STOCK MARKET PREDICTION USING MACHINE LEARNING: A COMPARATIVE ANALYSIS OFMODEL ACCURACIES FOR AN INDIAN STOCK

Mayank Gupta, Department of Instrumentation & Control Engineering, Bharati Vidyapeeth College of Engineering, Guru Gobind Singh Indraprastha University, India

ABSTRACT

This manuscript took a look at evaluates the performance of four machine getting to know models—Linear Regression, Random Forest, Long Short-Term Memory (LSTM) neural networks, and XGBoost—for predicting stock fees of an Indian inventory, Tata Steel (TATASTEEL.NS), listed at the National Stock Exchange (NSE). Historical inventory records over 3 years turned into used to educate and check the models, with performance assessed the usage of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Results had been visualised through comparative graphs. Linear Regression finished the bottom RMSE (3.03) and MAE (2.30), outperforming other models, which includes Random Forest (RMSE: 13.01, MAE: 11.26), LSTM (RMSE: 6.18, MAE: 4.83), and XGBoost (RMSE: thirteen.18, MAE: eleven.54). The findings spotlight the effectiveness of less complicated models in sure financial forecasting situations and contribute to the developing area of machine mastering applications in inventory market prediction, mainly for business region stocks in the Indian market.

Keywords:
machine learning, Random Forest, LSTM, stock market prediction, XGBoost.

I. Introduction

Stock marketplace prediction remains a tough but vital task for investors, financial analysts, and policymakers due to the inherent volatility and complexity of economic markets. The Indian stock marketplace, represented by means of the National Stock Exchange (NSE), is a dynamic surrounding inspired via macroeconomic factors, worldwide financial developments, and quarter-particular traits. Indian stocks, in particular the ones in the commercial and production sectors like Tata Steel (TATASTEEL.NS), are sensitive to international commodity expenses, government guidelines, and economic cycles, making their price prediction both tough and precious for stakeholders [1]. Traditional statistical methods, which includes time collection analysis and autoregressive fashions, frequently fail to seize the non-linear and chaotic patterns in inventory price movements, prompting researchers to explore device getting to know strategies [2].

Machine mastering gives powerful gear to model complex relationships in economic information, leveraging algorithms that can study from historical styles and adapt to new information. Recent research has carried out numerous machine learning fashions, consisting of selection trees, neural networks, and gradient boosting, to stock rate prediction with promising results. However, the effectiveness of these models can vary relying on the stock, market situations, and statistics traits. This study specializes in evaluating 4 machines getting to know models—Linear Regression, Random Forest, Long Short-Term Memory (LSTM) neural networks, and XGBoost—for predicting the inventory prices of Tata Steel, an Indian inventory listed at the NSE. The objectives are to: (1) check model overall performance the usage of RMSE and MAE, and (2) visualise predictions via graphs to apprehend version behaviour.

The desire of Tata Steel as the target inventory is prompted by way of its prominence inside the Indian marketplace and its exposure to each home and worldwide economic elements. Steel enterprise shares are motivated via raw fabric prices, global call for, and exchange regulations, which introduce additional complexity into charge prediction duties [3]. By comparing the performance of various device studying models on TATASTEEL.NS, this research aims to provide insights into the suitability of those models for Indian shares within the industrial region and contribute to the wider understanding of gadget gaining knowledge of applications in economic forecasting within the Indian marketplace context.

II. Materials and Methods

2.1 Data Collection

Historical remaining charges for TATASTEEL.NS had been retrieved the use of the yfinance Python library, masking a 3years period from May 2022 to May 2025. The dataset consisted of each day ultimate fees, which were wiped clean to do away with any missing values or anomalies, which include the ones caused by market holidays or data retrieval mistakes. An appearance-back length of one hundred fifty days was selected to create input sequences for version schooling, based on initial experiments that showed this window length successfully captured brief- to medium-term tendencies within the stock price statistics.

2.2 Data Preprocessing

The raw remaining fees were normalized the usage of the MinMaxScaler to scale values among 0 and 1, a step vital for ensuring compatibility with system getting to know algorithms, mainly LSTM, that is sensitive to the size of enter information. Normalization also enables mitigate the effect of fee volatility on model training. The dataset was then divided into schooling (80%) and checking out (20%) units without shuffling to preserve the temporal order of the data, that's crucial for time series prediction tasks. The training set was used to healthy the models, even as the check set become reserved for comparing their predictive overall performance on unseen data. To prepare the statistics for model input, sequences of one hundred fifty days (the appearance-lower back length) had been created to are expecting the following day's ultimate price. This sequence generation procedure worried growing overlapping windows of a hundred and fifty days as enter capabilities and the corresponding next day's charge as the target variable, resulting in a supervised gaining knowledge of dataset appropriate for regression duties. For LSTM, the enter data was in addition reshaped right into a three-dimensional layout (samples, time steps, functions), with the feature dimension set to one (closing charge), to fulfill the version's input necessities. Mathematical Expressions and Symbols

2.3 Machine Learning Models Four device mastering fashions had been applied using Python libraries (scikit-examine, tensorflow, xgboost), each selected for its awesome method to regression duties and suitability for economic time series records. Below is in depth explanation of each version and its configuration:

Linear Regression: Linear Regression is an essential statistical technique that fashions the relationship between input functions and the target variable as a linear equation. In this study, the input functions consist of the beyond 150 days' closing prices, and the target is tomorrow's ultimate charge. The model minimizes the sum of squared residuals between predicted and actual values using ordinary least squares. Despite its simplicity, Linear Regression serves as a robust baseline, in particular for datasets with linear or near-linear developments, and its interpretability makes it a precious benchmark for comparing more complex models [4]. The version assumes a linear relationship, expressed as y=β0 + β1*x1 + β2*x2 +…+β150*x150, where y is the predicted price, $x1$, $x2$, …, $x150$ are the tagged prices, and $\beta 0$, $\beta 1$, …, $\beta 150$ are the coefficients learned during training.

Random Forest: Random Forest is an ensemble gaining knowledge of technique that mixes multiple choice bushes to enhance predictive accuracy and reduce overfitting. Each tree is trained on a random subset of the statistics (the usage of bootstrap sampling) and a random subset of features at each cut up, introducing range many of the trees. In this look at, the Random Forest version turned into configured with a hundred decision timber (n_estimators=100) and a random country of 42 for reproducibility. The input capabilities had been the 150 lagged charges, and the model predicts the following day's charge by way of averaging the predictions from all trees. Random Forest is in particular powerful for taking pictures non-linear relationships and coping with noisy facts, as the ensemble technique reduces variance and mitigates the danger of overfitting inherent in single choice trees [5]. The version's robustness makes it a famous choice for monetary prediction duties, in which information regularly famous complicated styles.

Long Short-Term Memory (LSTM): LSTM is a form of recurrent neural community (RNN) designed to version sequential statistics by way of retaining a reminiscence of beyond inputs thru its cell state and gates (forget, input, and output gates). This architecture is in particular perfect for time series prediction, as it may seize long-term dependencies inside the facts, which are often found in stock charge actions [6]. In this study, the LSTM model turned into configured with two stacked LSTM layers, every containing 50 gadgets, accompanied through dropout layers with a 20% dropout charge to save you overfitting. The first LSTM layer returns sequences to allow stacking, while the second outputs a unmarried vector. A dense layer with one unit was delivered as the output layer for regression. The model turned into compiled with the Adam optimizer and mean squared error loss feature, and it was trained for 20 epochs with a batch length of 32. The input data was reshaped right into a 3-dimensional layout (samples, 150time steps, 1 characteristic), and the model turned into trained to predict the next day's scaled rate, which become later inverse-converted to the authentic charge scale for evaluation. LSTM's capability to maintain memory of past fee moves makes it a strong candidate for stock prediction, although its performance depends heavily on hyperparameter tuning and education duration.

XGBoost: XGBoost (Extreme Gradient Boosting) is a scalable and efficient implementation of gradient boosting, a gadget gaining knowledge of technique that builds an ensemble of choice bushes in a sequential manner to minimize a loss characteristic. XGBoost is understood for its excessive overall performance in regression and class obligations, specifically in financial programs, due to its potential to deal with non-linear relationships and incorporate regularization to prevent overfitting [7]. In this examine, the XGBoost model became configured with 100 estimators (bushes), a gaining knowledge of fee of 0.1, and a random state of 42 for reproducibility. The model uses the 150 lagged prices as enter features and predicts day after today's fee through iteratively including timber that accurate the errors of the preceding ones, optimizing a squared errors loss function. XGBoost additionally includes capabilities like L1 and L2 regularization, early preventing, and parallel processing, which decorate its efficiency and generalization capacity. Its tree-based shape lets in it to seize complex styles inside the records, making it an aggressive version for stock price prediction.

2.4 Evaluation Metrics

Model performance was assessed using two widely used metrics for regression tasks, each providing a different perspective on prediction accuracy:

Root Mean Squared Error (RMSE): RMSE measures the square root of the common squared variations among predicted and real values. It calculates the error by averaging the squared differences among predicted and actual fees, taking the square root to convey the result back to the unique units. RMSE gives a higher weight to large mistakes because of the squaring operation, making it particularly touchy to outliers or tremendous deviations in predictions. This metric is beneficial for information the general value of prediction errors inside the equal gadgets as the target variable (stock fees in this situation), taking into account direct evaluation across fashions. A lower RMSE indicates better predictive accuracy, with values toward zero representing close to-ideal predictions. In the context of stock rate prediction, RMSE can spotlight fashions that produce big errors at some stage in volatile intervals, which is critical for assessing their reliability in real-world economic packages.

Mean Absolute Error (MAE): MAE measures the average absolute variations between expected and actual values. It computes the error through averaging the absolute variations between expected and actual prices, imparting a direct degree of the typical prediction error. Unlike RMSE, MAE does no longer square the errors, making it less sensitive to outliers and supplying a more straightforward interpretation of the common mistake significance. MAE is also expressed within the same units because the target variable, making it easy to understand the typical prediction mistakes in sensible phrases (ex.an MAE of 2.30 manner the average prediction blunders is ₹2.30). A decrease MAE suggests better version overall performance, and evaluating MAE with RMSE can reveal whether or not big mistakes disproportionately affect the model's performance (ex. a huge hole among RMSE and MAE indicates the presence of outliers). MAE is especially beneficial in financial programs, because it offers a clear degree of the common deviation in expected prices, which could at once inform funding choices or chance assessments.

Both metrics have been calculated at the check set after inverse-reworking the predictions from the scaled range (zero to at least one) again to the authentic rate scale the usage of the MinMaxScaler. This ensured that the errors have been significant inside the context of real inventory charges, taking into account a truthful evaluation across all models. RMSE and MAE together provide a complete view of version overall performance, balancing sensitivity to massive errors (RMSE) with a focal point on common errors (MAE), that is critical for evaluating the practical utility of the models in stock rate prediction.

2.5 Implementation

The fashions had been educated and tested using the prepared dataset on a system with Python 3.9 and the vital libraries mounted (scikit-analyze for Linear Regression and Random Forest, tensorflow for LSTM, and xgboost for XGBoost). The training manner involved becoming every version at the schooling set and producing predictions for the test set. For Linear Regression, Random Forest, and XGBoost, the input was a -dimensional array of shape (samples, a hundred and fifty functions), while LSTM required a 3-dimensional input of shape (samples, 150time steps, 1 characteristic). The predictions were inverse-transformed to the unique price scale, and the RMSE and MAE have been computed for every version. Results have been visualized the usage of matplotlib to plot the actual versus expected stock costs, enabling a qualitative assessment of version performance alongside the quantitative metrics

III. Results and Discussion

3.1 Model Performance

Table 1 affords the RMSE and MAE for each version based on the take a look at set for TATASTEEL.NS. Linear Regression executed the bottom errors (RMSE: 3.03, MAE: 2.30), outperforming the extra complicated fashions. LSTM followed with an RMSE of 6.18 and MAE of 4.83, indicating mild performance. Random Forest and XGBoost exhibited higher mistakes, with RMSE values of 13.01 and thirteen.18, and MAE values of 11.26 and eleven.54, respectively, suggesting challenges in capturing the underlying styles in the statistics.

Model	**RMSE**	**MAE**
Linear Regression	3.03	2.30
Random Forest	13.01	11.26
LSTM	6.18	4.83
XGBoost	13.18	11.54

Table 1: Model Performance Results

3.2 Graphical Comparison

Figure 1 illustrates the real versus expected stock expenses for the check length, as generated by the code. Linear Regression predictions align maximum carefully with the real costs, reflecting its low RMSE and MAE and suggesting that the stock price actions of TATASTEEL.NS for the duration of the check period may also have observed exceedingly linear tendencies. LSTM captures popular developments but indicates moderate deviations, mainly for the duration of intervals of volatility, which may be attributed to inadequate hyperparameter tuning or the version's sensitivity to noise. Random Forest and XGBoost show off larger discrepancies, with predictions deviating notably from the real prices, steady with their higher mistake metrics.

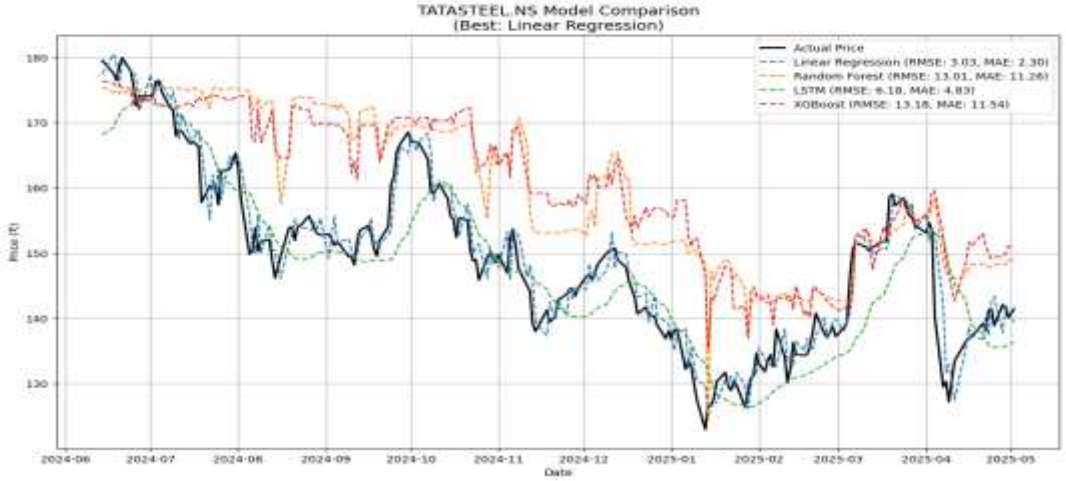

Figure 1: Actual vs. predicted stock prices for TATASTEEL.NS

3.3 Discussion

The superior overall performance of Linear Regression, with an RMSE of 3.03 and MAE of 2.30, indicates that the stock rate data for TATASTEEL.NS over the test duration may additionally showcase surprisingly linear patterns, which Linear Regression correctly captured due to its simplicity and focus on linear relationships [5]. The version's assumption of a linear dating between the beyond 150 days' prices and tomorrow's rate seems to keep properly for this dataset, as evidenced with the aid of the low errors metrics. This result is unexpected given the not unusual assumption that stock price actions are surprisingly non-linear and better applicable to complicated fashions like Random Forest or XGBoost. The better mistakes for Random Forest (RMSE: thirteen.01, MAE: eleven.26) and XGBoost (RMSE: thirteen.18, MAE: 11.54) may also imply overfitting to noise inside the training records or lack of ability to generalize well to the take a look at set. Random Forest's ensemble of one hundred choice bushes and XGBoost's iterative boosting approach, while designed to address non-linear patterns, may additionally have struggled with the specific characteristics of TATASTEEL.NS data, which include capacity linear trends or restricted function range (handiest historical costs had been used). The near RMSE and MAE values for those models suggest that their mistakes are always big as opposed to being skewed by outliers, as a massive gap among RMSE and MAE might imply the presence of extensive outliers.

LSTM, with an RMSE of 6.18 and MAE of 4.83, finished fairly properly, leveraging its capability to capture sequential dependencies thru its reminiscence cells and gates. The version's structure, with two LSTM layers, 50 units every, and 20% dropout, became meant to balance learning capability with overfitting prevention. However, its overall performance became probably hindered by using the restricted schooling duration (20 epochs) and the complexity of its structure. The 20-epoch schooling won't had been enough for the model to fully converge, and the dropout price might have reduced its capability to examine problematic styles within the records. Additionally, LSTM's sensitivity to hyperparameter settings, such as the range of units, mastering price, and batch size, indicates that similarly tuning should improve its performance, potentially making it more aggressive with Linear Regression.

The findings have essential implications for inventory marketplace prediction in the Indian context, in particular for business region shares like Tata Steel. The steel industry is situation to cyclical developments pushed with the aid of worldwide call for, uncooked material charges, and government guidelines, which may also result in periods of linear price movements interspersed with sudden volatility [4]. Linear Regression's fulfillment on this have a look at, as evidenced through its low RMSE and MAE, indicates that less difficult fashions may be effective in the course of strong durations, but their overall performance may also degrade for the duration of marketplace disruptions. Conversely, the underperformance of Random Forest and XGBoost highlights the need for careful characteristic engineering and hyperparameter tuning whilst applying ensemble strategies to financial facts. The moderate performance of LSTM suggests its capability for time series prediction, but its computational complexity and education necessities may also limit its realistic application without giant optimization.

A key trouble of this study is its reliance on historical remaining fees as the only enter feature. Stock charges are stimulated by way of a huge variety of things, which include buying and selling extent, macroeconomic signs, and market sentiment, which had been not taken into consideration right here. Incorporating these functions may want to improve the overall performance of complex models like Random Forest, XGBoost, and LSTM, doubtlessly making them extra competitive with Linear Regression. Additionally, the appearance-back period of a hundred and fifty days, while effective for shooting medium-time period traits, won't be foremost for all models or market conditions. Future research ought to discover adaptive look-again intervals or hybrid models that integrate the strengths of linear and non-linear procedures to obtain more robust predictions across exceptional marketplace situations.

IV. Conclusion

This demonstrates that Linear Regression changed into the handiest model for predicting inventory prices of TATASTEEL.NS, an Indian stock, accomplishing the lowest RMSE (3.03) and MAE (2.30) compared to Random Forest, LSTM, and XGBoost. The findings underscore the capacity of less complicated fashions in particular financial forecasting scenarios, in particular for commercial zone stocks for the duration of intervals of linear charge moves, and make a contribution to the information of machine studying applications in stock marketplace prediction in the Indian market. Future research could explore hybrid models, comprise extra functions consisting of buying and selling volume and sentiment analysis, and evaluate model performance under various marketplace situations to beautify predictive accuracy.

References

[1] J. S. M. O. Shafiq, “Short-term stock market price trend prediction using a comprehensive deep learning system,” Big Data, pp. 1-33, 2020.

[2] E. F. Fama, “Efficient capital markets: A review of theory and empirical work,” International journal of Computer Application, vol. 182, pp. 1-6, 1970.

[3] R. N. B. S. K. Gupta, “Global steel industry: Challenges and opportunities,” Journal of Business Strategy, pp. 22-30, 2019.

[4] G. A. F. S. A. J. Lee, “Linear Regression Analysis,” Wiley online library, 2003.

[5] Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5-32, 2001.

[6] S. H. J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, p. 1735–1780, 1997.

[7] T. C. C. Guestrin, “XGBoost: A scalable tree boosting system,” International Conference of Knowledge Discovery Data Mining, pp. 785-794, 2016.