

STOCK MARKET TREND ANALYSIS USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

PRIYANSHU KUMAR SINGH

(Reg No. 37110600)

UTKARSH CHAUHAN

(Reg No. 37110798)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119 TAMILNADU.

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **PRIYANSHU KUMAR SINGH (Reg No. 37110600), UTKARSH CHAUHAN (Reg No. 37110798)** who carried out the project entitled “**STOCK MARKET TREND ANALYSIS USING MACHINE LEARNING**” under our supervision from **November 2020 to March 2021**.

Internal Guide

Mr. MURARI D. KAMLESH, M.E, (Ph.D)

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D

Dr. L. LAKSHMANAN M.E., Ph.D

Submitted for Viva voice Examination held on _____

External Examiner

Internal Examiner

DECLARATION

I, **PRIYANSHU KUMAR SINGH (Reg No. 37110600)** and **UTKARSH CHAUHAN (Reg No. 37110798)** hereby declare that the Project Report on **“STOCK MARKET TREND ANALYSIS USING MACHINE LEARNING”** done by me under the guidance of **Mr.MURARI D. KAMLESH,M.E.,(Ph.D)** at Sathyabama Institute of Science and Technology(Deemed to be University) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA M.E., Ph.D, Dean, School of COMPUTING, Dr. S. VIGNESHWARI and Dr. L. LAKSHMANAN M.E., Ph.D, Head of the Department, Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. MURARI D. KAMLESH M.E., (Ph.D), DEPARTMENT OF COMPUTER SCIENCE** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

ABSTARCT

In the past decades, there is an increasing interest in predicting markets among economists, policymakers, academics and market makers. The aim of the project is to examine a number of different forecasting techniques to predict future stock returns based on past returns and numerical news indicators to construct a portfolio of multiple stocks in order to diversify the risk. We do this by applying supervised learning methods for stock price forecasting by interpreting the seemingly chaotic market data. We apply supervised learning algorithm like support vector machine. 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. Our aim is to create software that analyses previous stock data of certain companies, with help of certain parameters that affect stock value. We are going to implement these values in data mining algorithms and we will be able to decide which algorithm gives the best result. This will also help us to determine the values that particular stock will have in near future. We will determine the patterns in data with help of machine learning algorithms.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1.	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 STOCK MARKET	2
	1.3 PROBLEM DEFINITION	2
	1.5 DOMAIN DESCRIPTION	2
	1.5.1 MACHINE LEARNING	2
	1.5.2 MACHINE LEARNING METHODS	3
	1.5.3 ADVANTAGES	4
2.	LITERATURE SURVEY	5
3.	AIM AND SCOPE	11
	3.1. EXISTING SYSTEM	11
	3.2. PROPOSED SYSTEM	11
	3.3 FEASIBILITY STUDY	12
4.	ALGORITHM AND METHODS	14
	4.1 HARWARE REQUIREMENT	14
	4.2 SOFTWARE REQUIREMENT	14
	4.3 PYTHON LANGUAGE	14
	4.3.1 PYTHON PLATFORM	15
	4.3.2 PYTHON LIBRARIES	16
	4.4 ANACONDA NAVIGATOR	18

	4.5 SYSTEM ARCHITECTURE	19
	4.6 MODULE DESCRIPTION	20
	4.6.1 DATA PREPROCESSING	20
	4.6.2 FEATURE SELECTION	21
	4 6.3 BUILDING AND TRAINING MODEL	21
	4.6.4 CREATING MODEL	23
	4.6.5 VISUALIZATION	24
5.	RESULTS AND DISCUSSION	25
	5.1 RESULTS	25
6.	CONCLUSION AND FUTURE WORK	28
	6.1 CONCLUSION	28
	6.2 FUTURE WORK	28
	REFERENCES	29
	APPENDIX	
	SOURCE CODE	31

LIST OF FIGURES

FIG. NO.	TITLE	PAGE NO.
4.1	ANACONDA NAVIGATOR	19
4.2	DATA FLOW DIAGRAM	19
4.3	ARCHITECTURE DESIGN	20
4.4	SCALED DATA	23
4.5	TRAINED DATA	24
5.1	SAMPLE DATASET	25
5.2	GRAPH OF CLOSE PRICE HISTORY	26
5.3	PREDICTED GRAPH	26

LIST OF ABBREVIATIONS

API	Application Programming Interface
SVM	Support Vector Machine
WNN	Wavelet Neural Networks
MLP	Multi Layered Perceptron
FFNN	Feed Forward Neural Network
RFID	Radio Frequency Identification
NN	Neural Network

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In recent times stock market predictions is gaining more attention, maybe due to the fact that if the trend of the market is successfully predicted the investors may be better guided. The profits gained by investing and trading in the stock market greatly depends on the predictability. If there is a system that can consistently predict the direction of the dynamic stock market will enable the users of the system to make informed decisions. More over the predicted trends of the market will help the regulators of the market in taking corrective measures.

1.2 STOCK MARKET

A stock market, equity market or share market is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange as well as those only traded privately. Examples of the latter include shares of private companies which are sold to investors through equity crowd funding platforms. Stock exchanges list shares of common equity as well as other security types, e.g. corporate bonds and convertible bonds.

Stock price prediction is one of the most widely studied problem, attracting researchers from many fields. The volatile nature of the stock market makes it really difficult to apply simple time-series or regression techniques. Financial institutions and active traders have created various proprietary models to beat the market for themselves or their clients, but rarely did anyone achieve consistently higher than the average returns on investment. The challenge of stock market price forecasting is so appealing because an improvement of just a few points of percentage can increase the profit by millions of dollars. This paper discusses the application of Support Vector Machines and Linear Regression in detail along with the pros and cons of the given methods. The

paper introduces the parameters and variables which can be used to recognize the patterns in stock prices which can be helpful in future stock prediction and how boosting can be integrated with various other machine learning algorithms to improve the accuracy of our prediction systems.

1.3 PROBLEM DEFINITION

The aims of this project are as to identify factors affecting share market, To generate the pattern from large set of data of stock market for prediction of NEPSE and to predict an approximate value of share price to provide analysis for users through web application

The objective of the system is to give a approximate idea of where the stock market might be headed. It does not give a long term forecasting of a stock value. There are way too many reasons to acknowledge for the long term output of a current stock. Many things and parameters may affect it on the way due to which long term forecasting is just not feasible.

1.4 MOTIVATION

Stock price prediction is a classic and important problem. With a successful model for stock prediction, we can gain insight about market behavior over time, spotting trends that would otherwise not have been noticed. With the increasingly computational power of the computer, machine learning will be an efficient method to solve this problem.

Thus, our motivation is to design a public service incorporating historical data and users predictions to make a stronger model that will benefit everyone.

1.5 DOMAIN DESCRIPTION

1.5.1 MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves. The process of learning begins with observations or data, such as examples, direct

experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

1.5.2 MACHINE LEARNING METHODS

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant

characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

1.5.3 ADVANTAGES

Machine Learning undoubtedly helps people to work more creatively and efficiently. Basically, you too can delegate quite complex or monotonous work to the computer through Machine Learning - starting with scanning, saving and filing paper documents such as invoices up to organizing and editing images. In addition to these rather simple tasks, self-learning machines can also perform complex tasks. These include, for example, the recognition of error patterns. This is a major advantage, especially in areas such as the manufacturing industry: the industry relies on continuous and error-free production. While even experts often cannot be sure where and by which correlation a production error in a plant fleet arises, Machine Learning offers the possibility to identify the error early this saves down times and money. Self-learning programs are now also used in the medical field. In the future, after "consuming" huge amounts of data (medical publications, studies, etc.), apps will be able to warn a in case his doctor wants to prescribe a drug that he cannot tolerate. This "knowledge" also means that the app can propose alternative options which for example also take into account the genetic requirements of the respective patient.

CHAPTER 2

LITERATURE SURVEY

1. Survey of stock market prediction using machine learning approach

Authors: Ashish Sharma ; Dinesh Bhuriya ; Upendra Singh

2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)

Stock market is basically nonlinear in nature and the research on stock market is one of the most important issues in recent years. People invest in stock market based on some prediction. For predict, the stock market prices people search such methods and tools which will increase their profits, while minimize their risks. Prediction plays a very important role in stock market business which is very complicated and challenging process. Employing traditional methods like fundamental and technical analysis may not ensure the reliability of the prediction. To make predictions regression analysis is used mostly. In this paper we survey of well-known efficient regression approach to predict the stock market price from stock market data based. In future the results of multiple regression approach could be improved using more number of variables.

2. Short-term prediction for opening price of stock market based on self-adapting variant PSO-Elman neural network

Authors: Ze Zhang ; Yongjun Shen ; Guidong Zhang ; Yongqiang Song ; Yan Zhu, 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)

Stock price is one of intricate non-linear dynamic system. Typically, Elman neural network is a local recurrent neural network, having one context layer that memorizes the past states, which is quite fit for resolving time series issues. Given this, this paper takes Elman network to predict the opening price of stock market. Considering that Elman network is limited, this paper adopts self-adapting variant PSO algorithm to optimize the weights and thresholds of

network. Afterwards, the optimized data, regarded as initial weight and threshold value, is given to Elman network for training, accordingly the prediction model for opening price of stock market based on self-adapting variant PSO-Elman network is formed. Finally, this paper verifies that model by some stock prices, and compares with BP network and Elman network, so as to draw the result that shows the precision and stability of this predication model both are superior to the traditional neural network.

3. Combining of random forest estimates using LSboost for stock market index prediction

Authors: Nonita Sharma ; Akanksha Juneja,2017 2nd International Conference for Convergence in Technology (I2CT)

This research work emphasizes on the prediction of future stock market index values based on historical data. The experimental evaluation is based on historical data of 10 years of two indices, namely, CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex from Indian stock markets. The predictions are made for 1-10, 15, 30, and 40 days in advance. This work proposes to combine the predictions/estimates of the ensemble of trees in a Random Forest using LSboost (i.e. LS-RF). The prediction performance of the proposed model is compared with that of well-known Support Vector Regression. Technical indicators are selected as inputs to each of the prediction models. The closing value of the stock price is the predicted variable. Results show that the proposed scheme outperforms Support Vector Regression and can be applied successfully for building predictive models for stock prices prediction.

4. Using social media mining technology to assist in price prediction of stock market

Authors: Yaojun Wang ; Yaoqing Wang,2016 IEEE International Conference on Big Data Analysis (ICBDA)

Price prediction in stock market is considered to be one of the most difficult tasks, because of the price dynamic. Previous study found that stock price volatility in a short term is closely related to the market sentiment; especially for small-cap stocks. This paper used the social media mining technology to

quantitative evaluation market segment, and in combination with other factors to predict the stock price trend in short term. Experiment results show that by using social media mining combined with other information, the stock prices prediction model can forecast more accurate.

5. Stock market prediction using an improved training algorithm of neural network

Authors: Mustain Billah ; Sajjad Waheed ; Abu Hanifa,2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)

Predicting closing stock price accurately is an challenging task. Computer aided systems have been proved to be helpful tool for stock prediction such as Artificial Neural Net-work(ANN), Adaptive Neuro Fuzzy Inference System (ANFIS) etc. Latest research works prove that Adaptive Neuro Fuzzy Inference System shows better results than Neural Network for stock prediction. In this paper, an improved Levenberg Marquardt(LM) training algorithm of artificial neural network has been proposed. Improved Levenberg Marquardt algorithm of neural network can predict the possible day-end closing stock price with less memory and time needed, provided previous historical stock market data of Dhaka Stock Exchange such as opening price, highest price, lowest price, total share traded. Moreover, improved LM algorithm can predict day-end stock price with 53% less error than ANFIS and traditional LM algorithm. It also requires 30% less time, 54% less memory than traditional LM and 47% less time, 59% less memory than ANFIS.

6. Efficacy of News Sentiment for Stock Market Prediction Authors:

Sneh Kalra ; Jay Shankar Prasad,2019 International Conferenceon

Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)

Stock Market trend prediction will always remain a challenging task due to stochastic nature. The enormous amount of data generated by the news, blogs, reviews, financial reports and social media are considered a treasure of knowledge for researchers and investors. The present work focuses to observe

fluctuations in stock prices with respect to the relevant news articles of a company. In this paper, a daily prediction model is proposed using historical data and news articles to predict the Indian stock market movements. Classifier Naïve Bayes is used to categorize the news text having negative or positive sentiment. The count of the positive and negative sentiment of news articles for each day and variance of adjacent days close price along with historical data is used for prediction purpose and an accuracy ranging from 65.30 to 91.2 % achieved with various machine learning techniques.

7. Literature review on Artificial Neural Networks Techniques Application for Stock Market Prediction and as Decision Support Tools
Authors: Muhammad Firdaus ; Swelandiah Endah Pratiwi ; Dionysia Kowanda ; Anacostia Kowanda

This literature review is aiming to explore the use Artificial Neural Network (ANN) techniques in the field of stock market prediction. Design: Content analysis research technique. Data sources: Information retrieved from ProQuest electronic databases. Review methods: Utilizing key terms and phrases associated with Artificial Neural Network Stock Market Prediction from 2013-2018. Out of the 129 scholarly journal reviewed, there are 4 stock market studies met the inclusion criteria. The analysis and the evaluation includes 6 ANN derivatives techniques used to predict. Results: Findings from the reviewed studies revealed that all studies shows consistency that the accuracy rate of ANN stock market prediction is high. 2 Studies shows accuracy above 90%, 2 studies shows accuracy above 50%. Conclusion: This study reveals that the ability of ANN shows consistency of an accuracy rate of stock market prediction. Four method in predicting stock market had an accuracy above 95%. The highest accuracy achieved by using Signal Processing/Gaussian Zero-Phase Filter (GZ-Filter) with 98.7% prediction accuracy. 2018 Third International Conference on Informatics and Computing (ICIC)

8. Stock Market Movement Prediction using LDA-Online Learning Model

Authors: Tanapon Tantisripreecha ; Nuanwan Soonthomphisaj, 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)

In this paper, an online learning method namely LDA-Online algorithm is proposed to predict the stock movement. The feature set which are the opening price, the closing price, the highest price and the lowest price are applied to fit the Linear Discriminant Analysis (LDA). Experiments on the four well known NASDAQ stocks (APPLE, FACBOOK GOOGLE, and AMAZON) show that our model provide the best performance in stock prediction. We compare LDA-online to ANN, KNN and Decision Tree in both Batch and Online learning scheme. We found that LDA-Online provided the best performance. The highest performances measured on GOOGLE, AMAZON, APPLE FACEBOOK stocks are 97.81%, 97.64%, 95.58% and 95.18% respectively.

9. Stock Market Prediction Analysis by Incorporating Social and News Opinion and Sentiment

**Authors: Zhaoxia Wang ; Seng-Beng Ho ; Zhiping Lin
2018 IEEE International Conference on Data Mining Workshops (ICDMW)**

The price of the stocks is an important indicator for a company and many factors can affect their values. Different events may affect public sentiments and emotions differently, which may have an effect on the trend of stock market prices. Because of dependency on various factors, the stock prices are not static, but are instead dynamic, highly noisy and nonlinear time series data. Due to its great learning capability for solving the nonlinear time series prediction problems, machine learning has been applied to this research area. Learning-based methods for stock price prediction are very popular and a lot of enhanced strategies have been used to improve the performance of the learning based predictors. However, performing successful stock market prediction is still a challenge. News articles and social media data are also very useful and important in financial prediction, but currently no good method exists that can take these social media into consideration to provide better analysis of the

financial market. This paper aims to successfully predict stock price through analyzing the relationship between the stock price and the news sentiments. A novel enhanced learning-based method for stock price prediction is proposed that considers the effect of news sentiments. Compared with existing learning-based methods, the effectiveness of this new enhanced learning-based method is demonstrated by using the real stock price data set with an improvement of performance in terms of reducing the Mean Square Error (MSE). The research work and findings of this paper not only demonstrate the merits of the proposed method, but also points out the correct direction for future work in this area.

10. Stock Price Prediction Using News Sentiment Analysis

Saloni Mohan ; Sahitya Mullapudi ; Sudheer Sammeta ; Parag Authors:Vijayvergia ; David C. Anastasiu,2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)

Predicting stock market prices has been a topic of interest among both analysts and researchers for a long time. Stock prices are hard to predict because of their high volatile nature which depends on diverse political and economic factors, change of leadership, investor sentiment, and many other factors. Predicting stock prices based on either historical data or textual information alone has proven to be insufficient. Existing studies in sentiment analysis have found that there is a strong correlation between the movement of stock prices and the publication of news articles. Several sentiment analysis studies have been attempted at various levels using algorithms such as support vector machines, naive Bayes regression, and deep learning. The accuracy of deep learning algorithms depends upon the amount of training data provided. In our paper, we improve the accuracy of stock price predictions by gathering a large amount of time series data and analyzing it in relation to related news articles, using deep learning models. The dataset we have gathered includes daily stock prices for S&P500 companies for five years, along with more than 265,000 financial news articles related to these companies.

CHAPTER 3

AIM AND SCOPE

3.1 EXISTING SYSTEM

Nowadays, as the connections between worldwide economies are tightened by globalization, external perturbations to the financial markets are no longer domestic. With evolving capital markets, more and more data is being created daily.

The intrinsic value of a company's stock is the value determined by estimating the expected future cash flows of a stock and discounting them to the present, which is known as the book value. This is distinct from the market value of the stock, that is determined by the company's stock price. This market value of a stock can deviate from the intrinsic value due to reasons unrelated to the company's fundamental operations, such as market sentiment.

The fluctuation of stock market is violent and there are many complicated financial indicators. Only few people with extensive experience and knowledge can understand the meaning of the indicators and use them to make good prediction to get fortune. Most people have to rely solely on luck to earn money from stock trading. However, the advancement in technology, provides an opportunity to gain steady fortune from stock market and also can help experts to find out the most informative indicators to make better prediction. The prediction of the market value is of paramount importance to help in maximizing the profit of stock option purchase while keeping the risk low.

3.2 PROPOSED SYSTEM

We apply support vector machine algorithmSupport Vector Machine is a machine learning technique used in recent studies to forecast stock prices. This study uses daily closing prices for technology stocks to calculate price volatility and momentum for individual stocks and for the overall sector. These are used as parameters to the SVM model.

Advantages

- SVM works relatively well when there is a clear margin of separation between classes.

- SVM is more effective in high dimensional spaces.
- SVM is effective in cases where the number of dimensions is greater than the number of samples.
- SVM is relatively memory efficient.

3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

1. Economic feasibility
2. Technical feasibility
3. Social feasibility

ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. A feasibility study evaluates the project's potential for success.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 4

METHODS AND ALGORITHMS

4.1 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list, especially in case of operating systems. The minimal hardware requirements are as follows,

1. PROCESSOR : PENTIUM IV
2. RAM : 8 GB
3. PROCESSOR : 2.4 GHZ
4. MAIN MEMORY : 8GB RAM
5. PROCESSING SPEED : 600 MHZ
6. HARD DISK DRIVE : 1TB
7. KEYBOARD :104 KEYS

4.2 SOFTWARE REQUIREMENTS

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. These requirements are need to be installed separately before the software is installed. The minimal software requirements are as follows,

1. FRONT END :PYTHON
2. IDE : ANACONDA
3. OPERATING SYSTEM :WINDOWS 10

4.3 PYTHON LANGUAGE

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and

other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging and testing is fast.

4.3.1 PYTHON PLATFORM

Apart from Windows, Linux and MacOS, CPython implementation runs on 21 different **platforms**. IronPython is a .NET framework based **Python** implementation and it is capable of running in both Windows, Linux and in other environments where .NET framework is available.

4.3.2 PYTHON LIBRARIES

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is – “Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.” They are typically used to solve various types of life problems.

In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit- learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

4.3.2.1 NumPy

NumPy is a very popular python library for large multi- dimensional array and matrix processing, with the help of a large collection of high- level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and

random number capabilities. High- end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

4.3.2.2 SciPy

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

4.3.2.3 Skikit

Skikit- learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit- learn supports most of the supervised and unsupervised learning algorithms. Scikit- learn can also be used for data- mining and data- analysis, which makes it a great tool who is starting out with ML.

4.3.2.4 Theano

We all know that Machine Learning is basically mathematics and statistics. Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi- dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit- testing and self- verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large- scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used

by individuals for their own projects.

4.3.2.5 TensorFlow

TensorFlow is a very popular open- source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that

can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

4.3.2.6 Keras

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

4.3.2.7 PyTorch

PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports 15on Computer Vision, Natural Language Processing(NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

4.3.2.8 Pandas

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

4.3.2.9 Matpoltlib

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles,

font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar charts, etc,

4.4 ANACONDA NAVIGATOR

Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning. The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source packages long with free community support.

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.

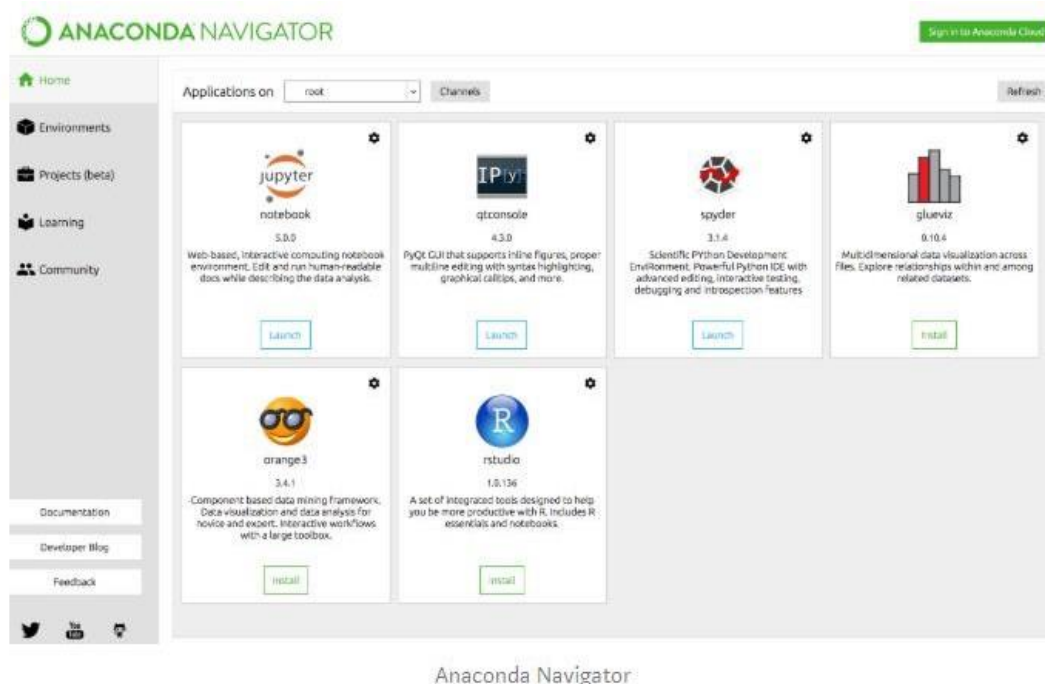


Fig 4.1 Anaconda Navigator

4.5 SYSTEM ARCHITECTURE

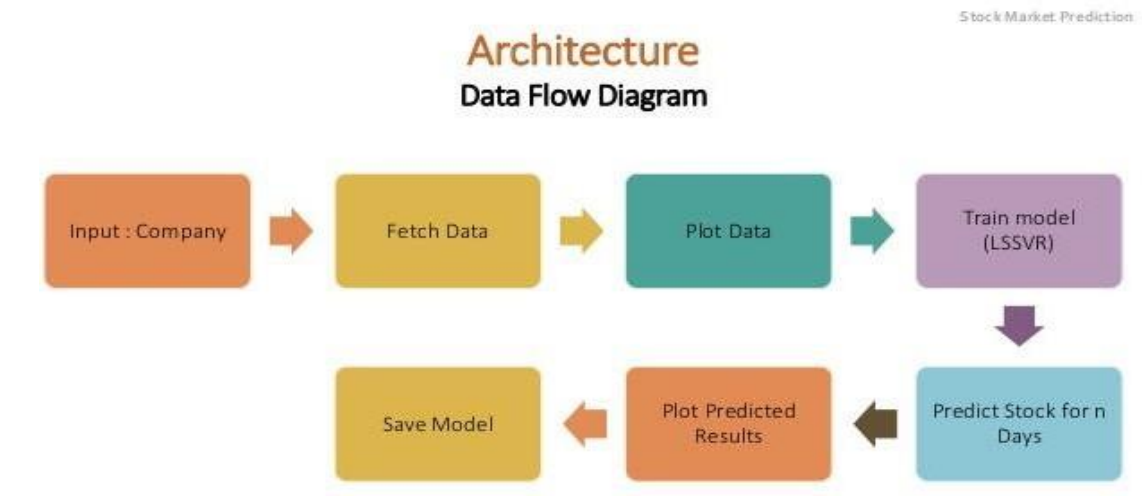


Fig 4.2 Data Flow Diagram

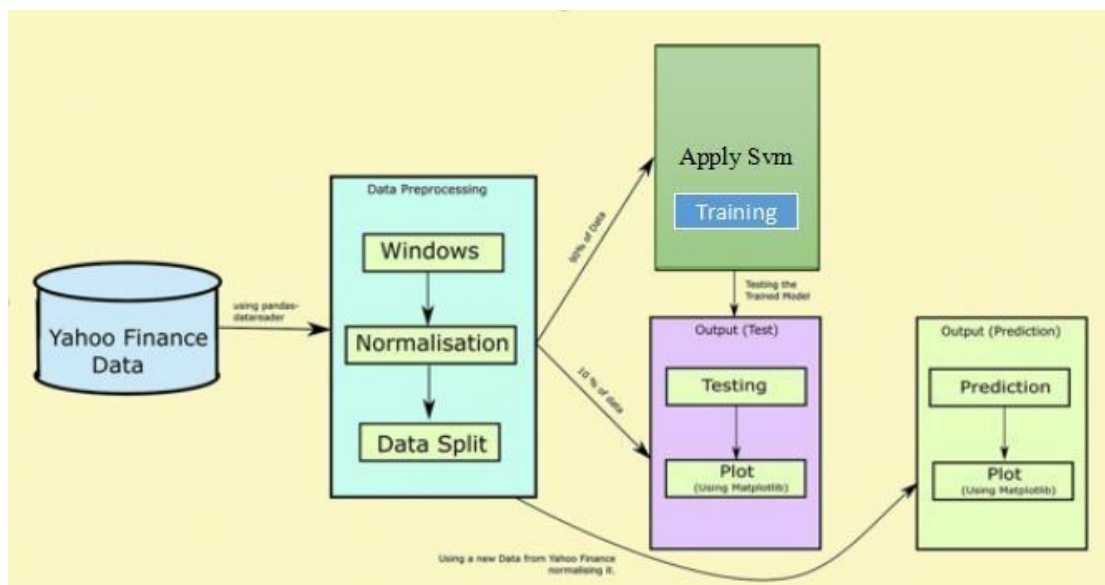


Fig 4.3 Architecture Design

4.6 MODULE DESCRIPTION

The implementation of this project is divided into following steps

1. Data Preprocessing
2. Feature selection

3. Building and Training Model
4. Visualization

4.6.1 Data Preprocessing

The entries are present in the dataset. The null values are removed using `df = df.dropna()` where `df` is the data frame. The categorical attributes (Date, High, Low, Close, Adj value) are converted into numeric using Label Encoder. The date attribute is splitted into new attributes like total which can be used as feature for the model.

Data goes through a series of steps during preprocessing:

Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

Data Integration: Data with different representations are put together and conflicts within the data are resolved.

Data Transformation: Data is normalized, aggregated and generalized.

Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.

```
import math

import pandas_datareader as web

import numpy as np

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import Dense, LSTM

import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')
```

4.6.2 Feature Selection

Features selection is done which can be used to build the model. The attributes used for feature selection are Date, Price, Adj close, Forecast X coordinate , Y coordinate, Latitude , Longitude, Hour and month,

```
df = web.DataReader('AAPL', data_source='yahoo', start='2012-01-01',  
end='2019-12-17')
```

4.6.3 Building and Training Model

After feature selection location and month attribute are used for training. The dataset is divided into pair of xtrain ,ytrain and xtest, y test. The algorithms model is imported from sklearn. Building model is done using model. Fit (xtrain, ytrain). This phase would involve supervised classification methods like linear regression, Ensemble classifiers (like Adaboost, Random Forest Classifiers), etc.

```
data = df.filter(['Close'])
```

```
dataset = data.values
```

```
training_data_len = math.ceil( len(dataset) * .8 )
```

```
training_data_len
```

```
scaler = MinMaxScaler(feature_range=(0,1))
```

```
scaled_data = scaler.fit_transform(dataset)
```

```
scaled_data
```

```
array([[0.01316509],
       [0.01457064],
       [0.01748985],
       ...,
       [0.97658263],
       [0.99755134],
       [1.          ]])
```

Fig 4.4 Scaled Data

```
train_data = scaled_data[0:training_data_len , :]
x_train = []
y_train = []
for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])
if i <= 61:
    print(x_train)
    print(y_train)
    print()
```

```
[array([0.01316509, 0.01457064, 0.01748985, 0.02026915, 0.01984303,
        0.02080338, 0.02036454, 0.01962679, 0.01862193, 0.02173194,
        0.02453668, 0.02367172, 0.01893356, 0.02345548, 0.01900352,
        0.03569838, 0.03440731, 0.0360927 , 0.03973694, 0.04194384,
        0.0417594 , 0.0410789 , 0.04397903, 0.04670744, 0.04979839,
        0.05479095, 0.0652785 , 0.06543749, 0.07127594, 0.07563885,
        0.06814046, 0.07102789, 0.07097066, 0.07906688, 0.07791571,
        0.08004628, 0.08387497, 0.08600558, 0.09214292, 0.09661397,
        0.09790501, 0.09835659, 0.09071194, 0.08886753, 0.08914103,
        0.09632775, 0.09835024, 0.10269405, 0.11293358, 0.12659476,
        0.12403805, 0.1240444 , 0.13392141, 0.13701233, 0.13481179,
        0.13280207, 0.13070964, 0.13766105, 0.14243103, 0.14442805]])]
[0.13949272033425864]

[array([0.01316509, 0.01457064, 0.01748985, 0.02026915, 0.01984303,
        0.02080338, 0.02036454, 0.01962679, 0.01862193, 0.02173194,
        0.02453668, 0.02367172, 0.01893356, 0.02345548, 0.01900352,
        0.03569838, 0.03440731, 0.0360927 , 0.03973694, 0.04194384,
        0.0417594 , 0.0410789 , 0.04397903, 0.04670744, 0.04979839,
        0.05479095, 0.0652785 , 0.06543749, 0.07127594, 0.07563885,
        0.06814046, 0.07102789, 0.07097066, 0.07906688, 0.07791571,
        0.08004628, 0.08387497, 0.08600558, 0.09214292, 0.09661397,
        0.09790501, 0.09835659, 0.09071194, 0.08886753, 0.08914103,
        0.09632775, 0.09835024, 0.10269405, 0.11293358, 0.12659476,
        0.12403805, 0.1240444 , 0.13392141, 0.13701233, 0.13481179,
        0.13280207, 0.13070964, 0.13766105, 0.14243103, 0.14442805)]),
        array([0.02036454, 0.01962679, 0.01862193, 0.02173194, 0.02453668,
        0.02367172, 0.01893356, 0.02345548, 0.01900352, 0.03569838,
        0.03440731, 0.0360927 , 0.03973694, 0.04194384, 0.0417594 ,
        0.0410789 , 0.04397903, 0.04670744, 0.04979839, 0.05479095,
        0.0652785 , 0.06543749, 0.07127594, 0.07563885, 0.06814046,
        0.07102789, 0.07097066, 0.07906688, 0.07791571, 0.08004628,
        0.08387497, 0.08600558, 0.09214292, 0.09661397, 0.09790501,
        0.09835659, 0.09071194, 0.08886753, 0.08914103, 0.09632775,
        0.09835024, 0.10269405, 0.11293358, 0.12659476, 0.12403805,
        0.1240444 , 0.13392141, 0.13701233, 0.13481179, 0.13280207,
        0.13070964, 0.13766105, 0.14243103, 0.14442805, 0.13949272])],
        [0.13949272033425864, 0.13293562570222134]]
```

Fig 4.5 Trained Data

4.6.4 Creating Models

Now, its time to train some prediction-model using our dataset. Scikit-learn provides a wide range of machine learning algorithms which have a unified/consistent interface for fitting, predicting accuracy, etc. We will not go into the details of how the algorithm works as we are interested in understanding its implementation only.

```
model = Sequential()
```

```

model.add(LSTM(50, return_sequences=True, input_shape=
(x_train.shape[1], 1)))

model.add(LSTM(50, return_sequences= False))

19model.add(Dense(25))

model.compile(optimizer='adam', loss='mean_squared_error')

model.fit(x_train, y_train, batch_size=1, epochs=1)

test_data = scaled_data[training_data_len - 60: , :]

x_test = []

y_test = dataset[training_data_len:, :]

for i in range(60, len(test_data)):

x_test.append(test_data[i-60:i, 0])

x_test = np.array(x_test)

x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1 ))

predictions = model.predict(x_test)

predictions = scaler.inverse_transform(predictions)

rmse=np.sqrt(np.mean(((predictions- y_test)**2)))

```

4.6.5 Visualization

Visualizations are the easiest way to analyze and absorb information. Visuals help to easily understand the complex problem. They help in identifying patterns, relationships, and outliers in data. It helps in understanding business problems better and quickly. It is also a precursor to many high-level data analysis for Exploratory Data Analysis(EDA) and Machine Learning(ML). Using math plotlib library from sklearn. Analysis of the Stock dataset is done by plotting various graphs.

```

train = data[:training_data_len]

valid = data[training_data_len:]

```

```
valid['Predictions'] = predictions  
  
plt.figure(figsize=(16,8))  
  
plt.title('Model')  
  
plt.xlabel('Date', fontsize=18)  
  
plt.ylabel('Close Price USD ($)', fontsize=18)  
  
plt.plot(train['Close'])  
  
plt.plot(valid[['Close', 'Predictions']])  
  
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
```

CHAPTER 5

RESULT AND DISCUSSION

5.1 RESULTS

[23]

	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	58.928570	58.428570	58.485714	58.747143	75555200.0	50.994907
2012-01-04	59.240002	58.468571	58.571430	59.062859	65005500.0	51.268970
2012-01-05	59.792858	58.952858	59.278572	59.718571	67817400.0	51.838169
2012-01-06	60.392857	59.888573	59.967144	60.342857	79573200.0	52.380054
2012-01-09	61.107143	60.192856	60.785713	60.247143	98506100.0	52.296970
...	...	...	...	...	...	...
2019-12-11	271.100006	268.500000	268.809998	270.769989	19689200.0	270.128906
2019-12-12	272.559998	267.320007	267.779999	271.459991	34327600.0	270.817261
2019-12-13	275.299988	270.929993	271.459991	275.149994	33396900.0	274.498535
2019-12-16	280.790009	276.980011	277.000000	279.859985	32046500.0	279.197357
2019-12-17	281.769989	278.799988	279.570007	280.410004	28539600.0	279.746094

Fig 5.1 Sample Dataset

Fig 5.1 shows the sample dataset which represents different values of stock market. First column represents date of different sample data. The other column represents high, low on that day. The next column represents the value on which it opens and on which value it closes. The volume column represents the total stock value till that day. Adj Close is the last column which represents the previous value on which it closed.



Fig 5.2 Graph of Close Price History

Fig 5.2 represents the graph which shows the close price on each date. Using the history of closed price the prediction is done. The price value is taken in US dollar.

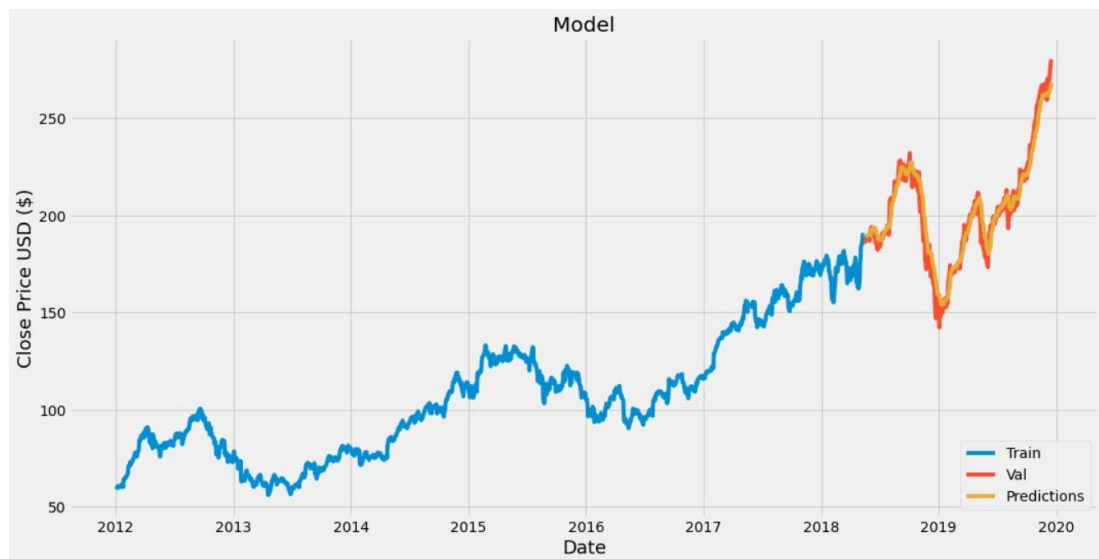


Fig 5.3 Predicted Graph

Figure 5.3 shows the predicted graph which predicts the closed price of different stocks on different future dates. It takes the previous data's and creates a model. Based on the model the prediction is done for the future years.

	Close	Predictions
Date		
2018-05-17	186.990005	188.909317
2018-05-18	186.309998	189.257950
2018-05-21	187.630005	189.337479
2018-05-22	187.160004	189.412064
2018-05-23	188.360001	189.421021
...	...	...
2019-12-11	270.769989	263.593048
2019-12-12	271.459991	264.419495
2019-12-13	275.149994	265.314331
2019-12-16	279.859985	266.540955
2019-12-17	280.410004	268.280457

Fig 5.4 Output Predicted Values

After creating the model, the fig 5.4 shows the original closed values and the predicted values. Finally, we can conclude after seeing the values that success rate is really high and can be really helpful for stocks prediction.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

By measuring the accuracy of the support vector machine algorithms, we found that the most suitable algorithm for predicting the market price of a stock based on various data points from the historical data. The algorithm will be a great asset for brokers and investors for investing money in the stock market since it is trained on a huge collection of historical data and has been chosen after being tested on a sample data. The project demonstrates the machine learning model to predict the stock value with more accuracy as compared to previously implemented machine learning models.

6.2 FUTURE WORK

Future scope of this project will involve adding more parameters and factors like the financial ratios, multiple instances, etc. The more the parameters are taken into account more will be the accuracy. The algorithms can also be applied for analyzing the contents of public comments and thus determine patterns/relationships between the customer and the corporate employee.

REFERENCES

1. Survey of stock market prediction using machine learning approach, Ashish Sharma ; Dinesh Bhuriya ; Upendra Singh
2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)
2. Short-term prediction for opening price of stock market based on self-adapting variant PSO-Elman neural network, Ze Zhang ; Yongjun Shen ; Guidong Zhang ; Yongqiang Song ; Yan Zhu, 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)
3. Combining of random forest estimates using LSboost for stock market index prediction, Nonita Sharma ; Akanksha Juneja, 2017 2nd International Conference for Convergence in Technology (I2CT)
4. Using social media mining technology to assist in price prediction of stock market, Yaojun Wang ; Yaoqing Wang, 2016 IEEE International Conference on Big Data Analysis (ICBDA)
5. Stock market prediction using an improved training algorithm of neural network, Mustain Billah ; Sajjad Waheed ; Abu Hanifa, 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)
6. Efficacy of News Sentiment for Stock Market Prediction, Sneh Kalra ; Jay Shankar Prasad, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)
7. Literature review on Artificial Neural Networks Techniques Application for Stock Market Prediction and as Decision Support Tools, Muhammad Firdaus ; Swelandiah Endah Pratiwi ; Dionysia Kowanda ; Anacostia Kowanda

- 8.** Stock Market Movement Prediction using LDA-Online Learning Model, Tanapon Tantisripreecha ; Nuanwan Soonthomphisaj, 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)
- 9.** Stock Market Prediction Analysis by Incorporating Social and News Opinion and Sentiment, Zhaoxia Wang ; Seng-Beng Ho ; Zhiping Lin
2018 IEEE International Conference on Data Mining Workshops (ICDMW)
- 10.** Stock Price Prediction Using News Sentiment Analysis Saloni Mohan ; Sahitya Mullapudi ; Sudheer Sammeta ; Parag, Vijayvergia ; David C. Anastasiu, 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)

APPENDIX

SOURCE CODE

```
import quandl, math

import numpy as np

import pandas as pd

from sklearn import preprocessing, svm

from sklearn import preprocessing, model_selection, svm

from sklearn.linear_model import LinearRegression

import matplotlib.pyplot as plt

from matplotlib import style

import datetime

# In[ ]:

style.use('ggplot')

df = quandl.get("WIKI/GOOGL")

df.head(5)

df['HL_PCT'] = (df['Adj. High'] - df['Adj. Low']) / df['Adj. Low'] * 100.0

#calculating high low percentage

df['PCT_change'] = (df['Adj. Close'] - df['Adj. Open']) / df['Adj. Open'] * 100.0

#adding HL_PCT, PCT_change to the dataframe

df = df[['Adj. Close', 'HL_PCT', 'PCT_change', 'Adj. Volume']]

#adding HL_PCT, PCT_change to the dataframe

df = df[['Adj. Close', 'HL_PCT', 'PCT_change', 'Adj. Volume']]
```

```

forecast_col = 'Adj. Close'

df.fillna(value=-99999, inplace=True)

forecast_out = int(math.ceil(0.01 * len(df)))

print(forecast_out)

df['label'] = df[forecast_col].shift(-forecast_out)

print(df['label'] )

df[forecast_col]

df.head(-1)

# In[6]:

X = np.array(df.drop(['label'], 1))

X = preprocessing.scale(X)

X_lately = X[-forecast_out:]

X = X[:-forecast_out]


df.dropna(inplace=True)


y = np.array(df['label'])

print(y)

# In[8]:

X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
test_size=0.2)

clf = LinearRegression(n_jobs=-1)

clf.fit(X_train, y_train)

confidence = clf.score(X_test, y_test)

```

```

forecast_set = clf.predict(X_lately)

df['Forecast'] = np.nan

clf = svm.SVR(kernel='rbf')

clf.fit(X_train, y_train)

accuracy = clf.score(X_test, y_test)

print(accuracy)

# In[9]:

last_date = df.iloc[-1].name

print(last_date)

last_unix = last_date.timestamp()

print(last_unix)

one_day = 86400

next_unix = last_unix + one_day

print(next_unix )

# In[10]:

for i in forecast_set:

    next_date = datetime.datetime.fromtimestamp(next_unix)

    next_unix += 86400

    df.loc[next_date] = [np.nan for _ in range(len(df.columns)-1)]+[i]

print(df.head())

df['Adj. Close'].plot()

df['Forecast'].plot()

plt.legend(loc=4)

plt.xlabel('Date')

```

```

plt.ylabel('Price')

plt.show()

import quandl

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import style

import datetime

import random

style.use('ggplot')


df = quandl.get("WIKI/GOOGL")

df.head(-1)

df = df[['Adj. Close']]


df.dropna(inplace=True)

last_close = df['Adj. Close'][-1]

last_date = df.iloc[-1].name.timestamp()

df['Forecast'] = np.nan


for i in range(1000):

    # Create np.Array of current predictions to serve as input for future
    predictions

    modifier = random.randint(-100, 105) / 10000 + 1

    last_close *= modifier

    next_date = datetime.datetime.fromtimestamp(last_date)

```

```
last_date += 86400
```

```
# Outputs data into DataFrame to enable plotting
```

```
df.loc[next_date] = [np.nan, last_close]
```

```
df['Adj. Close'].plot()
```

```
df['Forecast'].plot()
```

```
plt.legend(loc=6)
```

```
plt.xlabel('Date')
```

```
plt.ylabel('Price')
```

```
plt.show()
```