Stock Market Prediction Analysis Using Different Algorithms

Naman Rohilla, Yogesh Goyal, Ajay Khasiya

Dr. Bhavesh N. Gohil, Dr. A.K. Shukla

Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India

Abstract. With around 21 major stock exchange groups worldwide, the stock market is one of the significant choices for investors to invest funds. Prediction of the stock price with high precision is challenging due to the high volume of investors and market volatility. The volatility of the market is due to non-linear time series data. To handle such data, various algorithms are available. Many works have been done to predict a stock price however, the prediction with high accuracy is still a challenge to achieve.
This paper deals with the prediction of the daily high price of a stock. The experiment is done using AI (Artificial Intelligence) and ML (Machine Learning) algorithms (LSTM, XGBoost, and Regression specifically). The model is trained using previously available stock data, and the acquired knowledge (trained model) is used to predict the stock price precisely. The accuracy achieved in the proposed algorithm is around 97%. Different types of datasets are utilised to achieve favourable outcomes. It can be observed that building a proper model can help an investor to create a better stock portfolio, as better prediction improves the net profit for the investor.

Keywords: Stock Market Prediction, Machine Learning, time series data, algorithms, Volatility of the market.

1 Introduction

In Stock Market, the purchase and selling of stocks, shares, currencies and other derivatives through tangible forums supported by traders (a flexible and interconnected system) take place. Investors can purchase and sell shares in public corporations (Public corporations are the companies whose stocks are available on a stock exchange platform).

Due to market volatility, accurately foreseeing the stock price is a typical task [14]. The market is volatile because of two reasons,

1. Known variables (open and close price, high and low price, volume, etc.)
2. Unknown variables or Sentiments (Company reputation, inside information, election results, natural disaster, rumours in the market, country budget, etc.).

In stock market prediction, as the data is time series data, the strategy implied is also based on the same. The performance of multi-time guessing algorithms has been demonstrated in this paper. Long Short-term Memory (LSTM), XGBoost and Regression are the algorithms utilised in this project to anticipate high stock prices.

Time series forecasting modelling is essential in predicting the data like the stock market. Time series analysis is a statistical subset commonly used in econometrics and operations research. Stock prices, for example, high and low prices, are highly volatile, and several variables influence their worth.

In this paper, the prediction is solely based on known variables. The available data of a public corporation is used as the training data set for model training. However, risk strategies and safety measures have been implemented.

Individual observations are required to evaluate the prediction during real-life implications. Many factors are integrated and examined, such as the trading strategy to be used, sophisticated mathematical functions to show the condition of a particular stock, machine learning algorithms to anticipate future stock values, and unique stock-related concerns. Work done here is to anticipate future stock value using machine learning algorithms.

~~**Motivation for work:** The prediction of stock prices accurately is a challenging task to be achieved. Many different approaches have been made to create a framework for the same. Even after a lot of research and different algorithms, a favourable outcome is yet to be achieved. The motivation behind the work is to achieve a good result. If we can create a framework to predict a stock price accurately, it will help investors and companies to invest accordingly. The goal can be achieved with the proper stock market knowledge and the machines' computational power. Thus, our motivation is to create a platform for investors to invest in funds with low risk and high outcomes.~~

2 ~~Survey of~~ Related Work~~s~~

Different approaches ~~had been made~~ earlier to predict stock prices; a few are discussed here, along with their drawbacks.

~~Kranthi Vanukuru~~[7] ~~created~~ a model based on the SVM algorithm for the prediction of the stock price of IBM Inc. SVM requires a considerable dataset value for the training of the model, and overfitting is also not an issue. The model worked well for the selected corporation, but if considered a new corporation with a relatively low dataset, the prediction of such stock price would not be that accurate using the SVM model.

~~Somaraju Dinesh~~[12] ~~and others~~ worked with the LSTM algorithm to predict stock prices. In the model, they predicted the stock's closing price and achieved around 95% accuracy. The datasets used were Google, Nifty50, TCS, Infosys and Reliance. The performance of this model is high due to consistent change in the dataset, if a dataset provided to the model has rapid change, the model will fail, in this paper, we have discussed the performance of the LSTM model with the dataset of Tesla, and due to extreme change in the dataset, the model achieved around 30% accuracy.

~~Sreelekshmy Selvin~~[14] ~~and others created~~ a model based on CNN. The dataset used were Infosys, TCS and CIPLA. The accuracy achieved in the model is excellent. With every type of dataset, the model will perform outstanding until the prediction is based on sentiment analysis. CNN algorithm has a fixed input and output ratio, which doesn't allow prediction based on sentiments. This factor of CNN limits the future goal in stock price prediction.

Aparna Nayak[2] and others compared models based on Decision Boosted Tree, SVM and LR. They concluded that Decision Boosted Tree performs better than SVM and LR while predicting a curve motion up/down. The accuracy achieved is good, but in the practical world, prediction of the motion of the curve is not sufficient to invest with low risk. Along with the motion of the curve, one needs to predict the value of the stock, which may lead to a change in conclusion.

Mehar Vijh[8] and others created a framework based on ANN and RF and concluded that ANN performs better. They predicted the next day's closing price using the framework. The dataset used were Nike, Goldman Sachs, Johnson and Johnson, Pfizer and JP Morgan Chase and Co. The analyses done are comparative rather than result oriented. The accuracy achieved is not sufficient for actual investment.

3 ~~Models and Results~~

~~3.1 Experimental Setup/Structure of Model~~

Fig. 1. shows the structure of our program to its controllable levels. For this demonstration exercise, we have predicted high prices of ONGC, Tata Steel and Tesla stocks(dataset considered is of past seven years (2015-2022)). All the analysis is done in jupyter notebook, an open-source web application that allows one to create and share documents that contain live code, equations, visualisations, and narrative text.

We have used NumPy and Pandas for data cleaning, which provide high-performance, easy-to-use data structures and data analytic tools for manipulating numeric and time series data.

For data pre-processing and modelling, we have used sklearn, which contains a lot of efficient tools for machine learning and statistical modelling, including classification, regression, clustering, and dimensionality reduction. For visualisation, we have used matplotlib to create static, animated, and interactive visualisations.

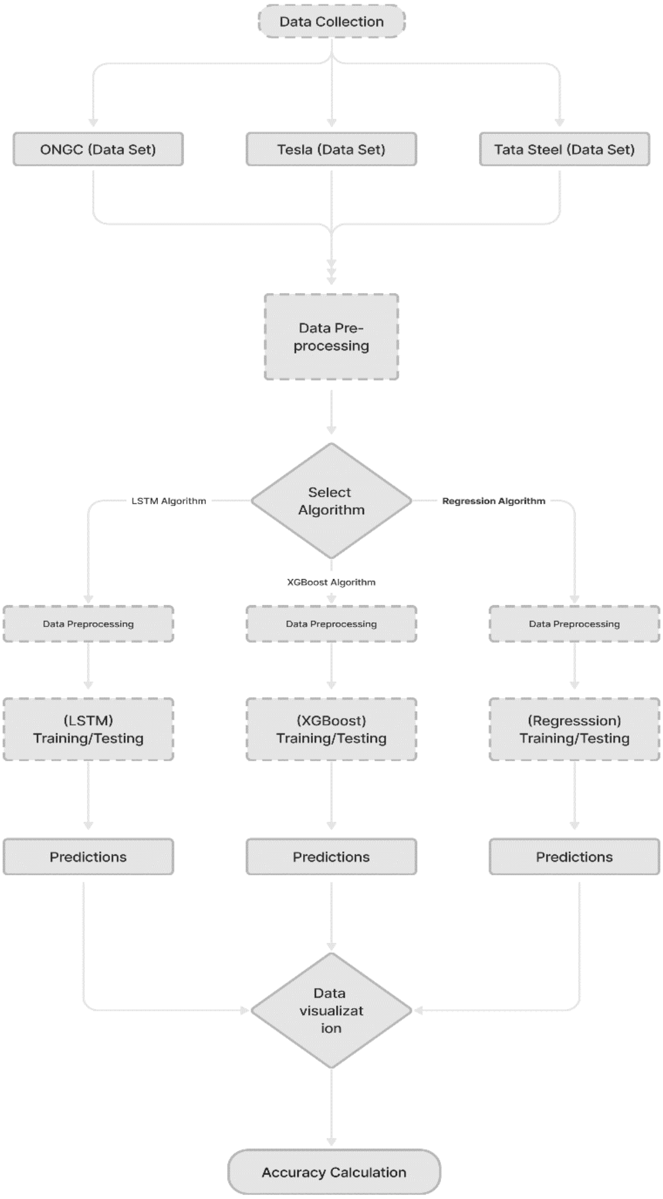


Fig. 1. Experimental Structure of the Model

LSTM: Long Short-Term Memory[5][13] network is a particular type of RNN that can learn long-term dependencies. They are currently widely utilised and function wonderfully in a variety of environments. It was specially designed to solve the issue of long-term dependencies. It is created so that it remembers data for a long time and doesn’t have to work hard to access that data. The initial stage starts with the decision made by “forget the gate layer”. It decides which information cells can ignore.

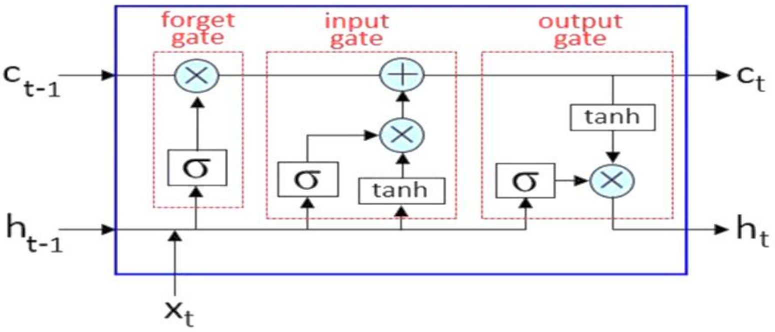
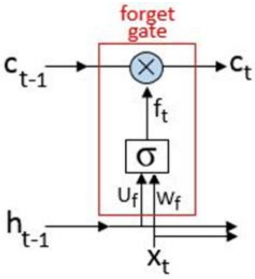


Fig. 2. Structure of LSTM

This is a sigmoid layer. h_{t-1} and x_t are two dependent variables which return values from 0 to 1. This value is stated to C_{t-1} cell. "1" means the information will be kept, while "0" means to reject the information.



.

Fig. 3. Inside the forget gate

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$c_t = c_{t-1} \otimes f_t$$

It further moves to the "input gate layer"[5]. It decides what other information needs to be kept. Again, it's a sigmoid layer which is updated by the tan(h) layer, it creates a C_{t-1} vector for new cell value.

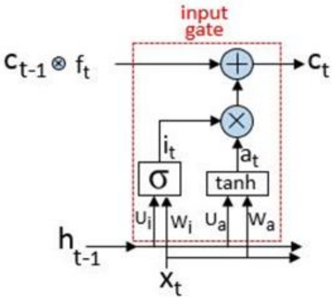


Fig. 4. Inside the input gate

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$a_t = \tan h\,(W_i x_t + U_a h_{t-1} + b_a)$$
$$c_t = (c_{i-1} \otimes f_t + i_t \otimes a_t)$$

Finally, both the above states are joined together. The transition is made from the previous cell state C_{t-1} to the new cell state C_t. As it's the preceding phase, we know that all the remaining data needs to be carried out. We now move to the items we decided to forget earlier. We append that value i_t C_t. This states by what value each state value has been updated. At last, we determine what should be produced. The output which needs to get filtered depends on the cell condition. A sigmoid layer is used to decide which cell state should be output. Further, it is multiplied by the sigmoid gate output and is processed through tan(h) function. It is done to get values ranging from -1 to 1. The value generated is the required result.

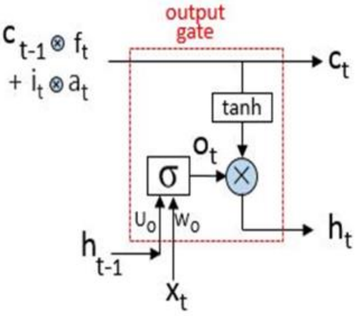


Fig. 5. Inside the output gate

$$O_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$h_t = \tan h\,(c_t x_t \otimes O_t)$$

XGBoost: Extreme Gradient Boosting[10] is one of the methods for ensemble learning. Relying on a few methods of machine learning models is not always been enough. It is a technique for methodically integrating the predictive abilities of several learners. Because of this, a single model is created that includes the outcomes of several models.

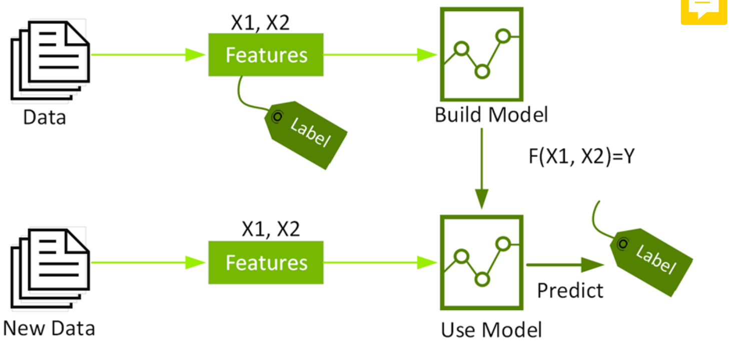


Fig. 6. XGBoost model

Regression: Regression[9] is a machine learning algorithm which comes under supervised learning. It is entirely based on independent variables and dependent variables. It is mainly based on predicting and creating the relationship between variables.

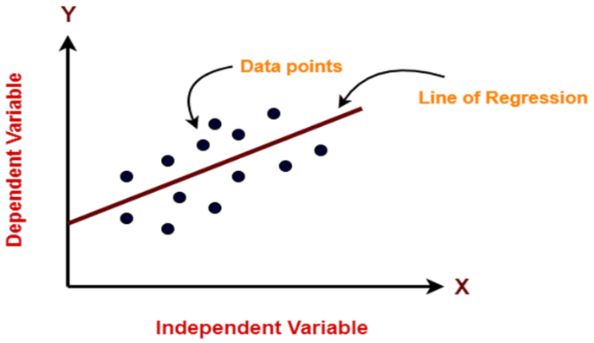


Fig. 7. Relationship between Dependent and Independent variable

~~Our~~ Regression model used (y) as a dependent variable and (x̂) as an independent vector. Thus, the linear relationship between x̂ and y is identified as a result of this regression approach. The above figure x̂ indicates the person's experience gain, and y indicates that person's salary. The line which fits the best for our model is called the regression line. Linear Regression Hypothesis Function*:*

$$y = \theta_1 x_1 + \theta_2 x_2 \ldots.. + \theta_n x_n + c$$

To train, there are some essential variables to define[15]

- Parameter
- Univariate

x: independent vector or input data
y: the labels of the data (supervised learning) or dependent variable
Now it will find the best line for the model to predict the value of y and $\hat{x}$ after training the model. To find the best fit line, we have to obtain the best value for the coefficient of $\hat{x}$

c: intervene of the data
θ_i: coefficient of x_i

After getting the best θ_i value, we will get the best fit line. So, when we use our model to predict the value of y for the input value of x_i, it will indicate the value of y. The model tries to forecast the value of y using the best-fit regression line such that the error difference between anticipated and projected values are as small as possible. As a result, updating the θ_i values is crucial to discover the optimal value that minimises the discrepancy between the predicted and actual values of y.

$$minimize\ \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

J: Cost function, also known as Root Mean Squared Error (RMSE) between the predicted y value (prediction) and real y value (actual (y)). Gradient Descent: Gradient Descent is used by the model to update θ_i values to minimise the Cost function (RMSE value) and get the best fit line for the model. Starting with random θ_i integers, the aim is to iterative update the values until the cost is as low as feasible

3.2 Data Set

Date	**Close (₹)**	**Open(₹)**	**High (₹)**	**Low (₹)**	**Volume(M)**	**Change (%)**
2015-04-07	212.67	213.33	213.67	210.00	3.49	0.55 ↑
2015-04-08	208.67	212.67	212.67	207.93	5.98	1.88 ↓
2015-04-09	206.77	209.67	210.53	205.87	5.71	0.91 ↓
2015-04-10	206.50	206.80	210.53	206.10	8.89	0.68 ↓
2015-04-13	207.90	209.00	209.00	206.73	2.80	0.68 ↑
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2022-03-31	163.90	161.85	166.25	161.15	3.35	1.17 ↑

2022-04-01	167.95	163.90	168.25	163.55	2.83	2.47 ↑
2022-04-04	168.05	166.95	169.05	165.35	1.77	0.06 ↑
2022-04-05	171.85	170.25	172.75	169.30	2.00	2.26 ↑
2022-04-06	173.00	170.50	173.50	170.50	1.29	0.67 ↑

Table 1. Data Set of ONGC [1752 rows x 6 columns]
("M" represents Million, ₹ represents INR, and "%" represents percentage).

Date	**Close (₹)**	**Open (₹)**	**High (₹)**	**Low (₹)**	**Volume(M)**	**Change (%)**
2015-03-16	294.62	297.01	297.01	291.82	2.91	0.09 ↓
2015-03-17	300.36	298.05	303.88	295.43	4.54	1.95 ↑
2015-03-18	299.63	303.20	303.20	297.46	3.25	0.24 ↓
2015-03-19	304.01	301.89	311.42	301.35	6.45	1.46 ↑
2015-03-20	296.88	304.01	308.35	295.34	6.06	2.35 ↓
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2022-04-07	1349.50	1377.00	1381.00	1341.00	5.57	1.55 ↓
2022-04-08	1370.75	1360.95	1379.95	1356.25	5.23	1.57 ↑
2022-04-11	1357.90	1370.70	1384.00	1353.00	4.48	0.94 ↓
2022-04-12	1320.25	1353.00	1359.95	1296.10	6.92	2.77 ↓
2022-04-13	1319.50	1344.95	1346.00	1315.00	4.49	0.06 ↓

Table 2. Data Set of Tata Steel [1733 rows x 6 columns]
("M" represents Millions, ₹ represents INR, and "%" represents percentage).

Date	**Close ($)**	**Open ($)**	**High ($)**	**Low ($)**	**Volume(M)**	**Change (%)**
2015-01-02	43.86	44.57	44.65	42.65	2.38	1.39 ↓
2015-01-05	42.02	44.91	43.30	41.43	2.68	4.20 ↑
2015-01-06	42.26	42.01	42.84	40.84	3.13	0.57 ↑
2015-01-07	42.19	42.67	42.96	41.96	1.48	0.17 ↓
2015-01-08	42.12	42.56	42.76	42.00	1.72	0.17 ↓
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2022-04-18	1004.29	989.03	1014.92	973.41	1.69	1.96 ↑
2022-04-19	1028.15	1005.06	1034.94	995.33	1.64	2.38 ↑
2022-04-20	977.20	1030.00	1034.00	975.25	2.16	4.96 ↓
2022-04-21	1008.78	1074.73	1092.22	996.41	3.49	3.23 ↑
2022-04-22	1027.81	1014.47	1034.50	994.29	1.02	1.89 ↑

Table 3. Data Set of Tesla [1840 rows x 6 columns]
("M" represents Millions, $ represents USD, and "%" represents percentage).

3.3 ~~Result~~

We examined the company’s historical stock price to determine the pattern for predicting future stock prices. Stock price prediction is based on the companies’ historical data without sentimental value. The accuracy achieved is shown in the table below.

Algorithms accuracy on multiple stocks (in %)			
Algorithms	**ONGC stock**	**Tata Steel stock**	**Tesla**
LSTM	97.55%	89.19%	<30%
XGBoost	Failed	Failed	Failed
Regression	99.95%	96.53%	97.41%

Table 4. Algorithms accuracy table

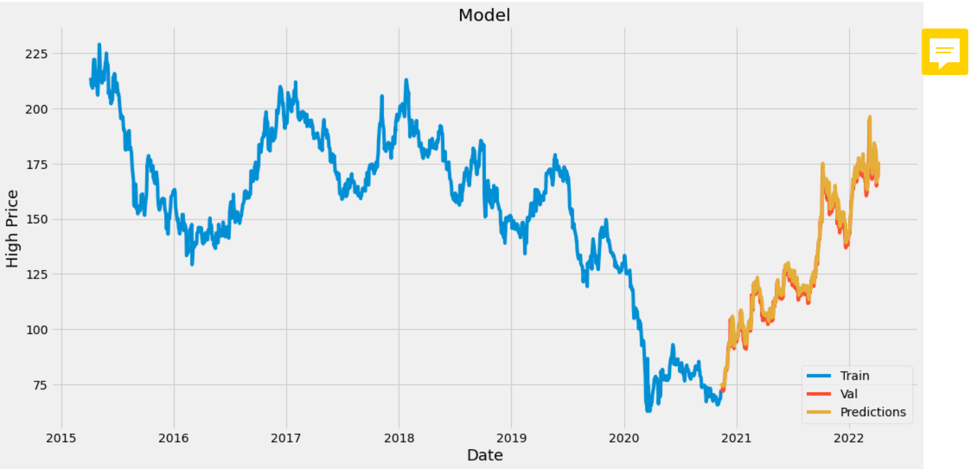


Fig. 8. Line graph: Prediction graph of high price. Algorithm: LSTM. Dataset: ONGC. (~~Blue line represent training data points, red line represents actual data points, orange line represents predicted data points by the model)~~

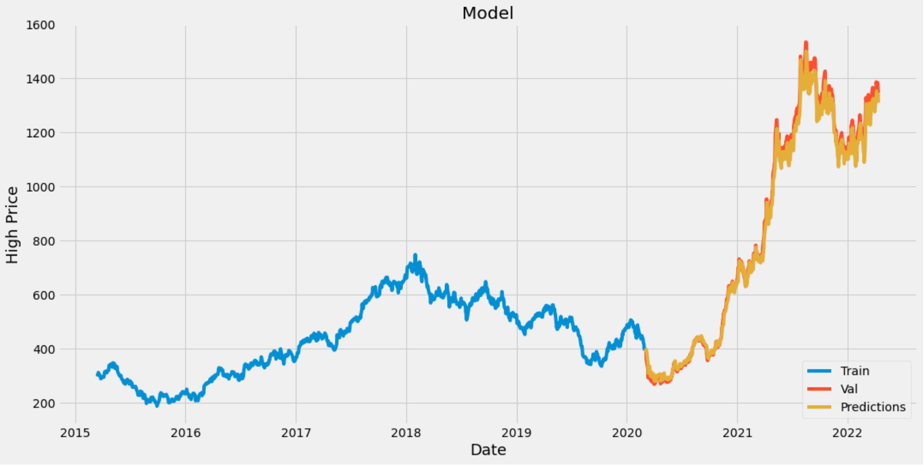


Fig. 9. Line graph: Prediction graph of high price. Algorithm: LSTM. Dataset: Tata steel. (~~Blue line represents training data points, red line represents actual data points, orange line represents predicted data points by the model)~~

LSTM performed well for the dataset with fewer variations (ONGC) but for dataset with moderate variation (Tata Steel) it performed well with one specific epoch while failed at every other epoch value, this result probably occurred due to the memory-based methodology of LSTM.

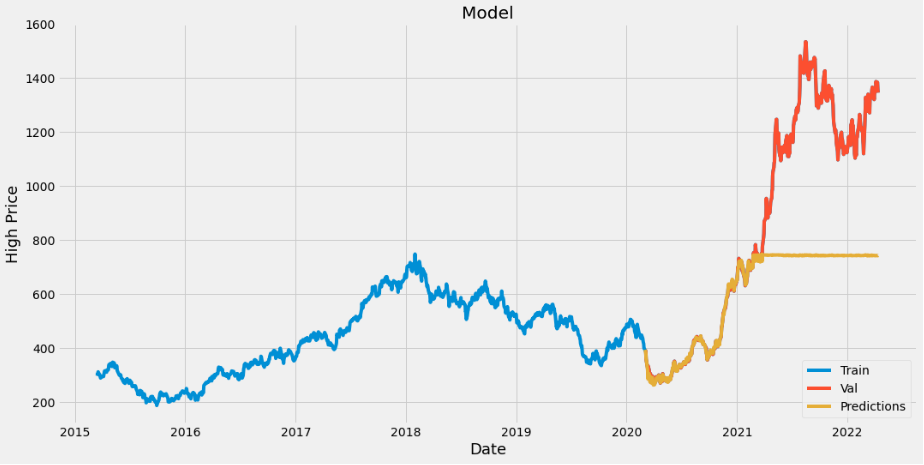


Fig. 10. Line graph: Prediction graph of high price. Algorithm: XGBoost. Data set: Tata steel. ~~(Blue line represents training data points, red line represents actual data points, orange line represents predicted data points by the model~~)

XGBoost failed to predict all the cases. The algorithm cannot extrapolate target values beyond the limits of the training data set when making the prediction, and the input space of any given problem is limited.

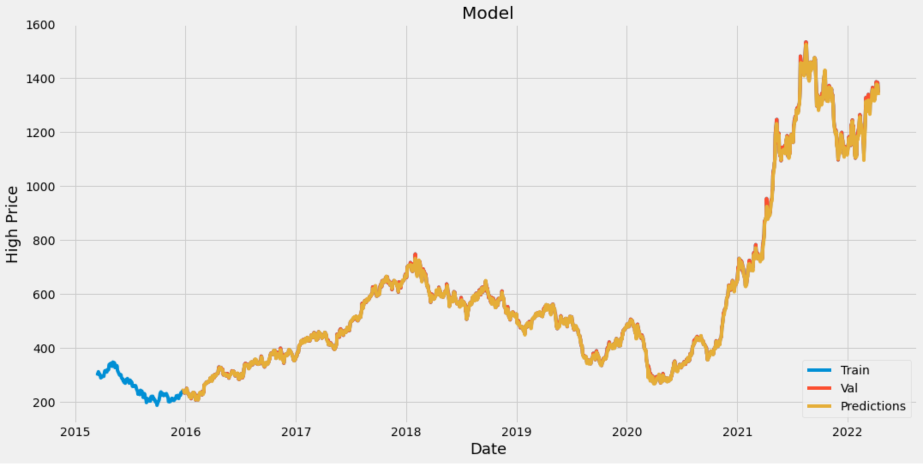


Fig. 11. Line graph: Prediction graph of high price. Algorithm: Regression. Dataset: Tata steel. (~~Blue line represents training data points, red line represents actual data points, orange line represents predicted data points by the model)~~

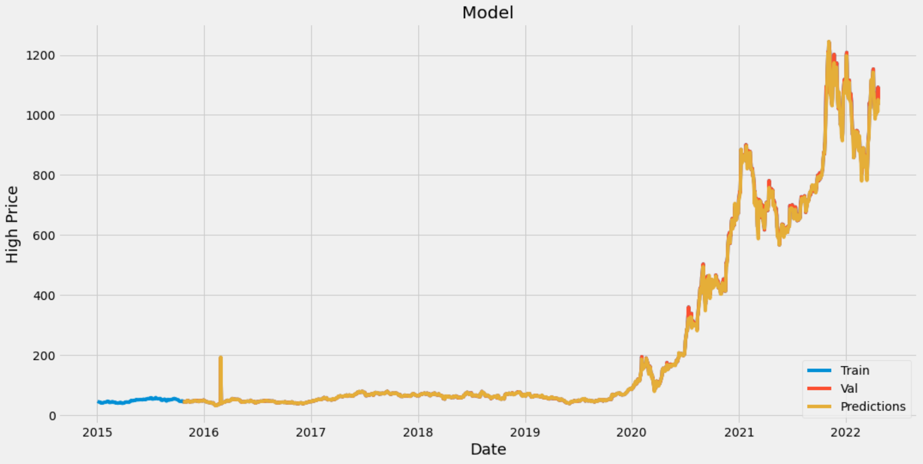


Fig. 12. Line graph: Prediction graph of high price. Algorithm: Regression. Dataset: Tesla. (~~Blue line represents training data points, red line represents actual data points, orange line represents predicted data points by the model)~~

Regression performed the best in the considered case, as the algorithm creates the optimal value of slopes in a given linear expression, considering high price as the dependent variable and other values as an independent variable, linear expressions are generated to obtain optimal values of sloped to predict high price accurately.

4 Conclusions and Future Work

In this ~~project~~, we examined the dataset of Tata Steel, ONGC and Tesla to determine the pattern for predicting the future high price of the stock. The models are based on LR, LSTM and XGBoost. The dataset collected is readily available on investing.com[16]. We pre-processed seven years of data of these companies. We trained our model with 80% of the data and tested the model with 20% of the data, the accuracy achieved was very different for each type of dataset and model. The best accuracy achieved is in the LR model.

~~Our proposed model~~ is very different from previous works. We ~~have~~ compared different algorithms with different types of datasets to understand which algorithm performs best under which circumstances. We observed that LR is the best algorithm among the performed models with each type of dataset. Still, there are some stones left to be unturned.

We aim to create a model which, along with the available dataset, considers sentiments and human reactions for predicting stock price. Also, integrating this model with one's Demat account for automatic trading to achieve the best profit will be our goal to achieve.

References

[1]. Amit Kumar Das Subramanian Chandramouli, Saikat Dutt. Machine Learning. Pearson Education India, 2018.

[2]. Aparna Nayak, M. M. Manohara Pai, Radhika M. Pai. Prediction Models for the Indian Stock Market. Procedia Computer Science, Volume 89, 2016, Pages 441-449, ISSN 1877-0509.

[3]. Cheng Soon Ong Marc Peter Deisenroth, A. Aldo Faisal. Mathematics for Machine Learning. Cambridge University Press, illustrated edition, 2020.

[4]. Chi-Feng Wang. The Vanishing Gradient Problem. https://towardsdatascience.com/ the-vanishing-gradient-problem-69bf08b15484. Medium American Publisher, 2019.

[5]. Chris Colah's Blog. Understanding LSTM Networks. https://colah.github.io/posts/ 2015-08-Understanding-LSTMs/. San Francisco, CA, 2015.

[6]. J. Brownlee. Ensemble Learning Algorithms with Python, volume 450 pages. Machine Learning Mastery Vermont, Victoria, 2021.

[7]. Kranthi Vanukuru. Stock Market Prediction Using Machine Learning. Research paper published by Sreenidhi Institute of Science Technology India, 2018.

[8]. Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar. Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science, Volume 167, 2020, Pages 599-606, ISSN 1877-0509.

[9]. Mohit Gupta. ML, Linear Regression. https://www.geeksforgeeks.org/ ml-linear-regression/. Blog published by Geeks-For-Geeks, India, 2017.

[10]. Ramya Bhaskar Sundaram. An end-to-end guide to understand the math behind XGBoost. https://www.analyticsvidhya.com/blog/2018/09/ an-end-to-end-guide-to-understand-the-math-behind-xgboost/. Blog Published by Analytics Vidhya, India, 2018.

[11]. Rory Mitchell. Gradient Boosting, Decision Trees and XGBoost with CUDA. https://developer.nvidia.com/blog/gradient-boosting-decision-trees-xgboost-cuda/. Published by NVIDIA Developer, Santa Clara, California, 2017

[12]. Sasumana Rahul Oruganti Naga Sandeep Somaraju Dinesh, Adduri Maruthi Siva Rama Raju. Stock Price Prediction. Report published by Anil Neerukonda Institute of technology and sciences, India, 2021.

[13]. Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural Computation, MIT Press Direct, U.S, 9(8):1735–1780, 1997.

[14]. Sreelekshmy Selvin, Vinayakumar Ravi, E.A Gopalakrishnan, Vijay Menon, and Soman Kp. Stock price prediction using LSTM, RNN and CNN-sliding window model. Pages 1643–1647, 09. International Conference on Advances in Computing, Communications and Informatics, MIT Manipal India, 2017.

[15]. Suvarna Gawali. Linear Regression in Machine Learning. https://www.analyticsvidhya.com/ blog/2021/06/linear-regression-in-machine-learning/. Blog Published by Analytics Vidhya, India, 2021.

[16]. **Dataset**: https://in.investing.com/equities/