

FEATURE EXTRACTION BASED REINFORCEMENT LEARNING MACHINE FOR STOCK PRICE PREDICTION

Submitted in Partial Fulfilment of The Requirements
For The Award Of
Bachelor of Engineering Degree in Computer Science and
Engineering By

TOLETY VENKATA SAI JEYANTH(37110787)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by
NAAC JEPPIAARNAGAR, RAJIV GANDHI SALAI, CHENNAI**

I-

600119

MARCH-2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NACC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai –

600119 www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of

TOLETY VENKATA SAI JEYANTH (37110787) who
carried out the project entitled **"FEATURE EXTRACTION**

BASED REINFORCEMENT LEARNING MACHINE FOR STOCK PRICE PREDICTION"

under my supervision from **November-2020 to April-2021.**

Internal Guide

Dr. J. Albert Mayan M.E., Ph.D.,

Head of the Department

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

Submitted for viva voce Examination held on _____.

Internal Examiner

External Examiner

DECLARATION

I, **TOLETY V E N K A T A S A I J E Y A N T H (37110787)** hereby declare that the project report entitled **“FEATURE EXTRACTION BASED REINFORCEMENT LEARNING MACHINE FOR STOCK PRICE PREDICTION”** done by me under the guidance of **Dr. J. Albert Mayan M.E., Ph.D.**, is submitted in partial fulfilment 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** 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 M.E., Ph.D.** and **Dr. L. Lakshmanan M.E., Ph.D., Heads** of the 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 **Dr.L.Lakshmanan M.E., Ph.D.**, for his 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.

ABSTRACT

The goal of this project is to predict stock worth movement. During this project we are going to realize economical technique which may predict stock movement additional accurately. Social media offers a robust outlet for people's thoughts and feelings it's a fast-ever-growing supply of texts starting from everyday observations to concerned discussions. This project contributes to the sphere of sentiment analysis that aims to extract emotions and opinions from text. A basic goal is to classify text as expressing either positive or negative feeling. Sentiment classifiers are designed for social media text like product reviews, blogposts, and even email corpus messages. With increasing complexity of text sources and topics, it's time to re-examine the quality sentiment extraction approaches, and probably to re-outline and enrich the definition of sentiment. Next, in contrast to sentiment analysis up to now, we have a tendency to examine sentiment expression and polarity classification inside and across varied social media streams by building topical datasets inside every stream. Totally different data processing ways are a unit accustomed predict market additional with efficiency in conjunction with varied hybrid approaches. We have a tendency to conclude that stock prediction is incredibly advanced task and varied factors sought to be thought off for prognostication the market additional accurately and with efficiency.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1 PROJECT INTRODUCTION	1
	1.2 MACHINE LEARNING	2
	1.3 ML VS TRADITIONAL PROGRAMMING	2
	1.4 HOW DOES MACHINE LEARNING WORK	3
	1.5 INFERRING	4
	1.6 PYTHON OVERVIEW	5
2	LITERATURE SURVEY	7
	2.1 A MACHINE LEARNING APPROACH TO STOCK WORTH MOVEMENT	7
	2.2 DYNAMIC PREDICTION IN STOCK MARKET ANALYSIS USING NEURAL NETWORKS	
	2.3 STOCK PRICE PREDICTION THROUGH DISCOVERING MULTI FREQUENCY TRADING PATTERNS	7
	2.4 TIME SERIES ANALYSIS: DETERMINING & CONTROL HOBOKEN	8
	2.5 ROBUST FINITE-HORIZON FILTERING FOR CONDITIONAL SYSTEMS WITH MISSING MEASUREMENT	8
3	AIM AND SCOPE OF THE PROJECT	9
	3.1 AIM	9
	3.2 SCOPE OF THE PROJECT	9
	3.3 EXISTING SYSTEM	9
	3.3.1 DRAWBACKS OF EXISTING SYSTEM	10
	3.4 PROPOSED SYSTEM	10
	3.4.1 EXAMPLE OF APPLICATION OF	11

	MACHINELEARNINGINSUPPLYCHAIN	
	3.4.2 DEEPLARNING	12
	3.4.3 REINFORCEMENTLEARNING	13
	3.4.4 ADVANTAGESOFPROPOSEDSYSTEM	14
4	METHODOLOGY	15
	4.1 MODULES	15
	4.2 RANDOMFORESTALGORITHM	17
	4.3 SVMALGORITHM	17
	4.4 LOGISTICREGRESSIONALGORITHM	18
	4.5 SUPERISEDLEARNING	18
	4.6 UNSUPERISEDLEARNING	21
	4.7 REQUIREMENTANALYSIS	22
	4.7.1 HARDWAREENVIRONMENT	22
	4.7.2 SOFTWAREENVIRONMENT	23
	4.8 PYTHON	23
	4.8.1 FEATURESOFPYTHON	23
	4.8.2 HISTORYANDVERSIONOFPYTHON	25
	4.8.3 ANACONDANAVIGATOR	26
	4.8.4 NUMPY	28
5	RESULT	31
	OUTPUT	31
6	CONCLUSIONANDFUTUREWORK	32
	REFERENCES	34
	APPENDIX	36
	A. SOURCECODE	36
	B. SCREENSHOTS	41
	C. PUBLICATION	46

LISTOFFIGURES

FIGNO.	TITLE	PAGENO
1.1	PRE-PROCESSING	3
1.2	LEARNINGPHASE	4
1.3	INFERENCE	4
3.1	ARTIFICIALINTELLIGENCE	13
4.1	ALGORITHMSDESCRIPTION	18
5.1	FINALOUTPUT	31

LIST OF TABLES

TAB. NO.	TITLE	PAGE NO.
4.1	SUPERVISED LEARNING ALGORITHMS	21
4.2	UNSUPERVISED LEARNING ALGORITHMS	22

LIST OF ABBREVIATIONS

GPL	-	General Public License
GUI	-	Graphical User Interface
MFC	-	Master Of Finance And Control
NLP	-	Natural Language Processing

CHAPTER

1 INTRODUCTION

N

1.1 PROJECT INTRODUCTION

Stock market prediction is the act of trying to determine the future value of a stock from social media. Social media offers a robust outlet for people's thoughts and feelings. Analysis of social media is strongly related to sentiment analysis. This is used to extract emotions and opinions from text. Data mining methodologies like NLP, Random forest, Neural network is used for analyzing social network content and improve the average accuracy. Recent analysis reveals the existence of attention-

grabbing communication patterns among completely different participants of various social network platforms. These patterns are shown to be helpful in predicting product sales and stock costs. Compared to a social network, which may be thought of as representing connections among folks within the public, a company network connects solely staff in a very huge corporation. While participants of a social network will specific opinions on any problems with interest, members of a company communication network are a unit expected to chiefly say company-specific business.

If human communication patterns will be discovered within the social networks to predict product sales or stock performance, one might surprise if such patterns additionally exist among members in company communication network to permit constant to be done. In contrast to social networks, in a very company communication network, e-mail has long been used as a tool for interorganizational data exchange. Within the same means, a social network platform is ready to capture participants' behaviour and their opinions concerning varied problems and events.

Thus, we tend to argue that a company communication network within the sort of Associate in Nursing e-mail scheme additionally contains perceived data, like structure stability and hardness, a couple of company's

developments. we tend to believe our argument is in line with company communications, that suggests that “employee communications will mean the success or failure of any major amendment program” ensuing from a merger, acquisition, new venture, new

method improvement approach, or alternative management problems. In alternative words, worker communication will serve a crucial “business operation that drives performance and contributes to a company’s financial success”.

Based on these broad company communication theories, we tend to anticipate that each company has its own communication approach with identifiable patterns. We tend to believe that these communication patterns will reflect how a company manages major company activities (such as mergers, acquisitions, new ventures, new method improvement approaches, going considerations, or bankruptcy) which will afterwards influence the company’s performance with in the exchange.

1.2 MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporates to make actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, uses an algorithm to formulate answers.

A typical machine learning task is to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user’s historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation. Machine learning is also used for a variety of tasks like fraud detection, predictive maintenance, portfolio optimization, automatize tasks and soon.

1.3 ML VS TRADITIONAL PROGRAMMING

Traditional programming differs significantly from machine learning. In traditional programming, a programmer codes all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on

logical foundation; the machine will execute an output following the logical statement.

When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

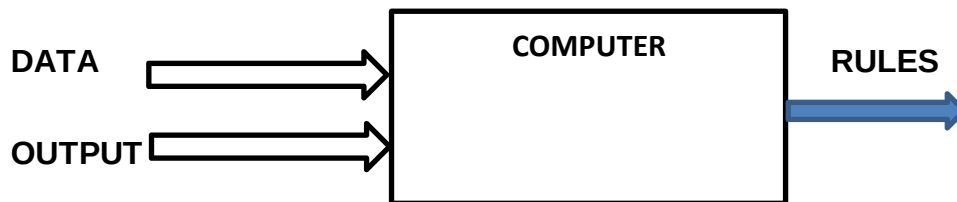


Fig1.1 Pre-Processing

1.4 HOW DOES MACHINE LEARNING WORK?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulty to predict.

The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

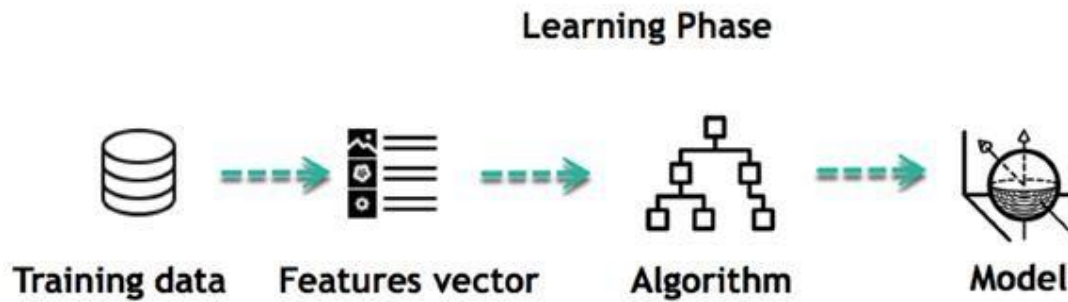


Fig1.2 Learning Phase

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant. This is the model.

1.5 INFERRING

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

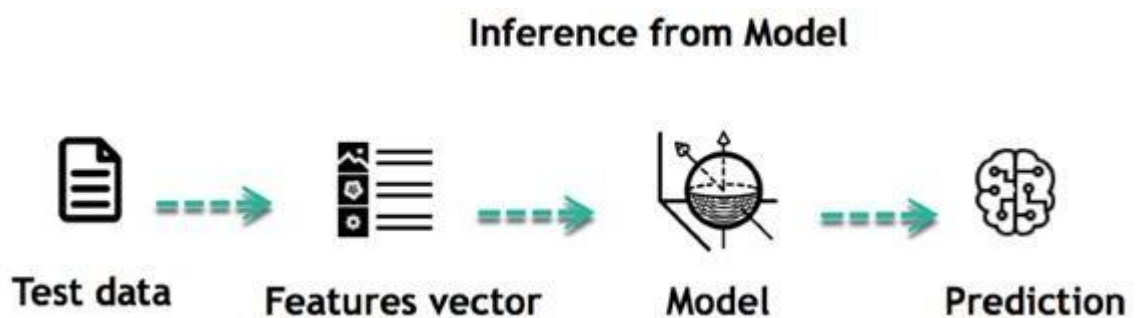


Fig1.3 Inference

The life of Machine Learning programs is straightforward and can be summarized into the following points:

1. Define a question
2. Collect data

3. Visualizedata
4. Trainalgorithm
5. TesttheAlgorithm
6. Collectfeedback
7. Refinethealgorithm
8. Loop4-7untiltheresultsare satisfying
9. Usethemodeltomake aprediction

Once the algorithm gets good at drawingthe right conclusions, itappliesthatknowledgetonewsetsofdata.

1.6 PYTHONOVERVIEW

Python is a high-level, interpreted, interactive and object-oriented scriptinglanguage. Python is designed to be highly readable. It uses English keywordsfrequently where as other languages use punctuation, and it has fewer syntacticalconstructionsthanotherlanguages.

➤ **PythonisInterpreted:**

Python is processed at runtime by the interpreter. You do not need tocompileyourprogrambefore executingit. ThisissimilartoPERLandPHP.

➤ **PythonisInteractive :**

YoucanactuallysitataPythonpromptandinteractwiththeinterpreterdirectly towriteyourprograms.

➤ **PythonisObject-Oriented:**

Python supports Object-Oriented style or technique of programmingthatencapsulatescodewithinobjects.

➤ **PythonisaBeginner'sLanguage:**

Python is a great language for the beginner-level programmers andsupportsthedevelopmentofawiderangeofapplicationsfromsimpletextprocessingt oWWWbrowsers togames.

1.7 HISTORYOFPYTHON

Python was developed by Guido van Rossum in the late eighties and earlynineties at the National Research Institute for Mathematics and Computer ScienceintheNetherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

CHAPTER 2 LITERATURE SURVEY

2.1 TITLE: A Machine Learning Approach to Stock Worth Movement.

2.1.1 Author: Yiwen Meng, William Speier, Member, Chrisandra Shufelt, SandyJoung, Jennifer E Van Eyk, C. Noel Bairey Merz, Mayra Lopez, Brennan Spiegel and Corey W. Arnold

2.1.2 Description: Wavelet depend prediction method for time series. The most prominent univariate time series models is the auto regressive integrated moving average (ARIMA) model. The popularity of ARIMA model is due to its statistical properties and Box-Jenkins methodology in the model selection procedure. But, most works use neural network to extract newly pattern form Multivariate timeseries. Therefore, we chose the SVM with the linear kernel as the prediction model. To assess the effectiveness of sentiment analysis on the message boards, six sets of features are designed.

2.2 TITLE: Dynamic Prediction in Stock Market Analysis using Neural Networks

2.2.1 Author: E. Maiorino, A. Sadeghian, Member, IEEE, and Mihaela van der Schar, Fellow, IEEE

2.2.2 Description: Two serious challenges are associated with financial time series. It's not convenient to copy volatility effect directly, so a neural network combined with Empirical Mode Decomposition (EMD) and de trending was used to solve the problem indirectly. The logical idea behind EMD for forecasting purposes by dividing signal into intrinsic mode functions (IMF) components, the remaining component will decrease the complication of time series, separating trends and oscillations at distinct scales, improving in this way the forecasting exactly at specific time horizons. Pattern recognition of impure will digest a new research direction. The work that analyze the extraction of useful characteristics from financial time series had received attention.

2.3 TITLE: Stock price prediction through discovering multi-frequency trading patterns.

2.3.1 Author: L. Zhang, and G.-J. Qi.

2.3.2 Description: Invented a new novel State Frequency Memory (SFM) recurrent network to take the multi-frequency trading patterns from old market data to make huge and short term predictions. SFM was used to decompose the hidden states of memory cells in multiple frequency components, each models a particular frequency of trading pattern critical the fluctuation of stock price. But, most works use neural network to extract newly pattern from Multivariate time series and join with regression model to predict but lack of consideration of constant patterns along data.

2.4 TITLE: Time Series Analysis: Determining and Control. Hoboken.

2.4.1 Author: G. E. P. Box, and G. M. Ljung.

2.4.2 Description: Planned classical time series analysis method for the time series on stationary and non-stationary conditions. The most prominent univariate time series models is the auto regressive integrated moving average (ARIMA) model. The popularity of ARIMA model is due to its statistical properties and Box-Jenkins methodology in the model selection procedure. So, the effect of model depends on smoothing of non-stationary time series data, mainly upon nonlinear financial time series.

2.5 TITLE: Robust finite-horizon filtering for conditional systems with missing measurements

2.5.1 Author: Z. Wang

2.5.2 Description: Created a fit finite-horizon filters which provides good moving performance. The noise inputs are nonstationary. Joined Stationary Wavelet Transform along some prediction methods gives away to deal with financial time series and used for stock market prediction. Research shows that bridging wavelet multi-resolution and multivariate complex network provides novel method for analyzing multivariate non-linear time series that was in nonlinear science and engineering. Missing data problem in fragmented time series is a challenge in trend analysis. Most classical methods replaces missing data along straight line values will oppose assumptions about noise statistics.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 AIM

The main goal of this project is to review totally different techniques to predict stock worth movement and to find the performance and accuracy of stock's price.

3.2 SCOPE OF THE PROJECT

The Random Forest Algorithm, SVM Algorithm and Logistic Regression Algorithm has long been recognized as being able to efficiently handle high dimensional data and has been shown to perform well on. Therefore, we chose the SVM with the linear kernel as the prediction model. To assess the effectiveness of sentiment analysis on the message boards, six sets of features are designed. The first one used only the historical prices. The other methods incorporated the mood information into the prediction model. All the feature values were scaled into $[-1, 1]$.

3.3 EXISTING SYSTEM

In existing system, we tend to propose that a company's performance, in terms of its stock worth movement, is foreseen by internal communication patterns. To get early warning signals, we tend to believe that it's vital for patterns in company communication network to be detected earlier for the prediction of serious stock worth movement to avoid attainable adversities that an organization could face within the securities market in order that stakeholders' interests are protected to the maximum amount as attainable. Despite the potential importance of such data regarding corporate communication, very little work has been tried in this vital direction. We attempt to bridge these research gaps by employing a data-mining method to examine the linkage between a firm's communication data and its share price. As Enron Corporation's e-mail messages constitute the only corpus available to the public, we make use of Enron's e-mail corpus as the training and testing data for our proposed algorithm.

3.3.1 DrawbacksOfExistingSystem

1. However accuracy would decrease when setting more levels of stock market movement.

2. The average of prediction accuracies using Decision Tree as the classifier are 43.44%, 31.92%, and 12.06% for “two levels,” “three levels,” and “five levels,” respectively.

3.4 PROPOSED SYSTEM

In this Proposed System, opinions can now be found almost everywhere - blogs, social networking sites like Facebook and Email, news portals, e-commerce sites, etc. While these opinions are meant to be helpful, the vast availability of such opinions becomes overwhelming to users when there is just too much to digest. Over the last few years, this special task of summarizing opinions has stirred tremendous interest amongst the Natural Language Processing (NLP) and Random Forest communities. Opinions mainly include opinionated text data such as blog/review articles, and associated numerical data like aspect rating is also included. For example, sentiment prediction on reviews of a product can give a very general notion of what the users feel about the product. If the user needs more specifics, then the topic-based summaries or textual summaries may be more useful. The approaches utilized to address this summarization task vary greatly and touch different areas of research including text clustering, sentiment prediction, Random Forest, NLP analysis, and so on. Some of these approaches rely on simple heuristics, while others use robust statistical models. Some of the applications of machine learning are given below:

1. Augmentation

Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

2. Finance Industry

Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

3. Government organization

The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

4. Healthcare industry

Healthcare was one of the first industries to use machine learning with image detection.

5. Marketing

Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

3.4.1 Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistic hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3% due to the potential reduction in inventory costs.

For example, everybody knows the Google car. The car is full of sensors on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

3.4.2 Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

Examples of deep learning applications:

- **AI in Finance:** The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants. Underwrite is a Fintech company providing an AI solution for credit makers company.
- **AI in HR:** Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30,000 resumes

a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate. At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under Armour chose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, they hired higher quality staffs.

- **AI in Marketing:** AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers. For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

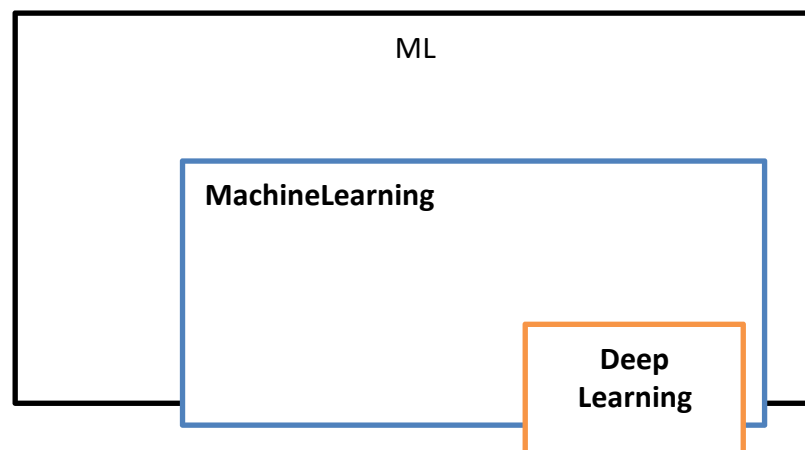


Fig3.1 Artificial Intelligence

3.4.3 Reinforcement learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially

learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

- Q-learning
- Deep Q network
- State-Action-Reward-State-Action (SARSA)
- Deep Deterministic Policy Gradient (DDPG)

3.4.4 Advantages Of Proposed System

1. It classifies the approaches in various ways and describes the techniques used in an intuitive manner.
2. It also provides various aspects of evaluation in opinion summarization, which was not covered by other previous surveys.
3. Finally, this project provides insights into the weaknesses of the approaches and describes the challenges that remain to be solved in this area.

CHAPTER 4 METHODOLOGY

4.1 MODULES

1. DataCollection
2. Pre-Processing
3. DataScoring
4. ResultwithAnalysis

1. DataCollection

Our data consists of mainly daily email about the stock market. Initially we are going to try to analyze the NYSE then upon perfection we will be using our model with the most accuracy to data from Bangladesh and implement a system

to help investors of Bangladesh. We started off with finding a tagged data; unfortunately there were no free sources to find such. What we did then was built our own web script to scrap chat from the email API. We run this script every day for 8 hours and we collect a huge amount of chat daily. Then we store it in JavaScript Object Notation (JSON) to our file storage and parse it.

2. Pre-Processing

These noisy words will interfere with our learning algorithm. Also the biasness seems to increase as a result of including these words. If we include these words then our learning algorithm seems to look for quantifiers like “a”, “an”, “this” etc. Also auxiliary verbs seem to be of no interest to us. So we need only words that give a sense of “good”, “bad” or “neutrality”. The following list of things we had to do in order to pre-process our data:

1. Delete all hyper-text links from the tweet data.

For example: “http://” or “https://t.co/3k7Bai5crQ” is removed from the tweet feed.

2. Change all word blocks from the email corpus data to lowercase. This increases uniformity and change helps us to remove repetitions if present.

3. Removing whitespaces from the tweet data. We keep the emotions

because

they provide helpful insights about the tweet.

4. We remove punctuation marks like commas, full stop etc.

5. Clear out any tag to any person in the email data. Tags starting with “@” are removed. We keep hashtags because sometimes tags like “#Stock #Crash” helps us better understand the mood.
6. Do not take corpus chat in our data email. So remove tweets with “RT” from the dataset.

3. Data Scoring

Our approach for scoring a tweet was simple and effective. The first problems with tweets in the CSV files were that they contain a lot of noisy words which have little to no significance to the actual context. We must remove them to get words of interest to us. Considering this objective, we first collected a list of positive, negative and neutral words in the dictionary. Then what we did was scored based on the unique positive, negative and neutral words on the tweets and our list of words. Suppose, the tweet consists of n words. Now considering the score for positive, negative and neutral score be $Score_{pos}$, $Score_{neg}$ and $Score_{neu}$ respectively and notate the set of all positive, negative and neutral words as $list_{pos}$, $list_{neg}$ and $list_{neu}$ and frequency of positive, negative and neutral words as $frequency_{pos}$, $frequency_{neg}$ and $frequency_{neu}$ respectively we come up with the following formula for scoring the data.

4. Results With Analysis

The experimental setup consists of two components. The first one is collecting data and scoring it. We collect data from Twitter feed and score it. The second component being the learning models. Some of the models are simple and some cutting edge. Based on the learning models we have worked on previously we have come up with interesting results. We used rapid table and meta-chart to make visual representations. We have used three learning models on data that we have collected from a dedicated machine for 3 months. We have divided seventy percent of it for training set and the rest 30% for test set.

➤ ALGORITHMS

4.2 RANDOM FOREST ALGORITHM

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more

powerful prediction model. The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

How Random Forest Works:

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs.
5. Finally, the new record is assigned to the category that wins the majority vote.

Advantages Of Using Random Forest:

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features. The random forest algorithm also works well when data has missing values or it has not been scaled well.

4.3 SVM ALGORITHM

1. Identify the correct hyperplane.
2. Classify two classes.
3. Find hyperplane for segregating two classes.
4. Plot the data points on x and y axis.

4.4 LOGISTIC REGRESSION ALGORITHM

1. Guessing probabilities using basic logistic Function.
2. They transformed into binary values for Prediction.
3. Maps values between the range 0 and 1.

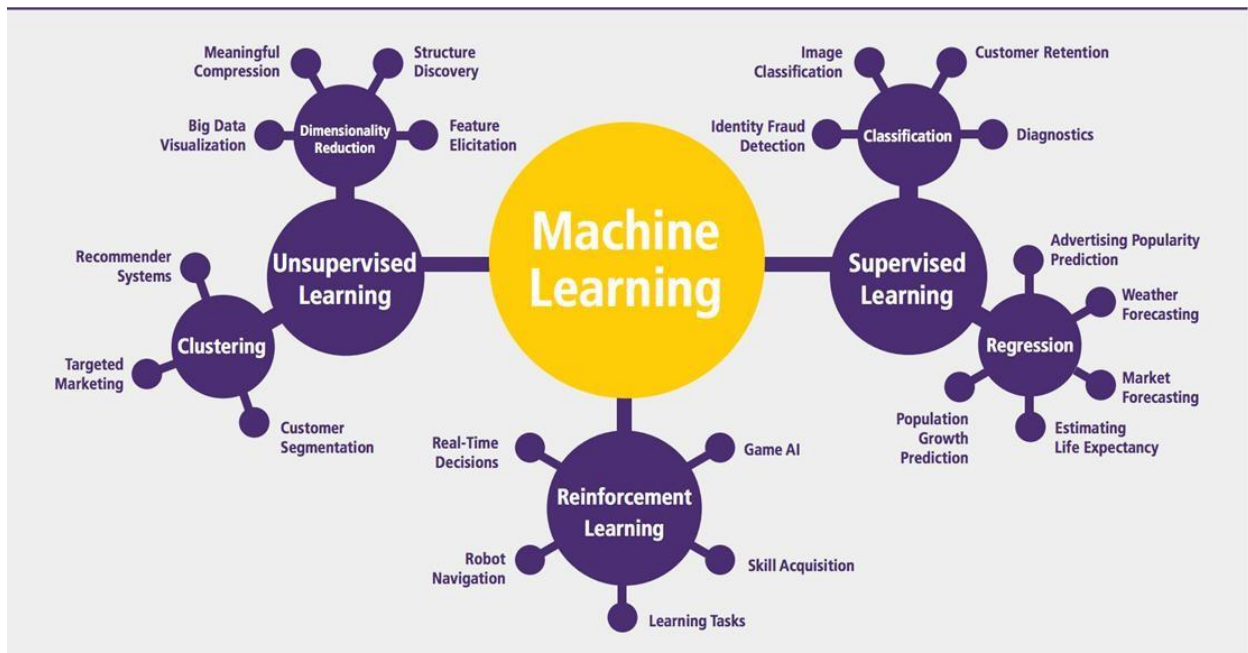


Fig4.1 Algorithms Description

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms:

4.5 SUPERVISED LEARNING

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task
- Regression task

1. Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

2. Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of features like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Algorithm Name	Description	Type
Linear	Finds a way to correlate each feature to the	Regression

regression	output to help predict future values.	
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divides the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple	Regression Classification

decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.

AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
-----------------	---	------------------------------

Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification
--------------------------------	---	------------------------------

Table 4.1 Supervised Learning Algorithms

4.6 UNSUPERVISED LEARNING

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

Algorithm	Description	Type
-----------	-------------	------

K-means

Puts data into some groups (k) that each

Clustering

clustering contains data with similar characteristics (as determined by the model, not in advance by humans)		
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommenders system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithm reduces the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

Table 4.2 Unsupervised Learning Algorithms

4.7 REQUIREMENT ANALYSIS

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements; it gives the wants in terms of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

4.7.1 Hardware Environment

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the

starting point for the system design. It shows what the systems do and not how it should be implemented.

- Harddisk : 40GB
- Monitor : 15' colour with VGA card support
- Ram : 4GB DDRam
- Processor : Pentium iv and above (or) equivalent
- Processor speed : Minimum 500MHZ

4.7.2 Software Environment

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

- Operating system : Windows
- Frontend : MySQL
- Backend : Python

➤ SOFTWARE FEATURES

1. Python 2.7
2. Anaconda Navigator

4.8 PYTHON

Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and scientific applications. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

4.8.1 Features Of Python

A simple language which is easier to learn, Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like C++, Java, C#. Python makes programming fun and allows

you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

➤ ***Free and open source***

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

➤ ***Portability***

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, MacOSX and Linux.

➤ ***Extensible and Embeddable***

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

➤ ***High-level Interpreted language***

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower level operations.

➤ ***Largest standard libraries to solve common tasks***

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server. You can use MySQLdb library using `import MySQLdb`. Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

➤ ***Object-oriented***

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

4.8.2 HistoryAndVersionsOfPython

Python is predominantly a dynamic typed programming language which was initiated by Guido van Rossum in the year 1989. The major design philosophy that was given more importance was the readability of the code and expressing an idea in fewer lines of code rather than the verbose way of expressing things as in C++ and Java [K-8][K-9]. The other design philosophy that was worth mentioning was that, there should be always a single way and a single obvious way to express a given task which is contradictory to other languages such as C++, Perl etc. [K-10]. Python compiles to an intermediary code and this in turn is interpreted by the Python Runtime Environment to the Native Machine Code. The initial versions of Python were heavily inspired from lisp (for functional programming constructs). Python had heavily borrowed the module system, exception model and also keyword arguments from Modula-3 language [K-10]. Python's developers strive not to entertain premature optimization, even though it might increase the performance by a few basis points [K-9]. During its design, the creator had conceptualized the language as being a very extensible language, and hence they had designed the language to have a small core library which was extended by a huge standard library [K-7]. Thus as a result, python is used as a scripting language as it can be easily embedded into any application, though it can be used to develop a full-fledged application. The reference implementation of python is CPython. There are also other implementations like Jython, Iron Python which can use python syntax as well as can use any class of Java (Jython) or .Net class (Iron Python). Versions: Python has two versions 2.x version and 3.x version. The 3.x version is a backward incompatible release was released to fix many design issues which plagued the 2.x series. The latest in the 2.x series is 2.7.6 and the latest in 3.x series is 3.4.0.1.5.2 Paradigms: Python supports multi-paradigmss such as: Object-

Oriented, Imperative, Functional, Procedural, and Reflective. In Object-

Oriented Paradigm, Python supports most of the OOP's concepts such as Inheritance (It also has support for Multiple Inheritance), Polymorphism but its lack of support for encapsulation is a blatant omission as Python doesn't have private, protected members: all class members are public [K-11]. Earlier Python 2.6 versions didn't support some OOP's concepts such as Abstraction through Interfaces and Abstract Classes [K-19].

1.5.3

Management:21AComparativeStudiesofProgrammingLanguages(Comparative Studies of Six Programming Language) Just like the other Managed Languages, Python compiles to an intermediary code and this in turn is interpreted by the Python Runtime Environment to the Native Machine Code. The reference implementation (i.e. CPython) doesn't come with a JIT compiler because of which the execution speed is slow compared to native programming languages [K-17]. We can use PyPy interpreter as it includes a JIT compiler rather than using the Python interpreter that comes by default with the python language, if speed of execution is one of the important factors [K-18]. The Python Runtime Environment also takes care of all the allocation and deallocation of memory through the Garbage Collector. When a new object is created, the GC allocates the necessary memory, and once the object goes out of its scope, the GC doesn't release memory immediately but instead it becomes eligible for Garbage Collection, which would eventually release the memory. But the main disadvantage in having python as a dynamic programming language is that there would be no way to guarantee that a particular piece of code would run successfully for all the different data-type scenarios simply because it had run successfully with one type. Basically, we don't have any means to find out an error in the code, till the code has started running.

1.5.4 Strengths and Weaknesses and Application Areas: Python

is predominantly used as a scripting language used in developing standalone applications that are being developed with Static-Typed languages, because of the flexibility it provides due to its dynamic typed nature. Python favours rapid application development, which qualifies it to be used for prototyping. To a certain extent, Python is also used in developing websites. Due to its dynamic typing and of the presence of a Virtual Machine, there is a considerable overhead which translates to way less performance when we compare with native programming languages [K-13]. And hence it is not suited.

4.8.3 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line

commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux.

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

➤ **What Applications Can I Access Using Navigator?** The following applications are available by default in Navigator:

1. JupyterLab
2. Jupyter Notebook
3. QtConsole
4. Spyder
5. VS Code
6. Glueviz
7. Orange3 App
8. Rodeo
9. RStudio

Advanced conda users can also build their own Navigator applications

➤ **How can I run code with Navigator?**

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way.

Jupyter Notebooks are an increasingly popular system that combine

yourcode,descriptive

text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

➤ **What's new in 1.9?**

1. Add support for Offline Mode for all environment related actions.
2. Add support for custom configuration of main window links.
3. Numerous bug fixes and performance enhancements.

➤ **Python's Standard Library**

1. Pandas
2. Numpy
3. Sklearn
4. tkinter
5. matplotlib

4.8.4 Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the `ndarray` object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an `ndarray` will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy)

objects, thereby allowing for arrays of different sized elements. • NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences. • A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays. The points about sequence size and speed are particularly important in scientific computing. As a simple example, consider the case of multiplying each element in a 1-D sequence with the corresponding element in another sequence of the same length. If the data are stored in two Python lists, *a* and *b*, we could iterate over each element.

The Numeric Python extensions (NumPy henceforth) is a set of extensions to the Python programming language which allows Python programmers to efficiently manipulate large sets of objects organized in grid-like fashion. These sets of objects are called arrays, and they can have any number of dimensions: one dimensional arrays are similar to standard Python sequences, two-dimensional arrays are similar to matrices from linear algebra. Note that one-dimensional arrays are also different from any other Python sequence, and that two-dimensional matrices are also different from the matrices of linear algebra, in ways which we will mention later in this text. Why are these extensions needed? The core reason is a very prosaic one, and that is that manipulating a set of a million numbers in Python with the standard data structures such as lists, tuples or classes is much too slow and uses too much space. Anything which we can do in NumPy we can do in standard Python – we just may not be alive to see the program finish. A more subtle reason for these extensions however is that the kinds of operations that programmer typically want to do on arrays, while sometimes very complex, can often be decomposed into a set of fairly standard operations. This decomposition has been developed similarly in many array lang

uages. In some ways, NumPy is simply the application of this experience to

the Python language—

thus many of the operations described in NumPy work the way they do because experience has shown that way to be a good one, in a variety of contexts. The languages which were used to guide the development of NumPy include the infamous APL family of

languages, Basic,

MATLAB, FORTRAN, S and S+, and others. This heritage will be obvious

to users of NumPy who already have experience with these other languages. This

tutorial, however, does not assume any such background, and all that is expected of the reader

is a reasonable working knowledge of the standard Python language. This document

is the “official” documentation for NumPy. It is both a tutorial and the most authoritative source

of information about NumPy with the exception of the source code. The tutorial

material will walk you through a set of manipulations of

simple, small, arrays of numbers, as well as image files. The result of various manipulations can

be displayed simply since the data set has a natural graphical representation.

All users of NumPy, whether interested in image processing or not, are

encouraged to follow the tutorial with a working NumPy installation at their side, testing the

examples, and, more importantly, transferring the understanding

gained by working on images to their specific domain. The best way to learn is by doing—

the aim of this tutorial is to guide you along this “doing.”

CHAPTER 5 RESULT

The proposed system is trained and tested over the dataset taken from Yahoo Finance. It is split into training and testing sets respectively and yields the following results upon passing through the different models. Our experimental results incontestable the existence of dependence between e-mail communication network and stock worth for Enron the contributions of the planned algorithmic program may be characterized as below. First, we tend to use “adjusted residual” to find the connection between e-mail communication frequency and stock worth.

The result of application of linear regression algorithm on the dataset to predict varying prices with respect to the time. Therefore, joint forces communication patterns will function an honest proxy to predict a company’s stock performance.

OUTPUT



Fig5.1FinalOutput

CHAPTER 6 CONCLUSION AND FUTURE WORK

CONCLUSION

The findings and theoretical implications from this paper are 2-fold. On one hand, we tend to capture the communications among nodes in Enron's major company communication network and identified employees' communication patterns. This paper demonstrates that a company's e-mail scheme contains significant data concerning employees' communication patterns. notwithstanding we tend to solely concentrate on the communication frequency, an organization (Enron in our case) has identifiable patterns of e-mail exchange. Such

identifiable patterns will reveal necessary data concerning major company activities and structure stability which will later influence the focal company's performance within the securities market. The "adjusted residual" live is probabilistic, thus it will work effectively even once the periods of the stock knowledge covered are unequal with missing or inaccurate values. Instead, our prediction rule cannot solely offer condition half and conclusion half, however additionally gift the weighted half for the principles.

FUTURE ENHANCEMENT

This project has some limitations that open fascinating opportunities for future analysis. First, during this paper, we tend to solely create use of the communication frequency between completely different individuals within the e-mail communication network. we can utilize text-mining algorithms to investigate e-mail content and create our experimental results a lot of correct for predicting stock worth movements. Second, during this paper, we tend to rework original information into separate information as none, weak, and strong. A future study will consider modeling the communication level of none, weak, and robust with the utilization of fuzzy sets. Last however not least, a future study will create use of in public out their comments like those on social network sites (e.g., Twitter and Facebook) to predict a company's stock costs. during this direction, future work will extend our algorithmic rule by together with text-mining methods for analyzing

the content of comments on social network sites and determine the patterns and relationships between corporate/consumer communication and companies' stockworth. Finally, we tend to conclude that it's possible to use computation ways like algorithms and data-mining techniques to explore a corporation's communication patterns and utilize such patterns to predict a corporation's structure performance like stock performance. This paper provides a springboard to each research ers and practitioners to any apply algorithms within the areas of knowledge science, management, and finance like for the investigation of the communication and structure performance. we glance forward to a lot of studies that repose on our projected algorithms and extend our findings from Enron to general company analysis.

REFERENCES

- [1] E. Maiorino, and A. Sadeghian, "Data driven detrending of non stationary fractal time series along echo state networks," *Inf. Sci.*, vols. 382–383, pp. 359–373, Mar. 2017.
- [2] G. Lai, and H. Liu, "Modeling long and short temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 95–104.
- [3] X. Qiu, and G. A. J. Amaratunga, "Empirical Mode Decomposition depend upon ensemble deep learning for load demand time series forecasting," *Appl. Soft Comput.*, vol. 54, pp. 246–255, May 2017.
- [4] L. B. Godfrey and M. S. Gashler, "Neural decomposition of time-series information for effective generalization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2973–2985, Jul. 2018.
- [5] S. Jeon, and V. Chang, "Pattern graph tracking based stock market prediction using big data," *Future Gener. Comput. Syst.*, vol. 80, pp. 171–187, Mar. 2018.
- [6] X. Dai and M. Bikdash, "Trend analysis of fragmented time series for mHealth apps: Hypothesis testing based adaptive spline changing method with important weighting," *IEEE Access*, vol. 5, pp. 27767–27776, 2017.
- [7] Z. Wang, and X. Liu, "Robust finite-horizon filtering for conditional systems with missing measurements," *IEEE Signal Process. Lett.*, vol. 12, no. 6, pp. 437–440, Jun. 2005.
- [8] G. E. P. Box, and G. M. Ljung, *Time Series Analysis: Determining and Control*. Hoboken, NJ, USA: Wiley, 2015.
- [9] F. E. Tay and L. Cao, "Application of support vector machines in financial time series Determining," *Omega*, vol. 29, no. 4, pp. 309–317, Aug. 2001.
- [10] K.-J. Kim, "Financial time series forecasting by using support vector machines," *Neurocomputing*, vol. 55, nos. 1–2, pp. 307–319, 2003.

- [11] M. R. Hassan and B. Nath, "Stock market forecasting using private Markovmodel: A new approach," in Proc. 5th Int. Conf. Intell. Syst. Design Appl. (ISDA), Sep. 2005, pp. 192–196.
- [12] A. Gupta and B. Dhingra, "Stock market prediction using private Markovmodels," in Proc. Students Conf. Eng. Syst. (SCES), Mar. 2012, pp. 1–4.
- [13] C. Stolojescu, and A. Isar, "A wavelet depend prediction method for timeseries," in Proc. Int. Conf. Stochastic Modeling Techn. Data Anal., Chania, Greece, 2010.
- [14] C. Stolojescu, and A. Isar, "Forecasting WiMAX BStraffic by statistical approaching in the wavelet domain," in Proc. Int. Symp. Signals, Circuits Syst. (ISSCS), Jul. 2009, pp. 1–4.
- [15] L. Zhang, and G.-J. Qi, "Stock price prediction through discovering multi-frequency trading patterns," in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2017, pp. 2141–2149.

APPENDIX

A. SOURCECODE

```
from sklearn.svm import SVC
from sklearn.ensemble import
GradientBoostingClassifier
from sklearn.metrics import scorer
er
from sklearn.metrics import
accuracy_score
import pandas as pd
import numpy as np
import matplotlib.pyplot as
plt
import seaborn
from sklearn.model_selection import
train_test_split
import
warnings
warnings.filterwarnings
('ignore')
d = pd.read_excel('Book3.xlsx')
d.head()
d.rename(columns=lambda x: x.strip(),
inplace=True)
d.describe()
da = d[(d['TRADINGCODE'] == 'GP')]
da.sort_values('DATE', ascending=False).reset_index(drop=True)
da = da[da['TRADE'] != 0]
plt.figure(figsize=(12, 6))
plt.plot(da['DATE'],
da['CLOSEP'])
plt.xlabel('Time Points')
plt.ylabel('CLOSEP')
plt.title('Time Series Plot of ' + da.loc[1, 'TRADINGCODE'])
import plotly as py
import plotly.graph_objects as go
from datetime import datetime
from plotly.offline import download_plotlyjs, init_notebook_mode,
plot, iplot
init_notebook_mode(connected=True)
trace = go.Candlestick(x=da['DATE'],
```

```

        open=da['OPENP'],
        high=da['HIGH'],
        low=da['LOW'],close
        =da['CLOSEP'])
dat=[trace]
iplot(dat)
importmpl_finance
from mpl_finance import
candlestick_ohlcimportmatplotlib.datesas
mdates
# Converting date to pandas datetime
formatda['DATE'] =
pd.to_datetime(da['DATE'])da['DATE']=da['DATE'].
apply(mdates.date2num)
ohlc=da[['DATE','OPENP','HIGH','LOW','CLOSEP']]
f1,ax=plt.subplots(figsize=(16,8))
candlestick_ohlc(ax, ohlc.values, width=40.6, colorup='green',
colordown='red')ax.xaxis.set_major_formatter(mdates.DateFormatter('%m/%y')
)
import seaborn as
snsimport pandas as
pdimportnumpyasnp
import matplotlib.pyplot as
pltimportmatplotlib.datesasmdate
s
%matplotlibinline
sns.set(style='darkgrid', context='talk',
palette='Dark2')my_year_month_fmt=mdates.DateFormatte
r('%m/%y')
data_1 =
pd.read_excel('Sampledata.xlsx',index_col=[0,1,2,3,4])data_1.h
ead(10)
Fitting 3 folds for each of 32 candidates, totalling 96
fits[CV]C=1,gamma=1,kernel=linear.....

```

```

[CV].....C=1,gamma=1,kernel=linear,total=
0.0s
[CV]C=1,gamma=1,kernel=linear.....
[CV].....C=1,gamma=1,kernel=linear,total=
0.0s
[CV]C=1,gamma=1,kernel=linear.....
[CV].....C=1,gamma=1,kernel=linear,total=
0.0s
estimator=SV
C(C=1.0,cache_size=200,class_weight=None,coef0=0.0,

```

```

decision_function_shape='ovr',      degree=3,      gamma='auto',
kernel='rbf',max_iter=-1,  probability=False,  random_state=None,
shrinking=True,tol=0.001,verbose=False),
    fit_params=None,iid=True,n_jobs=1,
    param_grid={'C': [1,10,100,1000],'gamma':[1,0.1,0.001,0.0001],'kernel':['
linear','rbf']},
    pre_dispatch='2*n_jobs', refit=True,
    return_train_score='warn',scoring=None,verbose=2)
grid.best_params_
{'C': 10, 'gamma': 0.1, 'kernel':
'rbf'}Predic=grid.predict(X_test)
from sklearn.ensemble import
RandomForestClassifierrandom_forest =
RandomForestClassifier(n_estimators=100)random_forest.fit
(X_train,y_train.values.ravel())
RandomForestClassifier(bootstrap=True, class_weight=None,
    criterion='gini',max_depth=None, max_features='auto',
    max_leaf_nodes=None,min_impurity_decrease=0.0,
    min_impurity_split=None,min_samples_leaf=1,
    min_samples_split=2,min_weight_fraction_leaf=0.0,
    n_estimators=100, n_jobs=1,oob_score=False, random_state=None,
    verbose=0,warm_start=False)
from sklearn import
metricslabels= ['Yes','No']
rf_cm=confusion_matrix(y_test,
y_pred)print(rf_cm)
fig=plt.figure()
ax =
fig.add_subplot(111)cax
=
ax.matshow(rf_cm)fig.col
orbar(cax)
ax.set_xticklabels([" +
labels)ax.set_yticklabels([" +

```

```
labels)plt.xlabel('Predicted')plt  
.ylabel('True')  
plt.show()
```

```

from sklearn import
svmclf=svm.SVC()
start =
time.time()print(start)
clf.fit(X_train,y_train)
SVC(C=1.0, cache_size=200, class_weight=None,
    coef0=0.0,decision_function_shape='ovr', degree=3, gamma='auto',
    kernel='rbf',max_iter=-1, probability=False, random_state=None,
    shrinking=True,tol=0.001,verbose=False)
labels = ['Yes',
'No']svm_cm=confusion_matrix(y_test,
svm_pred)print(svm_cm)
fig=plt.figure()
ax =
fig.add_subplot(111)cax =
ax.matshow(svm_cm)fig.col
orbar(cax)
ax.set_xticklabels([''] +
labels)ax.set_yticklabels([''] +
labels)plt.xlabel('Predicted')plt
.ylabel('True')
plt.show()
from sklearn.linear_model import
LogisticRegressionlr=LogisticRegression()
start_lr=time.time()lr.fit(X_train,
y_train)
LogisticRegression(C=1.0, class_weight=None, dual=False,
    fit_intercept=True,intercept_scaling=1, max_iter=100, multi_class='ovr',
    n_jobs=1,penalty='l2', random_state=None, solver='liblinear',
    tol=0.0001,verbose=0, warm_start=False)
lr_pred=lr.predict(X_test)end
_lr=time.time()lr_time=end_lr
-
start_lrprint(lr.score(X_test,y

```

```
_test))lr_acc=lr.score(X_test,  
y_test)
```

```

labels = ['Yes',
'No']lr_cm=confusion_matrix(y_test,
lr_pred)print(lr_cm)
fig=plt.figure()
ax =
fig.add_subplot(111)cax
=
ax.matshow(lr_cm)fig.col
orbar(cax)
ax.set_xticklabels([''] +
labels)ax.set_yticklabels([''] +
labels)plt.xlabel('Predicted')plt
.ylabel('True')
plt.show()
import matplotlib.pyplot as plt;
plt.rcParamsdefaults()importnumpyasnp
importmatplotlib.pyplotasplt
objects = ('Random Forest','SVM','Logistic
Regression')y_pos=np.arange(len(objects))
performance=[rf_acc,svm_acc,lr_acc]
plt.bar(y_pos, performance, align='center',
alpha=0.5)plt.xticks(y_pos,objects)
plt.ylabel('Rated')plt.title('Algorit
hm Performance')plt.show()
import matplotlib.pyplot as plt;
plt.rcParamsdefaults()importnumpyasnp
importmatplotlib.pyplotasplt
objects = ('Random Forest','SVM','Logistic
Regression')y_pos=np.arange(len(objects))
performance =
[rf_time,svm_time,lr_time]plt.bar(y_pos, performance,
align='center', alpha=0.5)plt.xticks(y_pos,objects)
plt.ylabel('Time
Rate')plt.title('Algorithm Time

```

```
Efficiency')plt.show()
```

B. SCREENSHOTS

```
d=pd.read_excel("Book3.xlsx")
```

```
d.head()
```

	DATE	TRADING CODE	LTP	HIGH	LOW	OPENP	CLOSEP	YCP	TRADE	VALUE(mn)	VOLUME
0	2018-08-16	1JANATAMF	6.2	6.3	6.1	6.2	6.2	6.2	56	0.757	122741
1	2018-08-16	1STPRIMFMF	11.2	11.2	10.9	11.0	11.1	10.9	145	2.640	238810
2	2018-08-16	AAMRANET	80.1	80.4	78.5	78.5	79.7	78.3	545	15.488	195035
3	2018-08-16	AAMRATECH	30.8	31.6	30.7	31.0	30.9	31.0	195	5.100	164899
4	2018-08-16	ABB1STMF	6.1	6.1	5.9	6.0	6.1	6.0	109	11.214	1857588

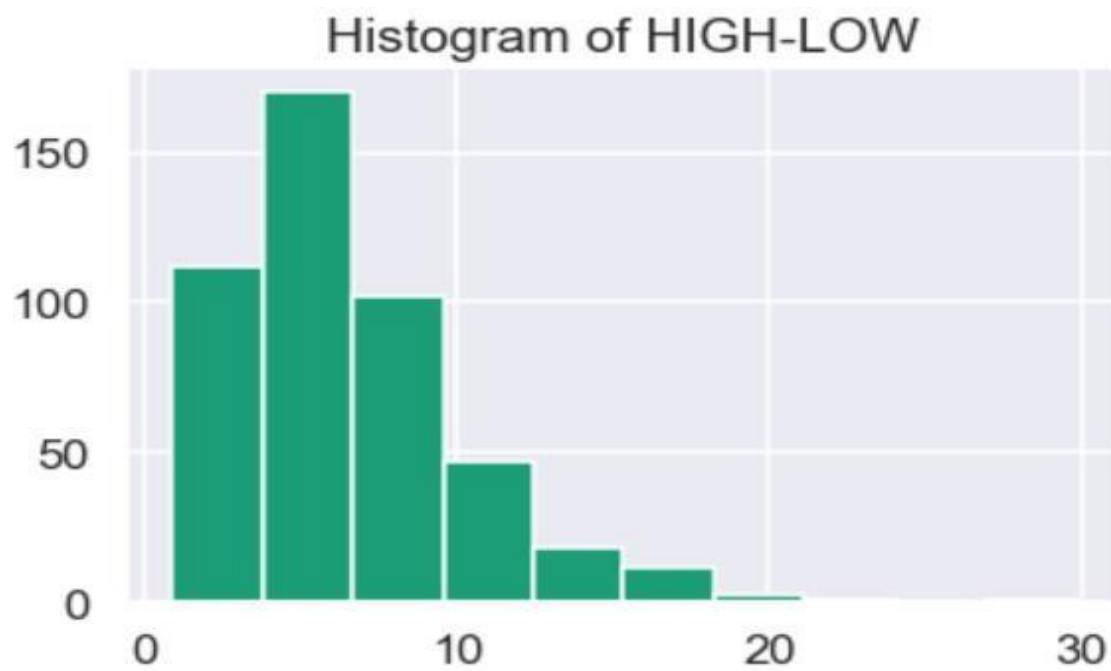
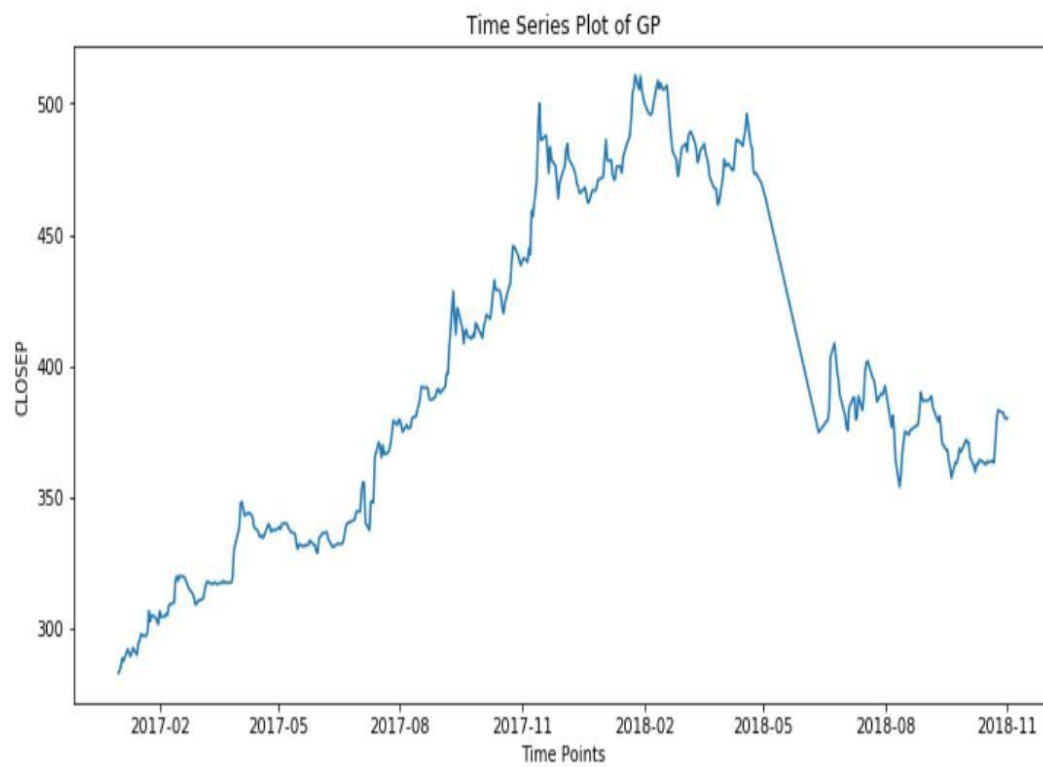
```
d.rename(columns=lambda x: x.strip(), inplace=True)
```

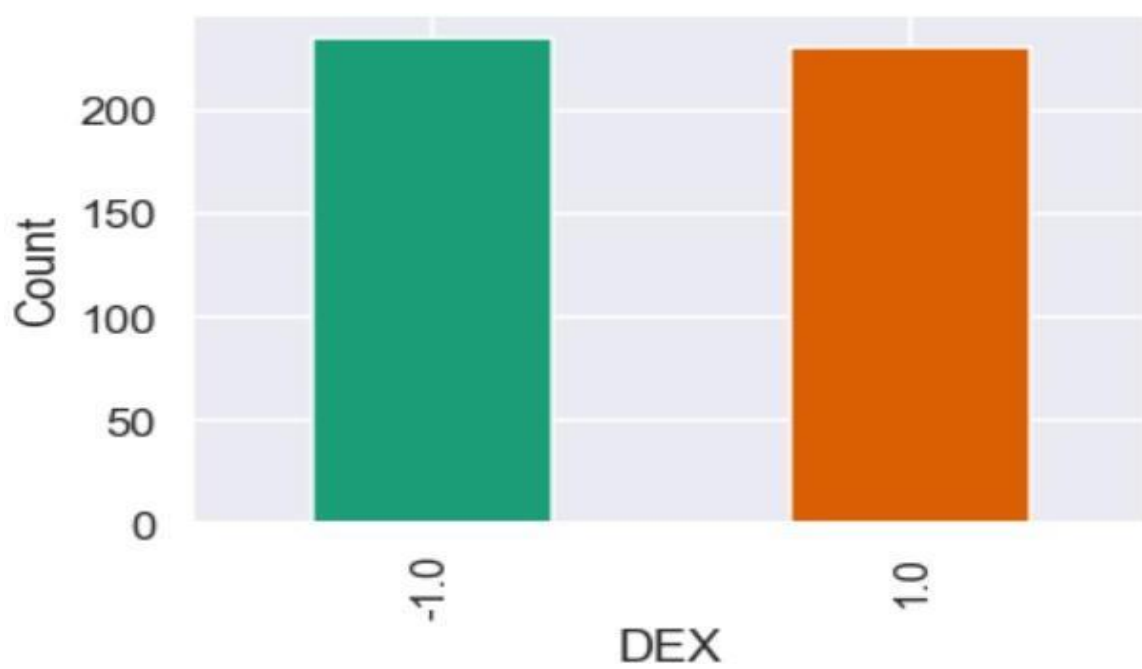
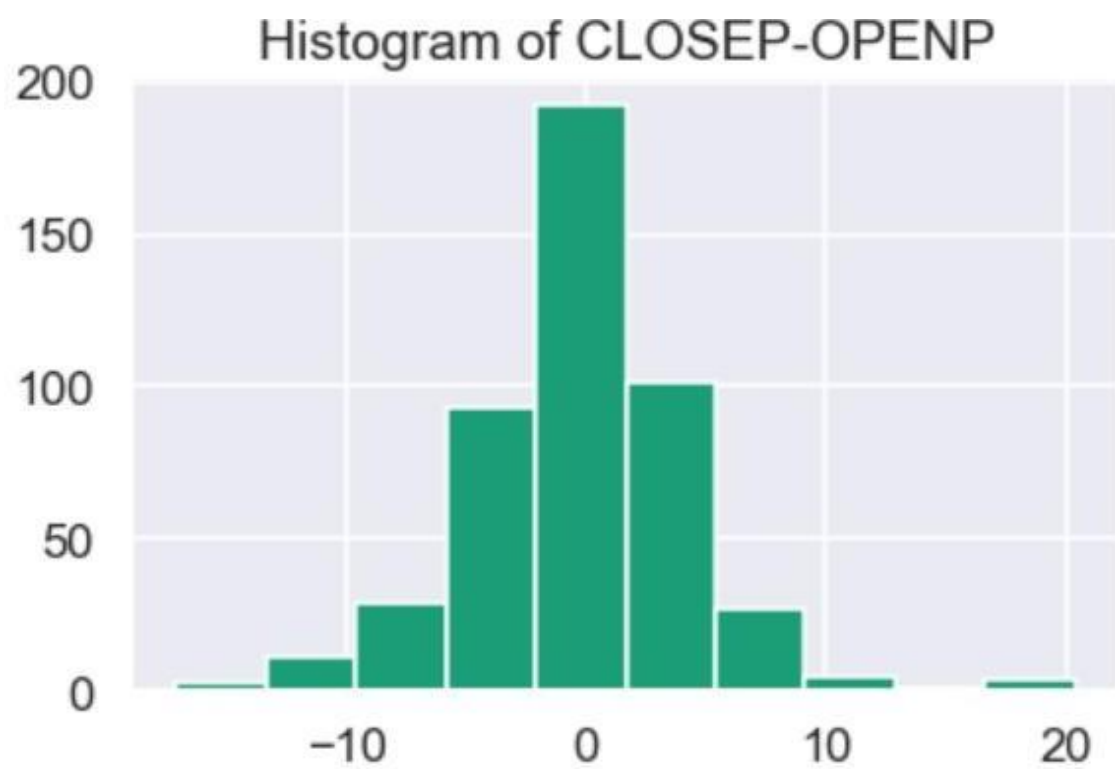
```
da=d[(d['TRADING CODE']=='GP')].sort_values('DATE', ascending=False).reset_index(drop=True)  
da=da[da['TRADE']!=0]
```

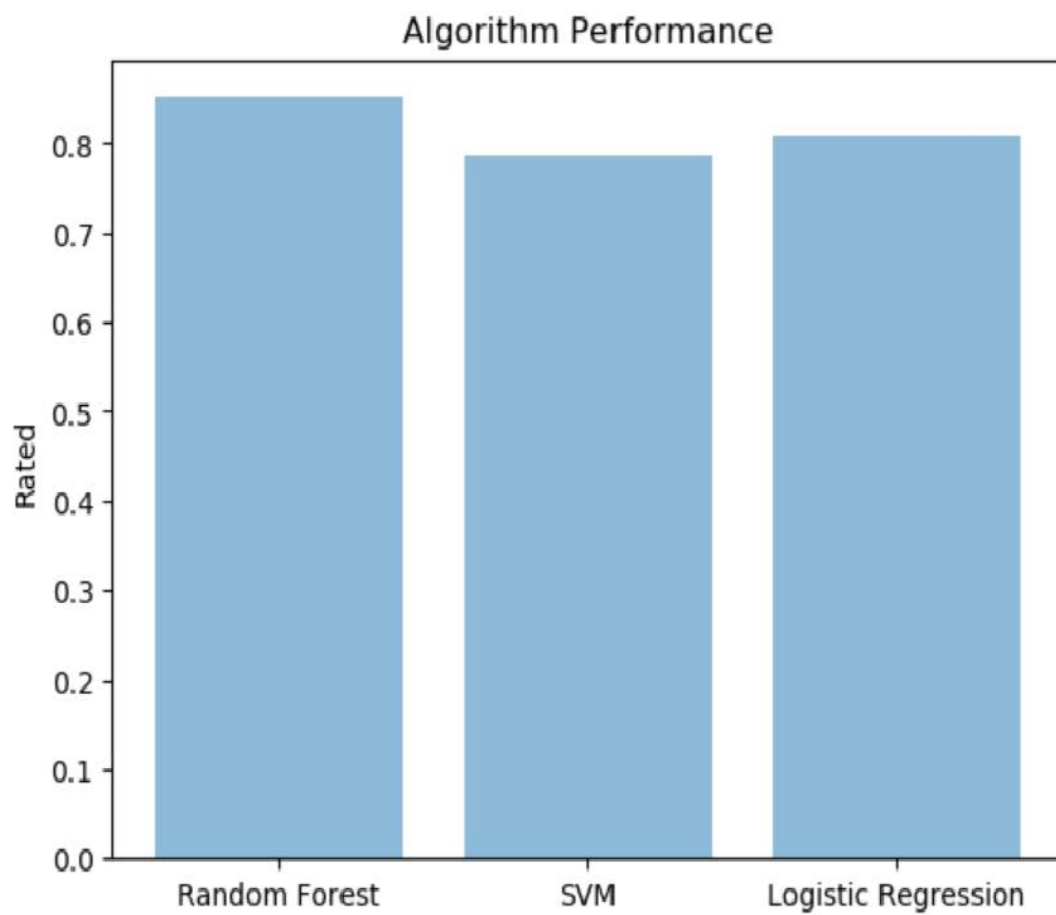
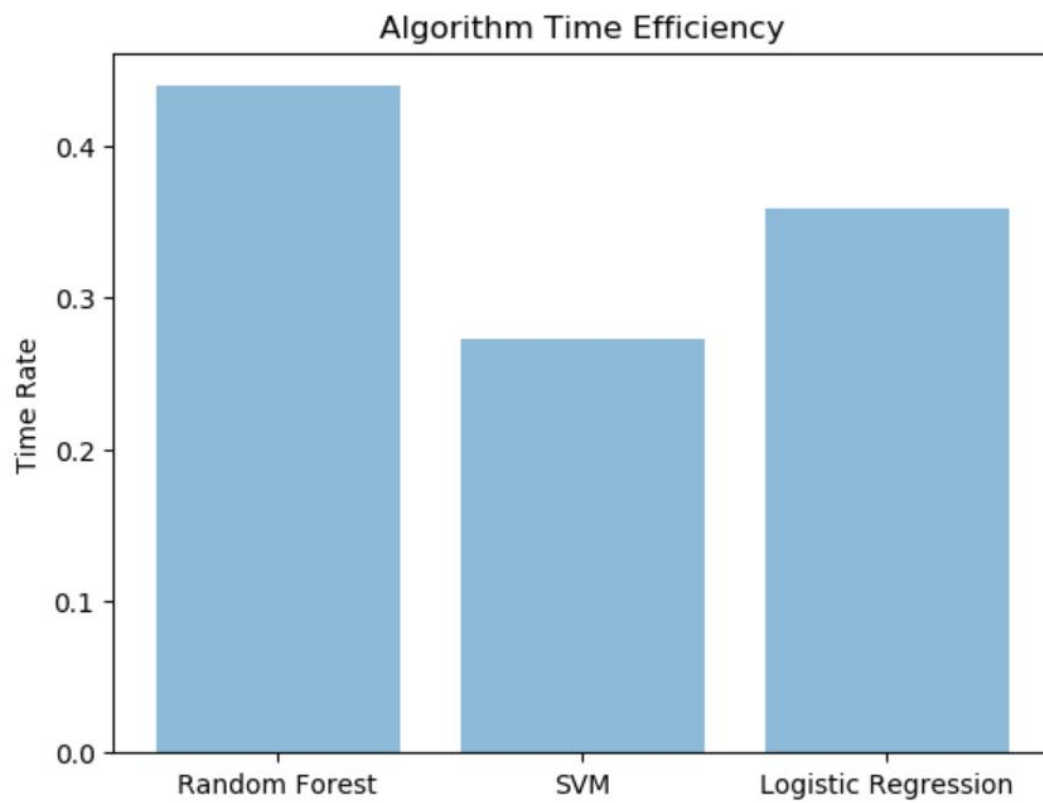
```
da.head()
```

	DATE	TRADING CODE	LTP	HIGH	LOW	OPENP	CLOSEP	YCP	TRADE	VALUE(mn)	VOLUME
0	2018-11-01	GP	380.1	381.0	379.0	379.1	380.1	379.8	452	30.257	79609
1	2018-10-31	GP	380.5	381.9	379.0	380.5	379.8	380.2	825	59.663	156974
2	2018-10-30	GP	380.0	385.5	379.0	383.0	380.2	382.1	788	63.648	166421
3	2018-10-29	GP	381.0	384.0	380.1	384.0	382.1	382.3	868	87.504	229100
4	2018-10-28	GP	382.0	387.0	379.2	383.0	382.3	383.0	1107	75.984	198438

```
Text(0.5,1,'Time Series Plot of GP')
```









TurnitinOriginalityReport

Processed on:23-Feb-
20204:30AMCSTID:1262244996
WordCount:1507
Submitted:1

SimilarityIndex

6%

SimilaritybySource

InternetSources: 2%
Publications: 6%
StudentPapers: 2%

Featureextractionbasedr
einforcement

learningmachine for
stock

Price Prediction By Edgane
sh

4%match(publications)

[MinWen,PingLi,LingfeiZhang,YanChen."Stock Market Trend Prediction Using High-Order Information of Time Series",IEEEAccess,2019](#)

1%match(Internetfrom29-Oct-
2019)[https://ijsiet.org/vol32018/11.p
df](https://ijsiet.org/vol32018/11.pdf)

1% match (Internet from 15-Sep-
2019)<http://ijraset.com/fileserve.php?FID=12891>

<1%match(publications)
[Liheng Zhang, Charu Aggarwal, Guo-Jun Qi. "Stock Price Prediction via Discovering Multi-Frequency Trading Patterns", Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'17, 2017](#)

Featureextractionbasedreinforcementlearningmachineforstockpricepred
iction B.Ganesh Reddy [1], B. Sai Manikanta [2], DR.
L.Lakshmanan[3].[1],[2]UGStudent,DepartmentofComputerSciencean
d Engineering, Sathyabama Institute of Science and
Technology.[3]Professor,DepartmentofComputer science and
Engineering,Sathyabama Institute of Science and TechnologyAbstract—To
analysisstockmovement,thesentimentalanalysisfromsocialmedia.Torealizet
he technique which predicts the stock move correctly. Social
mediagives strong path for people's opinions. It contributes to
sentimentanalysis that predicts data from text. Main goal is
differentiate eithernegative or positive text. Classifiers are created for
social media datalike reviews of product, blogs, and emails. As
complexity increases, ofdata sources, the time has come to re-check
the quality
extractionmethods.Toconcludetheseisincrediblyadvancedmethodanddiffer
entfactorsmightbethoughtformarketadditionalcorrectlywithefficiency.Ke
ywords—Python,RandomForest,SVM,Logistic Regression.
I.INTRODUCTION These project predict the future value of a stock
fromsocialmedia.Socialmediaprovidesastrongpathforpeople'sopinions.A
nalysis of social media is strongly linked to sentiment analysis. It
isusedtoextractopinionsfromdata.DataminingmethodologieslikeNLP,Ran
domforest,Neuralnetworkisusedforanalyzingsocialcontentandimproves
the average accuracy. Previous analysis reveals the existenceof
attention withdrawing patterns. These patterns are helpful
inpredicting product sales and stock costs. Company network
connectsstaffinaveryhugecorporation.Ifhumancommunicationpatterns

performance, its surprises such patterns additionally. II. RELATED WORK
 E. Maorino [1] Two serious challenges are associated with financial time series. It's not convenient to copy volatility effect so direct, so neural network along with EMD [3]–[4] and detrending [7] was used to avoid it. The logical idea behind EMD for sending purposes, dividing signal into IMF parts, the remaining parts will decrease the complication of time series, separating oscillations and parts at unique scales, increasing forecasting exactly at accurate time outlook. The work [2], [5] that analyse the extraction of useful characteristics from time series had received huge attention. F. X. DAI [6] uses HASF algorithm for health apps, that contain sampling which is non-uniform and naturally built to missing data. Z. WANG [7] For a better performance required horizon filters are designed. The inputs are non stationary, [13]–[14] joined Static Wavelet along some prediction methods gives a path to describe time series used in prediction of stock market. Research shows for analyzing multivariate time series is provided by bridging multiresolution wavelet and complex network. In trend analysis a most common challenge is data problem missing. Traditional methods replacing missing data restrict these assumptions about fault statistics.

G. E. P. Box [8]. For both non stationary and stationary conditions an analysis method classical time series is proposed. The ARIMA model is one of the important time series models. ARIMA model is popular because of its statistical properties. Non stationary time series data is on what this model depends on. [9] F. E. TAY and Kim [1] For best prediction performance studied different parameters of SVM. The [work of Hasan, Nath \[11\], Dhingra and Gupta \[12\]](#) shows (HMM) Hidden Markov Model is used on [price forecasting and is really better than ANN and ARIMA method](#) in accurate price prediction, to guide the selection of parameters which is easily influenced by people. L. ZANG [15] [To capture the multi frequency trading patterns from past data to make short predictions a novel state frequency memory network](#) is invented. To split [the hidden states of memory cells](#) into different [frequency components](#) SFM was used. [To extract frequent pattern from varied time series and to combine with classification or regression model](#) to have the [prediction of constant data patterns](#) mostly all of these works use neural network. III. PROPOSED SYSTEM From blogs, social media etc. and almost everywhere the opinions can be found out. The vacancy of opinions are difficult to find out by the users when the data is too much though the opinions are helpful. The important task of compiling opinions has become huge feeling among NLP and Random Forest communities. The text data of opinions contains data like articles from blog, and numerical data like rating. For example, sentiment prediction on product reviews gives a general idea of users feels about

the particular product. If the user wants huge specifics, then the topic based summary may be more useful. Its goal is for helping users to know the availability of the opinions in good manner. Approaches used for reducing work are difficult, so many areas of research like clustering, prediction, Random Forest, etc. Some depends upon easy heuristics, and others use strong models.
 ADVANTAGES IN PROPOSED SYSTEM: a. Describes the methods in different ways and in direct manner. b. Provides various aspects of evaluation in opinion summarization. Data Natu Pre- set ral Proce Algo rith m Rand om Fores Neur al Pre- proce Pre- proce Fig 3.1 Proposed System 1V. ALGORITHMS Algorithm 4.1: Random forest Algorithm 1: Choose 'N' records in set of data. 2: A decision tree is to be built based on records. 3: choose the desired algorithms needed. Repeat 1, 2. 4: Each tree classify's where the new record belongs to. Then the new record belongs to that with huge votes finally. Algorithm 4.2: SVM Algorithm 1: Hyper plane of right is identified. 2: Two classes are identified. 3: Hyper plane of segregate class is found out. 4: Data points

are plotted on x and y axis. Algorithm 4.3 :Logistic

RegressionAlgorithm 1: Probabilities are guessed using basic logistical Functions.2: For prediction purpose these are transformed into binary values. 3:Between 0 and 1 the values are mapped. V . MODULES

There are fourmain modules: A.)Data Collection B.)Pre processing

C.)Data scoringD.)Result with Analysis A.) Data Collection: This data consists of dailyemail about stock market. Initially we try to analyze the NYSE thenbased on perfection we are using our model with accuracy data andimplement a system for investors. We started with a tagged data, butthere are no free sources to find such. So,we created our own

webscripttoslicechatfromemailAPI.Wedailyrunthisscriptfor8hourstocollectahugeamountofchat.ThenstoresinJavaScriptObjectNotation(JSON) to our file storage and define it. B.) Pre processing: Unwantedwords will prevent learning algorithm. Also increases the result withthese words. If these words are included then quantifiers like 'a',

'an'etcareused.Soweneedonlywordswithasenseof"good"or"bad".

Steps for pre-process data: 1. Remove hyper-text links in data.

Forexample: "http://". 2. 1.Remove all hyper-text links in data.

Forexample: "http://". 3. Removing blank spaces ,commas, full stops

etcfrom the data. 4. Tags start with "@" be removed. We keep hashtagslike"#ganesh"because sometimeshelpsusbetterunderstandtheflow.

5. Take bulk chat in our data. And remove tweets with "RT" from

dataset. C.) Data scoring: Our approach to calculate a tweet was

simple.The problems with tweets in CSV files were they contain a lot of

noisywords.Soremovethemtogetwordsofinteresttous.Consideringthis,w

e first collect a list of good, bad and neutral words in the

dictionary.Thenwecalculatebasedontheuniquegood,badandneutralwordsonthe tweets and on the list of words. Suppose, A tweet consists of

nwords. Now considering the score for good, bad and neutral score

beScorepos,ScorenegandScoreneurespectivelyandnotatethesetofallgood, bad and neutral words. Analysing the data with various kinds

offields in Microsoft Excel format then it will be converted as

commadelimited to MySQL backup through the Database. D.) Results

alonganalysis: This contains two components.First is to collect data

fromtwitter then scoring it.Second one is learning models. In these

somemodelsareshimple.Basedonthesedesignswehavetoknowinterestingresults. Table and meta charts are us for making

visualrepresentations.Andalsoused,3learningmodelsondatathatwehavecollected from a dedicated machine for 3 months. 1V. RESULTS

ANDDISCUSSION Fig 4.1 Graph of Stock Price Prediction Output Fig

4.4Histogram Representation Fig 4.3 Comparison of Algorithms Fig

4.4Histogram Representation CONCLUSION This project demonstrates

thata company e-mail scheme contains important data

regardingemployees'communicationpatterns.Generallydon'tconcentrate

onthecommunication frequency. Such patterns will reveal necessary

datapertaining to major company activities and structure stability, this mayslow down the company performance. So we use join

forcescommunication patterns to improve company stock

performance. forparticular problems. Analyzes the clinical data using

Hadoop ecosystemwhere Hive and Pig used for analysis and

MapReduce for

processing.REFERENCES[1]E.Maiorino,andA.Sadeghian,"Datadrivendetr

endingof non stationary fractal time series along echo state networks,"

Inf.Sci.,vols.382–383,pp.359–373,Mar.2017.