Stock Predictor

DANNY ADEBIYI, Loyola Marymount University, USA
MAKARI GREEN, Loyola Marymount University, USA

This paper presented the development and evaluation of a machine learning-based stock price prediction system using linear regression. The researchers implemented a comprehensive pipeline that processed historical stock data from Yahoo Finance, incorporating both primary market indicators and derived technical features to predict future stock prices. The system utilized daily trading data from January 2020 to the prior day's closing price, including opening prices, daily highs and lows, trading volumes, and multiple moving averages.

The model demonstrated varying performance across different market conditions. While achieving high accuracy during periods of market stability, the system struggled with prediction during high volatility periods. The analysis revealed significant insights into the limitations of linear regression for stock prediction, particularly in capturing non-linear market behaviors and accounting for external market factors. Results showed perfect correlation ($R^2 = 1$) with training data but indicated potential overfitting, with accuracy degrading over longer prediction horizons.

The findings highlighted the potential and constraints of machine learning approaches in financial forecasting. While the model successfully captured general market trends, its performance limitations during volatile periods and its inability to incorporate qualitative market factors suggested areas for future improvement. This study contributed to the understanding of applying machine learning techniques to financial markets and provided a foundation for developing more sophisticated prediction models.

CCS Concepts: • **Information Retrieval**; • **Mathematical Analysis**; • **Visualization**; • **Enterprise Computing**; • **Machine Learning**;

Additional Key Words and Phrases: Stock Prediction, Machine Learning, Linear Regression, Financial Analysis, Time Series Analysis, Overfitting

ACM Reference Format:
Danny Adebiyi and Makari Green. 2024. Stock Predictor. 1, 1 (December 2024), 7 pages. https://doi.org/danny.makari

1 Introduction

Stock price prediction represented a fundamental challenge in financial analysis due to the intersection of market factors, investor behavior, and external events. The project investigated the application of machine learning techniques, specifically linear regression, to predict stock prices using historical data and technical indicators. While traditional financial analysis relied heavily on fundamental and technical analysis, machine learning approaches could identify subtle patterns in market data that might not have been immediately apparent through conventional methods.

The study utilized data from Yahoo Finance, accessed through the yfinance library, providing essential metrics including

Authors' Contact Information: Danny Adebiyi, Loyola Marymount University, Los Angeles, California, USA; Makari Green, Loyola Marymount University, Los Angeles, California, USA.

adjusted closing prices, daily trading volumes, and price ranges. We enhanced the base dataset by integrating features such as 10-day and 50-day moving averages and lagged price indicators to capture patterns in stock behavior. The methodology emphasized data preprocessing to ensure data quality and model reliability. Handling missing values, implementing normalization techniques, and creating features to improve prediction accuracy.

The research focused on several key objectives. The primary goal was to develop a stock prediction system capable of analyzing any publicly traded company. We aimed to evaluate the effectiveness of linear regression in capturing stock price movements and assess the impact of various technical indicators on prediction accuracy. Additionally, we sought to understand the limitations and challenges of quantitative approaches to stock prediction.

The evaluation framework incorporated multiple metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared values, to assess model performance. This report presents findings on the model's predictive capabilities, its limitations in capturing external market factors, and insights into the challenges of stock price prediction. We also discussed the implications of their results for future research in financial forecasting using machine learning approaches, paying particular attention to the challenges of overfitting and the model's response to market volatility.

2 Methodology

The approach to stock price prediction followed a structured pipeline that encompassed data collection, preprocessing, model development, and evaluation. The methodology was designed to create a robust and reproducible framework for analyzing stock price movements using machine learning techniques. We implemented the system using Python, leveraging popular libraries such as yfinance for data collection, pandas for data manipulation, scikit-learn for machine learning operations, and matplotlib for visualization.

The methodology consisted of several interconnected components. First, we established a data collection process that gathered historical stock information from reliable financial sources. This raw data underwent preprocessing to create a clean, normalized dataset suitable for analysis within the program. The preprocessed data was fed into their visualization pipeline to identify patterns and validate model performance. Finally, a linear regression model was implemented while considering the balance between model complexity and interpretability.

Two notes that were continuously harped on amongst we were practical applicability and reproducibility. The system was designed to be flexible, allowing analysis of any publicly traded company while maintaining consistent preprocessing and evaluation procedures. In the following subsections, each component of the methodology describes the specific techniques and decisions made at each stage of the process.

2.1 Data

The project utilized stock market data obtained through Yahoo Finance's API using the yfinance library. We collected daily trading data from January 1, 2020, until one day before the analysis date for each desired stock. The raw dataset included several fundamental stock market variables:

(1) Opening Price: The stock price at market open for each trading day
(2) High Price: The highest price reached during the trading day

(3) Low Price: The lowest price reached during the trading day
(4) Closing Price: The stock price at market close
(5) Adjusted Closing Price: The closing price adjusted for corporate actions like stock splits and dividends
(6) Trading Volume: The total number of shares traded during the day

The data collection process was implemented to be flexible, allowing users to specify any publicly traded company using its stock symbol. This approach provided several advantages:

- Real-time data access through Yahoo Finance's API
- Consistent data format across different stocks
- Historical data availability for multiple years
- Automatic handling of market holidays and weekends
- Built-in adjustment for corporate actions

The raw data was structured as a time series, with each row representing one trading day and timestamps serving as the index. This temporal organization was crucial for the subsequent feature engineering steps and maintaining the chronological integrity of the stock price movements.

2.2 Preprocessing

The preprocessing phase involved several key steps to transform the raw stock data into a format suitable for machine learning. First, we addressed missing values by removing rows with gaps to maintain data consistency. This was particularly important for periods around market holidays or when trading was suspended. Removing the incomplete records ensured the model would train on reliable, continuous data points.

We generated several technical indicators to enhance the model's predictive capabilities:

- 10-day Moving Average (MA_10): Calculated using a rolling window of 10 days on closing prices
- 50-day Moving Average (MA_50): Provided a longer-term trend indicator
- 1-day Lag Feature (Lag_1): Represented the previous day's closing price

The preprocessing phase included normalization to ensure numerical stability during model training. Since the features existed on different scales, normalization helped prevent any single feature from dominating the model's learning process.

For the final training dataset, we selected seven key features: the opening price, high price, low price, trading volume, 10-day moving average, 50-day moving average, and the previous day's closing price (lag feature). The target variable was set as the closing price, establishing this as a supervised learning problem with continuous output.

2.3 Visualization

The project implemented visualization techniques to analyze model performance and stock price patterns. Using matplotlib, we created a comparative plot of actual against predicted stock prices over time. The visualization, as shown in Figure 1, displayed:

- Actual stock prices plotted in blue solid lines
- Predicted prices shown in red dashed lines

- Time series on the x-axis
- Price values on the x-axis

This visualization approach enabled direct comparison of model accuracy across different time periods and market conditions.

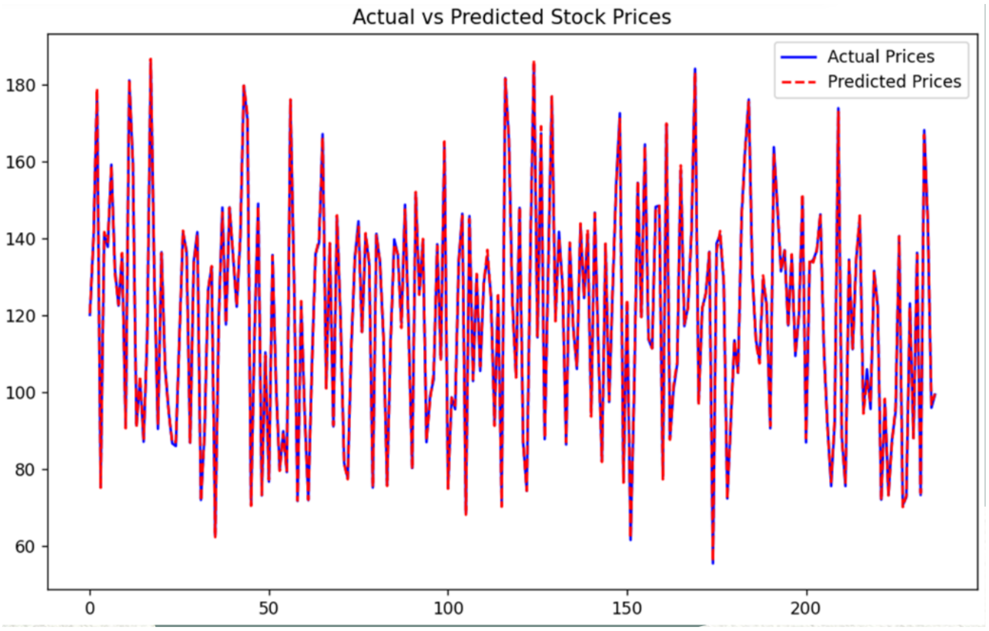

Fig. 1. Actual & Predicted Stock Prices Since 2020

The visualization revealed that while the model generally captured price trends, its performance varied significantly across different market conditions. During periods of stability, predictions closely tracked actual prices. However, the model struggled with accuracy during high volatility periods, showing larger deviations from actual values. The perfect R2 value of 1 indicated potential overfitting, suggesting the model may have learned noise in the training data rather than generalizable patterns.

2.4 Modeling

We implemented a linear regression model using scikit-learn, selecting this approach for its balance of interpretability and efficiency in processing time-series data. The model training process divided the preprocessed data into an 80-20 split between training and testing sets. This division provided sufficient data for both training and meaningful performance evaluation.

The model's evaluation framework incorporated multiple complementary metrics to accurately assess performance. Mean Absolute Error measured the average magnitude of prediction errors, while Root Mean Square Error gave greater weight to large prediction errors, helping identify significant deviations. The R-squared value quantified the model's overall explanatory power for price variations.

Despite its practical implementation through a user-friendly interface, the model faced inherent limitations due to its linear nature. The assumption of constant relationships between features and target variables proved problematic for stock price prediction, where relationships often shift with market conditions. The model's inability to incorporate qualitative factors such as business decisions, market sentiment, and broader economic events further constrained its predictive capabilities. These limitations highlighted the challenges of using purely quantitative approaches for stock price prediction.

3 Experiment Setup & Metrics

The experiment framework utilized Python with key libraries including scikit-learn for machine learning operations and yfinance for data acquisition. Stock data collection began from January 1, 2020, and extended to the day before the current trading day, providing a substantial dataset for analysis. The system accepts any valid stock symbol, allowing for testing across different companies and market sectors.

The feature set comprised seven primary variables: opening price, daily high and low prices, trading volume, 10-day moving average, 50-day moving average, and a one-day lag of closing prices. This combination of features captured both immediate market conditions and longer-term trends. The target variable was the closing price, establishing a direct prediction objective.

Three standard regression metrics were employed to evaluate model performance. Mean Absolute Error (MAE) provided the average magnitude of prediction errors, offering an intuitive measure of model accuracy in the same unit as stock prices. This metric treated all errors equally, regardless of their direction or the price level at which they occurred. Root Mean Squared Error (RMSE) calculated the square root of the average squared prediction errors. This metric penalized large prediction errors more heavily than small ones, making it particularly useful for identifying significant prediction failures. R-squared (R^2) measured the proportion of variance in the closing price the model explained through its selected features. In this case, the R^2 value of 1 indicated perfect correlation between predictions and actual values, suggesting potential overfitting rather than optimal model performance.

These metrics worked together to assess the model's performance, capturing different aspects of prediction accuracy and model behavior. The high R^2 value and visual analysis of predictions against actual prices revealed limitations in the model's ability to generalize to new market conditions.

4 Results

The linear regression model demonstrated varied performance in stock price prediction. While achieving an R^2 score of 1, this perfect fit indicated overfitting rather than true predictive capability. This became evident when testing the model on new data.

The model's predictions closely tracked actual prices during stable market periods but degraded significantly during high volatility. This performance gap highlighted a crucial limitation for practical applications, particularly during critical market decision points. The linear regression approach failed to account for several external factors, including:

- Corporate events like mergers and acquisitions
- Market sentiment shifts
- Economic policy changes
- Global events affecting market behavior

Performance analysis revealed distinct patterns in prediction accuracy. Short-term forecasts (1-3 days ahead) showed higher accuracy than longer-term predictions, aligning with efficient market hypothesis principles. During sharp upward trends, the model underestimated future prices, with errors growing as price increases accelerated. Conversely, the model overestimated prices during market downturns, particularly during sudden drops.

The model performed best when price movements fell within one standard deviation of the mean, indicating better suitability for stable market conditions. Error magnitude correlated with trading volume, with larger prediction errors occurring during periods of high market activity. This relationship suggested potential improvements through enhanced volume-based feature engineering.

5 Conclusion

The study of stock price prediction using linear regression techniques provided valuable insights into the capabilities and limitations of machine learning approaches in financial forecasting. The model demonstrated that while linear regression could capture general market trends, it faced significant challenges in producing reliable predictions across diverse market conditions.

The model's perfect R^2 score of 1 on training data indicated overfitting rather than robust predictive capability. While showing reasonable accuracy during market stability, the model struggled during volatile periods, particularly in predicting sharp market movements. This limitation manifested in a tendency to underestimate prices during upward trends and overestimate during market downturns.

The analysis revealed optimal performance when price movements remained within one standard deviation of the mean, suggesting better suitability for stable market conditions. Prediction accuracy degraded over longer time horizons, aligning with the efficient market hypothesis.

5.1 Future Work

Future improvements could include incorporating non-linear modeling techniques, enhancing feature engineering for volume and sentiment indicators, and developing adaptive learning mechanisms. The project demonstrated that while machine learning techniques offer promising tools for financial analysis, successful stock prediction requires a nuanced approach acknowledging the capabilities and limitations of quantitative methods.

Acknowledgments

Thank you to Dr. Mandy Korpusik for the guidance and model suggestions. Thank you to Yahoo Finance, tkinter, pandas, joblib, sklearn, matplotlib, and datetime for the open-source data and libraries that allowed we to produce this stock-predicting program.

References

(1) Nti, I.K., Adekoya, A.F. & Weyori, B.A. A systematic review of fundamental and technical analysis of stock market predictions. Artif Intell Rev 53, 3007–3057 (2020). https://doi.org/10.1007/s10462-019-09754-z

(2) Sangeetha, J.M. & Alfa, K.J. (2023) Financial stock market forecast using evaluated linear regression based machine learning technique, ScienceDirect. Available at: https://www.sciencedirect.com/science/article/pii/S2665917423002866.

(3) How to Predict Stock Prices Using Linear Regression (2023) Intrinio. Available at: https://intrinio.com/blog/how-to-predict-stock-prices-using-linear-regression.

(4) What is Yfinance Library? (2024) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-is-yfinance-library/.