© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717939

Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard

R. MAGESH[1], R. SARAVANAN[2]
[1] PG Student, Department of Computer Application, SCSVMV deemed to be University, Kanchipuram, Tamilnadu, India
[2] Assistant Professor, Department of Computer Application, SCSVMV deemed to be University, Kanchipuram, Tamilnadu, India

Abstract-: ***This project presents "Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard", a web-based intelligent financial analytics system designed to predict and analyze the movements of the Nifty 50 stock market index. The system combines machine learning techniques, technical indicator engineering, and interactive data visualization to support both daily and intraday market forecasting. The application is developed using Python technologies such as pandas, NumPy, scikit-learn, Dash, Plotly, and yfinance. The primary objective of the project is to provide an integrated platform that can collect historical market data, preprocess it, generate technical indicators, train predictive machine learning models, and display the results through an interactive dashboard. The system supports two prediction workflows: a daily prediction model for long-term trend analysis and a one-minute intraday prediction model for short-term tactical forecasting. These models use lagged market features, rolling averages, volatility measures, and technical indicators to improve forecasting accuracy. Several regression algorithms, including Linear Regression, Ridge Regression, Lasso Regression, Support Vector Regression, Decision Tree Regressor, Random Forest Regressor, and Gradient Boosting Regressor, are implemented and compared using evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score. The trained models are serialized using joblib and integrated into a Dash-based web application for real-time inference and visualization. The dashboard enables users to view historical market trends, predicted prices, technical charts, and analytical insights through an easy-to-use interface. In addition, the system supports PDF-based exploratory data analysis reports for offline analysis and documentation. The project demonstrates how machine learning and interactive visualization can be combined to create a practical decision-support system for financial market analysis and forecasting.***

I. INTRODUCTION

Stock markets are among the most dynamic and data-intensive financial systems in the world. Predicting stock market movements has always been a challenging task because prices are influenced by various factors such as economic conditions, investor sentiment, government policies, global events, and company performance. Despite these challenges, accurate forecasting of stock prices is highly valuable for investors, analysts, and researchers because it supports better financial decision-making and market analysis.

In India, the Nifty 50 index is one of the most important stock market indices and serves as a benchmark for the performance of the Indian equity market. It represents the top 50 companies listed on the National Stock Exchange (NSE) across different sectors of the economy. Since the Nifty 50 reflects the overall market trend, analyzing and predicting its movement can provide meaningful insights into market behaviour and investment opportunities.

Traditional stock market analysis methods mainly rely on historical charts, manual technical analysis, and statistical methods. However, these approaches often struggle to capture complex and non-linear market patterns. With the advancement of machine learning and data analytics, predictive systems can now learn hidden relationships from historical market data and generate more intelligent forecasting results. Machine learning algorithms are capable of analyzing large volumes of financial data, identifying trends, and improving prediction accuracy through data-driven learning techniques.

This project, titled "Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard," aims to develop an intelligent web-based platform called Stock Sense that combines machine learning, technical indicators, and interactive visualization for Nifty 50 market analysis and prediction. The system supports both daily forecasting and one-minute intraday forecasting, enabling users to analyze long-term market trends as well as short-term market fluctuations. The application is developed using Python technologies such as pandas, NumPy, scikit-learn, Dash, Plotly, and yfinance.

The project follows a complete machine learning workflow that includes historical data collection, preprocessing, feature engineering, model training, evaluation, and deployment. Various regression algorithms such as Linear Regression, Ridge Regression, Lasso Regression, Support Vector Regression, Decision Tree Regressor, Random Forest Regressor, and Gradient Boosting Regressor are used and compared to identify the most suitable model for stock prediction. Technical indicators and lagged features are generated from historical market data to improve forecasting performance.

An important feature of the system is the interactive dashboard developed using Dash and Plotly. The dashboard provides users with live market visualization, predicted values, historical trends, technical charts, and analytical insights in a user-friendly interface. In addition, the system also supports PDF-based exploratory data analysis reports, allowing users to review market analysis offline.

The main objective of this project is to create a practical and intelligent decision-support system that can help students, beginner investors, researchers, and analysts understand stock market behaviour more effectively. Rather than functioning as a fully automated trading platform, the system is designed as an educational and analytical tool that demonstrates how machine learning techniques can be applied to financial forecasting and visualization.

Overall, the project highlights the growing importance of machine learning and interactive analytics in the financial domain. By integrating predictive modelling with visualization technologies, the system provides a comprehensive platform for analyzing Nifty 50 market behaviour and generating data-driven forecasting insights.

II. LITERATURE REVIEW

Stock market prediction has become one of the most important research areas in financial analytics and machine learning because of its practical value in investment decision-making and market analysis. Financial markets generate large volumes of time-series data every day, making them suitable for predictive modelling and data-driven forecasting techniques. However, stock market behaviour is highly dynamic, non-linear, and influenced by many uncertain external factors such as economic conditions, political events, investor sentiment, and global market trends. Due to these complexities, researchers have explored different statistical, machine learning, and visualization-based approaches to improve forecasting accuracy and market understanding.

Traditional stock market forecasting methods mainly relied on statistical and mathematical models such as Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA), and linear regression. These techniques focused on identifying trends and patterns from historical price data. Although statistical methods were useful for basic forecasting, they often struggled to capture non-linear relationships and sudden market fluctuations. As stock markets became more volatile and data-rich, researchers began adopting machine learning approaches to improve prediction performance.

Machine learning techniques provide the ability to learn hidden patterns and relationships from large datasets without requiring explicit programming rules. Several studies have shown that regression-based machine learning algorithms can perform effectively in financial forecasting when combined with appropriate feature engineering and preprocessing methods. Algorithms such as Linear Regression, Ridge Regression, Lasso Regression, Support Vector Regression (SVR), Decision Tree Regressor, Random Forest Regressor, and Gradient Boosting Regressor are widely used in stock

prediction systems because they can model both linear and non-linear market relationships.

Linear Regression is one of the earliest and simplest approaches used in stock market prediction. Researchers commonly use it as a baseline model because of its simplicity, interpretability, and computational efficiency. However, financial markets are rarely linear in nature, which limits the prediction capability of ordinary linear models. To overcome this issue, regularized models such as Ridge Regression and Lasso Regression were introduced. These models help reduce overfitting by penalizing large coefficients and improving model generalization.

Support Vector Regression has also gained popularity in stock forecasting research due to its ability to handle non-linear relationships using kernel functions. Many researchers have reported that SVR performs well for short-term financial prediction tasks because it can manage noisy and high-dimensional data more effectively than traditional regression models. However, SVR requires careful parameter tuning and can become computationally expensive for large datasets.

Decision Tree and ensemble-based models have shown promising results in financial prediction because they can capture complex feature interactions and non-linear market behaviour. Random Forest Regressor improves forecasting stability by combining multiple decision trees and averaging their outputs, which reduces variance and improves robustness. Gradient Boosting Regressor further enhances predictive performance by sequentially correcting the errors made by previous models. Several studies have reported that ensemble models often outperform traditional statistical methods in stock prediction tasks because they can better adapt to dynamic market conditions.

Feature engineering is another important area discussed in financial forecasting literature. Researchers have found that raw stock prices alone are insufficient for effective prediction. Instead, predictive models perform better when technical indicators and lagged features are included as input variables. Commonly used indicators include Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), Bollinger Bands, and Moving Average Convergence Divergence (MACD). These indicators help summarize market momentum, volatility, and trend direction, making them useful for machine learning models.

Time-series analysis is also a major concept in stock market forecasting research. Since stock prices are sequential data, researchers emphasize the importance of preserving chronological order during preprocessing and model evaluation. Lagged variables, rolling averages, and volatility measures are commonly used to convert stock data into supervised learning datasets...

In recent years, visualization dashboards have become increasingly important in financial analytics systems. Researchers argue that prediction results are more useful when presented alongside interactive charts and technical indicators. Tools such as Plotly and Dash are widely used for building financial dashboards because they support dynamic visualization, zooming, filtering, and real-time interaction. Interactive dashboards help users understand prediction results more effectively compared to static graphs or numerical outputs alone.

III. PROBLEM STATEMENT

Forecasting stock market behaviour remains a complex and significant challenge in the field of financial analytics due to the highly dynamic, volatile, and non-linear nature of market data. The Nifty 50 index, which represents the performance of the top companies listed on the National Stock Exchange (NSE) of India, plays a crucial role in evaluating the overall condition of the Indian stock market. Accurate prediction of Nifty 50 movements can support investors, analysts, and researchers in making informed financial decisions. However, existing market analysis systems primarily focus on historical visualization and basic statistical analysis, with limited capability for intelligent forecasting and real-time analytical support.

Traditional forecasting techniques often fail to capture hidden patterns and temporal dependencies

present in financial time-series data. In addition, many existing stock prediction systems rely on a single forecasting model and do not effectively compare the performance of multiple machine learning algorithms. The absence of efficient feature engineering methods, such as lagged variables and technical indicators, further reduces prediction reliability and forecasting accuracy.

Another major limitation in existing approaches is the lack of integrated platforms that combine machine learning prediction, interactive visualization, and real-time market analytics within a unified environment. Most available systems either provide prediction outputs without sufficient interpretability or offer visualization dashboards without intelligent forecasting capabilities. Furthermore, few systems support both strategic daily forecasting and short-term intraday forecasting in the same application framework.

Therefore, there is a need for a comprehensive and intelligent stock market analysis system that can acquire historical and real-time Nifty 50 market data, preprocess and engineer meaningful predictive features, evaluate multiple machine learning regression models, and generate accurate forecasting outputs through an interactive and user-friendly dashboard. The proposed system addresses these challenges by integrating machine learning algorithms, technical indicator analysis, and web-based visualization technologies to develop a Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard capable of supporting both daily and one-minute prediction workflows for enhanced financial decision support.

IV. SYSTEM ANALYSIS

System analysis is one of the most important phases in software development because it helps identify the functional requirements, system behaviour, data flow, processing methods, and technical structure needed to develop an efficient application. In the proposed project, Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard, system analysis focuses on understanding how financial market data can be collected, processed, analyzed, and transformed into meaningful forecasting outputs using machine learning techniques and interactive visualization tools.

The primary goal of the system is to develop a web-based intelligent dashboard capable of predicting and analyzing Nifty 50 stock market movements using historical and near-real-time market data. The system combines machine learning algorithms, feature engineering, technical indicators, and visualization technologies to create an integrated financial analytics platform.

A. Existing System

Existing stock market analysis systems mainly focus on displaying historical market data, price charts, and basic technical indicators for investors and analysts. Many available platforms provide graphical representations of stock prices, trading volume, moving averages, and trend analysis tools. These systems help users monitor market behaviour and understand historical trends, but most of them have limited capability in intelligent prediction and machine learning-based forecasting.

Traditional stock analysis systems generally depend on statistical methods and manual technical analysis techniques. Investors often use moving averages, candlestick charts, Relative Strength Index (RSI), MACD, and Bollinger Bands to identify market trends and possible price movements. Although these approaches are useful for descriptive analysis, they do not provide automated predictive intelligence based on historical learning patterns.

B. System Architecture

The system architecture of the Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard is designed to provide an integrated environment for financial data acquisition, preprocessing, machine learning-based prediction, and interactive visualization. The architecture follows a modular and layered design approach that separates data processing, model training, prediction generation, and dashboard visualization into independent but interconnected components. This structure improves system scalability, maintainability, and performance while supporting both daily and one-minute forecasting workflows.

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717939

The architecture mainly consists of two major layers: the Offline Training Layer and the Online Inference and Visualization Layer. The offline layer is responsible for historical data collection, preprocessing, feature engineering, machine learning model training, evaluation, and model serialization. The online layer handles real-time data retrieval, feature recomputation, prediction generation, and interactive dashboard visualization.

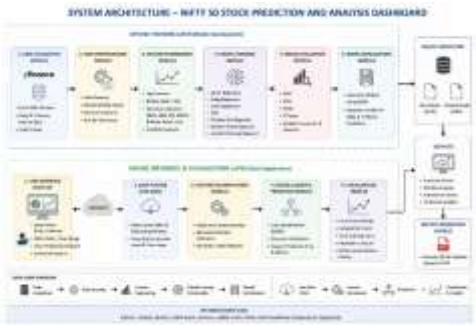

Fig. 1. System Architecture Diagram

C. Data Flow

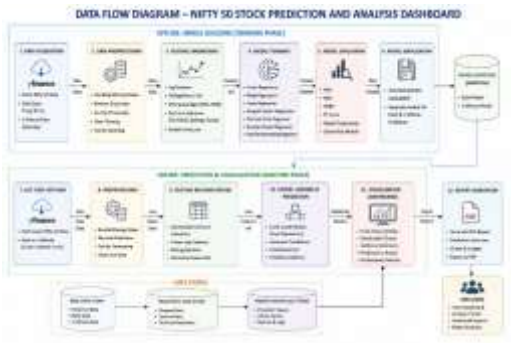

Fig. 2. Data Flow Diagram

The data flow of the Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard describes how market data moves through different modules of the system, beginning from data acquisition and ending with prediction visualization and report generation. The data flow architecture ensures that financial information is processed systematically, transformed into meaningful features, analyzed using machine learning models, and presented through an interactive dashboard. The entire workflow is designed to support both daily and one-minute forecasting operations efficiently and accurately.

The data flow process starts with the Data Acquisition Module, where historical and live Nifty 50 stock market data is collected using the yfinance library. The system retrieves market information such as Open, High, Low, Close, Adjusted Close, and Volume values. Depending on the selected forecasting mode, the system downloads either long-term daily data or short-term one-minute intraday data. This raw market data serves as the primary input for the system.

After data collection, the retrieved dataset is passed to the Data Preprocessing Module. In this stage, the system cleans and organizes the raw market data before it is used for analysis and prediction. Preprocessing operations include handling missing values, removing duplicate records, sorting timestamps in chronological order, filtering invalid entries, and preparing the dataset for feature generation.

D. System Database Design

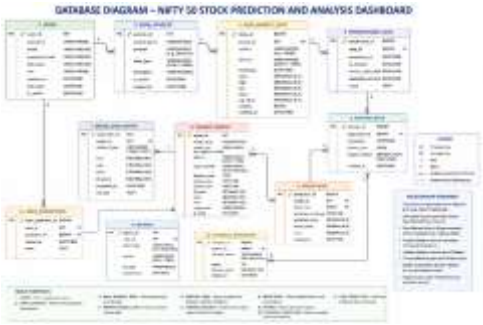

Fig. 3. System Database Diagram

The database design of the Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard is developed to efficiently store, manage, and retrieve stock market data, engineered features, machine learning model information, prediction outputs, and analytical reports. A well-structured database design is important because the system processes large volumes of time-series financial data for both daily and one-minute forecasting workflows. The database supports data consistency, fast retrieval, scalability, and smooth interaction between different system modules.

The database architecture is designed around the core functional components of the system, including market data acquisition, preprocessing, feature engineering, machine learning prediction, visualization, and report generation. The system mainly handles structured numerical data generated from stock market analysis and predictive modelling.

The primary entity in the database design is the Market Data Table. This table stores historical and live Nifty 50 stock market information collected using the yfinance API. The table contains attributes such as Date, Open Price, High Price, Low Price, Close Price, Adjusted Close Price, Volume, and Time Interval. The data is stored separately for daily and one-minute intervals to support different forecasting workflows. The Date-Time field acts as the primary key because each market record is uniquely identified by its timestamp.

After acquisition, the cleaned and processed data is stored in the Preprocessed Data Table. This table contains normalized and cleaned market records after missing-value handling, duplicate removal, and chronological sorting. Storing preprocessed data separately improves system efficiency because repeated preprocessing operations can be avoided during model training and prediction.

The Feature Engineering Table stores all engineered features generated from the historical market data. These features include lag values, rolling averages, rolling standard deviation, volatility measurements, moving averages, exponential moving averages, and technical indicators such as RSI, MACD, and Bollinger Bands. The feature table is important because machine learning models require structured feature inputs rather than raw stock prices. Each feature record is linked to the corresponding market timestamp using a foreign key relationship.

The Machine Learning Model Table stores information related to trained prediction models. This table contains attributes such as Model ID, Model Name, Algorithm Type, Training Date, Accuracy Metrics, File Path, and Forecasting Mode. Separate records are maintained for daily forecasting models and one-minute forecasting models. The system stores serialized model files using joblib, and the file paths are maintained in the database for runtime loading.

The Prediction Results Table stores the prediction outputs generated by the machine learning models. This table contains the prediction timestamp, actual market value, predicted value, model used, forecasting interval, and evaluation metrics. The table helps track prediction history and supports visualization of predicted versus actual market behaviour within the dashboard.

V. METHODOLOGY

The methodology of the Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard describes the complete workflow followed to collect stock market data, preprocess it, generate predictive features, train machine learning models, evaluate forecasting performance, and display the results through an interactive web dashboard. The methodology combines financial data analysis, machine learning techniques, technical indicator engineering, and visualization technologies to create an intelligent decision-support system for Nifty 50 market prediction.

The proposed system follows a structured workflow consisting of multiple stages: data acquisition, preprocessing, feature engineering, machine learning model development, model evaluation, model serialization, prediction generation, visualization, and report generation. Each stage plays an important role in ensuring accurate forecasting and efficient dashboard performance.

The first stage in the methodology is Data Acquisition. Historical and live Nifty 50 market data is collected using the yfinance library. The system retrieves stock market attributes such as Open, High, Low, Close, Adjusted Close, and Volume values. The project supports two forecasting workflows: daily prediction and one-minute intraday prediction. Therefore, separate datasets are collected for daily intervals and one-minute intervals depending on the selected prediction mode.

After collecting the data, the system performs Data Preprocessing. Financial market datasets often

contain missing values, duplicate records, irregular timestamps, and noisy information. The preprocessing stage cleans and organizes the raw market data before it is used for prediction. This stage includes handling missing values, removing duplicates, sorting records chronologically, filtering invalid entries, and preparing the dataset in a structured tabular format.

The next stage is Feature Engineering, which is one of the most important components of the methodology. Raw stock prices alone are not sufficient for effective prediction because machine learning models require informative input features that capture market behaviour and trends. Therefore, the system generates lagged values, rolling averages, rolling standard deviations, returns, volatility measures, and technical indicators from historical market data. Technical indicators such as Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), MACD, and Bollinger Bands are computed to improve forecasting performance and market analysis capability.

Once the features are generated, the system converts the dataset into a supervised learning format. The engineered features become the input variables, while the future stock price acts as the target variable. The dataset is then divided into training and testing subsets while preserving the time-series sequence to avoid data leakage.

The methodology then proceeds to Machine Learning Model Development. Multiple regression algorithms are implemented to compare prediction performance and identify the most suitable forecasting model for Nifty 50 analysis. The machine learning models used in the project include:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Support Vector Regression (SVR)
- Decision Tree Regressor
- Random Forest Regressor
- Gradient Boosting Regressor

These models are trained using historical market features and evaluated on unseen testing data. The use of multiple algorithms improves the reliability of the system because different models behave differently under varying market conditions.

The next stage is Model Evaluation. After training, the system evaluates each model using performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score. These evaluation metrics help measure forecasting accuracy and determine how effectively each model predicts future market behaviour. The best-performing models are selected separately for daily forecasting and one-minute intraday forecasting.

Once the optimal models are selected, the methodology includes Model Serialization using joblib. Serialization allows trained machine learning models to be saved permanently and reused during runtime without retraining.

The serialized models are then integrated into the Online Prediction and Inference Module developed using Dash. During runtime, the dashboard fetches updated market data, performs the same preprocessing and feature engineering operations, and generates predictions using the trained machine learning models...

The generated predictions are displayed through the Visualization Module using Plotly and Dash Bootstrap Components. The dashboard provides interactive charts, candlestick graphs, technical indicator plots, historical trends, predicted prices, and performance summaries.

VI. RESULTS AND DISCUSSION

The Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard was successfully implemented and tested using historical and near-real-time Nifty 50 stock market data. The system combined machine learning regression models, technical indicator analysis, and interactive visualization techniques to generate forecasting outputs for both daily and one-minute intraday prediction workflows. The results demonstrate that machine learning models can effectively identify historical market patterns and provide meaningful

forecasting insights when supported by proper preprocessing and feature engineering methods.

The first stage of evaluation focused on the performance of the machine learning models used in the project. Multiple regression algorithms, including Linear Regression, Ridge Regression, Lasso Regression, Support Vector Regression (SVR), Decision Tree Regressor, Random Forest Regressor, and Gradient Boosting Regressor, were trained and tested using engineered financial features such as lagged values, moving averages, rolling statistics, and volatility indicators. The models were evaluated using standard regression metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score.

The experimental results showed that ensemble-based algorithms such as Random Forest Regressor and Gradient Boosting Regressor produced better forecasting accuracy compared to simple linear models. These algorithms were more effective at capturing non-linear market relationships and handling complex feature interactions within the financial time-series data. Linear Regression models performed reasonably well as baseline predictors but were less effective during highly volatile market conditions because stock market behaviour is rarely completely linear.

The Support Vector Regression model also produced stable forecasting results, especially for short-term intraday prediction, because it handled non-linear relationships and noisy data effectively. However, the training time and parameter tuning requirements for SVR were higher compared to other regression techniques. Decision Tree Regressor provided interpretable prediction structures but occasionally suffered from overfitting when handling highly fluctuating market data.

The feature engineering process significantly improved model performance. The inclusion of lagged values, moving averages, rolling standard deviation, volatility measures, and technical indicators such as SMA, EMA, RSI, and MACD helped the machine learning models identify historical price behaviour more effectively than using raw stock prices alone. The results clearly indicate that technical indicators and engineered features play an important role in improving prediction accuracy for financial forecasting systems.

The dashboard visualization module was successfully integrated using Dash and Plotly. The system displayed historical market trends, predicted prices, technical indicator charts, candlestick visualizations, and model comparison outputs in a responsive web interface. Users were able to interact with charts using zooming, filtering, and hovering features, which improved usability and interpretability of the prediction results. The dashboard successfully supported both daily forecasting and one-minute intraday forecasting within a single platform.

The daily forecasting workflow produced more stable and consistent prediction outputs because daily stock market data contains smoother long-term trends and less short-term noise. In contrast, the one-minute intraday prediction workflow was more sensitive to rapid market fluctuations and volatility. Although intraday forecasting achieved useful short-term insights, the prediction accuracy was slightly lower compared to daily forecasting due to the highly dynamic nature of minute-level market behaviour.

The online inference system functioned effectively by loading serialized machine learning models using joblib and generating predictions from live market data. The runtime prediction process remained efficient because models were pre-trained offline and reused during dashboard execution. This reduced computational overhead and improved dashboard responsiveness.

The report generation module also produced successful PDF-based exploratory data analysis reports containing prediction summaries, technical indicator charts, and historical market analysis. This feature enhanced the usability of the system for academic documentation and offline review purposes.

A. Predictions R-squared

“R-squared” represents the performance comparison of different machine learning models used in the Nifty 50 stock prediction system based on the R^2 Score (Coefficient of Determination). The R^2 Score is

© MAY 2026 | IRE Journals | Volume 9 Issue 11 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I11-1717939

an important regression evaluation metric used to measure how well a machine learning model explains the variance in the target variable. The value of R^2 ranges from 0 to 1, where values closer to 1 indicate better predictive performance and stronger model The horizontal from the graph, it is observed that Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest Regressor, Gradient Boosting, and MLP Regressor achieved R^2 Scores close to 1.0.. Their strong R^2 values suggest that the engineered features and technical indicators effectively captured the market behaviour required for prediction.

Among these models, ensemble learning algorithms such as Random Forest Regressor and Gradient Boosting performed exceptionally well because they are capable of handling non-linear relationships and complex feature interactions present in stock market data.

The graph also shows that the Support Vector Regressor (SVR) produced a significantly lower R^2 Score compared to the other models. This indicates that SVR was less effective in explaining the variance of the Nifty 50 stock market data for the selected feature set and training configuration. The lower performance may be caused by factors such as insufficient parameter tuning, sensitivity to noisy financial data, or limitations in handling the complexity of the dataset.

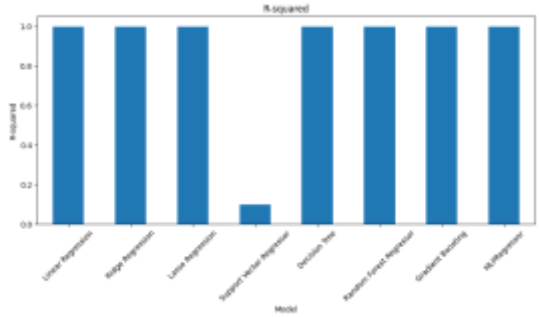

Fig. 4. prediction.ipynb

B. Score Predictions

The Random Forest Regressor and Gradient Boosting models also demonstrated strong performance with relatively high Recall and F1 Scores. These ensemble-based models performed well because they can capture complex non-linear relationships and interactions within stock market data. Gradient Boosting achieved a particularly strong F1 Score, indicating balanced predictive capability and improved handling of market variations.

The Decision Tree model showed moderate performance with scores around the mid-range level. While Decision Trees are capable of modeling non-linear behaviour, they may sometimes suffer from overfitting when handling highly volatile financial data, which can reduce generalization performance.
The Lasso Regression and Support Vector Regressor (SVR) models produced comparatively lower scores than the top-performing algorithms. Their performance indicates that they were able to capture some prediction patterns but were less effective in maintaining high recall and balanced prediction performance for the given dataset.

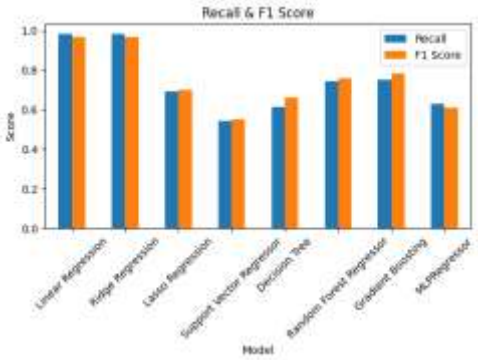

Fig. 5. Score Predictions

C. MAE & MSE MODEL

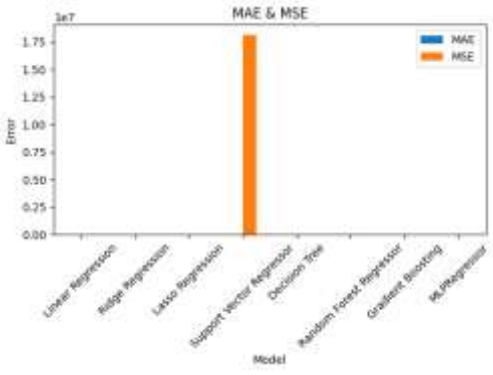

Fig. 6. MAE & MSE MODEL

Mean Absolute Error (MAE) measures the average absolute difference between the actual stock values and the predicted stock values. Lower MAE values indicate better prediction accuracy because the predicted values are closer to the actual market prices.

Mean Squared Error (MSE) measures the average squared difference between actual and predicted values. Since the errors are squared, larger prediction mistakes are penalized more heavily. A lower MSE value indicates a more accurate and stable forecasting model.

From the graph, it can be observed that most machine learning models achieved relatively low MAE and MSE values, indicating good prediction performance and effective forecasting capability. However, the Support Vector Regressor (SVR) shows an extremely large MSE value compared to all other models. This suggests that the SVR model generated significantly larger prediction errors and was less effective for the given Nifty 50 dataset and feature configuration.

The high MSE value for SVR indicates that the model struggled to handle the complexity and volatility of the stock market data. Possible reasons for this behaviour include insufficient parameter tuning, sensitivity to noisy financial time-series data, inappropriate kernel selection, or overfitting issues. Since MSE penalizes large prediction errors heavily, even a few major forecasting mistakes can produce a very large MSE value.

The remaining models, including Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest Regressor, Gradient Boosting, and MLP Regressor, maintained comparatively lower MAE and MSE values. This indicates that these algorithms produced predictions closer to the actual stock market values and handled the engineered financial features more effectively.

Among these models, ensemble learning algorithms such as Random Forest Regressor and Gradient Boosting generally provide better prediction stability because they combine multiple decision structures to reduce variance and improve generalization. Their lower error values demonstrate their suitability for stock market forecasting tasks.

VII. CONCLUSION

Mean Absolute Error (MAE) measures the average absolute difference between the actual stock values and the predicted stock values. Lower MAE values indicate better prediction accuracy because the predicted values are closer to the actual market prices.

Mean Squared Error (MSE) measures the average squared difference between actual and predicted values. Since the errors are squared, larger prediction mistakes are penalized more heavily. A lower MSE value indicates a more accurate and stable forecasting model.

From the graph, it can be observed that most machine learning models achieved relatively low MAE and MSE values, indicating good prediction performance and effective forecasting capability. However, the Support Vector Regressor (SVR) shows an extremely large MSE value compared to all other models. This suggests that the SVR model generated significantly larger prediction errors and was less effective for the given Nifty 50 dataset and feature configuration.

The high MSE value for SVR indicates that the model struggled to handle the complexity and volatility of the stock market data. Possible reasons for this behaviour include insufficient parameter tuning, sensitivity to noisy financial time-series data, inappropriate kernel selection, or overfitting issues. Since MSE penalizes large prediction errors heavily, even a few major forecasting mistakes can produce a very large MSE value.

The remaining models, including Linear Regression, Ridge Regression, Lasso Regression, Decision Tree, Random Forest Regressor, Gradient Boosting, and MLP Regressor, maintained comparatively lower MAE and MSE values. This indicates that these algorithms produced predictions closer to the actual stock market values and handled the engineered financial features more effectively.

Among these models, ensemble learning algorithms such as Random Forest Regressor and Gradient Boosting generally provide better prediction stability because they combine multiple decision structures to reduce variance and improve generalization. Their lower error values demonstrate their suitability for stock market forecasting tasks.

VIII. LIMITATIONS

Machine Learning-Based Nifty 50 Stock Prediction and Analysis Dashboard successfully provides stock market forecasting and interactive analytical visualization, the system still has several limitations related to financial data complexity, prediction reliability, and technical implementation. These limitations are important to understand because stock market forecasting is inherently uncertain and influenced by multiple unpredictable external factors.
One of the major limitations of the system is the dependency on historical market data for prediction. The machine learning models are trained mainly using past Nifty 50 price movements and technical indicators. However, stock markets are affected by real-world events such as economic policies, political changes, global crises, investor sentiment, and breaking news, which may not be reflected in historical price data alone. As a result, sudden market fluctuations and unexpected events can reduce prediction accuracy.

Another limitation is related to the volatile and non-linear nature of stock market data. Financial markets are highly dynamic and continuously changing, making it difficult for machine learning models to generalize perfectly across all market conditions. Even high-performing models may fail during periods of extreme volatility, market crashes, or unusual trading behaviour.

The project mainly uses regression-based machine learning models and technical indicators for forecasting. Although these algorithms perform effectively for pattern recognition and trend analysis, they may not capture deep sequential dependencies and long-term temporal relationships as efficiently as advanced deep learning architectures such as LSTM or Transformer-based models.

The system also depends on the quality and availability of data retrieved from yfinance. Since yfinance is an external data provider, issues such as API limitations, delayed updates, missing records, incomplete intraday data, or temporary connection failures may affect system performance and real-time prediction accuracy. Intraday one-minute data is especially sensitive to data availability and refresh delays.

IX. FUTURE ENHANCEMENTS

The current system focuses mainly on the Nifty 50 index, but future versions can support multiple stock indices and individual company stocks. Expanding the platform to include Sensex, Bank Nifty, NSE sectoral indices, and global stock markets would increase the practical usability and scalability of the system. Users would then be able to compare and analyze different financial markets within a single dashboard environment.

Another enhancement involves implementing real-time streaming data analytics. Currently, the system uses periodic data retrieval through yfinance, which may have limitations in live market scenarios. Future implementations can integrate real-time stock market APIs and WebSocket-based streaming services to provide continuously updating dashboards and instant prediction refreshes. This would make the system more suitable for active traders and real-time market monitoring.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2011.

[2] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.

[3] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson Education, 2021.

[4] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

[5] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[6] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[7] S. Patel, K. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques,” Expert Systems with Applications, vol. 42, no. 1, pp. 259–268, 2015.

[8] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” Expert Systems with Applications, vol. 42, no. 20, pp. 7046–7056, 2015.

[9] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock market index using fusion of machine learning techniques,” Expert Systems with Applications, vol. 42, no. 4, pp. 2162–2172, 2015.

[10] B. G. Malkiel, A Random Walk Down Wall Street, 12th ed. New York, NY, USA: W. W. Norton & Company, 2019.

[11] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.

[12] W. McKinney, Python for Data Analysis, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.

[13] Scikit-learn Developers, “Scikit-learn: Machine Learning in Python,” [Online]. Available: https://scikit-learn.org/

[14] Plotly Technologies Inc., “Plotly Python Graphing Library,” [Online]. Available: https://plotly.com/python/

[15] Dash Developers, “Dash: Analytical Web Applications for Python,” [Online]. Available: https://dash.plotly.com/