A Summer Internship Project Report on

**STOCK MARKET PREDICTION USING MACHINE LEARNING**

Submitted to

The Department of Information Technology

In partial fulfillment of the academic requirements of

Jawaharlal Nehru Technological University

For

The award of the degree of

Bachelor of Technology

in

INFORMATION TECHNOLOGY

By

**RISHIKA BHAT(18311A12L9)**

Under the Guidance of

**Mr.V. Raju**

Associate Professor



Sreenidhi Institute of Science and Technology

Yamnampet, Ghatkesar, R.R. District, Hyderabad – 501301

Affiliated to

Jawaharlal Nehru Technology University

Hyderabad - 500085

Sreenidhi Institute of Science and Technology

The Department of Information Technology



**CERTIFICATE**

This is to certify that this Internship Project report on

**“Stock Market Prediction Using Machine Learning”**

submitted by Rishika Bhat (18311A12L9 ) in the year 2019 in partial fulfillment of the academic requirements of Jawaharlal Nehru Technological University for the award of the degree of Bachelor of Technology in Information Technology, is a bonafide work that has been carried out by them as part of their Internship Project during summer (2020), under our guidance. This report has not been submitted to any other institute or university for the award of any degree.







****

****

**DECLARATION**

I, **RISHIKA BHAT(18311A12L9)** student of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR, of INFORMATION TECHNOLOGY** solemnly declare that the Internship project work, titled **“STOCK MARKET PREDICTION USING MACHINE LEARNING”** is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY** for partial fulfillment for the award of the degree of Bachelor of Technology in **INFORMATION TECHNOLOGY**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for the award of any degree

**RISHIKA BHAT**

**18311A12L9**

**ACKNOWLEDGEMENT**

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heartfelt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably runs the institution and has had a major hand in enabling me to do my project.

I profoundly thank **Dr. V.V.S.S.S. Balaram,** Head of the Department of Information Technology who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank my internal guide **Mr. V. Raju, Associate professor** for his technical guidance, constant encouragement, and support in carrying out my project at college.

I would like to thank my coordinator **Dr K Sreerama Murthy, Associate professor,** for his technical guidance, constant encouragement, and support in carrying out my project at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

**RISHIKA BHAT**

**18311A12L9**

**STOCK MARKET PREDICTION USING MACHINE LEARNING**

**ABSTRACT**

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – physical factors vs. physiological, rational and irrational behavior, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy. We can use machine learning as a game changer in this domain.

In this project, we will work with historical data about the stock prices of a publicly listed company. Using features like the latest announcements about an organization, their quarterly revenue results, etc., machine learning techniques have the potential to unearth patterns and insights we didn’t see before, and these can be used to make unerringly accurate predictions. We will implement a machine learning algorithm like linear regression to predict the future stock price of the company and measure the accuracy of the predictions.

**INDEX**

**Abstract**

**1. INTRODUCTION**

* 1. Scope

1.2 Block Diagram

1.3 Problem Definition

1.4 Softwares

1.5 Algorithm

1.6 Existing systems

1.7 Proposed system

**2. SYSTEM ANALYSIS**

2.1 Functional and Nonfunctional Requirements

2.2Performance Requirements

2.3 Software Requirements

2.4 Hardware Requirements

**3. SYSTEM DESIGN**

3.1 Architecture Design

3.2 Software Design

3.3 UML diagrams

**4. PROJECT IMPLEMENTATION**

**AND OUTPUT SCREENS**

**5. APPLICATIONS**

**6. MERITS AND DEMERITS**

**7. CONCLUSION**

**8. FUTURE ENHANCEMENT**

**9. BIBLIOGRAPHY**

**1. INTRODUCTION**

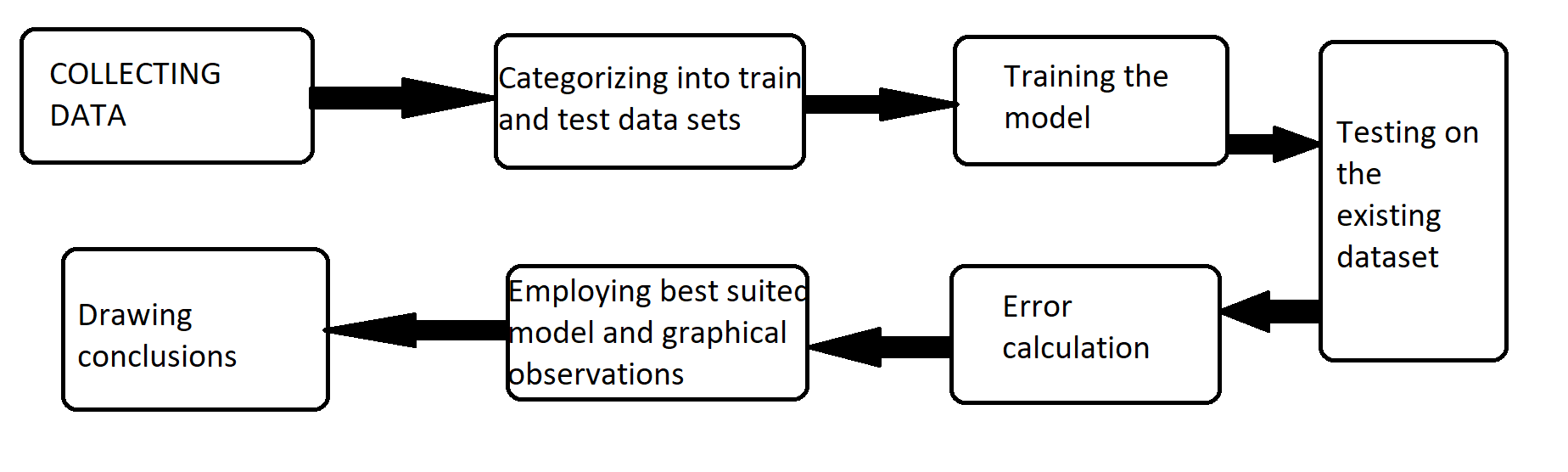
The stock market is basically an aggregation of various buyers and sellers of stock. A stock (also known as shares more commonly) in general represents ownership claims on business by a particular individual or a group of people. The attempt to determine the future value of the stock market is known as a stock market prediction. The prediction is expected to be robust, accurate, and efficient. The system must work according to real-life scenarios and should be well-suited to real-world settings. The system is also expected to take into account all the variables that might affect the stock's value and performance. There are various methods and ways of implementing the prediction system like Fundamental Analysis, Technical Analysis, Machine Learning, Market Mimicry, and Time-series aspect structuring. With the advancement of the digital era, prediction has moved up into the technological realm. The most prominent and promising technique involves the use of Artificial Neural Networks, Recurrent Neural Networks, which are basically the implementation of machine learning. Machine learning involves artificial intelligence which empowers the system to learn and improve from past experiences without being programmed time and again. Traditional methods of prediction in machine learning use algorithms like Backward Propagation, also known as Backpropagation errors. Lately, many researchers are using more of ensemble learning techniques. It would use low prices and time lags to predict future highs while another network would use lagged highs to predict future highs. These predictions were used to form stock prices.

**1.1 SCOPE**

Classically, within the signal processing community, linear parametric models have been a method of first choice in several applications. Historically, many computationally efficient algorithms have been developed for online and adaptive signal processing with e.g. LMS, recursive least squares, and Kalman filtering type algorithms. However, more recently considerable progress has been made also on the use of flexible nonlinear models. Moreover, many emerging applications in e.g. big data, network applications, bioinformatics, and brain-machine interfaces, are posing new challenges for predictive models toward handling large amounts of data in high dimensional input spaces. In this Machine Learning Section, we, therefore, take a broad view on the subject of signal 1 processing & machine learning in connection also to other related areas such as pattern recognition and neural networks, mathematics and statistics, optimization, and information theory. In general, one distinguishes between different types of learning models, such as supervised, unsupervised and semi-supervised learning, various tasks such as e.g. classification, regression, and clustering, and different types of models, including e.g. linear and nonlinear parametric models, kernel-based models and probabilistic models.

Here one is interested in characterizing the generalization error of the model, which is typically expressed in terms of the error on the training data and a complexity term. In many emerging applications one often has to cope with large amounts of data in often high-dimensional input spaces. This is posing new challenges for scalable optimization algorithms.

**1.2 BLOCK DIAGRAM**



**1.3 PROBLEM DEFINITION**

Stock market prediction is basically defined as trying to determine the stock value and offer a robust idea for the people to know and predict the market and the stock prices. It is generally presented using the quarterly financial ratio using the dataset. Thus, relying on a single dataset may not be sufficient for the prediction and can give a result that is inaccurate. Hence, we are contemplating the study of machine learning with various datasets integration to predict the market and stock trends. The problem with estimating the stock

price will remain a problem if a better stock market prediction algorithm is not proposed. Predicting how the stock market will perform is quite difficult. The movement in the stock market is usually determined by the sentiments of thousands of investors. Stock market prediction, calls for an ability to predict the effect of recent events on the investors. These events can be political events like a statement by a political leader, a piece of news on a scam etc. It can also be an international event like sharp movements in currencies and commodities etc. All these events affect the corporate earnings, which in turn affects the sentiment of investors. It is beyond the scope of almost all investors to correctly and consistently predict these hyperparameters. All these factors make stock price prediction very difficult. Once the right data is collected, it then can be used to train a machine and to generate a predictive result.

**1.4 SOFTWARE :**

* Python
* Anaconda Navigator
* Jupyter Notebook
* Numpy
* Pandas
* Matplotlib
* Sklearn
* **PYTHON**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [procedural](https://en.wikipedia.org/wiki/Procedural_programming), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

* **JUPYTER NOTEBOOK**

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

* **NUMPY**

NumPy is a library for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adds support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(math)), along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

* **PANDAS**

In [computer programming](https://en.wikipedia.org/wiki/Computer_programming), panda is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license). The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for data sets that include observations over multiple time periods for the same individuals.

* **MATPLOTLIB**

Matplotlib is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy). It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK+](https://en.wikipedia.org/wiki/GTK%2B). There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB), though its use is discouraged. [SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of matplotlib.

* **SKLEARN**

It features various classification, regression, and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, and DBSCAN, and **is** designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

**1.5 ALGORITHMS USED**

**LINEAR REGRESSION**

The most basic machine learning algorithm that can be implemented on this data is linear regression. The linear regression model returns an equation that determines the relationship between the independent variables and the dependent variable.

The equation for linear regression can be written as:

Here, x1, x2,….xn represent the independent variables while the coefficients θ1, θ2, …. θn represent the weights.

**1.6 EXISTING SYSTEM**

Feed forward back propagation artificial neural network is used in most of the existing systems, where the nonlinear data may not be evenly distributed. The data gets transformed by the Hidden layer (middle layer) of the multilayer network and learns the data transformation to make it linearly separable. This is an approach that gives comparatively less accurate results.

A)Feed Forward Neural Network (FFNN) . These FFNN systems make use of single perception, multiple perceptions, or Hidden layer networks. The inputs are fed in such a way that no cycle is formed. Thus only the forward nodes make use of the given input and thus, reuse cannot be possible in this case leading to lesser accuracy. In single perception FFNN, the inputs are fed directly to the output node using a few weights. Other networks used are hidden layers etc.

B) Hidden Layer Neural Network (HLNN). The hidden layer neural networks use the concept of abstraction. In these networks, the input is fed to output nodes as a result of a function. The estimation of output is based on the provided input. This results in a slightly more accurate result, however as the input is used only once, the possibility of the cycle being formed is limited, thus achieving high accuracy is impossible.

**1.7 PROPOSED SYSTEM**

Convolutional neural networks are more accurate and datasets can be trained more quickly. It is a type of network in which it contains all convolutional layers, it performs well in terms of large data and real-time applications.

Based on the application the user can easily and clearly understand the stock market.

This gives accurate results. It is easily understandable for both programmer and the user.

**2. SYSTEM ANALYSIS**

Machine Learning systems are large and complex. The more powerful way of structuring the complexity lies in architecting the system. Hence an efficient method is needed to structure and handle the complexity of these systems. The analysis of the system will not only satisfy the functional requirements but also satisfies the key non-functional requirements of the system such as performance, reliability, portability, maintainability, etc. Hence system analysis is the very first step in the development of large software-intensive systems in which non-functional i.e., quality requirements are addressed.

**2.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS :**

They refer to very important requirements in a software engineering process such as technical specifications, system design parameters and guidelines, data manipulation, data processing and calculation modules, etc. Functional Requirements are in contrast to other software design requirements referred to as Non-Functional Requirements which are primarily based on parameters of system performance, software quality attributes, reliability and security, cost, constraints in design/implementation, etc.

**Functional Requirements:**

These requirements are categorized by use cases. For any specific use case, there are specific requirements which are detailed below.

* **Text Requirements**

The system should provide text parser functions which can take the whole text and

separate into sentences, paragraphs, and words.

The system should provide a text­to­feature function that can take the necessary part

and obtain a feature vector.

The system should provide a well­trained Autoencoder to generate better inputs for

classifier.

The system needs a classifier that is well­trained to select summary sentences.

* **File Requirements**

A set of functions provide the reading from the file depending on the file extension.

The system should provide communication between the server and client with necessary

network functions such send files and receiving files.

* **Setting Requirements**

The system should take parameters such as summary length from the user before

summarizing.

* **Train System Requirements**

The system should provide a login screen for the admin.

The system should provide taking new data from the admin to train Autoencoders or

classifiers to improve reliability.

**Non­Functional Requirements:**

* **Usability**

The system should be easy to use. The user should reach the summarized text with

one button press if possible. Because one of the software’s features is time-saving.

The system also should be user-friendly for admins because anyone can be an admin

instead of a programmer. Training the Autoencoders and classifiers is used too many times,

so it is better to make it easy.

* **Reliability**

This software will be developed with machine learning, feature engineering, and deep

learning techniques. So, in this step, there is no certain reliable percentage that is measurable.

Also, user-provided data will be used to compare with results and measure reliability.

With recent machine learning techniques, user-gained data should be enough for reliability if

enough data is obtained.

The maintenance period should not be a matter because the reliable version is always

run on the server which allows users to access summarization. When admins want to update, it

takes longer as the upload and update time of the executable on the server. The users can be reached and

use the program at any time, so maintenance should not be a big issue.

* **Performance**

Calculation time and response time should be as little as possible because one of the

Software features are time-saving. The whole cycle of summarizing a page/file should not be more

than 30 seconds in order to 3 pages a long document.

The capacity of servers should be as high as possible. Calculation and response times

are very low, and this comes with that there can be so many sessions at the same time.

1-minute degradation of response time should be acceptable. A certain session limit

also acceptable at the early stages of development. It can be confirmed to the user with the “servers are

not ready at this time” message.

* **Supportability**

The system should require C, Java, Python, and Matlab knowledge to maintain. If

any problem acquires on the server side and deep learning methods, it requires code knowledge

and deep learning background to solve. Client-side problems should be fixed with an update

and it also requires code knowledge and network knowledge.

**MODULE**

**Machine learning** (**ML**) is the study of computer algorithms that improve automatically through experience. Machine learning algorithms build a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory, and application domains to the field of machine learning

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step

Machine learning solutions are widely used in most popular services that deal with cyber security, big data, and cloud computing. Here are some examples of successful machine-learning applications:

* **Amazon Machine Learning**

Amazon Machine Learning is a cloud-based service that allows engineers to create new models by applying off-the-shelf algorithms and advanced visualization tools. In addition, developers can test their models with the help of simple APIs. This service is effective for improving new applications, as developers can see how their product will interact with end users and can forecast product demand.

* **IBM Watson Analytics**

The artificial intelligence platform IBM Watson uses machine learning technology to process information and apply advanced analytics to the business. The IBM team provides Watson with a massive knowledge set from public and private sources in a way that doesn't violate privacy. Watson Analytics performs predictive analysis, detects patterns in data, and visualizes the results. This cloud application can interact with you in the form of questions and answers due to its ability to understand natural language. Using the Watson platform, companies can make better decisions about their business and meet new market challenges.

* **AI2**

AI2 was founded by PatternEx in cooperation with MIT’s Computer Science and Artificial Intelligence Laboratory. This cyber security platform can predict attacks and withstand intrusions significantly better than existing security systems. With the help of unsupervised learning, the platform clusters all available information and finds meaningful patterns. Analyzing system reports, security experts can confirm malicious attacks and apply supervised learning to train the AI2 with newly labeled data for even better detection of suspicious activity in the future.

* **Google Machine Learning**

Google has widely implemented machine learning technologies in its products and services to benefit from the massive information it can obtain by doing so. For instance, using semi-supervised machine learning, Gmail Inbox can automatically categorize your emails and offer a Smart Reply feature. The Cloud Vision API provides developers with powerful machine-learning models for processing image content. Additionally, the Cloud Machine Learning Engine allows technical professionals to train their machine learning models at scale.

**2.2 PERFORMANCE REQUIREMENTS**

Poor requirements definition can adversely impact system cost and performance for government acquisition programs. This can be mitigated by ensuring requirements statements are written in a clear and unambiguous manner that reflects high linguistic quality. This paper introduces a statistical model that uses requirements quality factors to predict system operational performance. This model is created using empirical data from current major acquisition programs within the federal government. Operational Requirements Documents and Operational Test Reports are the data sources, respectively, for the system requirements statements and the accompanying operational test results used for model development. A commercial-off-the-shelf requirements quality analysis tool is used to determine the linguistic quality metrics for the requirements statements. Following model construction, cross validation of the data is employed to confirm the predictive value of the model. In all, the results establish that requirements quality is indeed a predictive factor for end system operational performance; and the resulting statistical model can inform requirements decisions based on the likelihood of successful operational performance

**2.3 SOFTWARE REQUIREMENTS**

* IDE for Python:: Jupiter
* Package Manager::Anaconda
* Operating System Suggested:: Windows 10

**2.4 HARDWARE REQUIREMENTS**

* Processor:: INTEL Quad Core Processor or higher.
* System Specifications :: 8 GB RAM DDR4 , 64-bit-Operating System
* Hard disk:: 20 GB

**3. SYSTEM DESIGN**

System design is the process of envisioning and defining software solutions to one or more sets of problems. One of the main components of system design is the software requirements analysis (SRA). SRA is a part of the software development process that lists specifications used in software engineering. software design may be as simple as a flow chart or text describing a planned sequence of events. Furthermore, a software design may be platform-independent or platform-specific, depending upon the availability of the technology used for the design.

The design focuses on capabilities, and thus multiple designs for the same problem can and will exist. Depending on the environment, the design often varies, whether it is created from reliable frameworks or implemented with suitable design patterns. Design examples include operation systems, webpages, mobile devices or even the new cloud computing paradigm. Software design is both a process and a model.

The design should be traceable to the analysis model.

The design should not reinvent the wheel. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited; design time should be invested in representing ideas by integrating patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should, whenever possible, mimic the structure of the problem domain.

The design should exhibit uniformity and integration. A design is uniform if it appears fully coherent. In order to achieve this outcome, rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

**3.1 ARCHITECTURE DESIGN**

Machine Learning architecture is defined as the subject that has evolved from the concept of fantasy to the proof of reality. An earlier machine learning approach for pattern recognition has led the foundation for the upcoming major artificial intelligence program. Based upon the different algorithm that is used in the training data machine learning architecture is categorized into three types i.e. Supervised Learning, Unsupervised Learning, and Reinforcement Learning and the process involved in this architecture are Data Acquisition, Data Processing, Model Engineering, Excursion, and Deployment.

Types of Machine Learning Architecture

The Machine Learning Architecture can be categorized on the basis of the algorithm used in training.

**1. Supervised Learning**

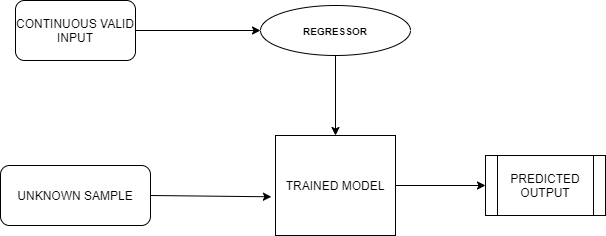
In supervised learning, the training data used is a mathematical model that consists of both inputs and desired outputs. Each corresponding input has an assigned output which is also known as a supervisory signal. Through the available training matrix, the system is able to determine the relationship between the input and output and employ the same in subsequent inputs post-training to determine the corresponding output. Supervised learning can further be broadened into classification and regression analysis based on the output criteria. Classification analysis is presented when the outputs are restricted in nature and limited to a set of values. However, regression analysis defines a numerical range of values for the output. Examples of supervised learning are seen in face detection and speaker verification systems.

**2. Unsupervised Learning**

Unlike supervised learning, unsupervised learning uses training data that does not contain output. Unsupervised learning identifies relation input based on trends, and commonalities, and the output is determined on the basis of the presence/absence of such trends in the user input.

**3. Reinforcement Training**

This is used in training the system to decide on a particular relevant context using various algorithms to determine the correct approach in the context of the present state. These are widely used in training gaming portals to work on user inputs accordingly.



**3.2 SOFTWARE DESIGN**

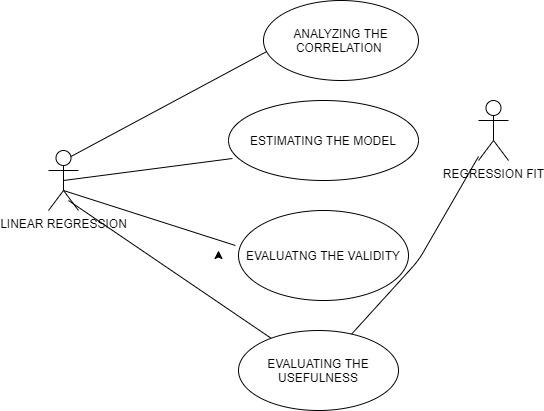
The software architecture of a system describes the structure, organization of components/modules, and their interactions not only to satisfy the systems’ functional and non-functional requirements but also to provide conceptual integrity to the overall system structure. Software architecture concerns the structure of large software-intensive systems. The architectural view is an abstract view that separates the details of implementation, algorithm, and data representation and concentrates on behavioral aspects and interaction among the various components. In other words, the software architecture of the system provides an abstract description of the system by exposing certain properties and hiding others. Hence software architecture plays an important function with respect to the following aspects in the development of large software-intensive systems.

**3.3 UML DIAGRAMS**

**3.3.1** **Use case diagram**: It captures the particular functionality of a system.

****

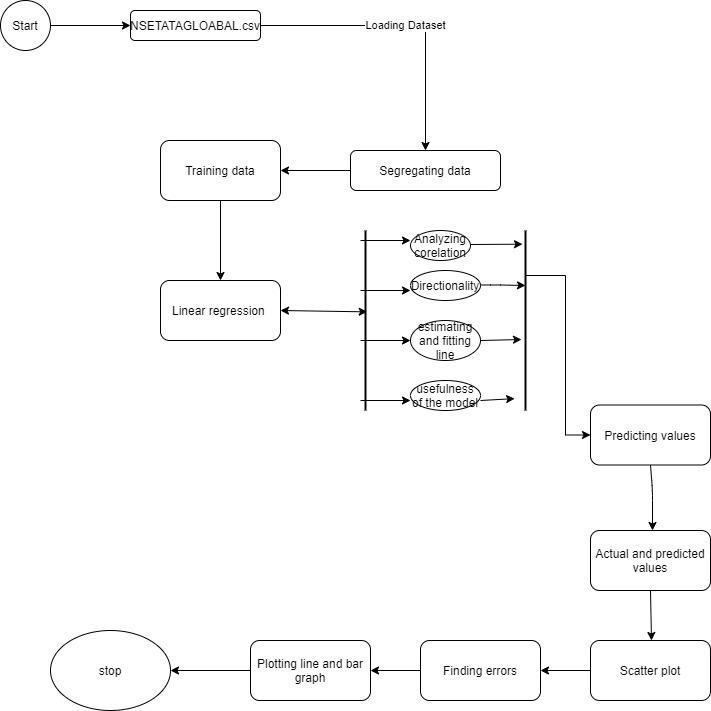
**Use case diagram for Linear Regression**



**3.3.2 Activity Diagram:** An activity diagram is basically a flowchart to represent the flow

from one activity to another activity. The activity can be described as an operation of

the system.

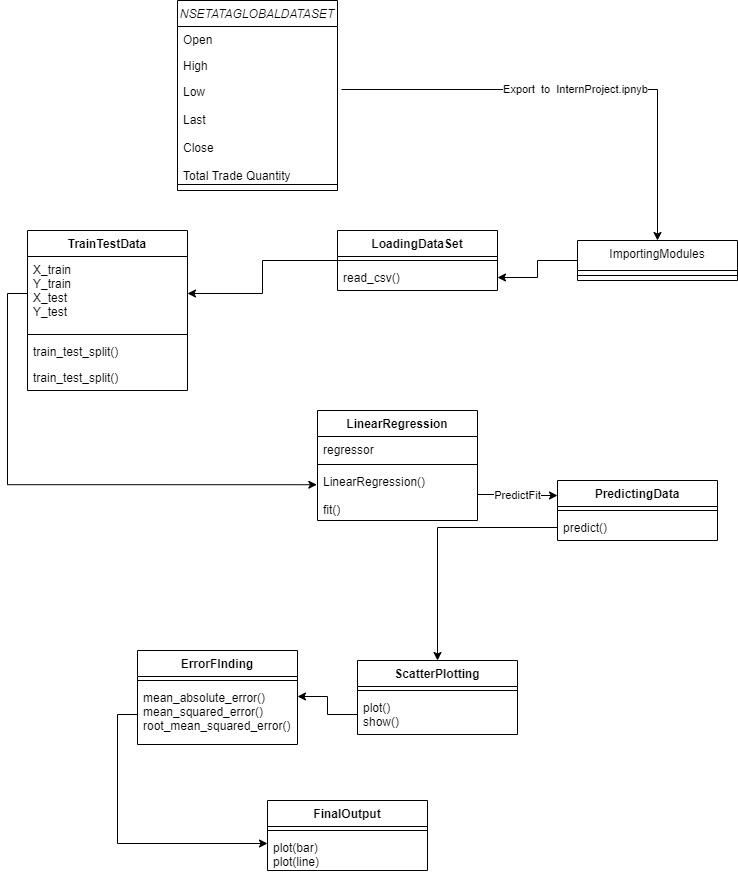
****

**3.3.3 Class Diagram:** A class diagram is a static diagram. It represents the static view

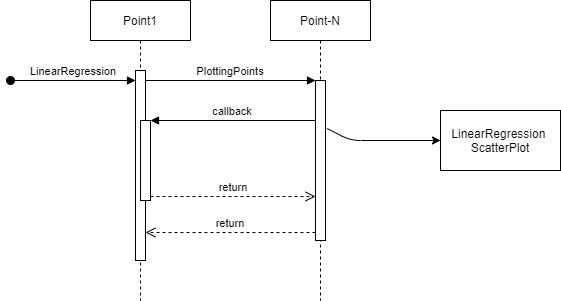
of an application. The class diagram is used for visualizing, describing, and

documenting different aspects of a system and constructing executable

code of the software application.

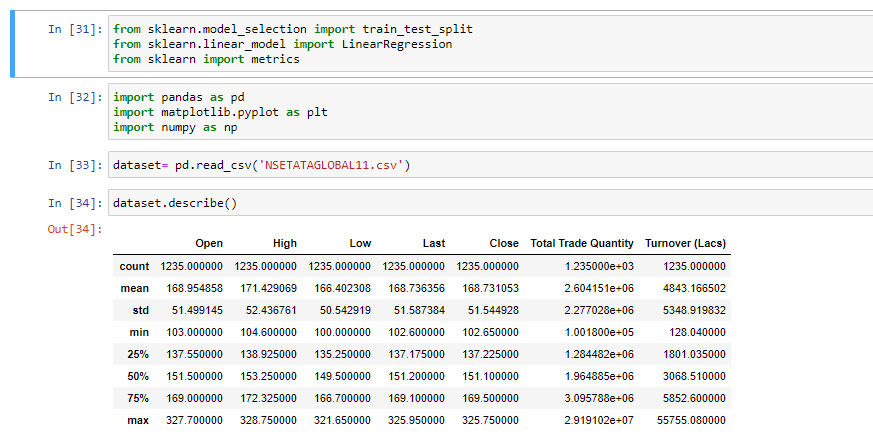


**3.3.4 Sequence Diagram:** A sequence diagram simply depicts the interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.

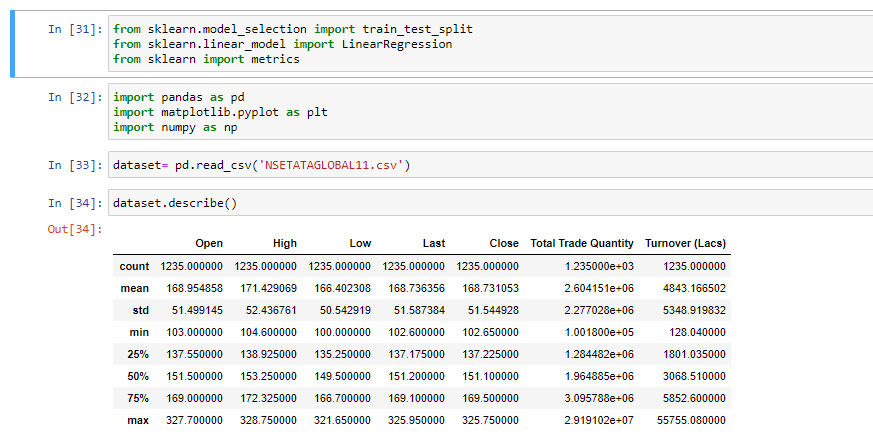


**4. PROJECT IMPLEMENTATION AND OUTPUT**

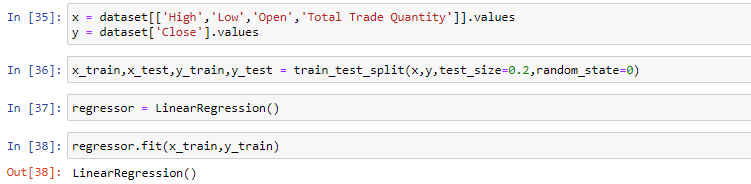
**Step 1: Importing modules**

****

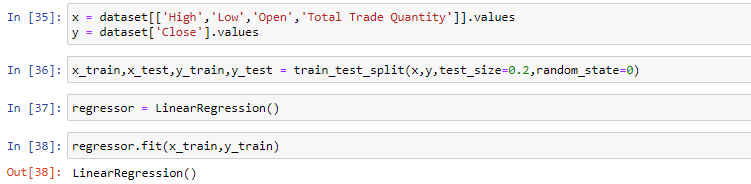
**Step 2: Importing Dataset**



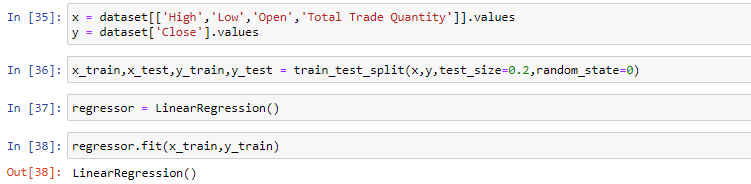
**Step 3: Declaring x and y values from the dataset**

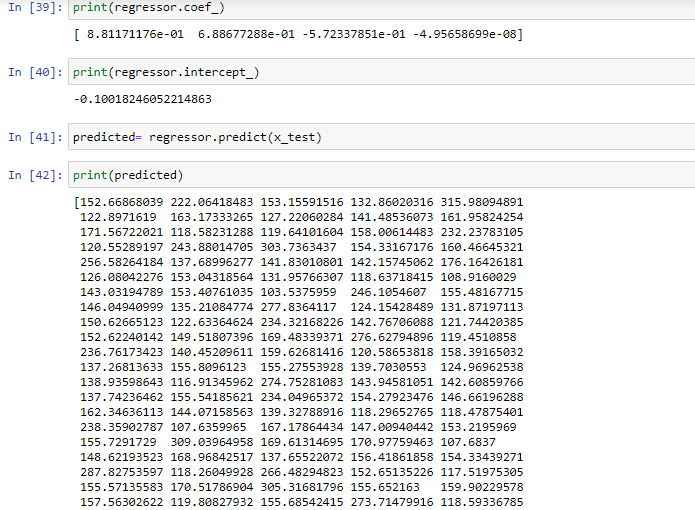


**Step 4: Training the dataset**

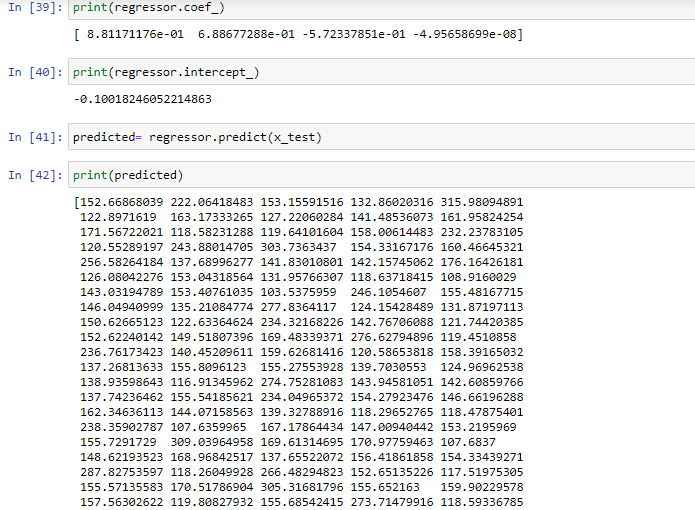
****

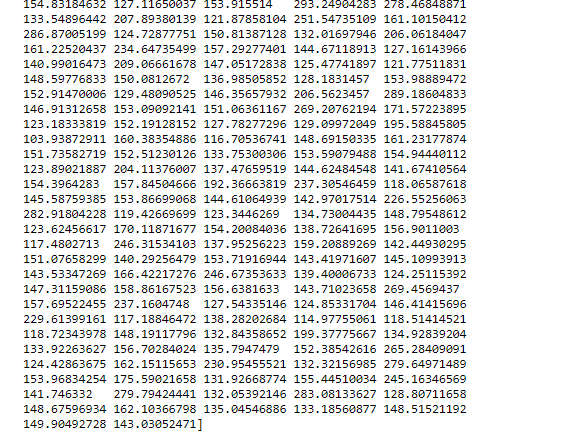
**Step 5: Applying Linear Regression**

****

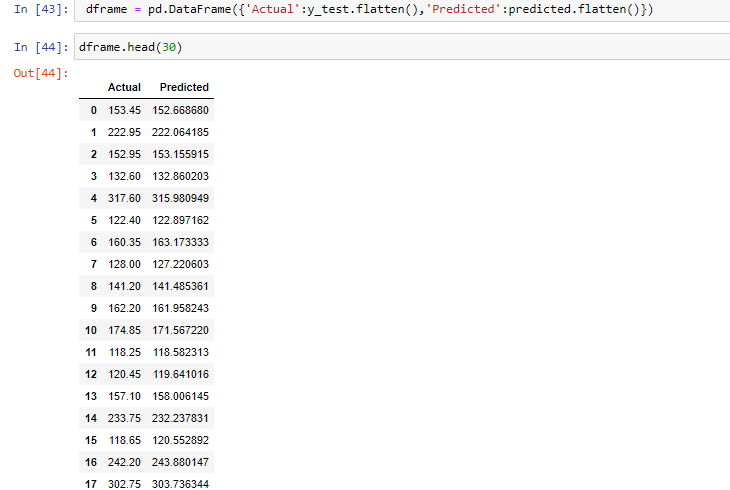
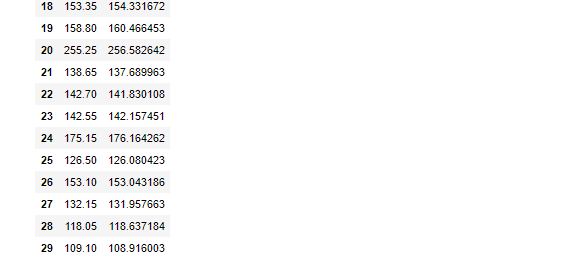
****

**Step 6: Predicting the regression set**

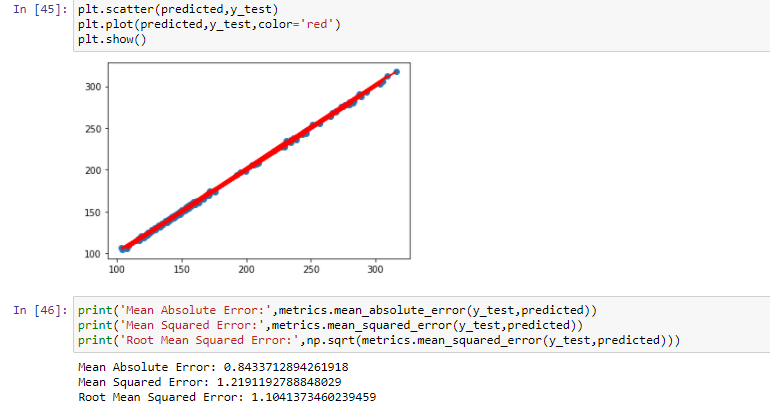
****

****

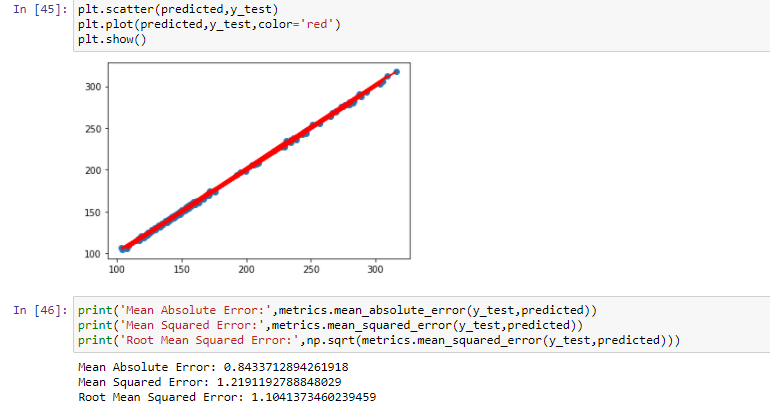
**Step 7: Finding the actual and predicted values**

** **

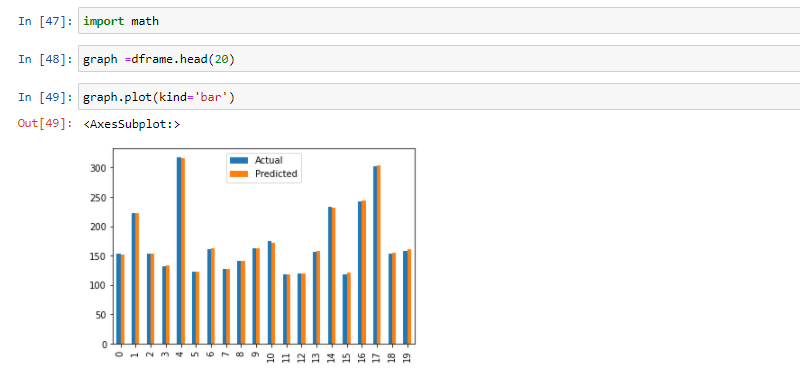
**Step 8:Plotting Scatter Plot**

****

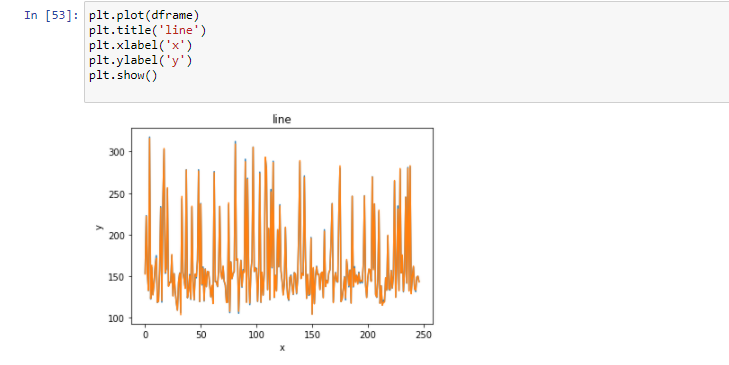
**Step 9:Finding Error**

****

**Step 10:Plot Bar Graph**

****

**Step 11:Plot Line Graph**

****

**5. APPLICATION**

* Mutual Fund investing.
* Trade analysis in businesses.
* Mobile applications for Stock Price Estimating.
* Used by the Government to get important estimations of national finance.
* Effective tool for Product and Service based companies.

**6. MERITS AND DEMERITS**

**MERITS**

* Energy efficient
* Less cost
* Time effective

**DEMERITS**

* The model might provide less accurate results if the dataset is not large.
* The accuracy of the model may vary from time to time based on the prevailing circumstances and other factors, which may ultimately provide misleading results, sometimes, when the factors are not properly dealt with.

**7. CONCLUSIONS**

By measuring the accuracy of the different algorithms, we found that the most suitable algorithm for predicting the market price of a stock based on various data points from historical data is the LSTM algorithm. 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 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.

**8. FUTURE ENHANCEMENT**

The 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 to analyzing the contents of public comments and thus determine patterns/relationships between the customer and the corporate employee. The use of traditional algorithms and data mining techniques can also help predict the corporation’s performance structure as a whole.

The algorithms yield more accurate results while working with a larger dataset, hence this is going to be very huge in the future which benefits both the customers and manufacturers in any company. Already machine learning algorithms are being implemented in the field of the stock market and the trend of machine learning and deep learning algorithms is expected to be huge in the near future.

**9. BIBLIOGRAPHY**

1. Pei-Yuan Zhou, Keith C.C. Chan, Member, IEEE, and Carol XiaojuanOu, “Corporate Communication Network and Stock Price Movements: Insights From Data Mining”, IEEE 2018.
2. <https://developers.google.com/machine-learning/crash-course/>
3. <https://www.statisticssolutions.com/how-to-conduct-linear-regression/>