

*A Project Report*  
*On*  
**STOCK PRICE PREDICTION USING LSTM**



Submitted in partial fulfillment of the  
Requirements for the award of the degree of

**MASTER OF COMPUTER APPLICATIONS**

**SUBMITTED BY**  
**CHEKKA VVS N MANIKUMAR**  
**(19A91F0004)**

**Under the Esteemed Guidance of**

**Mr. K.C.PRADEEP, M.Tech**

**Assistant Professor**



**DEPARTMENT OF MCA**

**ADITYA ENGINEERING COLLEGE**

**An Autonomous Institution**

Approved by AICTE, Affiliated to JNTUK & Accredited by NBA, NAAC with 'A' Grade  
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956 SURAMPALEM-  
533437, E.G.Dt, ANDHRA PRADESH.

**2019-2022**

# ADITYA ENGINEERING COLLEGE

## An Autonomous Institution

Approved by AICTE, Affiliated to JNTUK & Accredited by NBA, NAAC with 'A' Grade

Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956

SURAMPALEM-533437, E.G.dt, ANDHRA PRADESH.

## DEPARTMENT OF MCA



## CERTIFICATE

This is to certify that the project work entitled, “**STOCK PRICE PREDICTION USING LSTM**” is a bonafide work of **CHEKKA VVSNI MANI KUMAR** bearing Regd.No:**19A91F0004** submitted to the faculty of Master of Computer Applications in partial fulfillment of the requirements for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** from Aditya Engineering College, Surampalem for the academic year 2021-2022.

### Project Guide

**Mr.K.C.PRADEEP,M.Tech,**

Assistant Professor,

Department of MCA,

Aditya Engineering College,

Surampalem-533437.

### Head of the Department

**Mrs.D.Beulah,M.Tech.(Ph.D),**

Associate Professor,

Department of MCA,

Aditya Engineering College,

Surampalem-533437.

### External Examiner

## **DECLARATION**

I here by declare that the project entitled “**STOCK PRICE PREDICTION USING LSTM**” submitted to **ADITYA ENGINEERING COLLEGE, SURAMPALEM** has been carried out by me alone under the guidance of **Mr.K.C.PRADEEP**.

Place: Surampalem

(CHEKKA VVSN MANIKUMAR)

Date:

(Reg No: 19A91F0004)

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first person I would like to thank is my guide **Mr.K.C.PRADEEP, M.Tech, Assistant Professor, Department of MCA, Aditya Engineering College, Surampalem.** His wide knowledge and logical way of thinking have made a deep impression on me. His understanding, encouragement and personal guidance have provided the basis for this project. He is a source of inspiration for innovative ideas and his kind support is well known to all his students and colleagues.

I wish to thank **Mrs.D.Beulah.,M.Tech.,(Ph.D) Head of the Department, Aditya Engineering College, Surampalem,** who has extended her support for the success of this project.

I also wish to thank **Dr.M.Srinivasa Reddy, Principal, Aditya Engineering College, Surampalem,** who has extended his support for the success of this project.

I would like to thank all the **Management & Technical Supporting Staff** for their timely help throughout the project.

# **ABSTRACT**

## **ABSTRACT**

Stock is an unpredictable curve. Prediction in stock market is covered with the complexity and instability. The main aim for the persuasion of the topic is to predict the stability in the future market stocks. Many researchers have performed their research on the movement of future market evolution. Stock consists of fluctuating data which makes data as an integral source of efficiency. Impact on the same chances the efficiency of the prediction. In the recent trend of Stock Market Prediction Technologies machine learning has integrated itself in the picture for deployment and prediction of training sets and data models. Machine Learning employs different predictive models and algorithms to predict and automate things of requirement. We are applying a LSTM approach for predicting future values on a short term basis.

# CONTENTS

<b>CHAPTER NO</b>	<b>PAGENO</b>
<b>Chapter 1: INTRODUCTION</b>	<b>01</b>
1.1 Brief Information about the project	01
1.2 Motivation and Contribution of the project	02
1.3 Objective of the Project	03
1.4 Organisation of Project	03
<b>Chapter 2: LITERATURE SURVEY</b>	<b>04</b>
<b>Chapter 3: SYSTEM ANALYSIS</b>	<b>06</b>
3.1 Existing System	06
3.2 Proposed System	06
3.3 Feasibility Study	08
3.4 Functional Requirements	09
3.5 Non-Functional Requirements	10
3.6 Requirements Specification	10
3.6.1 Software Requirements	10
3.6.2 Hardware Requirements	10
<b>Chapter 4: SYSTEM DESIGN</b>	<b>11</b>
4.1 Introduction	11
4.2 System Architecture	12
4.3 Module Description	13
4.4 UML Diagrams	18
4.4.1 Use case Diagram	20
4.4.2 Class Diagram	25
4.4.3 Sequence Diagram	27
4.4.4 Collaboration Diagram	29
4.4.5 Activity Diagram	32

<b>Chapter 5: TECHNOLOGY DESCRIPTION</b>	<b>33</b>
5.1 Introduction to Python	33
5.2 Python application	36
5.3 Jupyter notebook	38
<b>Chapter 6: SAMPLE CODE</b>	<b>40</b>
<b>Chapter 7: TESTING</b>	<b>48</b>
7.1 Introduction	48
7.2 Sample Test Case Specifications	52
<b>Chapter 8: SCREENSHOTS</b>	<b>53</b>
<b>CONCLUSION</b>	<b>63</b>
<b>BIBLIOGRAPHY</b>	<b>64</b>



## **LIST OF TABLES**

<b>S.NO</b>	<b>TABLE NO</b>	<b>NAME OF THE TABLE</b>	<b>PAGE NO</b>
1	<b>4.4.1.1</b>	Use Case for Upload Data	<b>21</b>
2	<b>4.4.1.2</b>	Use Case for Data Preprocessing	<b>21</b>
3	<b>4.4.1.3</b>	Use Case for Data Visualisation	<b>21</b>
4	<b>4.4.1.4</b>	Use Case for Feature Extraction	<b>22</b>
5	<b>4.4.1.5</b>	Use Case for Feature Selection	<b>22</b>
6	<b>4.4.1.6</b>	Use Case for Splitting Data	<b>23</b>
7	<b>4.4.1.7</b>	Use Case for Perform Classification	<b>23</b>
8	<b>4.4.1.8</b>	Use Case for Accuracy	<b>23</b>

## **LIST OF FIGURES**

<b>S.NO</b>	<b>FIGURE NO</b>	<b>NAME OF THE FIGURE</b>	<b>PAGE NO</b>
<b>1</b>	<b>4.2</b>	System Architecture	<b>12</b>
<b>2</b>	<b>4.4.1</b>	Use case diagram	<b>20</b>
<b>3</b>	<b>4.4.2</b>	Class diagram	<b>25</b>
<b>4</b>	<b>4.4.3</b>	Sequence diagram	<b>27</b>
<b>5</b>	<b>4.4.4</b>	Collaboration diagram	<b>29</b>
<b>6</b>	<b>4.4.5</b>	Activity diagram	<b>32</b>

## **LIST OF SCREENS**

<b>S.NO</b>	<b>SCREEN NO</b>	<b>NAME OF THE SCREEN</b>	<b>PAGE NO</b>
<b>1</b>	<b>8.1</b>	Screenshot for importing the necessary packages, loading the dataset and displaying the head of the dataset	<b>53</b>
<b>2</b>	<b>8.2</b>	Screenshot for displaying the tail of the dataset and the shape of the dataset	<b>54</b>
<b>3</b>	<b>8.3</b>	Screenshot for describing the dataset	<b>55</b>
<b>4</b>	<b>8.4</b>	Screenshot for the summary of the dataset	<b>56</b>
<b>5</b>	<b>8.5</b>	Screenshot for Ploting the dataset into graph	<b>57</b>
<b>6</b>	<b>8.6</b>	Screenshot for LSTM Algorithm	<b>58</b>
<b>7</b>	<b>8.7</b>	Screenshot for testing and plotting the dataset	<b>59</b>
<b>8</b>	<b>8.8</b>	Screenshot for testing the closing price and predict the price	<b>60</b>
<b>9</b>	<b>8.9</b>	Screenshot for plotting the stock price into X-label and Y-label	<b>61</b>
<b>10</b>	<b>8.10</b>	Screenshot for plotting the predicted price and old data price	<b>62</b>

# **CHAPTER 1**

## **INTRODUCTION**

# 1. INTRODUCTION

## 1.1 Brief information about the project:

We all have heard the word stock one way or the other. Particularly stock is related with the associates and companies which are commercialized and are settling in the world of marketization. The other word used for stock is share which is prominently used in day to day life. People even term it as an investment plan and it's something people see as a long term investment that secures and provides abundant funds during the retirement age.

Buying a company stock is purchasing a small share of it. People invest on the same to get a long term benefit which they think is less value for now but has the potential to grow with time. It's an investment that provides the long time run and deals with long time goals with the fair objectives. The value of share you invest today has to give you a yield of best tomorrow but it's not the same.

Market is unpredictable so are the resources and the factors that are taken to drive it off or on the set. It's never been on the same level and the pattern of the same is still unpredictable till the time. Some closeness and prediction method had been derived and approximates values and the rough figures are generated hoping for the best but all of the resource can't be trusted and are still unpredictable in nature.

Knowing the market situation and researching on the same is the best way to find the reliability for which there are many agents who have taken the same as a profession and are making a fortune out of it. They predict and advise but the advisory cost and the charge is higher and the stock evaluation is never less the same.

Market is changing in an instantaneous rate even in a day there are many highs and lows in the market and having said the resources and the timing the external and internal agent. Stock is a fascinating resource to start with.

Stock in other term is defined as the fair share or the ownership representation explaining the security measures and the agreement. Stock in other term is defined as the fair share or the ownership representation explaining

the security measures and the agreement between two parties which are between two parties which are Stock in other term is defined as the fair share or the ownership representation explaining the security measures and the agreement between two parties which are individual and the company. Stock is there from the start and due to its tendency of uncertainty it has been a word of fancy. People researching on the same and implementing on the daily basis had made a fortune out of it. There are various agents available in market for making you understand and invest on the same and the charges of the same are hectic and insanely expensive.

The main resources for the company is the fund to carry out the daily work and create a profit out of it. In time of need for an higher budget estimation and to overgrow from the resources they need the finance and undergoing a finance loan for approval, passing and having one is hectic and the banks are vultures for which the interest rate is higher than the other form of investment hence limiting the margin of the product.

Stock is an other way for company to collect revenue and boost up the production for the upper yield and to gain the most out of the business plan for the bigger pictures. This is found to be an effective way to invest and grow in the commercial field and a better alternative to tackle the financial crisis during the requirement.

When an individual purchases a company stock then they're referred as a shareholder and they will get a share out of the same as they have invested in their profit or the gain. A investor can sell and buy the stock as per their needs. They can share their stock to their respective or the other individuals where as there are many stock brokers available out in the firm playing with the same

## **1.2 Motivation and Contribution of the project**

Many people are interested in the financial market. And need guidance and accurate predictions to invest wisely. Investors are always looking for the accurate future results. There are many applications that try to predict the stocks but they do not give detailed information about the prediction. Thus with a successful model for stock prediction , we try to gain insight about market behavior over time , spotting trends that would otherwise not have been noticed.

### 1.3 Objective of the project

Stock market prediction is a prediction system software that illuminate the risk that undergoes during the investment in stock market. It predicts the stock rates and its rate of exchange acknowledging the basic understanding and the statistical analysis in front of users.

Data is considered as the digital fuel that gives the possibilities of higher yearn and gives the upcoming terms. Knowledge is power and same holds correct with the stock. Stock is unpredictable and over-changing its dynamic in nature. The rise and fall of the same is uneven and can't be classified so easily. Dependencies of the same deal with flexible resources and the agents behind it.

Investment during a fiscal day determines the opening stock market for the next day. The main theme of the project is to predict the turning curves and bring the predictability method and undergo the process and algorithms to conclude to a viable resource source.

### 1.4 Organization of the Project

The **Literature Survey** consists of background of the project, possible approaches, introduction and comparison of technologies. The **System Analysis** consists of description of current system, proposed system, algorithms and requirement specifications. The **System Design** consists of modules description and unified modeling language diagrams such as: use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, and activity diagrams.

The **Technology Description** consists of technology used in this project. The **Sample code** consists of sample code for few modules. The **Testing** consists of testing techniques and test cases for modules. The **Screenshots** consists of output screens of this project. This is the main conclusion of this project

# **CHAPTER 2**

## **LITERATURE SURVEY**



## 2. LITERATURE SURVEY

The literature review plays a very vital role in the research process. It is a basis from where research thoughts are drawn and developed into concepts and finally theories. It also provides the researcher a bird's eye view about the research done in that area so far. A survey gives an oversight of a field and is thus distinguishing from a sort of study which consists of a microscopic examination of a turf; it is a map rather than a detailed plan. Depending on what is observed in the literature review, a researcher will understand where his/her research stands.

### **1. Research on Stock Price Prediction Method Based on Convolutional Neural Network:**

This paper intends for a prediction model for stock price which is centered at the convolutional neural networks, that has exceptional capability of learning on its own. The data set is taught and tested relating the behaviours of both Convolutional Neural Networks and Thai stock market. The result shows that the model on grounds of Convolutional Neural Networks can effectually recognize the altering trend in stock market price and envisage it which provides significant allusion for stock price forecast. The accuracy of the prediction is found to be elevated, and it could also be promoted in the field of finance.

### **2. Enhancing Profit by Predicting Stock Prices using Deep Neural Networks:**

The prediction of economic time series is quite a herculean task, which has fascinated the attentiveness of many scholars and is extremely vital for investors. This paper focuses on presenting a deep learning system, which makes use of a range of facts for a part of the stocks on the NASDAQ exchange to predict the value of the stock. This model has been trained on the smallest of data for a particular stock and accurately estimates the concluding value of that stock for multi-step-ahead. It consists of an auto encoder in order to remove noise and makes use of time series data engineering to syndicate the advanced features with the original features. These new features are given to a Stacked LSTM Autoencoder for multistep-ahead estimation of the stock concluding value. Further, this estimation is used by a profit maximization approach to offer assistance on the right time for buying and selling a particular stock. The results indicate that the suggested framework outclasses the state of the art time series forecasting methodologies with respect to analytical accuracy and effectiveness.

### **3. An LSTM-Method for Bit-coin Price Prediction: A Case Study Yahoo Finance Stock Market:**

Bit-coin is a type of Cryptocurrency and currently is one of a kind of investment on the stock market. Stock markets are inclined by several risks. And bit-coin is one kind of crypto currency that keeps rising in recent years, and sometimes suddenly falls without knowing influence on the stock market. There's a need for automation tools to predict bit-coin on the stock market because of its fluctuations. This research study studies how to create model prediction bit-coin stock market prediction using LSTM. Before confirming the results the paper tries to measure the results using RMSE (the Root Mean Square Error). The RMSE will at all times be larger or equal to the MAE. The RMSE metric assesses how well a model can calculate a continuous value. The method that is applied on this research to predict Bit-coin on the stock market Yahoo finance can forecast the result above \$12600 USD for the next couple of days after prediction.

### **4. Share Price Prediction using Machine Learning Technique:**

Lately stock market has been the talk of the town with more and more people from academics and business showing interest in it. This paper mostly deals with the approach towards predicting stock prices using RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory) on National Stock Exchange using numerous elements such as the present-day market price as well as anonymous events. A recommendation system along with models constructed on RNN and LSTM methods are used in selecting the company is also mentioned in this paper.

### **5. Stock Market Prediction Using Machine Learning Techniques:**

The Stock Market Prices play a crucial role in today's economy. Researchers have discovered that social media platforms such as twitter and web news tend to influence the decisionmaking process of any individual. In this research behavioural reflex towards web news is taken into count to reduce the gap and make the prediction much more accurate. Precise predictions were made for a day, a week and two weeks here after.

# **CHAPTER 3**

## **SYSTEM ANALYSIS**

### 3. SYSTEM ANALYSIS

#### 3.1 Existing System

As many have invested their time and effort in this world trade for getting it closer and more reliable to the people for carrying out the resources and make their lifestyle more deliberate than the previous. The existing models can't made up with the rarest information that they can't process causing it a major data loss which creates a problem in forecasting. Observation is the integral part in the resource and prediction management. Loss of sights is a major problem in the existing system as the stock varies each days and the loss margin can be higher with respect to time. An initial instance is taken for prediction.

##### **Disadvantages:**

- The existing system fails when there are rare outcomes or predictors, as the algorithm is based on bootstrap sampling.
- The previous results indicate that the stock price is unpredictable when the traditional classifier is used.
- The existence system reported highly predictive values, by selecting an appropriate time period for their experiment to obtain highly predictive scores.

#### 3.2 Proposed System

In this proposed system, we focus on predicting the stock values using machine learning algorithms like Long-Short Term Memory (LSTM). In this proposed system, we were able to train the machine from the various data points from the past to make a future prediction. We took data from the previous year stocks to train the model. We majorly used two machine-learning libraries to solve the problem. The first one was numpy , which was used to clean and manipulate the data, and getting it into a form ready for analysis. The other was scikit, which was used for real analysis and prediction. The data set we used was from the previous years stock markets collected from the public database available online, 80 % of data was used to train the machine and the rest 20 % to test the data.

**Advantages:**

- We used the python pandas library for data processing which combined different datasets into a data frame.
- The data frame features were date and the closing price for a particular day.
- We used all these features to train the machine on LSTM and predicted the object variable, which is the price for a given day.
- We also quantified the accuracy by using the predictions for the test set and the actual values.

**Algorithm: Long Short Term Memory Network (LSTM)**

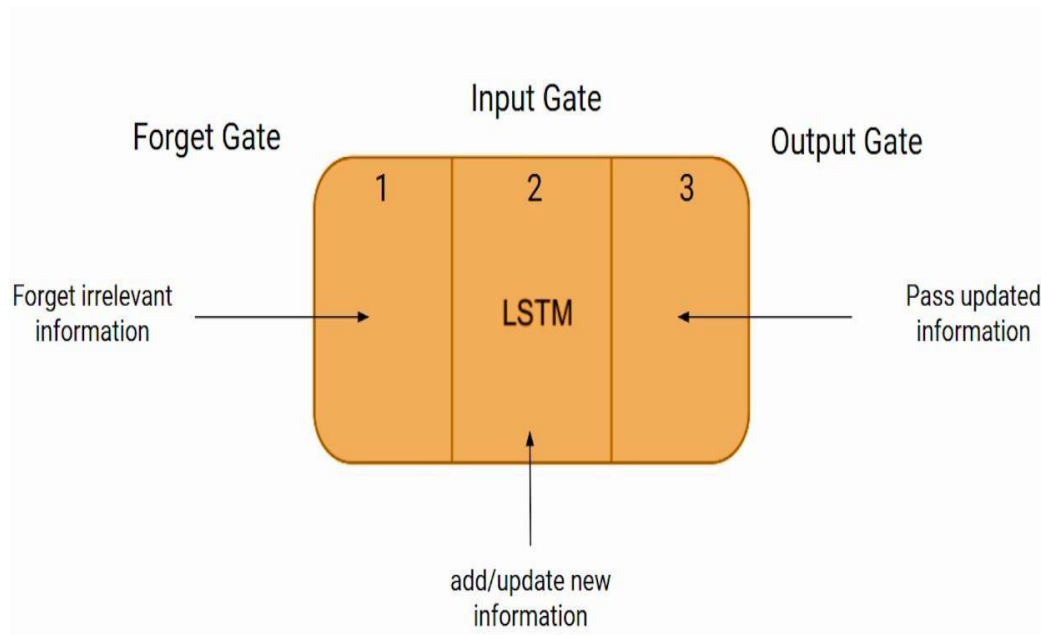
LSTM is a special kind of Recurrent Neural Network. These recurrent unit tries to remember all the past knowledge that the network is seen so far and to forget irrelevant data. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data. Each LSTM recurrent unit also maintains a vector called the Internal Cell State which conceptually describes the information that was chosen to be retained by the previous LSTM recurrent unit. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.

- The LSTM algorithm is well adapted to categorize, analyze and predict time series of uncertain duration.
- The LSTM can delete or add information to the cell state, which is carefully controlled by structures called gates.
- A Long Short Term Memory Network consists of different gates for different purposes as described below:

**Forget Gate:** It determines the extent of information to be written onto the internal cell state. The cell tries to learn new information from the input to this cell.

**Input Gate:** It determines to what extent to forget the previous data that means, whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten.

**Output Gate:** It determines what output (next hidden state) to generate from the current Internal cell state. This cell passes the updated information from the current timestamp to the next timestamp.



### Process in Long Short-Term Memory Include:

- In the first step, LSTM is to decide how much of the past it should remember which information to be omitted in from the cell in that particular time step. It looksat the previous state and the current input and computes the function.
- In the second step, It decides how much should this unit add to the current state. It takes the sufficient data and ignores the irrelevant data.
- In the Third step, It decides what part of the current cell state makes it to the output. The task of extracting useful information from the current cell state to be presented as output.

### 3.3 Feasibility Study

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

For the feasibility analysis, some understanding of the major requirements of the system is essential. The key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

## **Economic Feasibility**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

## **Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands 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 minimal or null changes are required for implementing this system.

## **Operational Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

## **3.4 Functional Requirements**

The functional requirements describe the core functionality of the software. It specifies the parameters of the project.

- The model shall accept the open price and close price of the day and also accept the high price and low price of the day as a input.
- The model should do pre-processing (like verifying for missing data values) on input for model training.
- The model shall use LSTM algorithm as main component of the software.
- It processes the given input data by producing the most possible outcomes of a open and close prices of the stock.

### 3.5 Non-Functional Requirements

The non-functional requirements are as follows:

#### Performance Requirements:

The system responds to the user input within a few seconds.

#### Security Requirements:

Security needs to be implemented like authentication to users. Our system needs to be secured in such a way that; it cannot be accessed by any unauthorized user.

#### Software Quality Attributes:

- **Reliability:** The system should never crash or hang, other than the operating system error. The system should provide strong protection in the face of prediction.
- **Maintainability:** The code in maintainability section is fully documented. Each function should be commented with pre and post conditions of the code. The module needs to permit future modifications.
- **Portability:** The software designed will run on any platform.

### 3.6 Requirements Specification

#### 3.6.1 Hardware Requirements

- Processo : 2.0 GHz or above.
- RAM : 4 GB (min. 512MB or more).
- Hard Disk : 500 GB (min. 20GB or more).

#### 3.6.2 Software Requirements

- Language : Python 3.9
- IDE : Jupyter Notebook 6.3.0
- Operating System : Windows 10.



# **CHAPTER 4**

## **SYSTEM DESIGN**

## 4. SYSTEM DESIGN

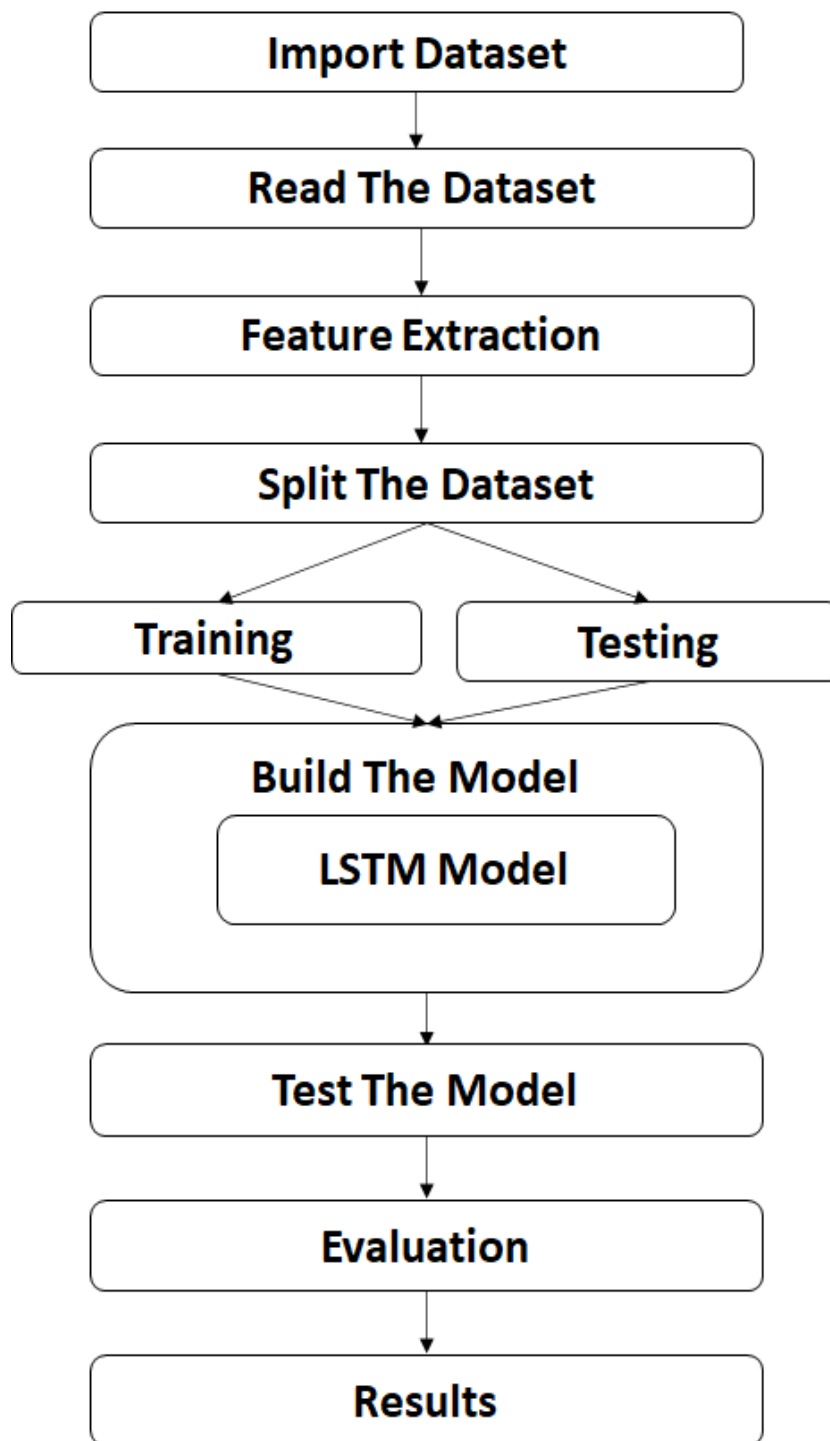
### 4.1 Introduction

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. During Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented. In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules. In other words, in system design the attention is on what components are needed, while in detailed design how the components can be implemented in software is the issue.

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blueprint for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified. During the system design activities, Developers bridge the gap between requirements specification, produced during requirements elicitation and analysis, and the system that is delivered to the user. Design is the place where the quality is fostered in development. Software design is a process through which requirements are translated into a representation of software.

## 4.2 System Architecture



**Fig 4.2 System Architecture**

## Architecture Description:

The system architecture has following steps:

- Import Dataset.
- Read the Dataset.
- Exploratory Data Analysis i.e. finding null values, duplicate values etc.
- Selecting Features (X) and Target (y) columns.
- Train Test Split will split the whole dataset into train and test data.
- Build the model i.e. Training the model.
- Test the model i.e. Model prediction.
- Evaluation of the system i.e. Accuracy score.

## 4.3 Modules Description

### 4.3.1 Data Collection:

Data used in my project is a set of stock prices on a company collected from a given dataset records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called labelled data.

### 4.3.2 Data Pre-processing

Organize your selected data by formatting, cleaning and sampling from it. Three common data pre-processing steps are:

- **Formatting:** The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
- **Cleaning:** Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be removed from the data entirely.
- **Sampling:** There may be far more selected data available than you need to

workwith. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploringand prototyping solutions before considering the whole dataset.

#### 4.3.3 Feature Extraction:

- Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, arelinear combinations of the original attributes.
- Finally, our models are trained using classifier algorithm. The module is classified on Natural Language Toolkit library in Python. The labelled data gathered and rest of the labelled data will be used to evaluate the models.

#### 4.3.4 Model Evaluation:

- Model Evaluation is an integral part of the model development process. It helpsto find the best model that represents our data and how well the chosen model will workin the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fittedmodels.
- To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form.

#### 4.3.5 Accuracy Score

Accuracy is used to evaluate the number of True positives. Higher the number of True Positives, accurate the model is. Sometimes Accuracy metrics can be misleading when the dataset is highly imbalanced the true positive will be high for thehigher instance class. So, classification metrics should be selected wisely.

Accuracy can be calculated as follows:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negative (TN)}}{\text{True Positives (TP)} + \text{False Positives (FP)} + \text{True Negative (TN)} + \text{False Negative (FN)}}$$

- **Precision Score (Specificity)**

For classification, when accuracy is seemed to be misleading, there are metrics like precision, recall, F-1 that can give a proper evaluation of the model. Precision can be calculated if TP, TN, FN, FP are available. Precision is the number of correct observations (TP) to the predicted positive observations (TP+FP).

The formula for precision is as follows:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

- **Recall Score (Sensitivity)**

A Recall is another metric for evaluating the prediction. A Recall is the number of True positive to the total number of the positive of the actual class. It means it is the ratio of true positive to a true positive and false negative.

The formula for the recall is as follows:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negative (FN)}}$$

- **F1 Score**

The F-1 score can be calculated if precision and recall value is available. F-1 is the harmonic mean of precision and recall.

The F-1 score can be calculated as follows:

$$\text{F-1 score} = 2 * \frac{\text{precision} * \text{recall}}{\text{Precision} + \text{recall}}$$

- **Confusion Matrix**

The Confusion matrix is considered to be the easiest method to evaluate the performance, as with the help of a confusion matrix you can also visualize the performance, that how many data instances are classified correctly.

TP = True Positives      FP = False Positive      FN = False Negative      TN = True Negative

Table 1 Confusion Matrix

Predicted \ Actual	Positives (0)	Negatives (1)
Positives (0)	TP	FP
Negatives (1)	FN	TN

The confusion matrix consists of the following:

- **True Positive**

The true positive rate represents the portion of the fraudulent transactions correctly being classified as fraudulent transactions, which means an actual class of the data matches the predicted class of the data.

The true positive value can be calculated as: True positive =  $TP / (TP + FN)$

- **False Positive**

The false positive rate indicates the portion of the non-fraudulent transactions wrongly being classified as fraudulent transactions, which means that the actual class of the data was 1 but the model predicted it to be 0.

The false positive value can be calculated as: False positive =  $FP / (FP + TN)$

- **False Negative**

The false negative rate indicates the portion of the non-fraudulent transactions wrongly being classified as normal transactions, which means that the actual value of the class was 0 but it was predicted to be 1.

The false negative value can be calculated as: False negative =  $FN / (FN + TP)$

- **True Negative**

The true negative rate represents the portion of the normal transactions correctly being classified as normal transactions which means that the actual value was 1 and the predicted value is also 1.

The true negative value is calculated as:  $\text{True negative} = \text{TN} / (\text{TN} + \text{FP})$

For the confusion matrix, higher that value of TP and TN, more accurate is the model.

### Dataset Description:

Predicting stocks price are always unfamiliar when compared to previous stock prices in dataset. This unfamiliarity is a very difficult problem in real-world when they are called concept drift problems. Concept drift can be said as a variable which changes over time and in unforeseen ways. These variables cause a high imbalance in the data. The table below shows basic features that are captured when any transaction is made.

Attribute Name	Description
Open Price	Open price by the stock during the day.
Close Price	Close price by the stock during the day.
High	The highest price by the stock during the day.
Low	The lowest price reached by the stock during the day.
Volume	Number of shares bought and sold in the market during the day.

The above data was then transformed into a format suitable for use with our prediction model by performing the following steps:

1. Transformation of time-series data into input-output components for supervised learning
2. Scaling the data to the  $[-1, +1]$  range



## 4.4 UML Diagrams




- The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules.
- A UML system is represented using five different views that describe the system from distinctly different perspectives. UML is specifically constructed through two different domains.
- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioural modeling, implementation modeling and environmental model views.
- These are divided into the following types:
  - Use case diagram
  - Class diagram
  - Sequence diagram
  - Collaboration diagram
  - Activity diagram

### 4.4.1 Use case diagram

Use Case diagrams identify the functionality provided by the system (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the Analysis phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

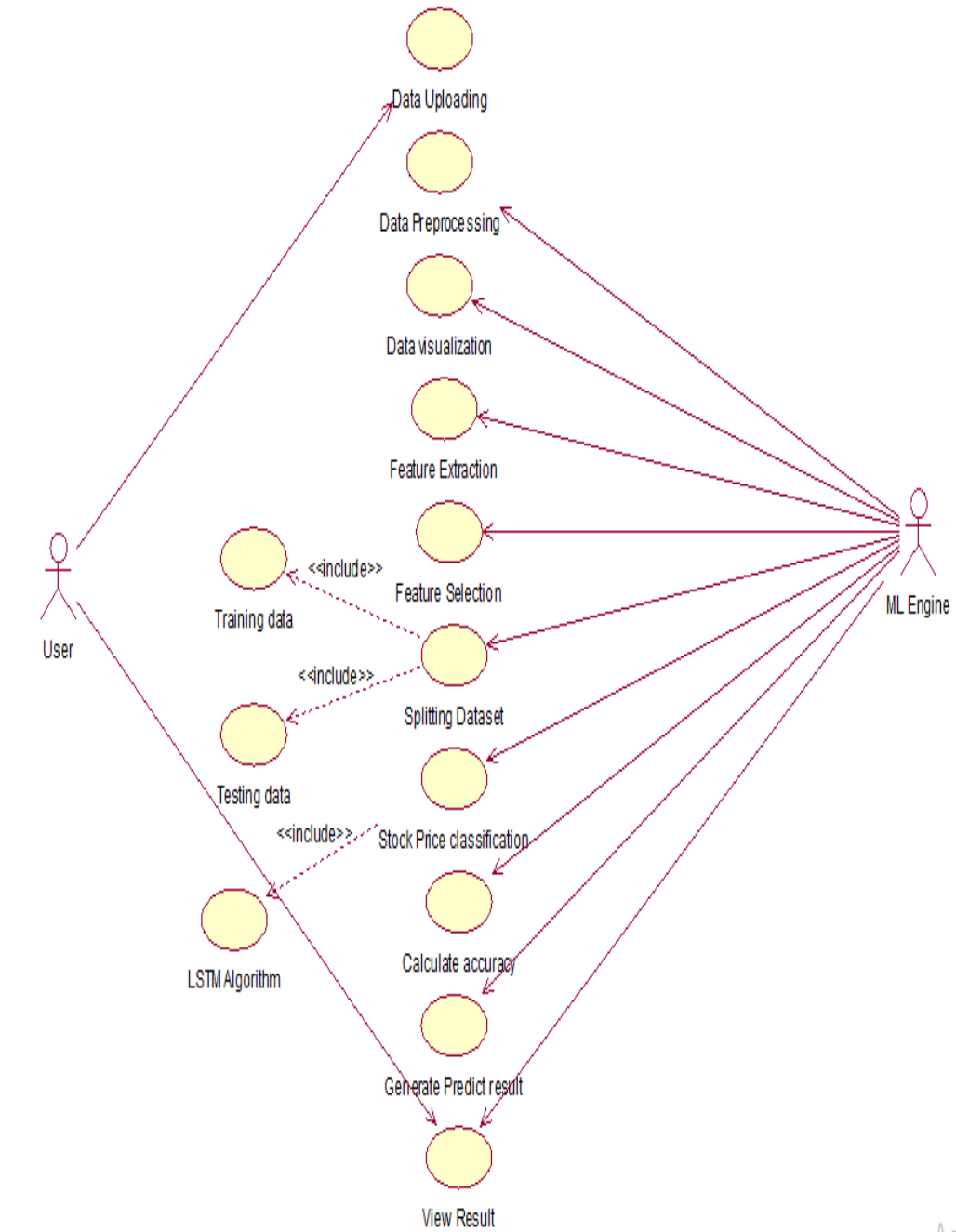
- Providing a high-level view of what the system does.
- Identifying the users ("actors") of the system.
- Determining areas needing human-computer interfaces.

**Graphical Notation:** The basic components of Use Case diagrams are the Actor, the Use Case, and the Association.

<b>Actor</b>	An Actor, as mentioned, is a user of the system, and is depicted using a stick figure. The role of the user is written beneath the icon. Actors are not limited to humans. If a system communicates with another application, and expects input or delivers output, then that application can also be considered an actor.	 <b>Actor Role Name</b>
<b>Use Case</b>	A Use Case is functionality provided by the system; Use Cases are depicted with an ellipse. The name of the use case is written within the ellipse.	 <b>Use Case Name</b>
<b>Directed Association</b>	These Associations are used to link Actors with Use Cases, and indicate that an Actor participates in the Use Case in some form.	

Behind each Use Case is a series of actions to achieve the proper functionality, as well as alternate paths for instances where validation fails, or errors occur. These actions can be further defined in a Use Case description. Because this is not addressed in UML,

there are no standards for Use Case descriptions. However, there are some common templates can follow, and whole books on the subject writing of Use Case description.



#### 4.4.1 Use case Diagram for Stock Price Prediction

**Use case Templates:**

<b>Use Case Name</b>	Data Upload
<b>Participating Actors</b>	User
<b>Flow Of Events</b>	It will load the load from the selected path where it is located in the system.
<b>Entry Condition</b>	User has to upload the data.
<b>Exit Condition</b>	Data upload successfully.

**Table 4.4.1.1: Use case for Upload Data**

<b>Use Case Name</b>	Data Pre-processing
<b>Participating Actors</b>	User
<b>Flow Of Events</b>	<ul style="list-style-type: none"> <li>It is subpart of data pre-processing some of the operations applied to handle unnecessary data like duplicates, outliers and missing values.</li> </ul>
<b>Entry Condition</b>	The file consists of features.
<b>Exit Condition</b>	The features are successfully generated.

**Table 4.4.1.2: Use case for Data Pre-processing**

<b>Use Case Name</b>	Data Visualization
<b>Participating Actors</b>	User
<b>Flow Of Events</b>	<ul style="list-style-type: none"> <li>It is subpart of data pre-processing some of the operations applied to visualize the data in the form of graphs.</li> </ul>
<b>Entry Condition</b>	The file consists of features.
<b>Exit Condition</b>	The features are successfully generated.

**Table 4.4.1.3: Use case for Data Visualization**

<b>Use Case Name</b>	Feature Extraction
<b>Participating Actors</b>	User
<b>Flow Of Events</b>	<ul style="list-style-type: none"> <li>It is a process of transforming raw data into numerical features that can be processed while preserving the information in the original data set.</li> </ul>
<b>Entry Condition</b>	Then the numerical input has been taken.
<b>Exit Condition</b>	It will Successful gives the result.

**Table 4.4.1.4: Use case for Feature Extraction**

<b>Use Case Name</b>	Feature Selection
<b>Participating Actors</b>	User
<b>Flow Of Events</b>	It is a process of reducing the number of input variables when developing a predictive model.
<b>Entry Condition</b>	Then the input variables had been reduced.
<b>Exit Condition</b>	It will Successful gives the result.

**Table 4.4.1.5: Use case for Feature Selection**

<b>Use Case Name</b>	Splitting dataset
<b>Participating Actors</b>	User
<b>Flow Of Events</b>	It is a process of splitting a dataset into training and testing in the ratio of 8:2
<b>Entry Condition</b>	Load the data Trained dataset.
<b>Exit Condition</b>	The data will train the training dataset.

**Table 4.4.1.6: Use case for Splitting dataset**

<b>Use Case Name</b>	Perform Classification
<b>Participating Actors</b>	User, ML Engine
<b>Flow Of Events</b>	It models the class with the features.
<b>Entry Condition</b>	It takes the input from the data instances.
<b>Exit Condition</b>	It generates the results from each class.

**Table 4.4.1.7: Use case for Perform Classification**



<b>Use Case Name</b>	Calculate Accuracy
<b>Participating Actors</b>	User, ML Engine
<b>Flow Of Events</b>	<ul style="list-style-type: none"> <li>• Model may give you satisfying results when evaluated.</li> <li>• Evaluation metrics such as KNN, Random Forest as name suggests gives us output and describes the complete detection of model.</li> </ul>
<b>Entry Condition</b>	Accuracy can be calculated.
<b>Exit Condition</b>	The accuracy evaluated.

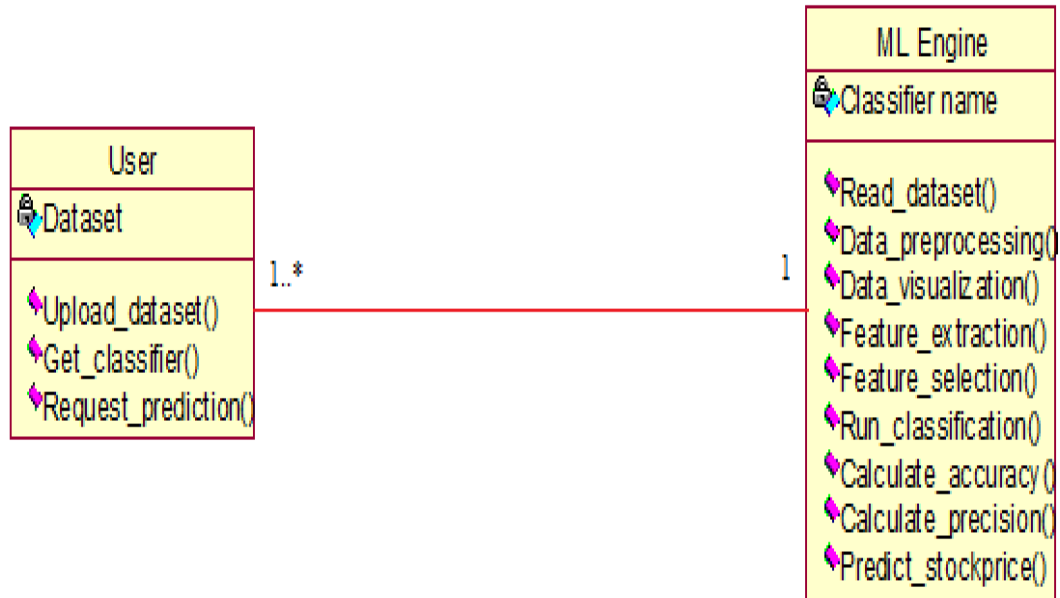
**Table 4.4.1.8: Use case for Accuracy**

#### 4.4.2 Class Diagram:

Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. Part of the popularity of Class diagrams stems from the fact that many CASE tools, such as Rational XDE, will auto-generate code in a variety of languages, these tools can synchronize models and code, reducing the workload, and can also generate Class diagrams from object-oriented code.

**Graphical Notation:** The elements on a Class diagram are classes and the relationships between them.

<p><b>Class</b></p>	<p>Classes are the building blocks in object-oriented programming. A Class is depicted using a rectangle divided into three sections. The top section is the name of the Class. The middle section defines the properties of the Class. The bottom section lists the methods of the class.</p>	 <pre> classDiagram     class SiteConfig {         +SqlDSN: string         +AdminEmail: string     }         </pre> <p>The diagram shows a class named 'SiteConfig' with two attributes: '+SqlDSN: string' and '+AdminEmail: string'.</p>
<p><b>Association</b></p>	<p>An Association is a generic relationship between two classes, and is modeled by a line connecting the two classes. This line can be qualified with the type of relationship, and can also feature multiplicity rule (e.g. one-to-one, one-to-many, many-to-many) for the relationship.</p>	 <pre> classDiagram     class A     class B     A "1..n" -- "1" B : owned by         </pre> <p>The diagram shows an association between two classes, 'A' and 'B'. The association is labeled 'owned by'. The multiplicity at class 'A' is '1..n' and at class 'B' is '1'.</p>



#### 4.4.2 Class Diagram for Stock Price Prediction

##### Description:



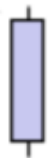

In the above diagram represents the class diagram of the project “Stock Price Prediction Using LSTM”. In the above class diagram, it describes the relation between User and ML Engine.

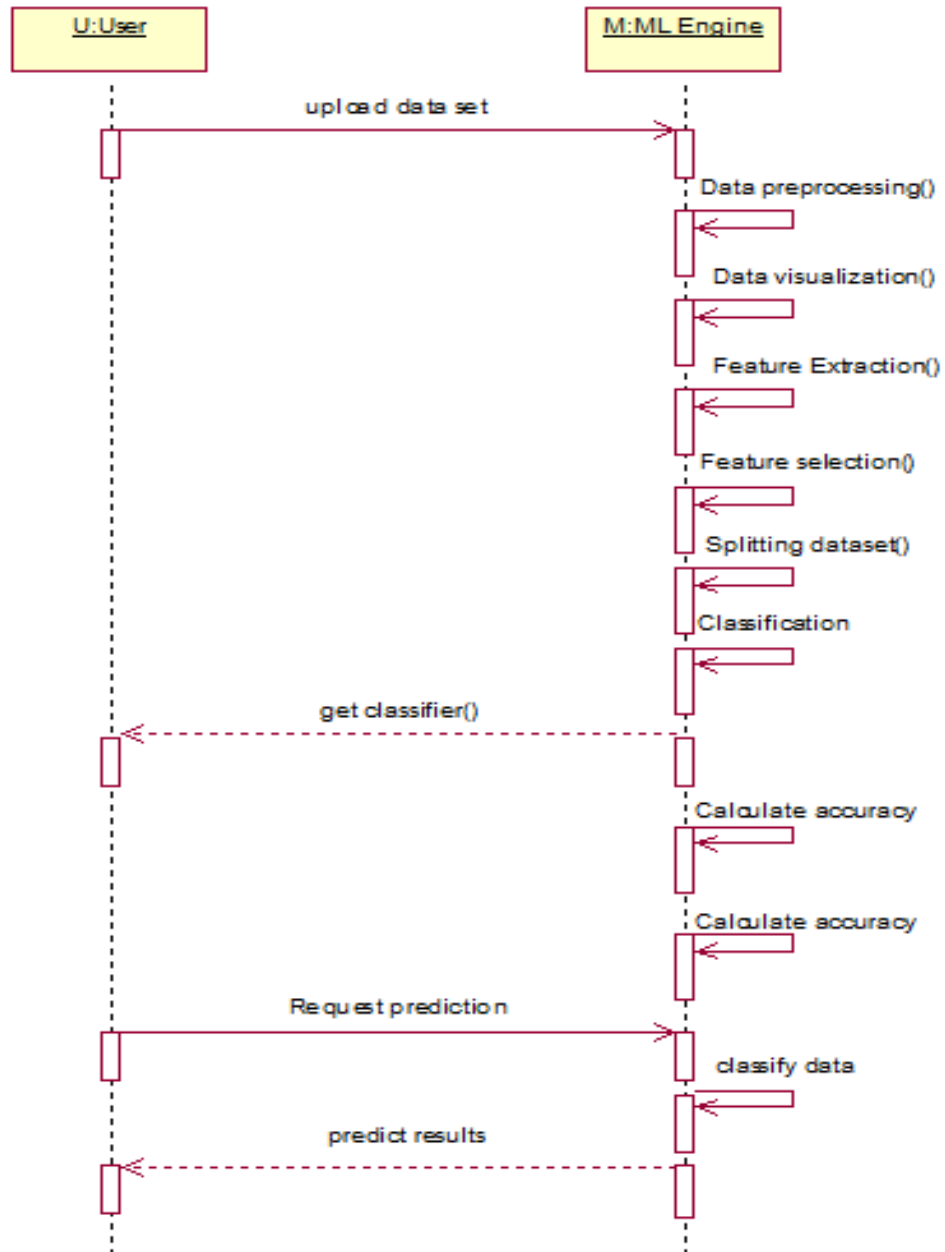
#### 4.4.3 Sequence Diagram

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

**Graphical Notation** in a Sequence diagram, classes and actors are listed as columns, with vertical lifelines indicating the lifetime of the object over time.



<b>Object</b>	Objects are instances of classes, and are arranged horizontally. The pictorial representation for an Object is a class (a rectangle) with the name prefixed by the object name (optional) and a semi-colon	
<b>Lifeline</b>	The Lifeline identifies the existence of the object over time. The notation for a Lifeline is a vertical dotted line extending from an object.	
<b>Activation</b>	Activations, modelled as rectangular boxes on the lifeline, indicate when the object is performing an action.	
<b>Message</b>	Messages, modelled as horizontal arrows between Activations, indicate the communications between objects.	



#### 4.4.3 Sequence Diagram for Stock Price Prediction




##### Description:

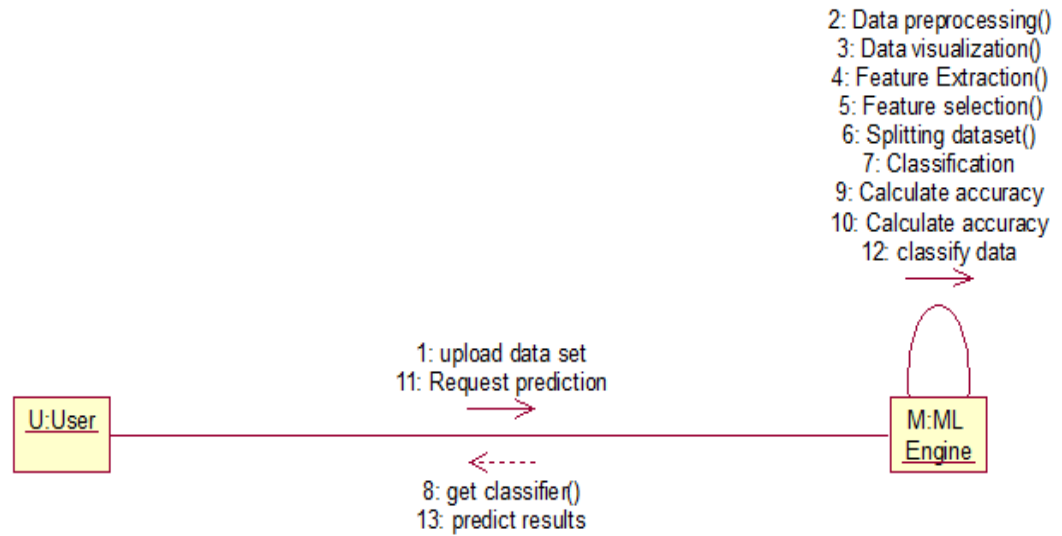
In the above diagram represents the sequence diagram of the project “Stock Price Prediction Using LSTM”. In the above Sequence diagram, it describes the relation between User and ML Engine.

#### 4.4.4 Collaboration Diagram

Like the other Behavioural diagrams, Collaboration diagrams model the interactions between objects. This type of diagram is a cross between an object diagram and a sequence diagram. Unlike the Sequence diagram, which models the interaction in a column and row type format, the Collaboration diagram uses the free-form arrangement of objects as found in an Object diagram. This makes it easier to see all interactions involving a particular object.

##### Graphical Notation:

Object	Objects are instances of classes, and are one of the entity types that can be involved in communications. An Object is drawn as a rectangular box, with the class name inside prefixed with the object name (optional) and a semi-colon.	
Actor	Actors can also communicate with Objects, so they too can be listed on Collaboration diagrams. An Actor is depicted by a stick figure.	
Message	Messages, modeled as arrows between objects, and labelled with an ordering number. Indicate the communications between objects.	



#### 4.4.4 Collaboration Diagram for Stock Price Prediction

##### Description:


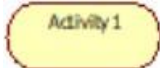


In the above diagram represents the Collaboration diagram of the project “Stock Price Prediction Using LSTM”. In the above Collaboration diagram, it describes the relation between User and ML Engine.




#### 4.4.5 Activity Diagram

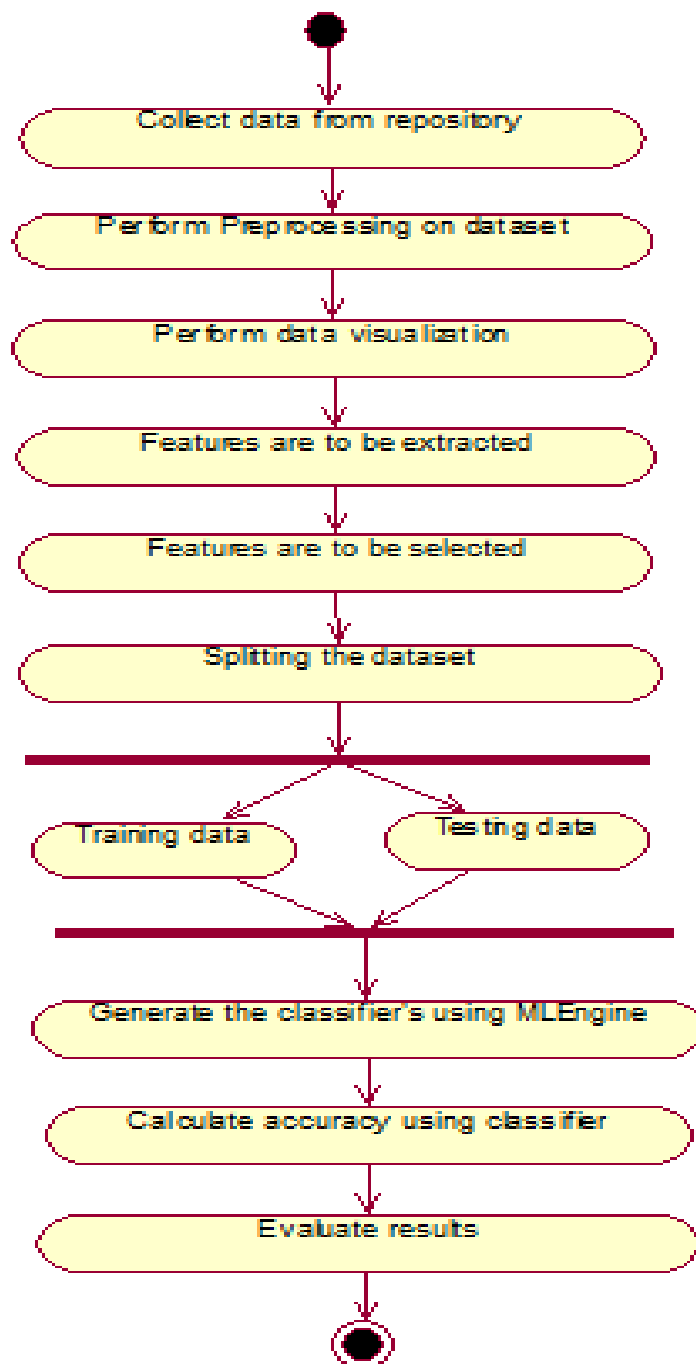
This shows the flow of events within the system. The activities that occur within a use case or within an object's behaviour typically occur in a sequence. An activity diagram is designed to be a simplified look at what happens during an operation or a process.

Each activity is represented by a rounded rectangle. The processing within an activity goes to completion and then an automatic transmission to the next activity occurs. An arrow represents the transition from one activity to the next. An activity diagram describes a system in terms of activities. Activities are the state that represents the execution of a set of operations. These are similar to flow chart diagrams and dataflow.

## Graphical Notation:

Starting point	This is starting point of process.	
Action	An action represents the execution of an atomic action, typically the invocation of an operation. An action is a simple action with an entry action, whose only exit transition is triggered by the implicit event of completing the execution of the entry action.	
End point	An end point represents the last or "final" activity of the enclosing composite activity. There may be more than one final point at any level signifying that the composite activity can end in different ways or conditions. When a final activity is reached and there is no other enclosing activity it means that the entire activity has completed its transitions and no more transitions can occur.	
Decision	An activity diagram expresses decision when guard conditions are used to indicate different possible transitions that depend on Boolean conditions of the owning object.	

<b>Synchronisation bar</b>	The fork will divide the state into multiple states and the join will merge the multiple states into single states.	
<b>Swim lane</b>	The swim lane will divide the business objects into meaning full order.	
<b>Transition</b>	A transition is a directed relationship between a source state vertex and a target state vertex. It may be part of a compound transition, which takes the static machine from one static configuration to another, representing the complete response of the static machine to a particular event instance.	



#### 4.4.5 Activity Diagram for Stock Price Prediction

##### Description:

In the above diagram represents the activity diagram of the project “Stock Price Prediction Using LSTM”. In the above activity diagram, it describes the relation between User and ML Engine.

# **CHAPTER 5**

## **TECHNOLOGY DESCRIPTION**



## 5. TECHNOLOGY DESCRIPTION

### 5.1 Introduction to Python

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming. Python is a programming language that lets you work quickly and integrate systems more efficiently.

### Features of Python

There are many features in Python, some of which are described below:

- **Easy to code:**

Python is a high-level programming language. Python language is very easy to learn as compared to other languages like C, C#, Javascript, Java, etc. Python is very easy to code and anybody can learn python basics in a few hours or days. It is a developer-friendly language.

- **Free and Open Source:**

Python language is freely available, anyone can download it, use it and share it completely free of cost. Since it is open-source, this means that source code is also available to the public.

- **Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

- **GUI Programming Support:**

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk (Tkinter) in python. PyQt5 is the most popular option for creating graphical apps with Python.

- **High-Level Language:**

Python is a high-level language. When programs are written in python, it is not necessary to remember the system architecture, or manage the memory

- **Extensible feature:**

Python is an extensible language. Python code can be written into C or C++ language and compile that code in C/C++ language.

- **Python is Portable language:**

Python language is also a portable language. For example, if there is a python code for windows and in order to run this code on other platforms such as Linux, Unix, and Mac then there is no need to change it, the code can run on any platform.

- **Python is Integrated language:**

Python is also an integrated language because it can easily integrated with other languages like c, c++, etc.

- **Interpreted Language:**

Python is an Interpreted Language because code is executed line by line at a time. Unlike other languages like C, C++, Java, etc. there is no need to compile the code as python makes it easier to debug the code. The source code of python is converted into an immediate form called bytecode.

- **Large Standard Library**

Python has a large standard library which provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

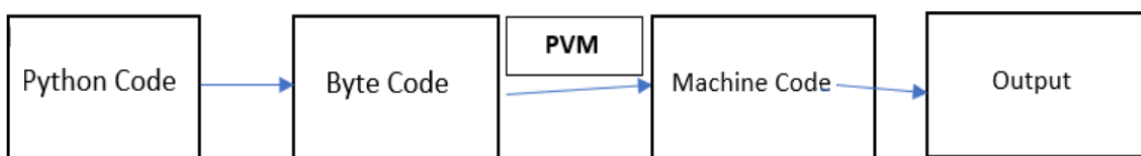
- **Dynamically Typed Language:**

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature there is no need to specify the type of variable.

## Python Virtual Machine

Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called “cpython”. It is the default and widely used implementation of Python.

Python doesn't convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code. So, within python, compilation happens, but it's just not into a machine language. It is into byte code (.pyc or .pyo) and this byte code can't be understood by the CPU. So we need an interpreter called the python virtual machine to execute the byte codes.



The Python source code goes through the following to generate an executable code :

- **Step 1:** The python compiler reads a python source code or instruction. Then it verifies that the instruction is well-formatted, i.e. it checks the syntax of each line. If it encounters an error, it immediately halts the translation and shows an error message.
- **Step 2:** If there is no error, i.e. if the python instruction or source code is well-formatted then the compiler translates it into its equivalent form in an intermediate language called “Byte code”.
- **Step 3:** Byte code is then sent to the Python Virtual Machine(PVM) which is the python interpreter. PVM converts the python byte code into machine-executable code. If an error occurs during this interpretation then the conversion is halted with an error message.

## 5.2 Applications of Python

- **Web Applications:**

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

- **Scientific and numeric computing:**

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

- **Creating software prototypes:**

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

- **GUI based desktop applications:**

Python has simple syntax, modular architecture; rich text processing tools and the ability to work on multiple operating systems create highly functional Graphical User Interface (GUI).

- **Data Analysis and Visualization:**

Data analysis is a process of collecting, inspecting, cleansing, transforming, and modeling data to discover useful information, make predictions, arrive at conclusions, support decision-making processes, and more. Data analysis is closely related to data visualization, which deals with the graphical representation of data.

- **Image Processing and Graphic Design Applications:**

Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribes'. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use Python in variable proportions.

- **Artificial Intelligence and Machine Learning:**

AI/ML applications require a language that is stable, secure, flexible, and is equipped with tools that can handle the various unique requirements of such projects. Python has all these qualities, and hence, it has become one of the most favoured languages for Data Science professionals.

Python's simplicity, consistency, platform independence, great collection of resourceful libraries, and an active community make it the perfect tool for developing AI and ML applications.

- **Scientific and Computational Applications:**

The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as FreeCAD, and finite element method software, such as Abacus, are coded in Python.

- **Games:**

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, and Vega Strike etc.

- **Operating Systems:**

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

## 5.3 Jupyter Notebook

### Introduction

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 modelling
- Data visualization
- Machine learning, and much more.

### Components

The Jupyter Notebook combines three components:

- **The notebook web application:**

An interactive web application for writing and running code interactively and authoring notebook documents.

- **Kernels:**

Separate processes started by the notebook web application that runs users' code in a given language and returns output back to the notebook web application. The kernel also handles things like computations for interactive widgets, tab completion and introspection.

- **Notebook documents:**

Self-contained documents that contain a representation of all content visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.

## Notebook web application

The notebook web application enables users to:

- Edit code in the browser, with automatic syntax highlighting, indentation, and tab completion/introspection.
- Run code from the browser, with the results of computations attached to the code which generated them.
- See the results of computations with rich media representations, such as HTML, LaTeX, PNG, SVG, PDF, etc.
- Create and use interactive JavaScript widgets, which bind interactive user interface controls and visualizations to reactive kernel side computations.
- Author narrative text using the Markdown markup language.

# **CHAPTER 6**

## **SAMPLE CODE**



## 6. SAMPLE CODE

### **Program 1: Data Exploration For Open Price(Data Pre-processing and Data Visualisation) [DataExploration.ipynb]**

```
#importing necessary packages

import pandas as pd

import datetime

import numpy as np

import matplotlib.pyplot as plt

import datetime as dt

from matplotlib import pyplot as plt

from sklearn import model_selection

from sklearn.metrics import confusion_matrix

from sklearn.preprocessing import StandardScaler

from sklearn.model_selection import train_test_split

import numpy as np

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Dropout

data = pd.read_csv('NSE-Tata_Global_Beverages_Limited.csv')

dataset_train=data.iloc[0:930,1:2]

dataset_test=data.iloc[930:,1:2]

training_set = data.iloc[0:930, 1:2].values

testing_set=data.iloc[930:,1:2].values
```

```

data.head()

data.drop('Last', axis=1, inplace=True)

data.drop('Total Trade Quantity', axis=1, inplace=True)

data.drop('Turnover (Lacs)', axis=1, inplace=True)

print(data.head())

data.to_csv('tata_preprocessed.csv',index= False)

data = data.iloc[::-1]

import math

import pandas as pd

import numpy as np

from IPython.display import display

from sklearn import linear_model

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error

from sklearn.model_selection import TimeSeriesSplit

data.tail(10)

plt.figure(figsize = (18,9))

plt.plot(range(data.shape[0]),(data['Open']))

plt.xticks(range(0,data.shape[0],500),data['Date'].loc[::500],rotation=45)

plt.xlabel('Date',fontsize=18)

plt.ylabel('Mid Price',fontsize=18)

plt.show()

from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler(feature_range = (0, 1))

training_set_scaled = sc.fit_transform(training_set)

X_train = []

y_train = []

for i in range(10,930):

```

```

X_train.append(training_set_scaled[i-10:i, 0])

y_train.append(training_set_scaled[i, 0])

X_train, y_train = np.array(X_train), np.array(y_train)

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

```

## **Program 2: Long-Short Term Memory Algorithm [LSTM.ipynb]**

```

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Dropout

regressor = Sequential()

    regressor.add(LSTM(units = 75, return_sequences = True, input_shape =
(X_train.shape[1], 1)))

regressor.add(Dropout(0.1))

    regressor.add(LSTM(units = 50, return_sequences = True))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))

regressor.add(Dropout(0.1))

    regressor.add(LSTM(units = 75))

regressor.add(Dropout(0.2))

    regressor.add(Dense(units = 1))

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

regressor.fit(X_train, y_train, epochs = 200, batch_size = 64)

real_stock_price = testing_set

dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)

inputs = dataset_total[len(dataset_total) - len(dataset_test) - 10:].values

inputs = inputs.reshape(-1,1)

inputs = sc.transform(inputs)

X_test = []

```

```

for i in range(10,305):

X_test.append(inputs[i-10:i, 0])

X_test = np.array(X_test)

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

predicted_stock_price = regressor.predict(X_test)

predicted_stock_price = sc.inverse_transform(predicted_stock_price)

plt.figure(figsize = (18,9))

plt.plot(real_stock_price, color = 'blue', label = 'TATA Stock Price')

plt.plot(predicted_stock_price, color = 'red', label = 'Predicted TATA Stock Price')

plt.title('TATA Stock Price Prediction')

plt.xlabel('Time')

plt.ylabel('TATA Stock Price')

plt.legend()

plt.show()

```

### **Program 3: Data Exploration For Close Price(Data Pre-processing and Data Visualisation) [DataExploration.ipynb]**

```

import pandas as pd

import datetime

import numpy as np

import matplotlib.pyplot as plt

import datetime as dt

from matplotlib import pyplot as plt

from sklearn import model_selection

from sklearn.metrics import confusion_matrix

from sklearn.preprocessing import StandardScaler

    from sklearn.model_selection import train_test_split

import numpy as np

import pandas as pd

```

```

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Dropout

import math

from IPython.display import display

from sklearn import linear_model

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error

from sklearn.model_selection import TimeSeriesSplit

data = pd.read_csv('NSE-Tata_Global_Beverages_Limited.csv')

dataset_train=data.iloc[0:930,1:2]

dataset_test=data.iloc[930:,1:2]

training_set = data.iloc[0:930, 3:4].values

testing_set=data.iloc[930:,3:4].values

data.head()

data.drop('Last', axis=1, inplace=True)

data.drop('Total Trade Quantity', axis=1, inplace=True)

data.drop('Turnover (Lacs)', axis=1, inplace=True)

print(data.head())

data.to_csv('tata_preprocessed.csv',index= False)

data = data.iloc[:-1]

plt.figure(figsize = (18,9))

plt.plot(range(data.shape[0]),(data['Close']))

plt.xticks(range(0,data.shape[0],500),data['Date'].loc[::500],rotation=45)

plt.xlabel('Date',fontsize=18)

plt.ylabel('Close Price',fontsize=18)

```

```

plt.show()

from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler(feature_range = (0, 1))

training_set_scaled = sc.fit_transform(training_set)

len(training_set_scaled)

X_train = []

y_train = []

for i in range(10,930):

    X_train.append(training_set_scaled[i-10:i, 0])

    y_train.append(training_set_scaled[i, 0])

X_train, y_train = np.array(X_train), np.array(y_train)

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

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import LSTM

from keras.layers import Dropout

regressor = Sequential()

    regressor.add(LSTM(units = 75, return_sequences = True, input_shape =
(X_train.shape[1], 1)))

regressor.add(Dropout(0.1))

    regressor.add(LSTM(units = 50, return_sequences = True))

regressor.add(Dropout(0.2))

    regressor.add(LSTM(units = 50, return_sequences = True))

regressor.add(Dropout(0.1))

    regressor.add(LSTM(units = 75))

regressor.add(Dropout(0.2))

    regressor.add(Dense(units = 1))

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

```

```

regressor.fit(X_train, y_train, epochs = 200, batch_size = 64)

real_stock_price = testing_set

dataset_total = pd.concat((dataset_train['Open'], dataset_test['Open']), axis = 0)

inputs = dataset_total[len(dataset_total) - len(dataset_test) - 10:].values

inputs = inputs.reshape(-1,1)

inputs = sc.transform(inputs)

X_test = []

for i in range(10,305):

    X_test.append(inputs[i-10:i, 0])

X_test = np.array(X_test)

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

predicted_stock_price = regressor.predict(X_test)

predicted_stock_price = sc.inverse_transform(predicted_stock_price)

# X_test

zz = regressor.predict([[0.29054816],

                        [0.3054365 ],

                        [0.30904579]])

prd =sc.inverse_transform(zz)

mse = np.mean((real - prd)**2)

(1- mse/100)*100

plt.figure(figsize = (18,9))

plt.plot(real, color = 'blue', label = 'TATA Stock Price')

plt.plot(prd, color = 'red', label = 'Predicted TATA Stock Price')

plt.title('TATA Stock Price Prediction')

plt.xlabel('Time')

plt.ylabel('TATA Stock Price')

plt.legend()

plt.show()

```

```
plt.figure(figsize = (18,9))  
plt.plot(real_stock_price, color = 'blue', label = 'TATA Stock Price')  
plt.plot(predicted_stock_price, color = 'red', label = 'Predicted TATA Stock Price')  
plt.title('TATA Stock Price Prediction')  
plt.xlabel('Time')  
plt.ylabel('TATA Stock Price')  
plt.legend()  
plt.show()
```



# **CHAPTER 7**

## **TESTING**

## 7. TESTING

### 7.1 Introduction

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

### System Testing and Implementation

The purpose is to exercise the different parts of the module code to detect coding errors. After this the modules are gradually integrated into subsystems, which are then integrated themselves too eventually forming the entire system. During integration of module integration testing is performed. The goal of this is to detect designing errors, while focusing the interconnection between modules. After the system was put together, system testing is performed. Here the system is tested against the system requirements to see if all requirements were met and the system performs as specified by the requirements. Finally acceptance testing is performed to demonstrate to the client for the operation of the system.

For the testing to be successful, proper selection of the test case is essential. There are two different approaches for selecting test case. The software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called —black box testing.

The focus here is on testing the external behaviour of the system. In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. The two forms of testing are complementary: one tests the external behaviour, the other tests the internal structure.

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Frequently the testing process starts with the testplan. This plan identifies all testing related activities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing.

The test plan specifies conditions that should be tested; different units to be tested, and the manner in which the module will be integrated together. Then for different test unit, a test case specification document is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual results are compared with the expected outputs. The final output of the testing phase is the testing report and the error report, or a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the errors encountered and the action taken to remove the error.

## Testing Techniques

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected. In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

### ➤ Black Box Testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been used to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structure or external database access.
- Performance errors.
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

- **White Box Testing**

In this testing, the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been used to generate the test cases in the following cases:

1. Guarantee that all independent paths have been executed.
2. Execute all logical decisions on their true and false sides
3. Execute all loops at their boundaries and within their operational.
4. Execute internal data structures to ensure their validity.

### **Testing Strategies:**

- **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.

This System consists of 3 modules. Those are Reputation module, route discovery module, audit module. Each module is taken as unit and tested. Identified errors are corrected and executable unit are obtained.

- **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 outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, 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.

- **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 and flows, emphasizing pre-driven process links and integration points.

- **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.
- Function : 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.

## 7.2 Sample Test Case Specification

Test Number	Test Case	Output	Result
T1	Open project in Jupyter Notebook	Project loads successfully	Pass
		Project loads unsuccessfully	Fail
T2	Run the main file	Program executes	Pass
		Program doesn't executes	Fail
T3	Data loading function is executed	Data is loaded	Pass
		Data is not loaded	Fail
T4	Data Preprocessing is performed	Data is preprocessed	Pass
		Data is not preprocessed	Fail
T5	Feature extraction is performed	Features are extracted from the data	Pass
		Features are not extracted from the data	Fail
T6	Feature selection is performed	Price report is generated and the features are selected.	Pass
		Price report is not generated and the features are not selected.	Fail
T7	Train/Test the data	The process of training/testing has been activated based on the data set.	Pass
		The process of training/testing has not been activated based on the data set.	Fail
T8	Classification model is called	Displaying results using classification model	Pass
		No results are displaying using classification model	Fail
T9	Accuracy is derived from the function	Accuracy is measured and displayed	Pass
		Accuracy is not measured and not displayed	Fail

# **CHAPTER 8**

## **SCREENSHOTS**

## 8. SCREENSHOTS

```

In [1]: import pandas as pd
import datetime
import numpy as np
import matplotlib.pyplot as plt
import datetime as dt
from matplotlib import pyplot as plt
from sklearn import model_selection
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout

In [2]: data = pd.read_csv('NSE-Tata_Global_Beverages_Limited.csv')
dataset_train=data.iloc[0:930,1:2]
dataset_test=data.iloc[930:,1:2]
training_set = data.iloc[0:930, 1:2].values
testing_set=data.iloc[930:,1:2].values
data.head()

Out[2]:

```

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	08-10-2018	208.00	222.25	206.85	216.00	215.15	4642146	10062.83
1	05-10-2018	217.00	218.60	205.90	210.25	209.20	3519515	7407.06
2	04-10-2018	223.50	227.80	216.15	217.25	218.20	1728786	3815.79
3	03-10-2018	230.00	237.50	225.75	226.45	227.60	1708590	3960.27
4	01-10-2018	234.55	234.60	221.05	230.30	230.90	1534749	3486.05

**Fig 8.1 Screenshot for importing the necessary packages, loading the dataset and displaying the head of the dataset.**

### Description:

The above screenshot displays the importing of necessary packages like numpy, pandas, matplotlib, seaborn etc, loading the existing dataset and displaying the head of the dataset, that is, it displays the first 5 rows in the dataset.



```
In [4]: import math
import pandas as pd
import numpy as np
from IPython.display import display
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import TimeSeriesSplit
```

```
In [5]: data.tail()
```

```
Out[5]:
```

	Date	Open	High	Low	Close
4	01-10-2018	234.55	234.60	221.05	230.90
3	03-10-2018	230.00	237.50	225.75	227.60
2	04-10-2018	223.50	227.80	216.15	218.20
1	05-10-2018	217.00	218.60	205.90	209.20
0	08-10-2018	208.00	222.25	206.85	215.15

```
In [6]: data.shape
```

```
Out[6]: (1235, 5)
```

**Fig 8.2 Screenshot for displaying the tail of the dataset and the shape of the dataset**

### Description:

The above screenshot displays the tail of the dataset, that is, it used to display the last 5 rows in the dataset. The shape of the dataset depicts the returns the rows and columns of the dataset.

In [7]: `data.describe()`

Out[7]:

	Open	High	Low	Close
<b>count</b>	1235.000000	1235.000000	1235.000000	1235.000000
<b>mean</b>	168.954858	171.429069	166.402308	168.731053
<b>std</b>	51.499145	52.436761	50.542919	51.544928
<b>min</b>	103.000000	104.600000	100.000000	102.650000
<b>25%</b>	137.550000	138.925000	135.250000	137.225000
<b>50%</b>	151.500000	153.250000	149.500000	151.100000
<b>75%</b>	169.000000	172.325000	166.700000	169.500000
<b>max</b>	327.700000	328.750000	321.650000	325.750000

**Fig 8.3 Screenshot for describing the dataset**

### Description:

The above screenshot displays the description of the dataset, which provides the overview of the dataset.

In [8]: `data.info()`

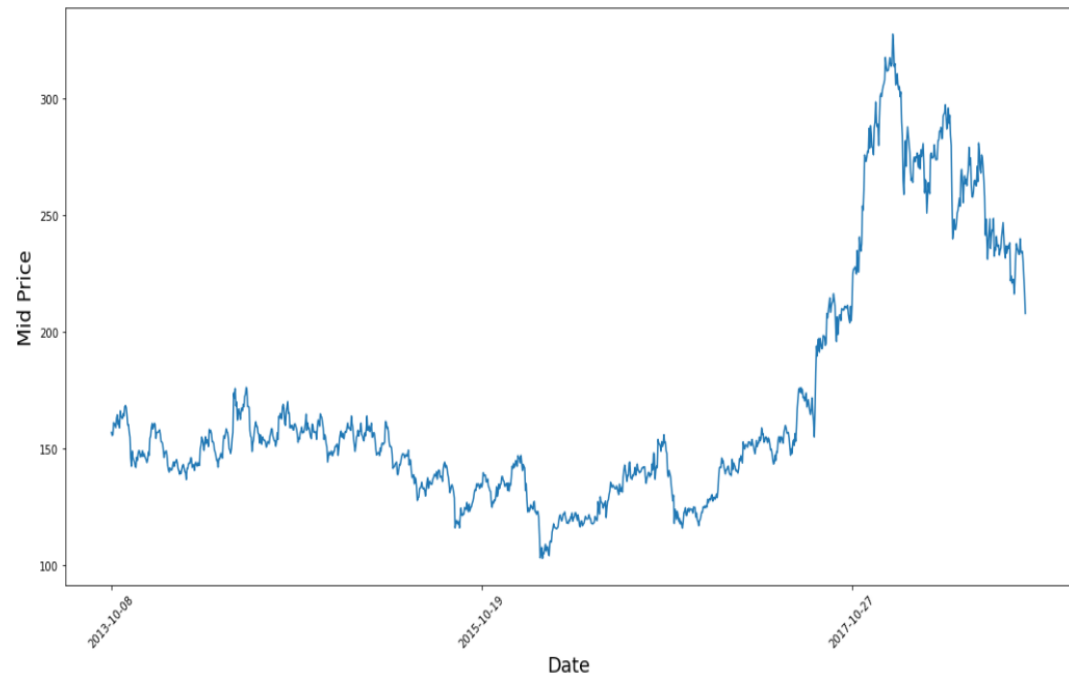
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1235 entries, 1234 to 0
Data columns (total 5 columns):
#   Column  Non-Null Count  Dtype
---  -
0   Date    1235 non-null    object
1   Open    1235 non-null    float64
2   High    1235 non-null    float64
3   Low     1235 non-null    float64
4   Close   1235 non-null    float64
dtypes: float64(4), object(1)
memory usage: 48.4+ KB
```

**Fig 8.4 Screenshot for the summary of the dataset**

**Description:**

The above screenshot displays the concise summary of the dataset.

```
In [5]: plt.figure(figsize = (18,9))
plt.plot(range(data.shape[0]),(data['Open']))
plt.xticks(range(0,data.shape[0],500),data['Date'].loc[::500],rotation=45)
plt.xlabel('Date',fontsize=18)
plt.ylabel('Mid Price',fontsize=18)
plt.show()
```



**Figure 8.5 Screenshot for Ploting the dataset into graph**

### Description:

The above screenshot displays the graph of the stock price and display the X-label has Mid price and Y-label has Date. Display the graph of the dataset.

```

In [9]: regressor = Sequential()

regressor.add(LSTM(units = 75, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.1))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.1))

regressor.add(LSTM(units = 75))
regressor.add(Dropout(0.2))

regressor.add(Dense(units = 1))

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

regressor.fit(X_train, y_train, epochs = 200, batch_size = 64)

Epoch 1/200
15/15 [=====] - 7s 30ms/step - loss: 0.0786
Epoch 2/200
15/15 [=====] - 0s 29ms/step - loss: 0.0112
Epoch 3/200
15/15 [=====] - 0s 28ms/step - loss: 0.0042
Epoch 4/200
15/15 [=====] - 0s 29ms/step - loss: 0.0038
Epoch 5/200
15/15 [=====] - 0s 30ms/step - loss: 0.0035
Epoch 6/200
15/15 [=====] - 0s 29ms/step - loss: 0.0033
Epoch 7/200
15/15 [=====] - 0s 30ms/step - loss: 0.0035
Epoch 8/200
15/15 [=====] - 0s 29ms/step - loss: 0.0031
Epoch 9/200
15/15 [=====] - 0s 30ms/step - loss: 0.0036
Epoch 10/200
15/15 [=====] - 0s 30ms/step - loss: 0.0036

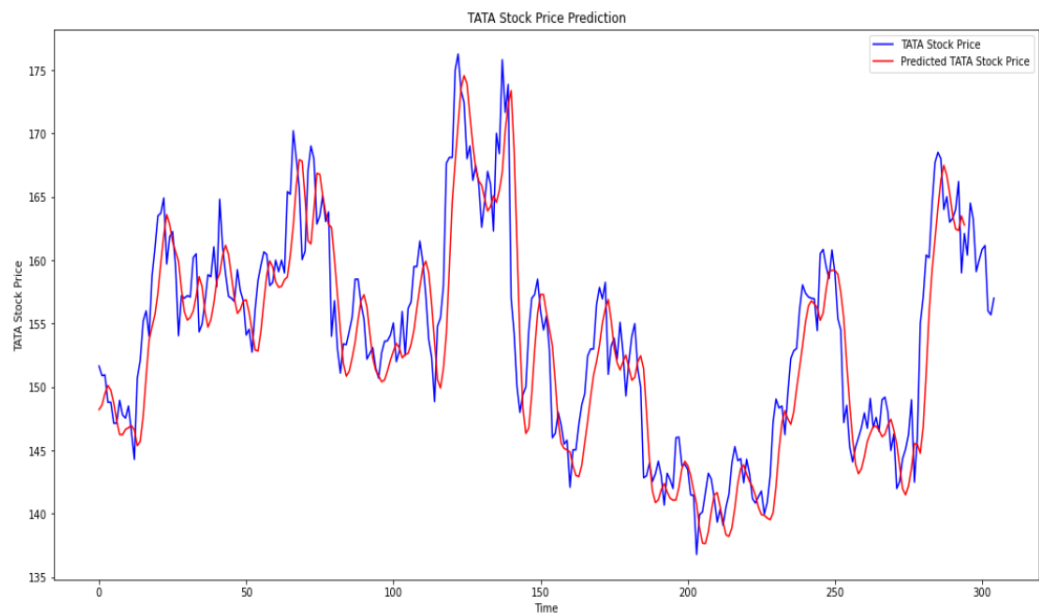
```

**Fig 8.6 Screenshot for LSTM Algorithm**

### Description:

The above screenshot displays the LSTM algorithm, starting with splitting the data into test and train sets and finally applying the model, that is LSTM algorithm.

```
In [11]: plt.figure(figsize = (18,9))
plt.plot(real_stock_price, color = 'blue', label = 'TATA Stock Price')
plt.plot(predicted_stock_price, color = 'red', label = 'Predicted TATA Stock Price')
plt.title('TATA Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('TATA Stock Price')
plt.legend()
plt.show()
```



**Fig 8.7 Screenshot for testing and plotting the dataset**

### Description:

The above screenshot displays the testing the data and plotting the dataset into graph model and displays the X-label has time and Y-label has stock price.

```

In [9]: # testing_set

In [10]: # X_test
zz = regressor.predict([[0.29054816],
                        [0.3054365 ],
                        [0.30904579]])

WARNING:tensorflow:Model was constructed with shape (None, 10, 1) for input KerasTensor(type_spec=TensorSpec(shape=(None, 10, 1), dtype=tf.float32, name='lstm_input'), name='lstm_input', description="created by layer 'lstm_input'"), but it was called on an input with incompatible shape (None, 1, 1).

In [11]: prd = sc.inverse_transform(zz)

In [12]:
real = ( [151.65],
        [150.55],
        [150.35])

In [13]: real
Out[13]: ([151.65], [150.55], [150.35])

In [14]: 151-147.5
Out[14]: 3.5

In [15]: mse = np.mean((real - prd)**2)

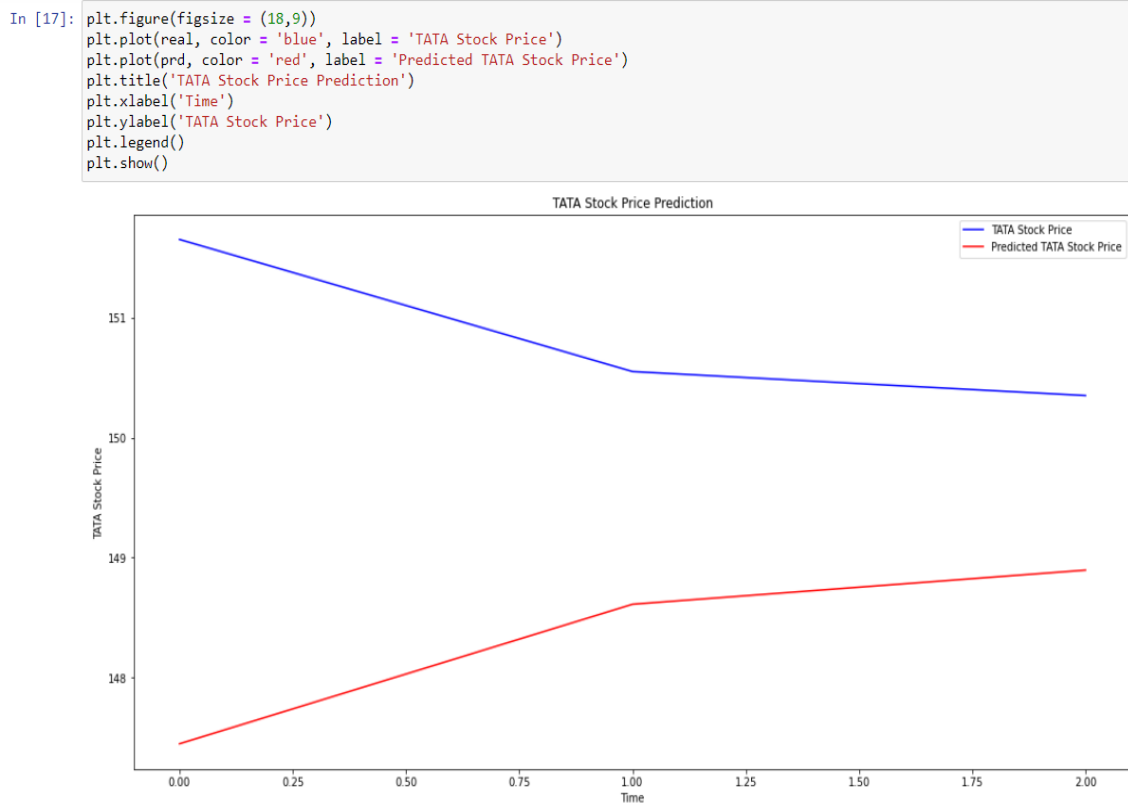
In [16]: (1- mse/100)*100
Out[16]: 92.16520684153771

```

**Fig 8.8 Screenshot for testing the closing price and predict the price**

### Description:

The above screenshot displays testing the closing price in the dataset and predict the future price of the stock.



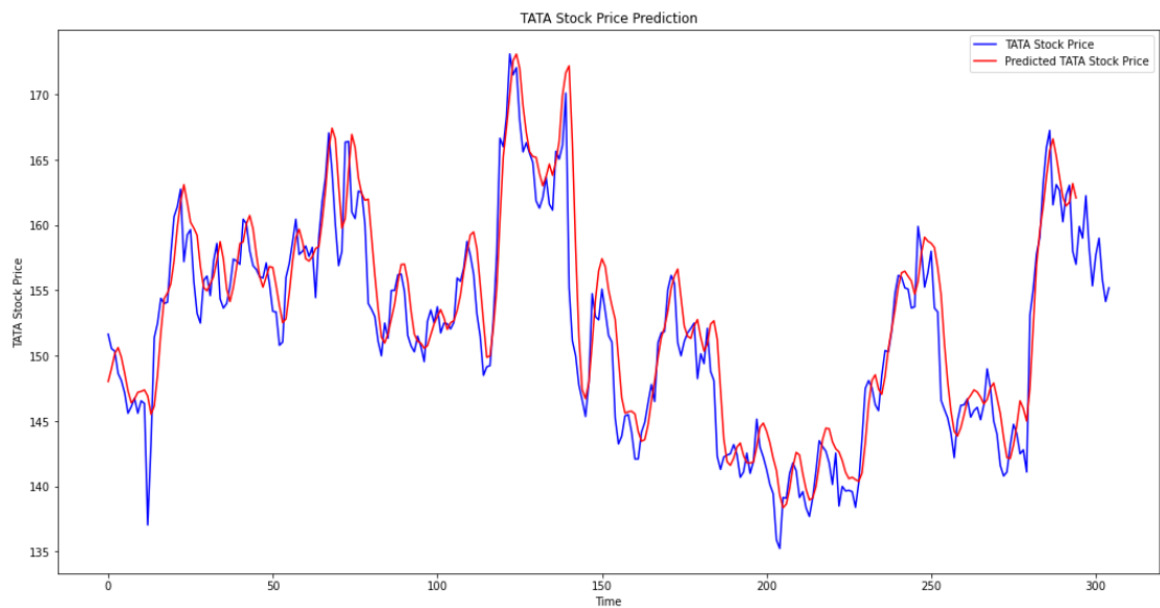
**Fig 8.9 Screenshot for plotting the stock price into X-label and Y-label**

### Description:

The above screenshot displays the graph of the stock price and X-label has time and Y-label has stock price.



```
In [18]: plt.figure(figsize = (18,9))
plt.plot(real_stock_price, color = 'blue', label = 'TATA Stock Price')
plt.plot(predicted_stock_price, color = 'red', label = 'Predicted TATA Stock Price')
plt.title('TATA Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('TATA Stock Price')
plt.legend()
plt.show()
```



**Fig 8.10 Screenshot for plotting the predicted price and old data price**

### Description:

The above screenshot displays the comparison of old dataset price and resultant(predicted) price of the stock.

## **CONCLUSION**

## CONCLUSION

Evaluating the Stock market prediction has at all times been tough work for analysis. The attempt to make of vast written data to forecast the stock market indices. If we join both techniques of textual mining and numeric time series analysis the accuracy in predictions can be achieved. Artificial neural network is qualified to forecast BSE/NSE market upcoming trends. Financial analysts, investors can use this prediction model to take trading decision by observing market behaviour.

The **future scope** of the project work on refining key phrases extraction will definitely produce better results. Enhancements in the preprocessor unit of this system will help in improving more accurate predictability in stock market.

- Twitter feeds message board, Extracting RSS feeds and news.
- Considering internal factors of the company likes Sales, Assets etc.

# **BIBLIOGRAPHY**

## **BIBLIOGRAPHY**

### **Books referred:**

- (1) Python Programming: A Modern Approach by Vamsi Kurama, Pearson
- (2) Learning Python by Mark Lutz, Orielly
- (3) Introduction to machine learning with python by Andreas C.Muller & Srash Guido,Oreilly
- (4) Hands on machine learning with scikit-learn, tensorflow and keras by Aurelian Geron, Oreilly

### **References:**

- [1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose<sup>1</sup>, Giridhar Maji, Narayan C. Debnath, Soumya Sen
- [2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.
- [3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, "Predicting Stock Prices Using LSTM" in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015
- [4] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning
- [5] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.

### **Websites referred:**

- (1) Wikipedia, URL: <https://en.wikipedia.org>
- (2) Google, URL: <http://www.google.co.in>.
- (3) For Python & Machine Learning: <https://www.javatpoint.com>

19A91F0004

---

ORIGINALITY REPORT

---

30%

SIMILARITY INDEX

---

PRIMARY SOURCES

---

1	<a href="http://www.slideshare.net">www.slideshare.net</a> Internet	896 words — 8%
2	<a href="http://ppm.smu.ca">ppm.smu.ca</a> Internet	292 words — 3%
3	<a href="http://mycodestock.com">mycodestock.com</a> Internet	252 words — 2%
4	<a href="http://www.coursehero.com">www.coursehero.com</a> Internet	217 words — 2%
5	<a href="http://docplayer.net">docplayer.net</a> Internet	184 words — 2%
6	<a href="http://ichi.pro">ichi.pro</a> Internet	180 words — 2%
7	<a href="http://www.irjet.net">www.irjet.net</a> Internet	148 words — 1%
8	<a href="http://www.journalppw.com">www.journalppw.com</a> Internet	121 words — 1%
9	<a href="http://www.ijeat.org">www.ijeat.org</a> Internet	118 words — 1%
10	<a href="http://grietinfo.in">grietinfo.in</a> Internet	

86 words — 1%

11 1croreprojects.com  
Internet

84 words — 1%

12 pdfcoffee.com  
Internet

84 words — 1%

13 www.geeksforgeeks.org  
Internet

83 words — 1%

14 machinelearningprojects.net  
Internet

79 words — 1%

15 marswebsolutions.files.wordpress.com  
Internet

75 words — 1%

16 analyticsindiamag.com  
Internet

73 words — 1%

17 stackoverflow.com  
Internet

71 words — 1%

18 www.dotnetcoders.com  
Internet

71 words — 1%

19 www.ukessays.com  
Internet

68 words — 1%

20 asianssr.org  
Internet

65 words — 1%

21 comp786.blogspot.com  
Internet

62 words — 1%

22 www.upgrad.com

---

EXCLUDE QUOTES            ON  
EXCLUDE BIBLIOGRAPHY   ON

EXCLUDE SOURCES        OFF  
EXCLUDE MATCHES        < 60 WORDS