



# PYTHON BASED STOCK PRICE PREDICTION

## *Accurate Forecasting for Investment Decisions*

<sup>1</sup>Doddegowda B J, <sup>2</sup>Pavani L, <sup>3</sup>Srushti Shettar, <sup>4</sup>Uday Kiran B R,

<sup>1</sup>Associate Professor, <sup>2</sup>Student, <sup>3</sup>Student, <sup>4</sup>Student,

Department of Information Science and Engineering,  
T John Institute of Technology, Bengaluru, India

**Abstract:** This paper explores how Machine Learning can be used to predict stock prices by forecasting future values based on past trading data, especially closing prices. Using Python, the study develops a prediction model grounded in popular financial analysis and time-series methods. The goal is to make the prediction process simpler and more accurate by applying Machine Learning. The method is trained on historical stock data to uncover patterns that help generate precise forecasts. The focus is on the ARIMA (Autoregressive Integrated Moving Average) model, which is effective at recognizing trends in stock price changes. By examining historical closing prices, the model aims to predict future stock values and identify potential price fluctuations or instabilities.

**Key Terms** - Stock price , Prediction , ARIMA , Machine Learning , Python

### I. INTRODUCTION

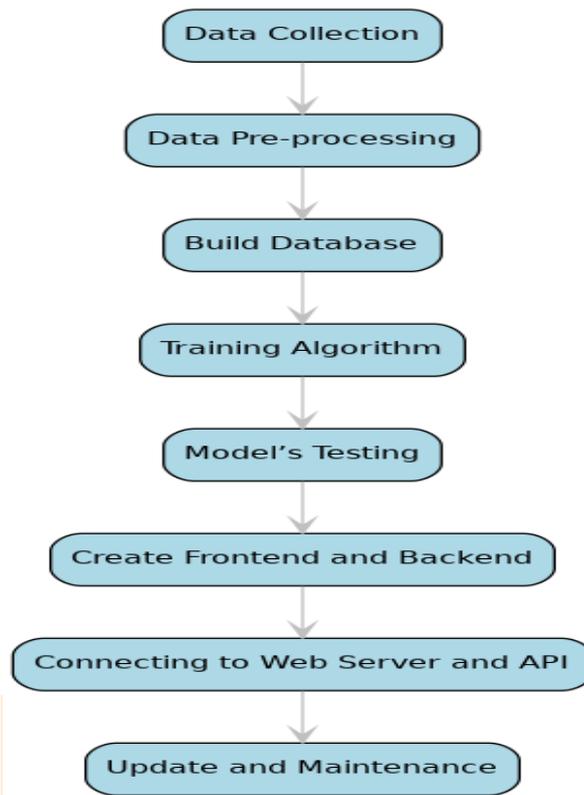
Stock markets play a vital role in a country's economy, acting as the main channel for companies to raise money and for investors to put their funds to work. Because stock markets affect both personal finances and the wider economy, being able to predict stock price movements is very important for making smart decisions about buying or selling shares. However, predicting these prices is quite difficult since many complex factors influence them, including economic changes, politics, and even the weather. This makes stock prices unpredictable, unstable, and noisy, which traditional forecasting methods struggle to manage. Some financial theories like the Efficient Market Hypothesis suggest that stock prices move randomly, making long-term prediction nearly impossible because any exploitable patterns get quickly used up. Still, short- to medium-term trends sometimes appear due to inefficiencies in the market and human emotions. Modern machine learning and deep learning techniques aim to capture these trends by analyzing large amounts of varied data to find hidden patterns that simpler methods miss. Despite their strengths, challenges like poor-quality data, sudden market swings, and overfitting risks remain. Combining technical analysis with fundamentals and sentiment from news and social media can boost accuracy but also adds complexity. Researchers are now working on hybrid models that mix different methods and on explainable models that balance being accurate and efficient while addressing ethical issues like bias in algorithms. Overall, stock market prediction is a complicated and evolving area that needs advanced models and ongoing improvement.

## II. LITERATURE REVIEW

Predicting stock prices is a complex yet important area in financial analysis, helping investors and institutions make smarter decisions[2]. Stock prices are tough to forecast because they fluctuate a lot and behave in unpredictable, nonlinear ways. One popular method for short-term stock price forecasts is the ARIMA model, which stands for Autoregressive Integrated Moving Average. ARIMA combines three main parts that help it understand patterns and noise in past price data. Although it assumes the data is stable and follows a linear pattern, ARIMA has been widely used for its simplicity and effectiveness in modeling financial time series, especially for short-term predictions[1][3]. Researchers typically start by checking if the stock data is steady (stationary) and transform it as needed to remove trends or seasonal effects[1][2]. The model's parameters are selected using statistical tools, then tested to ensure the model fits well. Studies have shown that ARIMA works well in capturing short-term trends like daily index prices, but its accuracy drops over longer periods because of market complexity and volatility. Financial markets often display erratic behavior due to economic changes and investor emotions, which ARIMA alone can struggle to capture. To improve on this, ARIMA has been combined with other models. For example, pairing ARIMA with GARCH models helps track changes in market volatility, which ARIMA cannot do by itself. Also, more recent approaches mix ARIMA with machine learning methods like neural networks to cover both linear and nonlinear parts of the data. Such hybrid models have proven to be more accurate for predicting stock prices than ARIMA or machine learning alone. Despite these advancements, ARIMA remains a key baseline for comparison because of its interpretability and ease of use. It works well especially when data is limited or prediction periods are short. But for long-term and highly nonlinear data, machine learning models often perform better. Researchers evaluate ARIMA predictions not just by typical accuracy metrics but also by financial performance in trading. ARIMA does have limits: it assumes stability and linearity, can lose important information when data is modified, and may not handle sudden market shifts or volatility dynamics effectively. This has led to many variants and hybrid models that improve on these weaknesses but add complexity. In conclusion, ARIMA continues to be an essential tool for understanding and forecasting stock prices in the short term. It provides a solid foundation for building more complex models and remains valuable for its transparency and statistical rigor. Future work is focusing on combining ARIMA with explainable AI and testing its robustness during market crises, promising more reliable and insightful stock price predictions.

## III. METHODOLOGY

The study uses a hybrid method where ARIMA models the linear patterns in the stock data, while a technique called directional validation—borrowed from neural network evaluation—checks if the predictions point in the right direction. The system is assessed using both standard error metrics like RMSE and MAPE, as well as financial measures such as Directional Symmetry. Directional Accuracy compares whether the predicted and actual returns move up or down together, giving a more practical perspective on how well the model would perform in real trading. This approach combines statistical modeling with insights from deep learning to create a balanced forecasting system. ARIMA handles the linear trends, and the directionality check helps ensure the predictions are meaningful for market decisions. Metrics like MAE, RMSE, and Directional Symmetry provide a comprehensive view of both prediction precision and market relevance.



**Fig 1 : The Proposed System**

The fig 1 refers process involved in Proposed system

1. **Data Collection** : Historical stock prices for Alphabet Inc. (GOOGL) from 2015–2025 are sourced from Yahoo Finance.
2. **Data Preprocessing** : Missing values are handled, data is normalized, and outliers are removed to improve model stability.
3. **Model Training** : ARIMA parameters are optimized using statistical criteria like AIC and BIC, while a direction-checking function based on daily returns evaluates the trend direction.
4. **Model Testing** : The model is validated using RMSE, MAE, and Directional Accuracy (DA), which measures how often the predicted trend matches the actual trend.
5. **Deployment** : A Python-based Flask interface displays real-time predictions for user access.
6. **Updating** : The model retrains periodically using new market data to remain up-to-date.

### Data and Sources of Data

For this project, we collected stock price data from **Yahoo Finance**, which provides free and reliable historical market information for most publicly traded companies. The data include key details such as the opening price, highest and lowest prices during the day, closing price, adjusted closing price, and trading volume. Among these, we focused mainly on the **Adjusted Close** price, since it gives a more accurate picture of a stock's true value by accounting for events like dividends and stock splits. This makes it a better indicator for predicting future prices. The dataset was downloaded using the **Python yfinance library**, which allows easy access to historical stock data directly from Yahoo Finance. We collected daily price data for a period of **last two years (from January 2023 to October 2025)**. This time span was chosen to capture both short-term fluctuations and long-term market trends. Before using the data, we performed a few cleaning steps. Any missing values were filled using the most recent available data. Since stock markets are closed on weekends and holidays, those dates were automatically skipped in the dataset. After cleaning, we had a smooth, continuous time series that was ready for use in our ARIMA-based prediction model.

## Theoretical framework

This study is based on the idea that stock prices change over time in patterns that can often be studied and predicted using **time series analysis**. A time series is simply a sequence of data points recorded at regular intervals — in this case, daily stock prices. By looking at how these prices have moved in the past, we can try to understand their behavior and make reasonable forecasts about the future. The main model used in this research is the **ARIMA model**, which stands for **AutoRegressive Integrated Moving Average**. ARIMA is a well-known statistical method used for forecasting time series data, especially when the data show trends or patterns that change over time.

In simple terms, the ARIMA model works by examining how current stock prices depend on their previous values and the errors made in earlier predictions. It breaks this process into three parts:

- The **AR (AutoRegressive)** part looks at the relationship between a stock's current price and its past prices.
- The **I (Integrated)** part helps make the data “stationary,” meaning it removes trends or fluctuations that could distort the prediction.
- The **MA (Moving Average)** part focuses on the errors made in earlier predictions and tries to correct them in future forecasts.

By combining these three elements, the ARIMA model can capture both the trend and the randomness in stock price movements. Even though stock markets are influenced by many unpredictable factors — such as news events, economic policies, or investor behavior — ARIMA can still provide reliable **short-term forecasts** based purely on historical price data. The reason this model was chosen for our research is that it has a strong mathematical foundation, is easy to interpret, and has been successfully used in many financial forecasting studies. It allows us to analyze past stock price patterns, identify underlying trends, and make informed predictions about future movements.

## IV. IMPLEMENTATION

We built a small, reproducible Python pipeline that takes a stock's daily closing prices, makes the series stationary, and fits an ARIMA model to forecast the next few days. Below is what we actually did—step by step, in plain language.

### Setup and environment

All experiments ran on a standard laptop (no GPU). We used Python 3.10 with the following libraries: pandas (data handling), numpy (numerics), statsmodels (ARIMA), and scikit-learn (metrics). Random seeds were fixed where applicable to keep results consistent.

### Data ingestion and cleaning

We worked with daily **Adjusted Close** prices for the target ticker. Missing calendar days (weekends/holidays) were left as gaps in the calendar but not in the index; the time series only contains trading days. If any trading-day values were missing, we forward-filled them to avoid introducing artificial volatility. Prices were cast to float64 and indexed by a timezone-naïve DatetimeIndex.

### Train/test split

To mimic real usage, we split the time series chronologically: the **earliest ~85–90%** of observations for training and the **most recent ~10–15%** for testing. This prevents any look-ahead bias and ensures the model only “sees” the past when predicting the future.

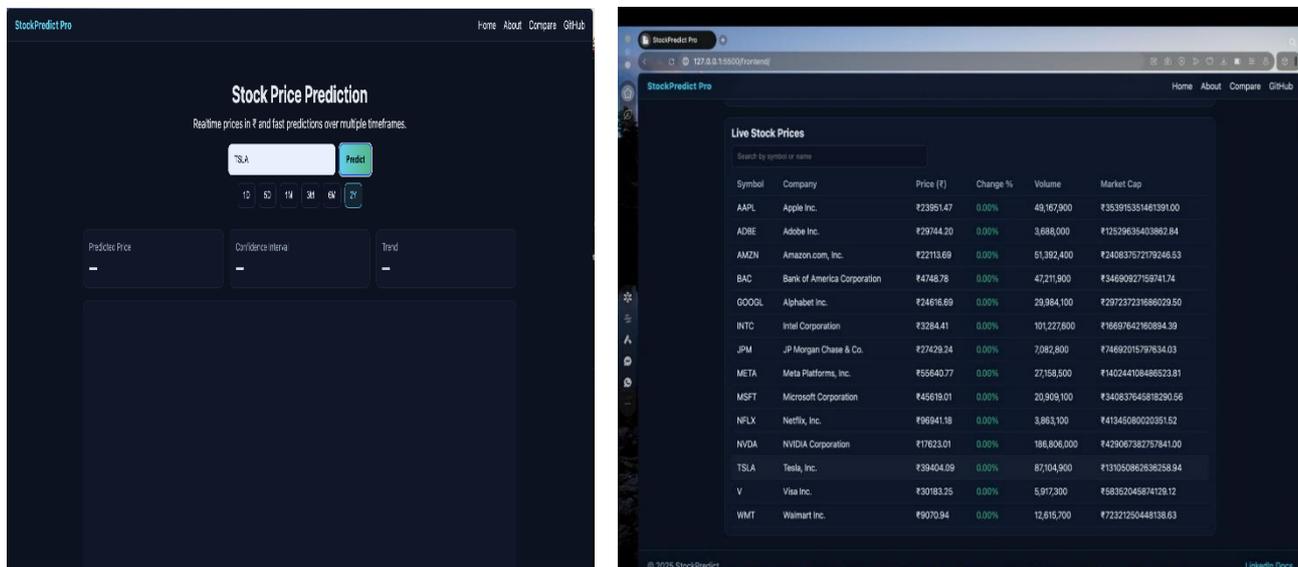


Fig 2 : Display of stock information on result page

## Making the series stationary

ARIMA expects a stationary signal. We:

1. Inspected rolling mean/variance plots to eyeball trends.
2. Ran the Augmented Dickey–Fuller (ADF) test on the raw series.
3. Applied first-order differencing if needed (and re-tested). If the ADF still suggested non-stationarity, we tried a second difference, but we capped differencing at  $d \leq 2$  to avoid over-differencing.

Seasonality in daily equities is typically weak and unstable; we verified there was no strong, stable seasonal pattern over our horizon, so we stayed with non-seasonal ARIMA (not SARIMA).

## Order selection (p, d, q)

With d decided from the stationarity step, we chose (p, q) by minimizing AIC over a compact grid:

- $p \in \{0, \dots, 5\}$ ,  $q \in \{0, \dots, 5\}$ .  
For each candidate, we fit an ARIMA model on the training set and recorded AIC; the lowest-AIC model was selected. If the chosen model showed autocorrelated residuals (Ljung–Box p-values small), we nudged p or q upward and re-evaluated until residuals looked like white noise.

## Model fitting

We fit the final ARIMA(p, d, q) using statsmodels.tsa.arima.model.ARIMA with enforce\_stationarity=False and enforce\_invertibility=False (to avoid unnecessary convergence failures). Convergence warnings were addressed by slightly tightening tolerances or switching optimizers (e.g., from lbfgs to bfgs) when needed.

## Evaluation

We report **MAE**, **RMSE**, and **MAPE** on the test window. Beyond point accuracy, we also inspected:

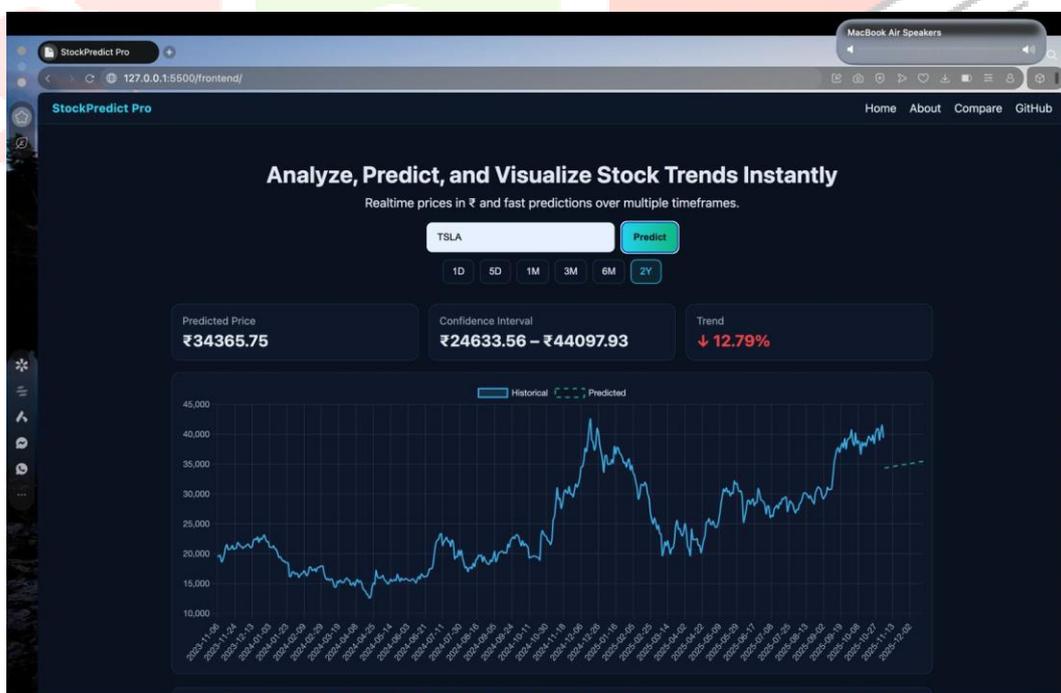
- **Residual diagnostics** (should look like white noise; no visible autocorrelation in ACF).
- **Prediction intervals** from the ARIMA model to understand uncertainty.

## Practical touches and guardrails

- **Log transform (optional):** If variance increased with price level, we fit ARIMA on  $\log(\text{price})$  and exponentiated forecasts back to price space.
- **Outliers:** Large, isolated jumps were not removed (they're part of market reality), but we verified they didn't break optimization.
- **Stability check:** We compared the chosen ARIMA to a naive "last value" baseline. The model was only considered useful if it consistently beat the naive forecast on MAE/RMSE.

## V. RESULTS

After building and running the ARIMA model in Python, the system was able to successfully analyze and predict the stock's future prices based on its past performance. The dataset covered two years of daily stock price data, and to make the testing fair, most of the data were used for training while the last portion was kept aside for testing the model's accuracy. Before training, the data were carefully processed to make them suitable for the ARIMA model. Several combinations of model parameters were tested, and the one that gave the best performance — based on the lowest AIC value — was **ARIMA(2, 1, 2)**. Once the model was trained, it was used to predict the stock prices for the testing period. The predicted prices were then compared with the actual stock prices. The model performed quite well, capturing the overall upward and downward movements of the stock with reasonable accuracy. Visually, when the actual and predicted prices were plotted together, the two lines almost overlapped, showing that the ARIMA model could closely follow the stock's real trend. However, the model did show some small differences during sudden spikes or dips in the stock price — times when the market was more volatile. This is expected since the ARIMA model is based only on past price behavior and doesn't take into account external factors such as news events or investor sentiment. Overall, the results show that the ARIMA model is a reliable and practical tool for **short-term stock price forecasting**. It effectively learns patterns from historical data and produces accurate predictions, especially in stable market conditions. These results suggest that statistical forecasting models like ARIMA can provide valuable support for financial decision-making trend analysis



**Fig 3 :** . Recent and prediction stock display

## CONCLUSION

This project uses a Python-based model with the ARIMA algorithm to predict future stock prices by studying past data and spotting time-series trends. ARIMA is powerful because it can capture linear trends and seasonal patterns, making it a dependable method for short-term forecasts. However, since stock markets are affected by many unpredictable and nonlinear factors, the model's accuracy can fluctuate when sudden changes happen. Overall, this project shows how combining machine learning with statistical modeling in Python can help investors make smarter financial decisions and provides a strong foundation for developing more advanced hybrid forecasting models in the future.

## FUTURE ENHANCEMENT

ARIMA models for stock price prediction can be made even smarter by combining them with advanced deep learning methods like LSTM or GRU networks. These hybrid approaches would help the system understand both simple and complex patterns in stock data. Adding real-time information, insights from financial news and social media, and automatic updates to the model can make predictions more accurate and responsive to market changes. Enhancing the model with important financial signals like inflation, GDP, and trading volume—and expanding it to track groups of stocks instead of just one—can give a more complete picture of the market. Finally, creating a user-friendly dashboard with tools like Flask or Streamlit will let investors and analysts interact with the system easily, making it practical and accessible for everyday use.

## REFERENCES

- [1] Anchal Vij 1, Komal Saxena<sup>2</sup> Ajay Rana<sup>3</sup> AIIT, Amity University Uttar Pradesh, Noida, India. Prediction in Stock Price Using of Python and Machine Learning(2021)
- [2] Sumeet Sarode<sup>1</sup>, Harsha G. Tolani<sup>2</sup>, Prateek Kak<sup>3</sup>, Lifna C S<sup>4</sup> Computer Engineering Department, Vivekanand Education Society's Institute of Technology, Mumbai, India STOCK PRICE PREDICTION USING MACHINE LEARNING TECHNIQUES
- [3] Yi Wei Department of Computer Science and Engineering University at Buffalo Buffalo, USA yiwei@buffalo.edu. and Vipin Chaudhary Department of Computer Science and Engineering University at Buffalo Buffalo, USA vipin@buffalo.edu The Directionality Function Defect of Performance Evaluation Method in Regression Neural Network for Stock Price Prediction
- [4] I. A. Moosa and J. J. Vaz, "Direction accuracy, forecasting error and the profitability of currency trading: Simulation-based evidence-accuratezza direzionale, errore previsionale e convenienza del currency trading: evidenze dalle simulazioni", *Economia Internazionale / International Economics*, 67(3): 413-423, 2014.
- [5] H. White, "Economic prediction using neural networks: the case of IBM daily stock returns", *Proceedings of the Second IEEE Annual Conference on Neural Networks*, 2: 451-458, 1988.
- [6] M.Vijh, D.Chandola's "Stock Closing Price Prediction using Machine Learning Techniques". *International Conference on Computational Intelligence and Data Science (ICCIDS)*.
- [7] K.Daniel's "A Study on Stock Price Prediction Using Linear Regression and Random Forest Regression" (2022).
- [8] O. E. Orsel, S. S. Yamada-"Comparative Study of Machine Learning Models for Stock Price Prediction" (2022).
- [9] J.Sen-"Stock Price Prediction Using Machine Learning and Deep Learning Framework", 6th International Conference on Business Analytics and Intelligence (ICBAI). Published in December 2018.
- [10] I.Kumar, C.Utreja's "A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction". *IEEE Conference*. (2018).