A PROJECT REPORT

ON

STOCK MARKET ANALYSIS AND PREDICTION

BY

AMAN KUMAR SINGH - (14207)
AMAN SONKAR - (14208)
ASHUTOSH RANA - (14215)
RAHUL KUMAR GUPTA- (14236)

UNDER THE SUPERVISION OF

Dr. ABHAY KUMAR AGARWAL

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KAMLA NEHRU INSTITUTE OF TECHNOLOGY SULTANPUR

(U.P.) – 228118

(An Autonomous State Government Institute)

AFFILIATED TO

DR A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY

LUCKNOW (U.P.) INDIA

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

KAMLA NEHRU INSTITUTE OF TECHNOLOGY

SULTANPUR (U.P.) – 228118

CERTIFICATE

This is to certify that **Aman Kumar Singh (14207)**, **Aman Sonkar (14208)**, **Ashutosh Rana (14215) & Rahul Kumar Gupta (14236)** have carried out the project work in this report entitled "**STOCK MARKET ANALYSIS AND PREDICTION"** for the award of Bachelor of Technology in Computer Science and Engineering at Kamla Nehru Institute of Technology, affiliated to Dr. A. P. J. Abdul Kalam Technical University (AKTU), Lucknow.

This report is the record of the candidates' own work carried out by them under our supervision and guidance. This project work is the part of their Bachelor of Technology in Computer Science and Engineering curriculum.

Their performance was good and we wish them good luck for their future endeavours.

Dr. Abhay Kumar Agarwal	**Dr. Neelendra Badal**	**Dr. Anil Kumar Malviya**
(Assistant Professor)	**(Professor)**	**(Professor)**
(Project Guide)	**(Project In-Charge)**	**(Head of Department)**

DECLARATION

We declare that the work presented in this project titled **"Stock Market Analysis and Prediction"**, submitted to **Department of Computer Science & Engineering** at **Kamla Nehru Institute of Technology**, affiliated to **Dr. A. P. J. Abdul Kalam Technical University (AKTU), Lucknow** for the award of the **Bachelor of Technology** degree in Computer Science & Engineering, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

| --- | --- | --- | --- |
| Aman Kumar Singh (14207) | Aman Sonkar (14208) | Ashutosh Rana (14215) | Rahul Kumar Gupta (14236) |

ACKNOWLEDGEMENT

Here, we gladly present this project report on **"STOCK MARKET ANALYSIS AND PREDICTION"** as part of the 8^{th} semester B.Tech in Computer Science and Engineering. We take this occasion to thank almighty for blessing us with his grace and taking our endeavour to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, **Dr. Abhay Kumar Agarwal** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We extend our sincere thanks to our respected head of the department **Dr. A.K. Malviya**, for allowing us to use the facilities available.

We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank our friends for the support and encouragement that they have given us during the course of our work.

Submitted By -

Aman Kumar Singh (14207)
Aman Sonkar (14208)
Ashutosh Rana (14215)
Rahul Kumar Gupta (14236)

ABSTRACT

This project aims at predicting stock market prices by using financial news, analyst opinions and quotes in order to improve quality of output. It proposes a novel method for the prediction of stock market closing price. Many researchers have contributed in this area of chaotic forecast in their ways. Fundamental and technical analysis are the traditional approaches so far. It uses a popular way to identify unknown and hidden patterns in data which is used for share market prediction. A multi-layered feed-forward network is built by using combination of data and textual mining.

The purpose of this project is to comparatively analyze the effectiveness of prediction algorithms on stock market data and get general insight on this data through visualization to predict future stock behavior and value at risk for each stock. The project encompasses the concept of Data Mining and Statistics. This project makes heavy use of Numpy, Pandas and Data Visualization Libraries.

This is an attempt to determine the market news, analyst recommendations in combination with the historical quotes which can efficiently help in the calculation of the closing index for a given trading day.

INDEX

LIST OF FIGURES

1. INTRODUCTION

1.1. PROJECT OVERVIEW

"Stock Market Analysis and Prediction" is the project on technical analysis, visualization and prediction using data provided by **Morning Star**. It uses data from the stock market, particularly some giant technology stocks and others. The Project used pandas to get stock information, visualized different aspects of it, and finally looked at a few ways of analyzing the risk of a stock, based on its previous performance history. It predicted future stock prices through **Monte Carlo Simulation** method.

The purpose of this project is to comparatively analyze the effectiveness of prediction algorithms on stock market data and get general insight on this data through visualization to predict future stock behavior and value at risk for each stock. The project encompasses the concept of Data Mining and Statistics. This project makes heavy use of Numpy, Pandas and Data Visualization Libraries.

1.2. PROBLEM STATEMENT

- This project is to collect the stock information for some previous years and then accordingly predict the results for the predicting what would happen next.
- In this project, we-are trying to review the possibility to apply two-known techniques which is monte carlo simulation and data-mining in stock market prediction. Extract useful information from a huge amount of data set and use data mining which is able to predict future trends and behaviors through monte carlo simulation. Therefore, combining both these methods could make the prediction much suitable and reliable.
- The most important for predicting stock market prices is monte carlo technique because it is able to learn nonlinear-mappings between inputs and outputs.
- It mainly focuses on monte-carlo simulation technique to predict future prices.

1.3. OBJECTIVES

The objective was to build a system capable of the following tasks:

- **Collecting fundamental and technical data from the internet:** The system should be able to crawl specific websites to extract fundamental data like news articles and analyst recommendations. Furthermore, it should be able to collect technical data in the form of historical share prices.
- **Simulating trading strategies:** The system should offer ways to specify and simulate fundamental and technical trading strategies. Additionally, combining the two approaches should be possible.
- **Evaluating and visualizing trading strategies:** The system should evaluate and visualize the financial performance of the simulated strategies. This allows a comparison to be made between technical, fundamental and the combined approaches.

1.4. SCOPE

- Analysis of stocks using data mining will be useful for new investors to invest in stock market based on the various factors considered by the software.
- Stock market includes daily activities like sensex calculation, exchange of shares. The exchange provides an efficient and transparent market for trading in equity, debt instruments and derivatives.

1.5. SYSTEM FEATURES

The main feature of our project is that it helps to determine the information on previous stock prices and their behavior with change in time. Our system is capable of training the new data taking reference to previously trained data. The computed or analyzed data will be represented in various diagram such as bar graphs and real-time graphs. The rich variety of on-line information and news make it an attractive resource from which one can get data. Stock market predictions can be aided by data mining and analysis of such financial information.

2. FEASIBILITY ANALYSIS

A feasibility study is a preliminary study which investigates the information of prospective users and determines the resources requirements, costs, benefits and feasibility of proposed system. A feasibility study takes into account various constraints within which the system should be implemented and operated. In this stage, the resource needed for the implementation such as computing equipment, manpower and costs are estimated. The estimated are compared with available resources and a cost benefit analysis of the system is made. The feasibility analysis activity involves the analysis of the problem and collection of all relevant information relating to the project. The main objectives of the feasibility study are to determine whether the project would be feasible in terms of economic feasibility, technical feasibility and operational feasibility and schedule feasibility or not. It is to make sure that the input data which are required for the project are available. Thus, we evaluated the feasibility of the system in terms of the following categories:

- Technical feasibility
- Operational feasibility
- Economic feasibility
- Schedule feasibility

2.1. TECHNICAL FEASIBILITY

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs etc. A number of issues have to be considered while doing a technical analysis; understand the different technologies involved in the proposed system. Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system. Is the required technology available? Our system "Stock Market Analysis and Prediction" is technically feasible since all the required tools are easily available. Although all tools seems to be easily available there are challenges too.

2.2. OPERATIONAL FEASIBILITY

Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? The proposed was to make a simplified web application. It is simpler to operate and can be used in any webpages. It is free and not costly to operate.

2.3. ECONOMIC FEASIBILITY

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could increase improvement in product quality, better decision making, and timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information. This is a web-based application. Creation of application is not costly.

2.4. SCHEDULE FEASIBILITY

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed in a given period of time using some methods like payback period. Schedule feasibility is a measure how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some project is initiated 10 with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable. The application development is feasible in terms of schedule.

3. REQUIREMENT DEFINITION

3.1. FUNCTIONAL REQUIREMENTS

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

- System should be able to process new stock data accessed from morning star
- System should be able to analyze data and classify each attribute polarity

3.2. NON-FUNCTIONAL REQUIREMENTS

3.2.1 **Reliability**: The reliability of the product will be dependent on the accuracy of the data-date of purchase, how much stock was purchased, high and low value range as well as opening and closing figures. Also, the stock data used in the training would determine the reliability of the software.

3.2.2 **Security:** The user will only be able to access the website using his login details and will not be able to access the computations happening at the back end.

3.2.3 **Maintainability:** The maintenance of the product would require training of the software by recent data so that there commendations are up to date. There is no updation of any data with recent values.

3.2.4 **Portability:** The website is completely portable and the recommendations completely trustworthy as the data is dynamically updated.

3.2.5 **Interoperability:** The interoperability of the website is very high because it synchronize all the data with the morning star server.

4. TECHNOLOGIES USED

4.1. FRONT-END

Technologies used in front-end are as follows :

- HTML
- CSS
- JavaScript
- Bootstrap

4.1.1. Hyper-text Markup Language (HTML)

- HTML is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
- HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` introduce content into the page directly. Others such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements.
- HTML can embed programs written in a scripting language such as JavaScript.

4.1.2. Cascading Style Sheet (CSS)

- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.
- The CSS specifications are maintained by the World Wide Web Consortium (W3C). The W3C operates a free CSS validation service for CSS documents.

4.1.3. Bootstrap

- Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.
- Bootstrap is the second most-starred project on GitHub,

4.1.4. JavaScript

- JavaScript often abbreviated as JS, is a high level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype based and multi paradigm. Alongside HTML and CSS. JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.
- As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.
- Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets. Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.
- The advent of Ajax returned JavaScript to the spotlight and brought more professional programming attention. The result was a proliferation of comprehensive frameworks and libraries, improved JavaScript programming practices, and increased usage of JavaScript outside Web browsers, as seen by the proliferation of Server-side JavaScript platforms.
- JavaScript typically relies on a run-time environment to provide objects .

4.2. BACK-END

Technologies used in back-end are as follows:

- Python
- Flask
- Pandas

4.2.1. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

4.2.2. Flask

- Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc.
- Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.
- Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.
- It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.
- Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application. Some of the popular Flask extensions are discussed later in the tutorial.
- Flask is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

4.2.3. Pandas

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze.

Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas

- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

4.2.4. Numpy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful Fourier transform, and random number capabilities

4.3. Application Programming Interface (API)

We have used an easy-to-use Python library for accessing the stock market datasets source called Morning Star. The API class provides access to the entire stock data API methods. Each method can accept various parameters and return responses. When we invoke an API method most of the time returned back to us will be a Morning Star model class instance. This will contain the data returned from Morning Star which we can then use inside our application.

4.3.1. Morning Star

Morningstar, Inc. is an investment research and investment management firm headquartered in Chicago, Illinois, United States.

Founder Joe Mansueto initially had the idea for Morningstar in 1982 while reviewing mutual fund annual reports he had requested from several prominent fund managers. However, it was only after a year working as a stock analyst for Harris Associates, seeing the fund industry and potential competitors up close, that he was convinced that the opportunity was there. Morningstar was subsequently founded in 1984 from his one-bedroom Chicago apartment with an initial investment of US$80,000. The name Morningstar is taken from the last sentence in Walden, a book by Henry David Thoreau; "the sun is but a morning star".

In July 1999, Morningstar accepted an investment of US$91 million from SoftBank in return for a 20 percent stake in the company. The two companies had formed a joint venture in Japan the previous year.

Morningstar's initial public offering occurred on May 3, 2005, with 7,612,500 shares at $18.50 each. The manner in which Morningstar went public is notable. They elected to follow Google's footsteps and use the OpenIPO method rather than the traditional method. This allowed individual investors to bid on the price of the stock, and allowed all investors equal access.

As of February 2015, Joe Mansueto owned approximately 55% of the outstanding shares in Morningstar.

5. SYSTEM DESIGN AND ARCHITECTURE

5.1. Use-Case Diagram

The use-case diagram for the project is as follows:

Fig-5.1: Use-case Diagram

5.2. System-Flow Diagram

The system flow diagram is as follows:

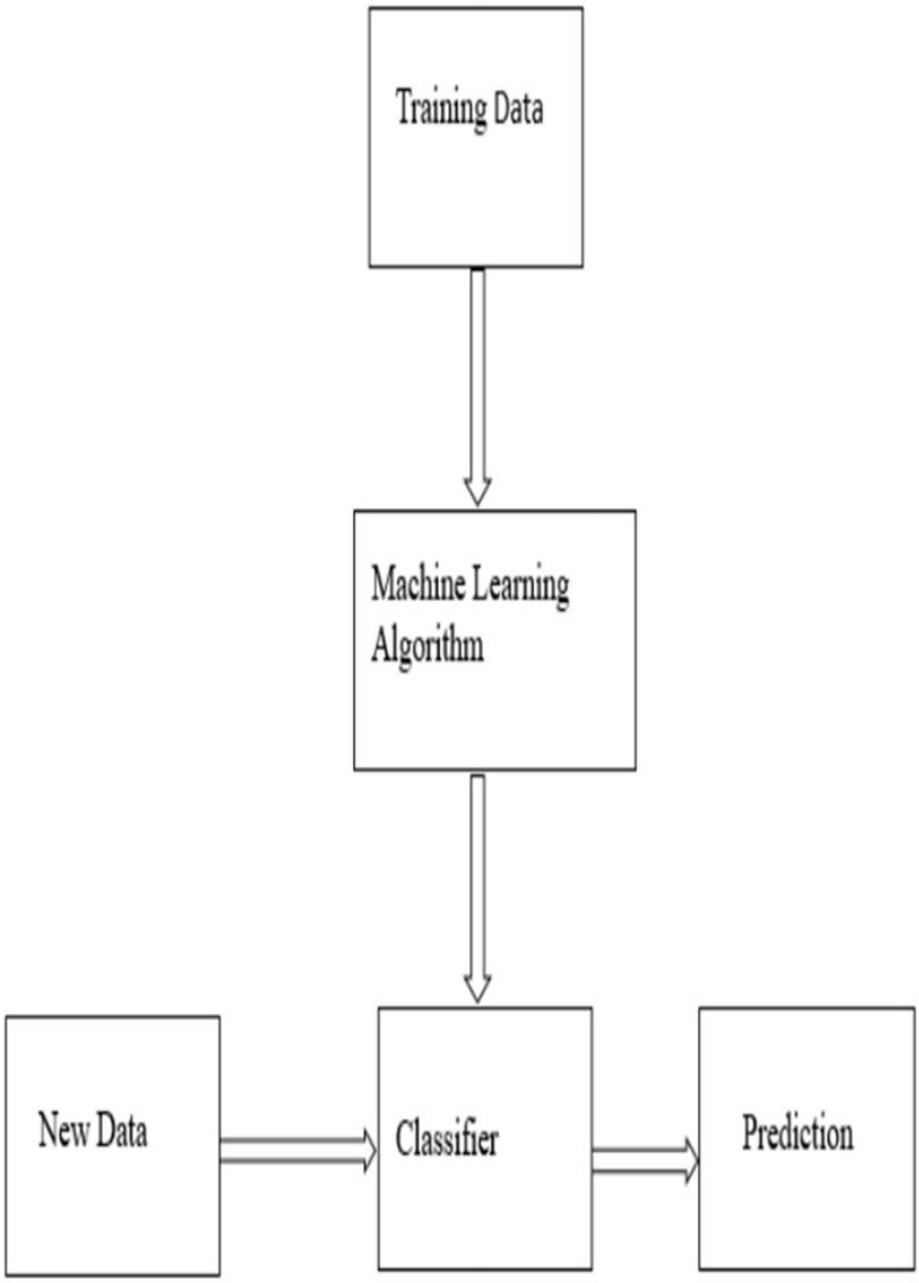


Fig.5.2: System-flow Diagram

5.3. Activity-Flow Diagram

The activity flow diagram is as follows:

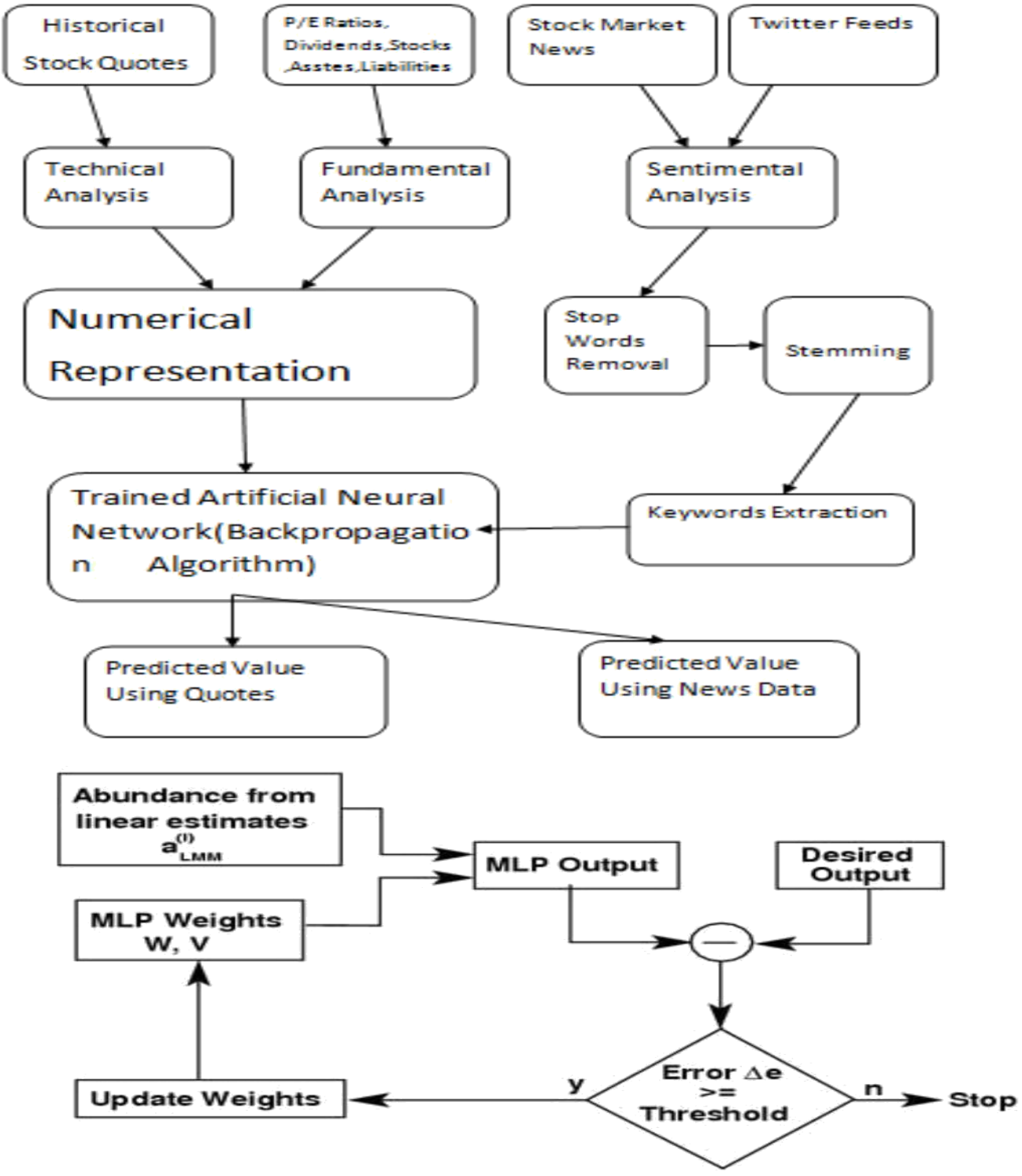


Fig.5.3: Activity-flow Diagram

5.4. Data Flow Diagram

- Data Flow diagram given below describes the flow of data in the system.
- It also tells about the data sources and data storage.

The data-flow diagram for the project is as follows:

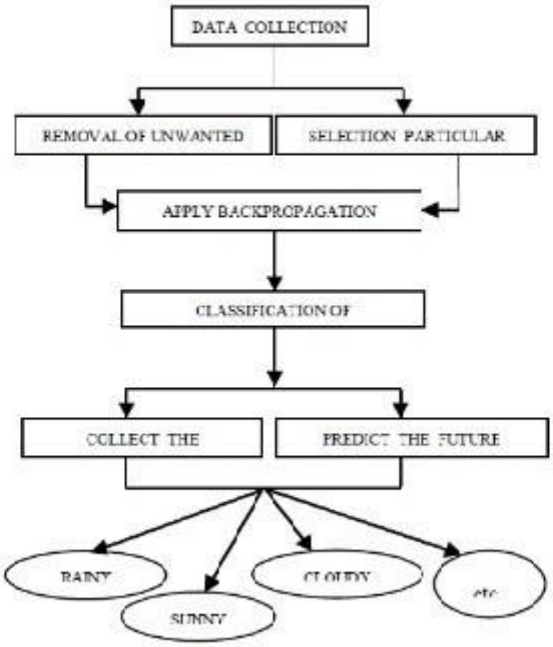


Fig. 5.4: Data-flow Diagram

5.5. FRAMEWORK OF STOCK ANALYSIS AND PREDICTION

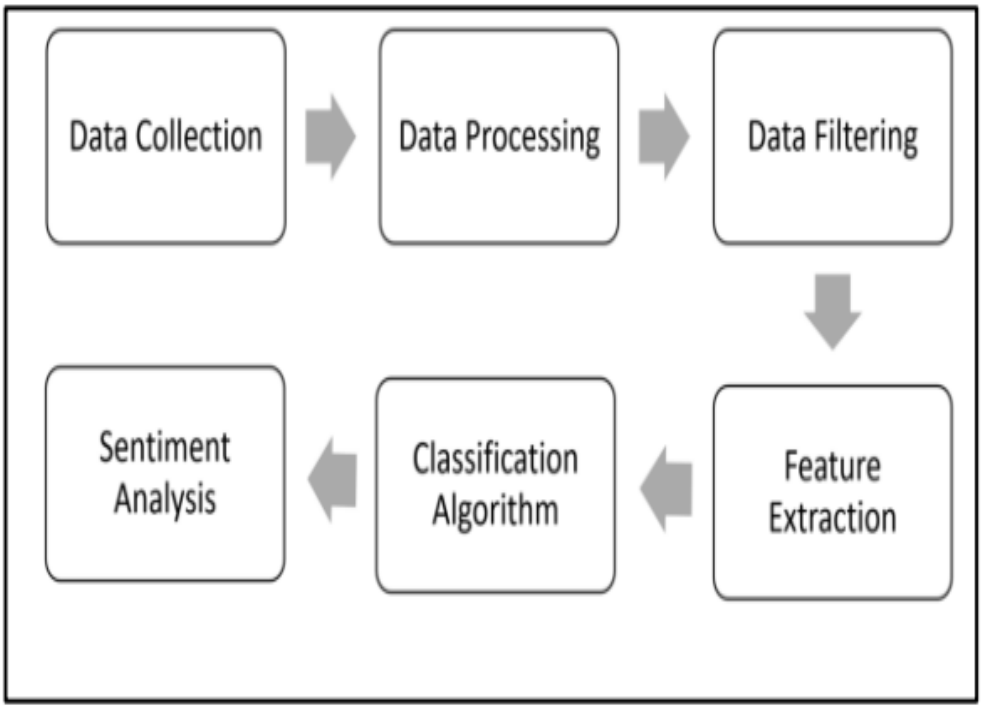


Fig.5.5: Framework of Analysis

5.5.1.Data Collection

- Data in the form of raw data is retrieved by using the Morning Star library which provides a package for real time stock streaming API. The API doesn't requires any one to register a developer account with Morning Star and fill in parameters such as consumerKey, consumerSecret, accessTokenaccess, and TokenSecret. This API allows to get all random data or filter data by using keywords. Filters supports to retrieve information which match a specific criterion defined by the developer. We used this to retrieve information related to specific keywords which are taken as input from users. Initially, we set at least set an application name and mode. We execute the program in local mode instead of cluster. Then,

input array of keywords is provided as an argument to Streaming Context “ssc” using “sc” where “sc” is spark context.

- For example, on inputting multiple keywords like, 'Google', 'Amazon', 'Apple', the output we obtained from 15 seconds’ window time was the live stream of stock data associated with these keywords. Only caveat of using filters is that famous keyword attributes have more information compared to niche words which makes it difficult to get data for niche specific keywords.

5.5.2. Data Processing

- Data processing involves Tokenization which is the process of splitting the raw data into individual words called tokens. Tokens can be split using whitespace or punctuation characters. It can be unigram or bigram depending on the classification model used. The bag of words model is one of the most extensively used model for classification. It is based on the fact of assuming text to be classified as a bag or collection of individual words with no link or interdependence.
- The next step in data processing is normalization by conversion of into attributes are normalized by converting it to lowercase which makes its comparison with an dictionary easier.

5.5.3. Data Filtering

- An attribute acquired after data processing still has a portion of raw information in it which we may or may not find useful for our application. Thus, these raw data are further filtered by removing stop words, numbers and punctuations. Stop words: For example, raw data contain many stop words which are extremely common words and holds no additional information.
- These words serve no purpose and this feature is implemented using a list stored in stopfile.dat. We then compare each word in a tweet with this list and delete the words matching the stop list as shown in fig.

- This helps to reduce the clutter from the stock stream.
- **Stemming:** It is the process of reducing derived words to their roots. Example includes words like “fish” which has same roots as “fishing” and “fishes”. The library to use stemming is Stanford NLP which also provides various algorithms such as porter stemming. In our case, we have not employed any stemming algorithm due to time constraints.

5.5.4. Feature Extraction

- TF-IDF is a feature vectorization method used in text mining to find the importance of a term to a document in the corpus. Feature extraction involves “mlib” library of Apache Spark. The recommended API is the Data Frame based API.
- This feature is useful for a case where we need to find trending topics or to create word clouds. However, this project is more focused towards finding sentiment in twitter streams so TF-IDF is not implemented.

5.5.5. Stock Analysis

- Stock analysis is done by using custom algorithm which finds polarity as below.
- **Monte Carlo Simulation:** For definite pattern we used a simple algorithm of monte carlo simulation in stock data. For both, positive and negative results, different lists were made. Next step is to compare every result in an attribute against both these list. If the current result matches an attribute in positive list, then a score of 1 is incremented and if a negative word is found then it is decremented. More positive results lead to higher score. However, Standford. NLP can be used to predict accurate stock analysis which provide complex algorithms to predict.

6. Methodology

6.1 MONTE CARLO SIMULATION

6.1.1. Introduction

Monte Carlo simulation is a computerized mathematical technique that allows people to account for risk in quantitative analysis and decision making. The technique is used by professionals in such widely disparate fields as finance, project management, energy, manufacturing, engineering, research and development, insurance, oil & gas, transportation, and the environment.

Monte Carlo simulation furnishes the decision-maker with a range of possible outcomes and the probabilities they will occur for any choice of action. It shows the extreme possibilities—the outcomes of going for broke and for the most conservative decision—along with all possible consequences for middle-of-the-road decisions. The technique was first used by scientists working on the atom bomb; it was named for Monte Carlo, the Monaco resort town renowned for its casinos. Since its introduction in World War II, Monte Carlo simulation has been used to model a variety of physical and conceptual systems.

6.1.2. Working

Monte Carlo simulation performs risk analysis by building models of possible results by substituting a range of values—a probability distribution—for any factor that has inherent uncertainty. It then calculates results over and over, each time using a different set of random values from the probability functions. Depending upon the number of uncertainties and the ranges specified for them, a Monte Carlo simulation could involve thousands or tens of thousands of recalculations before it is complete. Monte Carlo simulation produces distributions of possible outcome values.

By using probability distributions, variables can have different probabilities of different outcomes occurring. Probability distributions are a much more realistic way of describing uncertainty in variables of a risk analysis.

Common probability distributions include:

Normal

Or "bell curve." The user simply defines the mean or expected value and a standard deviation to describe the variation about the mean. Values in the middle near the mean are most likely to occur. It is symmetric and describes many natural phenomena such as people's heights. Examples of variables described by normal distributions include inflation rates and energy prices.

Lognormal

Values are positively skewed, not symmetric like a normal distribution. It is used to represent values that don't go below zero but have unlimited positive potential. Examples of variables described by lognormal distributions include real estate property values, stock prices, and oil reserves.

Uniform

All values have an equal chance of occurring, and the user simply defines the minimum and maximum. Examples of variables that could be uniformly distributed include manufacturing costs or future sales revenues for a new product.

Triangular

The user defines the minimum, most likely, and maximum values. Values around the most likely are more likely to occur. Variables that could be described by a triangular distribution include past sales history per unit of time and inventory levels.

PERT

The user defines the minimum, most likely, and maximum values, just like the triangular distribution. Values around the most likely are more likely to occur. However values between the most likely and extremes are more

likely to occur than the triangular; that is, the extremes are not as emphasized. An example of the use of a PERT distribution is to describe the duration of a task in a project management model.

Discrete

The user defines specific values that may occur and the likelihood of each. An example might be the results of a lawsuit: 20% chance of positive verdict, 30% change of negative verdict, 40% chance of settlement, and 10% chance of mistrial.

During a Monte Carlo simulation, values are sampled at random from the input probability distributions. Each set of samples is called an iteration, and the resulting outcome from that sample is recorded. Monte Carlo simulation does this hundreds or thousands of times, and the result is a probability distribution of possible outcomes. In this way, Monte Carlo simulation provides a much more comprehensive view of what may happen. It tells you not only what could happen, but how likely it is to happen.

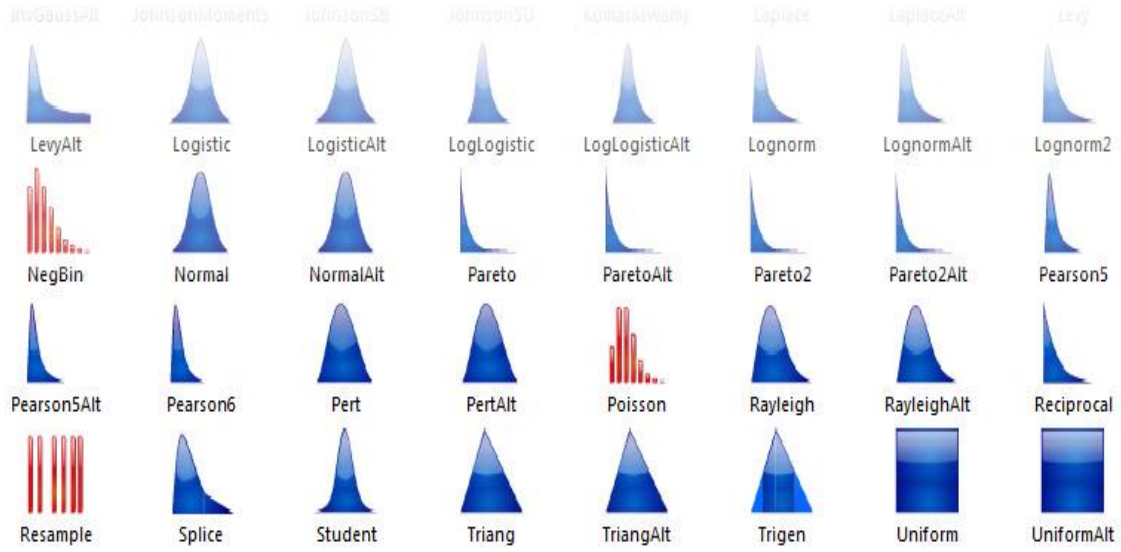


Fig-6.1: Distributions in Monte Carlo

7. WALKTHROUGH OF THE PROJECT

7.1 Meaning

A **walkthrough** is a form of software peer review "in which a designer or programmer leads members of the development team and other interested parties go through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems".

"Software product" normally refers to some kind of technical document. As indicated by the IEEE definition, this might be a software design document or program source code, but use cases, business process definitions, test case specifications, and a variety of other technical documentation may also be walked through.

A walkthrough differs from software technical reviews in its openness of structure and its objective of familiarization. It differs from software inspection in its ability to suggest direct alterations to the product reviewed, its lack of a direct focus on training and process improvement, and its omission of process and product measurement.

7.2 Process

A walkthrough may be quite informal, or may follow the process detailed in IEEE 1028 and outlined in the article on software reviews.

7.3 Objectives

A walkthrough is normally organized and directed by the author of the technical document. Any combination of interested or technically qualified personnel (from within or outside the project) may be included as seems appropriate.

In general, a walkthrough has one or two broad objectives:

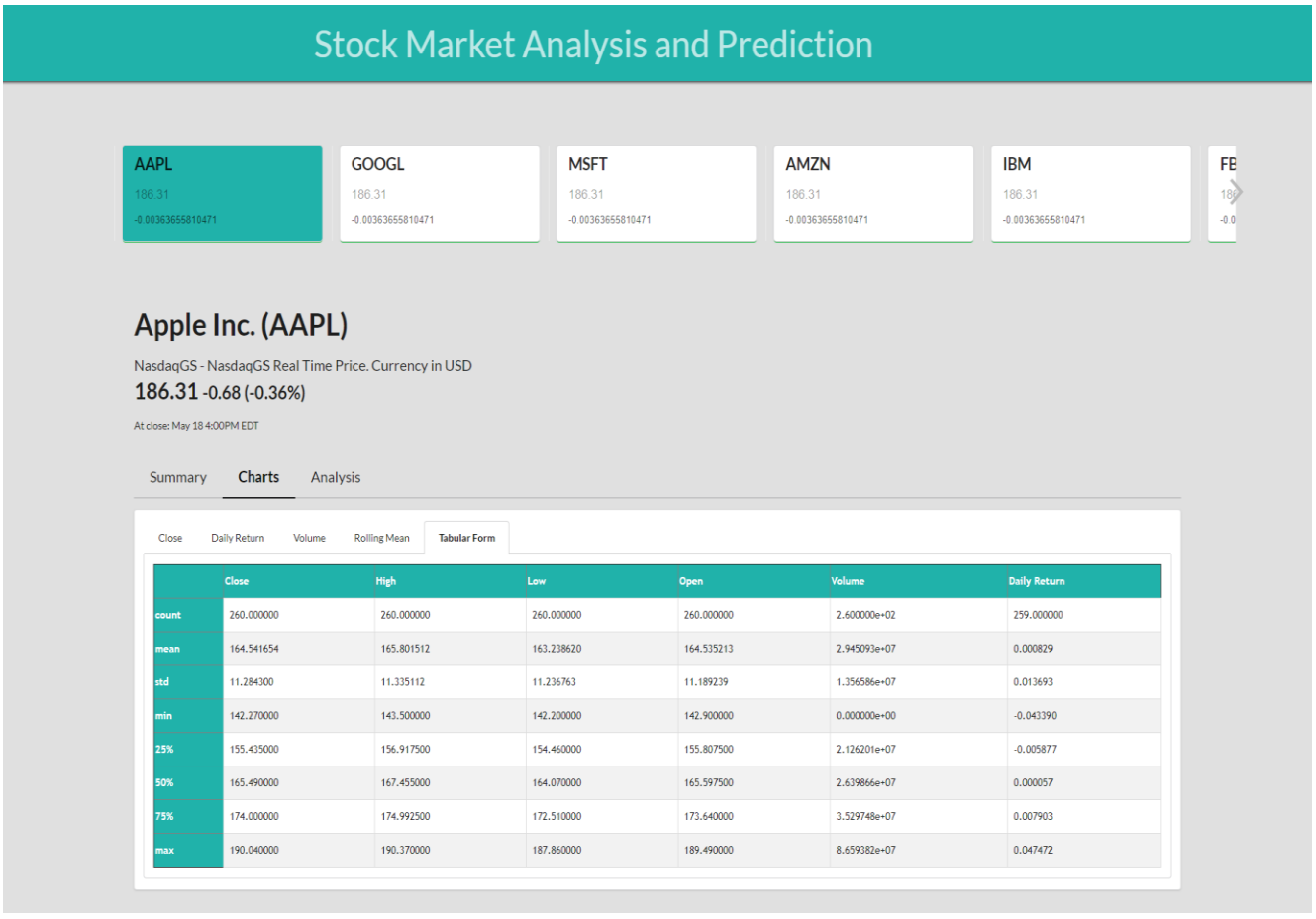
- To gain feedback about the technical quality or content of the document; and/or
- To familiarize the audience with the content.

7.4 Home Page

The homepage of the project looks as given below in the diagram. It consists of stock information of various companies which are accessed from morning star.

It contains various companies as modules:

- Apple
- Google
- Microsoft
- Amazon
- IBM etc.



	Close	High	Low	Open	Volume	Daily Return
count	260.000000	260.000000	260.000000	260.000000	2.600000e+02	259.000000
mean	164.541654	165.801512	163.238620	164.535213	2.945093e+07	0.000829
std	11.284300	11.335112	11.236763	11.189239	1.356586e+07	0.013693
min	142.270000	143.500000	142.200000	142.900000	0.000000e+00	-0.043390
25%	155.435000	156.917500	154.460000	155.807500	2.126201e+07	-0.005877
50%	165.490000	167.455000	164.070000	165.597500	2.639866e+07	0.000057
75%	174.000000	174.992500	172.510000	173.640000	3.529748e+07	0.007903
max	190.040000	190.370000	187.860000	189.490000	8.659382e+07	0.047472

Fig-7.1: Home Page

7.5 Contents of a Module

A particular module contains the following components:

- Summary
- Charts
- Analysis

Summary

The summary contains all the information about a particular company. All the attributes of a company are defined here. The past history, its evolvement over the period of time and its time of existence in the market since date are highlighted

The summary represents the preamble of any company in stock market.

Charts

Charts are used to describe companies' attributes in terms of following;

- Close
- Daily Return
- Volume
- Rolling Mean
- Standard Deviation etc.

Analysis

- Analysis of the various attributes are made based on their results.
- These results are used to make the analysis through graphs.
- Various graphs that are used are:
  - Line Graphs
  - Bar Graphs
  - Histograms
  - Plots

7.6 Analysis of various attributes

Close

The close is the end of a trading session in the financial markets when the markets close. It can also refer to the process of exiting a trade or the final procedure in a financial transaction in which contract documents are signed and recorded.

The close for Apple stock has been given below:

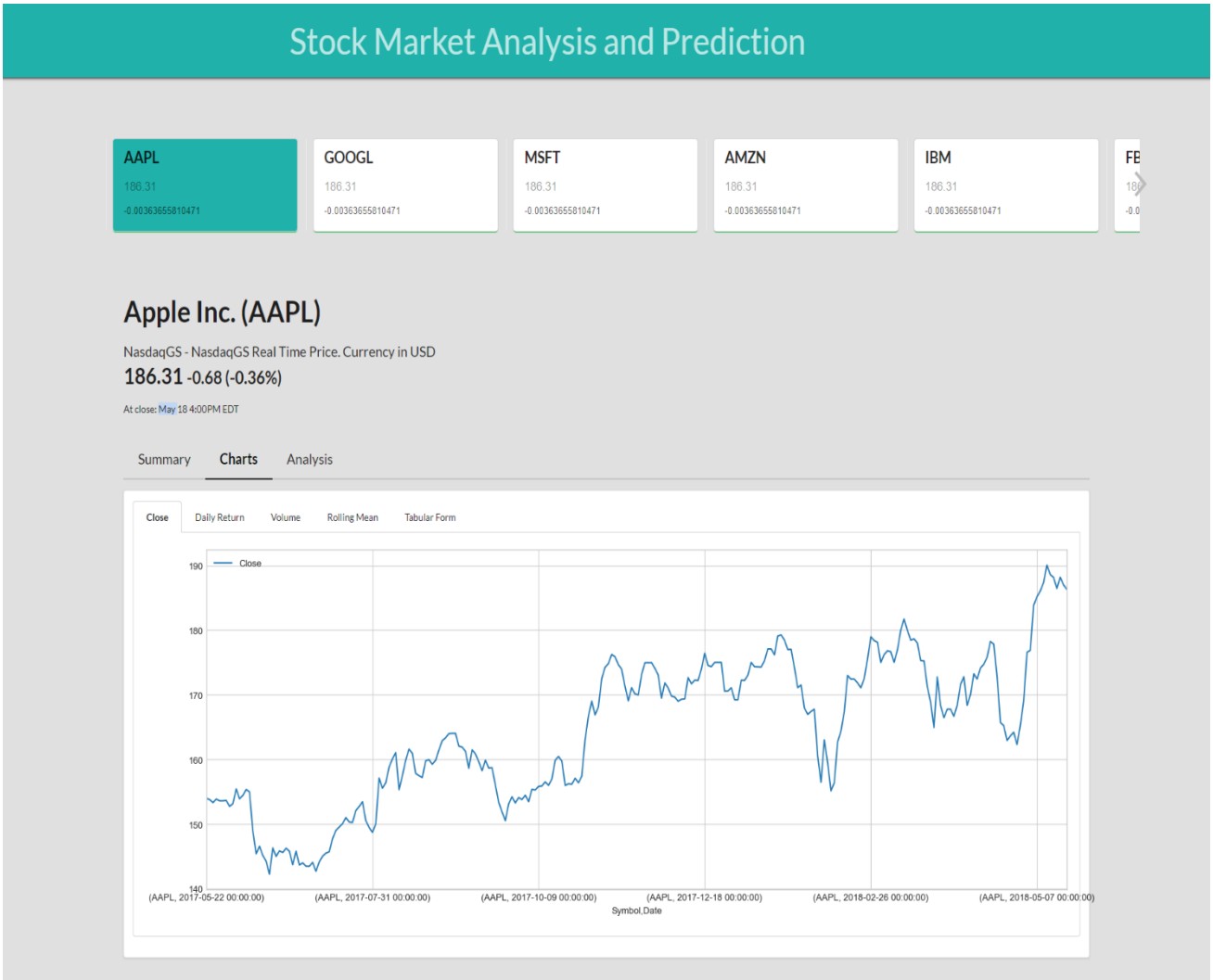


Fig-7.2: Close

Daily Return

The intraday return is one of the two components of the total daily return generated by a stock. Intraday return measures the return generated by a stock during regular trading hours, based on its price change from the opening of a trading day to its close. Intraday return and overnight return together constitute the total daily return from a stock, which is based on the price change of a stock from the close of one trading day to the close of the next trading day.

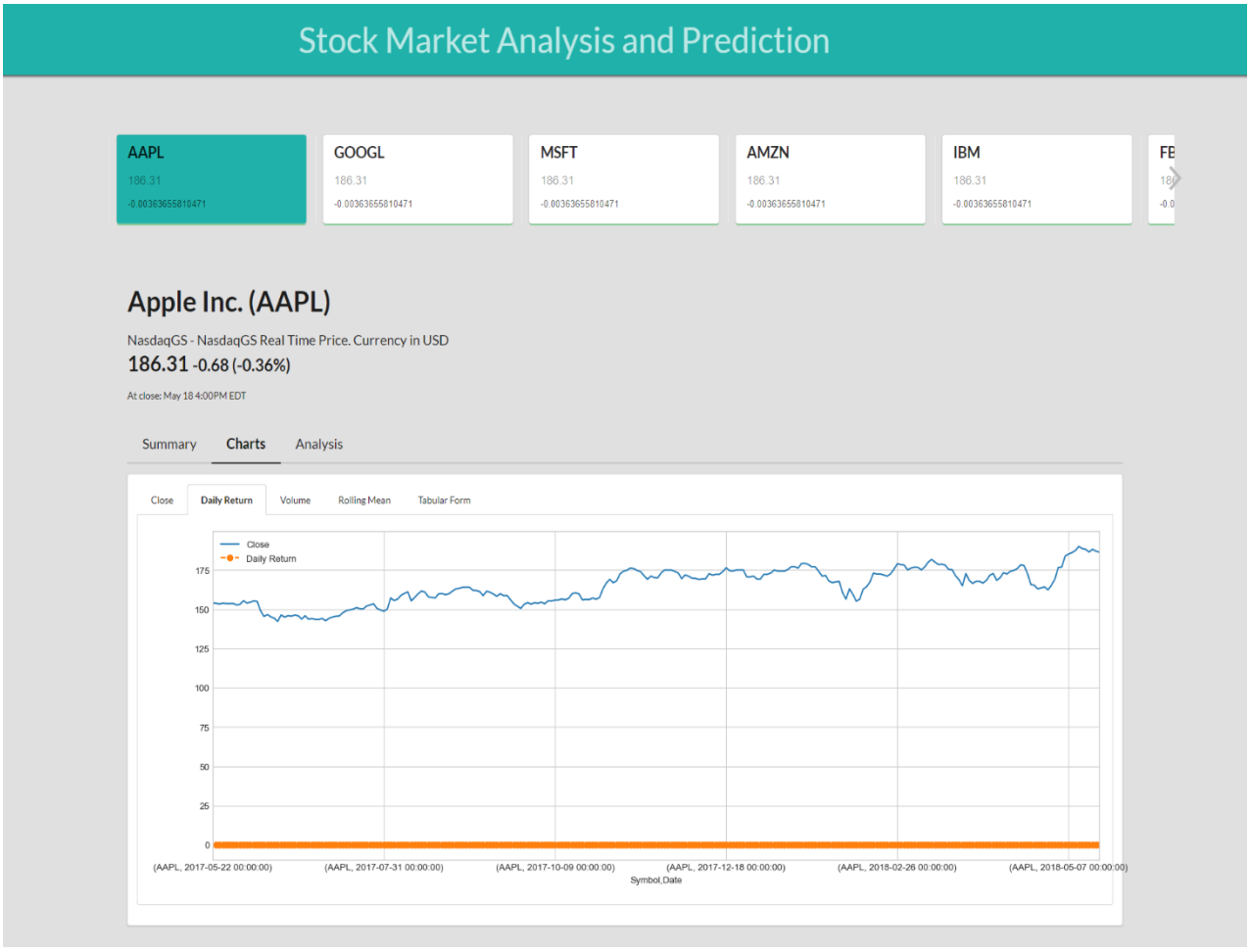


Fig-7.3: Daily Return

Volume

It is simply the amount of shares that trade hands from sellers to buyers as a measure of activity. If a buyer of a stock purchases 100 shares from a seller then the volume for that period increases by 100 shares based on that transaction. Volume is an important indicator in technical analysis as it used to measure the worth of a market move. If the markets have made strong price move either up or down the perceived strength of that move depends on the volume for that period. The higher the volume during that price move the more significant the move.

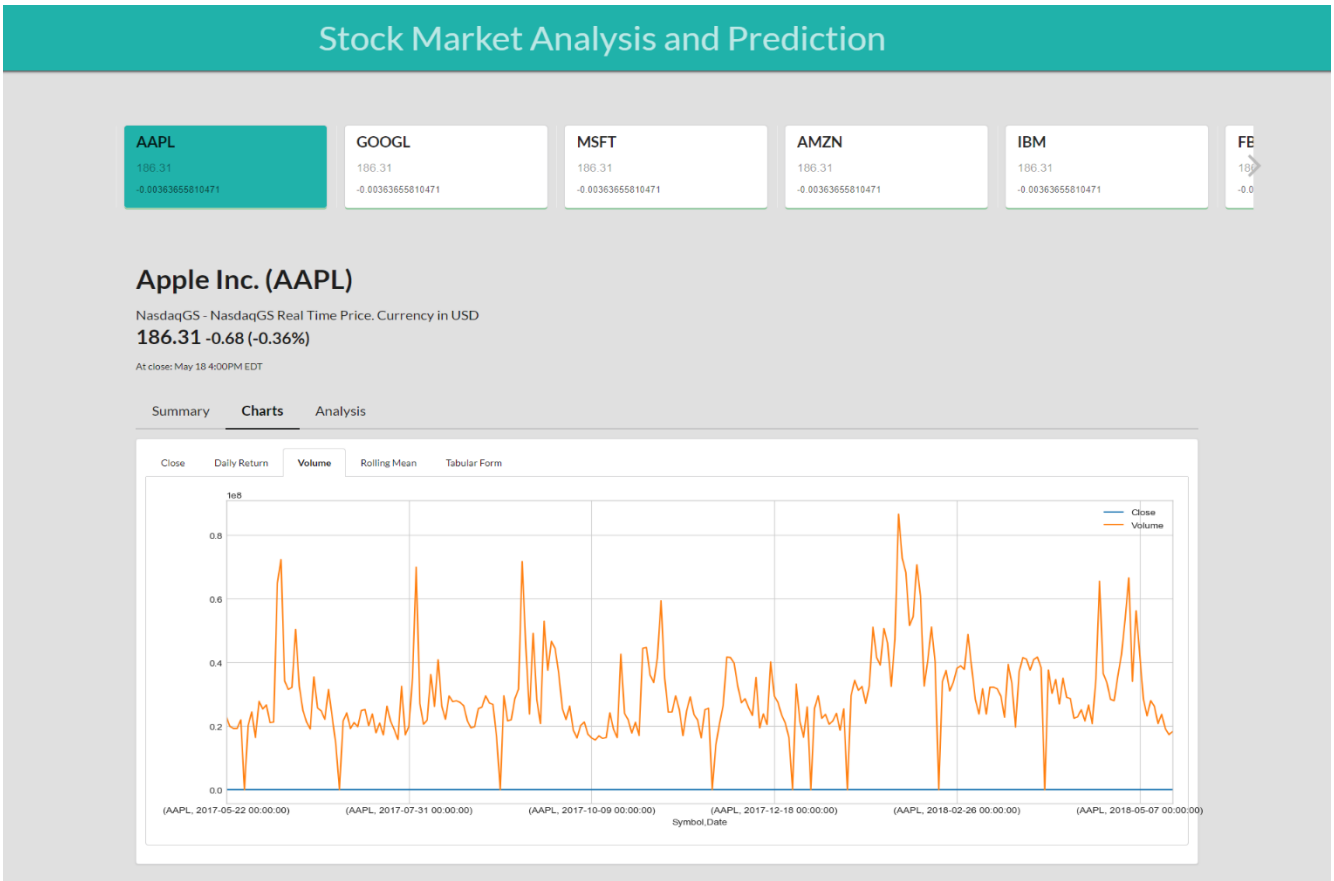


Fig-7.4: Volume

Rolling Mean

In stock applications a rolling mean is the unweighted mean of the previous *n* data. However, in science and engineering the mean is normally taken from an equal number of data on either side of a central value. This ensures that variations in the mean are aligned with the variations in the data rather than being shifted in time. An example of a simple equally weighted running mean for a n-day sample of closing price is the mean of the previous *n* days' closing prices.

If those prices are Pm, Pm-1, Pm-2, ….Pm-(n-1), then

The formula is

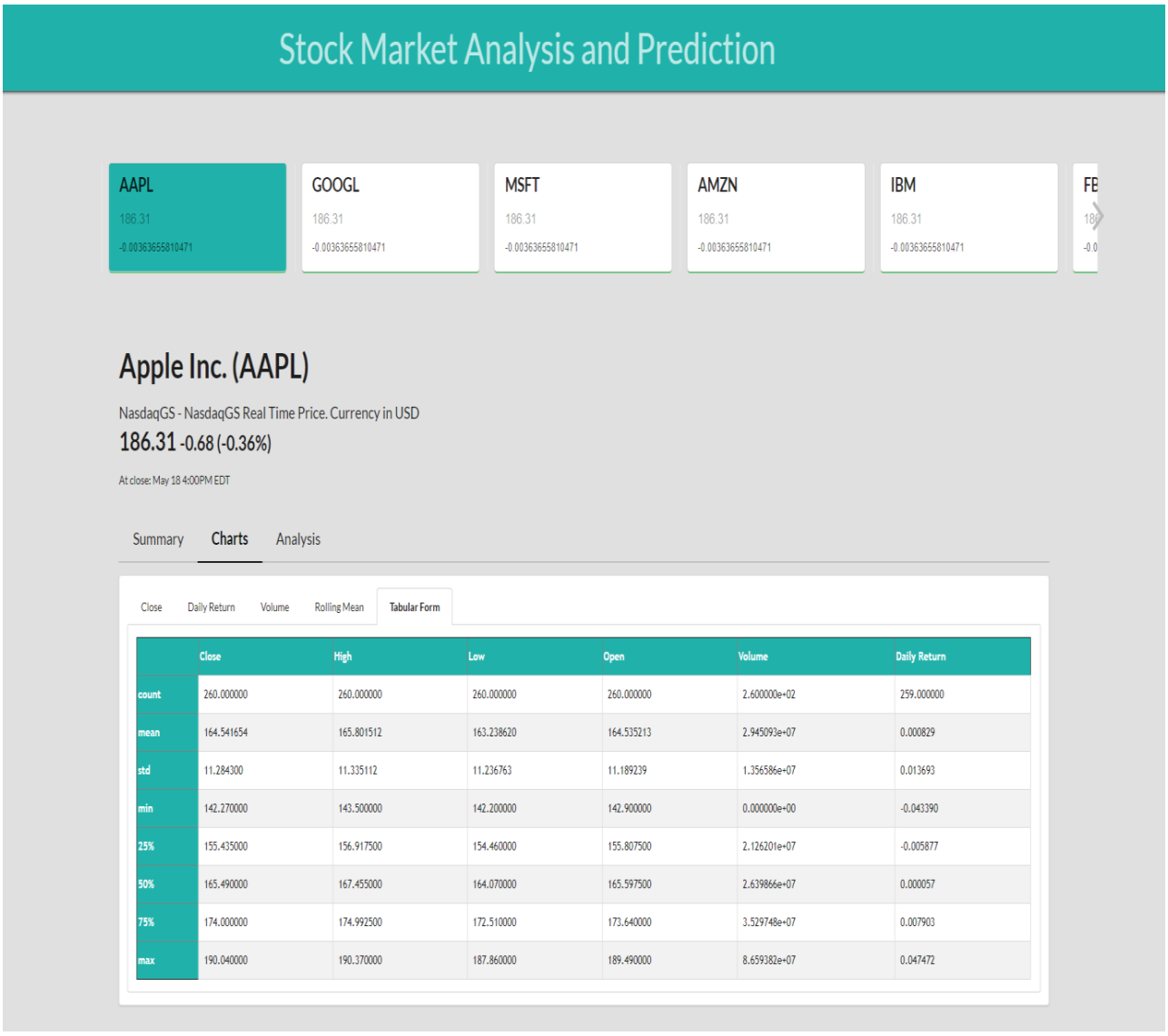
$$\begin{aligned}\overline{p}_{\mathrm{SM}} &= \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n} \\ &= \frac{1}{n}\sum_{i=0}^{n-1} p_{M-i}\end{aligned}$$

And a simple plot for apple stock is as given below:

Fig-7.5 Rolling Mean

Tabular Form

The tabular describes the various attributes discussed so far in terms of a table. It gives various discrete statistical parameter of each of these attributes. The parameters include count, mean, minimum, maximum, standard deviation, percentages etc.



	Close	High	Low	Open	Volume	Daily Return
count	260.000000	260.000000	260.000000	260.000000	2.600000e+02	259.000000
mean	164.541654	165.801512	163.238620	164.535213	2.945093e+07	0.000829
std	11.284300	11.335112	11.236763	11.189239	1.356586e+07	0.013693
min	142.270000	143.500000	142.200000	142.900000	0.000000e+00	-0.043390
25%	155.435000	156.917500	154.460000	155.807500	2.126201e+07	-0.005877
50%	165.490000	167.455000	164.070000	165.597500	2.639866e+07	0.000057
75%	174.000000	174.992500	172.510000	173.640000	3.529748e+07	0.007903
max	190.040000	190.370000	187.860000	189.490000	8.659382e+07	0.047472

Fig-7.6: Tabular form of the Attributes

8. TESTING

8.1 Unit Testing

- Unit testing is performed for testing modules against detailed design. Inputs to the process are usually compiled modules from the coding process. Each modules are assembled into a larger unit during the unit testing process.
- Testing has been performed on each phase of project design and coding. We carry out the testing of module interface to ensure the proper flow of information into and out of the program unit while testing. We make sure that the temporarily stored data maintains its integrity throughout the algorithm's execution by examining the loc al data structure. Finally, all error-handling paths are also tested.

8.2 System Testing

- We usually perform system testing to find errors resulting from unanticipated interaction between the sub-system and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, series of test cases must be developed which ultimately uncover all the possibly existing errors. Different software techniques can be used for this process. These techniques provide systematic guidance for designing test that
- Exercise the internal logic of the software components,
- Exercise the input and output domains of a program to uncover errors in program function, behavior and performance.
- We test the software using two methods:

  White Box testing: Internal program logic is exercised using this test case design techniques.

Black Box testing: Software requirements are exercised using this test case design techniques.

- Both techniques help in finding maximum number of errors with minimal effort and time.

8.3 Performance Testing

It is done to test the run-time performance of the software within the context of integrated system. These tests are carried out throughout the testing process. For example, the performance of individual module are accessed during white box testing under unit testing.

8.4 Analysis and Result

We collected dataset containing positive and negative data. Those dataset were trained data and was implemented using monte carlo simulation. Before training the classifier unnecessary words, punctuations, meaning less words were cleaned to get pure data. To determine positivity and negativity of raw data we collected data using morning star API. Those data were not stored in database.

Result

After facing a number of errors, successful elimination of those error we have completed our project with continuous effort. At the end of the project the results can be summarized as:

- A user-friendly web based application.
- No expertise is required for using the application.
- Organizations can use the application to visualize product or brand review graphically.

8.5. Testing Plan

Type of Test	Will Test be Performed	Comments/Explanation	Software Component
Requirement Testing	Yes	The tester should look at the project from organization perspective and make sure it defines and reflects what customers have in mind. It should align with client's goals without being biased towards technicalities.	Does input and output meet the requirements, i.e. input output synchronization according to the proposed algorithm.
Unit	Yes	Unit testing is a way by which individual units of source code with associated control data, usage, procedures, and operating procedures, are tested to determine if they are fit for use.	Check to see if the source data can be read by the module and produce the required result.
Integration	Yes	This testing is done after all the groups have been completed to test their relation. This is particularly necessary and important as these modules have been made solely for the purpose of integration with the existing organization cctv.	Check if all the variables data work together in cohesion.
Performance	Yes	Needed to assess various factors which will eventually lead to customer satisfaction.	Full table is being checked with proper input output entries.
Stress	No	The stress of the system is measured as the large number of data crawled is managed.	Several numerical data should be checked simultaneously on net.

8.6. Testing Types

Type of Test	Will Test be Performed	Comments/Explanation	Software Component
Compliance	Yes	Complacency was checked right as the data crawled were all under proper specifications.	Yahoo finance stock price crawler have been into use.
Security	Yes	Security of our predictions have been kept proper i.e pure live data pure algorithms based results.	Check to see if the source data can be read by the module and produce the required result.
Load	No	Though all the data crawled are quite large in number inspite of that load on a system was neutral.	Large data crawled were simultaneously stored also to depressurize the load on the system.
Volume	Yes	Myriad of data have been used i.e. 5k prices of BSE and 2.5k prices of NSE.	Accordingly entries were also made with proper companies code.
Stress	No	The stress of the system is measured as the large number of data crawled is managed.	Several numerical data should be checked simultaneously on net.

9. LIMITATIONS AND FUTUTRE ENHANCEMENT

9.1 Limitations

- The solution has a few limitations which are relevant to the proper functioning of our application.
- Parsing research firms Several notations were being used for the same research firm (e.g. CSFB and CS First Boston). A map was manually created to ensure the different expressions were mapped to the same firm.
- Parsing analyst recommendations: Different research firms tend to use different vocabulary for recommendations. For example, some use use ver-weight or simply buy to suggest a buying opportunity. In order to compare recommendations, all 96 different phrases found in the dataset were manually mapped to the three expressions Buy, Neutral and Sell.
- When working on a large project, small bugs can creep in and easily go unnoticed for some time (e.g. array indices of by one). Particularly when running simulations, the results may be greatly affected and the error may be hard to track down. In order to prevent this to a certain extent, unit tests were written using the JUnit4 framework. The behaviour of all relevant simulation server classes could be checked; when refactoring parts of the server, the behaviour could be revalidated.

9.2 Future Enhancements

From future perspective, we would like to extend this project by implementing some machine learning algorithms for applications like election results, product ratings, movies outcomes and running the project on clusters to expand its functionalities. Moreover, we would like to make a web application for users to input keywords and get analyzed results. In this project, we have worked only with unigram models, but we would like to extend it to bigram and further which will increase linkage between the data and provide accurate sentiment analysis results.

10. REFERENCES

1. Daily Stock Market Forecast from Textual Web Data Wuthrich, B.; Cho, V.; Leung, SPermunetilleke, D.; Sankaran, K.; Zhang, J.; Lam, W.IEEE International Conference on Systems, Man, and Cybernetics, vol.3, pp.2720-2725 vol.3, 11-14 Oct 1998
2. The Apache httpclient library is an open source Java library for working with HTTP.*http://hc.apache.org/httpcomponents-client/*
3. NekoHTML is an open source Java library for fixing HTML.*http://nekohtml.sourceforge.net*
4. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data Ronan Feldman and James Sanger Cambridge University Press, 11 Dec 2006
5. New Trading Systems And Methods Perry J. Kaufman Wiley, 4th Edition, 28 Feb 2005
6. Stock Market Prediction with Backpropagation Networks Freislben, B. Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, vol.604, pp.451-460, 1992
7. An Intelligent Forecasting System of Stock Price Using Neural Networks
8. Baba, N.; Kozaki, M. International Joint Conference on Neural Networks, vol.1, pp.371-377, 1992

11.APPENDIX (Codes with Outputs)

```
# For Data Processing
import numpy as np
import pandas as pd
from pandas import Series, DataFrame

# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline

# For reading stock data from morning star
from pandas_datareader.data import DataReader

# For time stamps
from datetime import datetime

# For division
from __future__ import division

# List of Tech_stocks for Analytics
tech_list = ['AAPL','GOOGL','MSFT','AMZN']

# set up Start and End time for data grab
end = datetime.now()
start = datetime(end.year-1,end.month,end.day)

#For-loop for grabing google finance data and setting as a dataframe
# Set DataFrame as the Stock Ticker
```

```
for stock in tech_list:
    globals()[stock] = DataReader(stock,'morningstar',start,end)

AAPL.head()
GOOGL.head()
MSFT.head()
```

 # Let's see a historical view of the closing price

```
AAPL['Close'].plot(legend=True, figsize=(10,4))
```

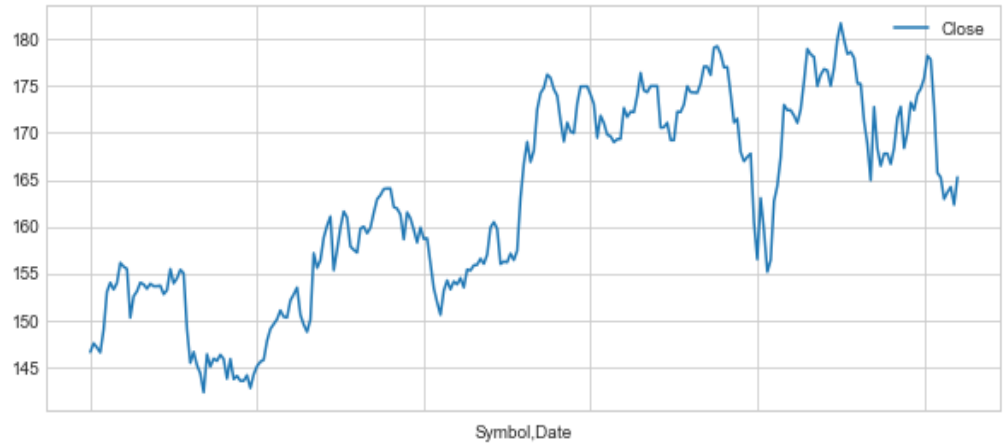


Fig-11.1: Closing Prices

 # Now let's plot the total volume of stock being traded each day over the past year

```
AAPL['Volume'].plot(legend=True, figsize=(10,4))
```

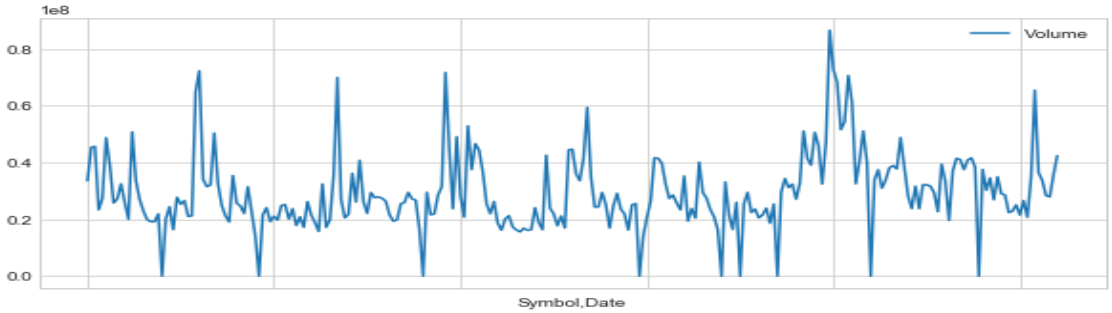


Fig-11.2: Volume

```
AAPL[['Close','MA for 10 days','MA for 20 days','MA for 50 days','MA for 100 days']].plot(subplots=False,figsize=(10,4))
```

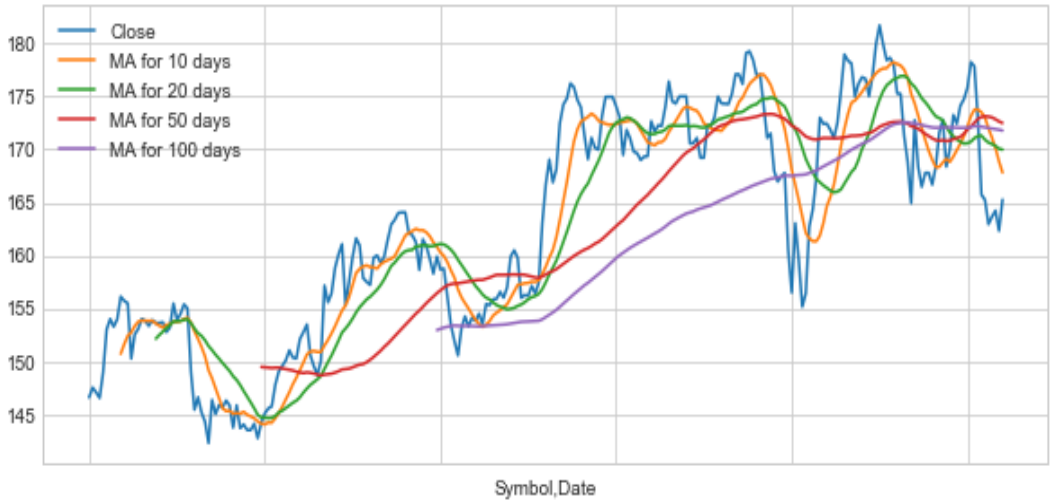


Fig.11.3: Analysis for Apple

#Daily Returns

 # We'll use pct_change to find the percent change for each day

```
AAPL['Daily Return'] = AAPL['Close'].pct_change()
```

 # Lets plot the daily return percentage

```
AAPL['Daily Return'].plot(figsize=(12,4), legend=True, linestyle='--', marker='o')
```

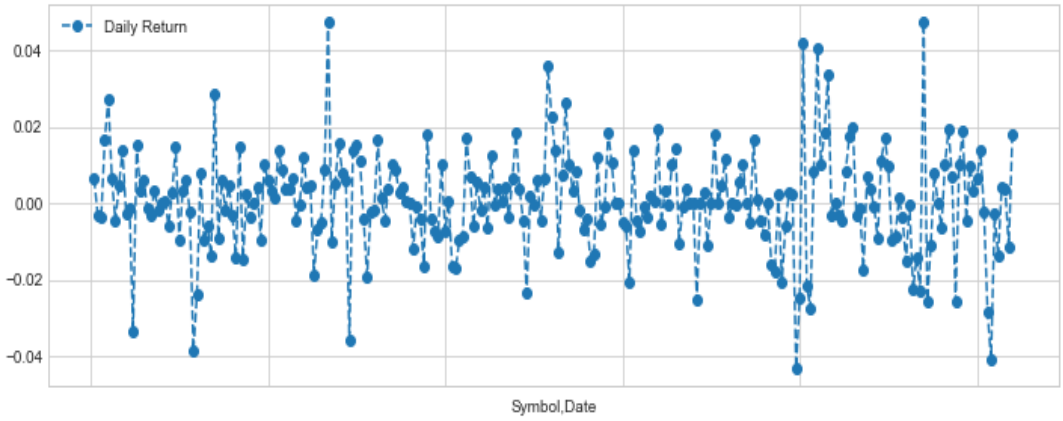


Fig.11.4: Daily Return

 # only with histogram

AAPL['Daily Return'].hist(bins=100)

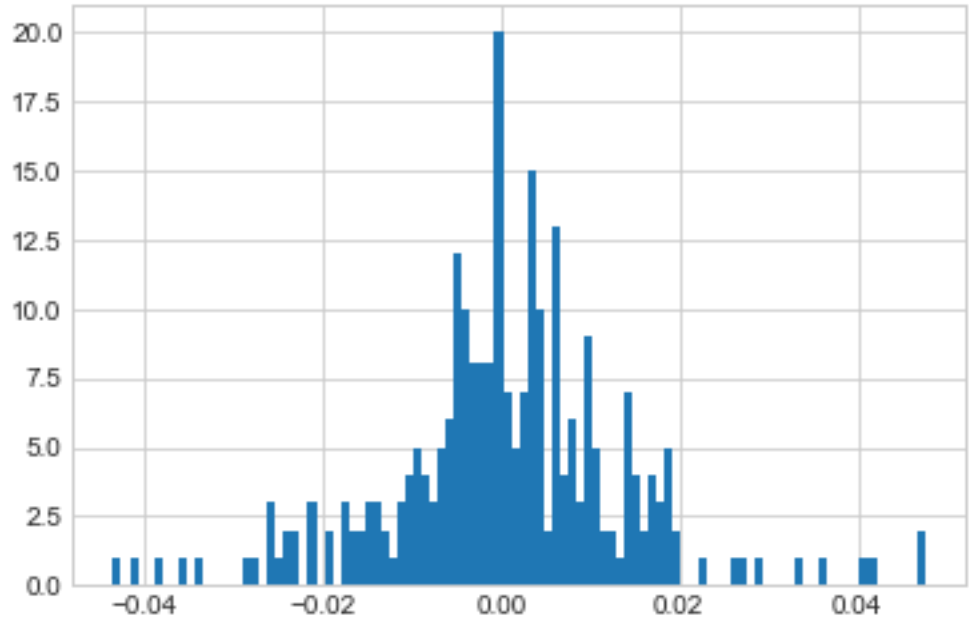


Fig-11.5: Histogram for Daily Return

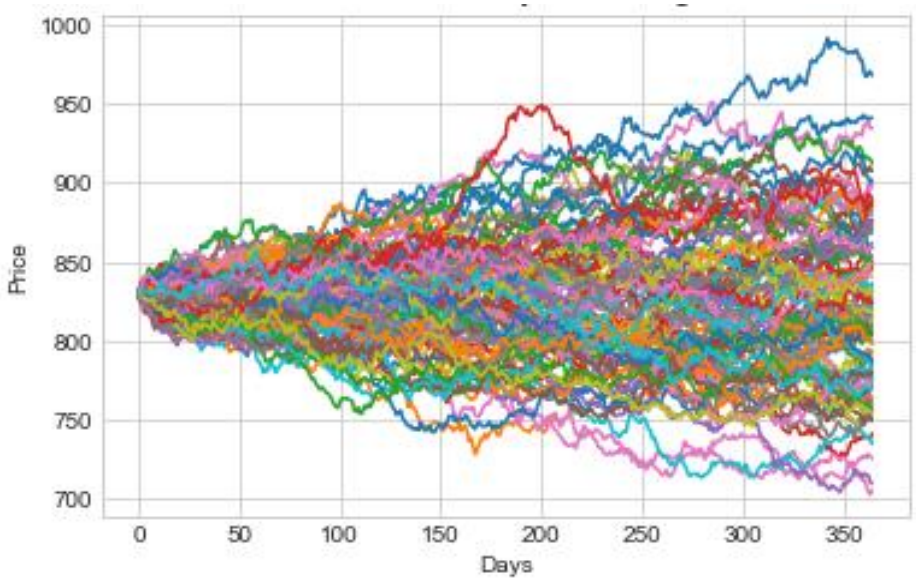
#Value at risk using Monte-Carlo Technique

```
def stock_monte_carlo(start_price,days,mu,sigma)

    # Define a price array
    price = np.zeros(days)
    price[0] = start_price

    # Schok and Drift
    shock = np.zeros(days)
    drift = np.zeros(days)

    # Run price array for number of days
    for x in range(1,days):
        # Calculate Schock
```

```
    shock[x] = np.random.normal(loc=mu * dt, scale=sigma * np.sqrt(dt))
     # Calculate Drift
     drift[x] = mu * dt
     # Calculate Price
     price[x] = price[x-1] + (price[x-1] * (drift[x] + shock[x]))
 return price
start_price = 830.09

for run in range(100):
 plt.plot(stock_monte_carlo(start_price, days, mu, sigma))
plt.xlabel("Days")
plt.ylabel("Price")
plt.title('Monte Carlo Analysis for Google')
```

 # For Google Stock - GOOGL

```
GOOGL.head()
```



11.6: Monte Carlo Analysis for Google

 # For Amazon Stock - AMZN

```
AMZN.head()
start_price = 824.95

for run in range(100):
    plt.plot(stock_monte_carlo(start_price, days, mu, sigma))

plt.xlabel("Days")
plt.ylabel("Price")
plt.title('Monte Carlo Analysis for Amazon')
```

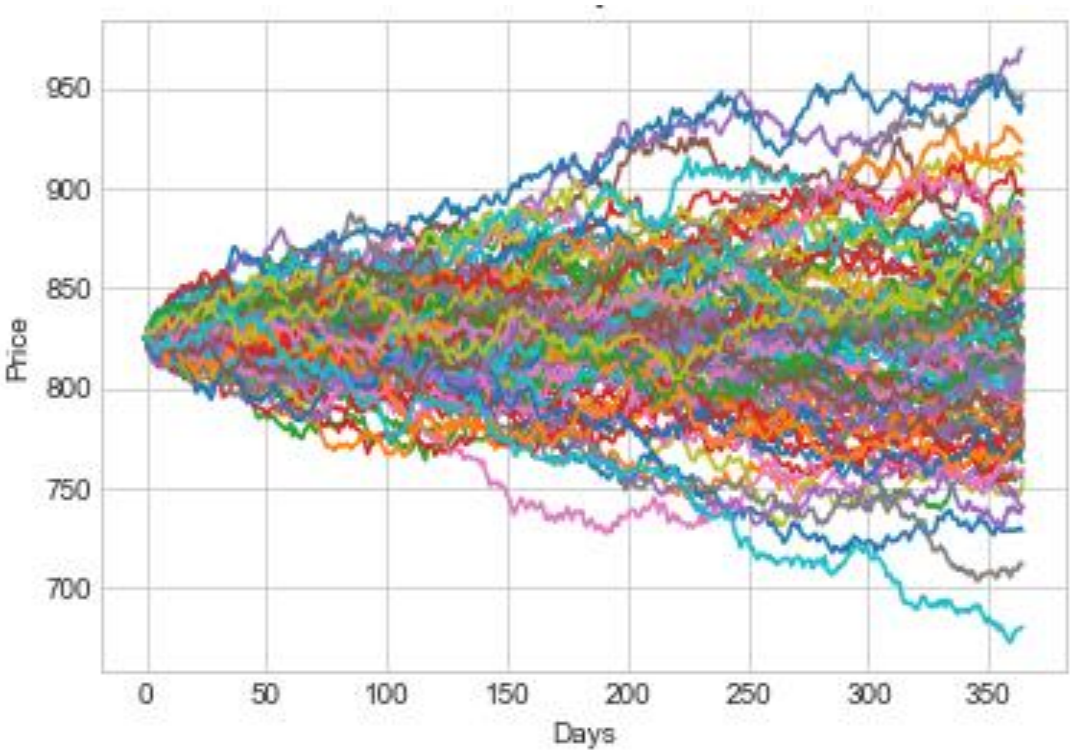


Fig-11.7: Monte Carlo Analysis for Amazon

 # For Apple Stock - AAPL

```
AAPL.head()
start_price = 117.10

for run in range(100):
    plt.plot(stock_monte_carlo(start_price, days, mu, sigma))

plt.xlabel("Days")
plt.ylabel("Price")
plt.title('Monte Carlo Analysis for Apple')
```

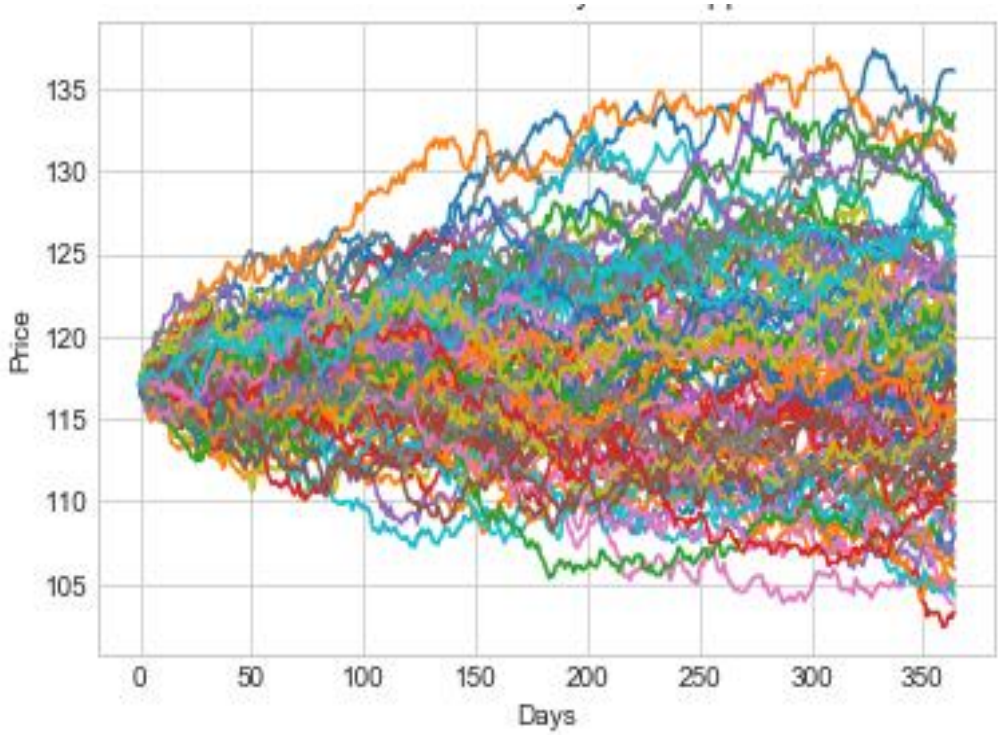


Fig-11.8: Monte Carlo Analysis for Apple

 # For Microsoft Stock - MSFT

```
MSFT.head()

start_price = 59.94

for run in range(100):
    plt.plot(stock_monte_carlo(start_price, days, mu, sigma))

plt.xlabel("Days")
plt.ylabel("Price")
plt.title('Monte Carlo Analysis for Microsoft')
```

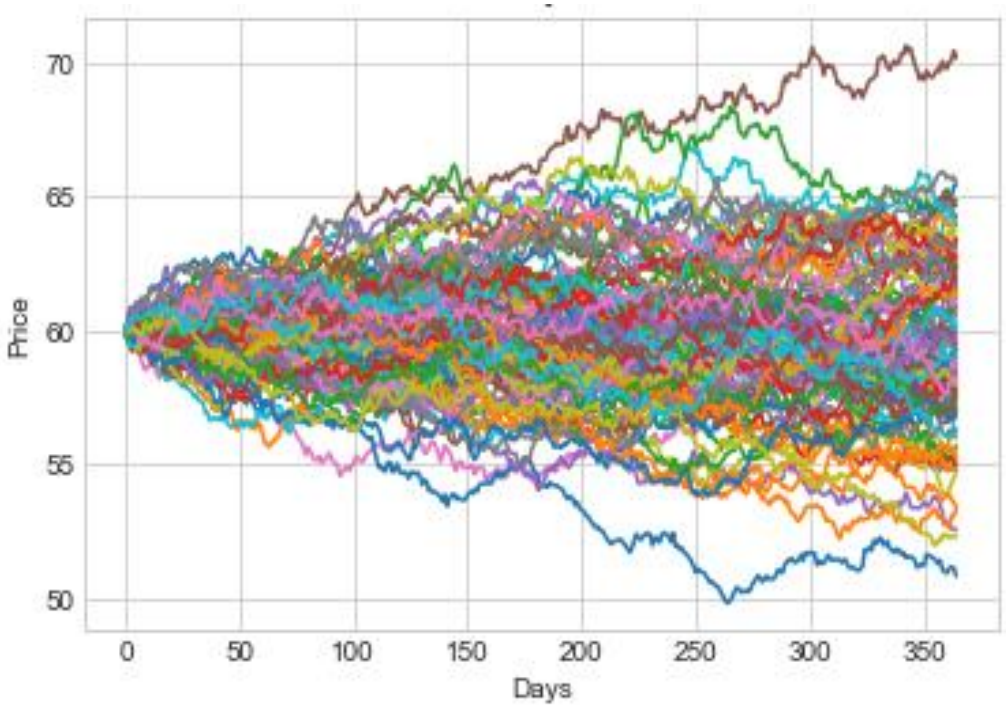


Fig-11.9: Monte Carlo Analysis for Microsoft

#Value at risk using Bootstrap Method

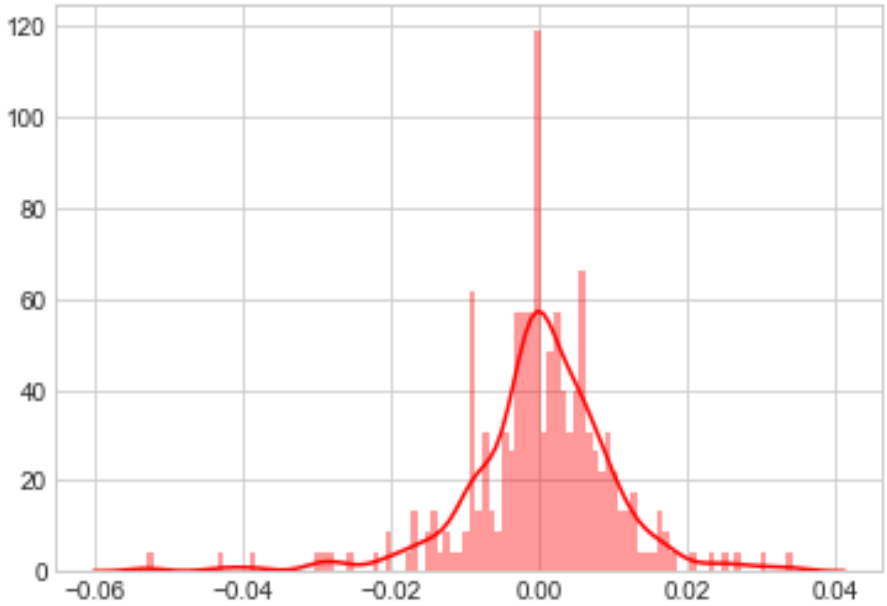
 # We'll use pct_change to find the percent change for each day

#For JNJ stocks

```
JNJ['Daily Return'] = JNJ['Close'].pct_change()
```

 # Note the use of dropna() here, otherwise the NaN values can't be read by seaborn

```
sns.distplot(JNJ['Daily Return'].dropna(),bins=100,color='R')
```



11.10: Daily Return for JNJ Stocks

```
(JNJ['Daily Return'].dropna()).quantile(0.05)
```

 # For WMT stocks

```
WMT['Daily Return'] = WMT['Close'].pct_change()
sns.distplot(WMT['Daily Return'].dropna(),bins=100,color='G')
```

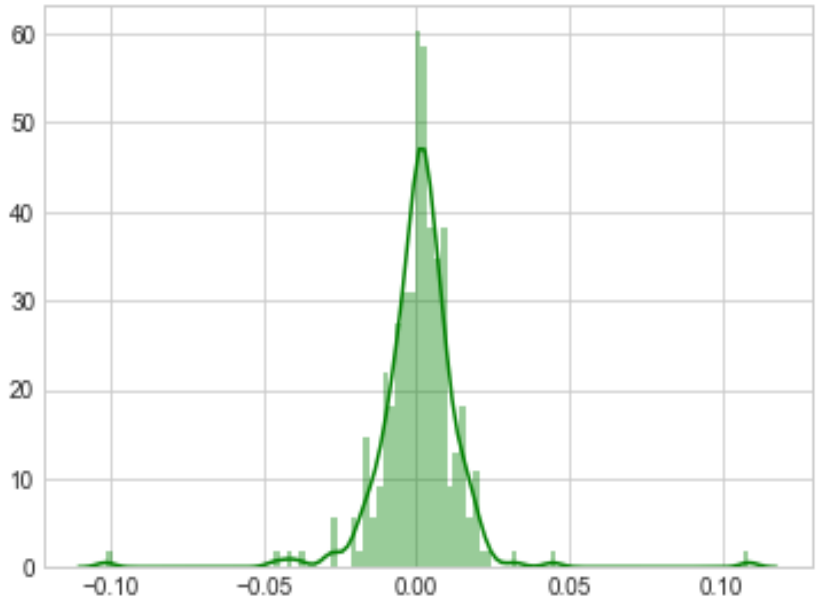


Fig-11.11: Daily Return for WMT Stocks

```
(WMT['Daily Return'].dropna()).quantile(0.05)
```

 # For NKE stocks

```
NKE['Daily Return'] = NKE['Close'].pct_change()
sns.distplot(NKE['Daily Return'].dropna(),bins=100,color='B')
```

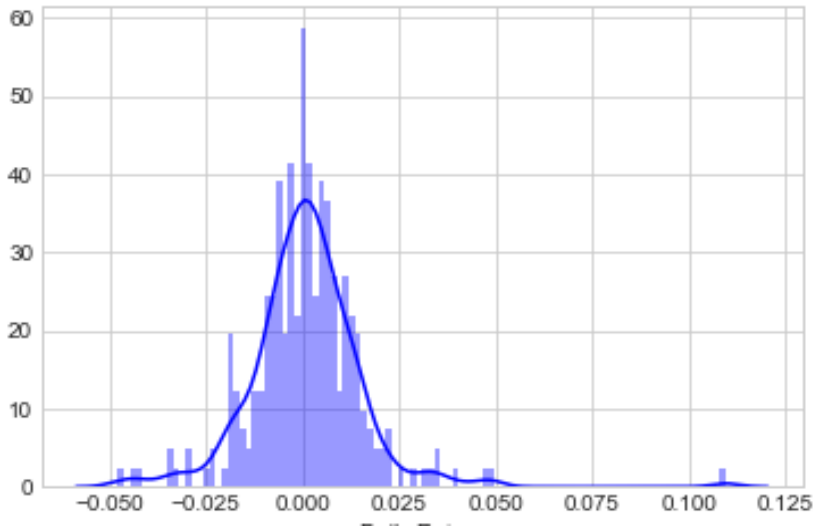


Fig-11.12: Daily Return for NKE Stocks