STOCK MARKET PREDICTION USING ML

A Project report submitted in
Partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

By

R SAMPADA **19671A0541**

Under the esteemed guidance of

Mrs. G. SWAPNA

ASSOCIATE PROFESSOR

Department of Computer Science and Engineering

J.B. Institute of Engineering & Technology

UGC AUTONOMOUS

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated by JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist-75 (TG)

2019-2023

J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

UGC AUTONOMOUS

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated by JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist-75 (TG)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **“STOCK MARKET PREDICTION USING ML”** submitted to the Department of Computer Science and Engineering, J.B Institute of Engineering & Technology, in accordance with Jawaharlal Nehru Technological University regulations as partial fulfillment required for successful completion of Bachelor of Technology is a record of bonafide work carried out during the academic year 2022-23 by,

R SAMPADA	**19671A0541**

Internal Guide	**Head of the Department**
Mrs. G. SWAPNA	Dr. G. SREENIVASULU
ASSOCIATE PROFESSOR	ASSOCIATE PROFESSOR

External Examiner

J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY
UGC AUTONOMOUS
(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated by JNTUH)
Yenkapally, Moinabad Mandal, R.R. Dist-75 (TG)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby certify that the Mini Project report entitled **"STOCK MARKET PREDICTION USING ML"** carried out under the guidance of our faculty, **Mrs. G. SWAPNA**, **Associate Professor** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering.** This is a record of bonafide work carried out by me and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to anyother university or institute for the award of any other degree or diploma.

Date: / /

R SAMPADA **19671A0541**

ACKNOWLEDGEMENT

At outset I express my gratitude to almighty lord for showering his grace and blessings upon me to complete this Mini Project. Although my name appears on the cover of this book, many people have contributed in some form or the other to this project development. I could not have done this Project without the assistance or support of each of the following.

First of all, I am highly indebted to **Dr. P. C. KRISHNAMACHARY**, Principal for giving us the permission to carry out this Mini Project.

I would like to thank **Dr. G. SREENIVASULU ,**Professor & Head of the Department of COMPUTER SCIENCE AND ENGINEERING**,** for being moral support throughout the period of the study in the Department.

I am grateful to **Mrs.G.SWAPNA,** Associate Professor COMPUTER SCIENCE AND ENGINEERING, for her valuable suggestions and guidance given by her during the execution of this Project work.

I would like to thank Teaching and Non-Teaching Staff of Department of Computer Science and Engineering for sharing their knowledge with us.

R SAMPADA **19671A0541**

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Stock trading is one of the most important activities. Stock prediction refers to the future predictions of the stock. This project explains the prediction of a stock using Deep Learning. We use different algorithms to predict the stock prices in the future. The programming language is used to predict the stock market using Deep learning is Python. In this, we propose a Deep Learning (DL) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a Deep learning technique called Long Short- Term Memory (LSTM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies. The rapid advancement in artificial intelligence and machine learning techniques, availability of large-scale data, and increased computational capabilities of the machine opens the door to develop sophisticated methods in predicting stock price. In the meantime, easy access to investment opportunities has made the stock market more complex and volatile than ever. The world is looking for an accurate and reliable predictivemodel which can capture the market's highly volatile and nonlinear behavior in a holistic framework. This study uses a long short-term memory (LSTM), a particular neural network architecture, to predict the next-day closing price of the S&P 500 index. A well- balanced combination of nine predictors is carefully constructed under the umbrella of the fundamental market data, macroeconomic data, and technical indicators to capture the behavior of the stock market in a broader sense. Single layer and multilayer LSTM models are developed using the chosen input variables, and their performances are compared usingstandard assessment metrics– Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Correlation Coefficient (R). The experimental results show that the single layer LSTM model provides a superior fit and high prediction accuracy compared toMultilayer LSTM models.

1. INTRODUCTION

Basically, quantitative traders with a lot of money from stock markets buy stocks derivatives and equities at a cheap price and later on selling them at high price. The trendin a stock market prediction is not a new thing and yet this issue is kept being discussed by various organizations. There are two types to analyze stocks which investors perform before investing in a stock, first is the fundamental analysis, in this analysis investors look at the intrinsic value of stocks, and performance of the industry, economy, political climateetc. to decide that whether to invest or not. On the other hand, the technical analysis it is anevolution of stocks by the means of studying the statistics generated by market activity, such as past prices and volumes. In the recent years, increasing prominence of deep learning in various industries have enlightened many traders to apply deep learning techniques to the field, and some of them have produced quite promising results. Thispaper will develop a financial data predictor program in which there will be a dataset storing all historical stock prices and data will be treated as training sets for the program. The main purpose of the prediction is to reduce uncertainty associated to investment decision making. Stock Market follows the random walk, which implies that the best prediction you can have about tomorrow's value is today's value. Indisputably, the forecasting stock indices is very difficult because of the market volatility that needs accurate forecast model. The stock market indices are highly fluctuating and it effects the investor's belief. Stock prices are considered to be a very dynamic and susceptible to quickchanges because of underlying nature of the financial domain and in part because of the mix of a known parameters (Previous day's closing price, P/E ratio etc.) and the unknown factors (like Election Results, Rumors etc. There have been numerous attempts to predict stock price with Deep Learning. The focus of each research projects varies a lot in three ways. (1) The targeting price change can be near-term (less than a minute), short-term (tomorrow to a few days later), and a long-term (months later), (2) The set of stocks can bein limited to less than 10 particular stocks, to stocks in particular industry, to generally all stocks. (3) The predictors used can range from a global news and economy trend, to particular characteristics of the company, to purely time series data of the stock price. The probable stock market prediction target can be the future stock price or the volatility of the prices or market trend. In the prediction there are two types like dummy and a real time prediction which is used in stock market prediction system. In Dummy prediction they have define some set of rules and predict the future price of shares by calculating the shares of the company. The stock price fluctuations are uncertain, and there are many interconnected reasons behind the scene for such behavior. The possible cause could be theglobal economic data, changes in the unemployment rate, monetary policies of influencing countries, immigration policies, natural disasters, public health conditions, and several others. All the stock market stakeholders aim to make higher profits and reduce the risks from the thorough market evaluation. The major challenge is gathering the multifaceted information, putting them together into one basket, and constructing a reliable model for accurate predictions.

Stock price prediction is a complex and challenging task for companies, investors, and equity traders to predict future returns. Stock markets are naturally noisy, non-parametric, non-linear, and deterministic chaotic systems (Ahangar, Yahyazadehfar, & Pournaghshband, 2010). It creates a challenge to effectively and efficiently predict the future price. Feature selection from the financial data is another difficult task in the stock prediction for which many approaches have been suggested (Hoseinzade & Haratizadeh, 2019). There has been a trend in which some researchers use only technical indicators, whereas others use historical data (Di Persio and Honchar, 2016, Kara et al., 2011, Nelson et al., 2017, Patel et al., 2015, Qiu and Song, 2016, Wang and Kim, 2018). The performance of the predictive model may not be top-notch due to the use of limited features. On the flip side, if all the available features from the financial market are included, the model could be complex and difficult to interpret. In addition, the model performance may be worse due to collinearity among multiple variables.

A proper model developed with an optimal set of attributes can predict stock price reasonably well and better inform the market situation. A plethora of research has been published to study how certain variables correlate with stock price behavior. A varying degree of success is seen concerning the accuracy and robustness of the models. One possible reason for not achieving the expected outcome could be in the variable selection process. There is a greater chance that the developed model performs reasonably better if a good combination of features is considered. One of the contributions of this study is selecting the variables by looking meticulously at multiple aspects of the economy andtheir potential impact in the broader markets. Moreover, a detailed justification is supplied why the specific explanatory variables are chosen in the present context in Section 4.

The field of quantitative analysis in finance has a long history. Several models ranging from naive to complex have been developed so far to find the solution to financial problems. However, not all quantitative analyses or models are fully accepted or widely used. One of the first attempts was made in the seventies by two British statisticians, Box and Jenkins, using mainframe computers (Hansen, McDonald, & Nelson, 1999). They developed the Auto-Regressive Integrated Moving Average (ARIMA) model utilizing onlythe historical data of price and volume. The ARIMA is used to handle only stationary time series data by default. Performance can be abysmal if it is used for non-stationary data. Therefore, it is essential to convert non-stationary time series data to stationary before implementation, which may lose the original structure and interpretability of the feature. With very few exceptions, almost all classical models assume that data has a linearrelationship. This assumption vividly raises the questions about the robustness of theclassical time series models as the real-world time series data are often nonlinear.

Things were getting more interesting from the eighties because of the development in data analysis tools and techniques. For instance, the spreadsheet was invented to model financial performance, automated data collection became a reality, and improvements in computing power helped predictive models to analyze the data quickly and efficiently. Because of the availability of large-scale data, advancement in technology, and inherent problem associated with the classical time series models, researchers started to buildmodels by unlocking the power of artificial neural networks and deep learning techniques in the area of sequential data modeling and forecasting. These methods are capable of learning complex and non-linear relationships compared to traditional methods. They are more efficient in extracting the most important information from the given input variables.

Several deep learning architectures have been developed to deal with various problems andthe intrinsic structure of datasets. Information flows only in the forward direction in a basic feedforward neural network architecture. Since each input is processed independently, it does not retain information from the previous step. Thus, these models are ineffective in dealing with sequential data where series of prior events are essential in predicting future events. Recurrent neural networks (RNN) are designed to perform such tasks. The RNN architecture consists of loops, allowing relevant information to persist over time. Information is being passed from one timestep to the next internally within the network. Therefore, the RNN is more suitable for sequential data modeling and time series applications such as stock market predictions, language translations, auto-completion in messages/emails, and signal processing. During the training process of the RNN, the costor error is calculated between the predicted values and the actual values from a labeled training dataset. The error is minimized by repeatedly updating the networks' parameters (weights and biases) until the lowest possible value is obtained. The training process utilizes a gradient, the rate at which cost changes with respect to each parameter. The gradient provides a direction to move in the error surface by adjusting the parameters iteratively. This strategy is called backpropagation, where the error is propagated backwardfrom the output layer all the way up to the input layer. One of the challenges of this technique is that parameters can be anywhere in the networks, and finding a gradient involves calculations of partial derivatives with respect to all the parameters. This process sometimes needs a long chain rule, especially for the parameters in earlier layers of the networks. As a result, gradients could ultimately vanish or decay back through the networks, known as the vanishing gradient problem, a common issue in neural networks training. Unfortunately, this problem also persists in the RNN architecture. LSTM, atypical recurrent neural network architecture, is designed to overcome the vanishing gradient problem (Hochreiter, 1998). Memorizing information for a longer period of time is the default behavior of the LSTM model.

This study considers the computational framework to predict the stock index price usingthe LSTM model, the improved version of neural networks architecture for time series data. The bird's-eye view of the proposed research framework via the schematic diagram isexpressed in Fig. 1. As outlined in the diagram, the proposed study utilizes the carefully selected features from fundamental, macroeconomic, and technical data to build the model.After that, the collected data has been normalized using the min–max normalization technique. Then input sequence for the LSTM model is created using a specific time step. The hyperparameters such as number of neurons, epochs, learning rate, batch size, andtime step have been incorporated in the model. The regularization techniques have been utilized to overcome the over-fitting problems. Once the hyperparameters are tuned, the input data is fed into the LSTM model to predict the closing price of the stock market index. The quality of the proposed model is assessed through RMSE, MAPE, and R.

In a nutshell, plenty of research has been done in predicting the stock market. Some research focuses on complex statistical or machine learning techniques without focusing on the type of attributable variables. Others use only the fundamental data without exploring additional factors that could influence the stock market prediction. There is a need to develop a model with a good combination of features of the stock market variables and simplicity in model architecture. Thus, our contribution is to create a model without addingany complexity in model architecture and maintaining well-balanced set of variables to capture the behavior of the stock market from multiple dimensions.Computational advances have led to introduction of deep learning techniques for the predictive systems in financial markets. In this paper we are using a Deep Learning technique i.e., Long Short- Term Memory (LSTM) in order to predict the stock market and we are using Python language for programming.

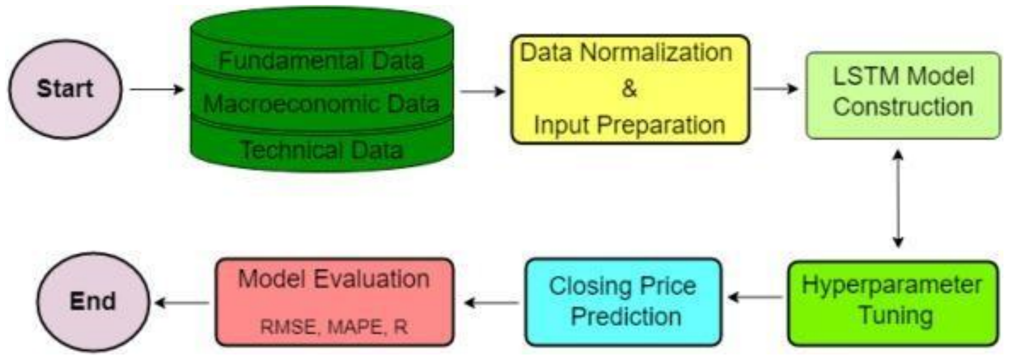


FIG 1.1 SCHEMATIC DIAGRAM OF THE PROPOSED RESEARCH FRAMEWORK

1.1 EXISTING SYSTEM

- **Stock Market Prediction Using Machine Learning**

The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India. In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

- **Forecasting the Stock Market Index Using Artificial Intelligence Techniques**

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools. Two techniques are used to benchmark the AI techniques, namely, Autoregressive Moving Average (ARMA) which is linear modelling technique and random walk (RW) technique. The experimentation was performed on data obtained from the Johannesburg Stock Exchange. The data used was a series of past closing prices of the All Share Index. The results showed that the three techniques have the ability to predict the future price of the Index with an acceptable accuracy. All three artificial intelligence techniques outperformed the linear model.However, the random walk method out performed all the other techniques. These techniques showan ability to predict the future price however, because of the transaction costs of trading in the market, it is not possible to show that the three techniques can disprove the weak form of market efficiency. The results show that the ranking of performances support vector machines, neuro-fuzzy systems, multilayer perceptron neural networks is dependent on the accuracy measure used.

- **Indian stock market prediction using artificial neural networks on tickdata**

The research work done by Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. A stock market is a platform for trading of a company's stocks and derivatives at an agreed price. Supply and demand of shares drive the stock market. In any country stock market is one of themost emerging sectors. Nowadays, many people are indirectly or directly related to this sector. Therefore, it becomes essential to know about market trends. Thus, with the development of the stock 5 market, people are interested in forecasting stock price. But, due to dynamic nature andliable to quick changes in stock price, prediction of the stock price becomes a challenging task. Stock m Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, whichare useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event. Experiments on three event-related tasks, i.e.,event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market1. Markets are mostly a nonparametric, non-linear, noisy and deterministic chaotic system (Ahangar et al. 2010). As the technology is increasing, stock traders are moving towards to use Intelligent Trading Systems rather than fundamental analysis for predicting prices of stocks, which helps them to take immediate investment decisions. One of the main aims of a trader is to predict the stock price such that he can sell it before its value decline, or buy the stock before the price rises. The efficient market hypothesis states that it is not possible to predict stock prices and that stock behaves in the random walk. It seems to be very difficult to replace the professionalism of an experienced trader for predicting the stock price. But because of the availability of a remarkable amount of data and technological advancements we can nowformulate an appropriate algorithm for prediction whose results can increase the profits for traders orinvestment firms. Thus, the accuracy of an algorithm is directly proportional to gains made by using the algorithm.

- **The Stock Market and Investment**

The research work done by Manh Ha Duong Boriss Siliverstovs. Investigating the relation between equity prices and aggregate investment in major European countries including France, Germany, Italy, the Netherlands and the United Kingdom. Increasing integration of European financial marketsis likely to result in even stronger correlation between equity prices in different European countries. This process can also lead to convergence in economic development across European countries if developments in stock markets influence real economic components, such as investment and consumption. Indeed, our vector autoregressive models suggest that the positive correlation between changes equity prices and investment is, in general, significant. Hence, 6 monetary authorities should monitor reactions of share prices to monetary policy and their effects on the business cycle.

1.2 PROPOSED SYSTEM

- The proposed work aims at improving the current technology using algorithms like LSTM algorithm. The proposed system is about the prediction of stocks in the future. This processhappens in an automated process.
- Automated process means it involves the assistance of deep learning. Large sets of data aretaken into account and used for training and testing in the machine. The trained and tested data are initially separated randomly through algorithms.
- It is then implemented using python programming language with the help of anaconda tool. The text editor used is Jupyter notebook. After all the training and testing of data, the modelwill predict whatever data is being presented at that time, based on present stocks, it will predict whether the price of stock would increase in future or not.
- These predictions happen through an algorithm called Long Short-Term Memory or LSTM.
- The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model,

ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM).

1.3 SYSTEM SCOPE

- The scope of the project is really bright as this can be used in several different areasfor more efficient results. This is mainly used for intraday trading of stock market. No human or computer can perfectly predict the volatile stock market.
- Under "normal" conditions, in most cases, a good neural network will outperform most other current stock market predictors and be a very worthwhile, and potentially profitable aid to investors.
- Should be used as an aid only!

2. LITERATURE SURVEY

During a literature survey, we collected some of the information about Stock market prediction mechanisms currently being used.

1.Survey of Stock Market Prediction Using Machine Learning Approach

The stock market prediction has become an increasingly important issue in the presenttime. One of the methods employed is technical analysis, but such methods do not always yield accurate results. So it is important to develop methods for a moreaccurate prediction. Generally, investments are made using predictions that are obtained from the stock price after considering all the factors that might affect it. The technique that was employed in this instance was a regression. Since financial stock marks generate enormous amounts of data at any given time a great volume of data needs to undergo analysis before a prediction can be made. Each of the techniques listed under regression hasits own advantages and limitations over its other counterparts. One of the noteworthy techniques that were mentioned was linear regression. The way linear regression models work is that they are often fitted using the least squares approach, but they may alternatively be also be fitted in other ways, such as by diminishing the "lack of fit" in some other norm, or by diminishing a handicapped version of the least squares loss function. Conversely, the least squares approach can be utilized to fit nonlinear models.

2.Impact of Financial Ratios and Technical Analysis on Stock Price Prediction Using Random Forests

The use of machine learning and artificial intelligence techniques to predict the prices of the stock is an increasing trend. More and more researchers invest their time every day in coming up with ways to arrive at techniques that can further improve the accuracy of the stock prediction model. Due to the vast number of options available, there can be n number of ways on how to predict the price of the stock, but all methods don't work the same way. The output varies for each technique even if the same data set is being applied. In the cited paper the stock price prediction has been carried out by using the random forest algorithm is being used to predict the price of the stock using financial ratios form the previous quarter. This is just one wayof looking at the problem by approaching it using a predictive model, using the random forest to predict the future price of the stock from historical data. However, there are always other factors that influence the price of the stock, such as sentiments of the investor, public opinion about the company, news from various outlets, and even events that cause the entire stock market to fluctuate. By using the financial ratio along with a model that can effectively analyze sentiments the accuracy of the stock price prediction model can be increased.

3.Stock Market Prediction via Multi-Source Multiple Instance Learning Accurately predicting the stock market is a challenging task, but the modern web hasproved to be a very useful tool in making this task easier. Due to the interconnectedformat of data, it is easy to extract certain sentiments thus making it easier to establishrelationships between various variable and roughly scope out a pattern of investment.Investment pattern from various firms show sign of similarity, and the key tosuccessfully predicting the stock market is to exploit these same consistenciesbetween the data sets. The way stock market information can be predictedsuccessfully is by using more than just technical historical data, and using othermethods like the use of sentiment analyzer to derive an important connection betweenpeople's emotions and how they are influenced by investment in specific stocks. Onemore important segment of the prediction process was the extraction of importantevents from web news to see how it affected stock prices.

4. Stock Market Prediction: Using Historical Data Analysis

The stock market prediction process is filled with uncertainty and can be influenced by multiple factors. Therefore, the stock market plays an important role in business and finance. The technical and fundamental analysis is done by sentimental analysis process. Social media data has a high impact due to its increased usage, and it can be helpful in predicting the trend of the stock market. Technical analysis is done using by applying machine learning algorithms on historical data of stock prices. The method usually involves gathering various social media data, news to extract sentiments expressed by individuals. Other data like previous year stock prices are also considered. The relationship between various data points is considered, and a prediction is made on these data points. The model was able to make predictionsabout future stock values.

5. A Survey on Stock Market Prediction Using SVM

The recent studies provide a well-grounded proof that most of the predictive regression models are inefficient in out of sample predictability test. The reason for this inefficiency was parameter instability and model uncertainty. The studies also concluded the traditional strategies that promise to solve this problem. Support vector machine commonly known as SVM provides with the kernel, decision function, and sparsity of the solution. It is used to learn polynomial radial basis function and the multi-layer perceptron classifier. It is a training algorithm for classification and regression, which works on a larger dataset. There are many algorithms in the market but SVM provides with better efficiency and accuracy. The correlation analysis between SVM and stock market indicates strong interconnection between the stock prices and the market index.

6. Predicting Stock Price Direction Using Support Vector Machines

Financial organizations and merchants have made different exclusive models to attempt and beat the market for themselves or their customers, yet once in a while has anybody accomplished reliably higher-than-normal degrees of profitability. Nevertheless, the challenge of stock forecasting is so engaging in light of the fact that the improvement of only a couple of rate focuses can build benefit by a large number of dollars for these organizations.

7. A Stock Market Prediction Method Based on Support Vector Machines (SVM)and Independent Component Analysis (ICA)

The time series prediction problem was researched in the work centers in the various financial institution. The prediction model, which is based on SVM and independent analysis, combined called SVM-ICA, is proposed for stock market prediction. Varioustime series analysis models are based on machine learning. The SVM is designed to solve regression problems in non-linear classification and time series analysis. The generalization error is minimized using an approximate function, which is based on risk diminishing principle. Thus, the ICA technique extracts various important features from the dataset. The time series prediction is based on SVM. The result ofthe SVM model was compared with the results of the ICA technique without using a preprocessing step.

8. Machine Learning Approach In Stock Market Prediction

The vast majority of the stockbrokers while making the prediction utilized the specialized, fundamental or the time series analysis. Overall, these techniques couldn't be trusted completely, so there emerged the need to give a strong strategy to financial exchange prediction. To find the best accurate result, the methodology chose to be implemented as machine learning and AI along with supervised classifier. Results were tried on the binary classification utilizing SVM classifier with an alternate set of a feature list. The greater part of the Machine Learning approach for taking care of business [2] issues had their benefit over factual techniques that did exclude AI, despite the fact that there was an ideal procedure for specific issues. Swarm Intelligence [2] optimization method named Cuckoo search was most easy to accommodate the parameters of SVM. The proposed hybrid CS-SVM strategy exhibited the performance to create increasingly exact outcomes in contrast with ANN. Likewise, the CS-SVM display [2] performed better in the forecasting of the stock value prediction. Prediction stock cost utilized parse records to compute the predicted, send it to the user, and autonomously perform tasks like buying and selling shares utilizing automation concept. Naïve Bayes Algorithm was utilized.

9. Corporate Communication Network and Stock Price Movements: Insights from Data Mining

This paper tries to indicate that communication patterns can have a very significant effect on an organization's performance. This paper proposed a technique to reveal the performance of a company. The technique deployed in the paper is used to find therelationships between the frequencies of email exchange of the key employees and theperformance of the company reflected in stock values. In order to detect association and non-association relationships, this paper proposed to use a data mining algorithm on a publicly available dataset of Enron Corp. The Enron Corporation was an energy, commodities, and services company based in Houston, Texas whose stock dataset is available for public use.

3.SYSTEM REQUIREMENTS ANALYSIS

3.1 Hardware Requirements:

• RAM: 4 GB

• Storage: 500 GB

• CPU: 2 GHz or faster

• Architecture: 32-bit or 64-bit

3.2 Software Requirements:

• Python 3.5 in Google Colab is used for data pre-processing, model training andprediction.

• Operating System: windows 7 and above or Linux based OS or MAC OS

3.3 Functional requirements:

Functional requirements describe what the software should do (the functions). Think aboutthe core operations.

Because the "functions" are established before development, functional requirements should be written in the future tense. In developing the software for Stock Price Prediction,some of the functional requirements could include:

• The software shall accept the tw_spydata_raw.csv dataset as input.

• The software should shall do pre-processing (like verifying for missing data values)on input for model training.

• The software shall use LSTM ARCHITECTURE as main component of the
software.

• It processes the given input data by producing the most possible outcomes of a CLOSING STOCK
PRICE

Notice that each requirement is directly related to what we expect the software to do. They represent some of the core functions.

3.4 Non-Functional requirements:

Product properties

• Usability: It defines the user interface of the software in terms of simplicity of
understanding the user interface of stock prediction software, for any kind of stock

trader and other stakeholders in stock market.

• Efficiency: maintaining the possible highest accuracy in the closing stock prices in
shortest time with available data.

Performance: It is a quality attribute of the stock prediction software that describes the responsiveness to various user interactions with it.

4.SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

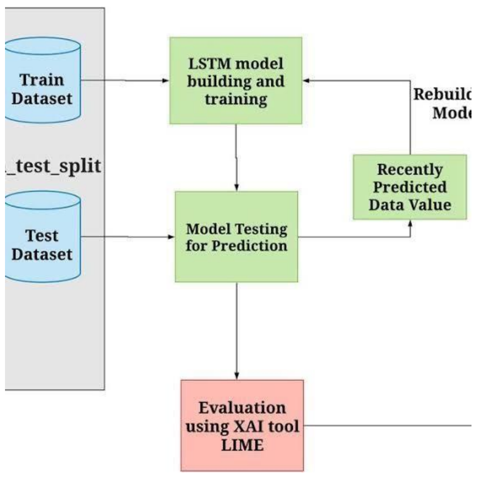

Train Dataset
_test_split
Test Dataset
LSTM model building and training
Model Testing for Prediction
Recently Predicted Data Value
Rebuilc Mode
Evaluation using XAI tool LIME

FIG 4.1.1 SYSTEM ARCHITECTURE

1) Preprocessing the data:

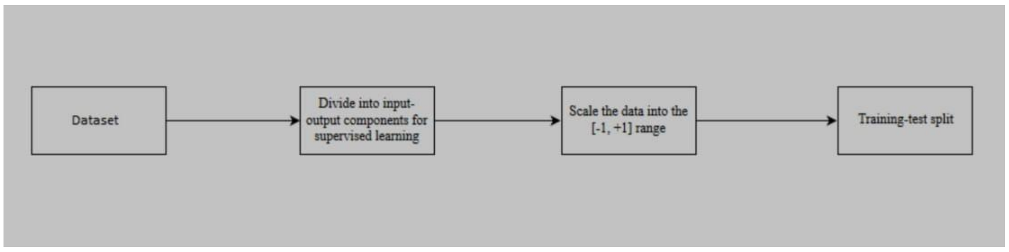

Dataset
Divide into input-output components for supervised learning
Scale the data into the [-1, +1] range
Training-test split

FIG 4.1.2 PREPROCESSING THE DATA

2) Overall Architecture

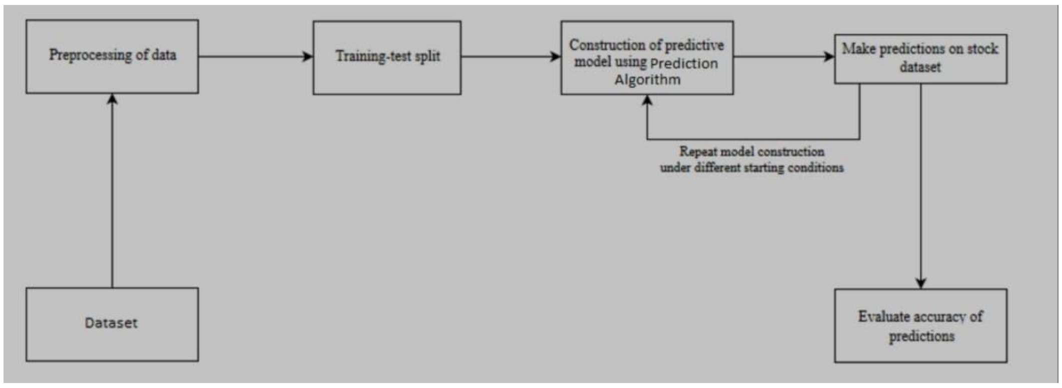

Preprocessing of data
Training-test split
Construction of predictive model using Prediction Algorithm
Make predictions on stock dataset
Repeat model construction under different starting conditions
Dataset
Evaluate accuracy of predictions

FIG 4.1.3 OVERALL ARCHITECTURE

4.2 STRUCTURE CHART:

A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming toarrange program modules into a tree. Each module is represented by a box, which contains the module's name.

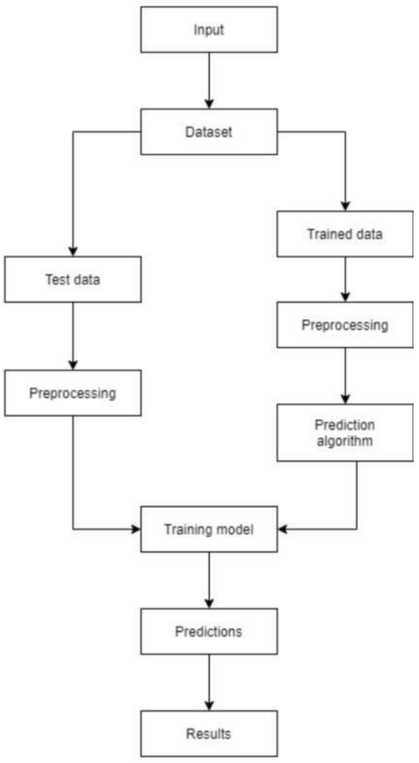


FIG 4.2.1 STRUCTRE CHART

4.3DATA FLOW DIAGRAM:

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process ora system (usually an information system). The DFD also provides information about the outputs and inputs ofeach entity and the process itself. Specific operations based on the data can be represented by a flowchart. There are several notations for displaying data-flow diagrams.

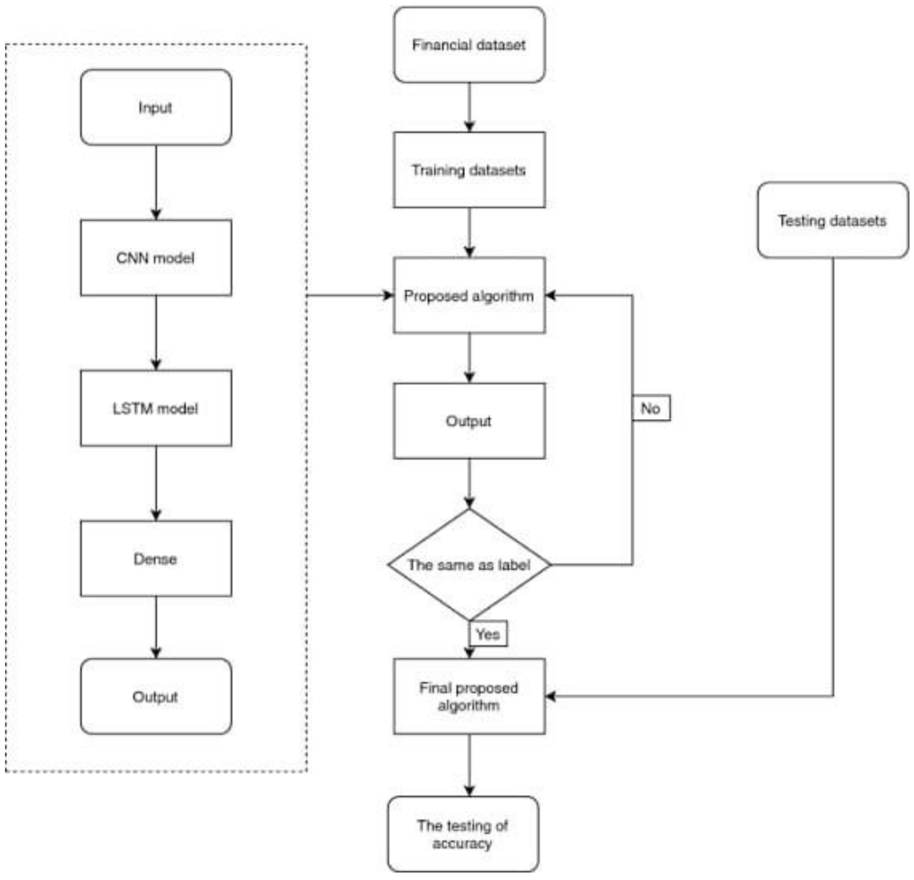


FIG 4.3.1 DATA FLOW DIAGRAM

4.4 UML DIAGRAMS

UML is an acronym that stands for Unified Modeling Language. Simply put, UML isa modern approach to modeling and documenting software. In fact, it’s one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the oldproverb says: “a picture is worth a thousand words”. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

UML diagrams, in this case, are used to communicate different aspects and characteristics of a system. However, this is only a top-level view of the system and will most probably not include all the necessary details to execute the project until the very end.

There are several types of UML diagrams and each one of them serves a different purpose. The two broadest categories that encompass all other types are Behavioral UML diagram and Structural UML diagram. As the name suggests, some UML diagrams try to analyze and depict the structure of a system or process, whereas other describe the behavior of the system, its actors, and its building components. The different types are as follows:

4.4.1 Use Case Diagram
4.4.2 Class Diagram
4.4.3 Sequence Diagram
4.4.4 Component Diagram
4.4.5 Collaboration Diagram
4.4.6 Object Diagram
4.4.7 State Machine Diagram
4.4.8 Communication Diagram
4.4.9 Deployment Diagram

4.4.1 USE CASE DIAGRAM:

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll usea set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

• Scenarios in which your system or application interacts with people, organizations, or external

systems.

• Goals that your system or application helps those entities (known as actors) achieve.

• The scope of your system

4.

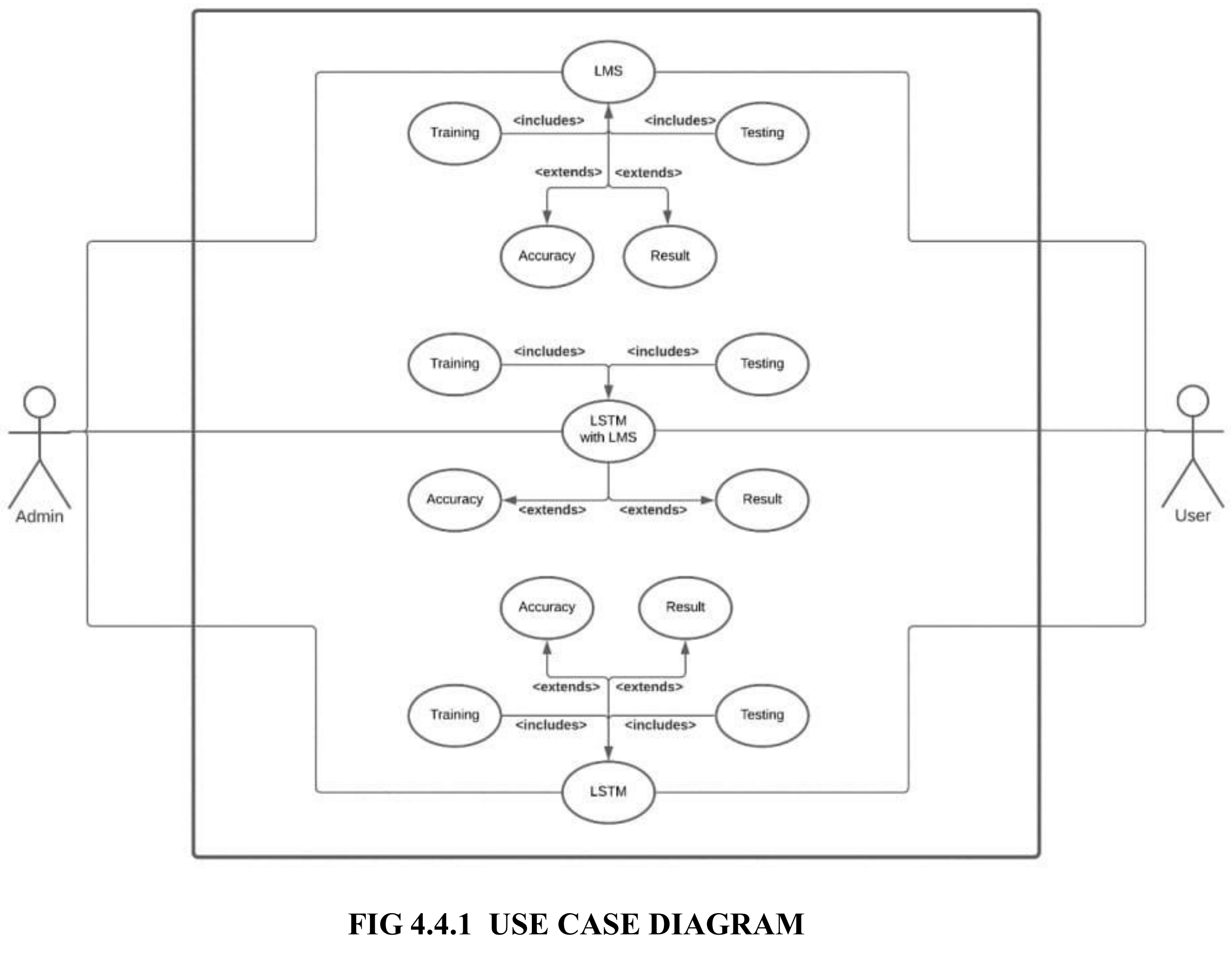


FIG 4.4.1 USE CASE DIAGRAM

4.4.2 SEQUENCE DIAGRAM:

A sequence diagram is a type of interaction diagram because it describes how and in what order agroup of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations. Try drawing asequence diagram to:

• Represent the details of a UML use case.

• Model the logic of a sophisticated procedure, function, or operation.

• See how objects and components interact with each other to complete a process.

• Plan and understand the detailed functionality of an existing or future scenario.

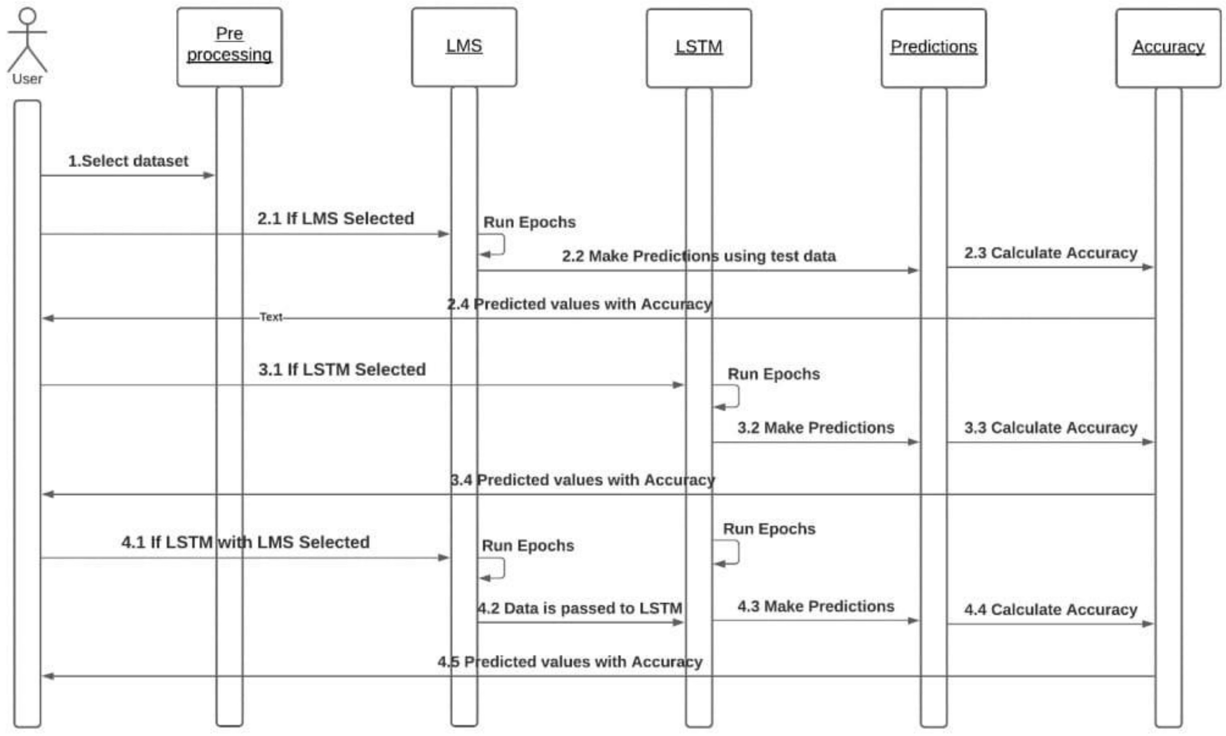


FIG 4.4.2 SEQUENCE DIAGRAM

4.4.3 ACTIVITY DIAGRAM:

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system.
An activity diagram portrays the control flow from a start point to a finish point showing the variousdecision paths that exist while the activity is being executed.

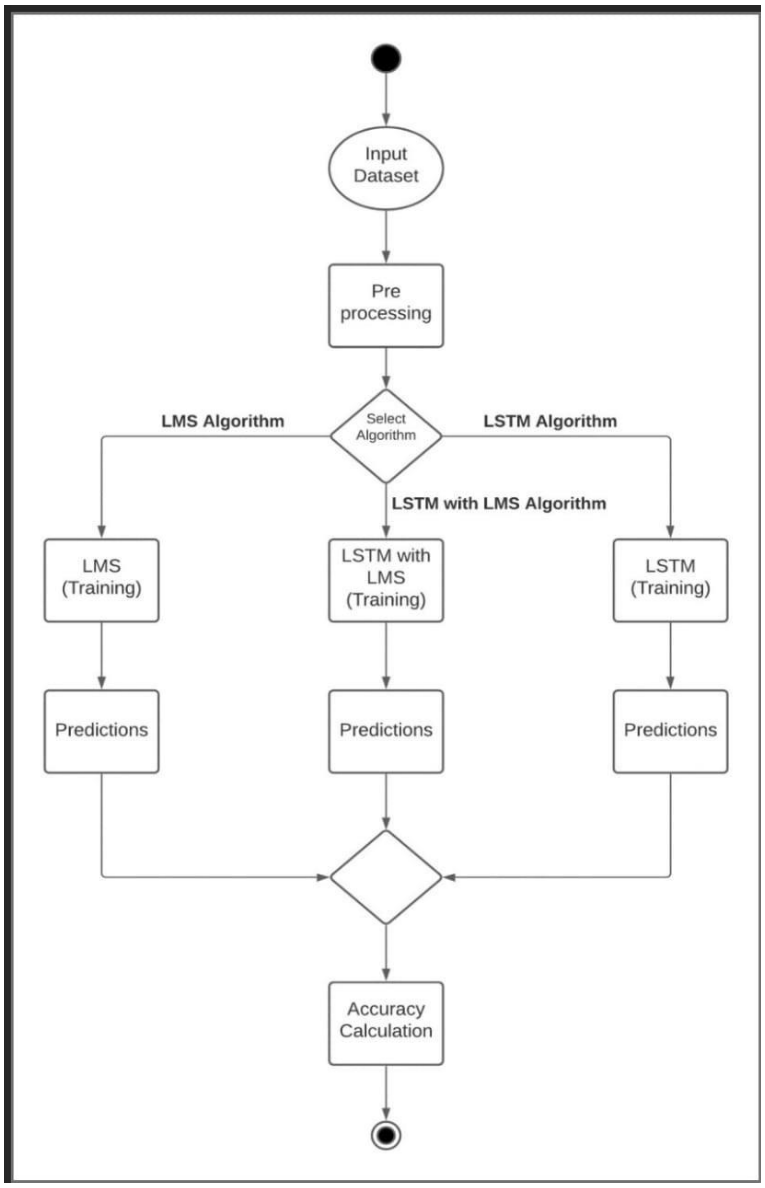


FIG 4.4.3 ACTIVITY DIAGRAM

4.4.4 COLLABORATION DIAGRAM:

Collaboration diagrams are used to show how objects interact to perform the behavior of a particularuse case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case.

They are the primary source of information used to determining class responsibilities and interfaces. The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying itquite different. The collaboration diagrams are best suited for analyzing use cases.

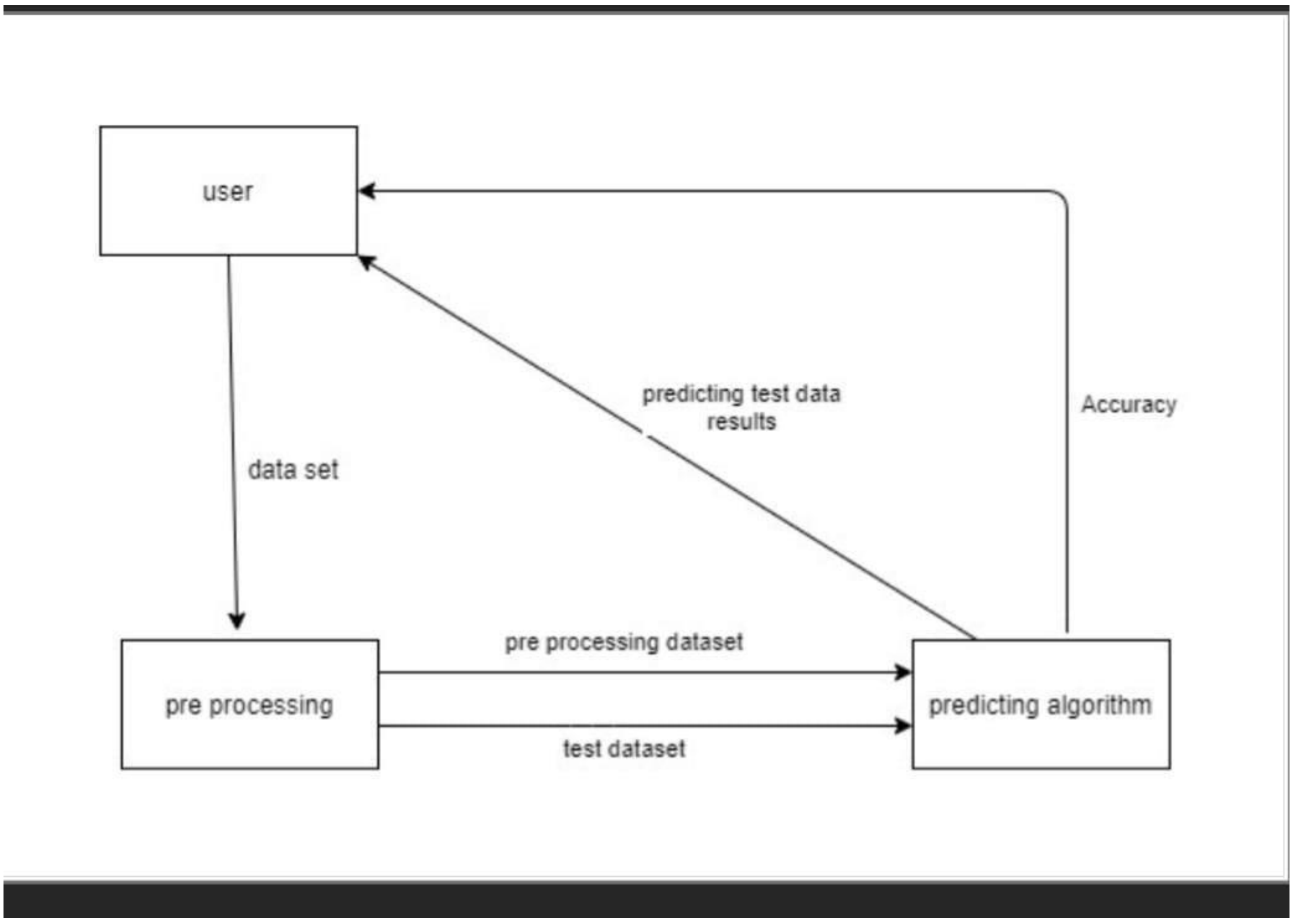


FIG 4.4.4 COLLABORATION DIAGRAM

4.4.5 FLOWCHART DIAGRAM:

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes witharrows.

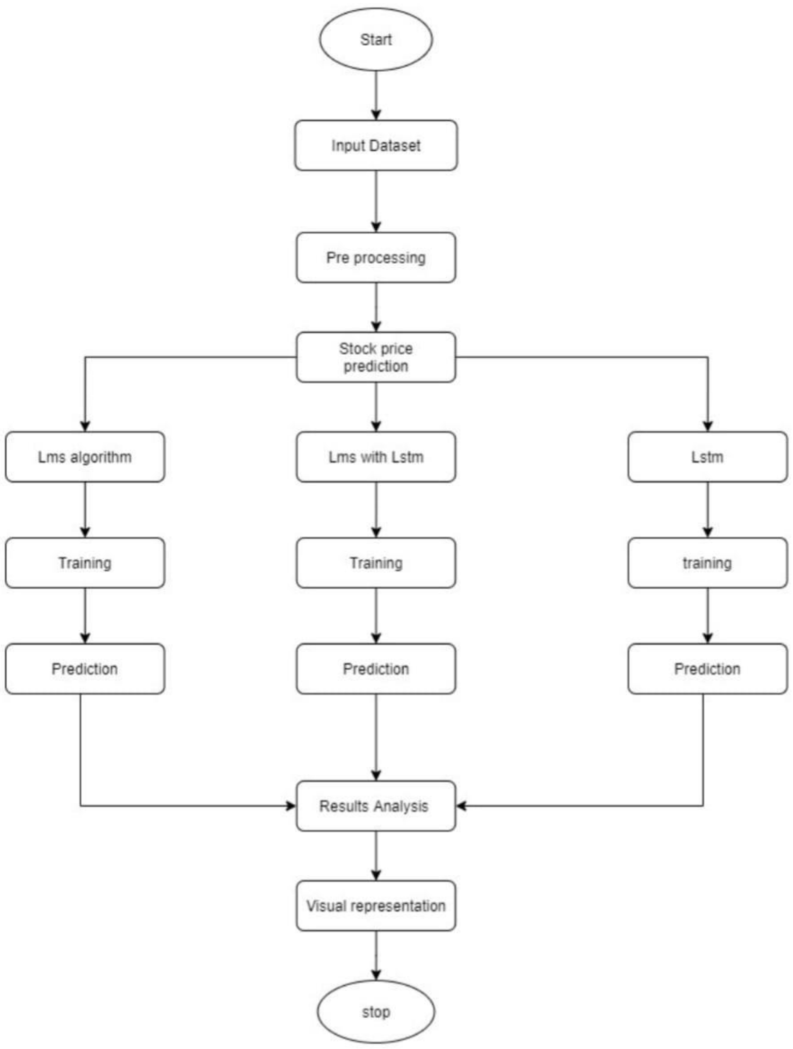


FIG 4.4.5 FLOWCHART DIAGRAM

:

4.4.6 COMPONENT DIAGRAM:

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Component diagrams are used in modeling the physical aspects of object-oriented systems that areused for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

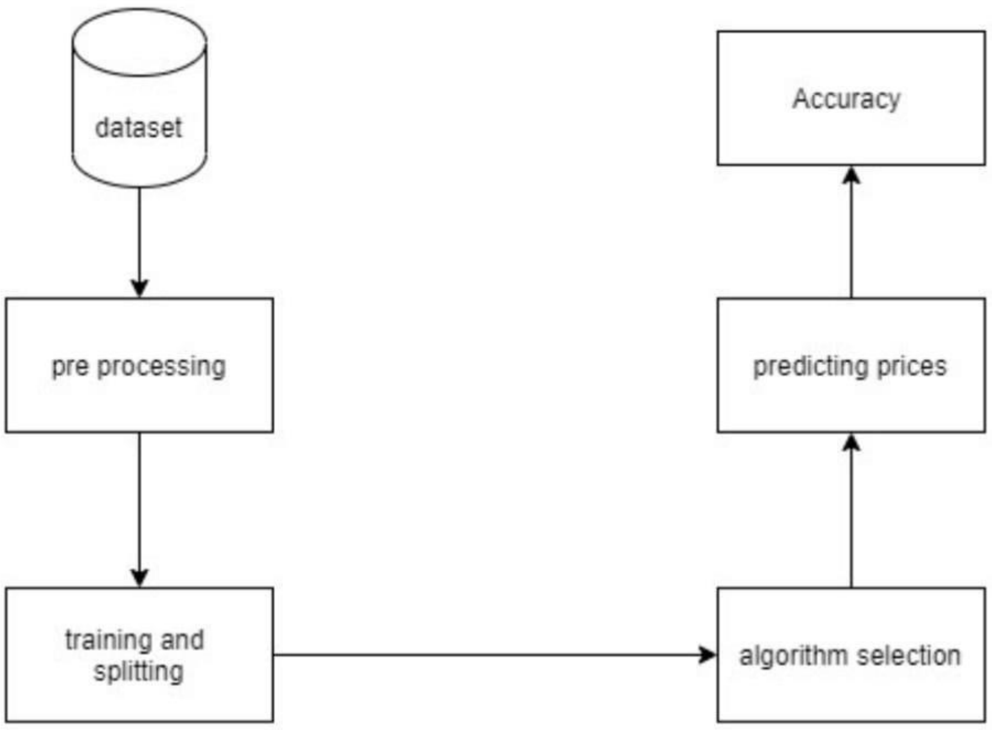


FIG 4.4.6 COMPONENT DIAGRAM

5.IMPLEMENTATION

5.1 MACHINE LEARNING IMPLEMENTATION

Machine Learning Workflow Undertaken:

We can define the Machine learning workflow in 5 stages.

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

What is the Machine learning model?

The machine learning model is nothing but a piece of code; which an engineer or data scientist models by training it with the data according to the need of the project and making the model learn through the data and allowing it to predict or give the solution that we want whenever we ask it to give. So, whenever we give our model the new data which we want it to predict, we will get the predicted value according to the model training, the trained model might or might not perform well on the test data that we want it to predict, due to various reasons, so before trying to train any model we need to make sure that the algorithm that is going to use is appropriate for the desired class that we want to predict and based on the data that we are using.

1. Gathering Data

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that uses different sensors data. Then the sensor data can be connected to the database where we want to store it. But the collected data cannot be used directly for performing the analysis, Since the collected data might be very irrelevant, extremely large values, unorganized text data, or noisy data to the project that we areworking on. Therefore, to solve this problem Data Preparation is done meaning data cleaning.

2. Data pre-processing

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps us to provide accurate and cleaned data for training so that we can get accurate results. In machine learning, there is an 80/20 rule, where, every data scientist should spend his/her 80% time for data pre-processing and 20% time to perform the analysis and build the actual machinelearning model.

What is data pre-processing?

Data pre-processing is a process of cleaning the raw data and making it a meaningful and understandable format. i.e., the data which is collected in the real world is not clean and consists of alot of irrelevant data and inconsistent data, so we first convert our raw into a meaningful way to generate an efficient machine learning model pipeline. In other words, whenever the data is gatheredfrom different

sources it is collected in a raw format and this data isn't feasible for the analysis.

Therefore, certain steps are executed to convert the data into a small clean, and meaningful way, where our model can understand. This part of the process is called data pre-processing.

Why do we need it?

As we know that data pre-processing is a process of cleaning the raw data into clean data so that it can be used to train the model. So, data pre-processing is an essential step to generate a machine learning model to help us predict, classify, forecast the data when we pass unknown data.

Below mentioned are some of the examples which mostly occur in raw data, when the data is collected.

1. Missing data: Missing data/ missing values are the empty spaces in our data that might have occurred due to various reasons, such as Structured missing values, missing completely at random, Missing at Random, missing not at random, each has its method of treating missing values and in thecase of Sensor data, missing values might be high due to technical issues, electricity or other environmental factors.

2. Noisy data: This type of data is also called outliers; this can occur due to human errors (humans manually gathering the data) or some technical problem of the device at the time of collection of data. Outliers can be easily termed as extremely low values or extremely high values which are not related to the particular data variable.

3. Inconsistent data: This type of data error might mostly happen due to human errors (mistakes with the name or values) or duplication of data. Inconsistent in numbering formats or if the data variable is not consistent through all the rows, then we can term it as an inconsistent data variable, making a data variable consistent is very important since it might result in a certain type of bias when the model is trained and it might also result in inaccurate analysis results when we visualize them onto plots.

Three Types of Data

1. Numeric e.g., Income, Age

2. Categorical e.g., Gender, Nationality

3. Ordinal e.g., Low/Medium/High, Education, RankingHow can data pre-processing be performed?

These are a lot of techniques that one can incur to pre-process the data to convert it from raw to ameaningful and understandable format.

1. Conversion of data: As we know Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features, we have a lot of transformation functions that we can use on our categorical/ string data type to convert it to numerical formats, such as 1. One Hot Encoding, 2. get dummies (), map ().

2. Ignoring the missing values: Whenever we encounter any missing values in our dataset it depends on the developer working on the project whether to delete the missing values or to impute some other values such as, mean, median, Predicting the missing value, or many other techniques.

3. Filling the missing values: The process of deleting the missing values can sometimes be efficientonly if the missing values are a handful if the missing values are very large then in this case, we might also check the importance of the variables to our model and we can then delete the entire column with the most number of missing values, or we can impute some other value in place of missing value, by further analysis the dataset or by using statistical measurements, or by using some predicting algorithms to predict the missing values.

4. Outlier's detection: As we have already seen, Outliers can be easily termed as extremely low values or extremely high values which are not related to the particular data variable. These are the error values are present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0, the entire value is changed, so this can be treated as an outlier, but in reality, outliers can be identified either by using plots or by statistical analysis.]

So, our main goal should be to train the best performing machine learning model nearly accurately using the cleaned/pre-processed data, so that it can help us to give the solutions whenever somethingnew data is passed onto it related to the data that we have trained.

Note: The better your data is, the better the results will be.

Supervised Learning:

Supervised learning is a branch of machine learning where for each row in the dataset, each row is tagged with a particular label known as the target class.

Supervised Learning is categorized into 2 other categories which are "Classification" and "Regression".

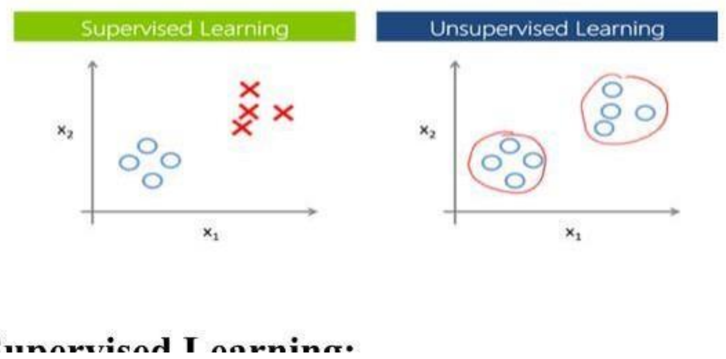


FIG 5.1.1 SUPERVISED AND UNSUPERVISED LEARNING

Classification:

The classification problem is when the target variable is categorical (i.e., the output variable consists of classes such as —Class A or B or something else, there might be 2 classes or more than 2classes.).

A classification problem can be described as a problem where the target class in a dataset consists of
categories, such as "Yes" or "No", or "spam" or "not spam".

Widely used Classification algorithms:

• K-Nearest Neighbour

• Naive Bayes

• Decision Trees/Random Forest

• Support Vector Machine

• Logistic Regression

Regression:

While a Regression problem is when the target variable is continuous (i.e., the output is numeric), Regression problem can be easily termed as the problem where we have to forecast about the future or what we do not know right now, it can be anything (Example: House Price Prediction, Stock market trends)

Widely used Regression Algorithms:

• Linear Regression

• Support Vector Regression

• Decision Trees/Random Forest

• Gaussian Progresses Regression

• Ensemble Methods

Unsupervised Learning:

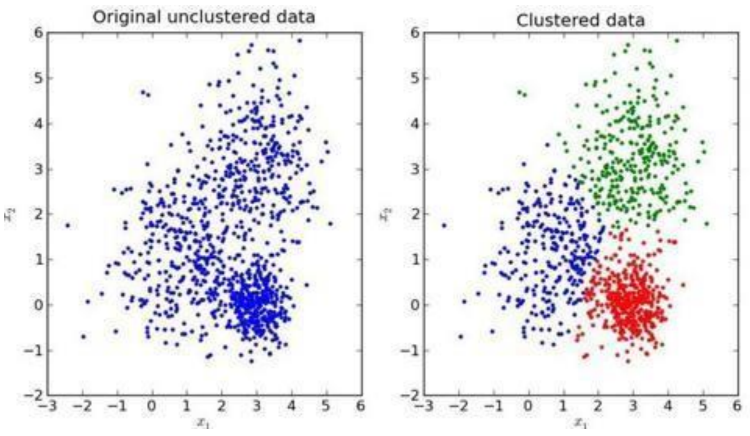


FIG 5.1.2 ORIGINAL UNCLUSTERD AND CLUSTERED DATA

Unsupervised Learning is another branch of Machine Learning where we won't be having any labelsfor each row of our data unlike supervised learning, so in this case, the model will try to segregate things based on the features and the data available. In simple terms it segregates the data in terms of clusters, the most important thing in unsupervised learning is the curse of finding the optimal k value(the number of clusters we would like to make).

Similar to Supervised, Unsupervised learning can also be categorized into 2 other categories as

"Clustering" and "Association".

Clustering:

Clustering is a process of learning to assign labels to examples by leveraging an unlabeled dataset, Because the dataset is completely unlabeled, deciding on whether the learned model is optimal is much more complicated than in supervised learning.

Clustering Algorithms available:

• DBSCAN
• HDBSCAN

• K-Means Clustering

• Hierarchical Clustering

• Gaussian Mixtures

• Spectral Clustering

Overview of models under categories:

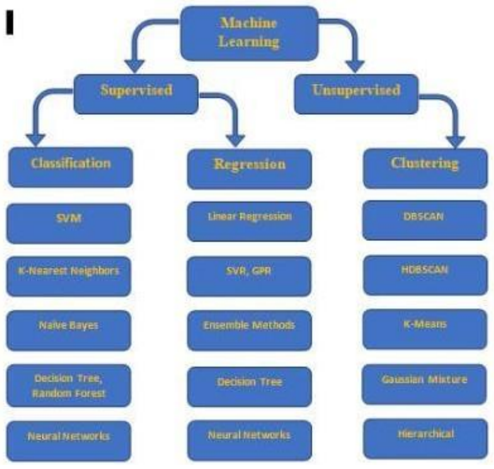


FIG 5.1.3 OVERVIEW OF MODELS

4. Training and Testing the model.

Before building any machine learning Project, training is the most important part, where we train ourmodel using the data available and make the machine learn and understand the data, after which when the model has learned from the data, we provide the model with another dataset to evaluate how good our model is performing, if it is performing well, we then test the model using test data, where we get to know the final performance of our model, which can be measure using various metrics, such as Accuracy, recall, precision, and through classification report.

This whole process of building and deploying a model is done using 3 different datasets which are
split using train_test_split (), which are 'Training data', 'Validation data', and 'Testing data'.

First, we train our classifier/regressor model using 'training data set', we then tune the parameters, tomake the model more efficient ad accurate using 'validation set', and then we test the performanceof our classifier on unseen data/ 'test data set' which should not be or by any means should be used for training or validating, if we use the same data for testing purpose, our model might perform well,but it will lead to overfitting. The test set will only be available during testing the classifier.

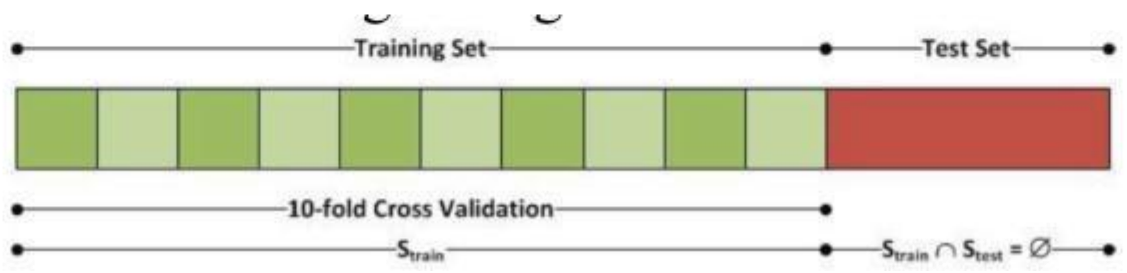


FIG 5.1.4 TRAINING SET

Training set: The training set is like a learning material which we give our model to learn so that it understands the data and applies what is understood to the data which we then use that trained modelto predict the values with the new data. Training a machine learning model can be done using various algorithms, where the algorithms need to be selected carefully according to the problem that we are trying to solve.

Validation set: Cross-validation/ Validation Set is primarily used for estimating how good our model is performing on unseen. Based on the validation data, we can then tune our parameters to make the model more efficient and reliable for deployment.

Test set: Test Dataset is simply an unknown dataset, related to the original data which we split at thestart so that the model is not aware of the values present in the dataset. The Test dataset is used for assessing the final performance of the model and how well the model is performing.

Simple Example of Train, Test, Validation Splits.

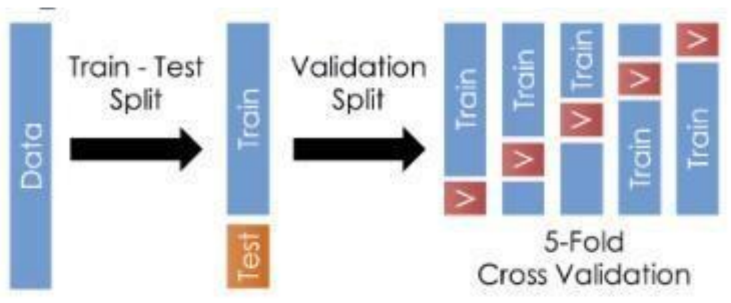
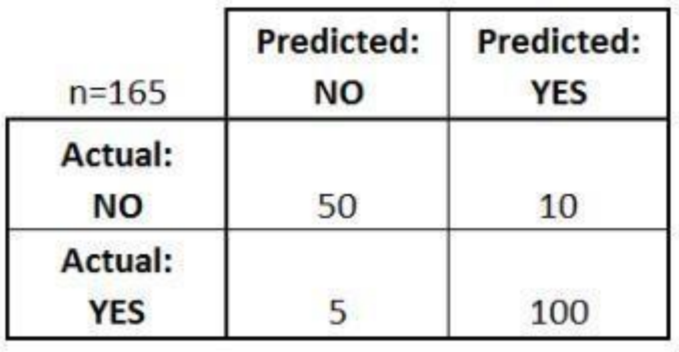


FIG 5.1.5 EXAMPLE OF TRAIN, TEST, VALIDATION SPLITS

Once the model is trained by the required parameters and achieved a good accuracy score, we can then use the trained model for predicting our test data/unseen data. Once this is done, we can evaluate our model or plot the evaluation metrics of our model performance using Classification report, Confusion Matrix, AUC_ROC Curve, etc.

Confusion Matrix

A confusion matrix is simply an evaluation metric matrix especially used for analyzing the behavior of the model and estimating how good the model is performing. It has 4 parameters, which are 'True positives', 'True Negatives', 'False Positives, and 'False Negative'. Which again derived into various formulas such as TPR, FPR, which helps us to get further performance analysis. Below mentioned is an ideal Confusion Matrix, In Confusion matrix the more the TP and Tn the better the model is, although depending on the project we are working on, we might care about FN and FP, when reducing the number of FP and FN, might become an important step



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

during model evaluation.

FIG 5.1.6 CONFUSION MATRIX

• True positives: Both Predicted and Actual value is True.

• True negatives: Both Predicted and Actual Value is False.

• False positives: In this case, the actual value is False, but the model has predicted True.

• False negatives: In this case, the actual value is True, but the model has predicted False.

• Using The TP, TN, FP, FN we can derive some formulas such as:

• Accuracy = (True Positives +True Negatives) / (Total number of classes)

• i.e., for the above example:

• Accuracy = (100 + 50) / 165 = 0.9090 (90.9% accuracy)

• Similarly, we can do it for Recall, Precision etc.

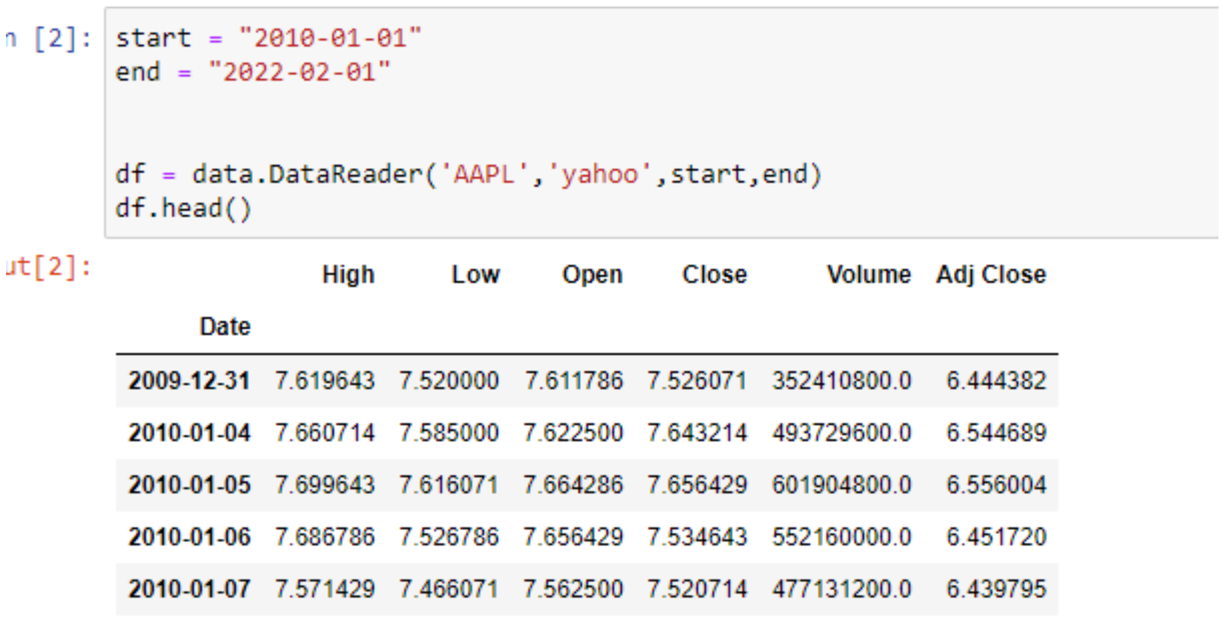
<u>5. Evaluation</u>

Model Evaluation is an integral part of the model development process. It helps us to find the best model that represents our data and it will help us to determine the best parameters which are used to fine-tune the model. This Evaluation step will help us determine the best model out of many models that we create.

Stock trend prediction is a great application of machine learning and when done with good and appropriate dataset can help in unique ways. Here we deployed the application using Streamlit and can select any stock we want and get the dataset from yahoo in real time and use the LSTM model we trained and we get a list of results like the 100 day MA, 200 day MA, and future trend of the stock.

Dataset Description:

We scrape the data from Yahoo website using pandas datareader library. You can mention the start date and end date to get the stock price details of that period.



```
n [2]: start = "2010-01-01"
       end = "2022-02-01"


       df = data.DataReader('AAPL','yahoo',start,end)
       df.head()
```

ut[2]:

Date	High	Low	Open	Close	Volume	Adj Close
2009-12-31	7.619643	7.520000	7.611786	7.526071	352410800.0	6.444382
2010-01-04	7.660714	7.585000	7.622500	7.643214	493729600.0	6.544689
2010-01-05	7.699643	7.616071	7.664286	7.656429	601904800.0	6.556004
2010-01-06	7.686786	7.526786	7.656429	7.534643	552160000.0	6.451720
2010-01-07	7.571429	7.466071	7.562500	7.520714	477131200.0	6.439795

FIG 5.1.7 DATASET DESCRIPTION

Sample images after getting the data.

Algorithm Description:

LSTM:

If you want to predict the next word for the sentence “I like numbers, my favorite subject is …” youranswer would be “mathematics”. How do you come to that conclusion, it is because of the word “numbers”. Recurrent Neural Networks help us achieve this. They are neural networks with loops, allowing the information to stay for some time.

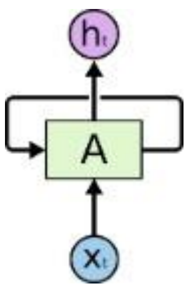


FIG 5.1.8 RECURRENT NEURAL NETWORKS

Xt is the input, the Network analyzes and throws out output ht. RNNs have performed decently but the problem comes when the sentence is too long and has immense data. E.g. predict the next word for the sentence "I like numbers, my favorite subject is mathematics. My brother likes space and technology, his favorite subject is …." your answer would be "astronomy". How do you come to that conclusion, it is because of the word "space and technology".

But how does the network know that word mathematics is not important now and "space and technology is important now". Also the problem with RNNs is it cannot remember the entire sequence for long. Long Short Term Networks (LSTMs) are a special kind of RNN which are very much capable of handling long term dependencies. LSTMs has a cell state which holds the importantword/information required for processing. The information in the cell state can be forgotten (removed) or added based on gates. It has 3 gates to help it decide:

1. Forget gate: To forget the information in the cell state. Done with the help of a sigmoid function which returns a value between 0 and 1. If 1 or closer to 1, remove the word else retain the word.
2. Input gate: What input should be given to the cell state done with the help of tan-h function.
3. Update gate: Any information that can be added to the cell state. E.g. : When "space" was in cell state, update it with the word "technology" because word "technology" is important in decision making as well.

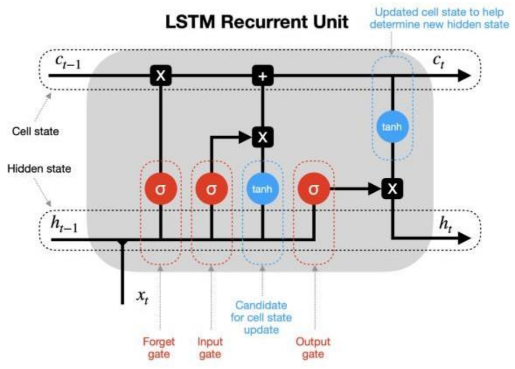


FIG 5.1.9 LSTM RECURRENT UNIT

Long short-term memory network:

Long short-term memory network (LSTM) is a particular form of recurrent neural network (RNN).

Working of LSTM:

LSTM is a special network structure with three “gate” structures. Three gates are placed in an LSTMunit, called input gate, forgetting gate and output gate. While information enters the LSTM’snetwork, it can be selected by rules. Only the information conforms to the algorithm will be left, and the information that does not conform will be forgotten through the forgetting gate. The experimental data in this paper are the actual historical data downloaded from the Internet. Three data sets were used in the experiments. It is needed to find an optimization algorithm that requires less resources and has faster convergence speed.

• Used Long Short-term Memory (LSTM) with embedded layer and the LSTM neural network with automatic encoder.

• LSTM is used instead of RNN to avoid exploding and vanishing gradients.

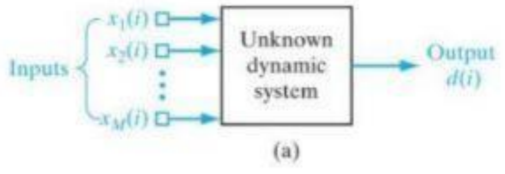
• In this project python is used to train the model, MATLAB is used to reduce dimensions of the input. MySQL is used as a dataset to store and retrieve data.

• The historical stock data table contains the information of opening price, the highestprice, lowest price, closing price, transaction date, volume and so on.

• The accuracy of this LSTM model used in this project is 57%.

LMS filter:

linear problems. The idea of the filter is to minimize a system (finding the filter coefficients) The LMS filter is a kind of adaptive filter that is used for solving by minimizing the least meansquare of the error signal.



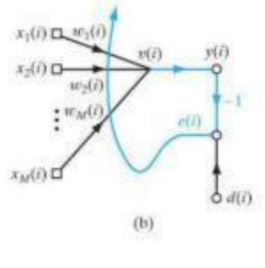


Fig. 1: LMS Inputs and Outputs Fig 2: LMS updating weights

FIG 5.1.10 LMS INPUTS AND OUTPUTSFIG 5.1.11 LMS UPDATING WEIGHTS 1

```
Algorithm 1: LMS
Input:
$x$ : input vector
$d$: desired vector
$\mu$: learning rate
$N$: filter order
Output:
$y$: filter response
$e$: filter error
begin
    $M = size(x)$ ;
    $x_n(0) = w_n(0) = [0\ 0\ \dots\ 0]^T$;
    while $n < M$ do
        $x_{n+1} = [x(n);\ x_n(1:N)]$;
        $y(n) = w_n^H * x_n$;
        $e(n) = d(n) - y(n)$;
        $w_{n+1} = w_n + 2\mu e(n) x_n$;
    end
end
```

In general, we don't know exactly if the problem can be solved very well with linear approach, so we usually test a linear and a non-linear algorithm. Since the internet always shows non-linear approaches, we will use LMS to prove that stock market prediction can be done with linear algorithms with a good precision.

But this filter mimetizes a system, that is, if we apply this filter in our data, we will have the filter coefficients trained, and when we input a new vector, our filter coefficients will output a response that the original system would (in the best case). So we just have to do a tricky modification for using this filter to predict data.

The system:

First, we will delay our input vector by l positions, where l would be the quantity of days we want topredict, this l new positions will be filled by zeros.

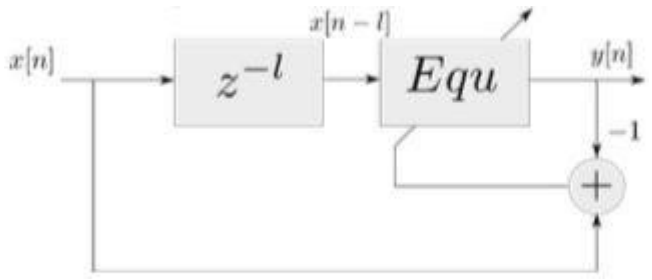
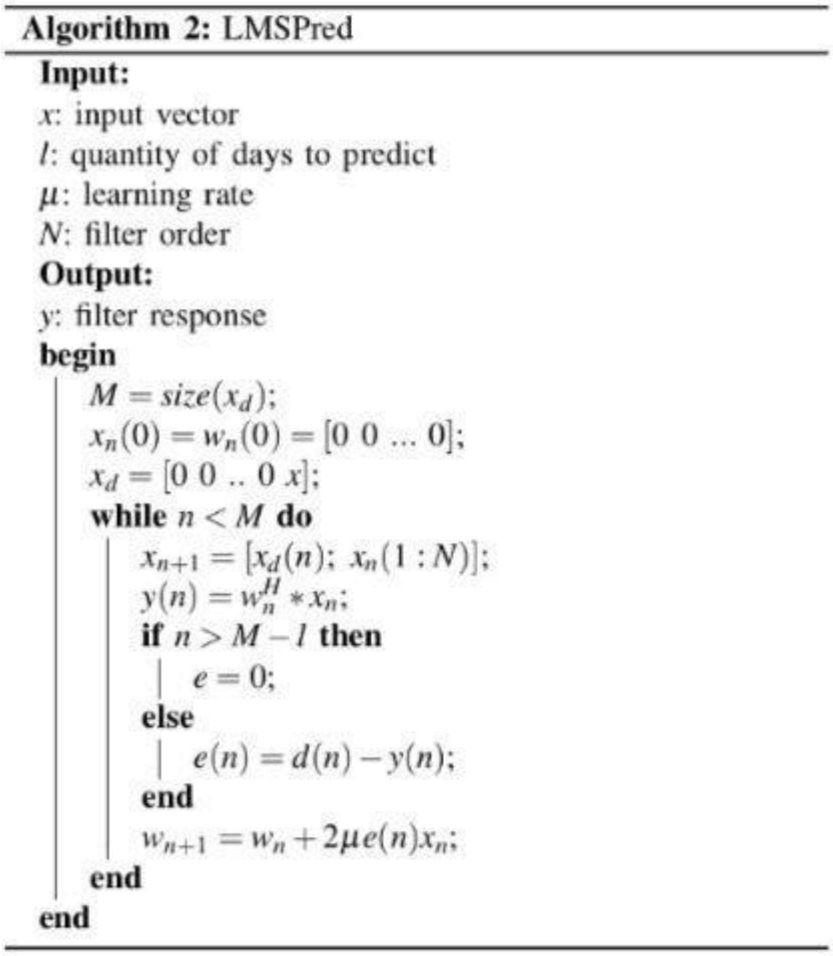


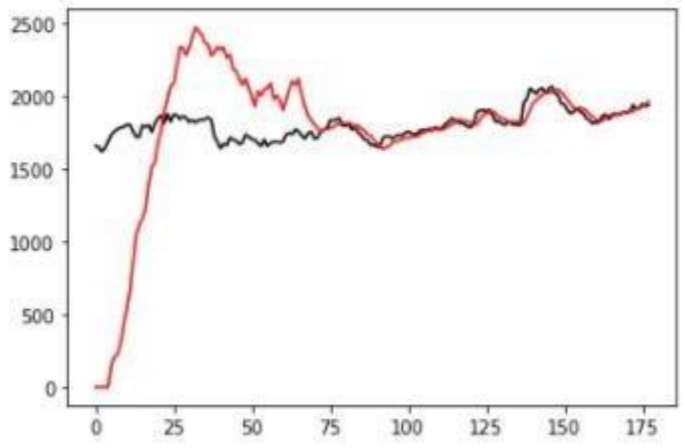
FIG 5.1.12 LMS UPDATING WEIGHTS 2

When we apply the LMS filter, we will train the filter to the first 178 data. After that, we will set the error as zero, so the system will start to output the answers as the original system to the last l values. We will call the tricky modification as the LMSPred algorithm



```
Algorithm 2: LMSPred
Input:
$x$: input vector
$l$: quantity of days to predict
$\mu$: learning rate
$N$: filter order
Output:
$y$: filter response
begin
    $M = size(x_d)$;
    $x_n(0) = w_n(0) = [0\ 0\ \dots\ 0]$;
    $x_d = [0\ 0\ ..\ 0\ x]$;
    while $n < M$ do
        $x_{n+1} = [x_d(n);\ x_n(1:N)]$;
        $y(n) = w_n^H * x_n$;
        if $n > M - l$ then
            $e = 0$;
        else
            $e(n) = d(n) - y(n)$;
        end
        $w_{n+1} = w_n + 2\mu e(n) x_n$;
    end
end
```

Results



One example of stock market prediction result

FIG 5.1.13 EXAMPLE OF STOCK MARKET PREDICTION RESULTS

LSTM Architecture

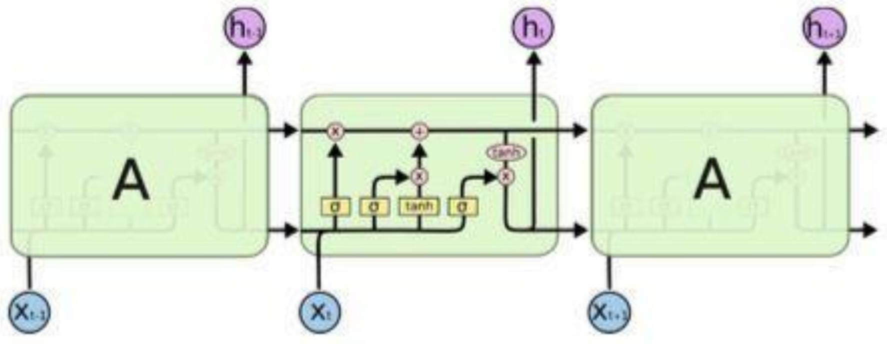


Fig. 4: LSTM Architecture

FIG 5.1.14 LSTM ARCHITECTURE

Forget Gate:

A forget gate is responsible for removing information from the cell state.

• The information that is no longer required for the LSTM to understand things or the information

that is of less importance is removed via multiplication of a filter.

• This is required for optimizing the performance of the LSTM network.

• This gate takes in two inputs; h_t-1 and x_t. h_t-1 is the hidden state from the previous cell or theoutput of the previous cell and x_t is the input at that particular

time step.

Input Gate:

1. Regulating what values need to be added to the cell state by involving a sigmoid function. This isbasically very similar to the forget gate and acts as a filter for all the information from hi-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 andx_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanhfunction) and then adding this useful information to the cell state via addition operation.

Output Gate:

The functioning of an output gate can again be broken down to three steps:

• Creating a vector after applying tanh function to the cell state, thereby scaling the values to therange -1 to +1.

• Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to beoutput from the vector created above. This filter again employs a sigmoid function.

• Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as aoutput and also to the hidden state of the next cell.

```
•  # LSTM
•  Inputs: dataset
•  Outputs: RMSE of the forecasted data
•
•  # Split dataset into 75% training and 25% testing data
•  size = length(dataset) * 0.75
•  train = dataset [0 to size]
•  test = dataset [size to length(dataset)]
•
•  # Procedure to fit the LSTM model
•  Procedure LSTMAlgorithm (train, test, train_size, epochs)
•      X = train
•      y = test
•      model = Sequential ()
•      model.add(LSTM(50), stateful=True)
•      model.compile(optimizer='adam', loss='mse')
•      model.fit(X, y, epochs=epochs, validation_split=0.2)
•      return model
•
•  # Procedure to make predictions
•  Procedure getPredictonsFromModel (model, X)
•      predictions = model.predict(X)
•      return predictions
•
•  epochs = 100
•  neurons = 50
•  predictions = empty
```

15

```
•  # Fit the LSTM model
•  model = LSTMAlgorithm (train, epoch, neurons)
•
•  # Make predictions
•  pred = model.predict(train)
•
•  # Validate the model
•  n = len(dataset)
•
•  error = 0
•  for i in range(n): error += (abs(real[i] - pred[i])/real[i]) * 100
•  accuracy = 100 - error/n
```

5.2 SAMPLE CODE

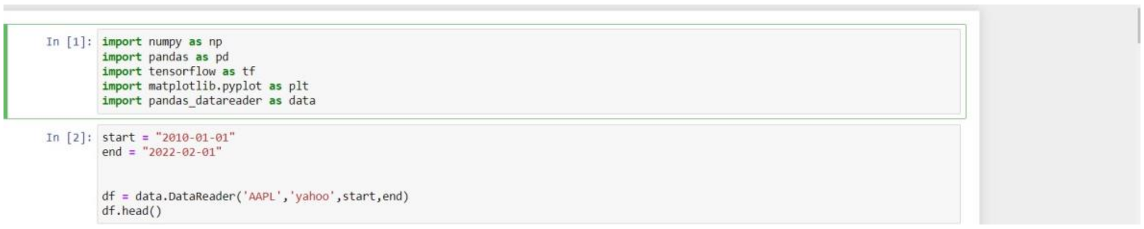
```
import numpy as np

import pandas as pd

import tensorflow as tf

import matplotlib.pyplot as plt

import pandas_datareader as data

from tensorflow.keras.models import load_model

import streamlit as st


start = "2010-01-01"

end = "2022-03-01"


st.title("Stock Trend Prediction")


ui = st.text_input("Enter stock name ","TCS")

df = data.DataReader(ui,'yahoo',start,end)


#Describing data

st.subheader("Data from 2010 - 2021 ")

st.write(df.describe())



#Visualizations

st.subheader("Closing price vs time")

fig = plt.figure(figsize=(10,5))

plt.plot(df.Close)

st.pyplot(fig)
st.subheader("Closing price vs time with 100 ma")
ma100 = df.Close.rolling(100).mean()
fig = plt.figure(figsize=(10,5))
plt.plot(ma100)
plt.plot(df.Close)
st.pyplot(fig)
st.subheader("Closing price vs time with 200 ma")
ma200 = df.Close.rolling(200).mean()
fig = plt.figure(figsize=(10,5))
plt.plot(ma200)
plt.plot(df.Close)
st.pyplot(fig)
st.subheader("Closing price vs time with 100 and 200 ma")
ma100 = df.Close.rolling(100).mean()
ma200 = df.Close.rolling(200).mean()
fig = plt.figure(figsize=(10,5))
plt.plot(ma100,"b")
plt.plot(ma200,"g")
plt.plot(df.Close,"r")
st.pyplot(fig)


#train test splitting
data_train = pd.DataFrame(df['Close'][0:int(len(df)*0.70)]) # first 70% values training
data_test = pd.DataFrame(df['Close'][int(len(df)*0.70):int(len(df))])
```

```
#splitting x_train , y_train


#scale down
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
data_train_array = scaler.fit_transform(data_train)
data_test_array = scaler.fit_transform(data_test)


#load model
model = load_model("stock.h5")
print(model)


#testing
past_100 = data_train.tail(100)
final_df = past_100.append(data_test,ignore_index=True)


#scaling down the data
input_data= scaler.fit_transform(final_df)


#input_data
x_test = []
y_test = []
for i in range(100,input_data.shape[0]):
    x_test.append(input_data[i-100:i])
    y_test.append(input_data[i,0])
x_test , y_test = np.array(x_test),np.array(y_test)


#predictions
```

```
y_pred = model.predict(x_test)

#scale back the values again
scaler = scaler.scale_ #This is the value by which all points in the data were divided to come into
range
scale_factor = 1/scaler[0]
y_pred = y_pred*scale_factor
y_test = y_test*scale_factor

#final graph
st.subheader("Predicted trend vs original")
fig2 = plt.figure(figsize=(12,6))
plt.plot(y_test,"blue",label="Original Price")
plt.plot(y_pred,"red",label="Predicted Price")
plt.xlabel("Time")
plt.ylabel("Price")
plt.legend()
st.pyplot(fig2)
```

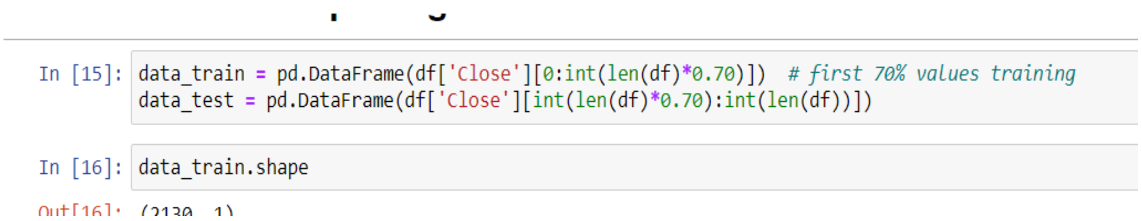
6. INPUT AND OUTPUT SCREENSHOTS

6.1 INPUT SCREENSHOTS:



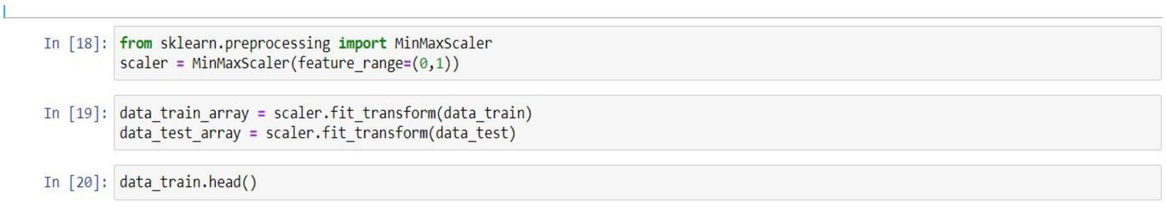
```
In [1]: import numpy as np
        import pandas as pd
        import tensorflow as tf
        import matplotlib.pyplot as plt
        import pandas_datareader as data

In [2]: start = "2010-01-01"
        end = "2022-02-01"


        df = data.DataReader('AAPL','yahoo',start,end)
        df.head()
```

FIG 6.1.1 INPUT SCREENSHOT 1



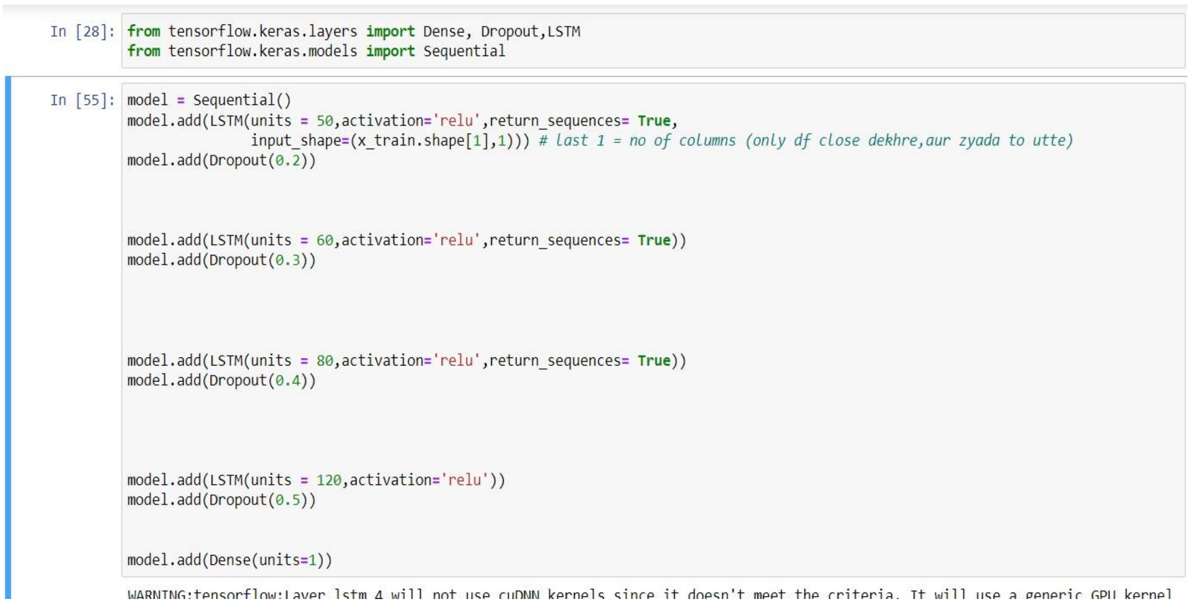
```
In [15]: data_train = pd.DataFrame(df['Close'][0:int(len(df)*0.70)])  # first 70% values training
         data_test = pd.DataFrame(df['Close'][int(len(df)*0.70):int(len(df))])

In [16]: data_train.shape

Out[16]: (2130, 1)
```

FIG 6.1.2 INPUT SCREENSHOT 2



```
In [18]: from sklearn.preprocessing import MinMaxScaler
         scaler = MinMaxScaler(feature_range=(0,1))

In [19]: data_train_array = scaler.fit_transform(data_train)
         data_test_array = scaler.fit_transform(data_test)

In [20]: data_train.head()
```

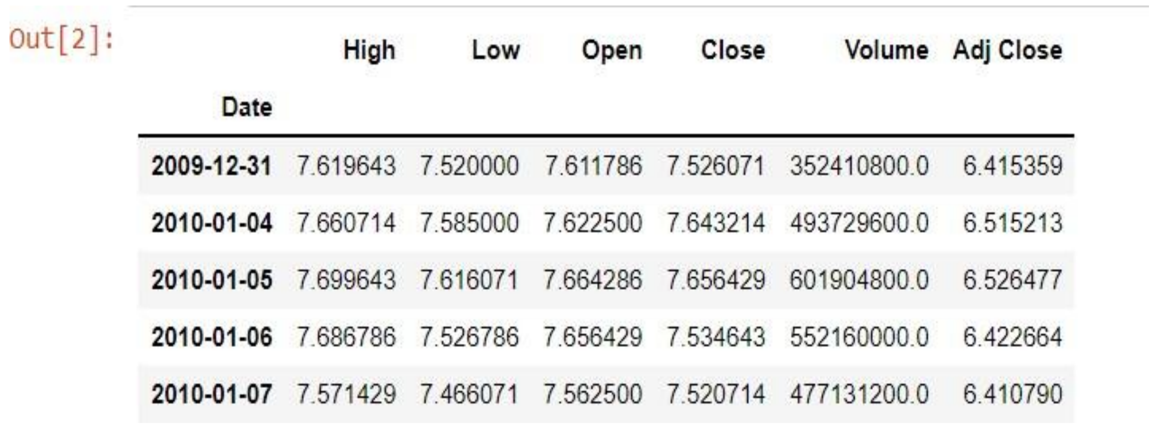
FIG 6.1.3 INPUT SCREENSHOT 3



```
In [28]: from tensorflow.keras.layers import Dense, Dropout,LSTM
         from tensorflow.keras.models import Sequential

In [55]: model = Sequential()
         model.add(LSTM(units = 50,activation='relu',return_sequences= True,
                        input_shape=(x_train.shape[1],1))) # last 1 = no of columns (only df close dekhre,aur zyada to utte)
         model.add(Dropout(0.2))



         model.add(LSTM(units = 60,activation='relu',return_sequences= True))
         model.add(Dropout(0.3))




         model.add(LSTM(units = 80,activation='relu',return_sequences= True))
         model.add(Dropout(0.4))




         model.add(LSTM(units = 120,activation='relu'))
         model.add(Dropout(0.5))


         model.add(Dense(units=1))

         WARNING:tensorflow:Layer lstm_4 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel
```

FIG 6.1.4 INPUT SCREENSHOT 4

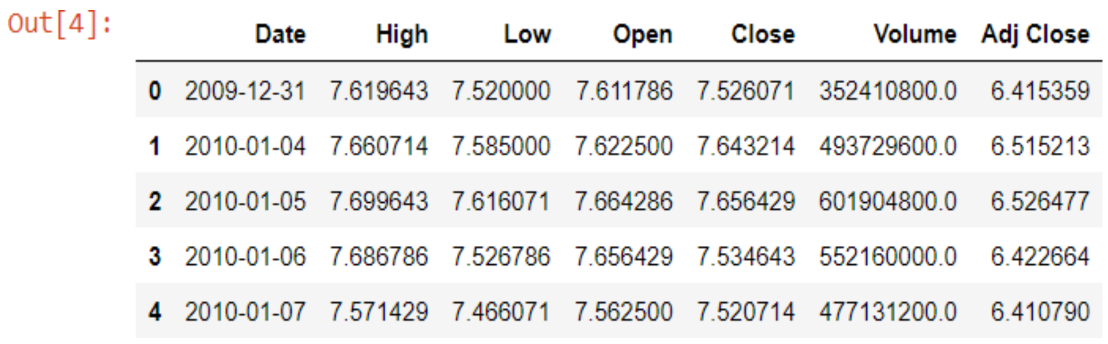
6.2 OUTPUT SCREENSHOTS:



Out[2]:

Date	High	Low	Open	Close	Volume	Adj Close
2009-12-31	7.619643	7.520000	7.611786	7.526071	352410800.0	6.415359
2010-01-04	7.660714	7.585000	7.622500	7.643214	493729600.0	6.515213
2010-01-05	7.699643	7.616071	7.664286	7.656429	601904800.0	6.526477
2010-01-06	7.686786	7.526786	7.656429	7.534643	552160000.0	6.422664
2010-01-07	7.571429	7.466071	7.562500	7.520714	477131200.0	6.410790

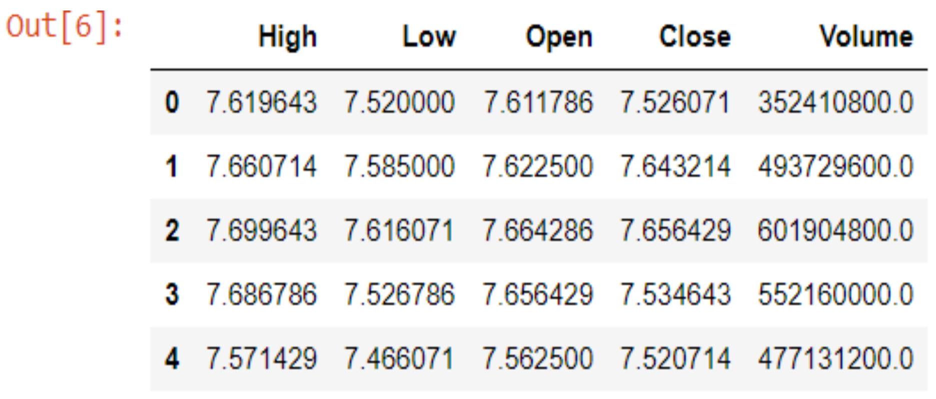
FIG 6.2.1 OUTPUT SCREENSHOT 1



Out[4]:

	Date	High	Low	Open	Close	Volume	Adj Close
0	2009-12-31	7.619643	7.520000	7.611786	7.526071	352410800.0	6.415359
1	2010-01-04	7.660714	7.585000	7.622500	7.643214	493729600.0	6.515213
2	2010-01-05	7.699643	7.616071	7.664286	7.656429	601904800.0	6.526477
3	2010-01-06	7.686786	7.526786	7.656429	7.534643	552160000.0	6.422664
4	2010-01-07	7.571429	7.466071	7.562500	7.520714	477131200.0	6.410790

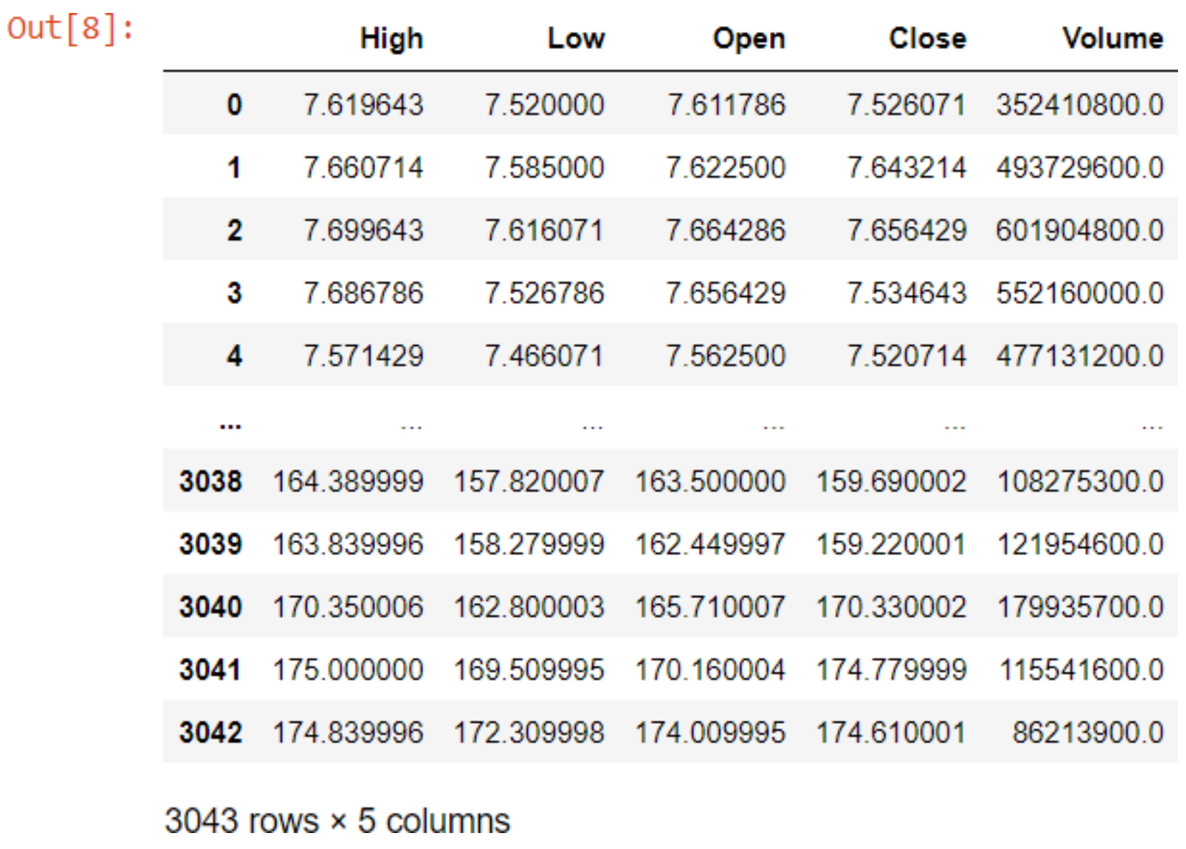
FIG 6.2.2 OUTPUT SCREENSHOT 2



Out[6]:

	High	Low	Open	Close	Volume
0	7.619643	7.520000	7.611786	7.526071	352410800.0
1	7.660714	7.585000	7.622500	7.643214	493729600.0
2	7.699643	7.616071	7.664286	7.656429	601904800.0
3	7.686786	7.526786	7.656429	7.534643	552160000.0
4	7.571429	7.466071	7.562500	7.520714	477131200.0

FIG 6.2.3 OUTPUT SCREENSHOT 3



```
Out[8]:
```

	High	Low	Open	Close	Volume
0	7.619643	7.520000	7.611786	7.526071	352410800.0
1	7.660714	7.585000	7.622500	7.643214	493729600.0
2	7.699643	7.616071	7.664286	7.656429	601904800.0
3	7.686786	7.526786	7.656429	7.534643	552160000.0
4	7.571429	7.466071	7.562500	7.520714	477131200.0
...	...	...	...	...	...
3038	164.389999	157.820007	163.500000	159.690002	108275300.0
3039	163.839996	158.279999	162.449997	159.220001	121954600.0
3040	170.350006	162.800003	165.710007	170.330002	179935700.0
3041	175.000000	169.509995	170.160004	174.779999	115541600.0
3042	174.839996	172.309998	174.009995	174.610001	86213900.0

3043 rows × 5 columns

FIG 6.2.4 OUTPUT SCREENSHOT 4

```
Out[12]: [<matplotlib.lines.Line2D at 0x179de6fe130>]
```

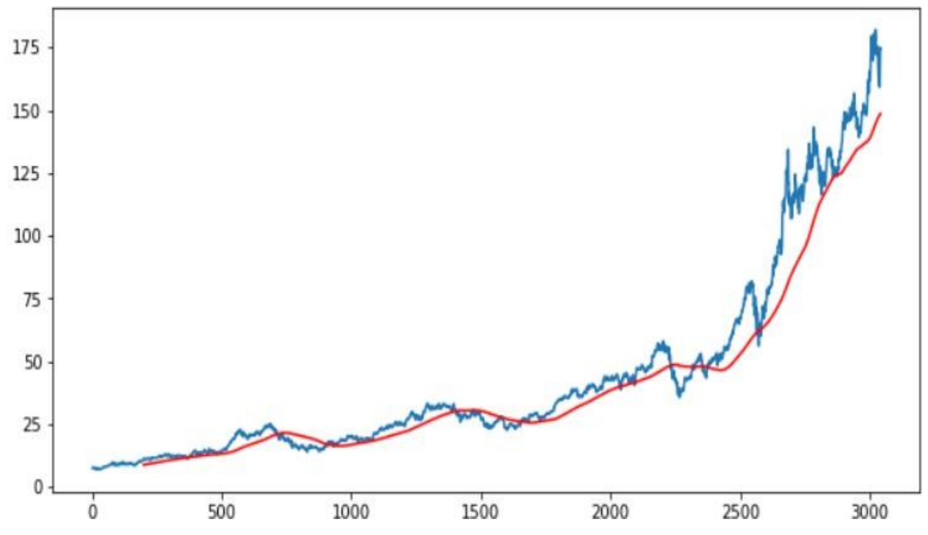
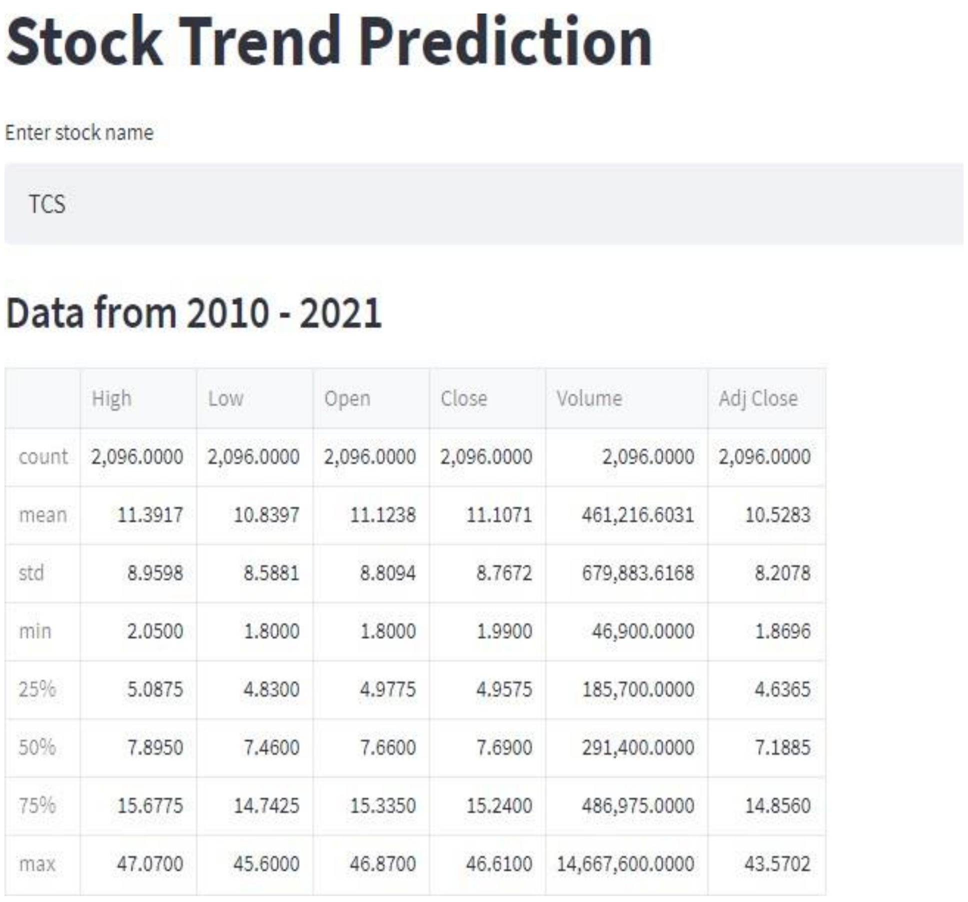


FIG 6.2.5 OUTPUT SCREENSHOT 5

7. RESULTS



Stock Trend Prediction

Enter stock name

TCS

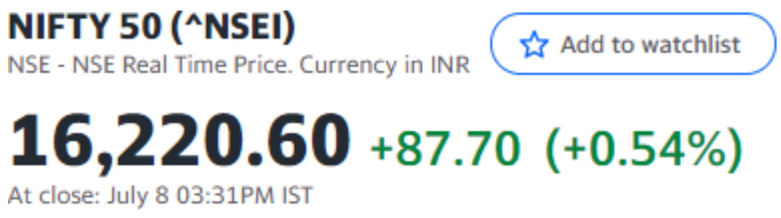
Data from 2010 - 2021

	High	Low	Open	Close	Volume	Adj Close
count	2,096.0000	2,096.0000	2,096.0000	2,096.0000	2,096.0000	2,096.0000
mean	11.3917	10.8397	11.1238	11.1071	461,216.6031	10.5283
std	8.9598	8.5881	8.8094	8.7672	679,883.6168	8.2078
min	2.0500	1.8000	1.8000	1.9900	46,900.0000	1.8696
25%	5.0875	4.8300	4.9775	4.9575	185,700.0000	4.6365
50%	7.8950	7.4600	7.6600	7.6900	291,400.0000	7.1885
75%	15.6775	14.7425	15.3350	15.2400	486,975.0000	14.8560
max	47.0700	45.6000	46.8700	46.6100	14,667,600.0000	43.5702

FIG 7.1.1 STOCK DATA 1

You can enter any stock
name.

Note: Be careful with the stock ticker according to the Yahoo website.



NIFTY 50 (^NSEI)

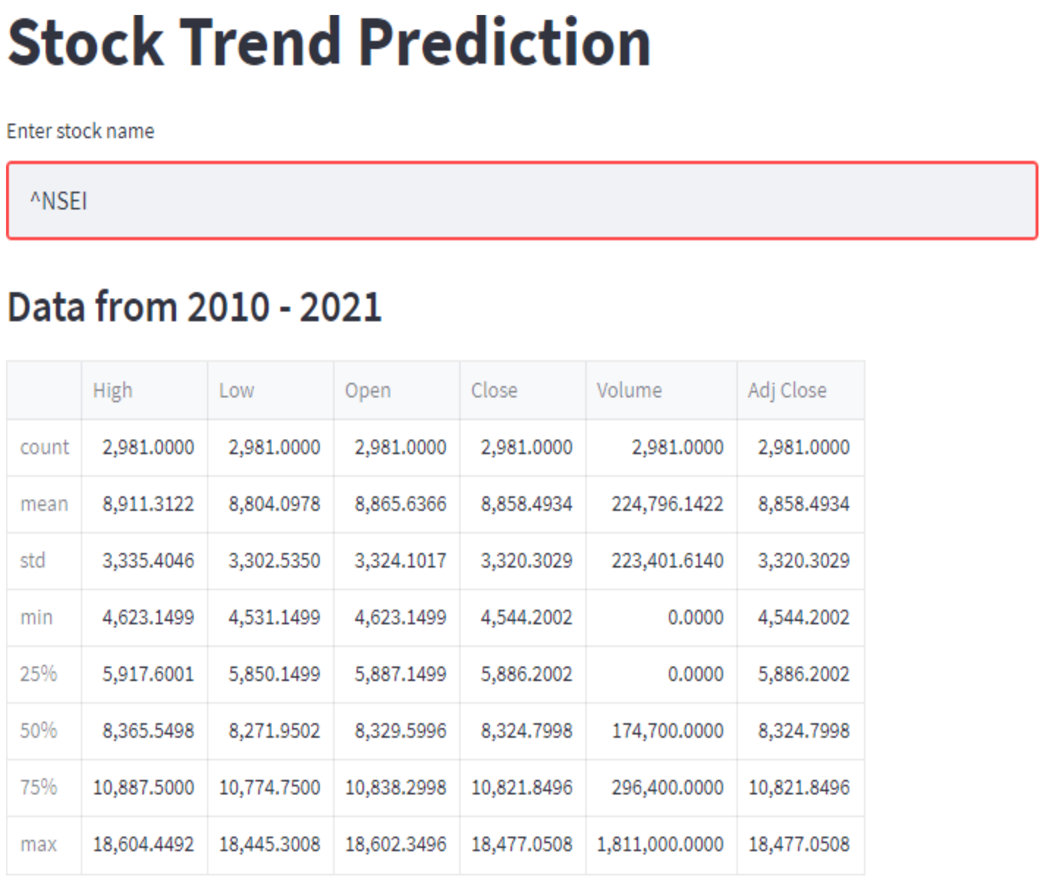
NSE - NSE Real Time Price. Currency in INR

16,220.60 +87.70 (+0.54%)

At close: July 8 03:31PM IST

FIG 7.1.2 STOCK PRICE 1

Ticker for NIFTY 50 (^NSEI), so in the input (^NSEI) .



Stock Trend Prediction

Enter stock name

^NSEI

Data from 2010 - 2021

	High	Low	Open	Close	Volume	Adj Close
count	2,981.0000	2,981.0000	2,981.0000	2,981.0000	2,981.0000	2,981.0000
mean	8,911.3122	8,804.0978	8,865.6366	8,858.4934	224,796.1422	8,858.4934
std	3,335.4046	3,302.5350	3,324.1017	3,320.3029	223,401.6140	3,320.3029
min	4,623.1499	4,531.1499	4,623.1499	4,544.2002	0.0000	4,544.2002
25%	5,917.6001	5,850.1499	5,887.1499	5,886.2002	0.0000	5,886.2002
50%	8,365.5498	8,271.9502	8,329.5996	8,324.7998	174,700.0000	8,324.7998
75%	10,887.5000	10,774.7500	10,838.2998	10,821.8496	296,400.0000	10,821.8496
max	18,604.4492	18,445.3008	18,602.3496	18,477.0508	1,811,000.0000	18,477.0508

FIG 7.1.3 STOCK DATA 2

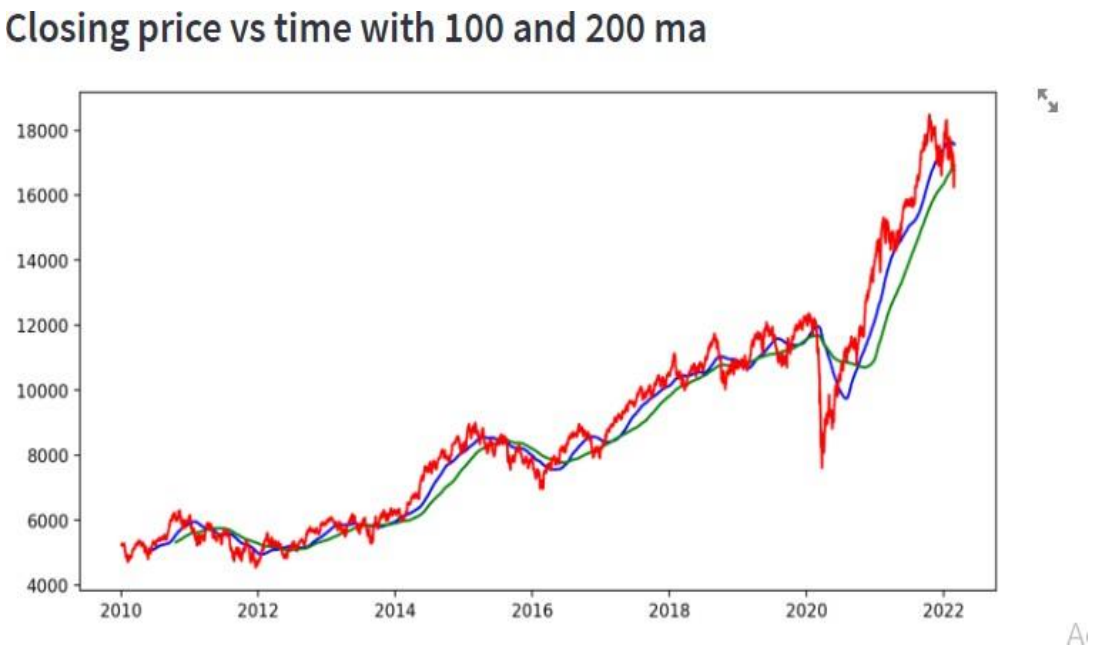


FIG 7.1.4 CLOSING PRICE VS TIME WITH 100 AND 200MA GRAPH

Predicted trend vs original

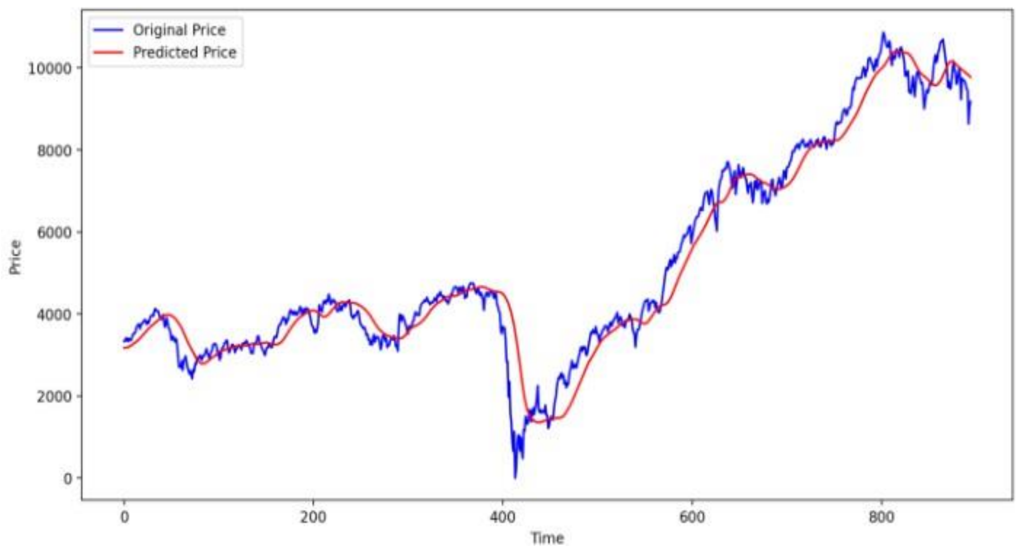


Figure 1: Predicted trend vs. original

FIG 7.1.5 PREDICTED TREND VS ORIGINAL PRICE GRAPH

8. CONCLUSION

In the project, we proposed the use of the data collected from different global financial markets with deep learning algorithms in order to predict the stock index movements. LSTM algorithm works on the large dataset value which is collected from different global financial markets. Also, LSTM does not cause a problem of over-fitting. Various deep learning based models are proposed for predicting the daily trend of Market stocks. Numerical results suggest high efficiency. The practical trading models are built upon our well-trained predictor. The model generates higher profit compared to the selected benchmarks.

9. FUTURE ENHANCEMENTS

Future scope of this project will involve adding more parameters and factors like the financial ratios,multiple instances, etc. The more the parameters are taken into account more will be the accuracy. The algorithms can also be applied for analyzing the contents of public comments and thus determine patterns/relationships between the customer and the corporate employee. The use of traditional algorithms and data mining techniques can also help predict the corporation's performance structure as a whole.

We are predicting the closing stock price of any given organization, we have developed an application for predicting close stock price using LSTM algorithm. We have used datasets belonging to Google, Nifty50, TCS, Infosys and Reliance Stocks and achieved above 93% accuracy for these datasets. In the future, we can extend this application for predicting cryptocurrency trading and also, we can add sentiment analysis for better predictions.

10. REFERENCES

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, GiridharMaji, Narayan C. Debnath, Soumya Sen

[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman - Stock price predictionusing LSTM, RNN and CNN-sliding window model - 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] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, "An innovative neural network approach for stock market prediction", 2018

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.

[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.

[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.

[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning.