

Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning

Rohit K. Sain¹, Amresh K. Singh¹, 

1. Department of Computer Science, Maharaja Ganga Singh University, Bikaner, IND

Received: July 03, 2026 | Review began: August 02, 2026 | Review ended: August 30, 2026 | Published: October 01, 2026

© Copyright 2026

This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

Forecasting stock index prices remains a challenging task because financial markets are volatile, non-stationary, and influenced by several economic and behavioral factors. This study evaluates 12 supervised machine learning models against a Naïve Persistence baseline for predicting next-day open and close prices of the NIFTY 50 index across 5-year, 10-year, and 20-year historical windows. The study compares two feature settings: one based on raw open, high, low, close, and volume variables and another that augments these variables with derived indicators such as the Simple Moving Average, Relative Strength Index, daily return, and rolling volatility. The evaluated models include Linear Regression, Lasso, Ridge, Elastic Net, Stochastic Gradient Descent Regressor, Support Vector Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting Regressor, AdaBoost Regressor, and eXtreme Gradient Boosting (XGBoost). The models are compared based on Mean Absolute Error, Root Mean Squared Error, and the R^2 score under chronological train-test splitting. The results show that linear models, particularly Linear Regression, Ridge, and Lasso, provide more stable generalization across 10-year and 20-year historical windows and consistently outperform the naïve benchmark. In contrast, Random Forest and XGBoost perform competitively in the 5-year setting but show weaker generalization for the 10-year and 20-year windows, often performing worse than a simple naïve forecast, even after hyperparameter tuning using TimeSeriesSplit. The findings suggest that simple linear models can serve as strong baselines for NIFTY 50 open and close price forecasting under the considered experimental setting.

Categories: AI applications, Machine Learning (ML)

Keywords: machine learning, time-series prediction, feature engineering, linear regression, xgboost, nifty 50 forecasting

Introduction

Predicting stock prices has become a common research topic in machine learning and financial analytics. The frequent fluctuations in stock markets have attracted researchers across several disciplines, including economics, finance, mathematics, statistics, and computer science [1,2]. In recent years, the number of retail investors, traders, and institutional investors has increased significantly, creating a strong demand for reliable forecasting tools for various investment horizons. However, predicting next-day stock prices is difficult since market values are affected by several uncertain and dynamic factors, such as investor sentiment, company performance, economic conditions, policy changes, global events, and sudden market shocks [3].

Financial time-series data are different from many other structured datasets because they are noisy, non-stationary, and highly sensitive to recent information [4]. A forecasting model that performs well in one time period may not necessarily perform equally well in another period because market behavior changes over time. Therefore, selecting an appropriate

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

forecasting model is an important research problem. In this context, machine learning techniques are increasingly used because they can learn hidden patterns from historical data and can be used for price prediction, trend analysis, portfolio planning, and risk management.

The NIFTY 50 [5,6] is one of the major benchmark indices of the Indian stock market. It represents the overall performance of 50 major companies traded on the National Stock Exchange across various sectors of the Indian economy [5,7]. Since the index reflects the overall movement of leading companies, forecasting NIFTY 50 prices is useful for understanding broader market behavior. In particular, next-day open and close price forecasting can help investors and analysts study short-term market movement, identify possible trading opportunities, and evaluate the behavior of different forecasting models under changing market conditions.

Over the years, researchers have investigated a wide range of statistical, machine learning, and deep learning approaches for forecasting stock markets. Traditional approaches, including the Autoregressive Integrated Moving Average (ARIMA) method, developed by Box and Jenkins in 1970 [8-10], and Holt-Winters Exponential Smoothing [11], have been commonly applied to time-series forecasting. Recent studies have indicated that deep learning approaches, such as Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks, and Long Short-Term Memory (LSTM) networks, have produced encouraging results in stock market prediction [9,11,12].

Although researchers have widely explored machine learning and deep learning methods for stock price forecasting, several limitations remain. Many existing works focus on a limited number of models, a single target variable, or a specific historical period [9,13,14]. Some studies mainly predict closing prices [6,13-15], while joint modeling of both open and close price forecasting is less commonly addressed in the reviewed literature. Similarly, several works use either raw market features or engineered technical indicators, but comparatively fewer studies analyze the effect of both feature settings under a common experimental framework. Moreover, the behavior of models may change when the length of the historical data window changes. Hence, a systematic comparison across multiple historical windows is useful for understanding model generalization.

To address these issues, this study performs a detailed comparative analysis of supervised regression models and a Naïve Persistence benchmark for next-day open and close price forecasting of the NIFTY 50 index. The experiments are conducted using 5-year, 10-year, and 20-year historical windows. Historical market data were obtained from Yahoo Finance [16] using the Python package *yfinance* [17]. Two feature pipelines are considered. The first pipeline uses raw OHLCV [13] features, namely Open, High, Low, Close, and Volume [18]. The second pipeline uses engineered technical indicators such as Simple Moving Average [3], Relative Strength Index (RSI) [19], daily return, and rolling volatility [7]. This allows the study to examine whether feature engineering improves forecasting performance compared with raw market data.

This study evaluates 12 supervised regression algorithms, including Linear Regression, Lasso, Ridge, Elastic Net, Stochastic Gradient Descent Regressor, Support Vector Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and eXtreme Gradient Boosting (XGBoost). These models are compared against a Naïve Persistence baseline to determine true predictive skill. In addition, Random Forest and XGBoost are further tuned using TimeSeriesSplit to examine whether hyperparameter optimization improves their generalization ability. Model performance is measured using commonly used regression metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2) [20].

This study makes three main contributions. First, it provides a comparative evaluation of 12 supervised regression models against a Naïve Persistence baseline for next-day open and close price forecasting of the NIFTY 50 index. Second, it analyzes the effect of 5-year, 10-year, and 20-year historical windows under a common experimental setting using both raw OHLCV features and engineered technical indicators. Third, it examines the effect of TimeSeriesSplit-based hyperparameter tuning on Random Forest and XGBoost and compares the generalization behavior of linear and non-linear models across different historical windows.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. *Cureus J Comput Sci* 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

The remainder of this paper is structured as follows. The Materials and Methods section reviews related studies and describes the methodology, including data collection, feature engineering, preprocessing, chronological train-test splitting, model training, and evaluation. The Results and Discussion section presents the findings across different historical windows. The Conclusion section summarizes the findings and future directions.

Materials And Methods

Related work

Stock market movement prediction has been studied using statistical, machine learning and deep learning techniques. Various approaches have been developed to enhance forecasting accuracy by identifying patterns in historical market data.



FIGURE 1: Historical trend of the NIFTY 50 index illustrating its long-term price movement and periods of market volatility over the selected study period

Data source: Google Finance [21]

Figure 1 shows the historical trend of the NIFTY 50 index for the selected period of study. The graph shows the long-term growth of the index and periods of market volatility. It highlights the dynamic nature of stock market data and the need for accurate prediction models.

Early studies employed statistical regression techniques, including linear regression, polynomial regression, and radial basis function models, using market variables such as open, high, low, and close (OHLC) prices, together with trading volume [13,22]. Among these, Linear Regression provided the best prediction results, demonstrating that simple regression models can effectively capture linear relationships in financial data [22].

Time-series forecasting methods have also been popular for predicting stock prices. The ARIMA model has shown strong results for short-term forecasting using historical stock prices from various exchanges [9]. Model selection utilized statistical measures like standard error, adjusted R^2 , and the Bayesian Information Criterion [9]. However, both regression

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

and ARIMA models rely on linear relationships and stationary data, which limits their effectiveness during rapid market changes or high volatility [20].

Research on stock-market forecasting has examined a variety of supervised learning techniques. Commonly investigated methods include Linear Regression and Multiple Linear Regression, Support Vector Machine and Support Vector Regression, along with tree- and instance-based approaches such as Random Forest, Decision Tree, and K-Nearest Neighbors. Logistic Regression has also been explored for market prediction. Comparative studies show that the performance of these models varies with the dataset, market conditions, prediction horizon, and feature selection. In many cases, Support Vector Regression and Support Vector Machine have achieved higher forecasting accuracy than Linear Regression, K-Nearest Neighbors, Decision Tree, and Random Forest. Linear Regression has remained effective for capturing overall market trends. Moreover, studies have found that proper feature engineering and tuning of hyperparameters can greatly improve prediction performance. However, no single model consistently outperforms the others in all market conditions [15,23,24].

Regression models remain popular because they are straightforward, easy to understand, and can model both linear and nonlinear relationships. Their performance can be improved by combining them with optimization techniques and additional financial features [20]. Hybrid regression methods have also been used for NIFTY 50 prediction, comparing machine learning models such as MARS, Random Forest, XGBoost, ANN, and Support Vector Machine with deep learning models. These studies found that LSTM-based models provide more accurate predictions than traditional machine learning methods [6].

ANNs have been effectively used to model nonlinear relationships in stock market data. Incorporating technical indicators such as RSI, Exponential Moving Average, and Moving Average Convergence Divergence has further improved prediction accuracy by providing more market information to the learning model [6,14,25]. Overall, ANN-based models have outperformed traditional statistical methods in capturing complex market behaviors [20].

Recently, deep learning models have gained considerable attention as methods for forecasting financial markets. These models are capable of capturing patterns over longer sequences. Among them, LSTM networks have consistently surpassed conventional machine learning approaches when forecasting the NIFTY 50 and other market indices. Studies have shown that LSTM generally offers higher forecasting accuracy compared with CNN, Recurrent Neural Networks, and Gated Recurrent Unit (GRU) architectures, although sudden market fluctuations still pose challenges for all deep learning models [7,13,26].

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. *Cureus J Comput Sci* 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

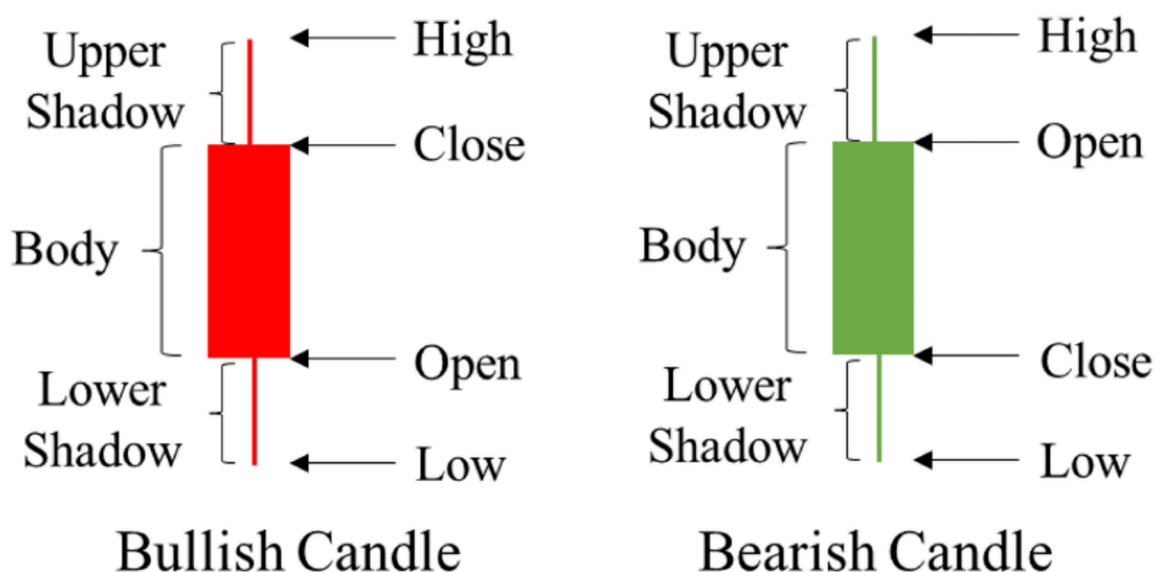


FIGURE 2: Example of a candlestick chart showing open, high, low, and close price movements. The body represents the difference between open and close prices, while the wicks indicate the highest and lowest prices during the trading period

Concept based on the study by Chen et al. [27].

The most popular type of chart used for technical analysis is the candlestick chart. Figure 2 illustrates the basic structure of bullish and bearish candlestick charts. Each candlestick summarizes the OHLC values for a given time period. The candlestick body shows the range between the opening and closing prices, whereas the upper and lower wicks indicate the highest and lowest price levels, respectively. The visualization is helpful for understanding short-term market behavior and is widely used in studies of stock market analysis and prediction [27]. These representations are often used as input for feature engineering in stock price prediction models.

Hybrid deep learning models have further improved forecasting accuracy by combining different neural network structures. CNN-BiLSTM models have been effective in extracting both local and temporal features from historical stock data, yielding better accuracy than single deep learning models and traditional SVM regression [28]. Likewise, hybrid LSTM-GRU models have surpassed standalone CNN and LSTM models for stock price prediction using long-term historical datasets [18]. Recent research has also integrated financial news sentiment with historical market data using hybrid LSTM-GRU frameworks, achieving strong prediction results for the NIFTY 50 index [29]. Machine learning-based trading systems that combine technical indicators with LSTM have also shown improved market trend predictions compared to traditional technical analysis [19].

Recent studies have increasingly leveraged deep learning methods to capture complex patterns in financial time-series data. For instance, comparative evaluations between LSTM models and Temporal Convolutional Network approaches have demonstrated that sequential deep learning models can significantly outperform Naïve Persistence baselines and traditional forecasting benchmarks across financial metrics [30].

While much of the existing literature emphasizes reported predictive accuracy, many studies engage insufficiently with established challenges in financial time series. Financial markets are highly non-stationary and subject to structural breaks, meaning models trained on past data often struggle when market regimes change. Furthermore, financial data exhibits extreme price-level autocorrelation. As a result, distinguishing between true predictive skill and whether a model simply memorizes the previous day's price is difficult, especially when forecasting raw price levels instead of returns or directional movements. Acknowledging these characteristics is essential for establishing rigorous benchmarks.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. *Cureus J Comput Sci* 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

In summary, past studies show that statistical models work well for simple and short-term forecasting. On the other hand, machine learning and deep learning methods usually perform better by capturing complex nonlinear relationships in financial data. However, most existing studies only compare a few models or focus on one forecasting approach. There has been limited research comparing several regression algorithms across different feature engineering techniques, hyperparameter tuning methods, and historical forecasting windows.

To fill this gap, this study offers a detailed comparison of 12 regression algorithms and a Naïve Persistence baseline: Linear Regression, Lasso, Ridge, Elastic Net, Stochastic Gradient Descent Regressor, Support Vector Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and XGBoost. These algorithms are evaluated for forecasting the open and close prices of the NIFTY 50 index using various feature engineering techniques, optimized hyperparameters, and multiple historical forecasting windows. This provides a thorough evaluation of regression-based stock market prediction models.

Methodology

This study aims to evaluate and compare 12 supervised regression models against a Naïve Persistence baseline for next-day open and close price forecasting of the NIFTY 50 index. Two feature pipelines are evaluated: a raw OHLCV pipeline and an engineered-feature pipeline. The models are trained and tested using chronological train-test splitting, ensuring that the original time sequence of the financial observations is maintained. In addition, Random Forest and XGBoost are further optimized using TimeSeriesSplit-based hyperparameter tuning to examine whether tuning improves their generalization performance. The complete methodology and experimental workflow utilized in this study are summarized in Figure 3.

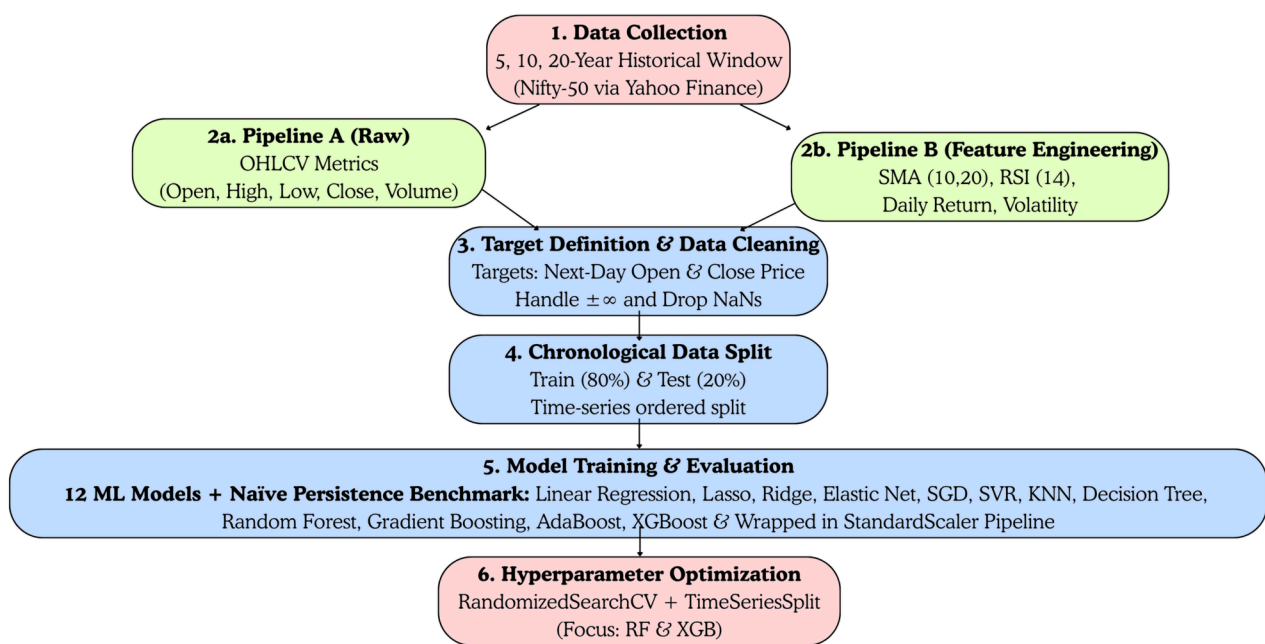


FIGURE 3: Comprehensive machine learning workflow for NIFTY 50 open and close price forecasting, detailing data collection from Yahoo Finance with 5-year, 10-year, and 20-year historical windows, raw and feature-engineered preprocessing pipelines, chronological data splitting, Naïve Persistence baseline evaluation, multi-model evaluation, and hyperparameter optimization

KNN, K-Nearest Neighbors; ML, Machine Learning; RF, Random Forest; RSI, Relative Strength Index; SGD, Stochastic Gradient Descent; SMA, Simple Moving Average; SVR, Support Vector Regression; XGB, Extreme Gradient Boosting

Data Collection

To examine algorithmic performance, historical data for the NIFTY 50 index (^NSEI) were collected using the Yahoo Finance API [17]. Three historical windows were considered in this study. For the 5-year historical window, data from 1 June 2021 to 1 June 2026 were used. For the 10-year historical window, data from 1 June 2016 to 1 June 2026 were used. For the 20-year historical window, data from 1 June 2006 to 1 June 2026 were used. Each dataset contains daily OHLCV data.

Feature Engineering

This study examines whether engineered technical indicators improve the predictive results achieved by machine learning models compared to using only raw market variables. To evaluate this, two feature pipelines are constructed: a raw pipeline acting as a baseline, containing only the original OHLCV variables, and an engineered pipeline.

The engineered pipeline augments the raw data with technical indicators computed from historical closing prices. These specific indicators were selected because they are standard financial metrics widely used in existing literature to reliably capture market trends, momentum, and historical volatility, providing a robust baseline for evaluation [18]. Specifically, these include 10-day and 20-day Simple Moving Average indicators, Daily Return, Rolling Volatility (20 days), and the 14-day RSI [7,19]. Furthermore, two target variables are defined to predict the next day's prices. Here, O_t represents the open price and C_t represents the close price for day t . The engineered features and target variables are summarized in Table 7.

Variable	Feature Definition
Engineered Features	
Simple Moving Average (SMA)	$SMA_{n,t} = \frac{1}{n} \times \sum_{i=0}^{n-1} C_{t-i}, \quad n \in \{10, 20\}$
Rolling Volatility	$\sigma_{n,t} = \sqrt{\frac{1}{n-1} \sum_{i=0}^{n-1} (C_{t-i} - \bar{C}_{n,t})^2}, \quad n = 20$
Daily Return	$R_t = \frac{C_t - C_{t-1}}{C_{t-1}}$
Relative Strength Index (RSI)	$RSI_t = 100 - \frac{100}{1 + RS_t}$
Target Variables	
Next-Day Open	$y_t^{(open)} = O_{t+1}$
Next-Day Close	$y_t^{(close)} = C_{t+1}$

TABLE 1: Engineered features and next-day target variables used in this study

Data Preprocessing and Cleaning

The construction of technical indicators and next-day target variables produced missing entries near the start and end of the dataset. Infinite values, if any, were first replaced with missing values, and incomplete records were removed using dropna(). The input variables were scaled with StandardScaler before model training [13].

Data Splitting

To avoid data leakage, a chronological train-test split was used instead of random sampling. For each dataset (5, 10, and 20 years), an 80:20 chronological split was applied, with the earlier portion used for training and the later portion used for model evaluation. This preserves the temporal order of financial data [13].

Baseline Benchmark, Model Training, and Hyperparameter Optimization

To determine whether candidate machine learning models truly predict market movements beyond just repeating past prices, we introduced a Naïve Persistence baseline as the main benchmark [31]. This baseline suggests that the target price on day $t + 1$ will simply match the observed price on day t :

$$\hat{y}_{t+1} = y_t$$

where $y_t \in \{O_t, C_t\}$ for open and close price targets, respectively.

Notably, training metrics for the Naïve Persistence baseline are completely omitted from our results. Because this baseline does not have a training phase, optimization steps, or learnable parameters, calculating training error is mathematically meaningless. It is applied exclusively to the test set as an out-of-sample benchmark to determine whether the machine learning approaches add predictive value beyond a simple persistence-based forecast.

Alongside the naïve baseline, 12 supervised machine learning regression models were evaluated: Linear Regression, Lasso, Ridge, Elastic Net, Stochastic Gradient Descent Regressor, Support Vector Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and XGBoost.

Hyperparameter tuning was applied selectively to Random Forest and XGBoost. These complex tree-based models are more prone to overfitting and are sensitive to configuration choices, so their settings require careful adjustment. In contrast, the linear models showed stable baseline performance, while full tuning of all remaining models was beyond the scope of this study. To preserve the chronological sequence and eliminate data leakage, tuning was conducted using RandomizedSearchCV [15], wrapped inside a 5-fold TimeSeriesSplit ($n_splits = 5$). The optimal hyperparameter configurations are summarized in Table 2.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Target Variable	Algorithm	Historical Window	n_estimators	max_depth	Regularization/Sa mple Parameters
Open Price	Random Forest	5 Years	100	10	min_samples_split =5, min_samples_leaf= 2
Open Price	Random Forest	10 Years	300	10	min_samples_split =5, min_samples_leaf= 2
Open Price	Random Forest	20 Years	100	10	min_samples_split =5, min_samples_leaf= 2
Open Price	XGBoost	5, 10, 20 Years	200	7	learning_rate=0.05, reg_alpha=10.0, reg_lambda=1.0
Close Price	Random Forest	5, 10, 20 Years	300	10	min_samples_split =5, min_samples_leaf= 2
Close Price	XGBoost	5, 10 Years	200	7	learning_rate=0.05, reg_alpha=10.0, reg_lambda=1.0
Close Price	XGBoost	20 Years	500	7	learning_rate=0.1, reg_alpha=1.0, reg_lambda=10.0

TABLE 2: Optimal hyperparameter configurations across target variables and historical windows

Results And Discussion

Experiments

A scikit-learn pipeline is used to train 12 regression algorithms, including Linear Regression, Lasso, Ridge, Elastic Net, SGD Regressor, SVR, KNN, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, and XGBoost. The overall experimental workflow is shown in Figure 4.

Experimental setup: All experiments and model training were performed using a computer with an 11th Generation Intel Core i3-1125G4 processor running at 2.00 GHz and 8 GB RAM. Python (version 3.13.5) was used for data preparation, feature creation, and model assessment. Historical stock data were retrieved using the yfinance library. Machine learning models and hyperparameter tuning pipelines were developed using the scikit-learn (version 1.6.1) and XGBoost (version 3.0.5) packages.

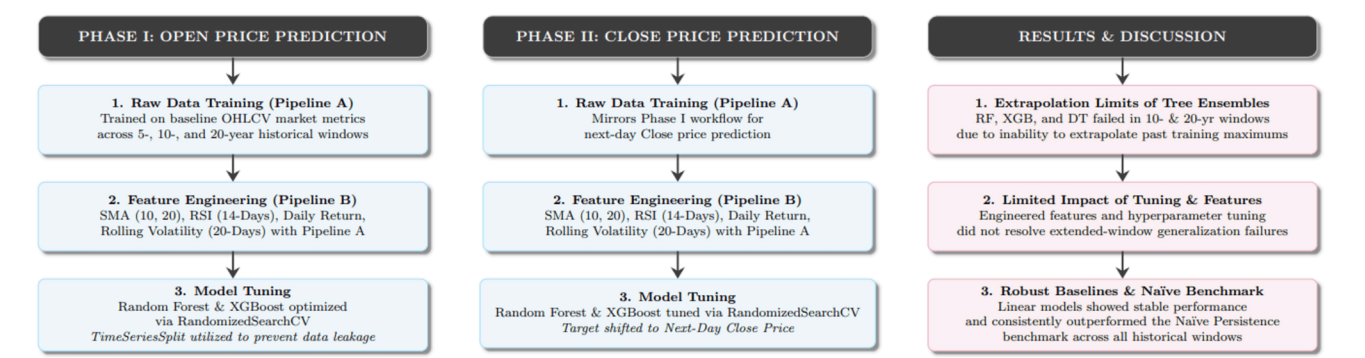


FIGURE 4: Overview of the experimental workflow, illustrating the multi-phase machine learning pipeline for next-day open and close price prediction, model tuning, and multi-window historical evaluation incorporating the naïve benchmark and tree extrapolation findings

DT, Decision Tree; OHLCV, Open, High, Low, Close, Volume; RF, Random Forest; RSI, Relative Strength Index; SMA, Simple Moving Average; XGB, Extreme Gradient Boosting

The experiments are constructed to compare the predictive performance of the 12 regression models for various historical windows and feature configurations. Three experimental settings are considered: baseline prediction using raw OHLCV features, prediction using engineered technical indicators, and prediction after hyperparameter tuning of selected ensemble models.

The evaluation examines the impact of historical data length, feature engineering, and model optimization on predicting next-day NIFTY 50 open and close prices.

Model performance is assessed using three standard evaluation measures: MAE, RMSE, and R^2 [13]. MAE represents the mean magnitude of the prediction errors, whereas RMSE places greater emphasis on larger deviations. The R^2 value reflects how much of the variation in stock prices is explained by the model [13,18]. The mathematical formulas are presented in Table 3.

Metric	Mathematical Formulation
Mean Absolute Error (MAE)	$MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Coefficient of Determination (R^2)	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

TABLE 3: Mathematical formulations of the regression performance evaluation metrics employed in this study

Forecasting Next-Day Open Price

Each model was evaluated using 5-year, 10-year, and 20-year historical windows with chronological splits to examine performance under different training-window lengths.

Baseline performance (raw OHLCV): Table 4 presents the baseline performance of the models for next-day open price forecasting using raw OHLCV features. In the 5-year historical window, Linear Regression, Ridge, Lasso, Gradient Boosting, Random Forest, and SGD Regressor achieved strong test performance. For the 10-year and 20-year historical windows, Linear Regression, Ridge, SGD Regressor, and Lasso showed more stable generalization than the remaining models. In contrast, SVR and several tree-based models showed weak test performance, especially in longer historical windows.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning

Historical Window	Model	Train R ²	Train RMSE	Train MAE	Test R ²	Test RMSE	Test MAE
5 Years	Linear Regression	0.9988	105.74	69.34	0.9658	159.62	90.09
	Ridge Regression	0.9986	113.85	76.82	0.9601	172.34	101.30
	Lasso Regression	0.9985	115.15	77.90	0.9599	172.81	101.93
	Gradient Boosting	0.9994	74.26	52.02	0.9550	183.14	117.65
	Random Forest	0.9998	45.91	31.14	0.9547	183.76	117.56
	SGD Regressor	0.9981	130.99	91.60	0.9522	188.59	117.73
	XGBoost	0.9999	23.14	16.42	0.9395	212.33	148.71
	Naïve Persistence	-	-	-	0.9247	236.80	164.24
	Decision Tree	1.0000	0.00	0.00	0.9358	218.71	143.54
	AdaBoost	0.9958	192.88	151.94	0.8884	288.33	223.22
	KNN	0.9964	178.52	114.05	0.8161	370.09	269.92
	Elastic Net	0.9856	359.34	304.31	0.5230	596.02	559.82
	SVR	0.0100	2975.12	2345.81	-48.8487	6092.79	6030.39
10 Years	Linear Regression	0.9996	81.29	50.92	0.9770	150.89	86.06
	Ridge Regression	0.9995	89.13	57.92	0.9721	166.19	100.72
	SGD Regressor	0.9994	100.58	67.28	0.9659	183.58	117.71
	Lasso Regression	0.9993	107.54	72.84	0.9626	192.29	125.56

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning

	Naïve Persistence	-	-	-	0.9411	241.39	167.65
	Elastic Net	0.9870	466.10	411.67	-0.4989	1217.49	1201.08
	Decision Tree	1.0000	0.00	0.00	-2.2373	1789.27	1544.02
	Random Forest	0.9999	34.81	22.18	-2.3956	1832.49	1588.53
	Gradient Boosting	0.9998	62.77	43.29	-2.4458	1846.00	1603.24
	XGBoost	0.9999	33.15	23.72	-3.0999	2013.59	1783.10
	KNN	0.9991	124.13	79.36	-3.1544	2026.94	1811.43
	AdaBoost	0.9977	194.97	145.47	-5.1154	2459.22	2254.23
	SVR	0.0648	3953.39	3045.96	-150.6748	12247.33	12201.97
20 Years	Linear Regression	0.9998	57.77	31.20	0.9982	122.95	72.74
	Ridge Regression	0.9997	62.80	37.21	0.9978	134.38	83.41
	SGD Regressor	0.9996	73.96	47.22	0.9972	152.95	100.64
	Lasso Regression	0.9995	84.92	56.11	0.9966	168.58	113.76
	Naïve Persistence	-	-	-	0.9950	203.85	142.97
	Elastic Net	0.9859	447.57	328.99	0.5428	1943.09	1895.22
	Random Forest	1.0000	24.08	13.59	-1.6416	4670.78	3836.93
	Gradient Boosting	0.9998	49.38	33.20	-1.6425	4671.60	3837.65
	Decision Tree	1.0000	0.00	0.00	-1.6815	4705.93	3878.31
	KNN	0.9995	84.46	46.55	-1.7767	4788.74	3977.70
	XGBoost	0.9999	33.06	23.21	-1.8482	4849.94	4025.03
	AdaBoost	0.9974	190.39	157.33	-2.2602	5188.91	4370.42

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

	SVR	0.2921	3168.10	2241.36	-23.5718	14245.33	13923.15
--	-----	--------	---------	---------	----------	----------	----------

TABLE 4: Comprehensive evaluation metrics for NIFTY 50 open price prediction across different historical windows (baseline raw OHLCV)

KNN, K-Nearest Neighbors; MAE, Mean Absolute Error; OHLCV, Open, High, Low, Close, Volume; RMSE, Root Mean Square Error; SGD, Stochastic Gradient Descent; SVR, Support Vector Regression

The trend is also visible in Figure 5, where several models track short-term movements well but struggle in extended historical-window settings.

How to cite this article:

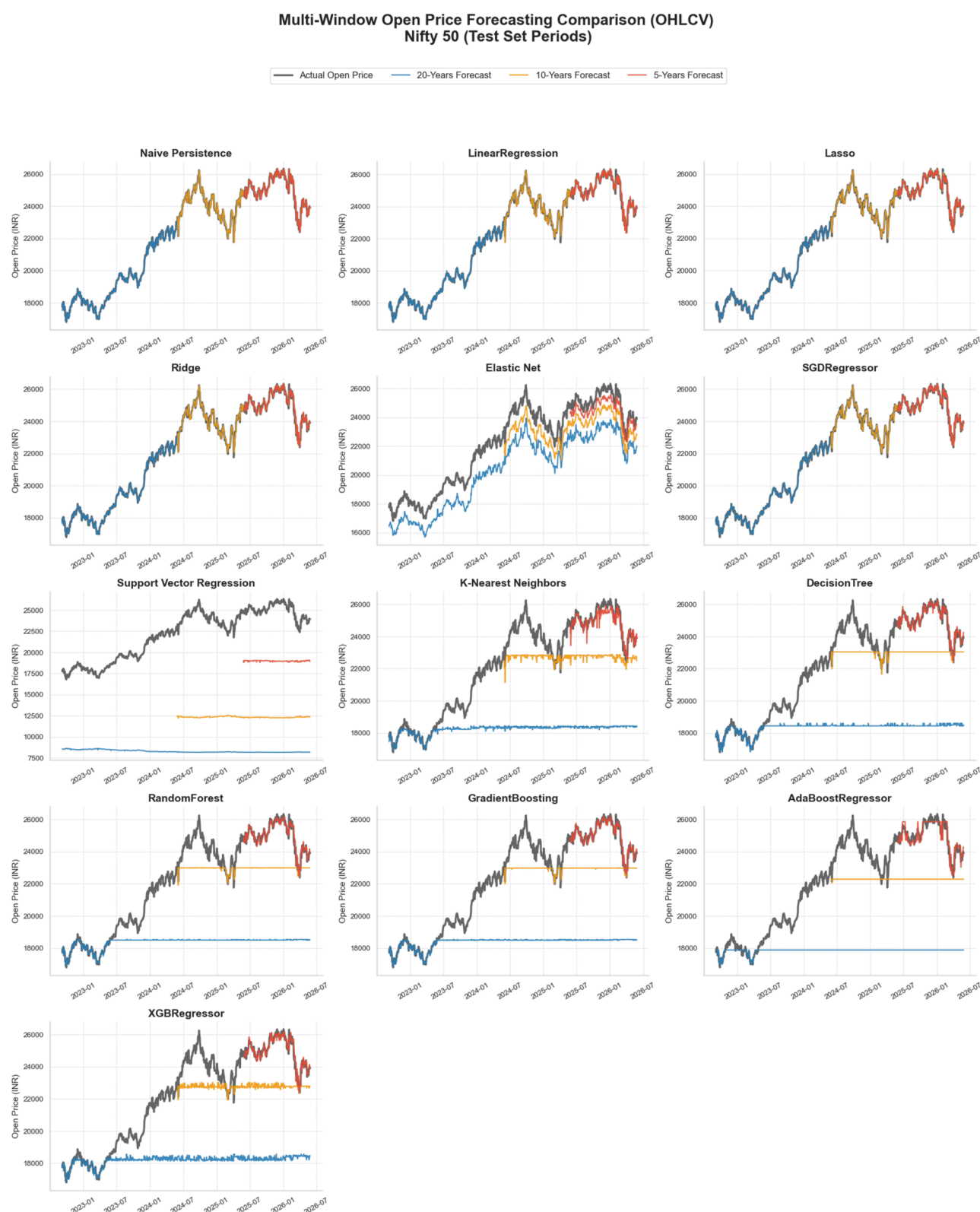


FIGURE 5: Actual vs. predicted open prices for the NIFTY 50 index for all historical windows

INR, Indian Rupee; SGD, Stochastic Gradient Descent; XGB, Extreme Gradient Boosting

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

The impact of feature engineering: Table 5 shows the performance of different models after adding engineered features such as SMA, rolling volatility, RSI, and daily return. For the 5-year historical window, feature engineering did not provide a clear improvement over raw OHLCV features. For the 10-year historical window, some linear models showed improvement in terms of reduced RMSE. However, for the 20-year historical window, engineered features did not consistently improve model performance. Overall, feature engineering mainly benefited selected linear models in the 10-year setting.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Historical Window	Model	Train R ²	Train RMSE	Train MAE	Test R ²	Test RMSE	Test MAE
5 Years	Linear Regression	0.9987	106.11	69.80	0.9657	161.08	90.96
	Ridge Regression	0.9986	113.06	75.57	0.9566	181.05	105.38
	Lasso Regression	0.9984	118.60	79.70	0.9532	188.08	108.26
	SGD Regressor	0.9984	120.48	82.63	0.9511	192.28	116.76
	Random Forest	0.9998	45.67	30.46	0.9497	194.97	130.97
	Gradient Boosting	0.9995	69.07	48.94	0.9422	209.11	143.10
	Naïve Persistence	-	-	-	0.9248	238.45	165.80
	XGBoost	1.0000	8.04	5.78	0.9101	260.72	189.08
	Decision Tree	1.0000	0.00	0.00	0.9054	267.37	186.41
	AdaBoost	0.9946	218.93	175.41	0.8625	322.40	245.76
	Elastic Net	0.9920	266.83	216.86	0.7337	448.71	409.36
	KNN	0.9939	232.11	165.65	0.5587	577.62	479.41
	SVR	-0.0383	3031.05	2420.24	-47.8984	6079.98	6019.04
10 Years	Linear Regression	0.9996	83.23	51.46	0.9783	145.24	83.25
	Ridge Regression	0.9995	90.50	57.29	0.9721	164.47	99.98
	SGD Regressor	0.9994	102.87	68.09	0.9665	180.41	117.85
	Lasso Regression	0.9993	104.92	67.03	0.9626	190.50	122.50

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning

	Naïve Persistence	-	-	-	0.9431	234.99	164.87
	Elastic Net	0.9932	338.19	284.04	0.2876	831.47	804.17
	Gradient Boosting	0.9998	60.28	42.26	-2.0167	1711.03	1468.12
	Random Forest	0.9999	35.05	22.48	-2.2124	1765.67	1521.69
	Decision Tree	1.0000	0.00	0.00	-2.2865	1785.92	1543.73
	XGBoost	1.0000	19.07	14.04	-3.0401	1980.12	1741.49
	KNN	0.9974	209.94	148.37	-4.4764	2305.37	2157.36
	AdaBoost	0.9978	192.67	142.32	-4.5179	2314.10	2101.64
	SVR	-0.0016	4091.61	3173.72	-154.8147	12296.97	12255.35
20 Years	Linear Regression	0.9998	57.74	31.62	0.9982	122.98	72.69
	Ridge Regression	0.9997	62.89	35.88	0.9977	137.63	85.89
	SGD Regressor	0.9996	70.94	42.62	0.9972	151.29	99.95
	Lasso Regression	0.9996	76.24	44.60	0.9964	171.51	114.67
	Naïve Persistence	-	-	-	0.9949	204.00	142.95
	Elastic Net	0.9925	327.59	240.66	0.7741	1362.16	1321.95
	Random Forest	1.0000	24.40	13.62	-1.6901	4700.49	3871.85
	Decision Tree	1.0000	0.00	0.00	-1.6919	4702.06	3877.89
	Gradient Boosting	0.9998	48.68	32.95	-1.7108	4718.53	3889.56
	XGBoost	1.0000	20.56	15.03	-1.9390	4913.20	4088.46
	KNN	0.9981	165.52	116.59	-2.0629	5015.62	4235.71
	AdaBoost	0.9974	193.67	160.47	-2.2975	5204.23	4394.46

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

	SVR	0.2013	3376.31	2510.22	-23.9484	14314.70	14000.96
--	-----	--------	---------	---------	----------	----------	----------

TABLE 5: Comprehensive evaluation metrics for NIFTY 50 open price prediction across different historical windows (engineered features)

KNN, K-Nearest Neighbors; MAE, Mean Absolute Error; RMSE, Root Mean Square Error; SGD, Stochastic Gradient Descent; SVR, Support Vector Regression

The trend is visible in Figure 6, where linear models follow the overall market trend more accurately, while ensemble models capture short-term fluctuations but struggle in extended historical-window settings.

How to cite this article:

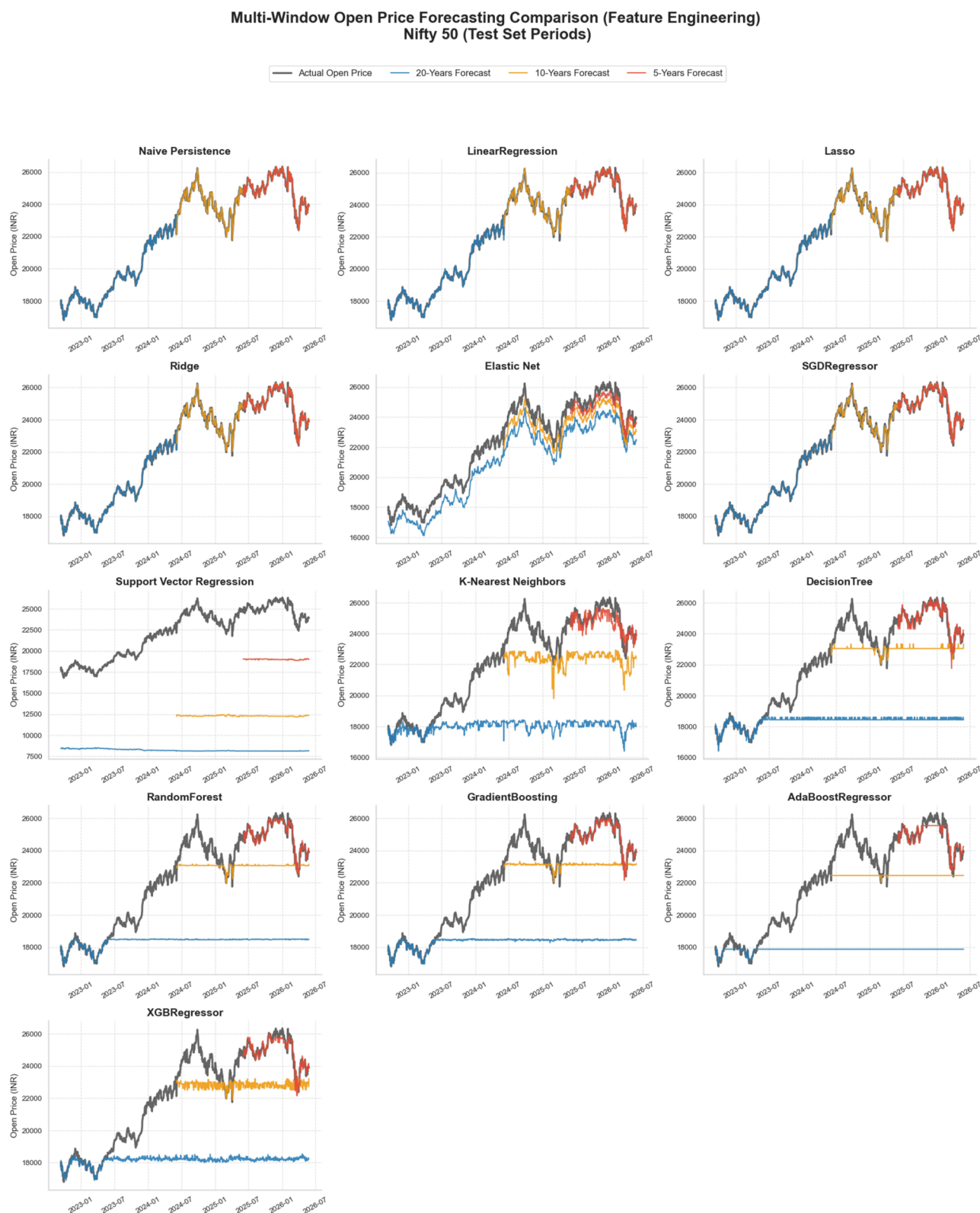


FIGURE 6: Actual vs. predicted open prices for the NIFTY 50 index using engineered features for all historical windows

INR, Indian Rupee; SGD, Stochastic Gradient Descent; XGB, Extreme Gradient Boosting

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Hyperparameter tuning: Hyperparameter tuning was applied to Random Forest and XGBoost using TimeSeriesSplit to reduce overfitting. We added stronger regularization, such as L1/L2 penalties in XGBoost, and limited model complexity, for example, by restricting max depth to 7-10, to encourage the models to learn general patterns instead of memorizing past data. Table 6 shows the results. Even after tuning, both models still achieved strong training performance ($R^2 > 0.999$) across all historical windows, but showed weak test performance on the 10-year and 20-year test sets, with strongly negative R^2 values. For the 5-year historical window, Random Forest achieved a test R^2 score of 0.9547 and an RMSE value of 183.70, while XGBoost achieved a test R^2 score of 0.9435 and an RMSE value of 205.05. For XGBoost, the test RMSE improved from 212.33 in the raw OHLCV setting to 205.05, even though the test R^2 score did not show the same clear improvement. However, for the 10-year and 20-year historical windows, the test R^2 scores had strongly negative values and large RMSE values. Figure 7 shows the performance of hyperparameter tuning.

Historical Window	Model	Train R^2	Test R^2	Test RMSE
5 Years	Random Forest	0.9995	0.9547	183.70
5 Years	XGBoost	0.9998	0.9435	205.05
10 Years	Random Forest	0.9999	-2.5985	1886.44
10 Years	XGBoost	0.9999	-3.0736	2007.12
20 Years	Random Forest	0.9999	-1.6847	4708.72
20 Years	XGBoost	0.9999	-1.8534	4854.38

TABLE 6: Performance of tuned random forest and XGBoost models for open price forecasting

RMSE, Root Mean Square Error

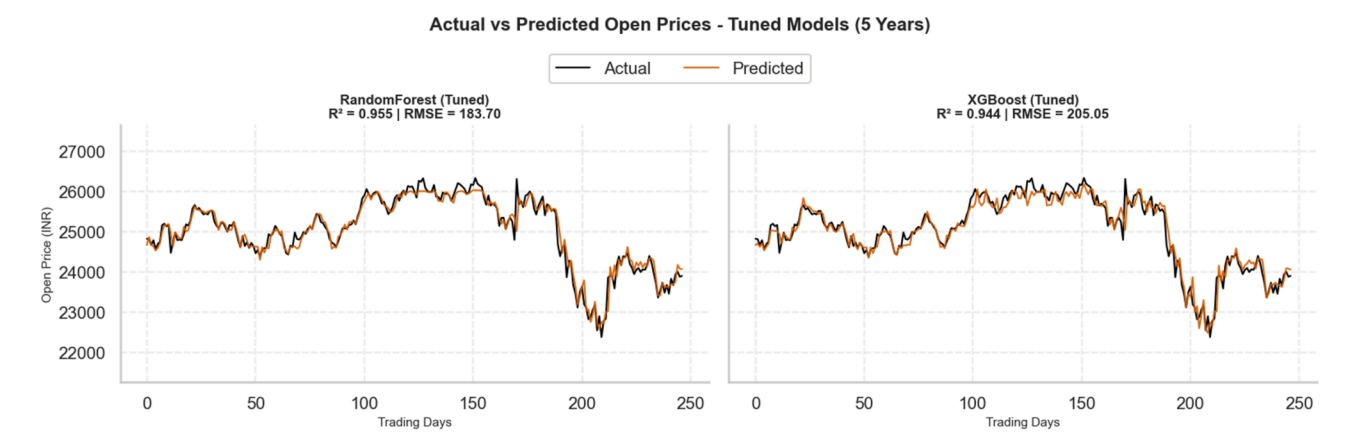


FIGURE 7: Actual vs. predicted open prices for the NIFTY 50 index using hyperparameter tuning (5-year historical window)

INR, Indian Rupee; RMSE, Root Mean Square Error

How to cite this article:

Summary of Phase I findings: The study moved step-by-step from raw OHLCV data to feature engineering and then hyperparameter tuning to test model limits in forecasting.

Linear Regression, Ridge, Lasso, and SGD Regressor showed stable test performance across the historical windows. Tree-based and ensemble models performed competitively in the 5-year setting but showed weak test performance in the 10-year and 20-year settings despite strong training performance. Feature engineering provided limited improvement, with some benefit for Random Forest in the 5-year setting and for linear models in the 10-year setting. Overall, the results suggest that linear models provide more consistent generalization for next-day NIFTY 50 open price forecasting under the present experimental setup.

Overall, while complex models work well when trained on shorter historical windows, Linear Regression performs better for next-day price forecasting when using 10-year and 20-year historical windows.

Forecasting Next-Day Close Price

Predicting the next day's closing price is important because it shows the final market value after a full day of trading. Our results show that tree-based models work well with 5 years of data but perform poorly over 10 and 20 years because they cannot predict prices higher than the peak prices seen in their training history. In contrast, models that follow linear trends perform much better at predicting the next day's price when trained on longer historical datasets.

Baseline performance (raw OHLCV): Table 7 shows predictive results of various baseline approaches across the 5-year, 10-year, and 20-year historical windows for next-day close price forecasting. For the 5-year historical window, Linear Regression, Lasso, Ridge, Gradient Boosting, SGD Regressor, and Random Forest outperform the other models. XGBoost and AdaBoost also show reasonable performance, while Support Vector Regression shows poor training and testing results. For the 10-year and 20-year historical windows, Linear Regression, Ridge, SGD Regressor, and Lasso achieve better performance and outperform the remaining models.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Historical Window	Model	Train R ²	Train RMSE	Train MAE	Test R ²	Test RMSE	Test MAE
5 Years	Linear Regression	0.9966	173.90	126.30	0.9479	195.77	145.98
	Lasso Regression	0.9964	178.19	129.22	0.9434	204.14	151.12
	Ridge Regression	0.9964	178.16	129.18	0.9424	205.90	152.31
	Gradient Boosting	0.9981	129.20	97.80	0.9372	214.99	165.64
	SGD Regressor	0.9961	187.73	136.95	0.9353	218.18	162.39
	Random Forest	0.9994	71.63	52.36	0.9285	229.40	170.42
	Naïve Persistence	-	-	-	0.9117	254.89	191.78
	XGBoost	0.9998	38.87	27.57	0.9085	259.52	202.56
	AdaBoost	0.9946	219.69	172.21	0.8778	299.88	233.93
	Decision Tree	1.0000	0.00	0.00	0.8343	349.24	251.29
	KNN	0.9950	210.84	144.56	0.7912	391.95	299.02
	Elastic Net	0.9837	382.08	314.51	0.4990	607.16	568.45
	SVR	0.0115	2972.72	2347.15	-49.3422	6086.54	6024.91
10 Years	Linear Regression	0.9989	133.05	93.01	0.9527	215.56	152.81
	Ridge Regression	0.9989	136.83	96.74	0.9489	224.15	159.21
	SGD Regressor	0.9988	142.97	101.47	0.9435	235.61	169.51
	Lasso Regression	0.9987	147.98	105.52	0.9415	239.89	173.26

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning

	Naïve Persistence	-	-	-	0.9228	275.57	203.62
	Elastic Net	0.9864	476.80	417.56	-0.5338	1228.05	1204.39
	Decision Tree	1.0000	0.00	0.00	-2.7169	1911.75	1677.99
	Random Forest	0.9998	54.11	38.22	-2.7193	1912.35	1677.50
	Gradient Boosting	0.9994	103.78	76.66	-2.7745	1926.51	1692.23
	KNN	0.9986	153.56	103.92	-3.4382	2089.01	1874.44
	XGBoost	0.9998	53.56	39.34	-3.6559	2139.65	1921.74
	AdaBoost	0.9973	212.77	168.08	-5.1022	2449.53	2245.83
	SVR	0.0618	3960.01	3046.27	-151.9414	12263.14	12218.14
20 Years	Linear Regression	0.9992	104.50	70.60	0.9960	182.35	130.60
	Ridge Regression	0.9992	107.02	73.45	0.9958	187.21	134.14
	SGD Regressor	0.9991	113.63	79.12	0.9953	197.53	142.86
	Lasso Regression	0.9990	120.94	84.71	0.9948	207.47	151.50
	Naïve Persistence	-	-	-	0.9934	233.82	172.75
	Elastic Net	0.9854	455.34	332.76	0.5370	1956.37	1904.93
	Gradient Boosting	0.9995	87.05	63.75	-1.8454	4849.96	4024.47
	Random Forest	0.9999	42.90	29.44	-1.8862	4884.63	4062.37
	Decision Tree	1.0000	0.00	0.00	-1.9008	4896.97	4088.85
	XGBoost	0.9997	60.67	45.59	-1.9245	4916.93	4098.39
	KNN	0.9991	111.81	73.14	-1.9269	4918.97	4101.94
	AdaBoost	0.9968	212.22	170.68	-2.3052	5227.17	4416.47

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

	SVR	0.2928	3163.91	2238.95	-23.5084	14233.95	13911.21
--	-----	--------	---------	---------	----------	----------	----------

TABLE 7: Comprehensive evaluation metrics for NIFTY 50 close price prediction across different historical windows (baseline raw OHLCV)

KNN, K-Nearest Neighbors; MAE, Mean Absolute Error; OHLCV, Open, High, Low, Close, Volume; RMSE, Root Mean Square Error; SGD, Stochastic Gradient Descent; SVR, Support Vector Regression

The trend is visible in Figure 8, where several models track short-term movements well but struggle to capture long-term market trends.

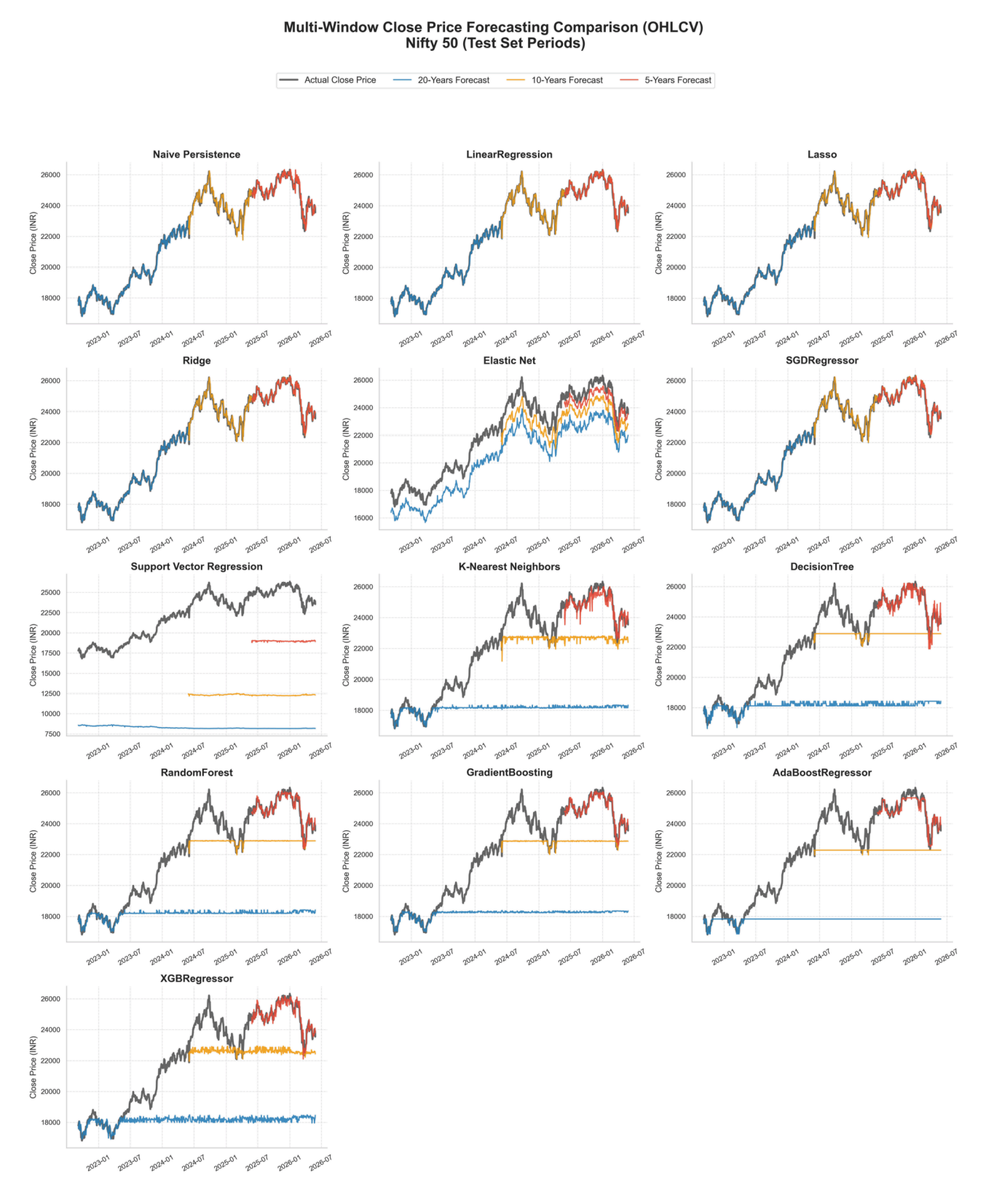


FIGURE 8: Actual vs. predicted close prices for the NIFTY 50 index for all historical windows
INR, Indian Rupee; OHLCV, Open, High, Low, Close, Volume; SGD, Stochastic Gradient Descent; XGB, Extreme Gradient Boosting

The impact of feature engineering: Table 8 presents the performance of the models after adding engineered features for next-day close price forecasting. For the 5-year historical window, Random Forest showed some improvement after feature engineering. For the 10-year historical window, linear models showed improvement in terms of reduced RMSE, while most non-linear models continued to show weak test performance. For the 20-year historical window, engineered features did not provide consistent improvement over raw OHLCV features. Hence, Random Forest showed improvement in the 5-year historical window, while feature engineering benefited linear models in the 10-year historical window.

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Historical Window	Model	Train R ²	Train RMSE	Train MAE	Test R ²	Test RMSE	Test MAE
5 Years	Linear Regression	0.9965	174.88	127.86	0.9473	198.27	145.72
	Ridge Regression	0.9964	178.88	128.17	0.9388	213.74	154.65
	Lasso Regression	0.9963	181.68	128.67	0.9380	215.13	158.37
	Random Forest	0.9994	70.17	50.08	0.9352	219.86	168.73
	SGD Regressor	0.9961	185.48	133.90	0.9309	227.06	160.41
	Naïve Persistence	-	-	-	0.9117	256.73	193.71
	Gradient Boosting	0.9984	120.14	90.71	0.9103	258.70	200.43
	AdaBoost	0.9951	207.52	163.70	0.8972	277.00	212.64
	Decision Tree	1.0000	0.00	0.00	0.8583	325.21	253.89
	XGBoost	1.0000	12.31	8.81	0.8121	374.46	296.61
	Elastic Net	0.9899	299.26	236.44	0.7052	469.10	421.52
	KNN	0.9927	254.44	185.35	0.5615	572.07	483.91
10 Years	SVR	-0.0376	3029.57	2421.42	-48.4953	6078.17	6018.08
	Linear Regression	0.9988	138.71	94.85	0.9578	201.08	147.51
	Ridge Regression	0.9988	141.76	96.60	0.9519	214.63	156.63
	SGD Regressor	0.9987	149.05	102.60	0.9448	229.86	169.81
	Lasso Regression	0.9987	149.72	101.23	0.9436	232.52	170.12

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning

	Naïve Persistence	-	-	-	0.9263	265.70	199.57
	Elastic Net	0.9925	354.26	294.71	0.2501	847.60	812.12
	Gradient Boosting	0.9994	100.32	74.10	-4.1411	2219.35	2005.01
	XGBoost	1.0000	28.87	20.77	-4.3891	2272.24	2074.99
	KNN	0.9970	224.18	158.90	-4.5564	2307.25	2158.45
	Random Forest	0.9998	53.29	36.88	-4.7554	2348.21	2130.68
	AdaBoost	0.9973	210.90	160.53	-4.9510	2387.77	2182.68
	Decision Tree	1.0000	0.00	0.00	-7.1013	2785.96	2563.83
	SVR	-0.0041	4096.35	3173.23	-157.2180	12311.91	12270.90
20 Years	Linear Regression	0.9992	104.43	70.49	0.9959	182.86	130.66
	Ridge Regression	0.9992	106.36	71.97	0.9956	190.08	135.95
	SGD Regressor	0.9991	111.49	76.16	0.9951	200.64	146.95
	Lasso Regression	0.9991	113.78	76.69	0.9946	211.32	152.32
	Naïve Persistence	-	-	-	0.9933	234.06	172.90
	Elastic Net	0.9920	337.79	245.79	0.7695	1376.52	1331.05
	Decision Tree	1.0000	0.00	0.00	-1.9152	4895.89	4085.51
	Gradient Boosting	0.9995	83.59	61.25	-1.9533	4927.77	4108.96
	Random Forest	0.9999	41.96	28.53	-2.0581	5014.42	4197.05
	KNN	0.9978	177.92	125.52	-2.1535	5092.00	4305.52
	XGBoost	0.9999	37.63	28.22	-2.1778	5111.59	4298.35
	AdaBoost	0.9972	199.34	163.83	-2.2920	5202.63	4395.92

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5.
DOI <https://doi.org/10.7759/s44389-026-00306-5>

	SVR	0.2024	3371.45	2507.20	-23.8654	14298.56	13984.13
--	-----	--------	---------	---------	----------	----------	----------

TABLE 8: Comprehensive evaluation metrics for NIFTY 50 close price prediction across different historical windows (engineered features)

KNN, K-Nearest Neighbors; MAE, Mean Absolute Error; RMSE, Root Mean Square Error; SGD, Stochastic Gradient Descent; SVR, Support Vector Regression

This trend is clearly visible in Figure 9, where the linear models follow the long-term market trend more accurately, while ensemble models capture short-term fluctuations but fail to predict long-term growth.

How to cite this article:

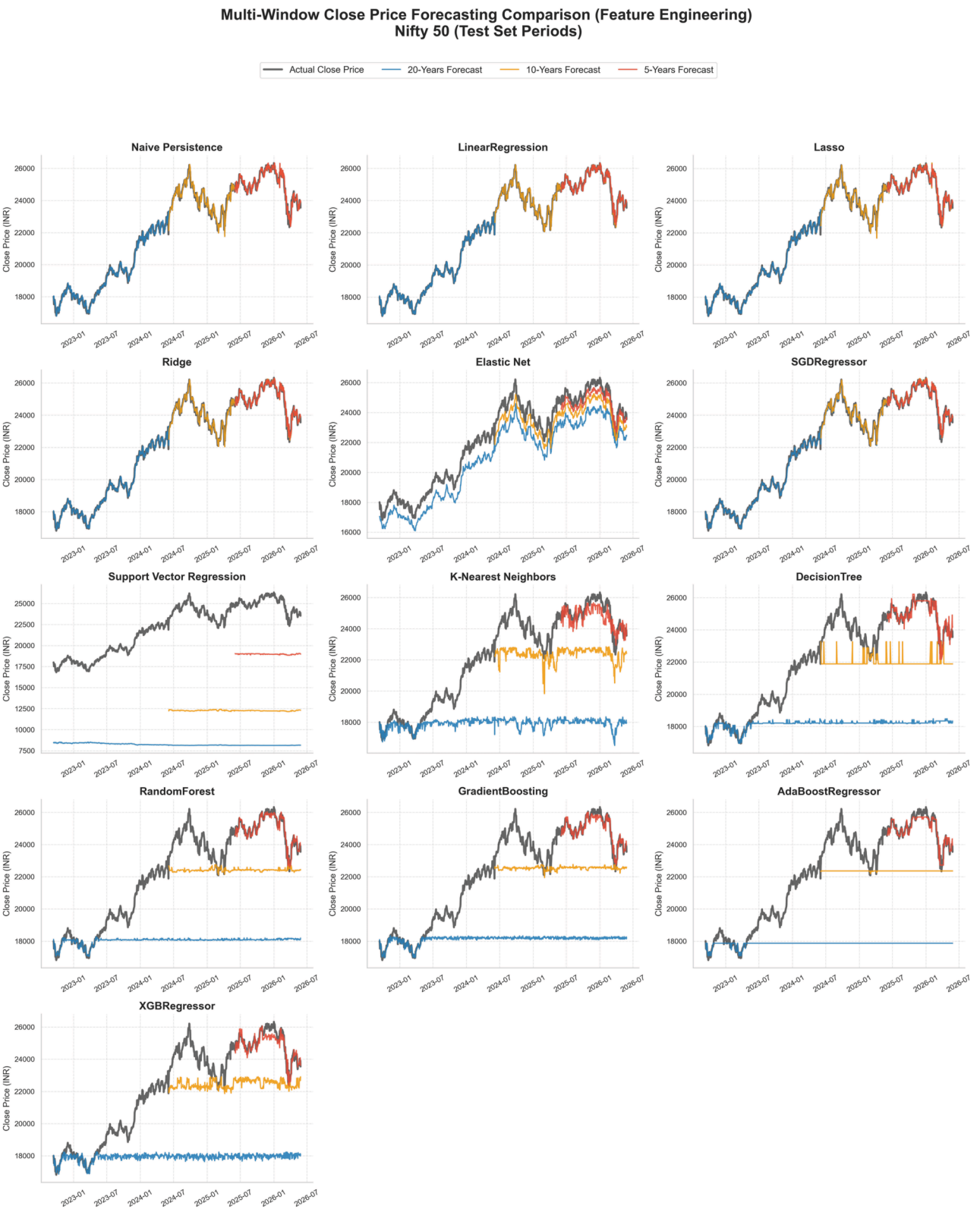


FIGURE 9: Actual vs. predicted close prices for the NIFTY 50 index using engineered features for all historical windows

INR, Indian Rupee; SGD, Stochastic Gradient Descent; XGB, Extreme Gradient Boosting

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Hyperparameter tuning: Hyperparameter tuning was applied to Random Forest and XGBoost using TimeSeriesSplit to reduce overfitting for close price prediction. Similar to Phase I, stronger regularization, such as L1/L2 penalties in XGBoost, and limited model complexity, such as restricting max depth to 7-10, were used to encourage the models to learn general patterns instead of memorizing past data. Table 9 shows the results. Even after tuning, both models achieved strong training performance ($R^2 > 0.999$) but showed weak test performance on the 10-year and 20-year test sets, with strongly negative R^2 values. For the 5-year historical window, Random Forest achieved a test R^2 score of 0.9337 and an RMSE value of 220.87, while XGBoost achieved a test R^2 score of 0.9198 and an RMSE value of 243.00. For XGBoost, the RMSE improved compared with the raw OHLCV setting, although the R^2 improvement was not equally clear. However, for the 10-year and 20-year historical windows, both models showed strongly negative R^2 values and large RMSE values. Figure 10 shows the predictions of the tuned models on the 5-year historical window.

Historical Window	Model	Train R^2	Test R^2	Test RMSE
5 Years	Random Forest	0.9988	0.9337	220.87
5 Years	XGBoost	0.9994	0.9198	243.00
10 Years	Random Forest	0.9996	-2.8811	1953.50
10 Years	XGBoost	0.9997	-3.7190	2154.10
20 Years	Random Forest	0.9997	-1.8891	4887.11
20 Years	XGBoost	0.9997	-1.9453	4934.42

TABLE 9: Performance of tuned random forest and XGBoost models for close price forecasting

RMSE, Root Mean Square Error

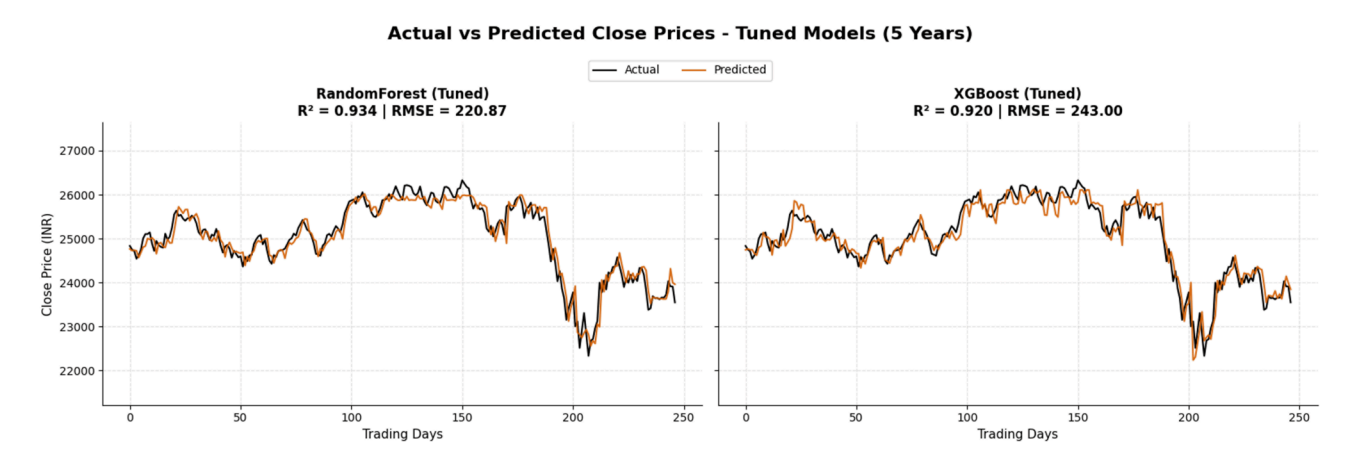


FIGURE 10: Actual vs. predicted close prices for the NIFTY 50 index using hyperparameter tuning (5-year historical window)

INR, Indian Rupee; RMSE, Root Mean Square Error

How to cite this article:

Summary of Phase II findings: The study moved step-by-step from raw OHLCV data to feature engineering and then to hyperparameter tuning to examine the model limits in close price forecasting. Raw data showed good predictive performance over the 5-year historical window, but weaker generalization across the extended 10-year and 20-year periods for several non-linear models. Feature engineering improved performance for the 10-year dataset but did not fully resolve the generalization issue for the 20-year dataset. Even after tuning, ensemble models continued to perform poorly in the 10-year and 20-year settings. The results indicate that linear model performance improved as the historical window increased, whereas non-linear model performance declined in longer historical windows. For example, in the baseline raw OHLCV setting, Linear Regression achieved an R^2 value of 0.9479 and an RMSE value of 195.77 for the 5-year window, while it recorded an R^2 value of 0.9960, together with an RMSE value of 182.35 for the 20-year window. A similar trend was observed with engineered features. Random Forest showed improvement in the 5-year historical window after feature engineering and tuning. Overall, complex models worked well when using shorter historical data, whereas Linear Regression performed better at next-day forecasting when using 10-year and 20-year historical windows.

Results

This section summarizes the major findings obtained from the open and close price forecasting experiments. The results are discussed with respect to model complexity, feature settings, historical window length, and the effect of hyperparameter tuning.

Generalization limitations of complex non-linear models: The results show that complex non-linear models, including tree-based ensemble approaches (Random Forest, XGBoost) and decision-tree approaches (Decision Tree), performed competitively in the 5-year historical window but showed weak generalization in the 10-year and 20-year historical windows. These models achieved very high training performance, but their test performance declined substantially in longer windows. This indicates that tree-based models may capture short-term patterns effectively but may struggle when the test data contain price ranges or market conditions that are different from those in the training data.

Limited effect of feature engineering and tuning: Using calculated technical indicators such as SMA, RSI, daily return, and rolling volatility provided limited improvement compared with raw OHLCV features. Feature engineering improved performance in some settings, particularly for selected linear models and Random Forest in the 5-year close price experiment. However, it did not consistently improve performance across all models and historical windows. Similarly, TimeSeriesSplit-based tuning of Random Forest and XGBoost improved short-window performance in some cases but did not fully resolve the weak generalization observed in longer historical windows.

Stable performance of linear models: Linear Regression, Ridge, and Lasso showed more stable performance across both open and close price forecasting tasks. In particular, Linear Regression achieved strong test performance across the 5-year, 10-year, and 20-year historical windows. Overall, the findings indicate that simple linear models are capable of capturing the general trend in NIFTY 50 price movements under the considered experimental setting. Therefore, linear models can serve as strong baseline methods before applying more sophisticated machine-learning techniques for stock-index prediction.

Comparing models to the naïve baseline: To check if the models really learned market patterns, we compared them to a "Naïve Persistence" baseline $\hat{y}_{t+1} = y_t$, which assumes tomorrow's price will be the same as today's. Since stock prices rarely change drastically overnight, this simple guess leads to very high accuracy. For instance, in the 20-year dataset, the naïve persistence approach recorded an R^2 value of 0.9950, along with an MAE of 142.97.

This comparison offers two important insights. First, the extremely high R^2 scores (often over 0.99) in linear models mostly show this natural day-to-day market stability, rather than finding complex rules. Still, Linear Regression showed additional predictive value over the naïve strategy, cutting the absolute error (MAE) in half to 72.74 in the same 20-year window.

Second, the baseline clearly shows the long-term shortcomings of complex algorithms. While models like Random Forest and XGBoost outperformed the naïve guess in the short 5-year window, they did poorly in the 10- and 20-year datasets. In the longer timeframes, these complex models performed much worse than simply guessing yesterday's price, resulting

How to cite this article:

in negative R^2 values and huge errors ($MAE > 3,800$). Ultimately, only the simple linear models consistently did better than the basic naïve benchmark across all timeframes.

Extrapolation limitations of tree ensembles: Decision trees (and models like Random Forest or XGBoost) predict numbers by taking averages of past training data. Because of this, they can never predict a number higher than the highest price they saw during training. Over 10 or 20 years, the NIFTY 50 index grew to new all-time highs. When tested on these newer, higher prices, tree models hit a ceiling. Instead of predicting the upward trend, they just output the highest historical price over and over (a flat line). This creates massive errors, causing the R^2 score to drop deep into negative numbers.

Conclusions

This study compared 12 supervised machine learning models alongside a Naïve Persistence baseline for next-day open and close price forecasting of the NIFTY 50 index across 5-year, 10-year, and 20-year historical training windows. The models were evaluated using raw OHLCV variables, engineered technical indicators, and optimized settings for Random Forest and XGBoost. The experimental results demonstrate that Linear Regression, Ridge, and Lasso provided the most stable generalization across extended historical training windows. Notably, these simple linear models were the only algorithms to consistently outperform the naïve benchmark (simply guessing yesterday's price). In contrast, more complex non-linear machine learning approaches, including Random Forest, XGBoost, and Decision Tree, performed competitively in the 5-year setting but showed poor generalization in the 10-year and 20-year datasets, often performing worse than the Naïve Persistence baseline. This severe degradation occurs because tree-based architectures cannot extrapolate beyond the maximum price levels observed during their training periods. Furthermore, feature engineering and hyperparameter tuning offered limited improvements and did not resolve the weak generalization of complex models over longer historical windows.

These findings show that simple linear models should serve as essential baselines for next-day NIFTY 50 price forecasting. However, these conclusions are strictly limited to the specific datasets, target variables, chronological 80/20 splits, and assessment methods applied in this study. The results should not be generalized as proof that linear models are universally superior across all financial forecasting problems or differing market regimes. Future research should address the limitations of this experimental design by employing rolling-origin or walk-forward validation, along with statistical significance testing of forecast errors and uncertainty estimates, to better handle market non-stationarity and structural breaks. Specifically, because the 80/20 chronological splits yielded different test periods for the 5-, 10-, and 20-year datasets, future studies should evaluate models over aligned out-of-sample periods to ensure that historical-window length effects are not confounded by differing market regimes. Further studies may also shift the focus from predicting absolute price levels to forecasting daily returns or directional movements, including macroeconomic indicators, and investigate combined models that bring together the trend-capturing ability of linear models with the short-term pattern detection capability of ensemble methods. Finally, while simple linear models worked well as basic benchmarks in our study, recent research highlights the value of deep learning. Studies comparing LSTM and Temporal Convolutional Network models show that deep learning can capture time-based patterns and beat naïve benchmark models in financial forecasting. Testing these deep learning models on longer historical datasets is a logical next step for future research.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Amresh K. Singh, Rohit K. Sain

Drafting of the manuscript: Amresh K. Singh, Rohit K. Sain

Critical review of the manuscript for important intellectual content: Amresh K. Singh, Rohit K. Sain

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

Supervision: Amresh K. Singh

Acquisition, analysis, or interpretation of data: Rohit K. Sain

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with [the ICMJE uniform disclosure form](#), all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The datasets (and/or code) supporting this study are available from the corresponding author upon reasonable request.

References

1. Jabeen A, Yasir M, Ansari Y, Yasmin S, Moon J, Rho S: [An empirical study of macroeconomic factors and stock returns in the context of economic uncertainty news sentiment using machine learning](#). Complexity. 2022, 2022:4646733. [10.1155/2022/4646733](#)
2. Jiang W: [Applications of deep learning in stock market prediction: recent progress](#). Expert Systems with Applications. 2021, 184:115537. [10.1016/j.eswa.2021.115537](#)
3. Billah MM, Sultana A, Bhuiyan F, Kaosar MG: [Stock price prediction: comparison of different moving average techniques using deep learning model](#). Neural Computing and Applications. 2024, 36:5861-5871. [10.1007/s00521-023-09369-0](#)
4. Shah P, Shah K, Vedant T, Shah V, Jyotinagar V: [Stock prediction and trading from OHLCV data: a quantum- enhanced learning-based comparison \[PREPRINT\]](#). Research Square. 2026, 1-20. [10.21203/rs.3.rs-9653068/v1](#)
5. National Stock Exchange of India Ltd. (NSE): [Overview of NIFTY 50 index](#). (2026). Accessed: June 1, 2026: <https://www.nseindia.com/>.
6. Mehtab S, Sen J, Dutta A: [Stock price prediction using machine learning and LSTM-based deep learning models](#). Machine Learning and Metaheuristics Algorithms, and Applications. Thampi SM, Piramuthu S, Li KC, Berretti S, Wozniak M, Singh D (ed): Springer, Singapore; 2021. 1366:88-106. [10.1007/978-981-16-0419-5_8](#)
7. Singh D, Gupta BK: [Closing price prediction of Nifty stock using LSTM with dense network](#). Advances in Distributed Computing and Machine Learning. Sahoo JP, Tripathy AK, Mohanty M, Li KC, Nayak AK (ed): Springer, Singapore; 2022. 302:382-392. [10.1007/978-981-16-4807-6_37](#)
8. Ariyo AA, Adewumi AO, Ayo CK: [Stock price prediction using the ARIMA model](#). 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK. 2014, 106-112. [10.1109/UKSim.2014.67](#)
9. Chatterjee A, Bhowmick H, Sen J: [Stock price prediction using time series, econometric, machine learning, and deep learning models](#). 2021 IEEE Mysore Sub Section International Conference (MysuruCon), Hassan, India. 2021, 289-296. [10.1109/MysuruCon52639.2021.9641610](#)
10. Vantuch T, Zelinka I: [Evolutionary based ARIMA models for stock price forecasting](#). ISCS 2014: Interdisciplinary Symposium on Complex Systems. Sanayei A, Rössler OE, Zelinka I (ed): Springer, Cham; 2015. 14:239-247. [10.1007/978-3-319-10759-2_25](#)
11. Moghar A, Hamiche M: [Stock market prediction using LSTM recurrent neural network](#). Procedia Computer Science. 2020, 170:1168-1173. [10.1016/j.procs.2020.03.049](#)

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>

12. Selvin S, Vinayakumar R, Gopalakrishnan EA, Menon VK, Soman KP: [Stock price prediction using LSTM, RNN and CNN-sliding window model](#). 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India. 2017, 1643-1647. [10.1109/ICACCI.2017.8126078](#)
13. Fathali Z, Kodia Z, Ben Said L: [Stock market prediction of NIFTY 50 index applying machine learning techniques](#). Applied Artificial Intelligence. 2022, 36:2111134. [10.1080/08839514.2022.2111134](#)
14. Vijn M, Chandola D, Tikkiwal VA, Kumar A: [Stock closing price prediction using machine learning techniques](#). Procedia Computer Science. 2020, 167:599-606. [10.1016/j.procs.2020.03.326](#)
15. Hoque KE, Aljamaan H: [Impact of hyperparameter tuning on machine learning models in stock price forecasting](#). IEEE Access. 2021, 9:163815-163830. [10.1109/access.2021.3134138](#)
16. [Yahoo finance: NIFTY 50 \(^NSEI\) historical data](#). (2026). Accessed: June 1, 2026: <https://finance.yahoo.com/quote/%5ENSEI/>.
17. Aroussi R: [yfinance: Download market data from Yahoo! Finance's API](#). (2026). Accessed: June 1, 2026: <https://pypi.org/project/yfinance/>.
18. Gupta A, Akansha, Joshi K, Patel M, Pratap V: [Visualization and forecasting of stock's closing price using machine learning](#). Multimedia Tools and Applications. 2024, 83:72471-72489. [10.1007/s11042-024-18376-9](#)
19. Dey R, Kassim S, Khan M, Mahajan RA, William P: [Smart trading in NIFTY-50: ML with MA, RSI & volume analysis](#). 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN), Rangpur, Bangladesh. 2025, 1-5. [10.1109/QPAIN66474.2025.11171705](#)
20. Saberironaghi M, Ren J, Saberironaghi A: [Stock market prediction using machine learning and deep learning techniques: a review](#). AppliedMath. 2025, 5:76. [10.3390/appliedmath5030076](#)
21. [Google Finance: NIFTY 50 \(INDEXNSE: NIFTY 50\) performance chart](#). (2026). Accessed: June 30, 2026: https://www.google.com/finance/beta/quote/NIFTY_50:INDEXNSE.
22. Bhuriya D, Kaushal G, Sharma A, Singh U: [Stock market predication using a linear regression](#). 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India. 2017, 510-513. [10.1109/ICECA.2017.8212716](#)
23. Nayak A, Pai MMM, Pai RM: [Prediction models for Indian stock market](#). Procedia Computer Science. 2016, 89:441-449. [10.1016/j.procs.2016.06.096](#)
24. Nayak RK, Sodha M, Tripathy R, Mishra N, Tripathy SK: [Exploring the efficacy of supervised learning in Indian stock market prediction: a comprehensive analysis](#). Proceedings of Third International Conference on Advanced Computing and Applications. Giri D, Das S, Corchado Rodríguez JM, De D (ed): Springer, Singapore; 2024. 529-544. [10.1007/978-981-97-4799-3_40](#)
25. Sangeetha JM, Alfia KJ: [Financial stock market forecast using evaluated linear regression based machine learning technique](#). Measurement: Sensors. 2024, 31:100950. [10.1016/j.measen.2023.100950](#)
26. Rana M, Uddin MM, Hoque MM: [Effects of activation functions and optimizers on stock price prediction using LSTM recurrent networks](#). Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence. Association for Computing Machinery, New York, NY; 2019. 354-358. [10.1145/3374587.3374622](#)
27. Chen X, Hu W, Xue L: [Stock price prediction using candlestick patterns and sparrow search algorithm](#). Electronics. 2024, 13:771. [10.3390/electronics13040771](#)
28. Eapen J, Bein D, Verma A: [Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction](#). 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA. 2019, 0264-0270. [10.1109/CCWC.2019.8666592](#)
29. Lamba D, Chhikara S, Chouhan AN, Tomar S, Bansal P, Dev A: [LGRNet: a stacked hybrid model for NIFTY50 open price prediction using market indicators and news sentiment](#). Proceedings of International Conference on Artificial Intelligence and Networks. Virdee B, Duarte Correia S, Swaroop A, Thakur N, Shetty SD (ed): Springer, Cham; 2026. 1952:309-321. [10.1007/978-3-032-24926-5_23](#)
30. Elend L, Tideman SA, Lopatta K, Kramer O: [Earnings prediction with deep leaning](#). KI 2020: Advances in Artificial Intelligence. Schmid U, Klügl F, Wolter D (ed): Springer, Cham; 2020. 12325:267-274. [10.1007/978-3-030-58285-2_22](#)
31. Baráth Z, Veres P, Bányai Á: [Beyond traditional forecasting methods: evaluating LSTM performance on diverse time series](#). Mathematics. 2026, 14:838. [10.3390/math14050838](#)

How to cite this article:

Sain R K, Singh A K (October 01, 2026) Open and Close Price Forecasting of the NIFTY 50 Stock Index Using Machine Learning: A Multi-Window Study of Feature Engineering and Model Tuning. Cureus J Comput Sci 3 : es44389-026-00306-5. DOI <https://doi.org/10.7759/s44389-026-00306-5>