N/A

Internal Guide Details

Faculty Name	Mr. Arjun CV
Full Contact Address with PIN Code	Department of Information and Communication Technology, Manipal Institute of Technology, Manipal-576104
Email Address	arjun.cv@manipal.edu

re-re

ORIGINALITY REPORT

11%	5%	5%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Manipal University Student Paper	2%
2	Submitted to Amrita Vishwa Vidyapeetham Student Paper	2%
3	link.springer.com Internet Source	1%
4	Submitted to Coventry University Student Paper	1%
5	"An Analysis on Impact of Exchange Rate on Nifty and S & P Sensex", International Journal of Innovative Technology and Exploring Engineering, 2019 Publication	<1%
6	Submitted to Xavier Institute of Management and Entrepreneurship, Bangalore Student Paper	<1%
7	digitalcommons.wpi.edu Internet Source	<1%
	Submitted to Oklahoma City University	

8	Student Paper	<1%
9	Submitted to Mont Blanc Palace Student Paper	<1%
10	Dilip Singh Sisodia, Ritvika Reddy. "chapter 5 Analysis of Public Sentiments About Mega Online Sale Using Tweets on Big Billions Day Sale", IGI Global, 2019 Publication	<1%
11	library.tradingtechnologies.com Internet Source	<1%
12	Submitted to Harrisburg University of Science and Technology Student Paper	<1%
13	www.tinylist.org Internet Source	<1%
14	scholarcommons.sc.edu Internet Source	<1%
15	forexpathway.com Internet Source	<1%
16	medium.com Internet Source	<1%
17	bitcoinbro.com Internet Source	<1%

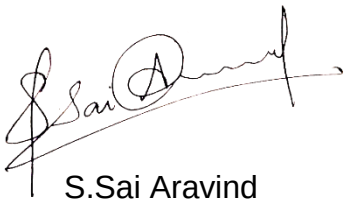
18	Gil Cohen. "Algorithmic setups for trading popular U.S. ETFs", Cogent Economics & Finance, 2020 Publication	<1%
19	www2.wealth-lab.com Internet Source	<1%
20	Submitted to University of Westminster Student Paper	<1%
21	www.ijbui.com Internet Source	<1%
22	Submitted to University of Leicester Student Paper	<1%
23	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
24	Submitted to University of South Africa Student Paper	<1%
25	community.scottrade.com Internet Source	<1%
26	first.ijs.si Internet Source	<1%
27	Submitted to National University of Ireland, Maynooth Student Paper	<1%

28	machinelearningmastery.com Internet Source	<1%
29	hdl.handle.net Internet Source	<1%
30	Rakhi Mahanta, Trilok Nath Pandey, Alok Kumar Jagadev, Satchidananda Dehuri. "Optimized Radial Basis Functional neural network for stock index prediction", 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016 Publication	<1%
31	Submitted to Birkbeck College Student Paper	<1%
32	Submitted to Xianjiatong-Liverpool University Student Paper	<1%
33	Submitted to City University Student Paper	<1%
34	Oguz Akbilgic, Hamparsum Bozdogan, M. Erdal Balaban. "A novel Hybrid RBF Neural Networks model as a forecaster", Statistics and Computing, 2013 Publication	<1%
35	Submitted to Universiti Teknologi MARA Student Paper	<1%

36	Submitted to University of Brighton Student Paper	<1%
37	"Breast Cancer Detection using Machine Learning Way", International Journal of Recent Technology and Engineering, 2019 Publication	<1%
38	wiredspace.wits.ac.za Internet Source	<1%
39	Submitted to Bond University Student Paper	<1%
40	Shipra Banik, A. F. M. Khodadad Khan, Mohammad Anwer. "Dhaka stock market timing decisions by hybrid machine learning technique", 2012 15th International Conference on Computer and Information Technology (ICCIT), 2012 Publication	<1%
41	upcommons.upc.edu Internet Source	<1%
42	Ou, Phichhang, and Hengshan Wang. "Prediction of Stock Market Index Movement by Ten Data Mining Techniques", Modern Applied Science, 2009. Publication	<1%
43	Submitted to UCSI University Student Paper	<1%

44	Submitted to British University in Egypt Student Paper	<1%

Exclude quotes	On	Exclude matches	< 3 words
Exclude bibliography	On		



S.Sai Aravind

Nagam Lahari Teja