Stock Market Price Prediction

End Term Report-Research Project(B.Tech)

CS4091-Research Project-VII Semester

submitted by

Subham Sarat Swain
(118CS0221)

under the guidance of

Prof. Pankaj Kumar Sa

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Contents

1 INTRODUCTION

A stock is a financial instrument.It represents ownership in a company or corporation and represents a proportionate claim on the fortune it has.Stocks are also called sharesor a company's equity. They are often called index funds.The stock market serves as the largest financial market in the world, representing trillions of dollars in notional capital.The value of stock markets around the world has grown more than 15-fold over the past 50 years.As the world's largest asset class and market, the stock market attracts investment by savers and other investors, including large institutions and individuals. According to the leading market experts and analysts, global investors have huge cashto invest in the market greater than America's GDP which can shift the market direction.

But despite its enormous influence, the stock market has a less-than-stellar reputation for accuracy, fairness, and transparency but the eyecatch lies majorly in its accuracy. The belief that the stock market is a zero-sum game is an assumption made by tradersin stock market. But it can contribute to the expansion and growth of personal wealthif you are willing to take on the risk of investing. It is a tremendous force multiplier on money, and through the power of compounding,one can earn descent amount of returns on his capital as compared to other asset classes.

In the given project an attempt is made to understand through the process of research, the existing market system and the technology and techniques it uses from a Computer Science perspective, the vulnerabilities and pain points of the current algorithms and techniques used and how we can think of an idea to exploit completely the algorithms which are considered to be used or are used on a very primitive scale as of now to the benefit of financial institutions and investors in general.Predicting the market with the help of machine learning algorithms is the subject of this study.

In addition,the present research and academic papers have been inspected ,to understand the algorithms and methods used for prediction and tried them on the dataset. However, the aforementioned techniques do not perform well in will provide real-life solutions to the problems that stock investors face which in general are more complex.

2 LITERATURE SURVEY

Literature survey is one of the most essential and the most time consuming portion of any research project or any development carried out.It ensures the authenticity of the project to be done,the things that need to be used ,the resources and surroundings along with the quantitative analysis of past and present data that needs to be done.

In the following project on the topic,"Stock Market Price Prediction",an attempt is made to conduct an extensive study of academic research papers by different academic pioneers,along side with the content available regarding the understanding of different concepts and technicalities of the stock market in general.Below mentioned are the major sources from which I have tried to gather information and carry on the research work further on the topic.

From the paper "**Machine Learning applications in financial Markets**"[2] by Prashant Pawar, Btech IIT Bombay,under Prof Saketh Nath, a brief and initial understandingof the financial markets is made which is later extended to the stock market in particular.The pain points and problems that day to day traders face while dealing with different financial instruments in market is well described .How Machine Learning algorithms and technologies are put into use in the real life market scenarios to avoid loss of capital and employment is understood there.The prime focus was to enlighten the use of ML algorithms for stock market. Regression techniques and algorithms of Machine Learning , in particular Linear Regression and Polynomial Regression are extensively used in most financial institutions.

From the book,**Technical analysis of Stock Trends**(Robert D. Eddwards ,John Magge)[1],a brief understanding of regressive analysis of the historical data of the stock marketand the parameters,models and the relationship(especially between the parameters of time and price) is inferred and how its used by the analysts and market experts in general.

From the paper **"Stock Market Price Prediction Using Linear and Polynomial Regression Models**" by Lucas Nunno,Computer Science Department,University of New Mexico[3],an

extensive and broad understanding of the current market institutions and the algorithms used is inferred.The predictive analysis using regression and its importanceto have a quick, easy and low cost prediction for institutions and small as well aslarge retailers in general is explained.The utility and applications of the Linear andPolynomial Regression models against the parameters of the market ,ie, Date,Open Price,Close Price,High Price,Low Price etc is highly explained and a hint is made toinfer the problems these models might posses in drastic situations.

From the paper **"Stock Market Prediction Using Machine Learning Algorithms**",by K. Hiba Sadia, Aditya Sharma, Adarrsh Paul, SarmisthaPadhi, Saurav Sanyal[4],a deep understanding of the actual market scenario is inferred and how Regression models such as Linear Regression and Polynomial Regression , fail to count for the analysis of the real life trading scenarios of stocks.The errors and the shortcomings as well as the primitive assumptions the models make while depicting the prices is clearly understood from the paper and how sometimes such subtle mistakes can lead to huge financial crisis is understood.

From the paper "**STOCK PRICE PREDICTION USING RECURRENT NEURAL NETWORKS**",by Israt Jahan,Software Engineering,North Dakota State University of Agriculture and Applied Science[5],an intuition is developed and a light is casted on howto analyse the problems of the current models of regression and how to overcome such shortcomings using the concept of Neural networks is inferred. The accuracy obtained through the use of Regression models and a prototype program of Neural networks is compared and then it was concluded and understood that in the futurethe Neural networks along with certain other suitable algorithms ,can be used for the real life parameters of the market such as Date,Open Price,Close Price,High,Low etc ,for generating higher accuracy predictions and hence greater profit for the consumers.It is still in the budding phase.

From the blog "**Basic Understanding of LSTM**" by Amey Laddad[6], the problems and working of theRNN models are understood and how the case of vanishing gradients and exploding gradient can cause a serious issue in generating the output is understood. Thereafter how the LSTM framework truly helps in overcoming the issue is seen and the working of LSTM is briefly understood. After that the advantages and applications of LSTM is understood from it.

3 GOALS AND OBJECTIVES

Market fluctuations are highly unstable and this can lead to loss on the side of medium scale investors and even complete loss of capital employed in some cases. Since these businesses make use of investor's money, risk investors must ensure that they are making optimum use of the risks they are taking and are also well aware of the fact that, no business investment will guarantee a guaranteed return.Qualitative and quantitative market intelligence are essential for investors seeking attractive risk-return ratios.

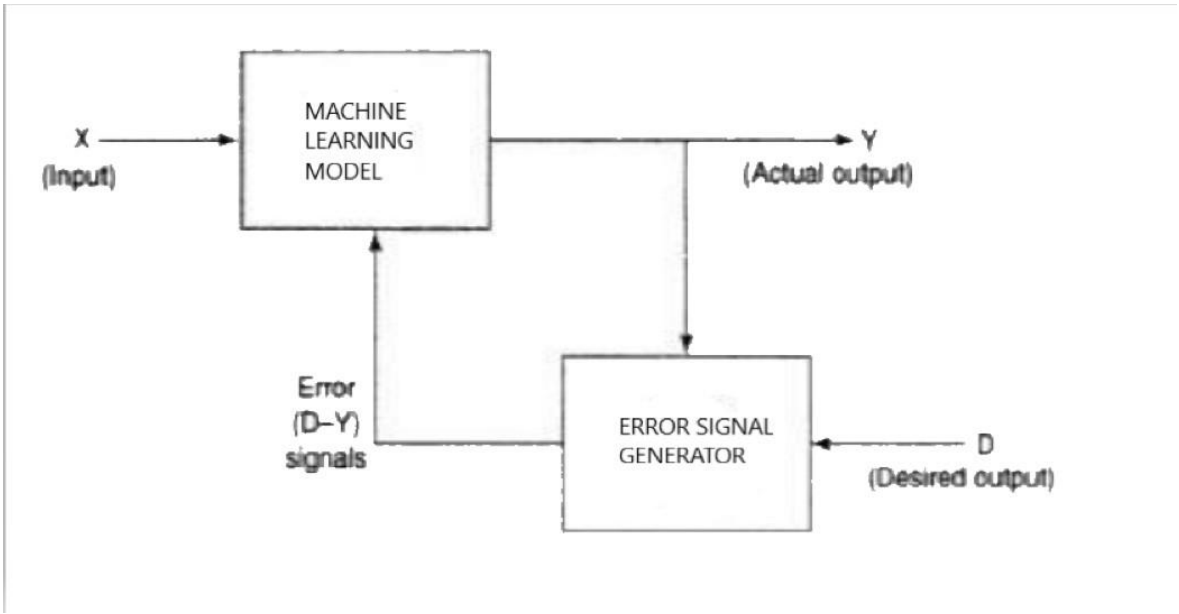
Finding the closest and approximate (though not accurate) method for price prediction is one the primary purpose of the project which can enable market participants to be aware of and interpret that information, even if they fail to utilize itto their advantage with the help of technology.The current problems of the used models (mainly Linear and Polynomial Regression) is to be analysed and then the new model is to be proposed considering the pain points and errors of the existing one so to avoid discrepancy and inconvenience.

So in a nutshell the main objective of the project is to understand the existing algorithms,collect and analyze the historical data, inspect and verify the anomalies,try to remove them and propose a new model on the same data set for an increased accuracy.

4 EXISTING ALGORITHMS USED

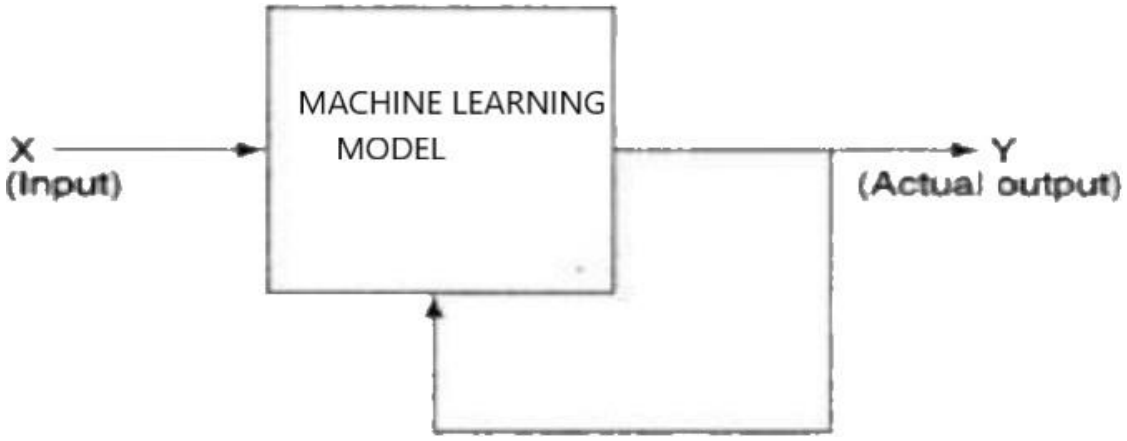
Generally two types of analysis is done for a selection of a stock, Fundamental analysis and Technical analysis. Our prime focus or attention would be technical analysis. Technical analysis do not yield or predict accurate results, investment decisions are madeby analysing all factors along with technical analysis that might affect the prices. Both Supervised and Unsupervised learning are used for the same along with Reinforcement Learning of Machine Learning Approach.

Supervised Learning : supervised machine learning is a type of artificial intelligence that uses datasets labeled with data to train algorithms that can predict and classify complex situations. It involves training models to produce outputs. The outputs are collected and verified by the algorithm. Supervised learning can be separated into two types of problems—classification and regression. A classification algorithm is a process used to identify entities in a given dataset. It uses a series of algorithms to classify the data. A regression algorithm is a type of statistical procedure that plots the relationship between a set of independent variables and the dependent ones. It is commonly used to predict the future revenue of a given business.

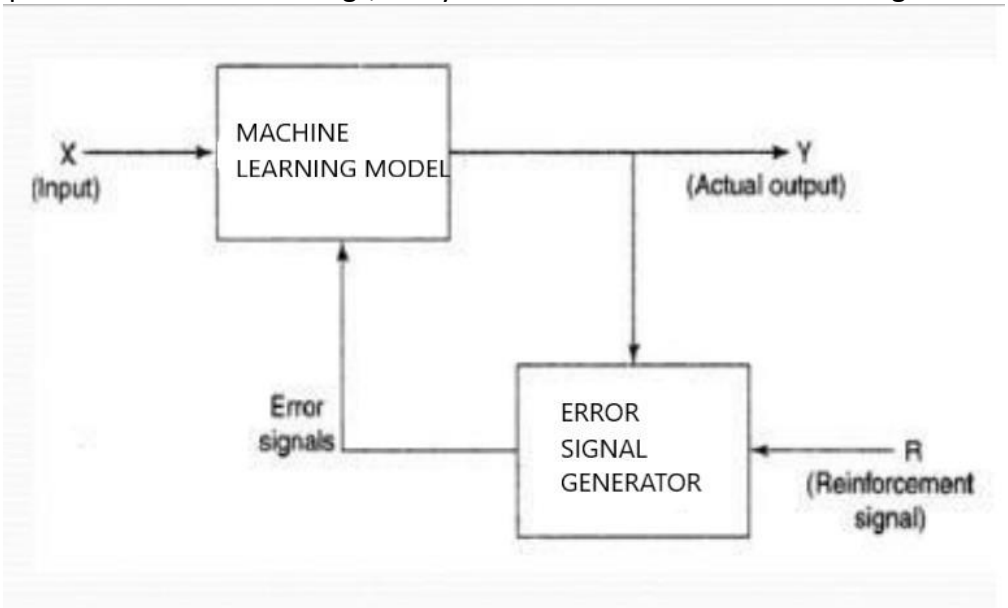


Unsupervised Learning: it is a type of artificial intelligence that tries to identify patterns from the datasets which are neither labelled nor classified. Allows system toidentify patterns from the data on its own.Mainly used to analyze and cluster unlabeled

datasets . It helps to identify co relations between uncorrelated dataset in the stock market. Some algorithms used for same : KNN algorithms , Neural networks etc.



Reinforcement Learning: a method where there is acceptance and rejection . Here the algorithms perceive the environment , takes actions and learns from the success and errors. Actions are taken sequentially. Output depends on the state of current input and next input depends on the output of the previous input. Algorithms used in finance markets: Deep Reinforcement Learning ,Policy Gradient Reinforcement learning etc.



A more intensive and deep approach: Taking an analysis from the history of stock market and the analysts and fund houses which operate in it as well as the researchwork done by many pioneers in the field , I have came to infer that, market predictions bylarge and whole was done and somehow followed now also , through regression models.

Assumptions in regression models : **a)**price movements establish trends in market **b)**stock prices intrinsically contains all information that could affect the prices including fundamental , technical and all psychological factors of the market.

Two types of regression models used extensively: 1) **Linear Regression Model** 2) **Polynomial Regression Model**.A brief discussion about both is done below.

Linear Regression Model:This model is used to establish a relationship between scalar dependent variable and independent variables. It is highly sensitive to the normalisation techniques and parameters. Machine learning libraries like scikit-learnin Python is leveraged here for implementation. Assumptions made during linear regression analysis while applying it for market :**a)**the trends are followed in the market linearly(data points vary linearly) **b)**the variables are completely independent of each other **c)**For any fixed value of independent variables the dependent variables are normally distributed (data is always normalized).Two basic types of Linear Regression techniques used a)Simple Linear Regression(dependent variable is predicted based on a single independent variable) b)Multiple Linear Regression(dependent variable is predicted based on multiple independent variables).Key thing is power of the variables are alwaysin linear fashion(both dependent and independent).

Polynomial Regression Model:It is a type of regression model where the relationship between the dependent variable and the independent variable is made to vary as a **n degree polynomial** unlike linear regression where variation happens only as exponent of 1.There can single independent variable or multiple independent variables like Linear regression .Being a non linear model of data, it is sometimes considered as a special case of multiple linear regression.Normalization is as such not much required in polynomial regression as it tends to sometimes deal with real life data but its a good practice to follow normalization procedure like Linear regression so as to maintain uniformity in range between the values of the data set.Normalization does not change the geometric description of the data used.

SIMPLE LINEAR REGRESSION $y = b_0 + b_1 x_1$

MULTIPLE LINEAR REGRESSION $y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$

POLYNOMIAL REGRESSION $y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$

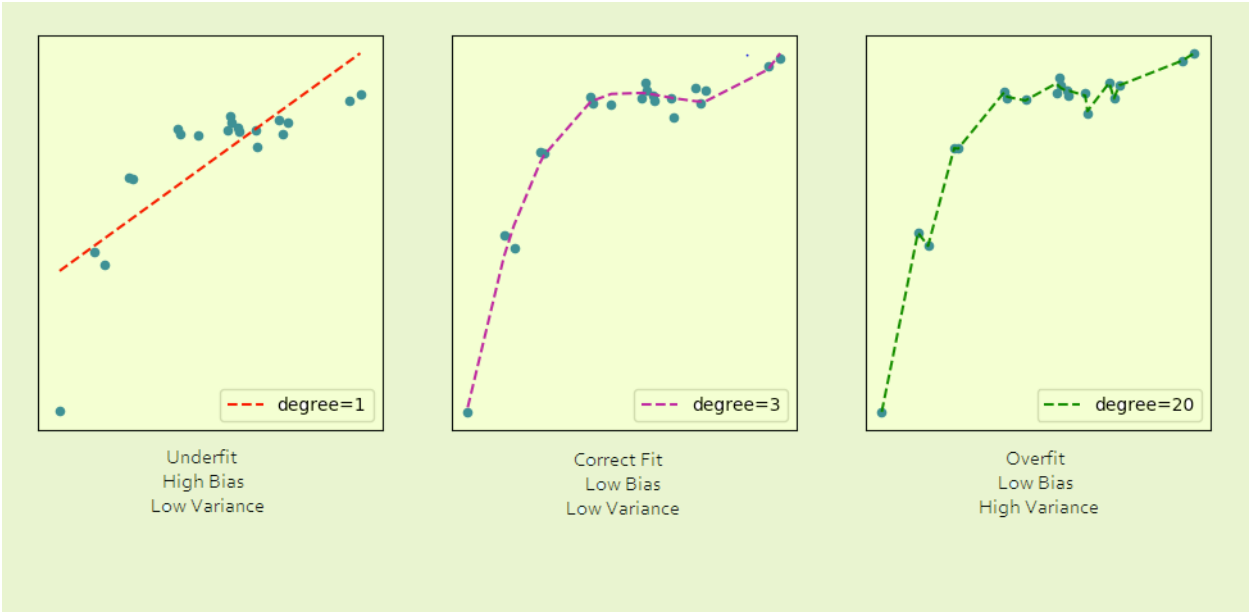
Some Key terms:

Overfitting:Model trained with lot of data,it performs good on training data and poorly on other generalised data.

Underfitting:Model cant capture the trend that is hidden in data,it performs poorly on training data and poorly on other generalised data.

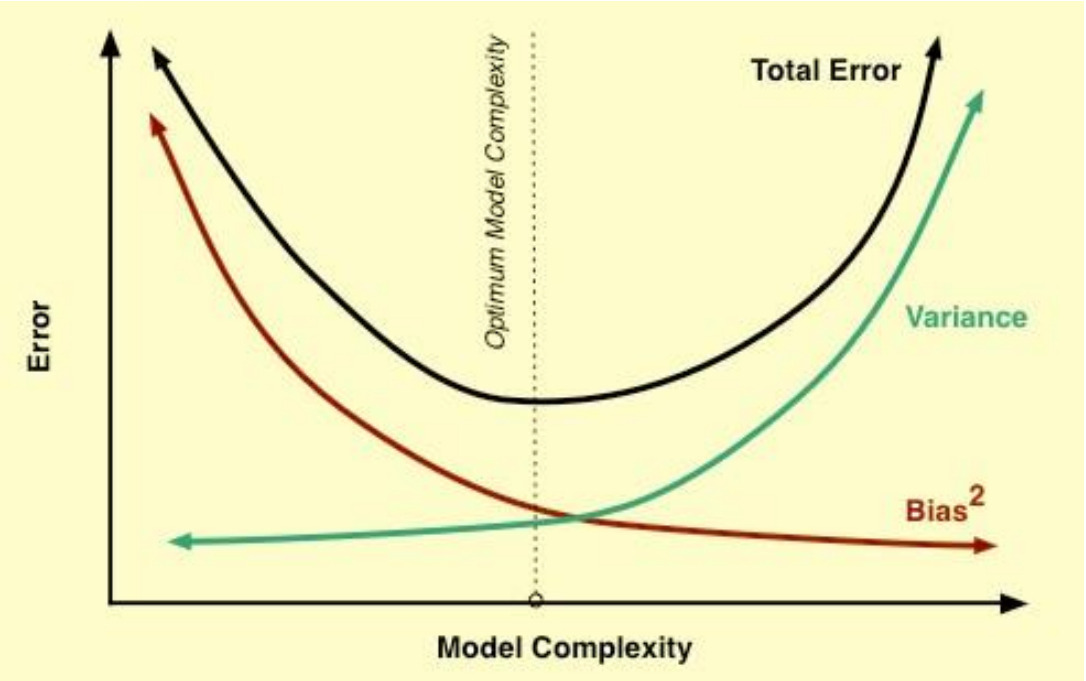
Bias:Assumptions that are made to make a function seem easier to us.

High degree polynomials seem to have low bias and high variance and low degree polynomials seem to have high bias and low variance.An appropriate degree must be choosen to have a bias variance tradeoff and reduction of error.



Bias Variance Tradeoff:Ideally a ML model must possess **low variance** and **low bias**.But its practically impossible to attain.So a trade off is made between them to fit well with appropriate polynomial degree chosen.Below is the diagram for bias variance

trade off and how it affects error.



Here the complexity of the model is plotted along with its error,with the indication of what should be the optimum complexity for which tradeoff should be done.

5 SAMPLE IMPLEMENTATION OF EXISTING ALGORITHM

Platforms used:Jupyter Notebook and Google Collaboratory are used mainly for implementing and executing the python notebooks.Both are open source and free platforms used for data cleaning ,visualization,machine learning etc.

Frameworks used:Pandas,Numpy,sklearn,matplotlib.Pandas is used for manipulation and analysis of data,Numpy is used to deal with arrays,matrix,and complex math functions,sklearn is used for importing and using regression models,matplotlib is used to plot graphs for data and variables used.

Dataset used:The implementation is done using past 26 years data set obtained from BSE India Sensex Archives for the year 1995 to 2021.Some glimpse of the data set used:

Indices : S&P BSE SENSEX
Period : 09-Jan-1995 to 11-May-2021

Date	Open	High	Low	Close
9/01/1995	3,723.09	3,744.84	3,709.02	3,709.02
10/01/1995	3,659.91	3,668.35	3,622.86	3,623.06
11/01/1995	3,588.66	3,602.91	3,574.89	3,600.79
12/01/1995	3,603.22	3,603.22	3,554.79	3,574.90
13/01/1995	3,561.11	3,613.60	3,559.60	3,603.66
16/01/1995	3,625.38	3,672.85	3,625.38	3,653.17
17/01/1995	3,640.51	3,669.10	3,636.30	3,636.30
18/01/1995	3,621.57	3,675.22	3,620.67	3,654.35
19/01/1995	3,663.18	3,675.84	3,656.08	3,656.08
20/01/1995	3,644.00	3,652.69	3,592.95	3,600.53
23/01/1995	3,538.90	3,541.27	3,481.54	3,483.93
24/01/1995	3,415.53	3,417.58	3,394.94	3,411.04
25/01/1995	3,443.73	3,468.09	3,440.01	3,450.68
27/01/1995	3,477.54	3,507.38	3,451.82	3,505.81
30/01/1995	3,493.78	3,579.77	3,493.78	3,575.79
31/01/1995	3,602.52	3,630.50	3,602.52	3,618.54
1/02/1995	3,636.89	3,636.89	3,569.66	3,569.66
2/02/1995	3,546.15	3,601.04	3,546.15	3,599.29

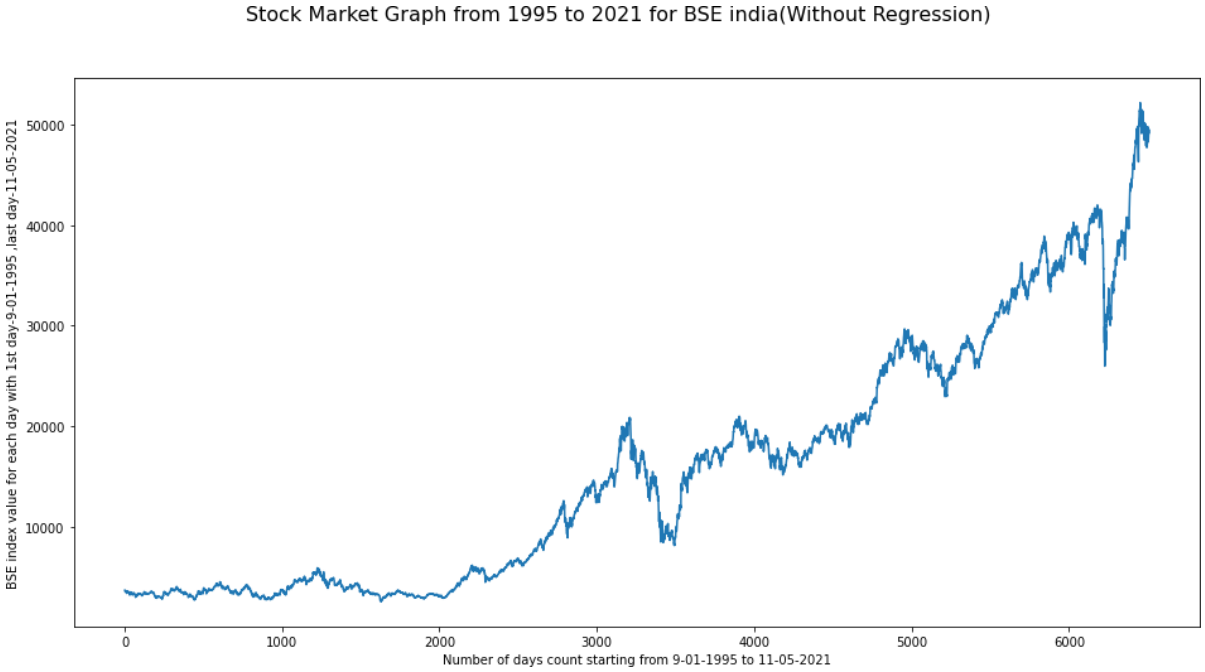
Code of Implementation:: **Linear Regression:**Data Implementation using Linear Regression

Polynomial Regression: Data Implementation using Polynomial Regression Analysis and description of implmentation:A dataset is obtained of the Indian stock market and a analytical study is attempted to do on it through the use of regression.Theentire dataset is divided into 2 parts,training set and testing set.Following a general convention 70 percent of dataset is kept aside for training set and 30 percent for testingdata set.

From the historical stock market data set of 1995-2021, five columns majorly are found;Date,Open Price,High Price,Low Price,Close Price.For keeping our model simple

for analytical purpose and from the inferences drawn from the literature survey,a regression analysis is attempted to be done between date(or in turn number of days)and the closing price index.

A general plot for the market index against the number of days is shown below:

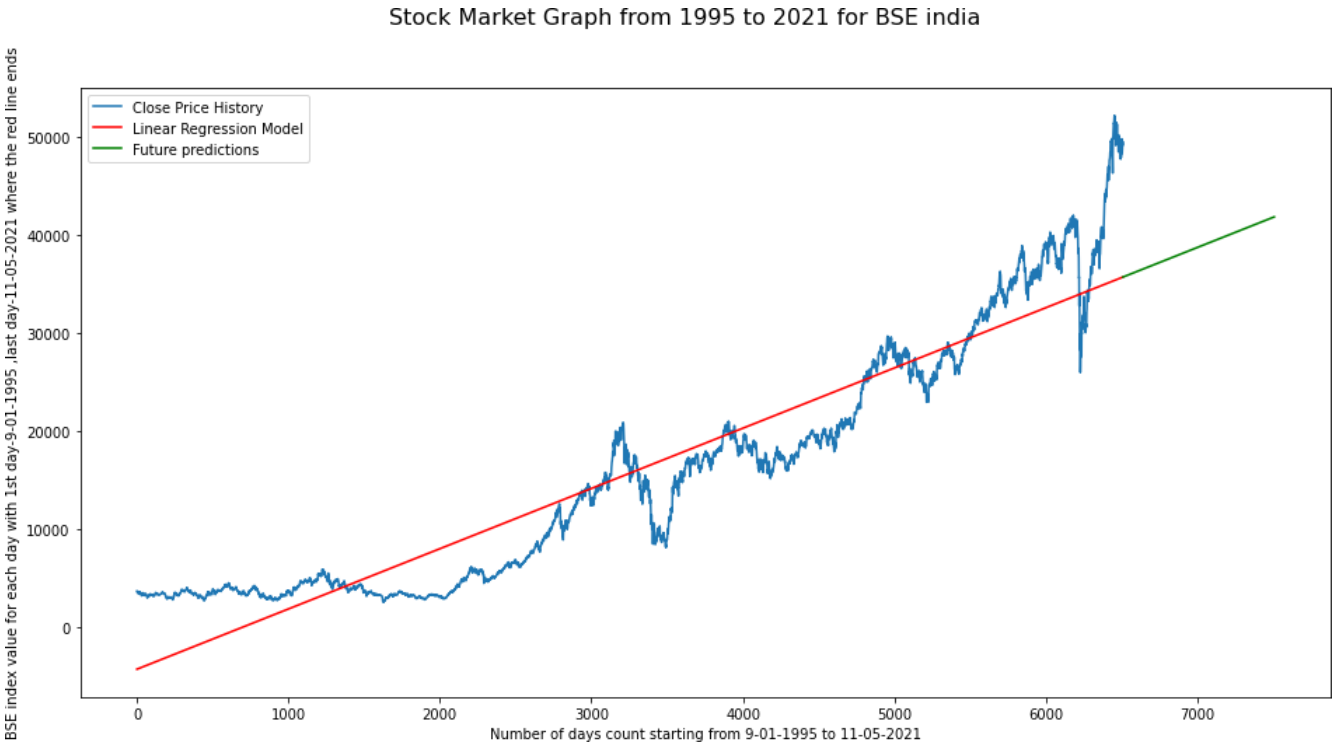


After this I have attempted to normalise the data set.Generally only independent variables are normalised and in case dependent variables are highly,they need to be normalised.Its good practice to normalise the data set as it removes discrepancy and in general the geometrical relationship between the data values are not changed through it.Date value-independent value,Market index-dependent variable;considered.The formula used :

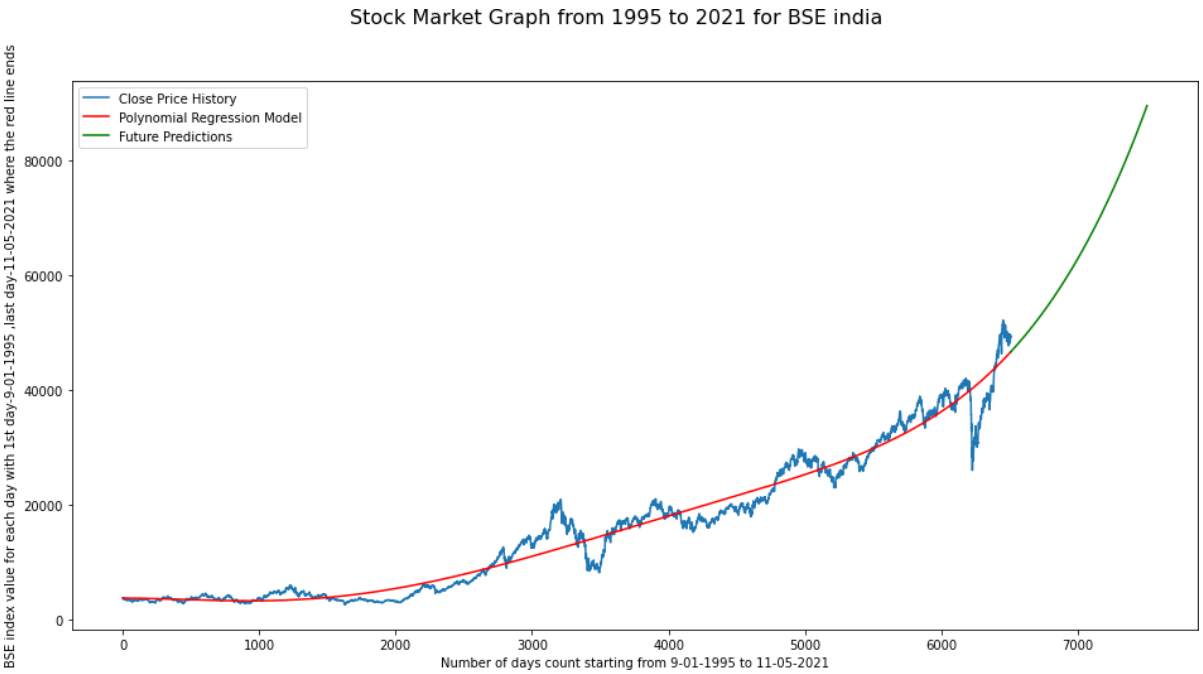
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

After the above procedures,the implementation is attempted using Linear Regression.A simple approach of simple linear regression is attempted as there are two values to be compared,Date and Closing Price(market index).A properfit is tried to achieve using the sklearn linear regression model.Predicted and learned

value of dependent variable is calculated.A new index for predicting the future value of the closing price is chosen and predicted value for dependent value is calculated for new index which is 1000 days ahead of the last date in data set(11,May,2021).The price predicted ahead(2024) will be around 41821.61647049 which can be seen with green line in the graph shown below:



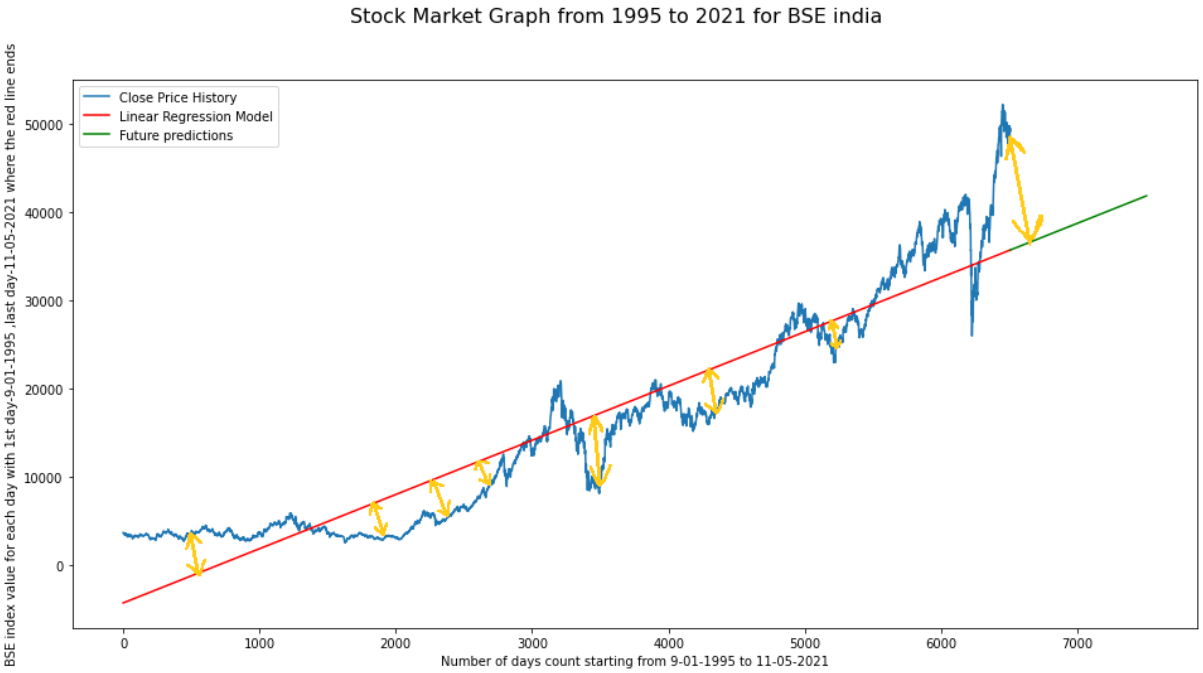
Then the implementation of polynomial regression is attempted.A simpler approach is attempted because of the need to establish a relationship between two variables only(Date and Closing Price).I have attempted to implement this model keeping linear model as its base,(infinite small linear models add up to a polynomial model).A degree 5 polynomial model is marked to train the model and the independent variable is transformedto this higher degree equation.Predicted and learned value of dependent variable is calculated.New index along with an extended transformation of the independent variableis calculated to fit the degree and used for predicting the values 1000 days ahead of thelast date of data set.The price predicted ahead(2024) will be around 89530.95080488 which can be seen with green line in the graph shown below:



This is the overall approach and implementation attempted to have a brief understanding of the existing models used for stock market price prediction in the market.

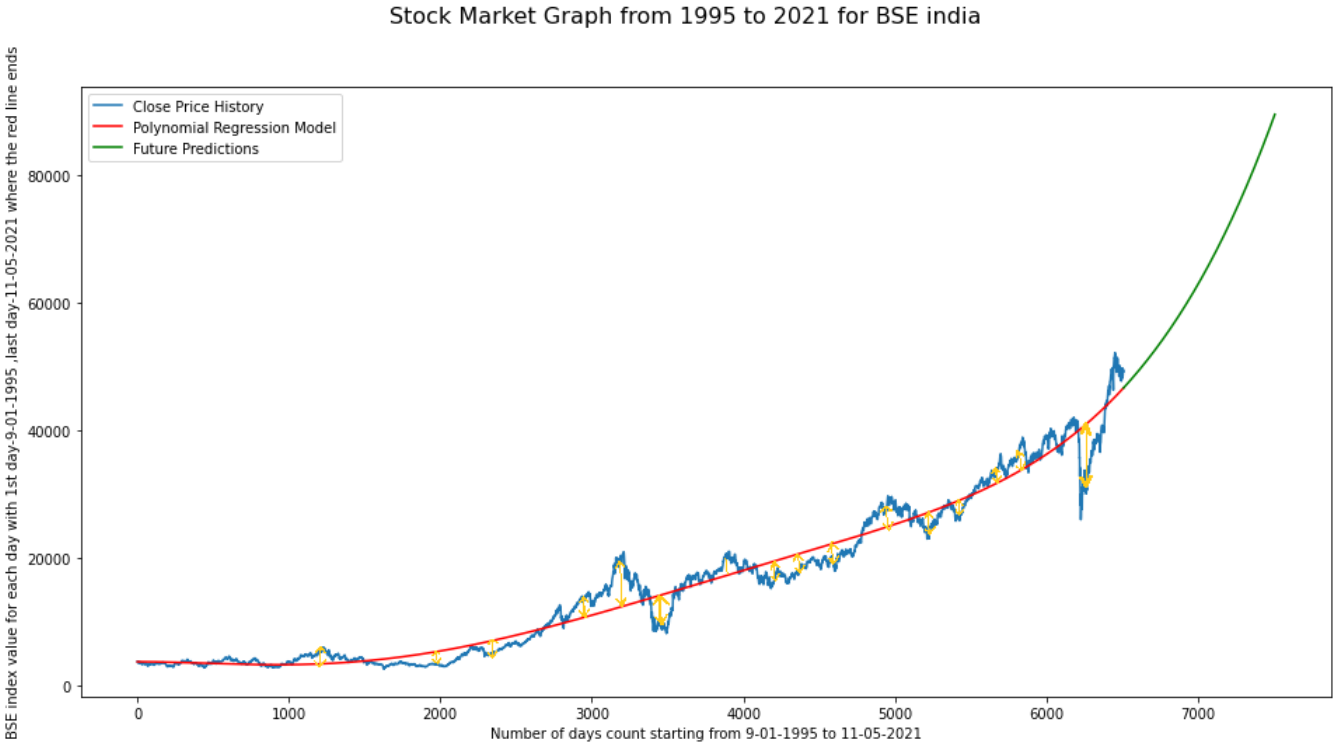
6 RESULT ANALYSIS AND SHORTCOMINGS

First analyzing the Linear regression model from the figure given below we see that the linear regression model and its predictive analysis regarding the predicted value does not seem to fir accurate on the real life data set of the stock market.Althoughit establishes a increasing trend over a long period of time which aligns with the actual graph,but following a linear model for calculating trading decisions during intra-day and short-term trading could incur huge capital loss for investors. The golden lines of theplot clearly indicates how greatly the model varies from the actual graph.This model has high bias,and the variance is also considerably high for certain range of values.Reasons for such discrepancies is that this model assumes independence of attributes and relationship between mean of dependent and independent variables but the mean alone does notprovide complete information of a variable.Also the predicted value for 1000 days aheadis already surpassed by the current market index(as of 2-11-2021,60029.06).This also accounts for error.



Having a distinct observation of the polynomial regression figure below,we see that,the given model performs better than the linear model.Carefully experimenting and having a correct fit with a degree 5 polynomial and trading off between bias and variance,we see that this model fits better to the given data-set.The length of the golden lines

(denoting how much the model's graph vary from the real one) is less which indicates considerably low variance.This model seem to have lower bias than linear model.Still it possess shortcomings.Firstly it can tend to over-fit with a lot of training data and higher degree,secondly ,though the error magnitude is less compared to linear model it still can't be overlooked for certain sections of the graph as it can cause huge losses for short time traders ,lastly the model tends to predict the values in increasing trend always which is not the case in real market that can face dips and lows.This requires search for an increasing accurate model which can work both in short term and long term in predicting the trends along the actual graph and can also reduce the error while depicting a model to fit for the existing graph or data.



This is the overall result and graph analysis attempted for the already existing and implemented algorithms and their shortcomings and need for better models thoroughly discussed.

7 PROPOSED ALGORITHM FOR BETTER SOLUTION

From the analysis and discussions done in the previous section,we got to infer that the regression models tend to cause faults in predicting the values of the stock market index(closing values) in terms of accuracy and in alignment with the real time data.This is because the variables are assumed to have linear dependencies which is obvious for linear regression model,and for polynomial regression model ,it is built on the top of infinite small amount of linear regression models that are made to vary in a continuous manner for a higher degree.So for data with non-linear dependencies which is the case of the real time stock market data,**Neural Networks** is one of the major choice of algorithm,which is inferred from the literature survey and can prove to yield high accuracy results.

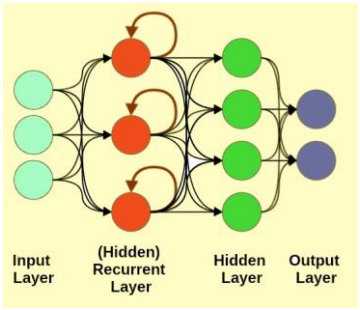
Neural Networks:It involves attempts to learn a function such that mapping is done from the input features and attributes to the output predictions.It has network of neurons which are weighted sum of inputs.Outputs of neurons are sub-sequentially merged and combined with the activation functions that results non linearity inside the system.Backpropagation algorithm is used to increase the accuracy of the neural networks which fine tunes the weights of a neural network byconsidering the error rates which has occured during the execution of the previous iteration.Back propagation is very rapid and efficient method to execute and improve accuracy.

Three types of neural networks are present:**a)**Artifical Neural Networks**b)**Convolutional Neural Networks**c)**Recurrent Neural Networks.

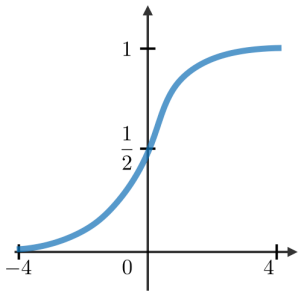
From the literature survey,I have inferred and analyzed to use **Recurrent Neural Networks** for continuation of the research project.The main centre of attention is given to RNN for the following project because of a number of reasons.The reasons are:a)RNN can handle large length of arbritary input and output data(which is the case of stock

market in our case)while CNN is especially ideal for image analysis and processing. b)RNN when compared to ANN,captures sequential information present in the input data at every neuron ,while ANN only works on the forward feeding mechanism,which makes RNN more apt for stock market.

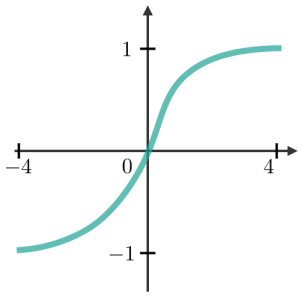
Diagram for RNN:



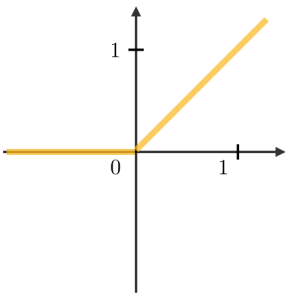
Some of the mathematical function used in RNN for activation are a)Sigmoid:



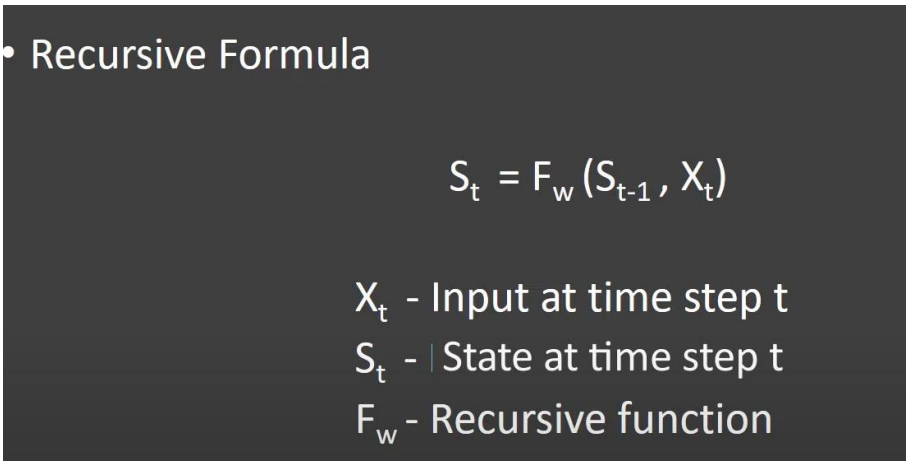
b)tanh

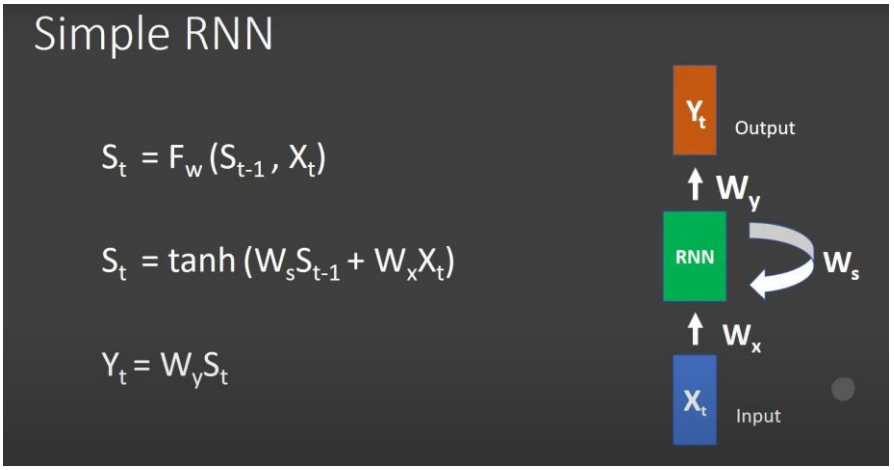


c)relu

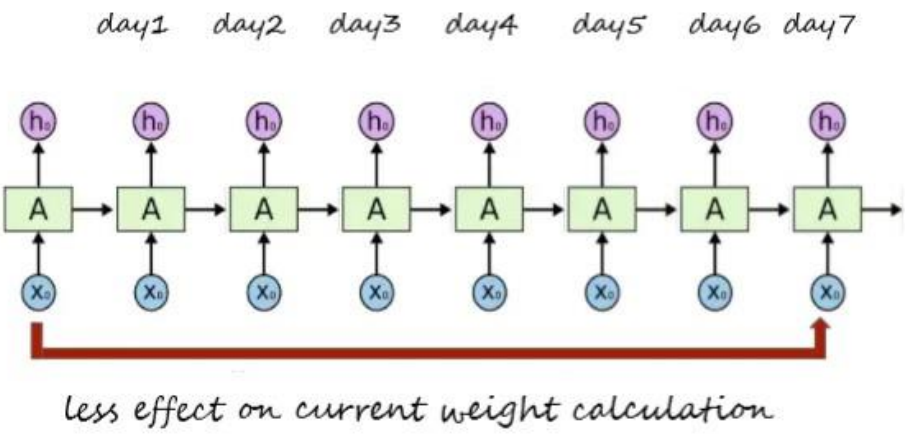


Mathematical Formulas used in Simple RNN are:





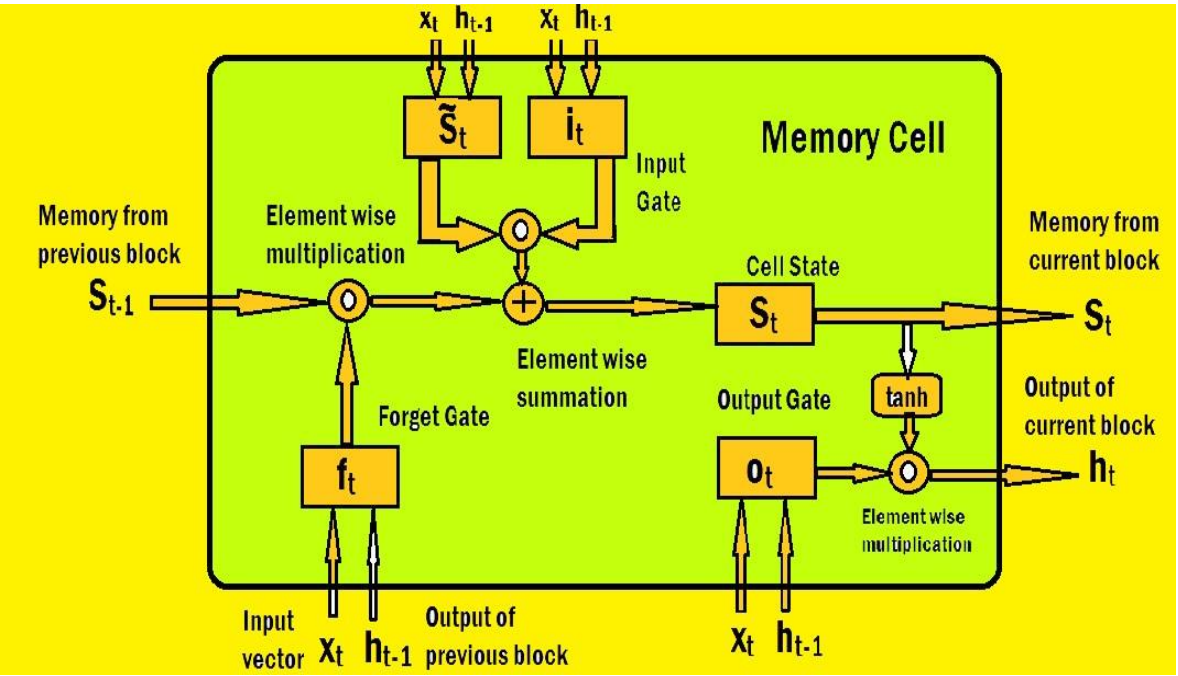
If we observe very closely for higher number or deep RNN network ,backpropagation technique can cause problem as when it would go back to each layer or step and multiply the gradient,for relative lower value of gradient the first layers (which are connected to input) will not have any weight changes and hence it can cause an inaccuracy model and from the stock market point of view the prediction would be inappropriate.Below image depicts same in diagram where hi=hidden layer outputs:



The above inaccuracy can be further reduced by using LSTM along with RNN ,which are generally used for dealing with time lags in time series data frame(in our case stock market price data).So in our case the model would fit appropriately as supposed. Theentire RNN model along with the LSTM with its implementation.

Stacked LSTM implementation is the implementation model which can overcome the shortcoming of the RNN . The implementation of LSTM (Long Short Term Memory) is one of the key implementations of the model with RNN.RNN proves to be very charismatic in the implementation of the time series data but they suffer serious issues the number of layers of the RNN increases or the number of neuron carrying the weight increases. This increases the loss of gradient, i.e., a case of **vanishing gradient** issue arises. In the vanishing gradient issue the RNN model has low propagation of the gradient which is required for adjusting the weights during backpropagation.So by using Long short Term Memory arrangement such issue is resolved which solves the problem of vanishing gradient and gives us much better accuracy than RNN.Also the case may occur of exploding gradients where the algorithm assigns excessively high importance to a weight .For both the cases LSTM arrangement works well.

Core LSTM idea: A memory block which can maintain its state over time and has explicit memory and gating units that determines the amount of data which can flow in and out of the memory.



In the above diagram a basic structure for LSTM is described. The unit involves Cell State Vector which forgets the data through one of the gates and takes input of the data through another gate.. Gates are restricted by the time steps which come before and the current input and sometimes by the cell state vector. 3 gates i) **Forget gate(**controls what information are thrown out of memory) ii)**Input gate**(regulates what are the input to be taken into the memory) iii)**Output gate** (regulates what are the output to be shown through the cell).The diagram above represents the overall working of the LSTM model in a nutshell.

Stacked LSTM: It is an architecture where there are multiple LSTM layers present rather than a single LSTM layer. The LSTM layer above provides the LSTM layer below a sequential layer of outputs rather than a single output which makes it very useful in the application of the time – series data where there is a dependence of adjacent data on each other, i.e., the data values are dependent on each other and also on the time . The main crucial advantage of using stacked LSTM than the feedforward layer of the RNN between the feature inputs is that a feed forward layer does not receive feedback from future timestep and thus cannot count for certain patterns but a LSTM layer is more complex in nature and hence much more complex patterns can be described by the inclusion of the LSTM layer .

Below is the sample implementation where a RNN along with LSTM network is used to predict the prices.Rather than using the entire BSE India index price here I have chosen a single stock (Apple Inc) as a sample data with prices from 2015 to 2020.

Steps followed:

1. Collection of the stock data.
2. Pre-process the data -Train and Test.
3. Create a Stacked LSTM Model.
4. Predict the test data and plot the output.
5. Predict the future 30 days and plot the output.

Below is the system algorithmic implementation of the above steps with code at each steps.:

Platform used: Google Collaboratory.

Libraries Used: TensorFlow,Keras,Numpy,Matplotlib,Scikitlearn etc.

Steps and code:

Step1)Data collection:

A small view of the data which is used in the following implementation.

		symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
2	0	AAPL	2015-05-2	132.045	132.26	130.05	130.34	45833246	121.6826	121.8807	119.8441	120.1114	45833246	0	1
3	1	AAPL	2015-05-2	131.78	131.95	131.1	131.86	30733309	121.4384	121.595	120.8117	121.5121	30733309	0	1
4	2	AAPL	2015-05-2	130.28	131.45	129.9	131.23	50884452	120.0561	121.1343	119.7059	120.9315	50884452	0	1
5	3	AAPL	2015-06-0	130.535	131.39	130.05	131.2	32112797	120.2911	121.079	119.8441	120.9039	32112797	0	1
6	4	AAPL	2015-06-0	129.96	130.655	129.32	129.86	33667627	119.7612	120.4016	119.1714	119.669	33667627	0	1
7	5	AAPL	2015-06-0	130.12	130.94	129.9	130.66	30983542	119.9086	120.6643	119.7059	120.4062	30983542	0	1
8	6	AAPL	2015-06-0	129.36	130.58	128.91	129.58	38450118	119.2083	120.3325	118.7936	119.411	38450118	0	1
9	7	AAPL	2015-06-0	128.65	129.69	128.36	129.5	35626800	118.554	119.5124	118.2867	119.3373	35626800	0	1
10	8	AAPL	2015-06-0	127.8	129.21	126.83	128.9	52674786	117.7707	119.07	116.8768	118.7844	52674786	0	1
11	9	AAPL	2015-06-0	127.42	128.08	125.62	126.7	56075420	117.4205	118.0287	115.7618	116.757	56075420	0	1
12	10	AAPL	2015-06-1	128.88	129.34	127.85	127.92	39087250	118.7659	119.1898	117.8168	117.8813	39087250	0	1
13	11	AAPL	2015-06-1	128.59	130.18	128.475	129.18	35390887	118.4987	119.9639	118.3927	119.0424	35390887	0	1
14	12	AAPL	2015-06-1	127.17	128.33	127.11	128.185	36886246	117.1901	118.2591	117.1348	118.1255	36886246	0	1
15	13	AAPL	2015-06-1	126.92	127.24	125.71	126.1	43988946	116.9598	117.2546	115.8447	116.2041	43988946	0	1
16	14	AAPL	2015-06-1	127.6	127.85	126.37	127.03	31494131	117.5864	117.8168	116.4529	117.0611	31494131	0	1
17	15	AAPL	2015-06-1	127.3	127.88	126.74	127.72	32918071	117.3099	117.8444	116.7939	117.697	32918071	0	1
18	16	AAPL	2015-06-1	127.88	128.31	127.22	127.23	35407220	117.8444	118.2407	117.2362	117.2454	35407220	0	1
19	17	AAPL	2015-06-1	126.6	127.82	126.4	127.71	54716887	116.6649	117.7891	116.4806	117.6878	54716887	0	1
20	18	AAPL	2015-06-2	127.61	128.06	127.08	127.49	34039345	117.5956	118.0103	117.1072	117.485	34039345	0	1
21	19	AAPL	2015-06-2	127.03	127.61	126.8792	127.48	30268863	117.0611	117.5956	116.9222	117.4758	30268863	0	1
22	20	AAPL	2015-06-2	128.11	129.8	127.12	127.21	55280855	118.0564	119.6137	117.1441	117.227	55280855	0	1
23	21	AAPL	2015-06-2	127.5	129.2	127.5	128.86	31938100	117.4942	119.0608	117.4942	118.7475	31938100	0	1
24	22	AAPL	2015-06-2	126.75	127.99	126.51	127.67	44066841	116.8031	117.9458	116.5819	117.6509	44066841	0	1
25	23	AAPL	2015-06-2	124.53	126.47	124.48	125.46	49161427	114.7573	116.5451	114.7112	115.6143	49161427	0	1
26	24	AAPL	2015-06-3	125.425	126.12	124.86	125.57	44370682	115.5821	116.2225	115.0614	115.7157	44370682	0	1
27	25	AAPL	2015-07-0	126.6	126.94	125.99	126.9	30238811	116.6649	116.9782	116.1027	116.9413	30238811	0	1

The data displayed in the image above is from 02-05-2015 to 02-05-2020.The raw data set has 1259 rows with respective columns for each rows. Here for a preliminary implementation the closing column and the date column are chosen to be implemented for our model. It is collected from Yahoo Finance.

Below is the code to upload the dataset into the drive and assign a variable to it so that the csv can be accessed and used it for future .

```
import pandas as pand
from google.colab import files

import io

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))

df = pand.read_csv(io.StringIO(uploaded['AAPL.csv'].decode('utf-8')))
```

Then the variable containing the closing column is extracted.

```
dataframe=df.reset_index()['close']
```

Step2) LSTM can yield wrong results if scale of data is not aligned perfectly,so we use normalization to scale down the values to a similar range of values.

Doing Normalization by MinMaxScaler.

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
dataframe=scaler.fit_transform(np.array(dataframe).reshape(-1,1))
```

Here the scaling is done between 0 to 1.

Step 3) Train and test split of the data is done in the ratio of 0.65:0.35 ie 65 percent of the data set is Training data and 35 percent of the data is testing data. The code for the same is.

```
##dataset divided into train test split
tr_sz=int(len(dataframe)*0.65)
tst_sz=len(dataframe)-tr_sz
tr_dt,tst_dt=dataframe[0:tr_sz,:],dataframe[tr_sz:len(dataframe),:1]
```

Step 4) Data Preprocessing is done with the help of introduction of the timesteps. The initial timestep taken is 1 here (initialized).But after that it is increased to 100.

```
import numpy
# creating dataset matrix .
def create_dataset(dataset, timeandstep=1):
    xdata, ydata = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]   ###i=0, 0,1,2,3-----99   100
        xdata.append(a)
        ydata.append(dataset[i + time_step, 0])
    return numpy.array(xdata), numpy.array(ydata)
```

```
# reshaping into different time steps.
timeandstep = 100
X_train, y_train = create_dataset(tr_dt, timeandstep)
X_test, ytest = create_dataset(tst_dt, timeandstep)
```

99 elements goes into the independent variable and the 100th element goes into the dependent variable.

Step 5) Before training the LSTM model we are here reshaping the X_train and X_test into the required dimensions.Here the number of dimensions taken is 3.

```
# [samplevlaues,timeandstep,features] reshaping to train the long short term memory model
X_train =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)
```

The reason for doing this transformation is that we are going to give this thing into the LSTM model as input.

Step 6) Create a stacked LSTM model

```
### Create the Stacked LSTM model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
```

The model used here is **sequential model** and the layers are going to be dense(more than 3 layers are going to be used here).

Step7) Parametrizing the LSTM model:

```
mdl=Sequential()
mdl.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
mdl.add(LSTM(50,return_sequences=True))
mdl.add(LSTM(50))
mdl.add(Dense(1))
mdl.compile(loss='mean_squared_error',optimizer='adam')
```

3 layers of LSTM is added here.

Step8) Fitting the data in X_train,Y_train with epoch =100.

```
mdl.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=100,batch_size=64,verbose=1
)
```

Step 9)Do the prediction for both X_train and X_test as performance metric needed to be done.

```
### prediction and performance measurements
tr_pr=mdl.predict(X_train)
tst_pr=mdl.predict(X_test)
```

After this do the reverse scaling to get the value in the original range:

```
##Taking back to original scaling
tr_pr=scaler.inverse_transform(tr_pr)
tst_pr=scaler.inverse_transform(tst_pr)
```

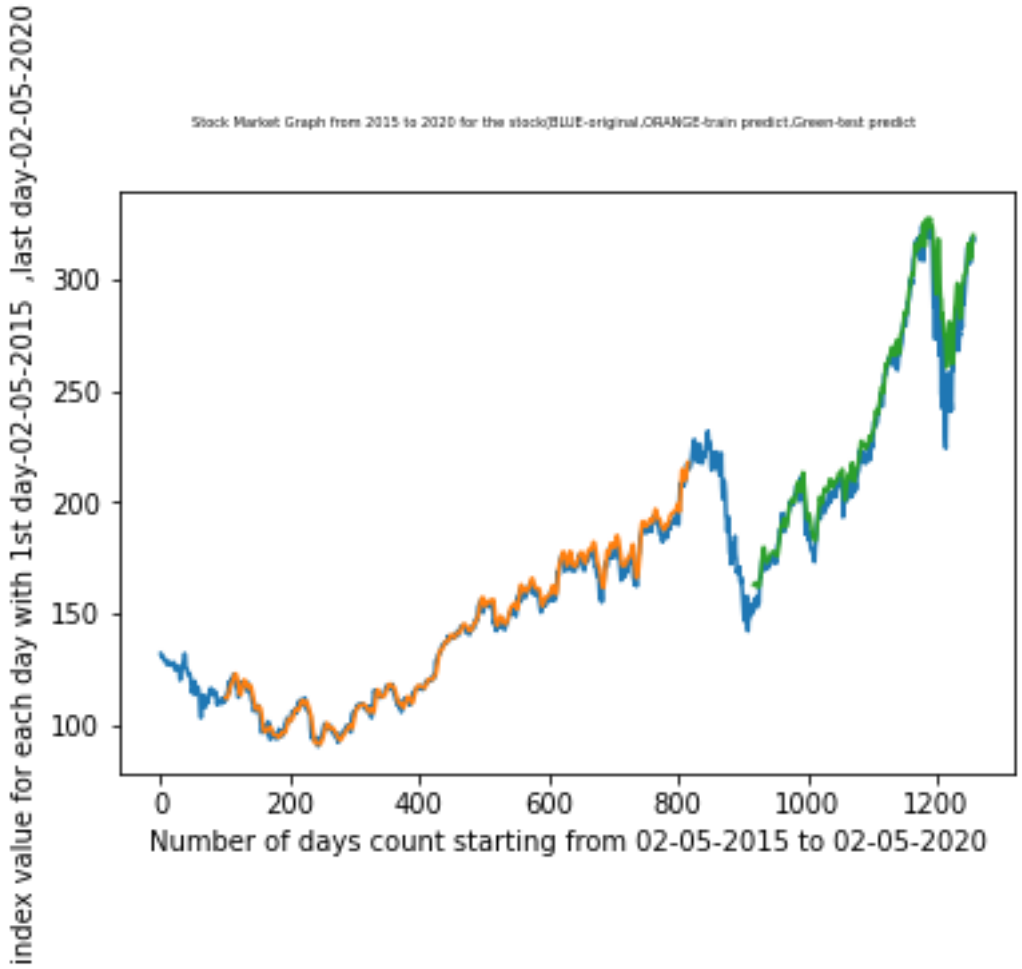
Step 10) Calculate RMSE(root mean square performance metric)

```
### root mean square error performance
import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,tr_pr))

### root mean square error for test data
math.sqrt(mean_squared_error(ytest,tst_pr))
```

Step11) Predicting the test data and plotting the output

```
### Plotting
# train predictions is shifted back for plotting
lk_bk=100
tr_pr_plot = numpy.empty_like(dataframe)
tr_pr_plot[:, :] = np.nan
tr_pr_plot[lk_bk:len(tr_pr)+lk_bk, :] = tr_pr
# test predictions is shifted back for plotting
tst_pr_plot = numpy.empty_like(dataframe)
tst_pr_plot[:, :] = numpy.nan
tst_pr_plot[len(tr_pr)+(lk_bk*2)+1:len(dataframe)-1, :] = tst_pr
# baseline and predicted values plotted
plt.plot(scaler.inverse_transform(dataframe))
plt.plot(tr_pr_plot)
plt.plot(tst_pr_plot)
plt.show()
```



Green colour is actually the test data here(predicted).

Orange colour is the train data here.

Step12)Reshaping and Predicting the output for the next 30 days.

```
x_in=tst_dt[341:].reshape(1,-1)
x_in.shape
```

Step 13)Predict from 1158column to future because the 100 timesteps are taken to create the model.

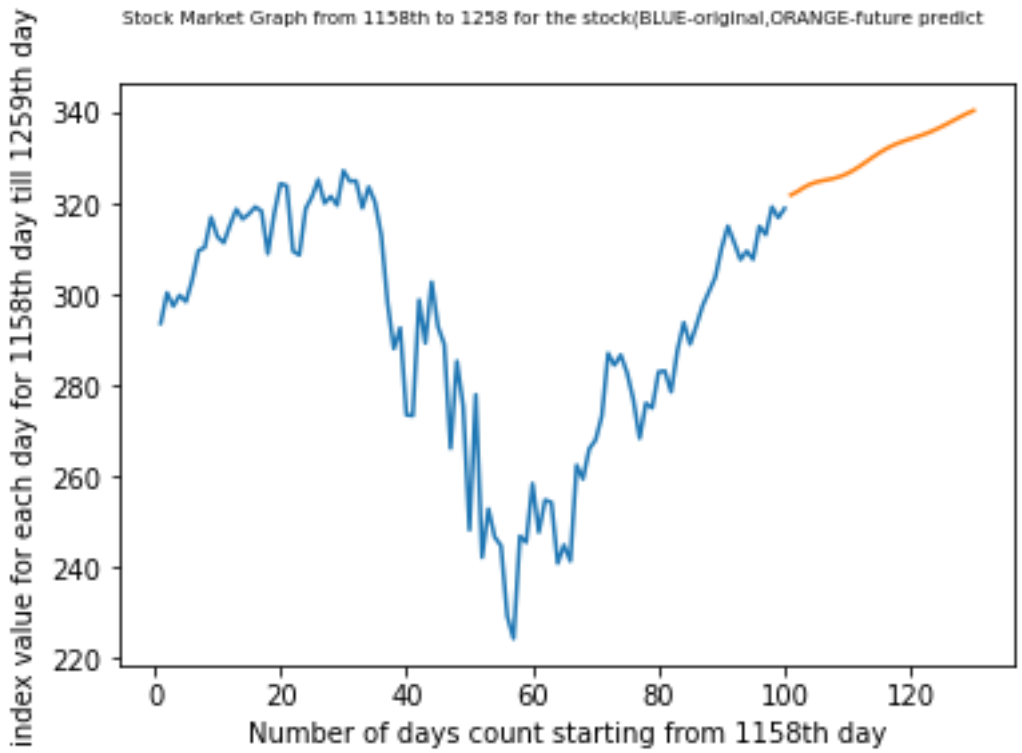
```
# prediction for future days in array format
from numpy import array

fin_out=[]
stepstoback=100
i=0
while(i<30):

    if(len(tmp_in)>100):
        #print(tmp_in)
        x_in=np.array(tmp_in[1:])
        print("{} day input {}".format(i,x_in))
        x_in=x_in.reshape(1,-1)
        x_in = x_in.reshape((1, stepstoback, 1))
        #print(x_in)
        yhat = mdl.predict(x_in, verbose=0)
        print("{} day output {}".format(i,yhat))
        tmp_in.extend(yhat[0].tolist())
        tmp_in=tmp_in[1:]
        #print(tmp_in)
        fin_out.extend(yhat.tolist())
        i=i+1
    else:
        x_in = x_in.reshape((1, stepstoback,1))
        yhat = mdl.predict(x_in, verbose=0)
        print(yhat[0])
        tmp_in.extend(yhat[0].tolist())
        print(len(tmp_in))
        fin_out.extend(yhat.tolist())
        i=i+1


print(fin_out)
```

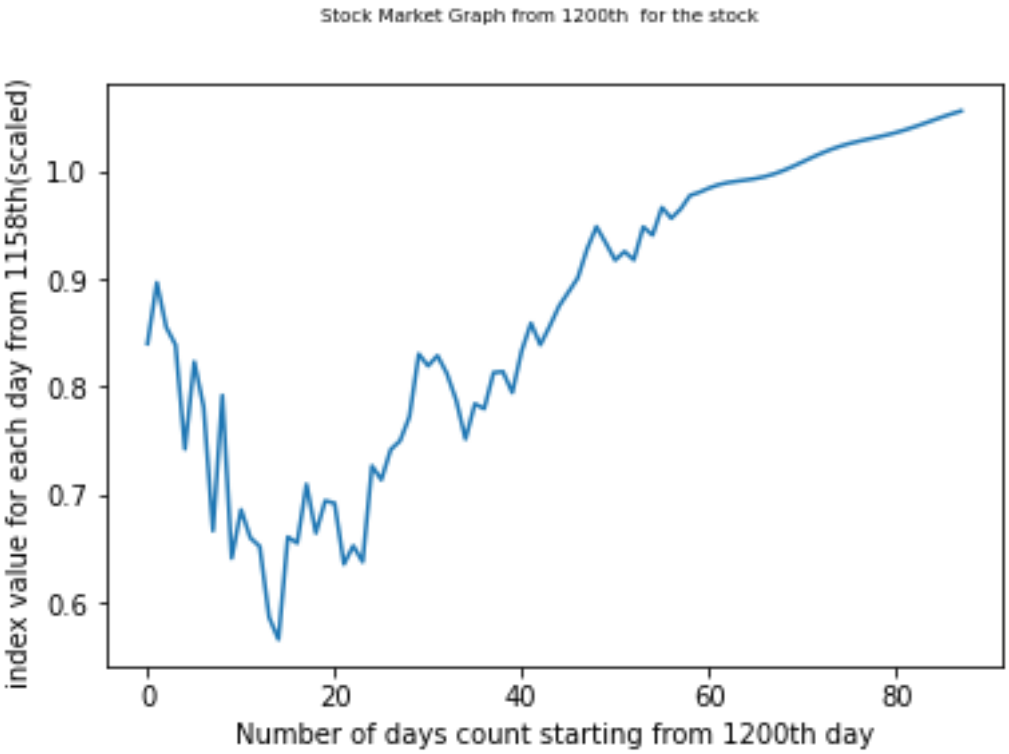
```
plt.plot(d_new,scaler.inverse_transform(dataframe[1158:]))
plt.plot(d_pred,scaler.inverse_transform(fin_out))
```

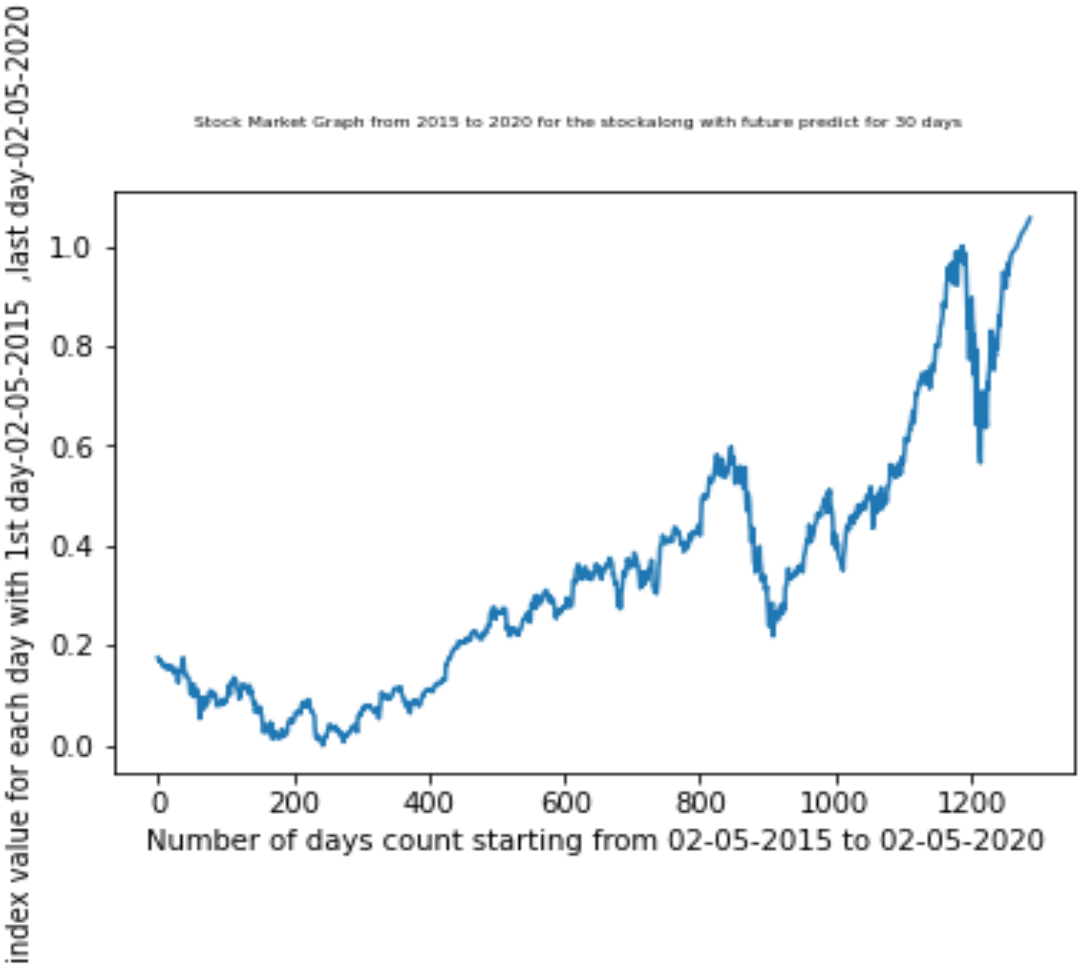


Orange curve indicates the new 30 days output ahead of future.

Step 14)Visualising the combined and total output

```
dataframe3=dataframe.tolist()
dataframe3.extend(fin_out)
plt.plot(dataframe3[1200:])
dataframe3=scaler.inverse_transform(dataframe3).tolist()
plt.plot(dataframe3)
```


Stock Market Graph from 1200th for the stock
index value for each day from 1158th(scaled)
1.0
0.9
0.8
0.7
0.6
0
20
40
60
80
Number of days count starting from 1200th day



As we can see the output gives the close to accurate values for the stock prices as when compared to the original data.LSTM model seem to have performed quite well. Still it has certain shortcomings .

The major shortcoming of our model is that we are considering only date and closing price to predict the future prices but in reality a lot of factors such as volume of stocks ,day high ,day low and many features together come into play for predicting the prices or determining the prices. The LSTM model proposed is relatively simple in nature .Moreover the model is unable to count financial crash in the market which may occur due to a variety of reasons such as economy ,government ,pandemic etc.

Future work: The future research work includes on top of the model proposed,the collection of more data,feature expansion as required,feature selection,choosing high weighted features,performing dimensionality reduction (PCA),model the data to a time series data,process it and then feed it to the exisiting LSTM.This will generate a

more general and multivariate dependency of the future stock prices on the high weighted features.Also after it we would try a variety of other models like Random forest ,SVM ,ANN,CNN etc and in end compare the performance of all on a single data.

8 CONCLUSION

The given project,"Stock market Price Prediction" lays the base for meticulous use of machine learning approach in solving or in particular having a close approach towards finding and predicting the market values.This along with other market analysis can help the investors to earn profits.

In the project the regression models that are used in the industry is identified and a sample implementation for the same is done.The shortcomings for the same is also seenand how a better advanced approach of neural networks is used to solve the problem isalso seen.

An extensive understanding of recurrent neural networks is performed. The algorithm frameworks and formulas are understood ,applied and the possible shortcomings of RNN also studied.A sample prediction of the stock prices using LSTM to overcome RNN is done,which gives close to accurate results.

For future work the recurrent neural network's implementation in stock market is to be done for multiple features using PCA and then a generalized result would be produced.Also other various models like Random Forest,SVM,ANN,CNN etc will be explored and the performance of all will be compared on a single dataset.

9 REFERENCES

[1]Technical analysis of Stock Trends(Robert D. Eddwards ,John Magge) pg151-168,pg251-289

[2]"Machine Learning applications in financial Markets" by Prashant Pawar, Btech IIT Bombayhttps://www.iith.ac.in/ saketha/research/ppBTP2010.pdf pg 13-15

[3]"Stock Market Price Prediction Using Linear and Polynomial Regression Models" by Lucas Nunno,Computer Science Department,University of New Mexicohttp://www.lucasnunno.com/assets/docs/ml_paper.pdf pg3-5

[4]"Stock Market Prediction Using Machine Learning Algorithms",by K. Hiba Sadia, Aditya Sharma, Adarrsh Paul, SarmisthaPadhi, Saurav Sanyalhttps://www.ijeat.org/wp-content/uploads/papers/v8i4/D6321048419.pdf pg28-30

[5]"STOCK PRICE PREDICTION USING RECURRENT NEURAL NETWORKS",by Israt Jahan,Software Engineering,North Dakota State University of Agriculture and Applied Science https://bit.ly/3mKmxmh pg-14-21

[6]"Basic Understanding of LSTM"by Amey Laddad:https://blog.goodaudience.com/basic-understanding-of-lstm-539f3b013f1e pg-1

10 APPENDIX

Code for Linear Regression: https://bit.ly/3sbzKYf

Code for Polynomial Regression: https://bit.ly/3m9UyeM

Code for RNN and LSTM: https://bit.ly/30vF90L

Data for RNN-LSTM- https://finance.yahoo.com/