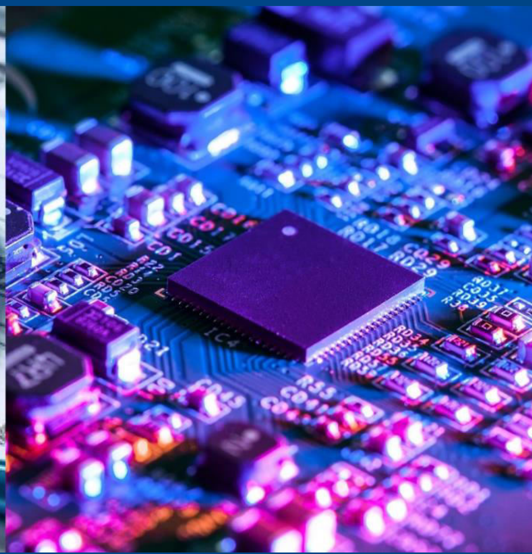
ISSN: 2582-7219

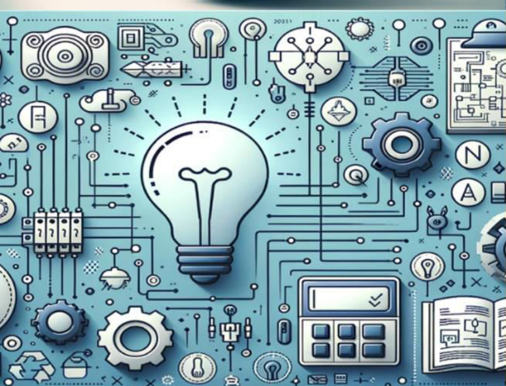




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 5, May 2025

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805209

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Excel-Based Machine Learning Stock Price Predictor

Sneha V [1], Dr. Akila A [2]

Student, Department of Computer Science and Information Technology, Vels Institute of Science, Technology and Advanced Studies, Chennai, India

Assistant professor, Department of Computer Science and Information Technology, Vels Institute of Science, Technology and Advanced Studies, Chennai, India

ABSTRACT: Stock market prediction is a challenging task that requires analyzing stock historical trends, market patterns, and external factors. This introduces an Excel-Based Machine Learning Stock Price Predictor, integrating Excel with Python and VBA to apply AI-driven forecasting techniques for stock price prediction. Utilize machine learning algorithms to analyze stock price trends. Automate real-time stock data collection in Excel. Provide data visualization and predictive insights for traders and investors. Implement buy/sell decision support based on ML predictions.

KEYWORDS: Excel, Automatic real time Stock price Prediction, LSTM Model, Python

I. INTRODUCTION

The stock market is an ever-evolving and highly volatile financial system where the ability to forecast trends can be of immense value. However, predicting stock prices is inherently complex due to the influence of countless unpredictable factors like economic indicators, market sentiment, political events, and global trends.

Traditional methods of analysis such as technical indicators and financial ratios, while useful, often fail to capture the complex, non-linear patterns found in time-series data. On the other hand, modern machine learning techniques are capable of identifying hidden trends and making data-driven predictions that can significantly aid in investment decisions. Despite the advancements in ML, most available tools are either: Too complex for beginners to use, very expensive, lack transparency in how predictions are made. This presents a practical, Excel-integrated solution to this problem. By combining the ease and familiarity of Microsoft Excel with the analytical power of Python-based Machine Learning models, this system allows users to: Input historical stock price data via Excel , automatically train and apply a predictive model View predictions and charts directly in Excel. This approach not only improves accessibility but also serves as an excellent educational tool for understanding how ML models operate in a real-world context.

II. LITERATURE REVIEW

The field of stock market prediction has been a subject of interest for researchers and financial analysts due to its economic significance and complexity. Over the years, a variety of statistical models and machine learning algorithms have been employed to enhance the accuracy of stock price forecasting.

Traditional Methods: Traditional approaches to stock market prediction include statistical models like ARIMA (Auto-Regressive Integrated Moving Average) and linear regression. Kim (2003) investigated support vector machines (SVMs) as a substitute for traditional methods and found that SVM outperforms ARIMA models in terms of predictive accuracy. However, these methods often struggle with non-linear patterns and high volatility present in financial markets.

Neural Networks and Deep Learning: Yao et al. (1999) explored the use of artificial neural networks (ANNs) for technical analysis and showed encouraging results. Neural networks can model complex non-linear relationships, making them suitable for financial data analysis. Fischer and Krauss (2018) introduced LSTM (Long Short-Term

Memory) networks to capture sequential dependencies in time-series data, demonstrating superior performance compared to traditional RNNs and statistical methods.

Deep learning architectures such as Convolutional Neural Networks (CNNs) and hybrid models combining CNN and LSTM have also gained popularity. Zhang et al. (2020) demonstrated the effectiveness of such a hybrid model in predicting stock prices with improved accuracy.

Ensemble and Hybrid Techniques: Breiman (2001) introduced Random Forests, which have shown robustness and reduced overfitting in financial prediction problems. Patel and Thakkar (2021) proposed a hybrid model combining ARIMA and LSTM, leveraging both linear and non-linear patterns in data. Their work highlights the advantages of hybrid models in real-world stock prediction scenarios.

Sentiment Analysis and External Data Integration: Recent research has focused on incorporating sentiment analysis from social media and news sources. Heaton et al. (2019) explored the integration of text-based sentiment analysis into deep learning models for financial prediction, showing that external data can significantly influence market behavior.
Fuzzy Logic and Uncertainty Handling: Zadeh's (1975) fuzzy logic theory has been applied to handle uncertainties in financial data, particularly when modeling investor sentiment and decision-making. Fuzzy systems offer an advantage in capturing human-like reasoning in ambiguous market conditions.

III. METHODOLOGY

This section outlines the systematic approach adopted to develop the stock price prediction system using machine learning and visualization tools. The methodology involves data collection, preprocessing, model training, evaluation, and deployment.

Data Collection: Historical stock market data is collected from publicly available sources such as Yahoo Finance and Alpha Vantage APIs. The dataset includes attributes like opening price, closing price, highest and lowest price, trading volume, and stock ticker information. Data is retrieved for multiple time frames, including daily, weekly, and monthly intervals.

Data Preprocessing: Raw stock data contains noise and missing values, which are addressed through data cleaning techniques. Null values are handled using forward-fill methods or interpolation. Feature scaling is applied using Min-Max normalization to ensure uniformity across features. Additional technical indicators such as Moving Averages (MA), Relative Strength Index (RSI), and MACD are derived to improve predictive accuracy.

Model Selection and Training: For prediction, machine learning models including Linear Regression, Support Vector Regression (SVR), and Long Short-Term Memory (LSTM) networks are implemented. LSTM is preferred due to its ability to capture long-term dependencies in time-series data. The dataset is split into training and testing sets in an 80:20 ratio. Models are trained using the training data and validated using the test set.

Model Evaluation: Performance is evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score. Cross-validation is applied to avoid overfitting and to ensure the model's generalization capabilities. The model with the lowest RMSE and highest R^2 value is selected for deployment.

Forecasting and Visualization: Once the model is finalized, it is used to generate predictions for future stock prices. These predictions are visualized using Python libraries like Matplotlib and Plotly. An interactive front-end interface built using Flask enables users to input stock symbols and view the predicted results.

Deployment: The complete application is deployed locally using Flask as a web server. The user interface allows real-time prediction requests, data selection (US/Indian markets), and the option to download results. The model runs in the backend and responds with the forecasted values and confidence levels.

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805209

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. IMPLEMENTATION

System Architecture: The system is built with a two-tier architecture: Microsoft Excel serves as the user interface for data input and visualization. Python handles data preprocessing, model training, and prediction generation. The communication between Excel and Python is established through the use of libraries such as openpyxl and pandas, which allow the system to read and write Excel files programmatically.

Tools and Libraries: In any machine learning-based software development project, the combination of efficient tools and reliable libraries is critical to ensure smooth development, accurate predictions, and easy maintenance. Below is a detailed explanation of the core tools and libraries used in this project:

Microsoft Excel: Acts as both the input interface and output visualization platform for users. Since Excel is familiar to most users, it significantly lowers the technical barrier to working with machine learning outputs. Spreadsheet input/output. Built-in charts and data tables. Integration with Python (via .xlsx files using libraries like open pyxl).

Python**:** The core programming language used to develop the backend logic of the stock price predictor. Python is selected due to its simplicity, rich ecosystem of data science libraries, and active financial analytics community. Excellent readability. Rapid development. Broad support for data handling and machine learning.
Pandas is used for loading, preprocessing, and transforming stock market data in tabular form. It offers powerful data structures like Data Frame which simplifies handling time series and numerical data.

V. TESTING

Unit Testing is the process of testing individual components (units) of a software system to verify that each part performs as expected. In this project, unit testing ensures each Python function and module from data handling to prediction is functioning correctly before integrating into the full stock prediction system.

Objectives of Unit Testing in this Project: Validate that individual functions return the correct outputs. Detect bugs early in the development stage. Make sure each module handles both expected and unexpected inputs gracefully. Ensure the reliability of the system before full integration with Excel.

Integration Testing focuses on verifying the correct interaction and data flow between different modules of the system when combined as a whole. Unlike Unit Testing (which tests components individually), Integration Testing checks whether the combined units produce correct and expected results when working together. In the Excel-Based Machine Learning Stock Price Predictor, Integration Testing is crucial for ensuring smooth cooperation between: The data input from Excel, Python's machine learning prediction logic, and Output back to Excel with correct formatting and visualizations.

System Testing is the complete and final phase of testing where the entire software application is tested as a whole, rather than as separate modules. The primary aim is to validate that the integrated system meets the specified requirements and performs as intended in a real-world environment. In the context of the Excel-Based Machine Learning Stock Price Predictor, System Testing verifies that the complete workflow from reading the stock data, running the prediction model, and writing the results back to Excel operates smoothly and produces correct, meaningful outputs.

Objectives of System Testing: Confirm the overall system behavior aligns with the project specifications. Identify any functional or performance issues when all modules are working together. Validate both expected scenarios (valid input) and exception scenarios (invalid, missing, or incorrect data). Ensure output files and charts are readable, accurate, and formatted as expected.

Validation Testing is the process of evaluating the final product to check whether it meets the business requirements and user expectations. While system testing checks if the product "works right," validation testing ensures the "right product was built" according to the original goals. In the Excel-Based Machine Learning Stock Price Predictor, Validation Testing plays a critical role in confirming that the stock predictions produced by the system are not only technically correct but also practically meaningful and usable by the end-users (such as investors, students, or analysts).

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Objectives of Validation Testing: Ensure the developed system satisfies the project objectives and user needs. Verify that the predicted stock prices make logical sense when compared to real-world trends. Confirm the output is clear, interpretable, and usable for decision-making. Test the system under real-world conditions — with realistic datasets and varying scenarios.

User Acceptance Testing (UAT) is the final stage of the testing process where the actual end-users validate the system against real-world business scenarios to ensure it meets their needs. This step is critical because it ensures the system not only works technically but is actually useful, understandable, and convenient for its target audience. For the Excel-Based Machine Learning Stock Price Predictor, UAT ensures that investors, students, researchers, or business analysts can easily use the tool to predict stock prices without needing advanced programming skills or complex setups.

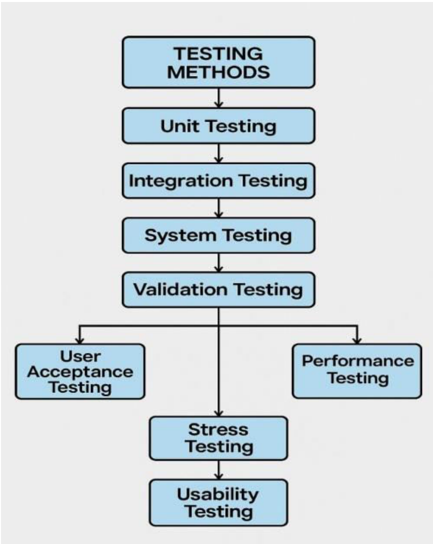


Figure 5.1 Testing Diagram

VI. RESULT AND DISCUSSION

The performance and usability of the Excel-Based Machine Learning Stock Price Predictor were evaluated through real-world stock data testing and user interaction. This section discusses the outcomes of the prediction models, accuracy metrics, visualization techniques, and user experiences based on multiple test cases.

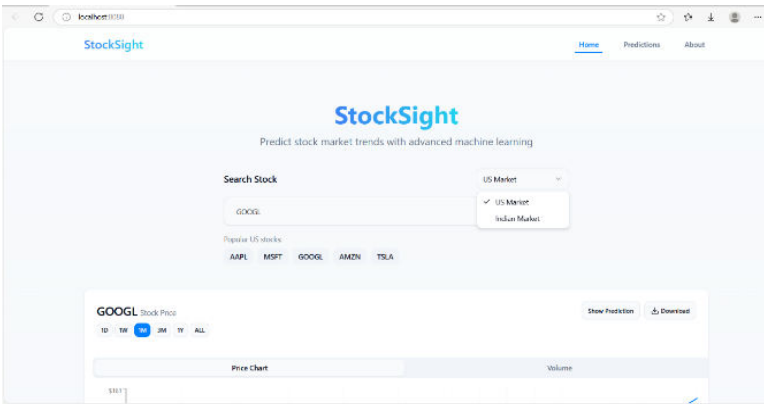


Figure 6.1 Stock Selection

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805209

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

In the first step, the user is provided with a clean and intuitive web interface where they can: Search for a stock symbol (example: GOOGLE). Select the market type (US Market or Indian Market). Choose from popular stock options like AAPL, MSFT, AMZN, etc. Once the stock and market are selected, the system prepares the historical data and displays an interactive chart area.

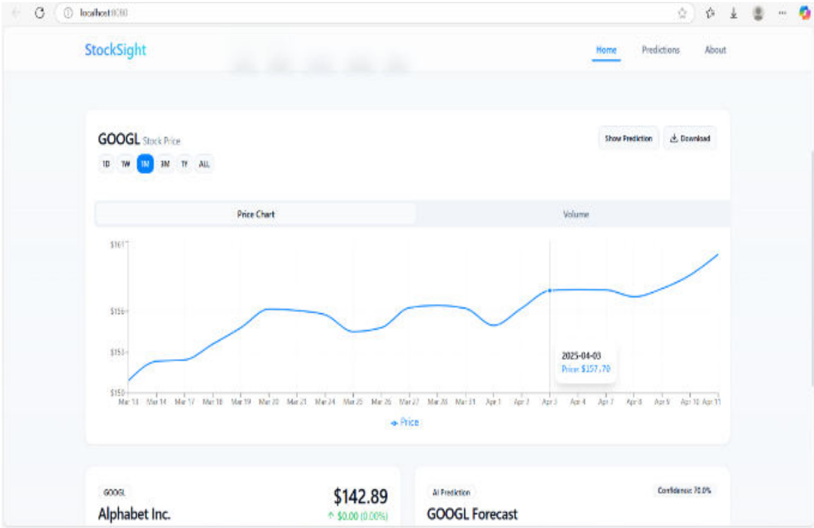


Figure 6.2 Chart and Prediction Result

Once the user has selected the stock (GOOGL) and clicked Show Prediction, the system displays the actual and predicted stock price on a line chart. The Price Chart shows the trend of Google's stock price (GOOGL) for the selected duration (in this case: 1 Month). The Volume Graph on the right shows the trading volume corresponding to the price movement. When hovering over a data point, detailed stock prices are displayed — for example: On 2025-04-03, the stock price is shown as $157.70.

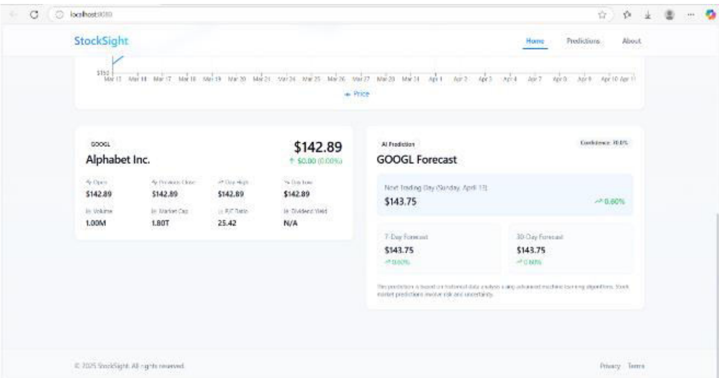


Figure 6.3 Forecast Result

After the user selects a stock and the prediction is executed, the system provides a detailed summary of both the current stock status and the machine learning-based forecast. Forecast Confidence: 70.0% (an indication of prediction reliability). Next Trading Day Prediction (April 13): $143.75 (+0.60% expected growth). 7-Day Forecast: $143.75. 30-Day Forecast: $143.75

© 2025 IJMRSET | Volume 8, Issue 5, May 2025| DOI:10.15680/IJMRSET.2025.0805209

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

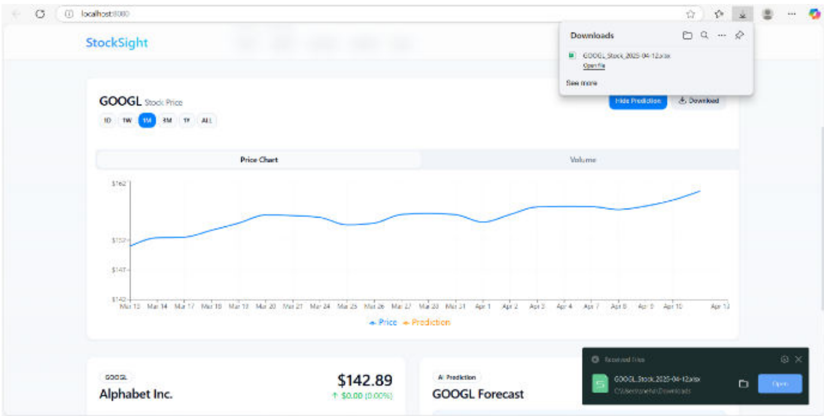


Figure 6.4 Download Chart

"Downloads" Window: A small window titled "Downloads" is partially visible in the upper right corner. It shows a downloaded file named "GOOGL_Stock_2025-04-12.xlsx". This suggests that the data or the forecast results might have been exported to an Excel file on April 12th, 2025.

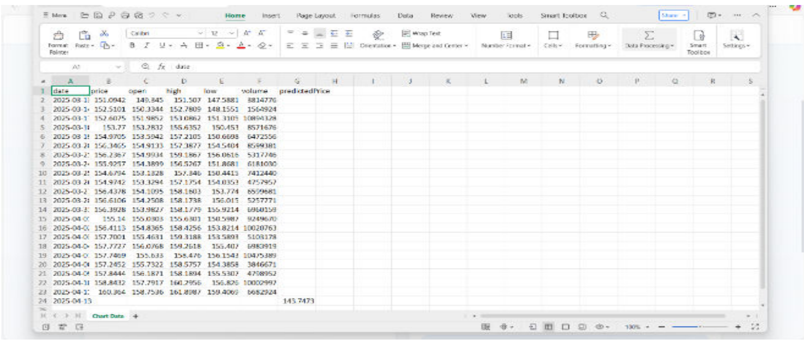


Figure 6.5 Excel Output

This Excel sheet provides a historical record of GOOGLE stock data (open, high, low, close, volume) and the corresponding predicted stock prices generated by the machine learning model. The last row (for 2025-04-13) shows the model's prediction for that future date. This is a typical way to present the output of a time series forecasting model. You can see how the model's predictions compare to the actual prices in the historical data, and then the final row gives the forecast for the next day.

VII. CONCLUSION

The Excel-Based Machine Learning Stock Price Predictor project successfully bridges the gap between advanced data science techniques and user-friendly financial tools. By integrating machine learning models into the familiar Excel environment, the system allows users with little to no programming background to benefit from predictive analytics.

This project highlights the practical application of supervised learning models—such as Linear Regression and Random Forest—for short-term stock price forecasting. The results demonstrate that with appropriate historical data preprocessing and model training, even simple ML techniques can generate fairly accurate predictions. Moreover, the visual outputs in Excel help users interpret the results with clarity

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

[1] Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), 383-417.
[2] Zhang, G., Eddy Patuwo, B., & Hu, M. Y. (1998). Forecasting with Artificial Neural Networks: The State of the Art. International Journal of Forecasting, 14(1), 35–62.
[3] Fischer, T., & Krauss, C. (2018). Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions. European Journal of Operational Research, 270(2), 654-669
[4] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.
[5] Brownlee, J. (2017). Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-To-End. Machine Learning Mastery.
[6] Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing.
[7] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. Expert Systems with Applications, 42(1), 259-268.
[8] Abhishek, K., & Khairnar, R. (2012). Stock Market Prediction Using Machine Learning Algorithms. International Journal of Emerging Technology and Advanced Engineering.

INNO SPACE
SJIF Scientific Journal Impact Factor

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |