TIME SERIES ANALYSIS & PREDICTIONS OF STOCK MARKET DATA USING DEEP LEARNING TECHNIQUES

A project work Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

KOMMOJU NAGA RAJESWAR ROHIT (20555A0509)

KANNURU KIRANMAYI (19551A0594)

SHAIK HAJI MASTAN VALI (19551A0561)

SAMMANGI GOPI CHAND (20555A0502)

Under the guidance of

K.V.K.SASIKANTH
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM, AP.

Jawaharlal Nehru Technological University, Kakinada, AP, India.

APRIL 2023

GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY

(Autonomous)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM, AP.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **"TIME SERIES ANALYSIS & PREDICTION OF STOCK MARKET DATA USING DEEP LEARNING TECHNIQUES**" is the Bonafide work of "**KOMMOJU NAGA RAJESWAR ROHIT (20555A0509), KANNURU KIRANAMAYI (19551A0594), SHAIK HAJI MASTAN VALI (19551A0561) , SAMMANGI GOPI CHAND (20555A0502)",** who carried out the project work under my supervision during the year 2022 to 2023, towards partial fulfillment of the requirements of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING of GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS), RAJAHMAHENDRAVARAM, AP.** The results embodied in this report have not been submitted to another university for the award of any degree.

Signature of the Head of the Department	Signature of the Supervisor
Dr. B. SUJATHA	**K.V.K.SASIKANTH**
Professor & Head of The Department	**Assistant Professor**
Department of Computer Science & Engineering	Department of Computer Science & Engineering

External Viva voice conducted on _____

Internal Examiner External Examiner

GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY

(Autonomous)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM, AP.

CERTIFICATION OF AUTHENTICATION

We solemnly declare that this project report **"TIME SERIES ANALYSIS & PREDICTION OF STOCK MARKET DATA USING DEEP LEARNING TECHNIQUES"** is the bonafide work done purely by us, carried out under the supervision of **K.V.K.SASIKANTH** towards partial fulfillment of the requirements of the Degree of Bachelor of Technology in Computer Science & Engineering as administered under the Regulations of Godavari Institute of Engineering & Technology, Rajamahendravaram, AP, India and award of the Degree from JawaharlalNehru Technological University, Kakinada during the year 2022- 2023.

We also declare that no part of this document has been taken up verbatim from any source without permission from the author(s)/publisher(s). Wherever few sentences, findings, images, diagrams or any other piece of information has been used for the sake of completion of this work, we have adequately referred to the document source. In the event of any issue arising hereafter about this work, we shall be personally responsible.

It is further certified that this work has not been submitted, either in part or full, to any other department of the Jawaharlal Nehru Technological University Kakinada, or any other University, institution or elsewhere, in India or abroad or for publication in any form.

NAME OF THE STUDENTS	SIGNATURE OF THE STUDENTS
KOMMOJU NAGA RAJESWAR ROHIT (20555A0509)	
KANNURU KIRANAMAYI (19551A0594)	
SHAIK HAJI MASTAN VALI (19551A0561)	
SAMMANGI GOPI CHAND (20555A0502)	

ACKNOWLEDGEMENT

We feel immense pleasure to express our sincere thanks and profound sense of gratitudeto all those people who played a valuable role for the successful completion of our project.

We would like to express my gratitude to **Dr. K.V.V. SATYANARAYANA RAJU**, chairman, Chaitanya group of Institutions, **Sri. K. SASI KIRAN VARMA**, Vice Chairman, GIET group of Institutions for providing necessary infrastructure.

We are thankful to Principal **Dr. P.M.M.S.SARMA** for permitting and encouraging for doing this project.

Our special thanks to **Dr. N. LEELAVATHY**, Vice Principal (Academics), for their content guidance and motivation and also for providing an excellent environment. We are truly grateful for her valuable suggestions and advice.

We are proudly grateful to express my deep sense of gratitude and respect towards **Dr. B. SUJATHA**, Professor and Head of the Department, Computer Science and Engineering, whose motivation and constant encouragement has led to pursue a project in the field of software development. We are very fortunate, for having her to enlighten us in all the aspects of life and transforming us to be an ideal individual in this field.

We are much obliged and thankful to our internal guide **K.V.K.SASIKANTH,** Professor and Head of the Department, Computer Science and Engineering, for providing this opportunity and constant encouragement given by him during the course. We are grateful to his valuable guidance and suggestions during my project work.

Our parents have put us ahead of themselves, because of their hard work and dedication, we have had opportunities beyond our wildest dreams. Our heart full thanks to them for giving us all support we needed to be successful student and individual.

Finally, we express our thanks to other professors, classmates, friends, neighbours and non-teaching staff who helped us for the completion of my project and without infinite love and patience this would never have been possible.

KOMMOJU NAGA RAJESWAR ROHIT (20555A0509)
KANNURU KIRANAMAYI (19551A0594)
SHAIK HAJI MASTAN VALI (19551A0561)
SAMMANGI GOPI CHAND (20555A0502)

INDEX

S.NO	CONTENTS	PAGE.NO
i.	**ABSTRACT**	
1.	**INTRODUCTION**	1
	1.1 INTRODUCTION OF PROJECT	1
	1.2 INTRODUCTION OF DOMAIN	2
	1.3 OBJECTIVE OF THE PROBLEM	4
	1.4 SCOPE OF THE PROJECT	4
2.	**LITERATURE SURVEY**	5
	2.1 LITERATURE SURVEY-1	5
	2.2 LITERATURE SURVEY-2	5
	2.3 LITERATURE SURVEY-3	5
	2.4 LITERATURE SURVEY-4	5
3.	**PROBLEM STATEMENT AND METHODOLOGY**	7
	3.1 PROBLEM DEFINITION	7
	3.2 METHODOLOGY	7
	3.2.1 EXISTING SYSTEM	7
	3.2.2 DISADVANTAGES OF EXISTING SYSTEM	8
	3.2.3 PROPOSED SYSTEM	9
	3.2.4 PROPOSED SYSTEM ADVANTAGES	9
	3.3 SYSTEM ARCHITECTURE	10

	3.4 UML DIAGRAMS	10
	3.4.1USE CASE DIAGRAM	11
	3.4.2 CLASS DIAGRAM	11
	3.4.3 DEPLOYMENT DIAGRAM	12
	3.4.4COMPONENT DIAGRAM	12
4.	**SYSTEM IMPLEMENTATION**	13
	4.1 MODULE DESCRIPTION	13
	4.2 FLOWCHART	14
	4.3 COMPARATIVE STUDY OF EXISTING AND PROPOSED SYSTEM	14
5.	**SYSTEM STUDY**	15
	5.1 FEASIBILTY STUDY	15
	5.2 ECONOMICAL FEASIBILITY	16
	5.3 TECHNICAL FEASIBILITY	16
	5.4 SOCIAL FEASIBILITY	16
	5.5 OPERATIONAL FEASIBILITY	17
6.	**SYSTEM TESTING**	18
	6.1 UNIT TESTING	18
	6.2 INTEGRATION TESTING	18
	6.3 FUNCTIONAL TESTING	19
	6.4 SYSTEM TESTING	19
	6.5 WHITE BOX TESTING	19
	6.6 BLACK BOX TESTING	20
	6.7 UNIT TESTING	20
	6.8 ACCEPTANCE TESTING	21

7.	**SYSTEM MAINTENANCE**	22
	7.1ABOUT THE SOFTWARE "PYTHON"	23
8.	**SOURCE CODE**	34
	8.1 STATIONARITY CHECK CODE	34
9.	**CONCLUSION AND FUTURE SCOPE**	46
	9.1 CONCLUSION	46
	9.2 FUTURE SCOPE	46
10.	**REFERENCES**	47

LIST OF FIGURES

S. No.	NAME OF THE FIGURE	Page. No
1	Stock Market Value	1
2	System Architecture	9
3	System UML Diagram	9
4	Use Case Diagram	10
5	Class Diagram	10
6	Deployment Diagram	11
7	Component Diagram	11
8	Detailed Flow Chart of Preparation of Our Project	13
9	Tesla Stock Market Price	37
10	Yearly Prediction	43
11	Accuracy Comparison	44

ABSTRACT

Stock market consists of various buyers and sellers. The stock market value is dynamic. It means the stock market value is changed day by day. Actually, stock has been represented as shares. The owner of the share may be an individual or group of peoples. In this current economic condition stock market value prediction is the critical task because the data is dynamic. Stock market prediction means to find the future value of the stock on a financial exchange. The expected prediction output to be accurate, efficient and robust value. Traditionally the stock values are predicted by using stock related news. But it does not provide a better result. Wrong prediction of stock value leads to heavy loss. Machine learning concepts play a very important role in various domains. It is also used to predict the stock market value with the help of collected data. This project describes about stock market value prediction using machine learning technique. This proposed concept is implemented by python programming language. This machine learning concept produces better prediction result compared with other machine learning techniques.

In our project we will be using the algorithms such as Decision Tree (DT) and Artificial Neural Network (ANN) and for analyzing our results.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION OF PROJECT:

Actually, traders buy the shares for less price and sales the same shares at high rate. Now stock market value prediction is one of the key research areas in current economic condition. Stock value analysis is divided into two categories. The first type is called fundamental analysis. In this analysis perform by using various parameters like political climate, current condition of the organization and economic level. The investors take their decision based upon the above-mentioned attributes. The second type of the analysis is technical analysis. Here the stock values are predicted by using various statistical values generated by the organization such as past price condition and volumes of data. Stock market data is a dynamic data. Every day the stock market value is changed. So, traditional methods are not able to produce better result. In this current scenario machine learning concepts are used in various fields. Traders are used various machine learning algorithms to predict the stock value. These algorithms are generating better result compared with traditional methods. Multivariate analysis with time series data is also used to predict the stock value. This proposed machine learning SVM concept uses historical data for predicting stock market value. The main aim of the proposed concept is gaining the market value with the help of machine learning approach. The following figure 1 shows the sample screen of stock market value.

Figure 1 Stock Market Value

The main objective of this proposed concept is to avoid unpredictable share value decision making.The stock market value is fluctuated condition.

1.2 INTRODUCTION OF DOMAIN:

Machine Learning:

In the statistical context, Machine Learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance. Machine Learning involves a high level of generalization in order to get a system that performs well on yet unseen data instances.

Machine learning is a relatively new discipline within Computer Science that provides a collection of data analysis techniques. Some of these techniques are based on well-established statistical methods (e.g. logistic regression and principal component analysis) while many others are not.

Most statistical techniques follow the paradigm of determining a particular probabilistic model that best describes observed data among a class of related models. Similarly, most machine learning techniques are designed to find models that best fit data (i.e. they solve certain optimization problems), except that these machine learning models are no longer restricted to probabilistic ones.

Therefore, an advantage of machine learning techniques over statistical ones is that the latter require underlying probabilistic models while the former do not. Even though some machine learning techniques use probabilistic models, the classical statistical techniques are most often too stringent for the oncoming Big Data era, because data sources are increasingly complex and multi-faceted. Prescribing probabilistic models relating variables from disparate data sources that are plausible and amenable to statistical analysis might be extremely difficult if not impossible.

Machine learning might be able to provide a broader class of more flexible alternative analysis methods better suited to modern sources of data. It is imperative for statistical agencies to explorethe possibleuse of machinelearningtechniques to determine whethertheir future needs might bebetter met with such techniques than with traditional ones.

Classes of Machine Learning:

There are two main classes of machine learning techniques:

1. Supervised machine learning
2. Unsupervised machine learning.

Examples of supervised learning Logistic regression (statistics) vs Support vector machines (machine learning) Logistic regression, when used for prediction purposes, is an example of supervised machine learning. In logistic regression, the values of a binary response variable (withvalues 0 or 1, say) as well as a number of predictor variables (covariates) are observed for a numberof observation units. These are called training data in machine learning terminology. The main hypotheses are that the response variable follows a Bernoulli distribution (a class of probabilistic models), and the link between the response and predictor variables is the relation that the logarithm of the posterior odds of the response is a linear function of the predictors. The response variables of the units are assumed to be independent of each other, and the method of maximum likelihood is applied to their joint probability distribution to find the optimal values for the coefficients (Theseparameterize the aforementioned joint distribution) in this linear function. The particular model with these optimal coefficient values is called the "fitted model," and can be used to "predict" thevalue of the response variable for a new unit (or, "classify" the new unit as 0 or 1) for which onlythe predictor valuesare known.

Other supervised machine learning techniques mentioned later in this briefing include decision trees, neural networks, and Bayesian networks. Examples of unsupervised learning Principal component analysis (statistics) vs Cluster analysis (machine learning). The main example of an unsupervised machine learning technique that comes from classical statistics is principal component analysis, which seeks to "summarize" a set of data points in high-dimensional space by finding orthogonal one-dimensional subspaces along which most of the variation in the data points is captured. The term "unsupervised" simply refers to the fact that there is no longer a response variable in the current setting. Cluster analysis and association analysis are examples of non-statistical unsupervised machine learning techniques. The former seeks to determine inherentgrouping structure in given data, whereas the latter seeks to determine co-occurrence patterns of items.

1.3 OBJECTIVE OF THE PROBLEM:

Machine learning approaches and artificial intelligence concepts are being used in combination with data mining techniques to find the solution of real-world problems. This technique provides better accuracy with minimum investment value. This method reduces large amount of time for predicting stock market value. Now most of the peoples invest their annual income into the sharemarket. With the help of proper guidance, the investors can able to yield high profit. But most of the investors said stock investment is risky compared with other investments. In this project we proposed a new machine learning model by using various dataset. The closing stock values are also included in this proposed system. This proposed system definitely helpful to the various investors.

1.4 SCOPE OF THE PROJECT:

ANN is the new learning algorithm to control the decision. In this project we presented a hypothetical and experiential architecture to implement ANN concept to predict the stock marketvalue. Initially four organization economic factors are considered for multivariate approach. Next, ANN concept was used to analyze the association among the factors and predict the stock marketvalue.

The main scopes of the project are as follows:

1. To find out the accuracy in predicting stock market.
2. To find out detail study of each company stocks.
3. Our work should be applicable to all datasets.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY-1:

The ARIMA model, originally presented by Box and Jenkins in 1970, is a method for forecasting time series data. This technique is widely used in industries such as food production to estimate demand and support supply chain decisions. ARIMA can also be applied to predict future stock prices using historical data [10]. While the model may not be effective in handling time series with significant seasonal patterns, its accuracy may decrease for long-term projections.

2.2 LITERATURE SURVEY-2:

Hochreiter and Schmid Huber introduced Long Short-Term Memory (LSTM) networks in 1997 and since then, they have achieved impressive accuracy results across various fields of application. In 2007, LSTMs networks had a significant impact on the field of speech recognition by producing better results than conventional models in certain speech applications [2]. In 2009, an LSTM network trained using Connectionist Temporal Classification (CTC) dominated in multiple handwriting recognition contests, marking the first time an RNN had taken first place in a pattern recognition competition. In 2014, Baidu, the Chinese search giant, achieved a remarkable feat by surpassing the Switchboard Hub5'00 speech recognition benchmark with CTC-trained RNNs, without relying on any standard speech processing techniques.

2.3 LITERATURE SURVEY -3:

Support Vector Machines (SVM) were first proposed by Boser in 1992. The major goal of SVM is to detect the optimal hyperplane that maximizes the margin between two classes [3]. Time series analysis is a crucial and challenging area in both statistics and machine learning, often involving large and multi-dimensional time series data. Although SVM is commonly utilized for analyzing specific types of data sets [8], it may not perform well when dealing with complex datasets with a high level of noise or when the target classes overlap.

2.4 LITERATURE SURVEY -4:

Friedman and Richard Olshen of Stanford University in 1977 [4]. It is a decision support tool that utilizes a tree-based model to depict different options and their possible outcomes such as utility, costs, and risks. The algorithm behind CART is based on conditional statements only. In this method, each terminal node represents a class label, internal nodes represent evaluations based on attributes, and the branches represent the outcome of these evaluations.

The decision tree displays the classification rules as the route from the root to the leaf nodes. However, this method may not be appropriate for modeling time-series data with a straightforward linear trend or strong seasonality, as it fails to perform well in these scenarios.

CHAPTER 3

PROBLEM STATEMENT AND METHODOLOGY

3.1 PROBLEM DEFINITION:

The main challenges in order to develop the next generation of intelligent Systems are: -

- Slowing down systems, crashing a system, and its lack of scalability.
- System sent irregular messages it would potentially lead to the detection of the scam.
- Less security.
- Time consumption.
- Common people can be fooled at any time.

3.2 METHODOLOGY:

Stock market values are dynamic in nature. So, the prediction of stock value is very difficult task.Traditionally stock market value has been predicted by using the help of stock related news. Aftercertain period computing concepts are used to predict stock market value. Now machine learning approaches are used to predict the stock market value in better way. The input data has been captured from the company historical value. Then the data has been preprocessed by using various data mining concepts. The main usage preprocessing is to remove unwanted data from the original data. After the selection of features the machine learning concept is applied to classify the data.

3.2.1 EXISTING SYSTEM:

Decision Tree (DT):

A **decision tree** is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are commonly used in operations research, specifically in decision analysis, to helpidentify a strategy most likely to reach a goal, but are also a popular tool in machine learning. A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).The paths from root to leaf represent classification rules.

In decision analysis, a decision tree and the closely related influence diagram are used as a visualand analytical decision support tool, where the expected values (or expected utility) of competingalternatives are calculated.

A decision tree consists of three types of nodes:

1. Decision nodes – typically represented by squares
2. Chance nodes – typically represented by circles
3. End nodes – typically representedby triangles

Decision trees are commonly used in operations research and operations management. If, in practice, decisions have to be taken online with no recall under incomplete knowledge, a decisiontree should be paralleled by a probability model as a best choice model or online selection model algorithm. Another use of decision trees is as a descriptive means for calculating conditionalprobabilities.

Decision trees, influence diagrams, utility functions, and other decision analysis tools and methods are taught to undergraduate students in schools of business, health economics, and publichealth, and are examples of operations research or science methods.

3.2.2 DISADVANTAGES OF EXISTING SYSTEM:

- Less Accuracy in predicting the stock market.
- Not user friendly model.
- More time consumingprocess
- Higher Computational Cost
- Lack of standards
- Cannot be implemented in all datasets.

3.2.3 PROPOSED SYSTEM:

Artificial Neural Network (ANN):

An Artificial Neural Network (ANN) is a class of neural networks where connections between nodes form a graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feed forward neural networks, ANNs can use their internal state (memory)to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feed forward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled. Both finite impulse and infinite impulse recurrent networks can have additional stored states, andthe storage can be under direct control by the neural network. The storage can also be replaced byanother network or graph, if that incorporates time delays or has feedback loops. Such controlledstates are referred to as gated state or gated memory, and are part of memory networks (LSTMs) and gated recurrent units. This is also called Feedback Neural Network (FNN).

3.2.4 PROPOSED SYSTEM ADVANTAGES:

- High output efficiency.
- User friendly.
- Less time consumption.
- Can be implemented in all datasets.
- Early prediction of stock rate is possible.

3.3 SYSTEM ARCHITECTURE:

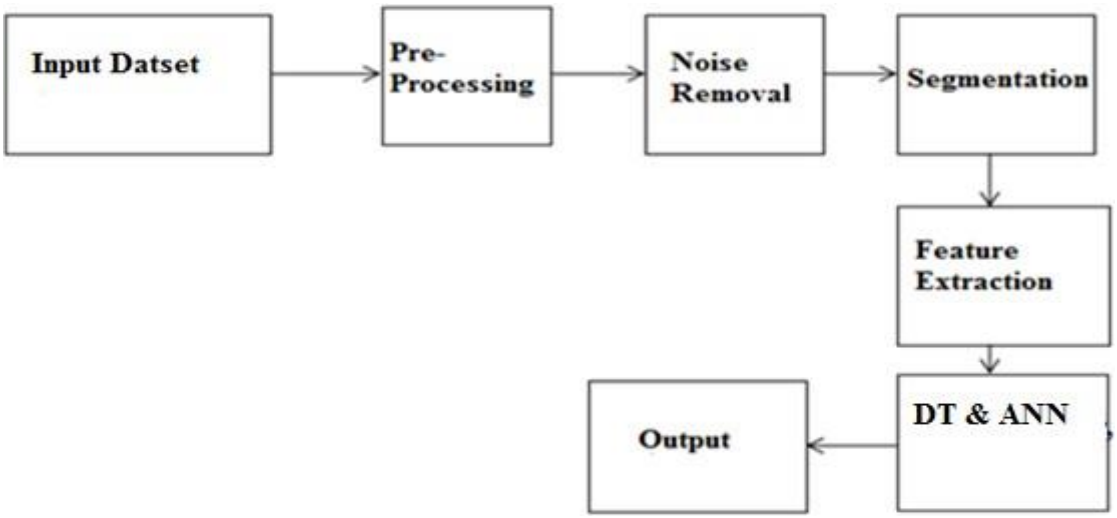

Figure 2 System Architecture

3.4 UML DIAGRAMS

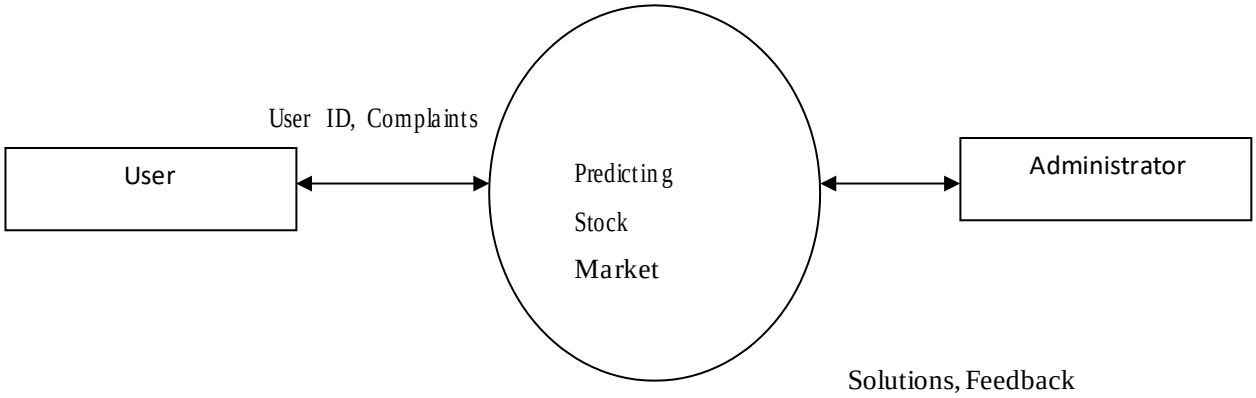

Figure 3 System UML Diagram

3.4.1 USE CASE DIAGRAM:

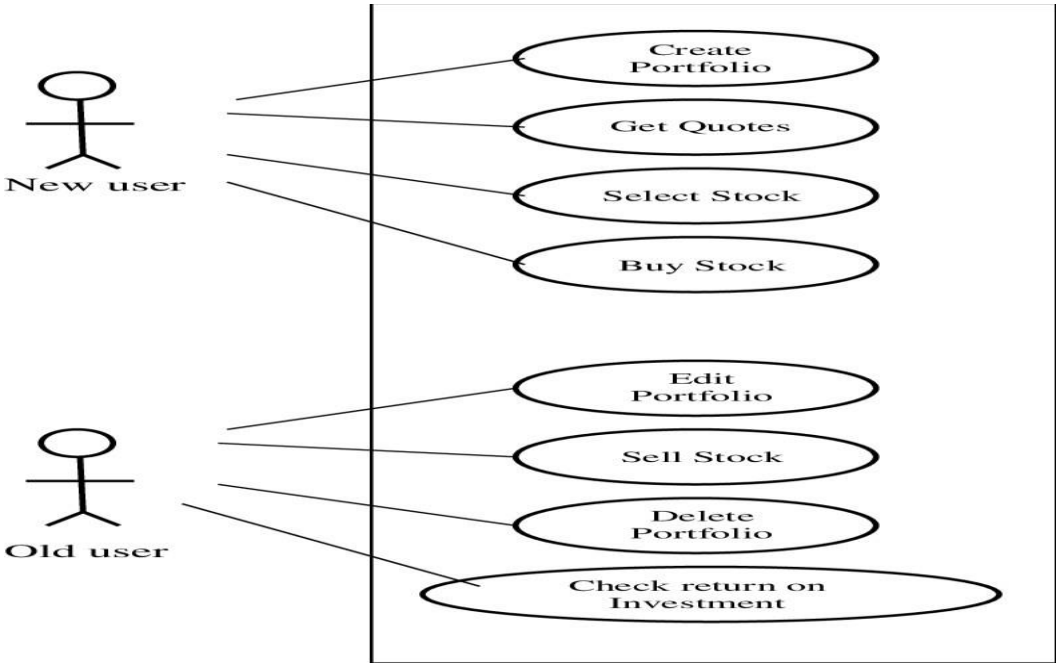

Figure 4 Use Case Diagram

3.4.2 CLASS DIAGRAM:

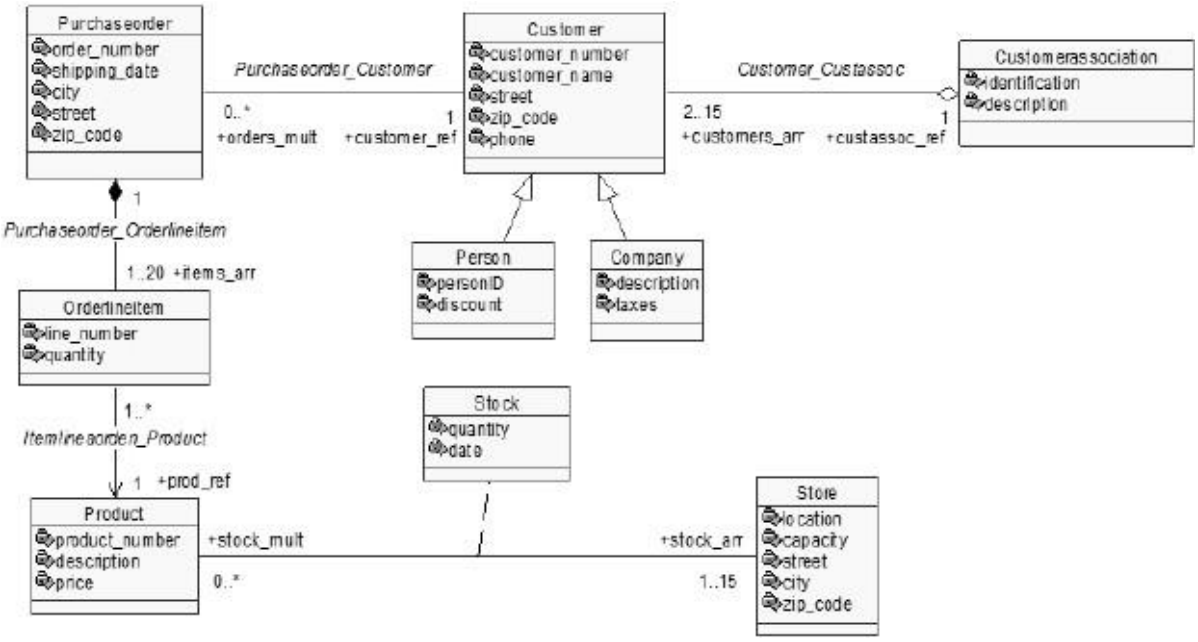

Figure 5: Class Diagram

3.4.3 DEPLOYMENT DIAGRAM:

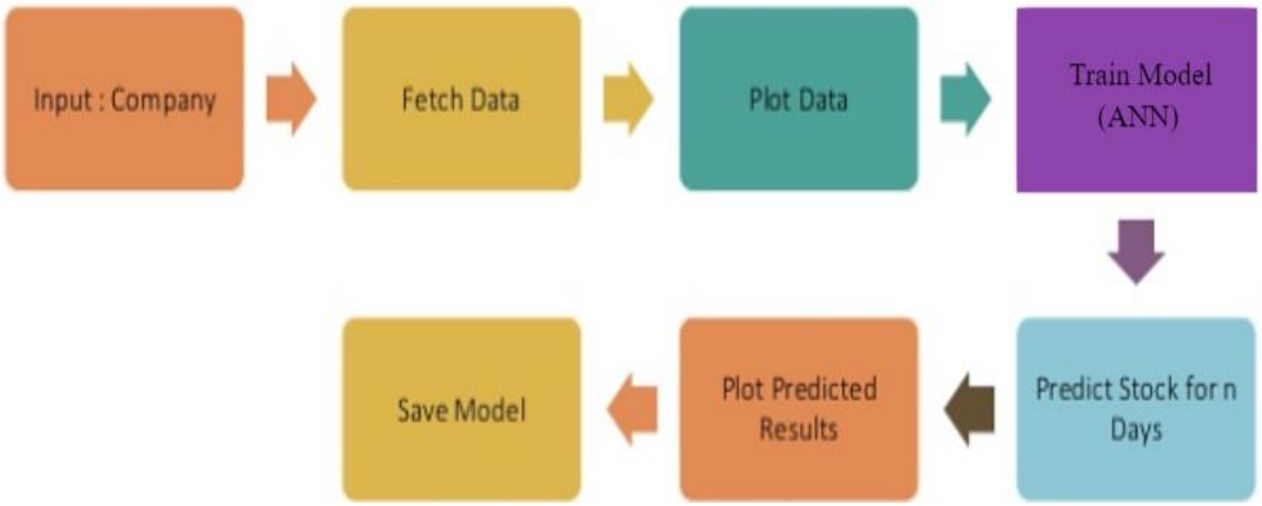

Figure 6 Deployment Diagram

3.4.4 COMPONENT DIAGRAM:

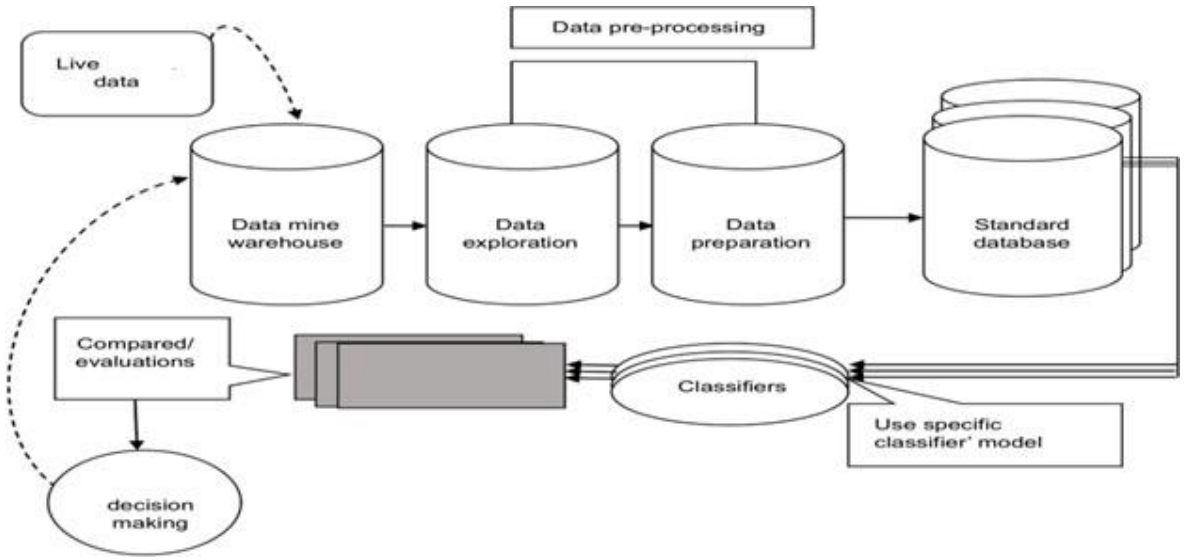

Figure 7 Component Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

1. Input dataset
2. Analysis of size of data set.
3. Oversampling.
4. Training and Testing.
5. Apply algorithms.
6. Predict results.

Input dataset:

Dataset can be taken from online source provider called UCI resipotary. We have collected set ofstock market datasets which we are going to analyze. Then for training the data set also for the comparison of the non stock datasets are also been taken.

Analysis of data set:

Here the analysis if dataset takes place. The size of data is taken into consideration for the data process.

Oversampling (Using SMOTE):

we have created a detailed history of all stocks that been doneover a certain amount of time and it is sampled to fix a threshold value.

Training and Testing Subset:

As the dataset is imbalanced, many classifiers show bias for majority classes. The features of minority class are treated as noise and are ignored. Hence it is proposed to select a sample dataset.

Applying algorithm:

Following are the classification algorithms used to test the sub-sampledataset.

a. Decision Tree (DT)

b. Artificial Neural Network (ANN)

Predicting results:

The test subset is applied on the trained model. The metrices used isaccuracy. The ROC Curve is plotted and the desirable results are achieved.

4.2 FLOWCHART:

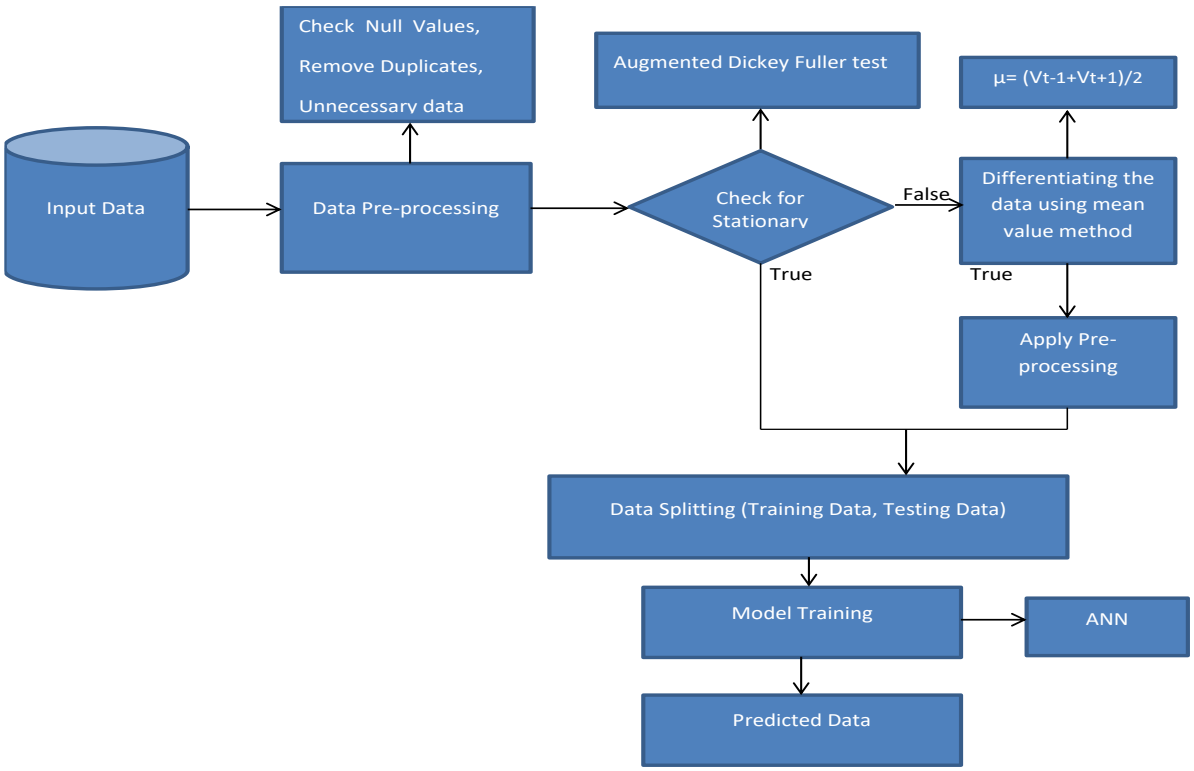

Figure 8 Detailed Flow Chart of Preparation of Our Project

4.3 COMPARATIVE STUDY OF EXISTING AND PROPOSED SYSTEM:

In our project we will be using the algorithms such as Decision Tree (DT) and Artificial Neural Network (ANN) for analyzing our results. In our proposed system we will be using the proposed ANN technique method which leads to high accuracy compared to existing system. From this weare getting less noise ratio and good accuracy so we can state that our proposed system works better than the existing system.

CHAPTER 5

SYSTEM STUDY

5.1 FEASIBILTY STUDY:

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

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution. The analystconducting the study gathers information using a variety of methods, the most popular of which are:

- Developing and administering questionnaires to interested stakeholders, such as potentialusers of the information system.
- Observing or monitoring users of the current system to determine their needs as well astheir satisfaction and dissatisfaction with the current system.
- Collecting, examining, and analyzing documents, reports, layouts, procedures, manuals,and any other documentation relating to the operations of the current system.
- Modeling, observing, and simulating the work activities of the current system.

The goal of the feasibility study is to consider alternative information systems solutions, evaluatetheir feasibility, and propose the alternative most suitable to the organization. The feasibility of aproposed solution is evaluated in terms of its components. These components are:

- Economical feasibility
- Technical feasibility
- Social feasibility
- Operational feasibility

5.2 ECONOMICAL FEASIBILITY:

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

5.3 TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is the technical requirements of thesystem. Any system developed must not have a high demand on the available technical resources.This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimalor null changes are required for implementing this system.

5.4 SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includesthe process of training the user to use the system efficiently. The user must not feel threatened bythe system, instead must accept it as a necessity.

The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5.5 OPERATIONAL FEASIBILITY:

The ability, desire, and willingness of the stakeholders to use, support, and operate the proposed computer information system. The stakeholders include management, employees, customers, and suppliers. The stakeholders are interested in systems that are easy to operate, makefew, if any, errors, produce the desired information, and fall within the objectives of the organization.

CHAPTER 6

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTING:

6.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2 INTEGRATION TESTING:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcomeof screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.3 FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: Identified classes of valid input must be accepted.

Invalid Input: Identified classes of invalid input must be rejected.

Functions: Identified functions must be exercised.

Output: Identified classes of application outputs must be exercised.

Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Beforefunctional testing is complete, additional tests are identified and the effective value of current testsis determined.

6.4 SYSTEM TESTING:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions andflows, emphasizing pre-driven process links and integration points.

6.5 WHITE BOX TESTING:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.6 BLACK BOX TESTING:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, Such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

6.7 UNIT TESTING:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.8 ACCEPTANCE TESTING:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 7

SYSTEM MAINTENANCE

Python is a remarkably powerful dynamic, object-oriented programming language that is used in a wide variety of application domains. It offers strong support for integration with other languages and tools, and comes with extensive standard libraries. To be precise, the following are some distinguishing features of Python:

- Very clear, readable syntax.
- Strong introspection capabilities.
- Full modularity.
- Exception-based error handling.
- High level dynamic data types.
- Supports object oriented, imperative and functional programming styles.
- Embeddable.
- Scalable
- Mature

With so much of freedom, Python helps the user to think problem centric rather than language centric as in other cases. These features make Python a best option for scientific computing.

Open CV

Open CV is a library of programming functions for real time computer vision originally developed by Intel and now supported by Willogarage. It is free for use under the open source BSD license. The library has more than five hundred optimized algorithms. It is used around the world, with forty thousand people in the user group. Uses range from interactive art, to mine inspection, and advanced robotics. The library is mainly written in C, which makes it portable to some specific platforms such as Digital Signal Processor. Wrappers for languages

such as C, Python, Ruby and Java (using Java CV) have been developed to encourage adoption by a wider audience. The recent releases have interfaces for C++. It focuses mainly on real-time image processing. Open CV is a cross-platform library, which can run on Linux, Mac OS and Windows. To date, Open CV is the best open source computer vision library that developers and researcher scan think off.

Tesseract

Tesseract is a free software OCR engine that was developed at HP between 1984 and 1994. HP released it to the community in 2005. Tesseract was introduced at the 1995 UNLV Annual TestOCR Accuracy and is currently developed by Google released under the Apache License. It can now recognize 6 languages, and is fully UTF8 capable. Developers can train Tesseract with their own fonts and character mapping to obtain perfect efficiency.

7.1 ABOUT THE SOFTWARE "PYTHON":

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-

68,SmallTalk, and 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.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python - Environment Setup

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Local-Environment Setup

Open a terminal window and type "python" to find out if it is already installed and which versionis installed.

- Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
- Win 9x/NT/2000
- Macintosh (Intel, PPC, 68K)
- OS/2
- DOS (multiple versions)
- PalmOS
- Nokia mobile phones
- Windows CE
- Acorn/RISC OS
- BeOS
- Amiga
- VMS/OpenVMS
- QNX
- VxWorks
- Psion
- Python has also been ported to the Java and .NET virtual machines

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available onthe official website of Python https://www.python.org/

You can download Python documentation https://www.python.org/doc/The documentation is available in HTML, PDF, and PostScript formats.

Installing Python

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the sourcecode manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platforms −Unix and Linux Installation

Here are the simple steps to install Python on Unix/Linux machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link to download zipped source code available for Unix/Linux.
- Download and extract files.
- Editing the *Modules/Setup* file if you want to customize some options.
- Run./configure script
- make
- make install

Windows Installation:

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version you need to install.

- To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if yourmachine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy touse. Just accept the default settings, wait until the install is finished, and you are done.

Mac in tosh Installation:

Recent Macs come with Python installed, but it may be several years out of date. See http://www.python.org/download/mac/ for instructions on getting the current version along with extra tools to support development on the Mac. For older Mac OS's before Mac OS X 10.3 (released in 2003), MacPython is available. Jack Jansen maintains it and you can have full access to the entire documentation at his website

− http://www.cwi.nl/~jack/macpython.html. You can find complete installation details for MacOS installation.

Setting up PATH:

Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.

The **path** variable is named as PATH in Unix or Path in Windows (Unix is case sensitive; Windows is not).

In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you must add the Python directory to your path.

Setting path at Unix/Linux:

To add the Python directory to the path for a particular session in Unix −

- **In the csh shell** − type setenv PATH "$PATH:/usr/local/bin/python" and press Enter.
- **In the bash shell (Linux)** − type export PATH="$PATH:/usr/local/bin/python" and pressEnter.
- **In the sh or ksh shell** − type PATH="$PATH:/usr/local/bin/python" and press Enter.
- **Note** − /usr/local/bin/python is the path of the Python directorySetting path at Windows

To add the Python directory to the path for a particular session in Windows −

At the command prompt − type path %path%;C:\Python and press Enter.

Note − C:\Python is the path of the Python directory Python Environment variables
Here are important environment variables, which can be recognized by Python −

S.No	Variable & Description
1	**PYTHONPATH** It has a role similar to PATH. This variable tells the Python interpreter where to locate the module files imported into a program. It should include the Python source library directory and the directories containing Python source code. PYTHONPATH is sometimes preset by the Python installer.
2	**PYTHONSTARTUP** It contains the path of an initialization file containing Python source code. It is executed every time you start the interpreter. It is named as .pythonrc.py in Unix and it contains commands that load utilities or modify PYTHONPATH.
3	**PYTHONCASEOK** It is used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.
4	**PYTHONHOME** It is an alternative module search path. It is usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.

Running Python:

There are three different ways to start Python −

Interactive Interpreter

You can start Python from Unix, DOS, or any other system that provides you a command-line interpreter or shell window.

Enter **python** the command line.

Start coding right away in the interactive interpreter.

```
$python #
Unix/Linuxor
python% #
Unix/Linuxor
C:> python # Windows/DOS
```

Here is the list of all the available command line options −

Sr.No.	Option & Description
1	**-d** It provides debug output.
2	**-O** It generates optimized bytecode (resulting in .pyo files).
3	**-S** Do not run import site to look for Python paths on startup.
4	**-v** verbose output (detailed trace on import statements).

5	**-X** disable class-based built-in exceptions (just use strings); obsolete starting with version 1.6.
6	**-c cmd** run Python script sent in as cmd string
7	**File** run Python script from given file

Script from the Command-line

A Python script can be executed at command line by invoking the interpreter on your application,as in the following −

Note − Be sure the file permission mode allows execution.

```
$python script.py # Unix/Linux

or

python% script.py # Unix/Linux

or

C: >python script.py # Windows/DOS
```

Integrated Development Environment

You can run Python from a Graphical User Interface (GUI) environment as well, if you have aGUI application on your system that supports Python.

- **Unix** − IDLE is the very first Unix IDE for Python.
- **Windows** − PythonWin is the first Windows interface for Python and is an IDE with a GUI.

- **Macintosh** − The Macintosh version of Python along with the IDLE IDE is available fromthe main website, downloadable as either MacBinary or BinHex'd files.

If you are not able to set up the environment properly, then you can take help from your system admin. Make sure the Python environment is properly set up and working perfectly fine.

Note − All the examples given in subsequent chapters are executed with Python 2.4.3 version available on CentOS flavor of Linux.

We already have set up Python Programming environment online, so that you can execute all the available examples online at the same time when you are learning theory. Feel free to modify anyexample and execute it online.

PANDA:

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-usedata structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

Audience

This tutorial has been prepared for those who seek to learn the basics and various functions of Pandas. It will be specifically useful for people working with data cleansing and analysis. After completing this tutorial, you will find yourself at a moderate level of expertise from where you can take yourself to higher levels of expertise.

Prerequisites:

You should have a basic understanding of Computer Programming terminologies. A basic understanding of any of the programming languages is a plus. Pandas library uses most of the functionalities of NumPy. It is suggested that you go through our tutorial on NumPy before proceeding with this tutorial. You can access it from Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can

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

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

Key Features of Pandas

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

Standard Python distribution doesn't come bundled with Pandas module.

NumPy:

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical andlogical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

Audience:

This tutorial has been prepared for those who want to learn about the basics and various functionsof NumPy. It is specifically useful for algorithm developers. After completing this tutorial, you will find yourself at a moderate level of expertise from where you can take yourself to higher levels of expertise.

Prerequisites:

You should have a basic understanding of computer programming terminologies. A basic understanding of Python and any of the programming languages is a plus. NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array. Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this opensource project.

Operations using NumPy

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy – A Replacement for MatLab

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat−plotlib** (plotting library). This combination is widely used as a replacement for MatLab,a popular platform for technical computing. However, Python alternative to MatLab is now seenas a more modern and complete programming language.

It is open source, which is an added advantage of NumPy.tandard Python distribution doesn't come bundled with NumPy module. A lightweight alternative is to install NumPy using popular Python package installer, **pip**.

```
pip install numpy
```

The best way to enable NumPy is to use an installable binary package specific to your operating system. These binaries contain full SciPy stack (inclusive of NumPy, SciPy, matplotlib, IPython, SymPy and nose packages along with core Python).

CHAPTER 8

SOURCE CODE

8.1 STATIONARITY CHECK CODE:

```
import numpy as np
import pandas as pd
import datetime as dt
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
import warnings
warnings.filterwarnings('ignore')
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)
```

Preprocessing of the data:

```
df = pd.read_csv("Stocks/tsla.us.txt")
from statsmodels.tsa.stattools import adfuller
def adtest(dataset):
    dftest = adfuller(dataset, autolag="AIC")
    print("1. ADF : ", dftest[0])
    print("2. P-VALUES : ", dftest[1])
    print("3. NUM OF LAGS : ", dftest[2])
    print("4. NUM OF OBSERVATIONS USED FOR THE ADF REGRESSION AND
CRITICAL VALUES CALCULATION : ", dftest[3])
    print("5. CRITICAL VALUES : ")
    for key, val in dftest[4].items():
        print("\t", key, ": ",val)
```

[out]

```
1. ADF :  -2.8584722829948017
2. P-VALUES :  0.05039461134316187
3. NUM OF LAGS :  23
4. NUM OF OBSERVATIONS USED FOR THE ADF REGRESSION AND CRITICAL VALUES CALCULATION :  1834
5. CRITICAL VALUES :
         1% :  -3.4339205977576532
         5% :  -2.863117219476073
         10% :  -2.5676096573745166
```

df.head()

[out]

	Date	Open	High	Low	Close	Volume	OpenInt
0	2010-06-28	17.00	17.00	17.00	17.00	0	0
1	2010-06-29	19.00	25.00	17.54	23.89	18783276	0
2	2010-06-30	25.79	30.42	23.30	23.83	17194394	0
3	2010-07-01	25.00	25.92	20.27	21.96	8229863	0
4	2010-07-02	23.00	23.10	18.71	19.20	5141807	0

df.tail()

[out]

	Date	Open	High	Low	Close	Volume	OpenInt
1853	2017-11-06	307.00	307.50	299.01	302.78	6482486	0
1854	2017-11-07	301.02	306.50	300.03	306.05	5286320	0
1855	2017-11-08	305.50	306.89	301.30	304.31	4725510	0
1856	2017-11-09	302.50	304.46	296.30	302.99	5440335	0
1857	2017-11-10	302.50	308.36	301.85	302.99	4621912	0

```
print(df.quantile([0, 0.05, 0.50, 0.95, 0.99, 1]).T)
print(df.isnull().sum())
print(df.shape)
print(df.dtypes)
df["Date"]=pd.to_datetime(df["Date"])
df.head()
```

[out]

	Date	Open	High	Low	Close	Volume	OpenInt
0	2010-06-28	17.00	17.00	17.00	17.00	0	0
1	2010-06-29	19.00	25.00	17.54	23.89	18783276	0
2	2010-06-30	25.79	30.42	23.30	23.83	17194394	0
3	2010-07-01	25.00	25.92	20.27	21.96	8229863	0
4	2010-07-02	23.00	23.10	18.71	19.20	5141807	0

```
Stock_df= df[["Date","Close"]]
Stock_df.head()
```

[out]

	Date	Close
0	2010-06-28	17.00
1	2010-06-29	23.89
2	2010-06-30	23.83
3	2010-07-01	21.96
4	2010-07-02	19.20

```
print("Min. Date:",Stock_df["Date"].min())
print("Max. Date:",Stock_df["Date"].max())
```

```
Min. Date: 2010-06-28 00:00:00
Max. Date: 2017-11-10 00:00:00
```

```
Stock_df.index=Stock_df["Date"]
Stock_df
```

[out]

	Date	Close
Date		
2010-06-28	2010-06-28	17.00
2010-06-29	2010-06-29	23.89
2010-06-30	2010-06-30	23.83
2010-07-01	2010-07-01	21.96
2010-07-02	2010-07-02	19.20
...	...	...
2017-11-06	2017-11-06	302.78
2017-11-07	2017-11-07	306.05
2017-11-08	2017-11-08	304.31
2017-11-09	2017-11-09	302.99
2017-11-10	2017-11-10	302.99

1858 rows × 2 columns

[out]

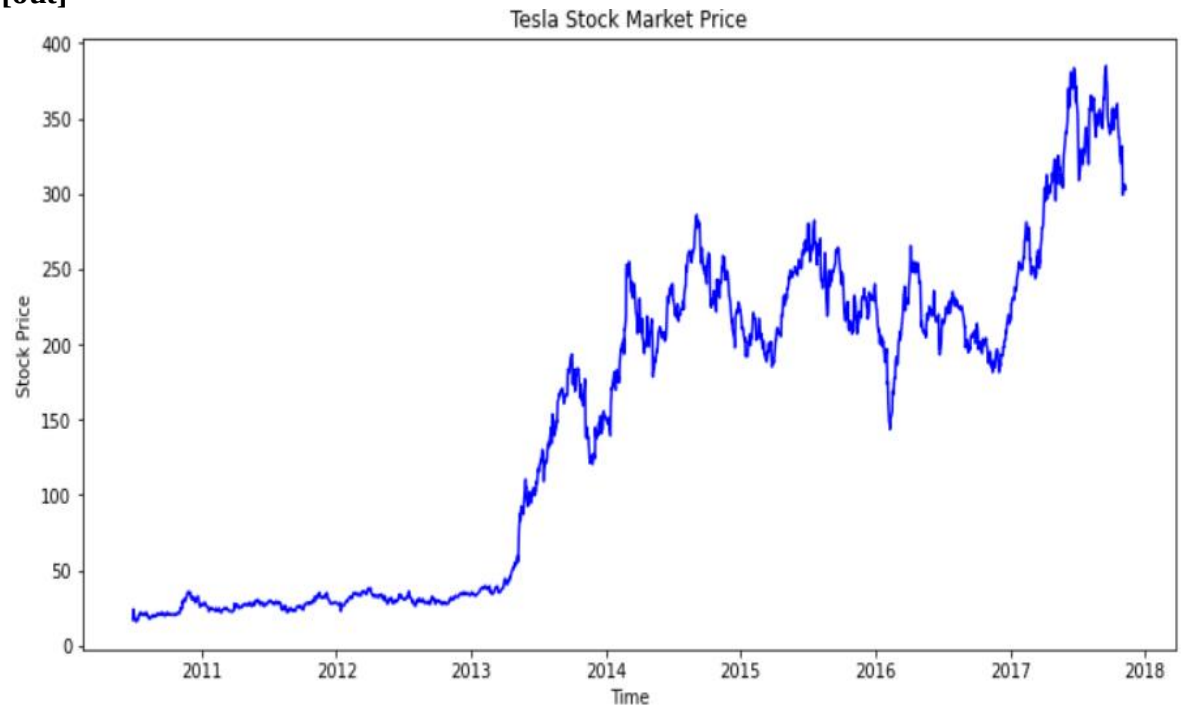

Figure 9 Tesla Stock Market Price

```
Stock_df=Stock_df.values
Stock_df[0:5]
```

[out]

```
array([[17.  ],
       [23.89],
       [23.83],
       [21.96],
       [19.2 ]])
```

```
Stock_df=Stock_df.astype("float32")
```

Training and testing the model

```
pos=int(round(len(Stock_df)*(1-0.20)))
train=Stock_df[:pos]
test=Stock_df[pos:]
print(pos, train.shape, test.shape)
```

[out] 1486 (1486, 1) (372, 1)

```
print(train.shape,test.shape)
```

[out] (1486, 1) (372, 1)

```
scaler_train=MinMaxScaler(feature_range=(0,1))
train=scaler_train.fit_transform(train)
scaler_test=MinMaxScaler(feature_range=(0,1))
test=scaler_test.fit_transform(test)
test=scaler_test.fit_transform(test)
train[0:5]
```

[out]
```
array([[0.00444049],
       [0.02993634],
       [0.02971432],
       [0.02279454],
       [0.01258141]], dtype=float32)
```

```
def create_features(data,lookback):
 X,Y=[],[]
 for i in range(lookback,len(data)):
        X.append(data[i-lookback:i,0])
        Y.append(data[i,0])
    return np.array(X),np.array(Y)
lookback=30
X_train,y_train=create_features(train,lookback)
X_test,y_test=create_features(test,lookback)
print(X_train.shape,y_train.shape,X_test.shape,y_test.shape)
```

[out] (1456, 30) (1456,) (342, 30) (342,)

```
X_train[0:5]
```

[out]
```
array([[0.00444049, 0.02993634, 0.02971432, 0.02279454, 0.01258141,
            0.00114713, 0.        , 0.00614268, 0.00592066, 0.00462551,
            0.00865897, 0.01494967, 0.01513469, 0.01791   , 0.02260953,
            0.01665186, 0.01635583, 0.01924215, 0.02031527, 0.01905713,
            0.01757696, 0.01820603, 0.01683688, 0.01531971, 0.01894612,
            0.02275755, 0.02020426, 0.01720693, 0.01402457, 0.01406157],
           [0.02993634, 0.02971432, 0.02279454, 0.01258141, 0.00114713,
            0.        , 0.00614268, 0.00592066, 0.00462551, 0.00865897,
            0.01494967, 0.01513469, 0.01791   , 0.02260953, 0.01665186,
            0.01635583, 0.01924215, 0.02031527, 0.01905713, 0.01757696,
            0.01820603, 0.01683688, 0.01531971, 0.01894612, 0.02275755,
            0.02020426, 0.01720693, 0.01402457, 0.01406157, 0.01195234],
           [0.02971432, 0.02279454, 0.01258141, 0.00114713, 0.        ,
            0.00614268, 0.00592066, 0.00462551, 0.00865897, 0.01494967,
            0.01513469, 0.01791   , 0.02260953, 0.01665186, 0.01635583,
            0.01924215, 0.02031527, 0.01905713, 0.01757696, 0.01820603,
            0.01683688, 0.01531971, 0.01894612, 0.02275755, 0.02020426,
            0.01720693, 0.01402457, 0.01406157, 0.01195234, 0.00777087],
           [0.02279454, 0.01258141, 0.00114713, 0.        , 0.00614268,
```

```
          0.00592066, 0.00462551, 0.00865897, 0.01494967, 0.01513469,
          0.01791   , 0.02260953, 0.01665186, 0.01635583, 0.01924215,
          0.02031527, 0.01905713, 0.01757696, 0.01820603, 0.01683688,
          0.01531971, 0.01894612, 0.02275755, 0.02020426, 0.01720693,
          0.01402457, 0.01406157, 0.01195234, 0.00777087, 0.00666074],
         [0.01258141, 0.00114713, 0.        , 0.00614268, 0.00592066,
          0.00462551, 0.00865897, 0.01494967, 0.01513469, 0.01791   ,
          0.02260953, 0.01665186, 0.01635583, 0.01924215, 0.02031527,
          0.01905713, 0.01757696, 0.01820603, 0.01683688, 0.01531971,
          0.01894612, 0.02275755, 0.02020426, 0.01720693, 0.01402457,
          0.01406157, 0.01195234, 0.00777087, 0.00666074, 0.00932504]],
    dtype=float32)

y_test[0:5]

[out]
array([0.16944242, 0.1735692 , 0.21287155, 0.21223283, 0.20181775],
        dtype=float32)

X_train=np.reshape(X_train,(X_train.shape[0],1,X_train.shape[1]))
X_test=np.reshape(X_test,(X_test.shape[0],1,X_test.shape[1]))
X_test=np.reshape(X_test,(X_test.shape[0],1,X_test.shape[1]))
y_train=y_train.reshape(-1,1)
y_test=y_test.reshape(-1,1)
print(X_train.shape,y_train.shape,X_test.shape,y_test.shape)
print(X_train.shape,y_train.shape,X_test.shape,y_test.shape)

[out](1456, 1, 30) (1456, 1) (342, 1, 30) (342, 1)

def ANN(m):
   from keras.models import Sequential
   from keras.layers import Dense
   model.add(Dense(11,activation='relu',input_dim=30))
   model.add(Dense(1,activation='sigmoid'))
   model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
   model.fit(X_train,y_train,epochs=3)

from lib.utils import *
ac=[]
model=ANN(X_train)
model.compile(loss="mean_squared_error",optimizer="adam")
callbacks=[EarlyStopping(monitor="val_loss",patience=3,verbose=1,mode="min"),
        ModelCheckpoint(filepath="mymodel.h5",monitor="val_loss",mode="min",
                    save_best_only=True,save_weights_only=False,verbose=1)]
history = model.fit(x=X_train,
y=y_train,epochs=200,batch_size=30,validation_data=(X_test,y_test),
                callbacks=callbacks,shuffle=False)
ac.append(accuracy_score(model,y_test,sample_weight=0.2)*100)
```

[out]

```
Epoch 1/200
40/49 [=====================>......]  - ETA: 0s - loss: 0.0061
Epoch 1: val_loss improved from inf to 0.00979, saving model to mymodel.h5
49/49 [==============================]  - 3s 15ms/step - loss: 0.0080 - val_loss: 0.009
8
Epoch 2/200
40/49 [=====================>......]  - ETA: 0s - loss: 0.0060
Epoch 2: val_loss improved from 0.00979 to 0.00892, saving model to mymodel.h5
49/49 [==============================]  - 0s 6ms/step - loss: 0.0072 - val_loss: 0.0089
Epoch 3/200
45/49 [==========================>...]  - ETA: 0s - loss: 0.0057
Epoch 3: val_loss improved from 0.00892 to 0.00798, saving model to mymodel.h5
49/49 [==============================]  - 0s 6ms/step - loss: 0.0066 - val_loss: 0.0080
Epoch 4/200
42/49 [========================>.....]  - ETA: 0s - loss: 0.0057
Epoch 4: val_loss improved from 0.00798 to 0.00743, saving model to mymodel.h5
49/49 [==============================]  - 0s 6ms/step - loss: 0.0063 - val_loss: 0.0074
Epoch 5/200
42/49 [========================>.....]  - ETA: 0s - loss: 0.0040
Epoch 5: val_loss improved from 0.00743 to 0.00705, saving model to mymodel.h5
49/49 [==============================]  - 0s 6ms/step - loss: 0.0049 - val_loss: 0.0070
Epoch 6/200
48/49 [============================>.]  - ETA: 0s - loss: 0.0047
Epoch 6: val_loss improved from 0.00705 to 0.00624, saving model to mymodel.h5
49/49 [==============================]  - 0s 5ms/step - loss: 0.0048 - val_loss: 0.0062
Epoch 7/200
44/49 [=========================>....]  - ETA: 0s - loss: 0.0035
Epoch 7: val_loss improved from 0.00624 to 0.00591, saving model to mymodel.h5
49/49 [==============================]  - 0s 6ms/step - loss: 0.0040 - val_loss: 0.0059
Epoch 8/200
34/49 [===================>..........]  - ETA: 0s - loss: 0.0025
Epoch 8: val_loss improved from 0.00591 to 0.00564, saving model to mymodel.h5
49/49 [==============================]  - 0s 5ms/step - loss: 0.0047 - val_loss: 0.0056
Epoch 9/200
47/49 [===========================>..]  - ETA: 0s - loss: 0.0039
Epoch 9: val_loss improved from 0.00564 to 0.00525, saving model to mymodel.h5
49/49 [==============================]  - 0s 5ms/step - loss: 0.0040 - val_loss: 0.0052
Epoch 10/200
47/49 [===========================>..]  - ETA: 0s - loss: 0.0039
Epoch 10: val_loss improved from 0.00525 to 0.00481, saving model to mymodel.h5
49/49 [==============================]  - 0s 6ms/step - loss: 0.0041 - val_loss: 0.0048
Epoch 11/200
45/49 [==========================>...]  - ETA: 0s - loss: 0.0032
Epoch 11: val_loss did not improve from 0.00481
49/49 [==============================]  - 0s 5ms/step - loss: 0.0035 - val_loss: 0.0048
Epoch 12/200
```

41/49 [=====================>.....] - ETA: 0s - loss: 0.0033
Epoch 12: val_loss improved from 0.00481 to 0.00449, saving model to mymodel.h5
49/49 [==============================] - 0s 7ms/step - loss: 0.0039 - val_loss: 0.0045
Epoch 13/200
33/49 [===================>.....] - ETA: 0s - loss: 0.0019
Epoch 13: val_loss improved from 0.00449 to 0.00430, saving model to mymodel.h5
49/49 [==============================] - 0s 4ms/step - loss: 0.0033 - val_loss: 0.0043
Epoch 14/200
36/49 [=====================>.....] - ETA: 0s - loss: 0.0020
Epoch 14: val_loss improved from 0.00430 to 0.00415, saving model to mymodel.h5
49/49 [==============================] - 0s 6ms/step - loss: 0.0032 - val_loss: 0.0041
Epoch 15/200
45/49 [==========================>...] - ETA: 0s - loss: 0.0031
Epoch 15: val_loss improved from 0.00415 to 0.00407, saving model to mymodel.h5
49/49 [==============================] - 0s 6ms/step - loss: 0.0033 - val_loss: 0.0041
Epoch 16/200
44/49 [=========================>....] - ETA: 0s - loss: 0.0030
Epoch 16: val_loss improved from 0.00407 to 0.00398, saving model to mymodel.h5
49/49 [==============================] - 0s 6ms/step - loss: 0.0033 - val_loss: 0.0040
Epoch 17/200
46/49 [===========================>..] - ETA: 0s - loss: 0.0034
Epoch 17: val_loss did not improve from 0.00398
49/49 [==============================] - 0s 5ms/step - loss: 0.0035 - val_loss: 0.0040
Epoch 18/200
42/49 [=======================>.....] - ETA: 0s - loss: 0.0029
Epoch 18: val_loss did not improve from 0.00398
49/49 [==============================] - 0s 5ms/step - loss: 0.0032 - val_loss: 0.0040
Epoch 19/200
46/49 [===========================>..] - ETA: 0s - loss: 0.0031
Epoch 19: val_loss improved from 0.00398 to 0.00391, saving model to mymodel.h5
49/49 [==============================] - 0s 6ms/step - loss: 0.0032 - val_loss: 0.0039
Epoch 20/200
43/49 [=========================>....] - ETA: 0s - loss: 0.0027
Epoch 20: val_loss improved from 0.00391 to 0.00385, saving model to mymodel.h5
49/49 [==============================] - 0s 6ms/step - loss: 0.0032 - val_loss: 0.0039
Epoch 21/200
44/49 [=========================>....] - ETA: 0s - loss: 0.0024
Epoch 21: val_loss did not improve from 0.00385
49/49 [==============================] - 0s 5ms/step - loss: 0.0027 - val_loss: 0.0041
Epoch 22/200
45/49 [==========================>...] - ETA: 0s - loss: 0.0029
Epoch 22: val_loss improved from 0.00385 to 0.00368, saving model to mymodel.h5
49/49 [==============================] - 0s 6ms/step - loss: 0.0031 - val_loss: 0.0037
Epoch 23/200
46/49 [===========================>..] - ETA: 0s - loss: 0.0029
Epoch 23: val_loss did not improve from 0.00368
49/49 [==============================] - 0s 5ms/step - loss: 0.0032 - val_loss: 0.0037
Epoch 24/200

46/49 [==========================>..] - ETA: 0s - loss: 0.0029
Epoch 24: val_loss did not improve from 0.00368
49/49 [==============================] - 0s 5ms/step - loss: 0.0031 - val_loss: 0.0038
Epoch 25/200
44/49 [=========================>....] - ETA: 0s - loss: 0.0031
Epoch 25: val_loss did not improve from 0.00368
49/49 [==============================] - 0s 5ms/step - loss: 0.0034 - val_loss: 0.0041
Epoch 25: early stopping

```
loss=model.evaluate(X_test,y_test,batch_size=30)
```

[out]12/12 [==============================] - 0s 2ms/step - loss: 0.0041

```
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
```

[out] 46/46 [==============================] - 0s 2ms/step
11/11 [==============================] - 0s 3ms/step

```
train_predict=scaler_train.inverse_transform(train_predict)
test_predict=scaler_test.inverse_transform(test_predict)
y_train=scaler_train.inverse_transform(y_train)
y_test=scaler_test.inverse_transform(y_test)
train_rmse=np.sqrt(mean_squared_error(y_train,train_predict))
test_rmse=np.sqrt(mean_squared_error(y_test,test_predict,))
train_prediction_data=result_df[lookback:pos]
train_prediction_data["Predicted"]=train_predict
test_prediction_data=result_df[pos+lookback:]
test_prediction_data["Predicted"]=test_predict
print("Daily prediction")
test_prediction_data.head()
```

[out] Daily prediction

Date	Close	Predicted
2016-07-07	215.94	221.558334
2016-07-08	216.78	221.641373
2016-07-11	224.78	223.143478
2016-07-12	224.65	225.323242
2016-07-13	222.53	226.999268

```
plt.figure(figsize=(14,5))
plt.plot(result_df,label="Real Values",)
plt.plot(train_prediction_data["Predicted"],color="yellow",label="Train Predicted")
```

```
plt.plot(test_prediction_data["Predicted"],color="red",label="Test Predicted")
plt.title("Yearly prediction")
plt.xlabel("Time")
plt.ylabel("Stock Values")
plt.legend()
plt.show()
```

[out]

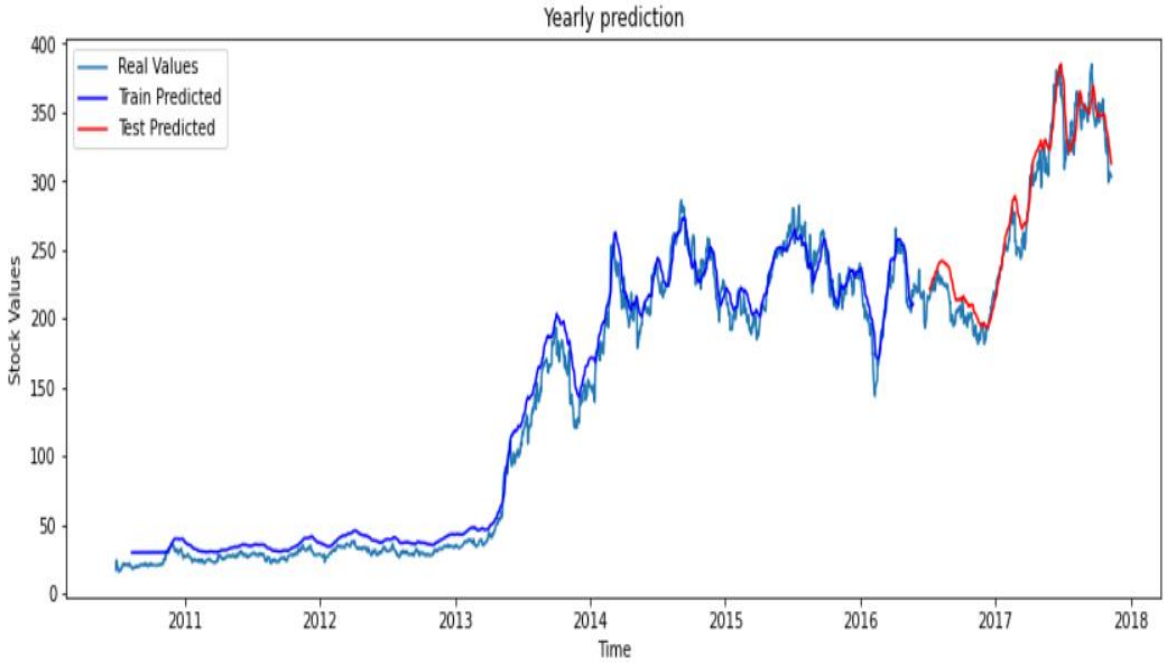

Figure 10 Yearly Prediction

```
from sklearn import tree
from sklearn import preprocessing
lab = preprocessing.LabelEncoder()
y_transformed = lab.fit_transform(y_train)
model=tree.DecisionTreeClassifier()
model.fit(X_train.reshape(1456,-1),y_transformed)
y_pred=model.predict(X_test.reshape(342,-1))
ac.append(accuracy_score(y_pred,y_test,sample_weight=0.8)*100)
import numpy as np
import seaborn as sns
import matplotlib as plt
plt.style.use('dark_background')
x=['Decision Tree','ANN']
ac.reverse()
ax=sns.barplot(x,ac)
ax.set_title('Accuracy comparison')
ax.set_ylabel('Accuracy')
#ax.yaxis.set_major_locator(ticker.LinearLocator())
print("the accuracy of {} is {} and {} is {}".format(x[0],ac[0],x[1],ac[1]))
ax.set_ylim(50,100)
```

[out]

```
the accuracy of Decision Tree is 94.19999999999999 and ANN is 96.00999999999999
(50.0, 100.0)
```

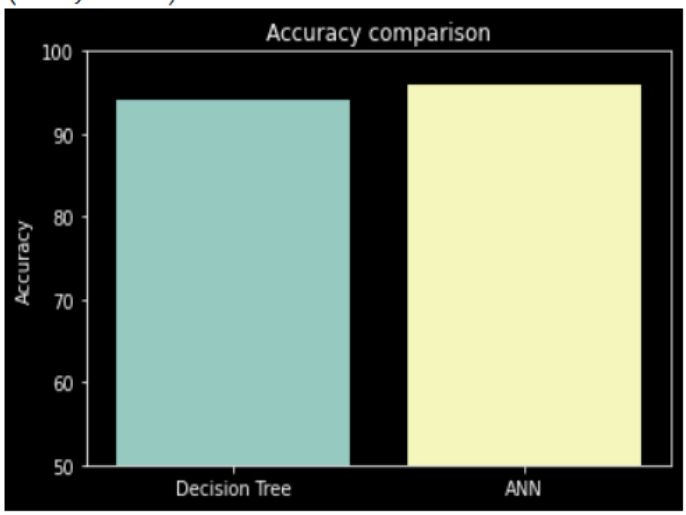

Figure 11Accuracy Comparison

CHAPTER 9
CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION:

Stock market values are not stable as it is changed day by day. Due to this reason the prediction of future stock value is a tedious process. In olden days stock market values are predicted by using various tock related news. But it does not provide the better result. If there is any error on the predicted value it leads to the huge loss. All traders are interested to buy the shares at the less amount and sales the shares in higher rate. For that reason, proper prediction methods are needed. In current situation various machine learning concepts are used to predict the stock market value. In this ANN machine learning concept is used to predict the stock market value. For this prediction historical values are used. This proposed ANN concept generates better result compared with other machine leaning concepts. The future work of this project is to add more attributes in this share market value prediction.

9.2 FUTURE SCOPE:

Further research is required to analyze the comprehensive legal landscape that aim to predict stocks of all companies from all sectors. In the project, we proposed the use of the data collected from different global financial markets with machine learning algorithms in order to predict the stock index movements. In future we can do to deep learning method of classification of stock market. Numerical results suggest the high efficiency. The practical trading models built upon our well- trained predictor. The model generates higher profit compared to the selected benchmarks.

REFERENCES

1. T. Leangarun, P. Tangamchit and S. Thajchayapong, "Stock Price Manipulation Detection Using Deep Unsupervised Learning: The Case of Thailand," in IEEE Access, vol. 9, pp. 106824-106838, 2021, doi: 10.1109/ACCESS.2021.3100359.
2. Sahnoune, E. Zeraoulia and D. Berkani, "Analysis and prediction of chaotic time series," 2022 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS), Oum El Bouaghi, Algeria, 2022, pp. 1-4, doi: 10.1109/PAIS56586.2022.9946871.
3. The Data Mining and Knowledge Discovery Handbook, January 2005, "Decision Trees," Lior Rokach and Oded Maimon (pp.165-192)
4. J. Kumari, V. Sharma and S. Chauhan, "Prediction of Stock Price using Machine Learning Techniques: A Survey," *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N),* Greater Noida, India, 2021, pp. 281-284, doi: 10.1109/ICAC3N53548.2021.9725685.
5. Dimitrios V. Vougas (2002) Application of the Dickey-Fuller test to the Nelson and Plosser (1982) Data, Applied Economics Letters, 9:8, 511-514, DOI: 10.1080/13504850110105736
6. Xiaoguo Wang and Yuejing Liu, "ARIMA time series application to employment forecasting," 2009 4th International Conference on Computer Science & Education, Nanning, China, 2009, pp. 1124-1127, doi: 10.1109/ICCSE.2009.5228480.
7. "A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance Ga YoungLee, LubnaAlzamil, BakhtiyarDoskenov, ArashTermehchyhttps://doi.org/10.48550/arXiv.2109.07127
8. Kerim Koc, Asli Pelin Gurgun,Scenario-based automated data preprocessing topredict severity of construction accidents,Automation in Construction,Volume 140,2022,104351,ISSN 0926-5805,https://doi.org/10.1016/j.autcon.2022.104351.
9. Xuefei Wang, Chi Cheng, Jiale Li, Jianmin Zhang, Guowei Ma, Jinzhao Jin,Automated monitoring and evaluation of highway subgrade compaction quality using artificial neural networks,Automation in Construction,Volume 145,2023,104663,ISSN 0926-5805,https://doi.org/10.1016/j.autcon.2022.104663.
10. Chuan-jin LI, Ya-nan DU, Zhi-wei YOU, Su-qun LIN. Applicability analysis of stationarity test methods in microtremor[J]. Progress in Geophysics, 2018, 33(2): 823-829. doi: 10.6038/pg2018BB0424