

DEVELOPING MACHINE LEARNING BASED MODELS TO PREDICT STOCK PRICES

Project report submitted in fulfillment of the requirement
for the degree of Bachelor of Technology

in

**Computer Science and Engineering/Information
Technology**

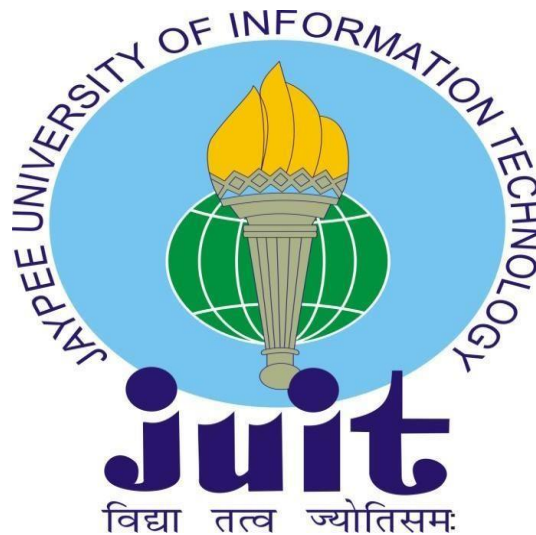
By

Aanjaneya Sharma (191550)
Sulbha Sharma (191529)

Under the supervision of

Mr. Prateek Thakral

to



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat,
Solan-173234, Himachal Pradesh**

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Developing a Machine Learning Based Model to Predict Stock Prices**” in fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Mr. Prateek Thakral**, Assistant Professor (Grade – II) and Computer Science.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)
Aanjaneya Sharma
191550

(Student Signature)
Sulbha Sharma
191529

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Mr. Prateek Thakral
Assistant Professor (Grade – II)
Computer Science
Dated:

PLAGIARISM CERTIFICATE

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: _____ Department: _____ Enrolment No _____

Contact No. _____ E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> All Preliminary Pages Bibliography/Images/Quotes 14 Words String 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for his divine blessing in making it possible to complete the project work successfully.

I am grateful and wish my profound indebtedness to Supervisor **Mr. Prateek Thakral, Assistant Professor (Grade-II)** Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of **Machine Learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Mr. Prateek Thakral** Department of CSE, for his kind help to finish my project.

I would also generously welcome each of those individuals who have helped me straightforwardly or in a roundabout way to make this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

(Student Signature)

Project Group No. :- 127

Aanjaneya Sharma

(191550)

(Student Signature)

Project Group No. :- 127

Sulbha Sharma

(191529)

TABLE OF CONTENT

Title	Page No.
Certificate	I
Plagiarism Certificate	II
Acknowledgment	III
List of Abbreviations	V
List of Figures	VI
List of Graphs	VIII
List of Tables	IX
Abstract	X
Chapter-1 Introduction	1- 8
Chapter-2 Literature Survey	9 - 16
Chapter-3 System Design & Development	17 - 46
Chapter-4 Experiment & Result Analysis	47 - 55
Chapter-5 Conclusions	56 – 58
References	59 - 60
Appendices	61 - 67

LIST OF ABBREVIATIONS

Abbreviations	Meaning
LSTM	Long short term memory
GA	Genetic algorithm
SVM	Support vector machine
HMM	Hidden Markov model
ML	Machine learning
TECH	Support Vector Machine
SMP	Stock market prediction
ANN	Artificial neural network
CNN	Convolution neural network
RNN	Recurrent neural network
PCA	Principal component analysis
LDA	Linear discriminant analysis
LR	Linear Regression
SH	Shareholder
SCP	Stock closing price
SOP	Stock opening price
TM	Traditional methods
NB	Naïve bayes
SDLC	Software development life cycle
AI	Artificial Intelligence
SGD	Stochastic Gradient Descent
INR	Indian national rupee

LIST OF FIGURES

Figure No.	Title	Page No.
1.1	Methodology for SMP	5
3.1	Waterfall model of SDLC	20
3.2	Design of model	20
3.3	Dataset	24
3.4	PCA and LDA	27
3.5	Linear Regression	30
3.6	Example	30
3.7	Typical RNN model	32
3.8	Typical LSTM model	32
3.9	Architecture of LSTM	33
3.10	Gates in LSTM	33
3.11	Forget gate equation	34
3.12	Updated equation	34
3.13	Input gate equation	35
3.14	New information equation	35
3.15	Updated Equation	36
3.16	Output gate equation	36
3.17	Hidden state equation	36
3.18	Final output equation	37
3.19	Dataset used in web application	46
4.4	Working of LSTM	50
4.5	LSTM model summary	50
4.6	LSTM model training – 1	50
4.7	LSTM model training - 2	51

4.9	Interface	52
4.10	Query Parameters	52
4.11	Information about selected Stock Ticker	53
4.12	Select Years of prediction and Raw data	53
4.14	Forecast Data	54

LIST OF GRAPHS

Figure No.	Title	Page No.
4.1	True Value and LR value	48
4.2	Red – Actual Price and Blue – Predicted Price for random 25 values	48
4.3	Actual Price and Orange – Predicted Price for random 25 values	49
4.8	True Value and LSTM value	51
4.13	Plot for stock opening and closing prices	53
4.15	Forecast plot for n future years	54
4.16	Plot for the selected years	54
4.17	Day-wise plot for the selected stock	55
4.18	Day of year plot for the selected stock	55

LIST OF TABLES

Table No.	Title	Page No.
4.1	Various measuring parameters of LR model	49
4.2	Various accuracies of LR model	49
4.3	Various measuring parameters of LSTM model	51

ABSTRACT

The prediction of a stock market direction may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. Forecasting accuracy is the most important factor in selecting any forecasting method. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections that are suitable for investment are a very difficult task. The key factor for each investor is to earn maximum profits on their investments.

In this project, Linear Regression and LSTM are used. Linear Regression is a very specific type of supervised machine learning algorithm characterized by prediction. Linear regression uses two types of variables i.e., dependent, and independent variables. In this project, we investigate the predictability of financial movement with Linear Regression and LSTM. LSTM is the special version or updated version of RNN.

These methods are applied to 22 years of data retrieved from Reliance. The results will be used to analyze the stock prices and their prediction in depth in future research efforts

CHAPTER 1 – INTRODUCTION

1.1 INTRODUCTION

A stock market is a public market for the trading of company stocks. It helps companies to raise capital. It also helps to generate personal wealth. For persons who invest in stock market, predicting the stock market price is always a difficult task. Predicting the stock market prices is not only difficult and interesting but also the tough part in the area of research. As stock price depend on many aspects such as economic, psychological political and social. These all play vital roles in the stock price as they have great influence on it. So predicting the stock price with full accuracy is one of the challenging tasks. All the investors can suffer a huge loss if they lack the sufficient information and knowledge.

In last decade, role of machine learning in various industries has increased too much. Also, it has helped various traders by applying machine learning algorithms in stock price prediction to predict stock prices and these algorithms have produced quite promising results. For maximizing the profits and minimizing the losses, these machine learning algorithms can predict the values of the stock early by analyzing the trend over the last few years, could prove to be highly useful for making stock price movements.

According to different researches, there are two main traditional approaches for SMP, (1) fundament analysis and (2) technical analysis.

1.1.1 Fundament Analysis

It is a method which calculates the real value of a stock and determines the value that one share of that company should cost. For the long-term predictions, Fundamental analysis is useful and the advantages are due to their systematic approach and their ability to predict changes. To determine accurate product value, reliable and accurate information on the financial report of the company,

it is necessary to have competitive strength and economic conditions in which they are interested. The above value of the product can be used to make an investment decision. On the basis of this idea, “if the intrinsic value is higher than the market value it holds, invest otherwise and avoid it as a bad investment”. An assumption is made that, if given sufficient time, the company will move to a cost agreeing with the prediction. If a company is undervalued, then the market value of that company should rise, and conversely, if a company is overvalued, then the market price should fall.

1.1.2 Technical Analysis

It is the study of the stock prices to make a profit, or to make better investment decisions. It predicts the direction of the future price movements of stocks based on their historical data, and helps to analyze financial time series data using technical indicators to forecast stock prices. Meanwhile, it is assumed that the price moves in a trend and has momentum. Technical analysis uses price charts and certain formulae, and studies patterns to predict future stock prices; it is mainly used by short-term investors. The price would be considered high, low or open, or the closing price of the stock, where the time points would be daily, weekly, monthly, or yearly. It is possible to extract rules from the data and the investors make future decisions based on these rules.

Web application is the type of software application which provides an interface for users to interact with the application and perform several tasks. It runs on a web server and is used through a web browser. It can be designed to be either static or dynamic. In static web application, the displayed content remains same to all users and does not change on the user input. In dynamic web application, the displayed content can be changed based on the user input. These applications can be hosted on a web server by a hosting provider.

1.2 PROBLEM STATEMENT

For data analysis time series forecasting and modeling plays a vital role. Time series analysis is one of the ways for analyzing the data recorded over an interval of time. It is being widely used in analytics & data science. Stock prices are volatile in nature and price depends on various factors. The main aim of this project is to predict stock prices using two different machine learning algorithms such as Linear Regression and Long short-term memory (LSTM). These prices will be predicted using the previous prices of the particular stock. It will help out those persons who regularly invest in share market so that they can invest with less risk and can gain more profit. After prediction of the prices the other aim will be to build the web application in Python using streamlit, yfinance, prophet, pandas and datetime libraries.

1.3 OBJECTIVES

The objective of this Python project is to build a stock price predictor which can predict the future prices of a particular stock. This system will tell us about the price of the stock of a particular company. This model can be used by various traders so that they can know the predicted price of the stock and can invest in market with low risk and can gain more profit by knowing the future price of the stock. On the other hand, this project will also be deployed as web application where one can select different stock tickers and can choose the number of years of prediction ranging from one to five which will give the result as future prices till the selected number of years.

1.4 METHODOLOGY

1.4.1 Basic steps in constructing a Machine Learning model:

1.4.1.1 Data Collection

- How accurate the model is going to be is completely depended on the data.

- This step determines that which data we are going to use for the training of our machine learning model.
- We can use different pre-collected datasets by downloading them from Kaggle or another repository.

1.4.1.2 Data preparation

- Preparing the data by wrangling it.
- Cleaning the data which may require various data pre-processing operations.
- Randomizing the data which remove a order or sequence.
- Visualizing data to detect the useful relationships which prevent the bias and variance problem.
- Splitting the data into training and testing data.

1.4.1.3 Choose a Model

- Choosing the best suitable algorithm for the model by analyzing the problems in the other algorithms.

1.4.1.4 Train the Model

- The focus of this step is to train our model using the training data so that it makes the correct predictions.
- With each iteration in this step the model is doing self-learning and rectifying itself to improve the accuracy.

1.4.1.5 Evaluating the Model

- Makes use of a measurement or a combination of metrics to "measure" the model's objective execution.
- In contrast to test data, which doesn't help tune the model, this concealed information is meant to be fairly representative of how the model would

operate in the real world.

- A good train/evaluation split would be 80/20, 70/30, or something similar, depending on the region, the availability of information, the characteristics of the dataset, etc.
- Test the model using hidden data from the beginning.

1.4.1.6 Making Predictions

- Using additional (test set) information that has been withheld from the model up until this stage (and for which class marks are known) is used to test the model; a more accurate prediction of the model's behavior in practice.

1.4.2 Proposed Methodology for SMP

We proposed using a stacking ensemble on selected models that are having high performance to achieve even better all-round performance. The basic architecture of a stacking ensemble is shown in Figure 1.1

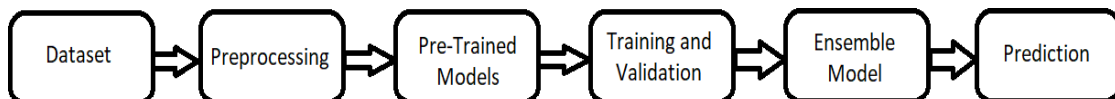


Figure 1.1 Methodology for SMP

Here, we first trained our two shortlisted models: Linear Regression and LSTM. Then we combined these best performing models by creating an ensemble model. The resultant model provided better performance over previous two models.

1.4.3 Basic steps in creating web application using streamlit:

1.4.3.1 Install Streamlit

- Firstly, we have to install streamlit using pip install streamlit in the command prompt or terminal.

1.4.3.2 Create a Python file

- Now we have to create a new python file using any IDE. This file must contain the code for creating the web application.

1.4.3.3 Import Streamlit

- Now we will have to import the streamlit into our python file.

1.4.3.4 Give the layout

- Using the different commands of streamlit we can add titles, text, images and different components which will define the layout for the web application.

1.4.3.5 Add interactivity

- Again, using the different commands, we can create many interactive components like sliders, dropdowns, buttons and etc.

1.4.3.6 Run the application

- Once the whole code is written we can run this file using streamlit run filename.py in the command prompt. This will open the web application in the web browser.

1.5 ORGANIZATION

This project report is divided into five chapters which are as follows :-

Chapter 1: - This chapter gives the brief introduction of the project. The chapter provides the introduction about the project and gave a brief overview of the SMP. The chapter also talks about the problem statement of the whole project and the objectives of the project. The chapter also provides a brief

introduction to methodology used for the project and also provides the information about the steps in designing the SMP model using the machine learning algorithms.

Chapter 2: - This chapter gives the knowledge about the previous work related to the SMP. This also provides the information related to the machine learning. We have mentioned various Journals and related papers which gives information about the work done earlier. The chapter gave us the information that how the various peoples have tried to use the various models in order to SMP model. The techniques and the results for those techniques are mentioned in this chapter, and these help us to find the approach that we are going to use to create our model or project.

Chapter 3: - This chapter gives the information about the steps that we are going to follow to build the whole project. It talks about the system development and the model development. The information about the data set that we are going to use is provided in the chapter. Also, the complete information about the libraries that we are going to use is provided in the chapter. It also gives the information about the machine learning algorithms that we are going to use. It gives the complete knowledge behind the various algorithms. The chapter also includes information about the data preprocessing which include data reduction, data cleaning and data transformation. It also includes information about model creation, model training and the validation of the model. The various accuracy measures are also discussed. It also provides information about the system required to run the project.

Chapter 4: - This chapter gives the information about that how the whole project work is done and at every stage how we have kept checks on the work. It provides information about the work done at different levels and also gives the results obtained at the different levels. It provides the information about the model that we have created using the different modules and libraries. It also consists of the results from the various performance measures that we

have used in the project. It provides the information about the accuracy of the model and the predictions made using the created model. The whole chapter is providing us the information about the performance our whole model or project.

Chapter 5: - This chapter consists of the whole conclusion of the work presented in this project report. It provides information about the whole phases of the project and it also mentions the future scope for the project. It also consists the information about the applications of the project and where the project can be utilized in order to make that sector more computerized. It gives the information that how we can improve the project and what we can do in future related to this project and how we can improve this project.

CHAPTER 2 – LITERATURE SURVEY

SMP model has been created by many people using different approaches and different machine learning algorithms. So all the major techniques used for SMP model presented till date in any paper or research work are as follows :-

- **Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar [1]**, the project stock price prediction is self-explanatory as in this project we just predict the prices of stock for the future, based on their past and present prices. The stock market is so dynamic in nature as the price of a particular share or stock varies non-linearly. Also, there are so many factors on which the stock market depends on, factors including supply and demand of the subject of company, global economy, political conditions, and world wars too.

The main aim is to build a model which can predict future stock prices for a particular stock which will help to increase the profit of a stockholder. Basically, we want to build an accurate model which will predict the share prices for a particular company based on the present and past prices of a particular share.

In the past times, many classical algorithms were used like linear regression, Random Walk Theory, Moving Average Convergence or Divergence, etc. Some neural networks were also used like ANN, CNN, and RNN. But for this prediction, the motive would be the same for all models i.e. to predict the prices for the future stock so that they would help to gain more profit and less loss.

In this paper, the dataset for the five companies has been collected and this dataset includes 10-year data. Attributes for the same are High, Low, Open, Close, Adjacent Close, and Volume. Then they make some other variables titled New Variables and this New Variable includes the difference between

high and low prices and the average of 7-day prices. Firstly, they use ANN i.e. Artificial Neural Network. In this neural network, there were 7 inputs, and only 1 hidden layer was taken into consideration. After the calculations between the input layer and the hidden layer, the whole data was sent to the output layer as input. After the ANN the next model they used was Random Forest. Random forest is one of the ensemble techniques. Under this model, new variables were created, and then with the use of these new variables training of each decision tree was performed.

In the section on results and conclusion, they calculated RMSE, MAPE, and MBE. After this, there were output graphs between the closing price and the date for each company.

- **Subba Rao Polamuri, Kudipudi Srinivas, A Krishna Mohan [2]**, stock market prediction is a challenging task nowadays as all of us want to gain more profit and loss and this is possible if we have a model which can predict the future price of a particular stock based on its past and present prices. The person who invests in the share market always wants to know the future prices of the share he/she has or the shares he/she want to buy or sell to get more and more profit. They want to make a model which helps to make all shareholders the best in their own way.

Under the literature survey of this paper, the major indicators were the fundamental and technical analysis. For the prediction techniques, they chose almost 6 techniques and for those 6 techniques, their advantages and disadvantages were discussed. Those 6 prediction techniques were Holt-Winters, Artificial Neural Network, Hidden Markov Model, ARIMA Model, Time Series Linear Model, and Recurrent Neural Network. And finally, they prepared a table of comparison of all these 6 prediction techniques.

Talking about ANN, the Stock closing price is the parameter that was used and its advantage is that this has a lower prediction error and the disadvantage is that prediction gets worse with increased noise variation. Now coming

towards SVM for stock prediction the parameter used here is consumer investment, net revenue, net income, price per earnings, and the ratio of a stock, and its advantage is that this does not lose much accuracy when applied to a sample from outside the training sample and disadvantage is that this model exaggerates to minor fluctuations in the training data which decrease the predictive ability. The next model is HMM i.e. Hidden Markov model and its used parameter is a technical indicator. The major advantage of this model is that this is used for optimization purposes and its disadvantage includes the evaluation of decoding learning.

Coming to the ARIMA model the parameters used under this are open, high, low, close prices, and moving average. This model is robust and efficient but only suitable for short-term predictions. Then Time Series linear model is a model which uses the parameters data and number of months. Its disadvantages include integrating the actual data and disadvantages of traditional and seasonal trends present in the data. The last model was RNN which used the input hidden and output layers as a parameter used and its advantage includes that this may contain the inputs which were previously used in the past but this only works for the smaller set of input nodes.

In the conclusion, this paper is basically a comparative paper that differentiates 6 different techniques which also includes their advantages and disadvantages too.

- **Troy J. Strader, John J. Rozycki, Thomas H. Root, Yu-Hsiang Huang [3]**, as we understand now that the strategy for the stock or share market is not easy this is much more complex than we thought. So, in order to solve this complex mystery in the past twenty years many models we developed. But as simple each model must have its advantages but along with this, they must have disadvantages too. But again, the motive for all of us is to provide sufficient knowledge to the shareholders and stockholders so that they may earn or gain more and more profit and less loss.

In this paper, the authors discussed the research taxonomy, artificial neural network, Support vector machine, genetic algorithm, and investment decision. For the research taxonomy, a chain model was built which includes the way how they proceed in their model.

Here the main point which was discussed was the genetic algorithm. Basically, genetic algorithms are the algorithms in which the basic optimization technique is search based. In simple words, GA is search based technique. This is used when we are searching for an optimal solution. Also, they are known as genetic because the main topic of where they came from is genes. As if we are having two algorithms and, in some cases, they both are given some optimal solution for the same problem both are also having some disadvantages hence in these cases we might mix up these algorithms with each other, and then the new algorithm must have all the good properties from the parent algorithm and hence that third algorithm will be the best.

The same, if genetic algorithms are having so many optimal solutions i.e., they are having so many advantages but similar they also have some disadvantages too. The biggest disadvantage of genetic algorithms is that they are highly computationally expensive. They take a lot of time to run but still the most time taken is the point where we decide the two or more parent algorithms which we are going to mutate with each other in order to have a best algorithm out of them.

- **Nusrat Rouf, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satayabrata Aich, and Hee-Cheoi Kim [4]**, the project stock price prediction is self-explanatory as in this project we just predict the prices of stock for the future, based on their past and present prices. The stock market is so dynamic in nature as the price of a particular share or stock varies non-linearly. Also, there are so many factors on which the stock market depends on, factors including supply and demand of the subject of company, global economy, political conditions, and world wars too. The

main aim is to build a model which can predict future stock prices for a particular stock which will help to increase the profit of a stockholder. Basically, we want to build an accurate model which will predict the share prices for a particular company based on the present and past prices of a particular share.

Here the complete model was divided into various steps. The machine learning models which were used in this paper were ANN, SVM, Naïve Bayes, Genetic Algorithm, Fuzzy Algorithm, Deep Neural Network, Regression Algorithm, and Hybrid approaches. Also, there were two main traditional approaches to the analysis of the stock markets. One is Fundamental Analysis and the other is Technical Analysis.

Under the section of Fundamental Analysis, there are all the basic calculations. Here are the basic and genuine prices of a share that company holds. All the past data also comes under this section and the calculations might be used to predict the future price of a particular stock based on the previous price. The Technical Analysis is the complete study of the procedure of a stock to make a profit or how to make investment decisions better. This is also an important step as we can perform calculations but how we can or what calculations we have to perform must be decided in this step only.

- **V Kranthi Sai Reddy [5]**, nowadays investors and traders invest a huge amount of money in stocks and shares. So every model which is used to predict the future price of a particular stock must be accurate because a lot of money is involved in this case. Hence In recent years, the usage of machine learning increased at a different level. Even most of the important works depend on artificial intelligence and machine learning

The methodology used in this is, SVM i.e. Support vector Machines, and the RBF i.e., Radial Basis Function. SVM is the machine or the model which is considered most appropriate for the time series predictions. Basically, time

series predictions are predictions that can find patterns in the data and are used to predict or forecast the values of any given variables. Also, the SVM involves plotting the points from the dataset in the space of n dimensions. RBF i.e., the Radial Basis Function, is basically a kernel that is used for kernelization in the algorithm. Mainly this is used in SVM. Kernelization is a technique to increase the dimensions of a dataset and in this technique; the main inputs can be replaced by similar inputs.

Here the environments which are used are WEKA and YALE. These both are Data Mining environments. After this, they mentioned a list of features that are used in SVM. The dataset used in this paper is the IBMInc.csv file

Coming toward the conclusion that SVM works better on the large dataset also, SVM does not give a problem of over fitting. The practical trading models built upon our well-trained predictor.

- **Jingyi Shen and M. Omir Shafiq [6]**, we are surrounded by big data so also the data for stock price or market price prediction increases. Today stock market is a new and large world and this is the field that is completely dedicated to investors. Here they collected data from a Chinese stock market and as this data is large enough so they applied some deep learning models in this. Also, some optimization techniques are also used in this. Techniques including PCA i.e., Principal Component Analysis, CNN i.e., Convolutional neural network, LSTM i.e., Long short-term memory.

According to the paper, for them there were 3 key contributions of the work, the first one was to extract the dataset and sanitization the dataset, the second includes feature extraction, and the third step was to run a model and that model is LSTM.

They chose a dataset of 3558 stocks from the Chinese stock market. Attributes of the dataset were daily price data and fundamental data of each stock Id. And for this data the algorithm they proposed was like first they

take raw data then sanitize it then there will be feature extraction and after that features will be finalized on which the model will work. They will also reduce the dimension of the dataset with the help of PCs technique. After that the final step will be taken into consideration i.e., their model LSTM will be proposed finally.

For the conclusion section, PCA and LSTM were chosen. With the help of PCA, the dimensions of raw data were decreased and then LSTM was proposed for the managed data.

- **Indronil Bhattacharjee and Pryonti Bhattacharja [7]**, the stock market is something that genuinely helps in a country's economic growth. But as this is very much important for a person as well as for the country also but to predict future stock prices is not easy. There are many reasons for the same but from them, one is like the prices or the trend is not stationary in nature. Many algorithms are used to predict the stock prices like KNN, Random Forest, Linear Regression, Lasso, and Ridge.

Here firstly they discussed Statistical methods, these methods include SMA i.e., Simple moving average, WMA i.e., Weighted Moving average, Exponential Smoothing, and Native Approach. Their respective formulas were also described in the paper.

After statistical methods, some machine learning methods were discussed and methods include Simple Linear Regression, Ridge Regression, Lasso Regression, KNN, Random Forest, SVM, and SLP i.e., Single Layer Perceptron. For the proposed system firstly the data set is taken into consideration and then some data-cleaning processes are done after this the whole dataset will be divided into testing and training datasets. Then scaling of the datasets must be done and then the final feature selection will be done but after this, the prediction will be done using both traditional and statistical machine learning approaches.

The conclusion of this paper is a comparative study between statistical and machine learning approaches. After studying it has been concluded that machine learning methods especially LSTM and MLP are found to be more precise, and accurate to predict stock prices. Also, the computation costs for the traditional methods are very high in comparison to machine learning approaches.

- **Xuan Ji, Jiachen Wang, Zhijun Yan [8]**, stock market prediction is a challenging task nowadays as all of us want to gain more profit and loss and this is possible if we have a model which can predict the future price of a particular stock based on its past and present prices. The person who invests in the share market always wants to know the future prices of the share he/she has or the shares he/she want to buy or sell to get more and more profit. They want to make a model which helps to make all shareholders the best in their own way.

In this paper, by declining all the traditional approaches they just used Deep Learning techniques to predict the stock prices. DNNs are comparatively faster than traditional methods and have very fewer computation costs. Similarly, to the other models in this firstly they collected the dataset and then sanitized it. After this, the feature selection step will be taken into consideration. In this step, text features are extracted from social media which can represent investors' attitudes in front of the world. The proposed methods are Doc-W-LSTM and this uses the DOC2Vec model to extract high-dimensional text feature vectors. Doc2Vec model generates a feature with a high or large dimension which will better represent the semantic information to the original document. Now the dataset becomes of large dimension and hence that must have so many outliers and noise. Now the noise from the dataset will be removed.

Now finally LSTM will be proposed and this is the improved version of RNN.

CHAPTER 3 – SYSTEM DESIGN & DEVELOPMENT

3.1 ANALYSIS

Firstly, we have to analyze about the types of trading. It is being categorized into many types which allow us to invest in stock market in different ways such as: -

3.1.1 TYPES OF TRADING

3.1.1.1 Intraday Trading

In this type of trading, traders have to complete the entire transaction in a day. In this a person has to buy and sell a stock within a day and it makes investors use of margin.

3.1.1.2 Position Trading

In this type of trading, traders have more time than the intraday trading. In this one can hold stocks for months and by understanding price he/she can sell it or buy it.

3.1.1.3 Swing Trading

It allows one to hold stock more than one day and he/she can benefit from the swing in prices of stocks.

3.1.1.4 Delivery Trading

In this type of trading, stocks are being delivered to a respective demat account. It does not allow the usage of margins like intraday allows. Also, it is the form of long – term investment.

3.1.1.5 Technical Trading

This type of trading is done with the help of technical market analysis. In this type of trading decisions are being made by analyzing the stock price changes. This trading has more risk than other types of trading.

So above types of trading make us understand that in how many types trader can invest and in which respect SMP model should be built according to need of the organization or individual.

Now we have to analyze about the type of dataset. We can develop SMP model using different machine learning algorithms on different type of datasets. SMP model can be categorized according to the type of data they use as the input. Before developing SMP model it's important to know about the type of datasets. So, let's look about the different types of data available on social media platforms and market.

3.1.2 TYPES OF DATA

Most of the models which have been developed used the market data for their analysis. Using market data they have developed SMP model which predict the future price while recent studies have considered textual data from online sources as well.

3.1.2.1 Market Data

Market data are the information which is related to historical price – related numerical data of financial markets. Analysts and traders use market data to analyze the historical trend and the latest stock prices in the market. Market data include the real time prices of the stock at the particular interval of time which will definitely help in predicting stock prices. These types of data are usually free and can be downloaded from the market websites such as Yahoo Finance and etc.

3.1.2.2 Textual Data

It is the type of the data which includes data in the text form like financial news websites, general news, and social platforms. It's a difficult task to convert the textual data into numerical data so that it can be used for SMP. This type of data uses the different sentiments to analyze the stock market data. As the recent studies has revealed that public sentiments also affect the market considerably. But one of the challenges is that using this type of data model complexity increases as collecting a lot of textual data is a complex task.

3.1.3 TYPES OF APPROACHES

Now moving towards different approaches to build SMP, as we have discussed earlier that there are two traditional approaches namely: -

- **Fundamental Analysis**
- **Technical Analysis**

These both approaches have been widely used to build SMP. About both approaches we have discussed above.

3.2 SYSTEM DEVELOPMENT

In order to develop a software or model we need to learn the concept of the Software Development Life Cycle (SDLC). In our project we are using the Waterfall model of the Software Development Life Cycle (SDLC). The user can develop a good and efficient software by using this model of Software Development Life Cycle (SDLC). The main reason that it provides us efficient and good software is that it uses the sequential way for the creation of the software. This method keeps check on the whole system design process as it doesn't allow a developer to move next step without completing the previous step. This structure gives us advantage that we do not have the overlapped processes. The output of the one step is used as an input for the next step and without completing the previous step we cannot move to the next stop. The

Waterfall model design is given below.

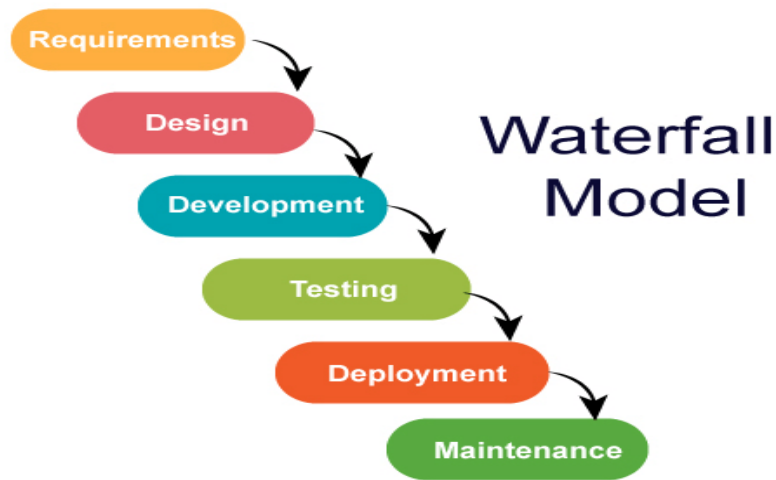


Fig. 3.1 Waterfall Model of SLDC [9]

3.3 DESIGN

For implementation of any machine learning model its flow chart or design is most important. So, for SMP model building we must have its pre – defined structure, basis of which model will be built. That structure gives an idea to build model easily and every existing model is built on those pre – defined steps only.

As every machine learning model have one pre – defined blueprint as shown in below figure which allows us to build our model easily using the steps mentioned in figure.

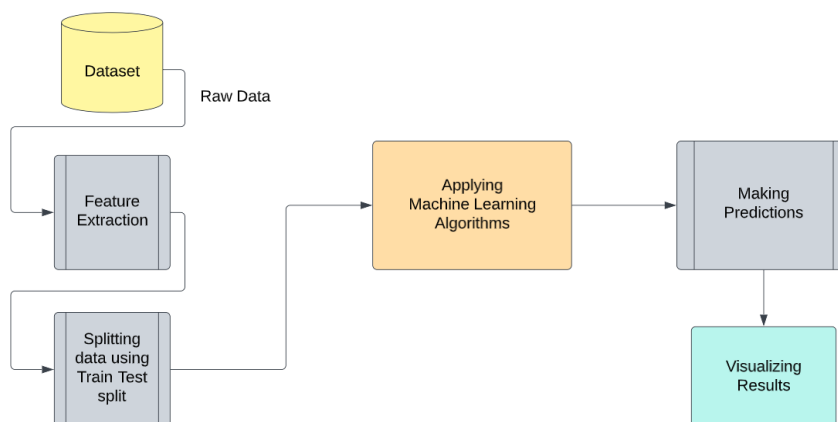


Fig. 3.2 Design of Model

Above design depicts that how we will build our SMP model. Following steps will be followed for implementation of SMP model :-

Step 1 :- Take dataset and upload it on any Python IDE

Step 2 :- Data preprocessing is the next step in which we basically remove outliers and make our dataset ready for further use.

Step 3 :- After exploratory data analysis, we do feature extraction where we extract important features which may contribute highly to our results.

Step 4 :- Now we split our dataset into training and testing part using train test split library. Generally, the train test split ratio is 70:30 or 80:20 (in percentage).

Step 5 :- After above steps we fit our data in to different machine learning algorithms for the required predictions.

Step 6 :- Once predictions are done we visualize our results and move towards the desired result.

3.4 MODEL DEVELOPMENT

In previous section we saw some basic steps which are required to develop a model. Now we will discuss them briefly to know about them deeply and know that how our model will be deployed.

3.4.1 IMPORTING LIBRARIES

3.4.1.1 Numpy

A Python library used for working with clusters is called NumPy. Additionally, it may operate in the areas of grids, Fourier transformation, and direct polynomial mathematics. A library called NumPy, which stands for Mathematical Python, contains a variety of schedules for managing those clusters in addition to multifaceted display objects. NumPy may be used to execute numerical and intelligent operations on exhibitions. This educational activity clarifies NumPy's fundamentals, including its

engineering and atmosphere. Additionally, it looks at various cluster capacities, ordering styles, etc. You are free to use it without restriction because it is an open-source task.

For Mathematical Python, use NumPy. NumPy aims to provide a display object that is ultimately 50x faster than typical Python records. The Nd array exhibit object in NumPy has a wealth of supporting features that make working with Nd array incredibly straightforward. In the field of information science, where efficiency and resources are crucial, exhibits are frequently used.

3.4.1.2 Pandas

Pandas is characterized as an open-source library that gives elite execution information control in Python. The name of Pandas is gotten from the word Board Information, and that implies an Econometrics from Multi-layered information. It is utilized for information examination in Python and created by Wes McKinney in 2008.

Information examination requires heaps of handling, for example, rebuilding, cleaning or combining, and so on. There are various instruments accessible for quick information handling, like Numpy, Scipy, Cython, and Panda. Yet, we lean toward Pandas since working with Pandas is quick, basic and more expressive than different devices.

Pandas is basically utilized for reshaping and turning of the information collections. It incorporates with different libraries like SciPy and scikit – learn.

3.4.1.3 Matplotlib

Python scripts can be used to create 2D charts and plots using the Matplotlib module. It provides a module called Pyplot that streamlines the plotting process by providing controls for line styles, text style properties, designing tomahawks, and other features.

To be more precise, it supports a very broad range of diagrams and plots, including histograms, bar graphs, power spectra, error outlines, and others. It is used in conjunction with NumPy to provide a powerful open-source alternative to MATLAB. Additionally, it can be used with Python and PyQt's wxPython illustration toolkits.

3.4.1.4 Sci-kit

Scikit-learn library, commonly known as sklearn library. This library is one of the most useful libraries. As this library contains most important tools for machine learning. The basics of the scikit-learn library in python have supervised learning algorithms, Unsupervised Learning, Clustering, and dimensionality reduction.

This library allows us to give many important features like normalization, different algorithms, different metrics for performance measure of model, and many more important features.

3.4.1.5 Keras

A powerful and easy-to-use free open-source Python library for building and analyzing machine learning models is called Keras. It covers Theano and TensorFlow, two efficient frameworks for mathematical computation, and enables you to characterize and prepare neural network models in just a few lines of code. It uses independent AI toolboxes, C#, Python, and C++ libraries. Theano and TensorFlow are tremendously powerful libraries for building brain organizations, but they are also challenging to understand.

Keras is based on a minimal architecture that offers a clear and straightforward technique for creating deep learning models in light of image processing libraries. The goal of Keras is to quickly characterize profound learning models. All things considered, Keras is the best option for situations requiring deep learning.

3.4.2 UPLOADING DATASET

We have taken a dataset from the yahoo finance website which gives us real-time prices of a particular stock. From this website, we have taken RELIANCE stock prices as our dataset which includes its real-time prices for the last 22 years. Dataset has 7 attributes which include Date, Open, High, Low, Close, Adj Close, and Volume. The dates are from 17-11-2000 to 17-11-2022. The total numbers of entries are approx 6000.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	17-11-2000	48.1198	48.1895	47.5783	48.0966	37.4534	2.2E+07
3	20-11-2000	48.0502	48.3674	47.8181	47.9961	37.3751	1.1E+07
4	21-11-2000	48.3906	48.3906	47.5938	47.8259	37.2426	2.4E+07
5	22-11-2000	47.9651	48.1972	47.5938	47.6789	37.1281	7411113
6	23-11-2000	47.5009	48.3519	47.2302	48.143	37.4895	1.6E+07
7	24-11-2000	48.3519	49.0327	48.2746	48.4834	37.7546	2.2E+07
8	27-11-2000	48.584	49.4969	48.5066	49.3112	38.3992	2E+07
9	28-11-2000	49.5046	51.0983	49.0869	50.9823	39.7005	4.4E+07
10	29-11-2000	50.7502	51.431	50.1313	50.6264	39.4234	3.4E+07
11	30-11-2000	50.5954	51.0983	50.3634	50.7734	39.5378	2E+07
12	01-12-2000	50.5954	51.0983	50.2086	50.7656	39.5318	2.3E+07
13	04-12-2000	50.8894	51.8951	50.8894	51.4619	40.074	3.2E+07
14	05-12-2000	51.7559	52.5759	51.7249	52.4057	40.809	2.4E+07
15	06-12-2000	53.3341	53.3341	52.0808	52.1814	40.6343	2E+07
16	07-12-2000	52.0499	52.661	51.9493	52.2278	40.6704	1.4E+07
17	08-12-2000	52.2974	52.9009	52.1659	52.5141	40.8933	2.8E+07
18	11-12-2000	52.7461	53.6822	52.6069	53.5275	41.6825	3.9E+07
19	12-12-2000	53.69	53.7209	53.1484	53.2722	41.4837	1.5E+07
20	13-12-2000	53.5352	54.0845	53.2877	53.7519	41.8572	2.6E+07

Fig. 3.3 Dataset

3.4.3 PRE-PROCESSING

Data pre-processing assumes a significant part in any acknowledgment cycle. Data pre- processing is a data mining strategy which is utilized to change the crude information in a valuable and proficient configuration. To shape the info pictures in a structure reasonable for division pre-handling is utilized. Data preprocessing is an essential step prior to building a model with these elements. It ordinarily occurs in stages:

- Data cleaning
- Data transformation
- Data reduction

3.4.3.1 Data Cleaning

Data cleaning is one of the significant pieces of AI. It has a huge impact in building a model. It doubtlessly isn't the fanciest piece of AI and simultaneously, there aren't any covered up stunts or insider facts to reveal. Be that as it may, legitimate information cleaning can represent the deciding moment of your undertaking. Proficient information researchers generally spend an exceptionally huge part of their experience on this step. In view of the conviction that, "Better information beats fancier calculations". On the off chance that we have a very much cleaned dataset, we can come by wanted results even with an exceptionally straightforward calculation, which can demonstrate extremely helpful on occasion. Clearly, various sorts of information will require various sorts of cleaning. Notwithstanding, this deliberate methodology can constantly act as a great beginning stage.

3.4.3.2 Data Transformation

In fact, by cleaning and smoothing the data, we have previously performed information adjustment. Notwithstanding, by information change, we comprehend the techniques for transforming the information into a proper configuration for the computer to gain from. Information change is the cycle where information is taken from its crude, soloed furthermore, standardized source state and change it into information that is combined, correspondingly displayed, de-standardized, and prepared for examination. Without the right innovation stack set up, information change can be tedious, costly, and monotonous. By and by, changing the information will guarantee greatest information quality which is basic to acquiring precise investigation, prompting important bits of knowledge that will ultimately engage information driven choices. Building and preparing models to handle information is a splendid idea, and more ventures have taken on, or

then again plan to send, AI to deal with numerous reasonable applications. However, for models to gain from information to make significant expectations, the actual information should be coordinated to guarantee its examination yield important bits of knowledge.

3.4.3.3 Data Reduction

Data reduction is a cycle that diminished the volume of unique information and addresses it in a lot more modest volume. Information decreases strategies guarantee the respectability of information while diminishing the information. The time expected for information decrease shouldn't eclipse the time saved by the information mining on the diminished informational index.

1. Data Reduction Techniques:

Dimensionality Reduction: - The dimensionality reduction refers to the strategy of reducing the component of an information data set. As a rule, AI datasets (highlightset) contain many segments (i.e., highlights) or a variety of focuses, making an enormous circle in a three-layered space. By applying dimensionality reduction, you can diminish or cut down the quantity of sections to quantifiable counts, in this way changing the three-layered circle into a two-layered object.

Types of dimensionality reduction are –

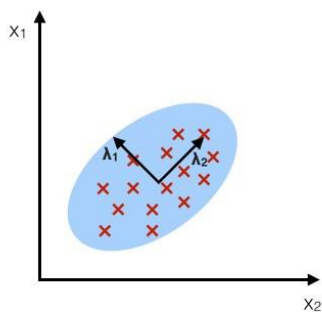
- a. **Principal Component Analysis:** - Principal Component Analysis is one of the main direct procedures of dimensionality reduction. This technique plays out immediate planning of the information to a lesser layered space in a manner that boosts the difference of the information in the low-layered portrayal. Basically, it is a factual system that symmetrically changes over the 'n' directions of a dataset into another arrangement of n facilitates, known as the primary parts. This transformation brings about the formation of the primary head part having the greatest change. Each succeeding head part bears the most elevated conceivable change, under

the condition that it is symmetrical(not corresponded) to the former parts. The PCA change is delicate to the general scaling of the first factors. Consequently, the information section ranges should initially be standardized prior to carrying out the PCA strategy. Something else to recall is that utilizing the PCA approach will make your dataset lose its interpretability.

- b. Linear Discriminant Analysis:** - The linear discriminant analysis is a speculation of Fisher's linear discriminant technique that is generally applied in stats, pattern recognition, and AI. This strategy plans to find a direct mix of elements that can describe or separate between at least two classes of items. It addresses information in a manner that expands class distinctness. While objects having a place with a similar class are compared through projection, objects from various classes are organized far separated.

PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation

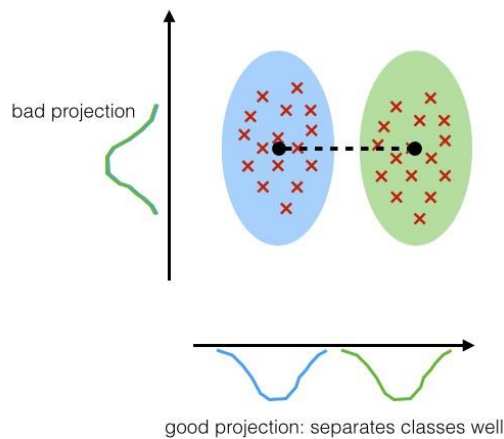


Fig. 3.4 PCA and LDA [10]

Above figures depict that how PCA and LDA work in respect to variance and what are their good projection and bad projection in respect to variance only. Also, how classes are separated using PCA and LDA.

3.4.4 IMPLEMENTING MODEL

Now, comes the real part where we at long last get to involve the fastidiously pre-arranged information for model building. Contingent upon the information type (subjective or quantitative) of the objective variable (usually alluded to as the Y variable) we are either going to fabricate an order (on the off chance that Y is subjective) or then again relapse (assuming Y is quantitative) model.

3.4.4.1 Learning Algorithms

The machine learning algorithms can be classified into three categories:

1. **Supervised Learning** – In supervised learning, every model is a pair comprising of an information object (commonly a vector) and an ideal result esteem (likewise called the administrative sign). A supervised learning calculation analyze the preparation information and produces a derived capability, which can be utilized for planning new models.

An ideal situation will consider the calculation to accurately decide the class marks for concealed examples. It is an AI task that lays out the numerical connection between input X and yield Y factors. Such X, Y pair comprises the marked information that are utilized for model structure in a work to figure out how to foresee the result from the info. Supervised learning problems can be further classified into classification and regression problems.

- a. **Classification:** When the output variable is a category then it is a classification problem. Example: - “Red” or “Blue”.
- b. **Regression:** When the output variable is a real value then it is a regression problem. Example: - “Weight” or “Rupees”.

2. Unsupervised Learning – In unsupervised learning, is an task that utilizes just the information X factors. Such X factors are unlabeled information that the learning calculation involves in demonstrating the inborn construction of the information. Unsupervised learning problems can be additionally classified into association and clustering problems.

a. Clustering: When we try to find the inherent grouping in the data, this kind of problem is known as clustering problem.

Example: - Grouping customers by ordering sequence.

b. Association: When we try to find the rules that can describe the large portions of our data, this kind of problem is known as association problem.

Example :- People selling A are also tend to sell B.

3. Reinforcement Learning – It is about making a reasonable move to expand compensation in a specific circumstance. It is utilized by different programming and machines to find the most ideal way of behaving or way it ought to take in a particular circumstance.

Reinforcement learning varies from the supervised learning in a manner that in supervised learning the preparation information has the response key with it so the model is prepared with the right respond to itself though in reinforcement learning, there is no response except for the support specialist chooses how to play out the given assignment. Without any a preparation dataset, it will undoubtedly gain from its insight. It is a machine learning task that settles on the following strategy and it does this by learning through experimentation with an end goal to augment the prize.

3.4.4.2 Algorithms used in our project

- **Linear Regression**

Linear Regression is a machine learning algorithm that is commonly used to make predictions. This is a supervised machine learning algorithm as this uses the supervised type of dataset to make predictions. This is one of the easiest but most powerful statistical methods that allow examining the relationship between two or more variables of interest. This algorithm uses at least two variables in which one of which is the dependent variable and another one is the independent variable. Here dependent variable means the factor that is to be predicted for example in our case the dependent variable is the future prices of a particular stock. Independent variables refer to the variables that have a huge impact on the dependent variable. In our model past prices of a stock, and date are the kind of independent variables because these are the factors on which the future prices of a particular stock depend. Talking about the output for regression models, the output must be in a continuous form. Linear regression basically draws an optimal line that is going to be the best fit line for our model. And the objective function which is used in linear regression is sum squared error and that should be of minimum value. Linear regression is of two types, first is Simple Linear Regression and another is Multiple Linear Regression. For Simple Linear regression there is just one (input) independent variable and one (output) dependent variable whereas in multiple linear regression there are multiple independent (input) variables and just one dependent variable (output).

Mathematics of Linear Regression

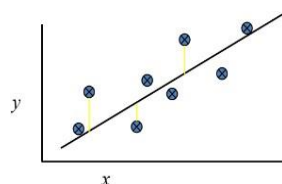


Fig. 3.5 Linear Regression

Size (feet ²)	Price (\$1000)
<i>x</i>	<i>y</i>
2104	460
1416	232
1534	315
852	178
...	...

Fig. 3.6 Example

Here there are some data points in the X and Y planes which are distributed accordingly.

Here our task is to draw a line that suits the given dataset. For such kind of problem, we use a linear regression model that will predict the best fit line for the given dataset.

The equation for the best fit line is: $Y = \beta_0 + \beta_1 X$

Here Y is the dependent variable and X is the independent variable β_0 the Y-intercept and β_1 the slope. Now find the values for the coefficients which minimize.

To find the values for the coefficients which minimize the objective function we take the partial derivatives of the objective function (SSE) with respect to the coefficients.

$$\beta_1 = \frac{n \sum xy - \sum x \sum y}{(n \sum x^2 - (\sum x)^2)}$$

$$\beta_0 = \frac{\sum y - \beta_1 \sum x}{n}$$

- **Long – Short term memory**

LSTM is an ANN used in the fields of artificial intelligence and deep learning. It is an advanced part of RNN, a sequential network, which allows information to persist. As RNN is not able to handle the problem of vanishing gradients but LSTM is capable of handling the vanishing gradients problem. As RNN process entire sequence of data at one time, so LSTM also do same. It has feedback connections which make them different from feed forward NN.

This is how typical RNN looks like where :-

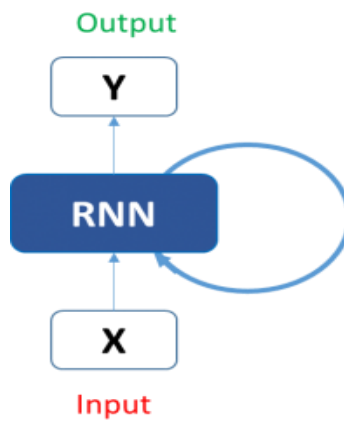


Fig. 3.7 Typical RNN model [11]

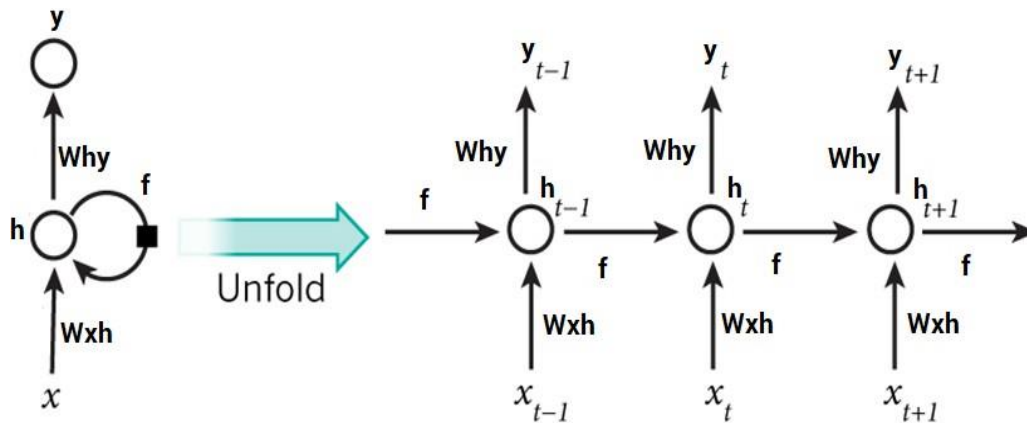


Fig. 3.8 Typical LSTM model [12]

This figure clearly depicts that how SMP is done using LSTM as here every prediction at time h_t is dependent on previous information i.e. for prediction of stock prices we must have its previous prices and its information to predict using LSTM.

Output of LSTM at particular point in time is dependent on three things:

- Cell state – current long – term memory of the network
- Previous hidden state – output at previous point in time
- Input data at the current time step.

LSTMs use a series of ‘gates’ which control how information in a sequence of data comes into, is stored in and leaves the network. There are three gates in a typical LSTM; forget gate, input gate and output gate. These gates can be thought of as filters and are each their own neural network.

➤ **Architecture of LSTM :-**

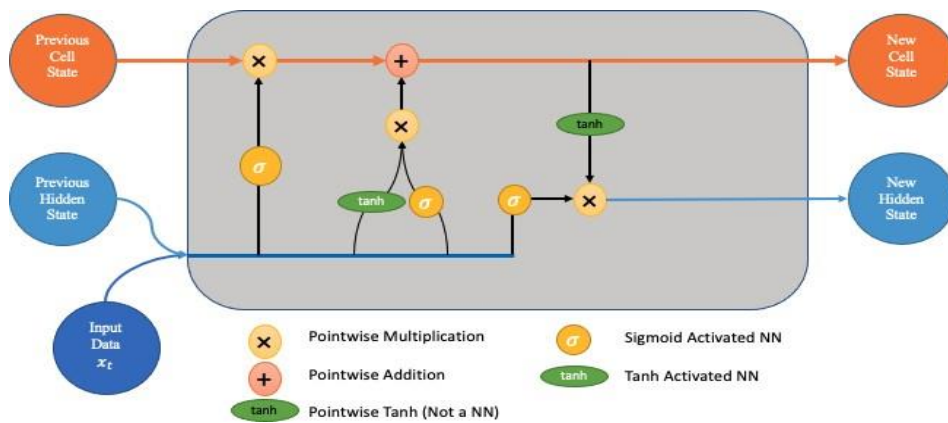


Fig. 3.9 Architecture of LSTM [13]

Above figure shows that how LSTM model looks now let us look at the architecture of LSTM model deeply

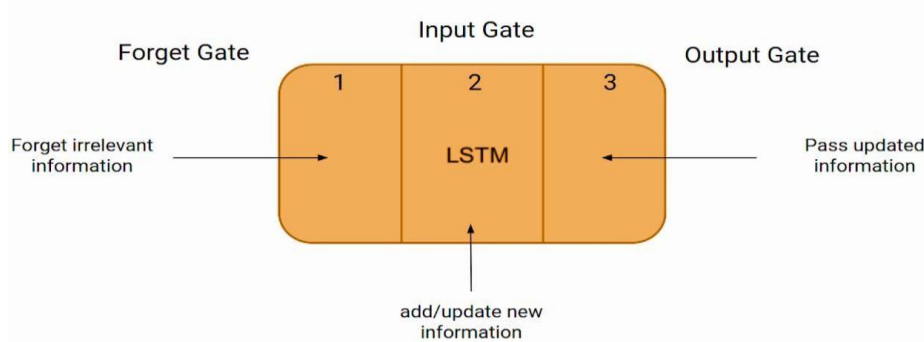


Fig. 3.10 Gates in LSTM [14]

Above figure shows that how different gates work in LSTM.

➤ **Types of Gates :-**

a. Forget Gate :- First gate is forget gate in which we decide which information are useful for further gates.

Equation for forget gate is :-

Forget Gate:

- $f_t = \sigma(x_t * U_f + H_{t-1} * W_f)$

Fig. 3.11 Forget Gate Equation

Here,

- X_t : input to the current timestamp.
- U_f : weight associated with the input
- H_{t-1} : The hidden state of the previous timestamp
- W_f : It is the weight matrix associated with hidden state

Later sigmoid function is applied which make f_t value between 0 and 1. If f_t is equal to 0 then model forgot everything and if f_t is equal to 1 then model don't forget anything.

$$C_{t-1} * f_t = 0 \quad \dots \text{if } f_t = 0 \text{ (forget everything)}$$

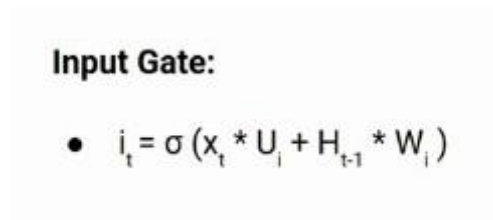
$$C_{t-1} * f_t = C_{t-1} \quad \dots \text{if } f_t = 1 \text{ (forget nothing)}$$

Fig. 3.12 Updated Equation

Above figure shows the new updated equation.

b. Input Gate :- This gate determines that what information should be added to LSTM network from the previous hidden state and new input data. This gate works after the forget gate.

Equation for input gate is :-



The image shows a light blue rectangular box containing the text "Input Gate:" followed by a bullet point and the equation $i_t = \sigma(x_t * U_i + H_{t-1} * W_i)$.

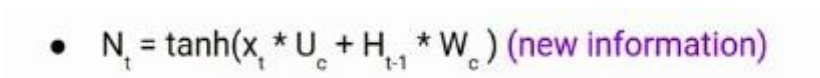
Fig. 3.13 Input Gate Equation

Here,

- X_t : input to the current timestamp.
- U_i : weight matrix of input
- H_{t-1} : The hidden state of the previous timestamp
- W_i : It is the weight matrix of input associated with hidden state

Again, sigmoid function is applied which make i_t value between 0 and 1.

Now new information has to be passed further so the information which has to be passed to the cell state uses the activation function \tanh at previous timestamp $t - 1$ and input x at timestamp t . Due to \tanh function, value of new information will be between -1 and 1.



The image shows a light blue rectangular box containing a bullet point and the equation $N_t = \tanh(x_t * U_c + H_{t-1} * W_c)$ with the text "(new information)" in purple.

Fig. 3.14 New Information Equation

If value of N_t is negative the information is subtracted from the cell state and if the value is positive the information is added to the cell state at the current timestamp.

Updated equation becomes :-

$$C_t = f_t * C_{t-1} + i_t * N_t \text{ (updating cell state)}$$

Fig. 3.15 Updated Equation

C_{t-1} is the cell state at the current time stamp.

- c. **Output Gate :-** This gate gives the output for the LSTM model after all previous processes.

Equation for output gate is :-

Output Gate:

- $o_t = \sigma(x_t * U_o + H_{t-1} * W_o)$

Fig. 3.16 Output Gate Equation

Here,

- X_t : input to the current timestamp.
- U_o : weight associated with the output.
- H_{t-1} : The hidden state of the previous timestamp
- W_o : It is the weight matrix of output associated with hidden state.

Again, sigmoid function is applied which make o_t value between 0 and 1.

For calculating current hidden state we will use below equation :-

$$H_t = o_t * \tanh(C_t)$$

Fig. 3.17 Hidden state Equation

And for the final output we will just apply SoftMax activation function as shown below :-

$$\text{Output} = \text{Softmax}(H_i)$$

Fig. 3.18 Final output Equation

➤ **Activation function used :-**

• **Sigmoid function: -**

The fundamental justification for our use of sigmoid capability is that it exists between (0 to 1). This makes it particularly useful for models where we need to predict the likelihood as a result of the input. The sigmoid is the best option because the range of likelihood for anything is only 0 to 1. The ability can be differentiated. That suggests that there are two locations where we can determine the sigmoid bend's slope. Although the capability is monotonous, the subsidiary is not. A brain network may become unresponsive during the preparation stage due to the strategic sigmoid capacity.

• **SoftMax function: -**

The SoftMax activation function changes the raw outputs of the neural network into a vector of probabilities, basically a likelihood conveyance over the info classes. Consider a multiclass characterization issue with N classes. The softmax enactment returns a result vector that is N passages long, with the section at file I comparing to the likelihood of specific information having a place with the class I. The mathematical operation known as Softmax converts a vector of integers into a vector of probabilities, with the probabilities of each element equal to its overall magnitude.

The softmax capacity's use as an enactment capability in a brain network model is its most well-known application in applied AI. The

organization is specifically made to produce N values, one for each class in the arrangementwork, and the results are standardized by using the softmax capability to convert them from weighted aggregate qualities into probabilities that add upto 1. Each value in the softmax capability's output is translated into the likelihood of enrollment for each class.

- **Tanh function :-**

Tanh i.e., hyperbolic tangent activation function, is one of the most important activation functions. This function is somewhat like a sigmoid function but better than a sigmoid. Tanh is an s-shaped graph with a range from (-1 to 1). Basically, tanh is used for classification between two classes. Also, tanh is used in feed-forward neural networks.

- **Different optimizers used in LSTM :-**

1. Gradient Descent
2. Adam
3. Stochastic Gradient Descent (SGD)
4. Mini Batch Stochastic Gradient Descent (MB-SGD)
5. SGD with momentum
6. Adaptive Gradient (AdaGrad)
7. AdaDelta
8. RMSprop

- **Gradient Descent**

The most basic yet most common improvement computation is gradient descent. It is heavily used in estimates for straight relapse and characterization. Brain networks also use a slope- plunge computation for backpropagation.

A first-request streamlining computation called Gradient Descent depends on the main request subsidiary of a bad luck capability. It determines how the loads should be modified in order for the

capability to be reduced to a minimum. Backpropagation involves moving the misfortune from one layer to the next and changing the model's limits, or loads, based on the misfortunes in order to restrict the misfortune.

- **Adam Optimizer**

A slope plunge advancement method computation is called adaptive movement estimation. The approach is incredibly effective when dealing with significant problems involving a lot of information or boundaries. It works well and takes minimal RAM. Naturally, the computation is a combination of the inclination drop with force and the RMSprop calculations. An extension of the stochastic angle plunge technique, the Adam optimization approach has recently gained increased popularity for machine learning applications in computer vision and natural language processing. Adam is a streamlined calculation that can be used in place of the conventional stochastic slope plunge approach to iteratively prepare information while refreshing network loads. Adam was first mentioned in a 2015 ICLR work (paper) by Diederik Kingma from OpenAI and Jimmy Ba from the University of Toronto titled "Adam: A Strategy for Stochastic Enhancement." Adam is portrayed by the developers as combining the advantages of two distinct stochastic angle drop augmentations. Every cycle, the Adam's actual step size is roughly constrained by the step size hyper-boundary. This characteristic increases the past unintuitive learning rate hyper-boundary with instinctive comprehension.

The Adam update rule's step size is independent of the magnitude of the inclination, which is helpful for traversing areas with modest angles (for example, saddle focuses or gorges). SGD fights to quickly navigate through these locations.

Adam was designed to combine the advantages of RMSprop, which performs brilliantly in online environments, and AdaGrad, which performs admirably with modest slopes. Having both of these enables

us to work with Adam on a wider range of projects. Adam can also be viewed as a combination of RMSprop and SGD with energy.

- **Stochastic Gradient Descent**

It is a different version of gradient descent. It makes too many tries to update the model's limits. In this, after calculating misfortune on each prepared model, the model boundaries are adjusted. In contrast to Slope Plummet, if the dataset has 1000 lines, SGD will refresh the model boundaries more than once in a single pattern of dataset.

- **MB-SGD**

The most effective gradient descent algorithm is the mini batch. It is an improvement for both regular slope drops and SGDs. After each bunch, the model bounds are refreshed. The borders are then renewed after each clump, dividing the dataset into various batches.

- **SGD with Momentum**

There is growing momentum to reduce the large SGD discrepancy and loosen the union. It minimises the transition to the unimportant course and speeds up the blending toward the important bearing. In this method, the additional hyperparameter referred to as energy, denoted by the symbol " β ," is used.

- **AdaGrad**

The fact that the learning rate is constant for all parameters and for each cycle is one of the limitations of all optimizers. AdaGrad optimizer modifies the pace of learning. For each boundary and time step 't', the learning rate " η " is altered. A second request has been sent to the optimizer algorithm. It functions using the error function's derivative.

- **AdaDelta**

It is an expansion of AdaGrad which will in general eliminate the smalling rate of learning issue of it. Rather than aggregating recently squared slopes, AdaDelta will minimize the screen amassed old inclinations to few decent size w .

For it dramatically moving normal is utilised as opposed to the amount of the multitude of angels.

- **RMSprop**

The RMSprop optimizer is like the slope plummet calculation with force. The RMSprop optimizer confines the motions in the upward heading. Subsequently, we can build our learning rate and our calculation could steer bigger strides in the even bearing uniting quicker. The contrast between RMSprop and gradient descent is in how the slopes are determined. The accompanying conditions show how the slopes are determined for the RMSprop and gradient with momentum. The worth of force is signified by beta and is normally set to 0.9. In the event that you are not keen on the number related to the enhancer, you can simply skirt the accompanying conditions.

3.4.4.3 Model Evaluation and Prediction

- **For linear regression**

After data pre-processing and splitting of the dataset, we fitted our model in the Linear Regression algorithm and evaluated the accuracy score for training and testing datasets. Along with this, we also computed coefficients and intercepts for the linear regression model. Then after this, we imported the math library from python. Under this, we calculated the mean absolute error, mean squared error, and root mean squared error. We also plot a graph between the Scaled INR and Time Scale. In this graph two values are shown, one if for true value, and another one is the value predicted by the model LR. Also, a bar graph is plotted for the actual and predicted prices for a particular stock.

- **LSTM**

After importing all the main libraries, and doing the basic data cleaning, and pre-processing steps. After the cleaning, sanitizing, and pre-processing of the dataset, the dataset is ready for splitting. Now the dataset is split into testing and training datasets so that we can compute the training and testing accuracies. We applied the LSTM model to the dataset and along with this; we calculated the shapes of testing and training datasets. For the LSTM models, we must call a particular number of epochs and batch size. For this, we tried a different number of epochs and batch sizes and their combinations too. Like firstly we tried epoch number 50 and batch size 8, then epoch number and batch size, and at last we tried with epoch number and batch size.

After all these, we plotted a graph between the time scale and scaled INR. This graph is plotted for both values i.e., for true and LSTM values. In the last, we computed the accuracy of our LSTM model

3.5 WEB APPLICATION DEVELOPMENT

In previous sections we saw some basic steps which are required to develop a model. Now we will see some basic steps to build web application and will discuss them briefly to know about them deeply and know that how our web application will be created.

3.5.1 IMPORTING LIBRARIES

3.5.1.1 Streamlit

An open - source Python library which is used to create web applications quickly and easily. It simplifies the process of creating web applications by providing the range of components such as sidebars, buttons, dropdowns and etc. It also allows to put on the images, logos and many other things to make an interactive website. With streamlit, we can create custom dashboards, data visualizations and machine learning models, all in one

place. So, with its simple and intuitive interface, it is an ideal tool for the development of web application without needing to be experts in web development.

3.5.1.2 Pandas

Pandas is characterized as an open-source library that gives elite execution information control in Python. The name of Pandas is gotten from the word Board Information, and that implies an Econometrics from Multi-layered information. It is utilized for information examination in Python and created by Wes McKinney in 2008.

Information examination requires heaps of handling, for example, rebuilding, cleaning or combining, and so on. There are various instruments are accessible for quick information handling, like Numpy, Scipy, Cython, and Panda. Yet, we lean toward Pandas since working with Pandas is quick, basic and more expressive than different devices.

Pandas is basically utilized for reshaping and turning of the information collections. It incorporates with different libraries like SciPy and scikit – learn.

3.5.1.3 yfinance

yfinance is a Python library which consists of all the stock prices data and financial data of the all stocks present at the Yahoo Finance. This library is used to retrieve all the historical, current and other financial stocks data from the Yahoo Finance. It gives data about stock prices like Opening Price, Closing Price, Adj Close, Volume, Dividend data and etc. of the particular selected stock. Users can specify the start date and close date and retrieve all the important information of the stock between the selected dates. Therefore, it is a powerful and flexible library for retrieving the financial data from Yahoo Finance.

3.5.1.4 Prophet

Facebook created the Python library Prophet for time series forecasting. The framework is based on a generalised additive model (GAM), which enables the incorporation of different seasonalities, trend changes, and holiday impacts. Prophet is a well-liked option for time series forecasting workloads since it is simple to use and requires little configuration.

Time series data having the following qualities can be used with Prophet:

- Daily, monthly, or hourly observations that have been made for at least a few months.
- Fourier series can be used to simulate seasonal trends
- Holidays and special occasions are examples of anomalies that may be modelled as regressors.

This library will be used as main algorithm for predicting the future prices of the particular stock and will be used in web application part for the forecasting of the stock price.

3.5.1.5 Datetime

A built-in Python library called the datetime module offers classes for dealing with dates and timings. It makes it simple for Python programmers to generate, modify, format, and interact with dates and times in their programs.

For representing dates and times, the datetime library offers a number of classes, including:

- A class for working with dates called **datetime.date** represents a date (year, month, and day).
- A class for working with times called **datetime.time** represents a certain time of day (hour, minute, second, and microsecond).
- a class for working with dates and times that reflects a precise moment in time (year, month, day, hour, minute, second, microsecond), is called **datetime.datetime**.

- A class for working with time lengths, **datetime.timedelta** provides the distinction between two dates or times.

3.5.1.6 Plotly

A free Python library for building interactive data visualizations is called Plotly. For making plots, charts, and dashboards that may be used for data exploration and communication, it offers a broad variety of graphing and charting capabilities. Plotly is a flexible tool for data visualization since it supports several languages and operating systems.

A variety of chart kinds, such as scatter plots, line charts, bar charts, and heatmaps, are available in the library. Additionally, it offers tools for making 3D animations, maps, and visualizations. Users have a variety of options for the colors, typefaces, and layouts that will display on their charts. Along with a number of other data formats, Plotly also supports CSV, Excel, and JSON.

3.5.2 UPLOADING DATASET

We have taken a dataset from the GitHub repository of data professor who is renowned youtuber known for making content about data science and python. We have uploaded his raw file which consists of several stock tickers which will be used to predict the prices of the selected stock ticker. Uploading of file is done using pandas command `pd.read_csv` which is the general way of uploading file in the Python.

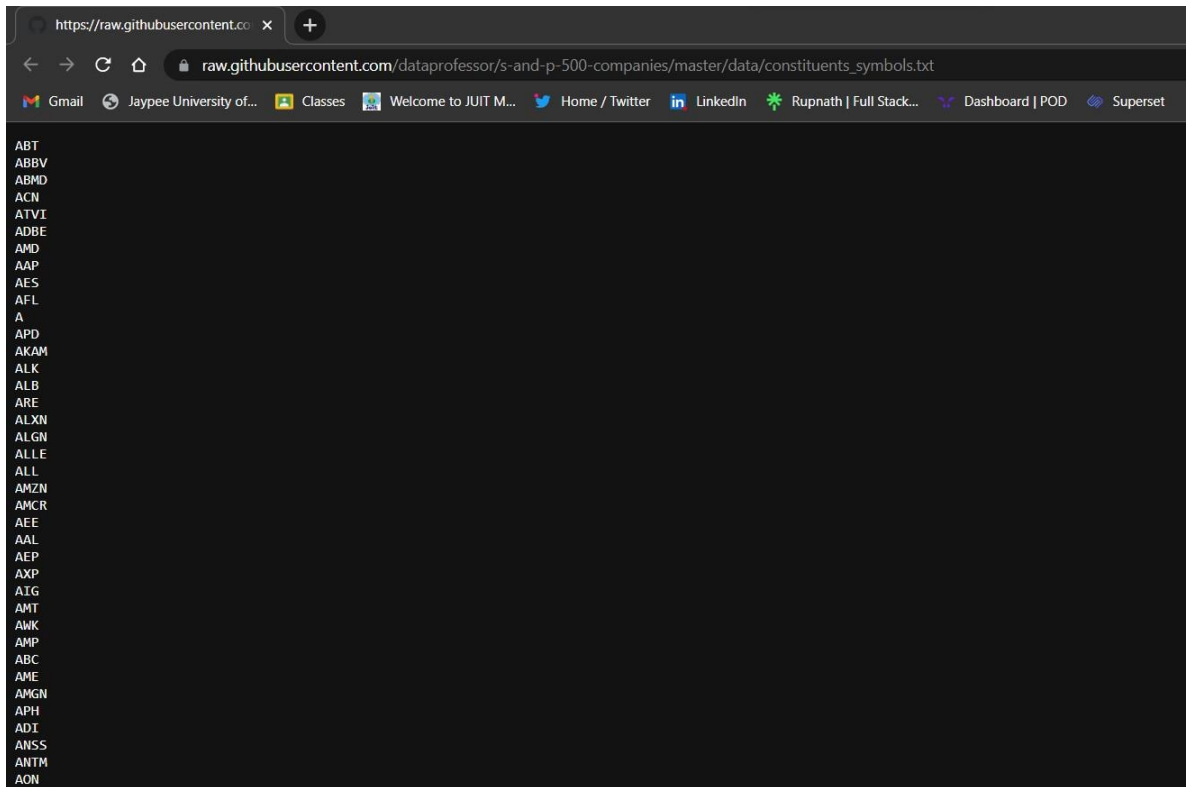


Fig. 3.19 Dataset used in web application [15]

3.5.3 IMPLEMENTING WEB APPLICATION USING STREAMLIT

Now after uploading the dataset we will write the code mainly using streamlit library and other libraries. Streamlit gives us many commands to make web application more attractive, some of the commands are :

- `st.write`
- `st.text`
- `st.markdown`
- `st.header`
- `st.subheader`
- `st.sidebar`
- `st.info`

Using above commands, we can create the web application in the python using streamlit and other libraries like yfinance, pandas and etc.

CHAPTER 4 – EXPERIMENT & RESULT ANALYSIS

4.1 System Configuration

4.1.1 Software Requirements

These are the software configurations used:

Operating System: Windows 11.

IDE: Google Colaboratory, Jupyter Notebook (Used for data augmentation).

Python: Python is an interpreted, high-level, general-purpose programming language that was developed by Guido Van Rossum and initially released in 1991. With its noticeable use of substantial Whitespace, Python places a strong emphasis on code readability. Its language constructs and object-oriented methodology are designed to aid programmers in creating clean, comprehensible code for both little and big projects. Python has garbage collection and dynamic typing. Programming paradigms including procedural, object-oriented, and functional programming are all supported.

Google Colaboratory: Colab is a completely cloud-based Jupyter notebook environment that is free to use. The notebooks you create can be simultaneously modified by your team members, exactly like you edit documents in Google Docs, and most significantly, it doesn't require any setup. Many well-known machine learning libraries are supported by Colab and are simple to load in your notebook.

Jupyter Notebook: A free, open-source, interactive web tool called Jupyter. Researchers can utilise a computational notebook to compile software code, computational results, explanatory text, and multimedia materials into a single document. Although computational notebooks have been around for a while, Jupyter has been incredibly popular in the last few years. An engaged user-developer community and a new architecture that permits the laptop have contributed to this quick uptake.

4.1.2 Hardware Requirements

These are the Hardware interfaces used Processor: Intel i5 9th Gen 4 cores 8 threads

RAM: 8GB DDR4 RAM

Monitor: 15'' full HD color monitor

Mouse: Scroll or optical mouse

Graphics Device: Nvidia GTX 1650 4GB DDR5 VRAM

Keyboard: Standard 110 keys keyboard

4.2 Results Obtained from Linear Regression and LSTM

4.3.1 Linear Regression

- Inline Graph of Linear Regression between true value and LR predicted value

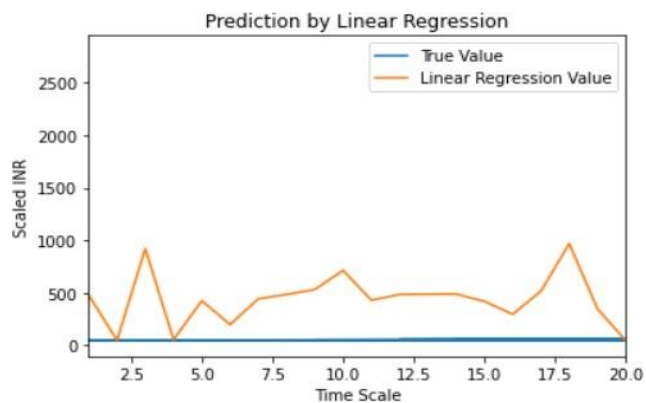


Fig 4.1 True Value and LR value

- Inline Graph of Linear Regression between Actual and Predicted values for random 25 Values.

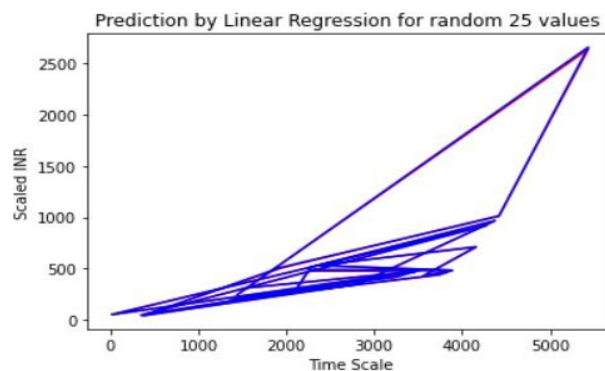


Fig 4.2 Red – Actual Price and Blue – Predicted Price for random 25 values

- Bar Graph of Linear Regression between Actual and Predicted values for random 25 Values.

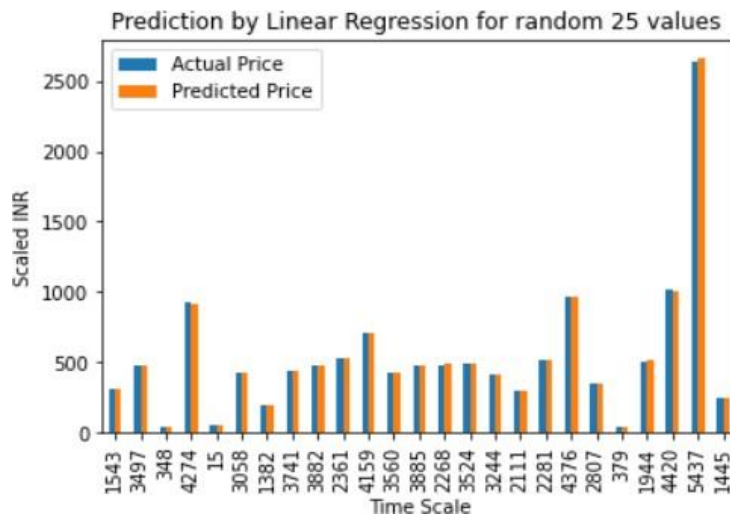


Fig 4.3 Blue – Actual Price and Orange – Predicted Price for random 25 values

- Various metrics of LR model

Table 4.1 Various measuring parameters of LR model

S.No	Metrics	Value
1.	MAE	2.87
2.	MSE	26.48
3.	RMSE	5.14

- Accuracies of LR model

Table 4.2 Various accuracies of LR model

S.No	Accuracies	Value
1.	Training Accuracy	99.992
2.	Testing Accuracy	99.993

4.3.2 LSTM

- Working of LSTM.

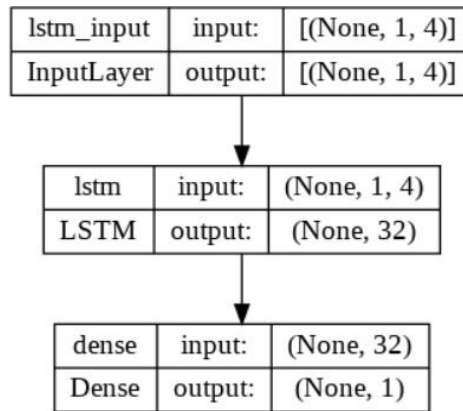


Fig 4.4 Working of LSTM

- Model Summary

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 32)	4736
dense (Dense)	(None, 1)	33

```

Total params: 4,769
Trainable params: 4,769
Non-trainable params: 0

```

Fig 4.5 LSTM model summary

- Training model

```

Epoch 1/50
653/653 [=====] - 5s 4ms/step - loss: 537560.8125 - accuracy: 0.0000e+00
Epoch 2/50
653/653 [=====] - 4s 6ms/step - loss: 492377.7500 - accuracy: 0.0000e+00
Epoch 3/50
653/653 [=====] - 5s 7ms/step - loss: 445401.4688 - accuracy: 0.0000e+00
Epoch 4/50
653/653 [=====] - 5s 7ms/step - loss: 395325.0312 - accuracy: 0.0000e+00
Epoch 5/50
653/653 [=====] - 5s 7ms/step - loss: 345547.8125 - accuracy: 0.0000e+00
Epoch 6/50
653/653 [=====] - 3s 4ms/step - loss: 298838.0312 - accuracy: 0.0000e+00
Epoch 7/50
653/653 [=====] - 2s 4ms/step - loss: 257071.3438 - accuracy: 0.0000e+00
Epoch 8/50
653/653 [=====] - 2s 4ms/step - loss: 221175.1094 - accuracy: 0.0000e+00
Epoch 9/50
653/653 [=====] - 2s 4ms/step - loss: 191070.4844 - accuracy: 0.0000e+00
Epoch 10/50
653/653 [=====] - 2s 4ms/step - loss: 165901.0781 - accuracy: 0.0000e+00

```

Fig 4.6 LSTM model training – 1

```

Epoch 35/50
653/653 [=====] - 2s 4ms/step - loss: 940.5564 - accuracy: 0.0000e+00
Epoch 36/50
653/653 [=====] - 3s 4ms/step - loss: 885.7559 - accuracy: 0.0000e+00
Epoch 37/50
653/653 [=====] - 2s 4ms/step - loss: 834.7105 - accuracy: 0.0000e+00
Epoch 38/50
653/653 [=====] - 2s 4ms/step - loss: 785.5075 - accuracy: 0.0000e+00
Epoch 39/50
653/653 [=====] - 2s 3ms/step - loss: 737.2723 - accuracy: 0.0000e+00
Epoch 40/50
653/653 [=====] - 2s 3ms/step - loss: 689.3998 - accuracy: 0.0000e+00
Epoch 41/50
653/653 [=====] - 3s 4ms/step - loss: 642.1830 - accuracy: 0.0000e+00
Epoch 42/50
653/653 [=====] - 2s 4ms/step - loss: 596.1466 - accuracy: 0.0000e+00
Epoch 43/50
653/653 [=====] - 2s 3ms/step - loss: 551.8044 - accuracy: 0.0000e+00
Epoch 44/50
653/653 [=====] - 2s 3ms/step - loss: 509.3723 - accuracy: 0.0000e+00
Epoch 45/50
653/653 [=====] - 3s 4ms/step - loss: 469.3961 - accuracy: 0.0000e+00
Epoch 46/50
653/653 [=====] - 3s 4ms/step - loss: 431.9879 - accuracy: 0.0000e+00
Epoch 47/50
653/653 [=====] - 4s 6ms/step - loss: 397.3802 - accuracy: 0.0000e+00
Epoch 48/50
653/653 [=====] - 2s 3ms/step - loss: 365.5277 - accuracy: 0.0000e+00
Epoch 49/50
653/653 [=====] - 2s 3ms/step - loss: 336.3944 - accuracy: 0.0000e+00
Epoch 50/50
653/653 [=====] - 2s 4ms/step - loss: 309.9590 - accuracy: 0.0000e+00

```

Fig 4.7 LSTM model training – 2

- Inline graph of LSTM between true value and LSTM value

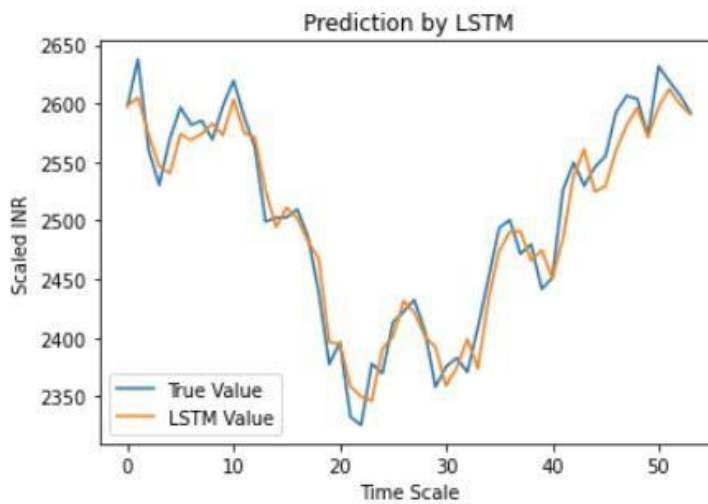


Fig 4.8 True Value and LSTM value

- Various metrics of LSTM model

Table 4.3 Various measuring parameters of LR model

S.No	Metrics	Value
1.	MAE	17.47
2.	MSE	420.34
3.	RMSE	20.50

4.3 Results of Web Application

- Interface

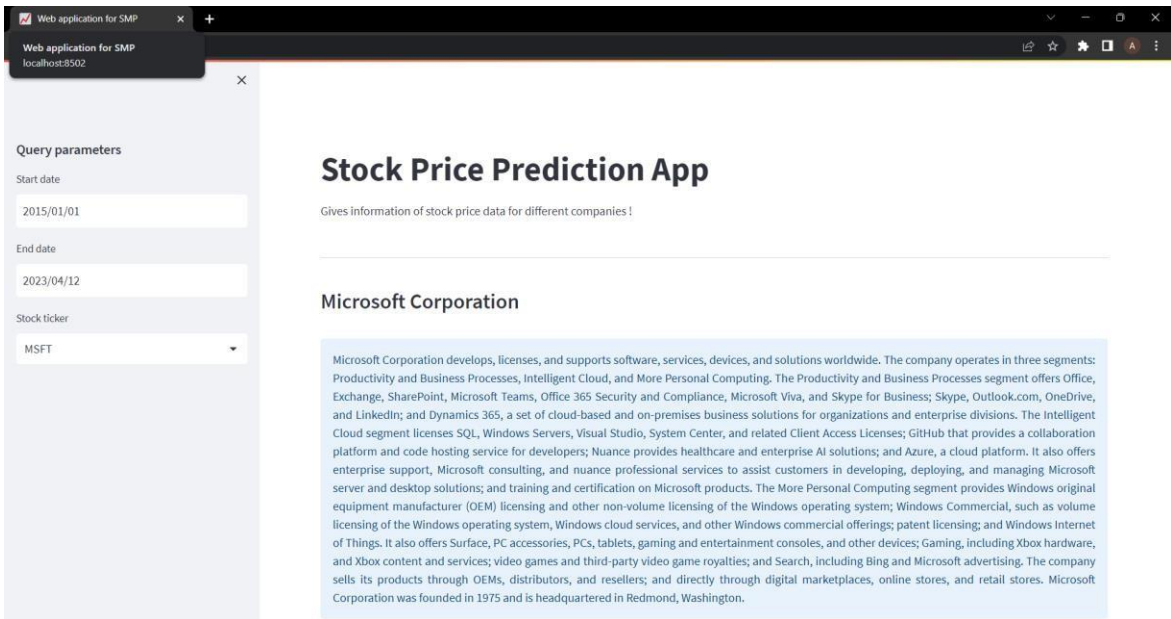


Fig 4.9 Interface

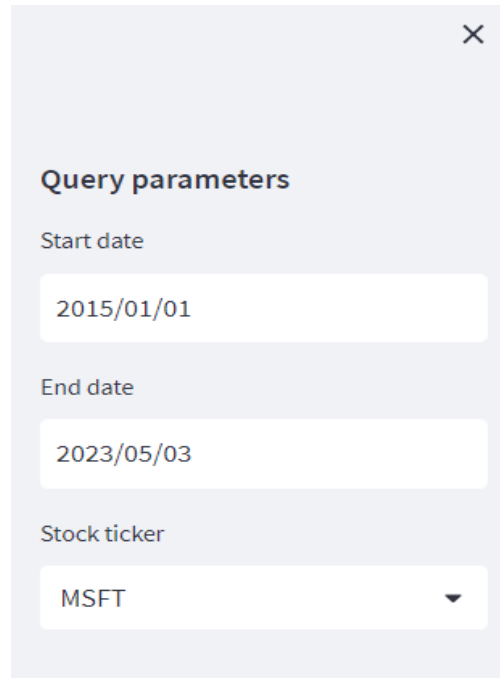


Fig 4.10 Query Parameters

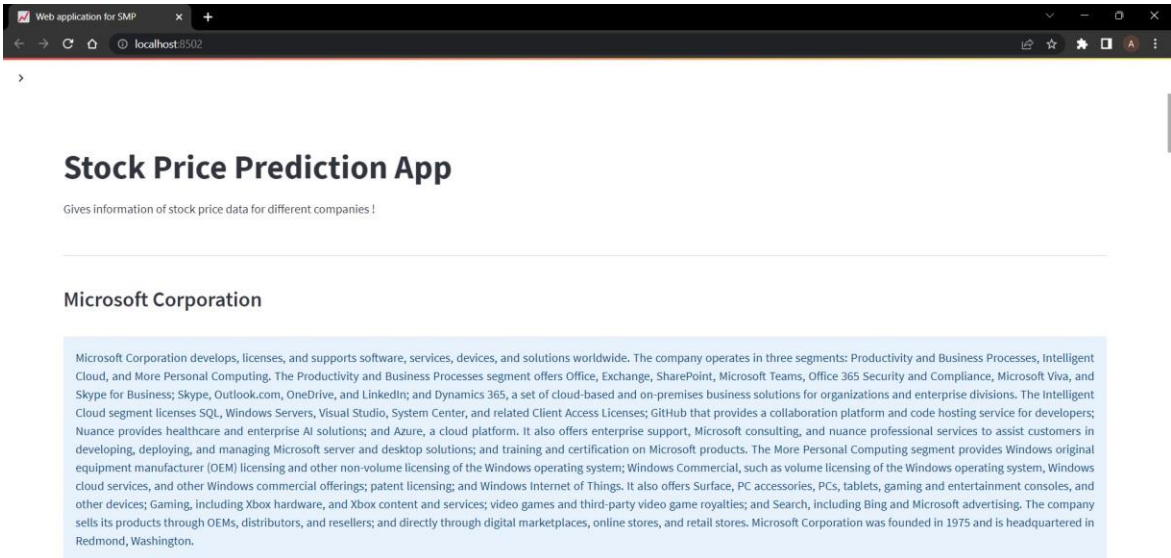


Fig 4.11 Information about the selected Stock Ticker

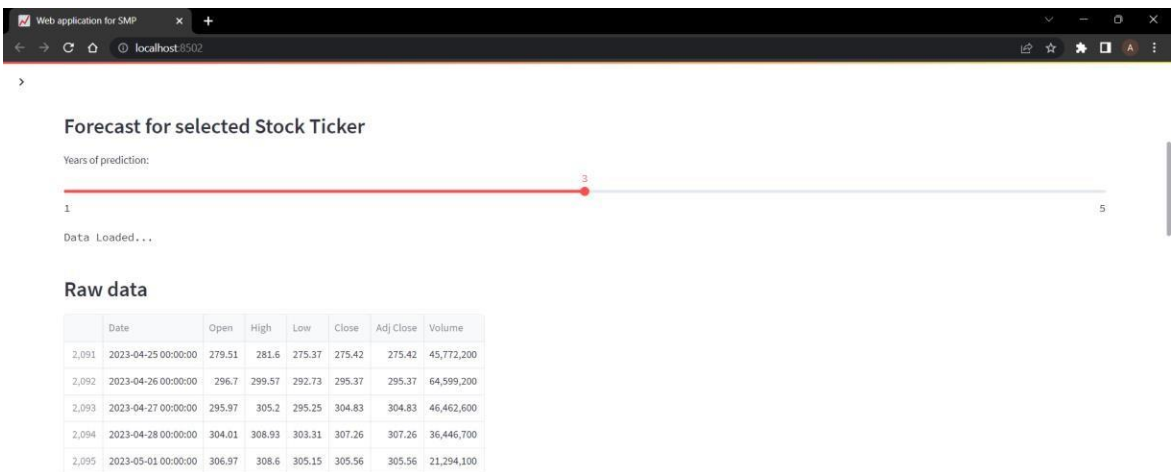


Fig 4.12 Select Years of Prediction and Raw Data



Fig 4.13 Plot for Stock opening and closing prices

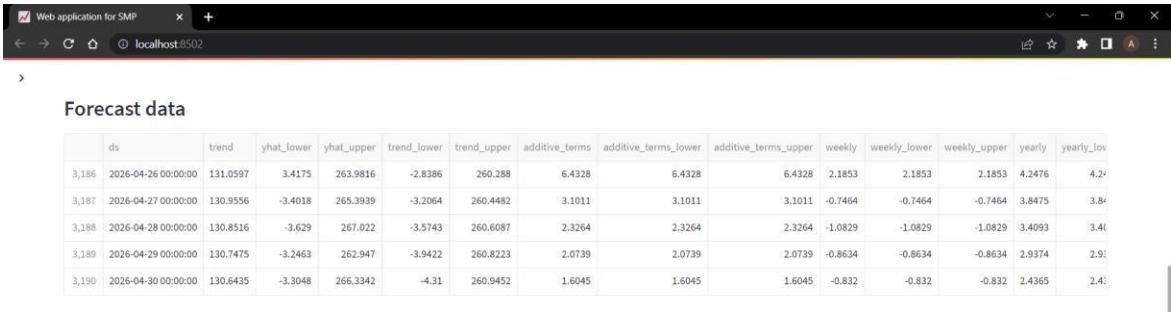


Fig 4.14 Forecast Data

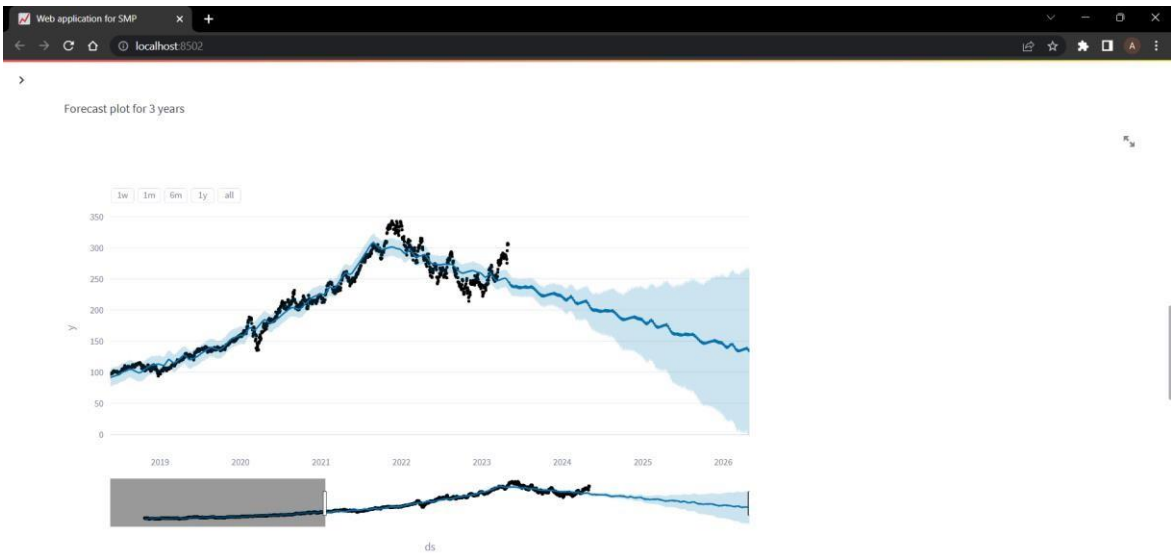


Fig 4.15 Forecast plot for n future years

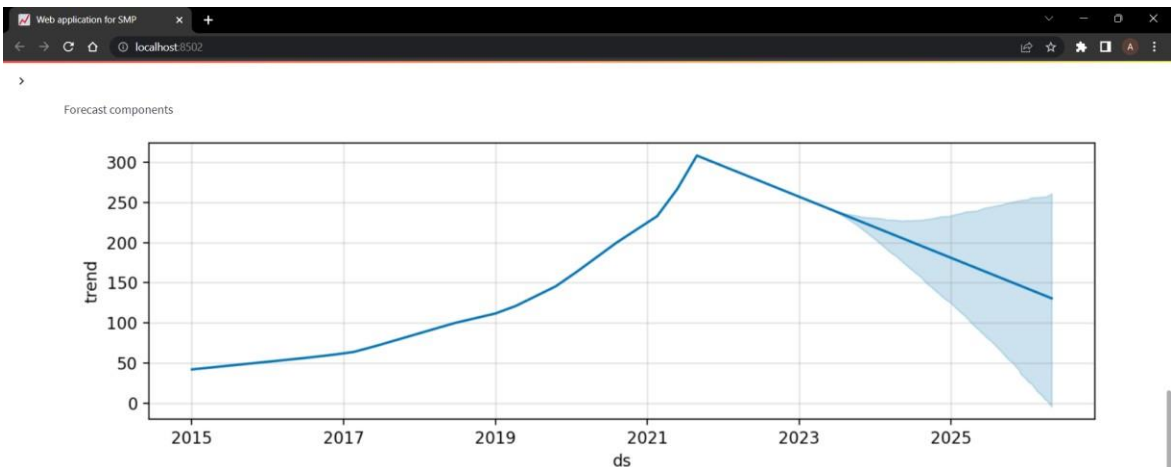


Fig 4.16 Plot for the selected years

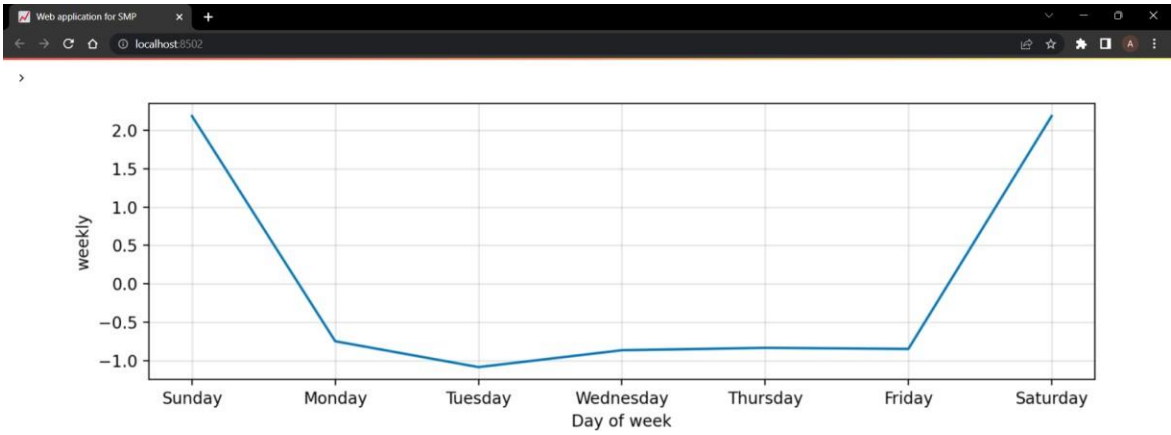
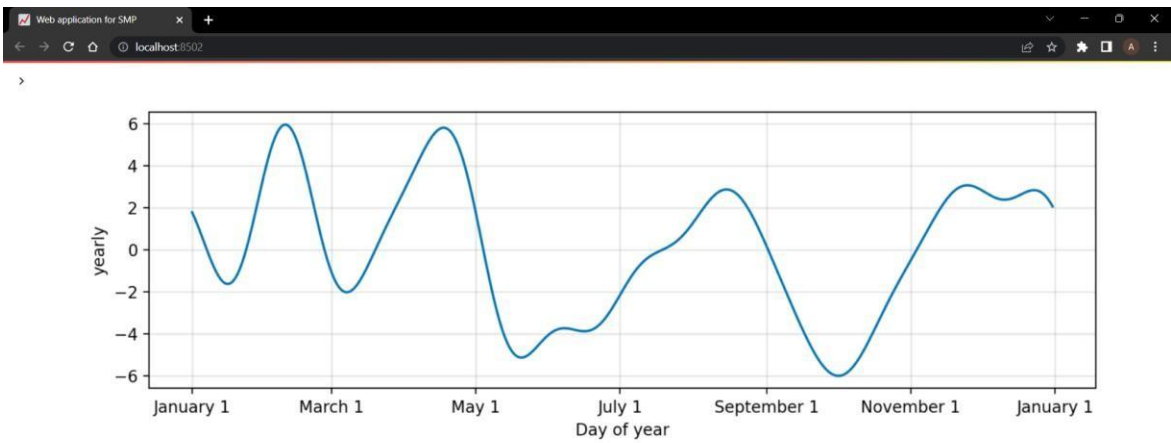


Fig 4.17 Day-wise Plot for the selected stock



Credits

- App built by --Aanjaneya Sharma and Sulbha Sharma
- Built in Python using `streamlit`, `yfinance`, `prophet`, `pandas` and `datetime`

Fig 4.18 Day of year plot for the selected stock

CHAPTER 5– CONCLUSION

5.1 Conclusion

Our project Developing Machine learning-based models to predict stock prices deals with the prediction of future prices for a particular stock. The principal reason for this project is to design an efficient model for the shareholders who invest money in the share market. In order to get more gain and less loss they might check the future prices of a stock and then decide accordingly whether to buy or sell a particular stock. There are various models present in Machine Learning which we can use to predict stock prices. We analyze the models and studied their characteristics. We also studied different research papers in order to choose our model or algorithm. While doing the literature survey, we found that models like Linear Regression, K Nearest Neighbors, SVM (Support Vector Machine), RF (Random Forest), and LSTM (Long Short-Term Memory) have been utilized in a lot of research. We also find that with the help of these models the accuracy achieved is nearly up to 98%. And no further better model using this model can be created. So, we are here trying to use the LSTM (Long Short-Term Memory) which has a lot of future scopes. The model has variations in accuracy if we do changes in the model or we can create our own model. In this project, Linear Regression, and LSTM models were used to achieve a high-performance model which is efficient for stock price prediction. We are using principles of Supervised Machine learning in order to make our model. Before preparing the model, we have done the pre-processing of the data and normalized the data in order to have the best data to train our model. In our model, we have used Linear regression and LSTM algorithms. Linear regression is simply a regression model which works for two or more variables. Here we used Multiple Linear Regression because in multiple linear regression we have two or more independent variables as input but there is only one dependent variable i.e., there is only one output, for our project the output variable is predicted future stock prices based on past stock prices. In LSTM we used ADAM optimizer and sigmoid function, tanh function, and SoftMax function as activation functions. As activation function introduces non-linearity to the

function. So, the activation function sigmoid, and tanh gives non-linearity to the function and are used in input layers, whereas SoftMax is an activation function that is used only in the output layer. We have trained the model on various fitting configurations that are on different numbers of epochs and batch sizes. While working on the model we found that the best model is trained on a batch size of 8 and the number of epochs is 50.

In terms of web application, we created the application in which we can check the future predicted prices of the particular stock by selecting the date intervals from starting date to end date. In this application one can also see the different graphs giving information about the stock opening prices, stock closing prices and stock future price. Other graphs give information about the stock's prices – yearly, day of week, day of year.

5.2 Future Scope

In this Project, for implementing Linear Regression and LSTM a thorough examination is needed to determine the optimal choice for the different parameters. The SoftMax, tanh, and sigmoid activation functions are used in this project. Also, we can change the parameters of the Linear Regression and LSTM model to improve the model. Also, we can improve both models by utilizing the progressed information increase method and further developing a LSTM model to show up as an official choice for stock price prediction.

In web application we can improve the interface and can add the databases so that a user can add his/her login information along with the different stock prices data. We can add more interactivity and layout to it so that the application can look more attractive and more reliable.

5.3 Applications

Stock price prediction is one of the most advanced topics in the field of predictions. The stock price prediction system can be used by shareholders in order to attain more and more gain and less loss. As of now, so many people are indulged in the share market and a huge amount of money is used in this process. So, to have more profit and less loss this project can be used by shareholders so

that they might get sufficient information about the future prices of stock and can buy or sell the stocks accordingly.

The web application can be used by different users who regularly invest in stock market so that they can get an idea of the stock prices and this application can act as their stock market dashboard which will provide them a lot of information of the selected stock ticker from the selected date to end date. This will also give them an option to select the numbers of years of prediction of the particular stock.

Hence our system can be used by many users who regularly invest in stock market and want to get the idea of the future price of the stock.

REFERENCES

- [1] Mehar Vijn, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, “*Stock Closing Price Prediction using Machine Learning Techniques*”, Elsevier, Volume 167, 2020, Pages 599-606, <https://www.sciencedirect.com/science/article/pii/S1877050920307924>
- [2] Subba Rao Polamuri, Kudipudi Srinivas, A.Krishna Mohan, “*A Survey on Stock Market Prediction Using Machine Learning Techniques*”, ResearchGate, May2020,DOI:10.1007/978-981-15-1420-3_101, https://www.researchgate.net/publication/341482418_A_Survey_on_Stock_Market_Prediction_Using_Machine_Learning_Techniques
- [3] Troy J. Strader, John J.Rozycki, Thomas H. Root, Yu-Hsiang Huang, “*Machine learning stock market prediction studies: Review and research directions*”, Volume 28 Issue: 4, 2020, <https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1435&context=jitim>
- [4] Nusrat Rouf, Majid Bashir Malik, TasleemArif, Sparsh Sharma, Saurabh Singh, SatayabrataAich ,andHee-Cheoi Kim, “*Stock Market Prediction Using Machine Learning Techniques: A decade Survey on Methodologies, Recent Developments, Recent Developments, and Future Directions*”, MDPI, Nov 2021, doi.org/10.3390/electronics10212717 <https://www.mdpi.com/2079-9292/10/21/2717>

[5] V Kranthi Sai ReddyVanukuru, “*Stock Market Prediction Using Machine Learning*”, Nov 2018, DOI:10.13140/RG.2.2.12300.77448

https://www.researchgate.net/publication/328930285_Stock_Market_Prediction_Using_Machine_Learning

[6] Jingyi Shen,M.Omair Shafiq, “*Short term stock market price trend prediction using a comprehensive deep learning system*”, Springer Open, Article 66, Aug 2020

<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6>

[7]Indronil Bhattacharjee, Pryonti Bhattacharja, “*Stock Price Prediction: A comparative study between traditional statistical approach and machine learning approach*”, ResearchGate, Dec 2019, DOI:10.1109/EICT48899.2019.9068850

https://www.researchgate.net/publication/337568173_Stock_Price_Prediction_A_Comparative_Study_between_Traditional_Statistical_Approach_and_Machine_Learning_Approach

[8] Xuan Ji, Jiachen Wang, Zhijun Yan, “*A stock Price Prediction method based on deep learning technology*”, Emerald Insight, Volume 5 Issue 1, April 2021

<https://www.emerald.com/insight/content/doi/10.1108/IJCS-05-2020-0012/full/html>

[9] <https://medium.com/@chathmini96/waterfall-vs-agile-methodology-28001a9ca487>

[10] <https://medium.com/analytics-vidhya/illustrative-example-of-principal-component-analysis-pca-vs-linear-discriminant-analysis-lda-is-105c431e8907>

- [11] <https://www.analyticsvidhya.com/blog/2019/01/fundamentals-deep-learning-recurrent-neural-networks-scratch-python/>
- [12] <https://www.analyticsvidhya.com/blog/2019/01/fundamentals-deep-learning-recurrent-neural-networks-scratch-python/>
- [13] <https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9>
- [14] <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>
- [15] https://raw.githubusercontent.com/dataprofessor/s-and-p-500companies/master/data/constituents_symbols.txt

APPENDICES

1. Code

• Linear Regression

```
Linear Regression
```

```
[1] import pandas as pd
import numpy as np
from sklearn import metrics
from matplotlib import pyplot as plt
```

```
[2] from google.colab import files
uploaded_files.upload()
```

Choose Files RELIANCE.csv

- RELIANCE.csv(text/csv) - 408007 bytes, last modified: 11/17/2022 - 100% done

Saving RELIANCE.csv to RELIANCE.csv

```
[3] dataset=pd.read_csv('RELIANCE.csv')
dataset.head()
```


	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-11-17	48.119831	48.189457	47.578289	48.096022	37.453396	21922558.0
1	2000-11-20	48.050205	48.367393	47.818115	47.996052	37.375076	10856730.0
2	2000-11-21	48.390602	48.390602	47.593761	47.825851	37.242550	24245818.0
3	2000-11-22	47.965103	48.197193	47.593761	47.678860	37.128071	7411113.0
4	2000-11-23	47.500927	48.351921	47.230156	48.143040	37.489532	16483956.0

```
[4] dataset['Date']=pd.to_datetime(dataset.Date)
dataset.shape
```

(5494, 7)

```
[5] dataset['Adj Close'].plot(figsize=(16,6))
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fa2a17b2990>



```
[6] dataset.drop('Adj Close',axis=1,inplace=True)
```

```
[8] dataset.head()
```

	Date	Open	High	Low	Close	Volume
0	2000-11-17	48.119831	48.189457	47.578289	48.096022	21922558.0
1	2000-11-20	48.050205	48.367393	47.818115	47.996052	10856730.0
2	2000-11-21	48.390602	48.390602	47.593761	47.825851	24245818.0
3	2000-11-22	47.965103	48.197193	47.593761	47.678860	7411113.0
4	2000-11-23	47.500927	48.351921	47.230156	48.143040	16483956.0

```
[7] dataset.isnull().sum()
```

Date 0
Open 10
High 10
Low 10
Close 10
Volume 10
dtype: int64

```
[11] dataset.isna().any()
```

Date False
Open False
High False
Low False
Close False
Volume False
dtype: bool

```
[12] dataset= dataset.dropna()
```

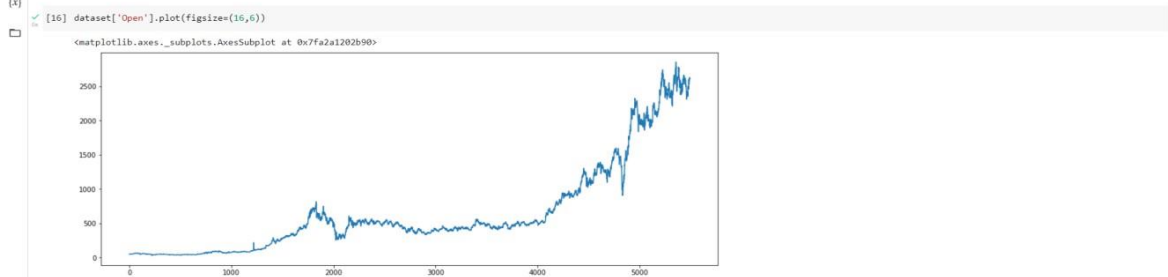
```
[13] dataset.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5484 entries, 0 to 5483
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  --
 0   Date    5484 non-null   datetime64[ns]
 1   Open    5484 non-null   float64
 2   High    5484 non-null   float64
 3   Low     5484 non-null   float64
 4   Close   5484 non-null   float64
 5   Volume  5484 non-null   float64
dtypes: datetime64[ns](1), float64(5)
memory usage: 299.9 KB
```

```
dataset.describe()

```

	Open	High	Low	Close	Volume
count	5484.000000	5484.000000	5484.000000	5484.000000	5.484000e+03
mean	651.517536	659.539088	642.942053	650.912065	1.566375e+07
std	668.352873	675.786272	660.566370	667.835748	1.601396e+07
min	32.477016	34.751492	31.502243	31.966421	0.000000e+00
25%	183.820068	189.237469	183.169987	188.711212	6.259475e+06
50%	459.475076	466.063986	453.080246	459.605896	9.863748e+06
75%	701.104217	711.375626	692.931702	699.742126	1.916798e+07
max	2856.149002	2856.149002	2786.100098	2819.850098	2.918015e+08

```
[15] print(len(dataset))
5484
```



```
[17] X=dataset[['Open','High','Low','Volume']]
Y=dataset['Close']

[18] from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2, random_state=0)
```

```
[19] # X_train.shape
print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
(4387, 4) (1097, 4) (4387,) (1097,)
```

```
[20] from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
lr=LinearRegression()
```

```
[21] lr.fit(X_train,Y_train)
LinearRegression()
```

```
[22] # accuracy score on the training data
lr_train_prediction = lr.predict(X_train)
lr_training_data_accuracy = lr.score(X_train, Y_train)
print('Accuracy score of the training data : ', lr_training_data_accuracy*100)
Accuracy score of the training data : 99.99279335644235
```

```
[23] # accuracy score on the training data
lr_test_prediction = lr.predict(X_test)
lr_testing_data_accuracy = lr.score(X_test, Y_test)
print('Accuracy score of the testing data : ', lr_testing_data_accuracy*100)
Accuracy score of the testing data : 99.99366980915586
```

```
[42] print(lr.coef_)
print(lr.intercept_)
[-5.88329364e-01  8.21985399e-01  7.65307477e-01 -5.12599903e-09]
0.1457279072211577
```

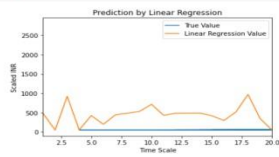
```
[26] plt.plot(lr.coef_)
plt.title('Linear Regression Coefficients')
plt.xlabel('values')
plt.legend()
plt.show()
```



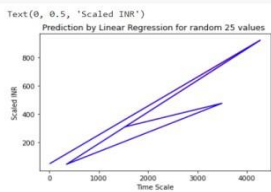
```
[27] print(X_test)
[28] lr_test_prediction.shape
[29] dframe=pd.DataFrame(Y_test,lr_test_prediction)
[30] dframe=pd.DataFrame({'Actual Price':Y_test,'Predicted Price':lr_test_prediction})
```

```
[31] print(dfr)
[32] import math
a = metrics.mean_absolute_error(Y_test,lr_test_prediction)
b = metrics.mean_squared_error(Y_test,lr_test_prediction)
c = math.sqrt(metrics.mean_squared_error(Y_test,lr_test_prediction))
print('Mean Absolute Error:', a)
print('Mean Squared Error:', b)
print('Root Mean Squared Error:', c)
```

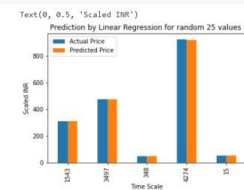
```
[33] x = Y_test
y = lr_test_prediction
plt.ylim(1,20)
plt.xlim(1,20)
plt.plot(x, label='True Value')
plt.plot(y, label='Linear Regression Value')
plt.title('Prediction by Linear Regression')
plt.xlabel('Time Scale')
plt.ylabel('Scaled INR')
plt.legend()
plt.show()
```



```
[40] graph=dfr.head(5)
plt.plot(graph['Actual Price'],color='red')
plt.plot(graph['Predicted Price'],color='blue')
plt.title('Prediction by Linear Regression for random 25 values ')
plt.xlabel('Time Scale')
plt.ylabel('Scaled INR')
```



```
[41] graph=plt(kind='bar')
plt.title('Prediction by Linear Regression for random 25 values ')
plt.xlabel('Time Scale')
plt.ylabel('Scaled INR')
```



• LSTM

```
[1] import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib
from sklearn.preprocessing import MinMaxScaler
from keras.layers import LSTM, Dense, Dropout
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.dates as mdates
from sklearn.preprocessing import MinMaxScaler
from sklearn import linear_model
from keras.models import Sequential
from keras.layers import Dense
import keras.backend as K
from keras.callbacks import EarlyStopping
from keras.optimizers import Adam
from keras.models import load_model
from keras.layers import LSTM
from keras.utils.vis_utils import plot_model
```

```
[2] from google.colab import files
uploaded=files.upload()
Choose File RELIANCE.csv
• RELIANCE.csv(text/csv) - 408007 bytes, last modified: 11/17/2022 - 100% done
Saving RELIANCE.csv to RELIANCE.csv
```

```
[3] df=pd.read_csv('RELIANCE.csv')
df.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-11-17	48.119831	48.189457	47.578289	48.096622	37.453396	21922558.0
1	2000-11-20	48.050205	48.367393	47.818115	47.996052	37.375076	10856730.0
2	2000-11-21	48.390602	48.390602	47.593761	47.825851	37.242550	24245818.0
3	2000-11-22	47.965103	48.197193	47.593761	47.678960	37.128071	7411113.0
4	2000-11-23	47.500927	48.351921	47.230156	48.143040	37.489532	16483956.0

```
[4] df['Date']=pd.to_datetime(df.Date)
df.shape
```

(5494, 7)

```
[5] df.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	5484.000000	5484.000000	5484.000000	5484.000000	5484.000000	5.484000e+03
mean	651.517536	659.539088	642.942093	650.912065	628.037471	1.566375e+07
std	668.302873	676.788272	660.566370	667.830748	670.959128	1.601396e+07
min	32.477016	34.751492	31.502243	31.966421	25.373877	0.000000e+00
25%	183.820068	189.237469	183.169987	188.711212	166.027012	6.259475e+06
50%	459.475876	466.603986	453.080246	459.608956	426.964726	9.863748e+06
75%	701.104217	711.375626	692.931702	699.742126	653.995239	1.916798e+07
max	2856.149902	2856.149902	2786.100098	2819.850098	2811.385742	2.918015e+08

```
[6] print('Dataframe Shape: ', df.shape)
print('Null Value Present: ', df.isnull().values.any())
```

Dataframe Shape: (5494, 7)
Null Value Present: True

```
[7] df.isna().any()
```

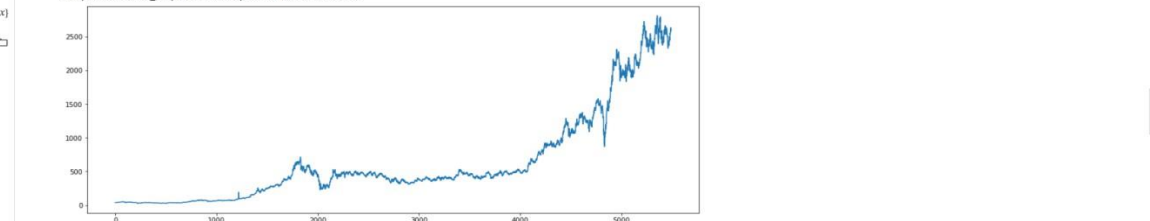
```
Date          False
Open          True
High          True
Low           True
Close         True
Adj Close     True
Volume        True
dtype: bool
```

```
[8] df=df.dropna()
```

```
[9] df.isna().any()
```

```
Date          False
Open          False
High          False
Low           False
Close         False
Adj Close     False
Volume        False
dtype: bool
```

```
[10] df['Adj Close'].plot(figsize=(16,6))
matplotlib.axes._subplots.AxesSubplot at 0x7f6094ebef10:
```



```
[11] output_var = pd.DataFrame(df['Adj Close'])
#Selecting the Features
features = ['Open', 'High', 'Low', 'Volume']
```

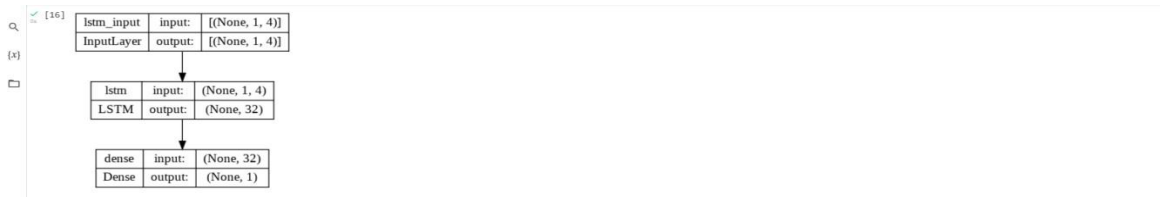
```
[12] #Scaling
scaler = MinMaxScaler()
feature_transform = scaler.fit_transform(df[features])
feature_transform = pd.DataFrame(columns=features, data=feature_transform, index=df.index)
feature_transform.head()

      Open   High    Low  Volume
0  0.005540  0.004763  0.005836  0.075128
1  0.005515  0.004826  0.005823  0.037206
2  0.005636  0.004834  0.005842  0.083090
3  0.005485  0.004766  0.005842  0.025398
4  0.005321  0.004820  0.005710  0.056490

[14] timesplit = TimeSeriesSplit(n_splits=100)
for train_index, test_index in timesplit.split(feature_transform):
    X_train, X_test = feature_transform.iloc[train_index], feature_transform.iloc[train_index+len(test_index)]
    y_train, y_test = output_var.iloc[train_index].values.ravel(), output_var.iloc[train_index+len(test_index)].values.ravel()

[15] trainX = np.array(X_train)
testX = np.array(X_test)
X_train = trainX.reshape(X_train.shape[0], 1, X_train.shape[1])
X_test = testX.reshape(X_test.shape[0], 1, X_test.shape[1])

[16] #Building the LSTM Model
lstm = Sequential()
lstm.add(LSTM(32, input_shape=(1, trainX.shape[1]), activation='relu', return_sequences=False))
lstm.add(Dense(1))
lstm.compile(loss='mean_squared_error', optimizer='adam')
plot_model(lstm, show_shapes=True, show_layer_names=True)
```



```
[17] lstm.summary()

Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
lstm (LSTM)                  (None, 32)                4736
dense (Dense)                (None, 1)                  33
-----
Total params: 4,769
Trainable params: 4,769
Non-trainable params: 0
```

```
[18] history=lstm.fit(X_train, y_train, epochs=50, batch_size=8, verbose=1, shuffle=False)

Epoch 1/50
679/679 [=====] - 2s 2ms/step - loss: 732637.4375
Epoch 2/50
679/679 [=====] - 2s 2ms/step - loss: 672116.5625
Epoch 3/50
679/679 [=====] - 2s 2ms/step - loss: 618537.6250
Epoch 4/50
679/679 [=====] - 2s 2ms/step - loss: 544930.8125
Epoch 5/50
679/679 [=====] - 2s 2ms/step - loss: 479072.3750
Epoch 6/50
679/679 [=====] - 2s 2ms/step - loss: 416240.6875
Epoch 7/50
679/679 [=====] - 2s 2ms/step - loss: 358824.5625
Epoch 8/50
679/679 [=====] - 2s 2ms/step - loss: 308119.4375
Epoch 9/50
679/679 [=====] - 2s 2ms/step - loss: 264241.3750
Epoch 10/50
679/679 [=====] - 2s 2ms/step - loss: 226070.2969
Epoch 11/50
679/679 [=====] - 2s 2ms/step - loss: 191412.8594
Epoch 12/50
679/679 [=====] - 2s 3ms/step - loss: 158223.1562
Epoch 13/50
679/679 [=====] - 2s 2ms/step - loss: 126029.1172
Epoch 14/50
679/679 [=====] - 2s 2ms/step - loss: 96387.0078
Epoch 15/50
679/679 [=====] - 2s 2ms/step - loss: 71118.2891
Epoch 16/50
679/679 [=====] - 2s 2ms/step - loss: 50867.8555
Epoch 17/50
679/679 [=====] - 2s 2ms/step - loss: 35402.5078
Epoch 18/50
```

```
[18] Epoch 44/50
679/679 [=====] - 2s 2ms/step - loss: 394.7025
Epoch 45/50
679/679 [=====] - 2s 2ms/step - loss: 361.3250
Epoch 46/50
679/679 [=====] - 2s 2ms/step - loss: 331.6437
Epoch 47/50
679/679 [=====] - 2s 2ms/step - loss: 305.4872
Epoch 48/50
679/679 [=====] - 2s 2ms/step - loss: 282.6887
Epoch 49/50
679/679 [=====] - 2s 3ms/step - loss: 262.9916
Epoch 50/50
679/679 [=====] - 2s 4ms/step - loss: 246.8348
```

```
[19] y_pred = lstm.predict(X_test)

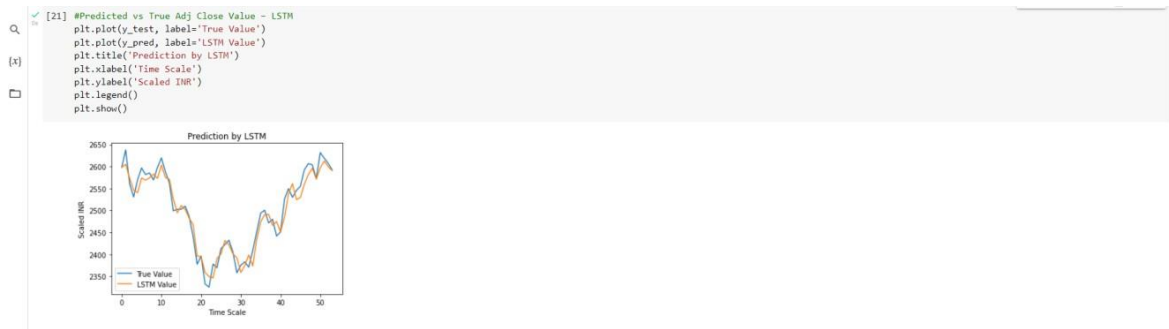
2/2 [=====] - 0s 8ms/step
```

```
[20] # from sklearn.metrics import mean_squared_error
# mean_squared_error(y_test, y_pred)

import math
from sklearn import metrics

a = metrics.mean_absolute_error(y_test, y_pred)
b = metrics.mean_squared_error(y_test, y_pred)
c = math.sqrt(metrics.mean_squared_error(y_test, y_pred))
print('Mean Absolute Error:', a)
print('Mean Squared Error:', b)
print('Root Mean Squared Error:', c)

Mean Absolute Error: 17.471697794278828
Mean Squared Error: 420.3452445930481
Root Mean Squared Error: 20.50232290724756
```



- **Web Application**

```
File Edit Selection View Go Run Terminal Help mp.py - smp - Visual Studio Code
mp.py > ...
1 import streamlit as st
2 import datetime
3 from datetime import date
4
5 import yfinance as yf
6 from prophet import Prophet
7 from prophet.plot import plot_plotly
8 from plotly import graph_objs as go
9
10 import pandas as pd
11
12
13 st.set_page_config(
14     page_title="web application for smp",
15     page_icon="📈",
16     layout="wide",
17     initial_sidebar_state="expanded",
18 )
19
20 #### App title ####
21
22 st.markdown('''
23 # Stock Price Prediction App
24 Gives information of stock price data for different companies !
25 ''')
26 st.write('----')
27
28 #### Sidebar ####
29
30 st.sidebar.subheader('Query parameters')
31 start_date = st.sidebar.date_input("Start date", datetime.date(2015, 1, 1))
32 end_date = st.sidebar.date_input("End date", datetime.date(2023, 5, 2))
33
34 #### Stock selection ####
35
36 stocks = pd.read_csv('https://raw.githubusercontent.com/dataprofessor/s-and-p-500-companies/master/data/constituents_symbols.txt')
37 tickerSymbol = st.sidebar.selectbox('Stock Ticker', stocks) # Select ticker symbol
```

```
File Edit Selection View Go Run Terminal Help mp.py - smp - Visual Studio Code
mp.py > ...
38 tickerData = yf.Ticker(tickerSymbol) # Get ticker data
39
40 string_name = tickerData.info['longName']
41 st.subheader('***%s***' % string_name)
42
43 string_summary = tickerData.info['longBusinessSummary']
44
45 mystyle = '''
46 <style>
47     p {
48         text-align: justify;
49     }
50 </style>
51 '''
52
53 st.markdown(mystyle, unsafe_allow_html=True)
54
55 st.info(string_summary)
56
57 st.write(' ')
58
59 #### Prediction ####
60
61 st.subheader('Forecast for selected Stock Ticker')
62
63 n_years = st.slider('Years of prediction:', 1, 5)
64 period = n_years * 365
65
66
67 @st.cache_data
68 def load_data(ticker):
69     data = yf.download(ticker, start_date, end_date)
70     data.reset_index(inplace=True)
71     return data
72
```

```
File Edit Selection View Go Run Terminal Help mp.py - smp - Visual Studio Code
mp.py > ...
74 data_load_state = st.text('Loading data...')
75 data = load_data(tickerSymbol)
76 data_load_state.text('Data Loaded...')
77
78 st.write(' ')
79
80 st.subheader('Raw data')
81 st.write(data.tail())
82
83 ##### Plot raw data #####
84
85 def plot_raw_data():
86     fig = go.Figure()
87     fig.add_trace(go.Scatter(x=data['Date'], y=data['Open'], name="stock_open"))
88     fig.add_trace(go.Scatter(x=data['Date'], y=data['Close'], name="stock_close"))
89     fig.layout.update(title_text="Time Series data with Rangeslider", xaxis_rangeslider_visible=True)
90     st.plotly_chart(fig)
91
92 plot_raw_data()
93
94 ##### Predict forecast with Prophet. #####
95
96 df_train = data[['Date', 'Close']]
97 df_train = df_train.rename(columns={"Date": "ds", "Close": "y"})
98
99 m = Prophet()
100 m.fit(df_train)
101 future = m.make_future_dataframe(periods=period)
102 forecast = m.predict(future)
103
104 ##### Show and plot forecast #####
105
106 st.subheader('Forecast data')
107 st.write(forecast.tail())
108
109 st.write(' ')
110
```

```
File Edit Selection View Go Run Terminal Help mp.py - smp - Visual Studio Code
mp.py > ...
111 st.write(f'Forecast plot for {n_years} years')
112 fig1 = plot_plotly(m, forecast)
113 st.plotly_chart(fig1)
114
115 st.write("Forecast components")
116 fig2 = m.plot_components(forecast)
117 st.write(fig2)
118
119
120 st.write('---')
121
122 st.write('\n')
123 st.write('\n')
124
125 st.markdown'''
126 **Credits**
127 - App built by -- Aanjaneya Sharma and Sulbha Sharma
128 - Built in "Python" using "streamlit", "yfinance", "prophet", "pandas" and "datetime"
129 '''
130
131 hide_streamlit_style = """
132 <style>
133 #MainMenu {visibility: hidden;}
134 footer {visibility: hidden;}
135 </style>
136 """
137 st.markdown(hide_streamlit_style, unsafe_allow_html=True)
138
```