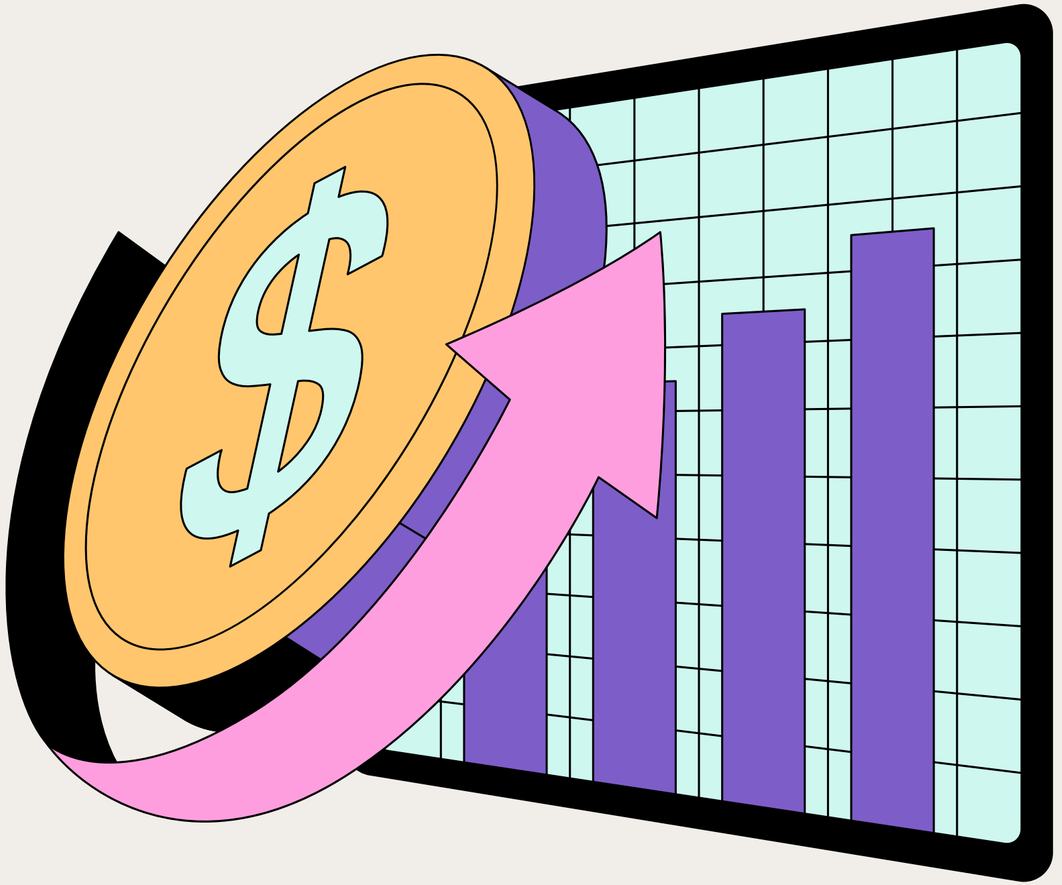
Stock Price Prediction

Using Machine Learning

Presented by Group 23, ECO723

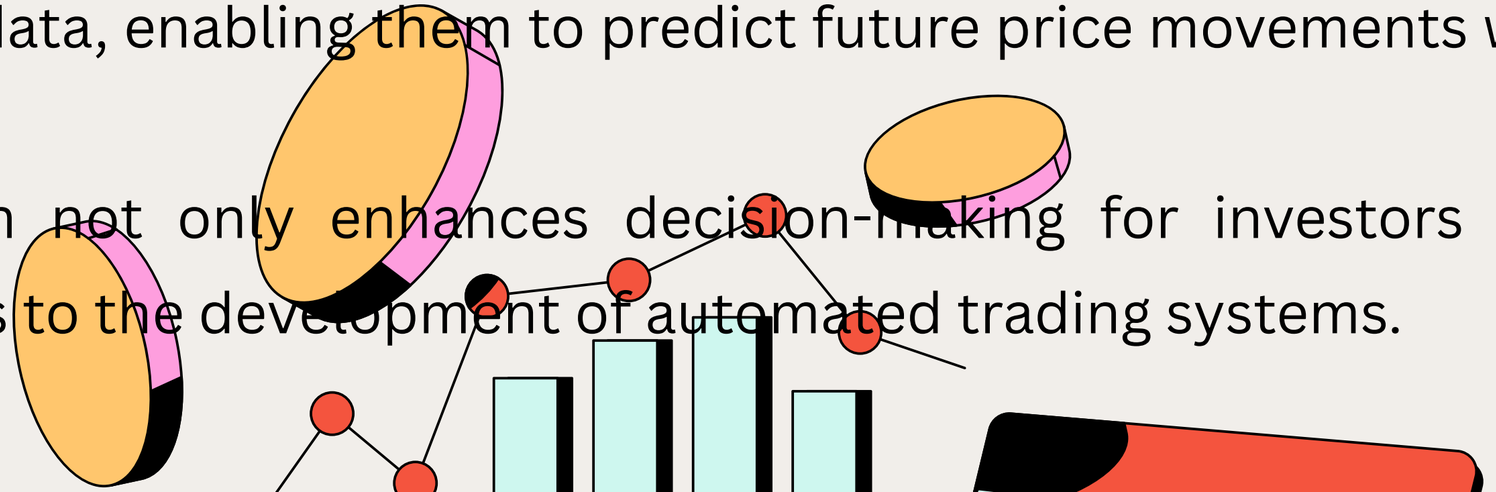


Overview

- INTRODUCTION
- MOTIVATION
- OBJECTIVES
- RESEARCH METHODS
- DATA SOURCE
- GRAPHICAL DATA STORY
- EMPIRICAL FINDINGS
- CONCLUSIONS
- LIMITATIONS
- RESOURCES
- REFERENCES

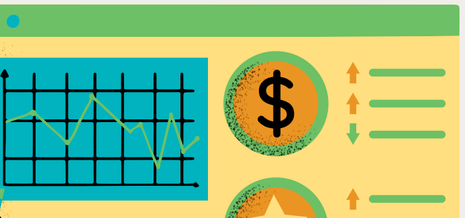
Introduction

- The prediction of stock prices using machine learning has emerged as a cutting-edge approach that leverages advanced computational techniques to analyze and forecast market trends.
- By employing algorithms capable of processing vast amounts of financial data, machine learning models can identify patterns and insights that might be missed by traditional methods.
- These models, ranging from linear regression to more complex neural networks, are trained on historical stock data, enabling them to predict future price movements with increasing accuracy.
- This innovative application not only enhances decision-making for investors and traders but also contributes to the development of automated trading systems.



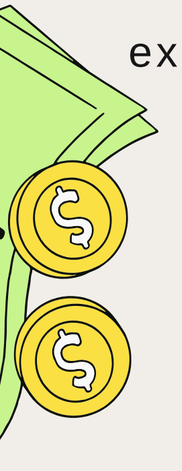
Motivation

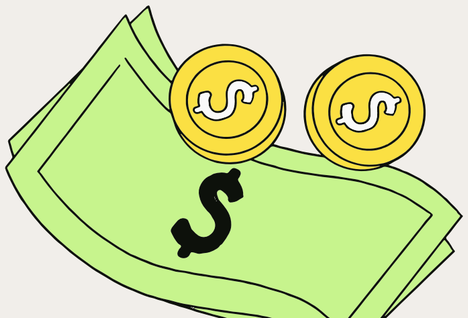
- In financial markets, traditional techniques often struggle with the complex challenge of non-linearity, where variable relationships aren't straightforward.
- This is where machine learning (ML) models like Long Short-Term Memory (LSTM) networks, random forest, linear regression come into play.
- LSTM is particularly skilled at capturing long-term dependencies in noisy financial series, characterized by unpredictable fluctuations and intricate patterns.
- By modeling these dependencies, machine learning models can help stakeholsers to take imformed desicions.



Objectives

- Apply an LSTM-based model to predict a company’s future stock prices, aiming for higher accuracy and reliability than traditional approaches.
- Compare LSTM forecasts with one classical machine-learning technique to demonstrate the accuracy gains achieved by the novel model.
- Assess the impact of training-data volume on prediction quality by envisioning tests with a much larger dataset than the one currently used.
- Identify promising research extensions, including sentiment analysis of news headlines and experimentation with other emerging deep-learning architectures.





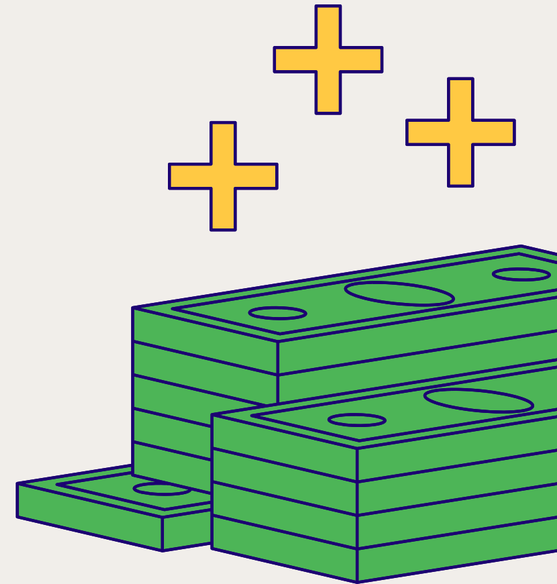
Research Methods

Preprocessing Pipeline:

- Normalization: StandardScaler implementation for feature scaling
- Train-Test Split: 95% training (5,136 obs) / 5% testing (270 obs)
- Sequence Creation: 60-day lookback windows for temporal pattern recognition
- Data Reshaping: 3D tensor preparation for LSTM input requirements

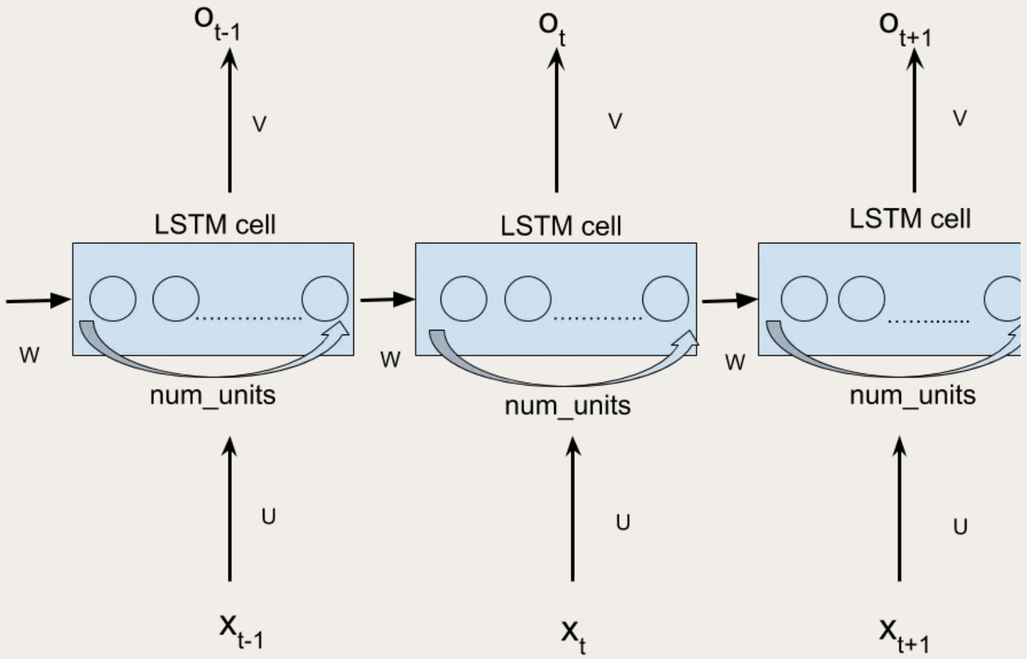
Feature engineering :

- Log return: To measure the volatility on the day.
- Seven days moving average: To take account of short term disruptions.



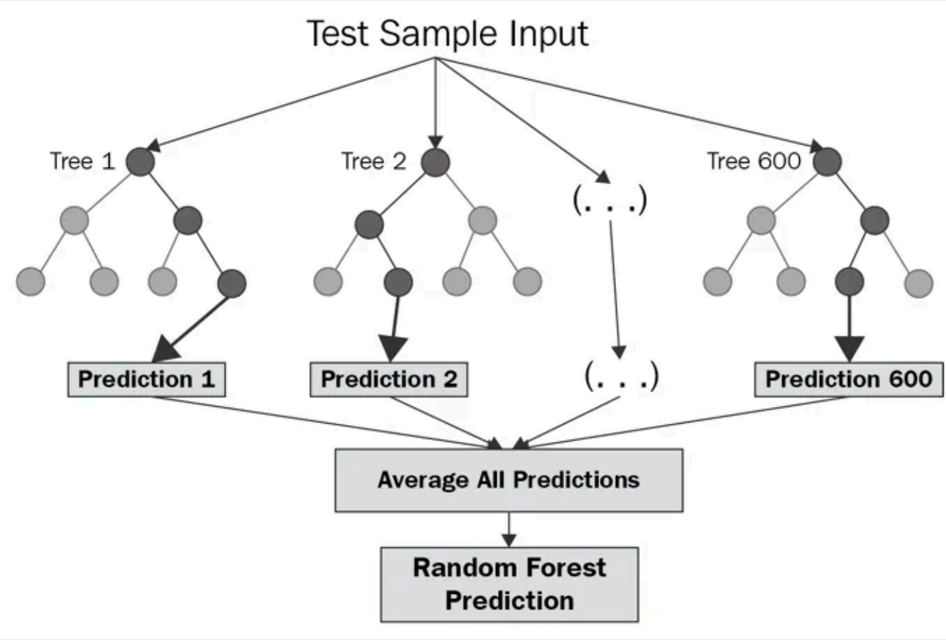
LSTM

- **Definition: Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture specialized in learning and retaining long-range dependencies in sequential data.**
- **Structure: Comprising memory cells and three key gates—input, output, and forget—LSTMs regulate information flow to maintain and update memory over time.**
- **Architecture: The model has 6 layers—2 LSTM layers (each with 50 units), 2 Dropout layers (0.01 rate), and 2 Dense layers—designed for sequence-to-one regression in time-series forecasting.**
- **Error Function: Mean Squared error**
- **Opitmiser : Adam**



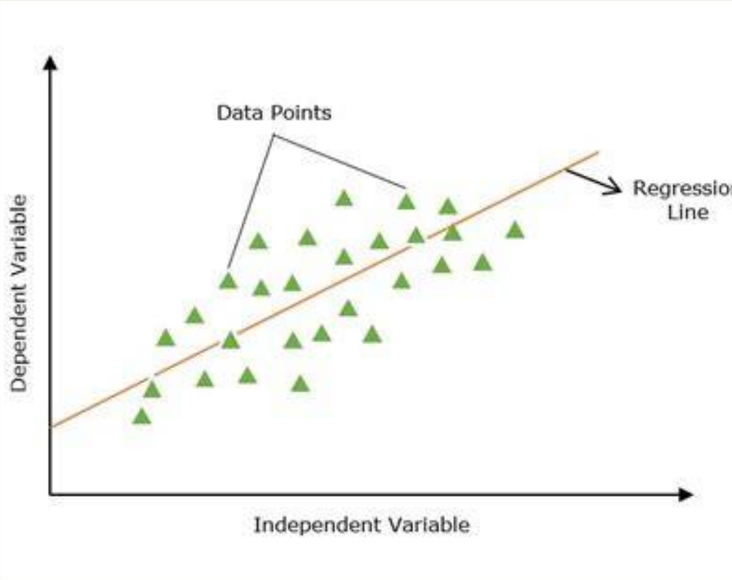
Random forest

- Defination: Random Forest is an ensemble machine learning algorithm used for classification and regression tasks.
- Mechanism: It creates multiple decision trees during training and merges their results to improve accuracy and prevent overfitting.
- Feature Selection: Each tree is built using a random subset of features, which enhances model robustness
- Voting System: For classification, it uses majority voting from the decision trees, while for regression, it averages the output.
- Error function: MAPE
- Artitechture : Has employed 100 desicion tree for training.



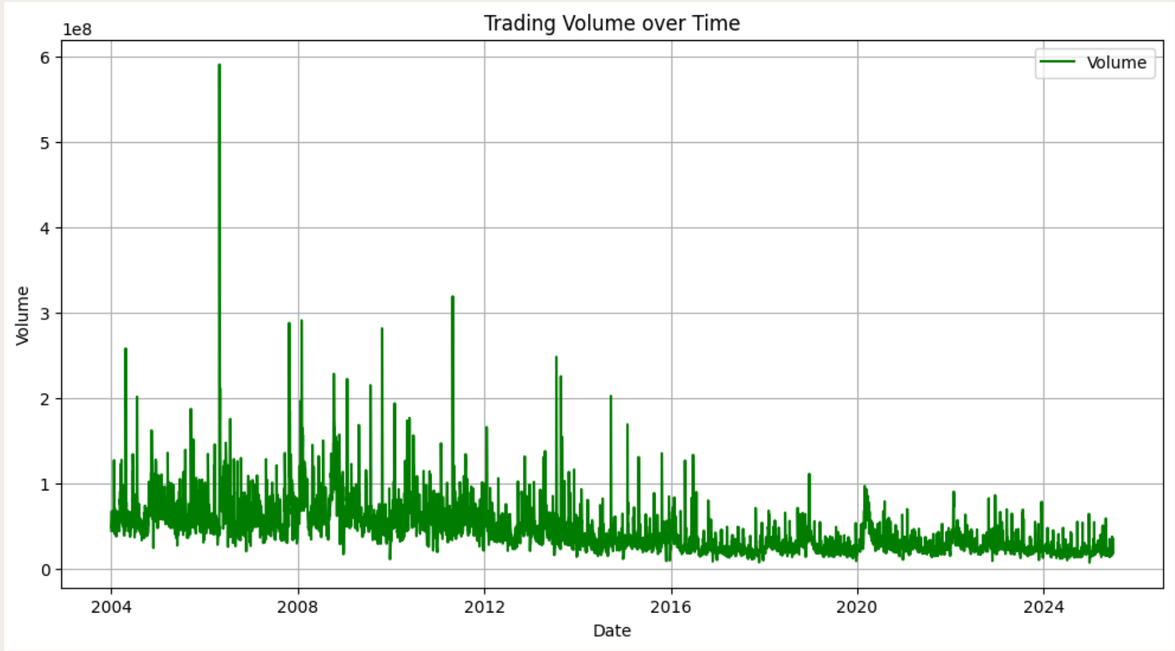
Linear Regression

- **Fundamental Concept: Linear regression models the relationship between a dependent variable and independent variables by fitting a linear equation to observed data. Simple linear regression involves one independent variable, while multiple linear regression includes two or more.**
- **Equation and Components: The basic equation is $Y = \beta 0 + \beta 1 X + \epsilon$, where (Y) is the dependent variable, $\beta 0$ is the y-intercept, β1 is the slope, (X) is the independent variable, and ϵ is the error term.**
- **Assumptions: Key assumptions include linearity, independence, homoscedasticity, normality of errors, and no multicollinearity. Violating these can lead to biased results.**
- **Applications: It is widely used in fields like economics, biology, engineering, and social sciences for predictive modeling and analysis, aiding in understanding trends and relationships between variables.**



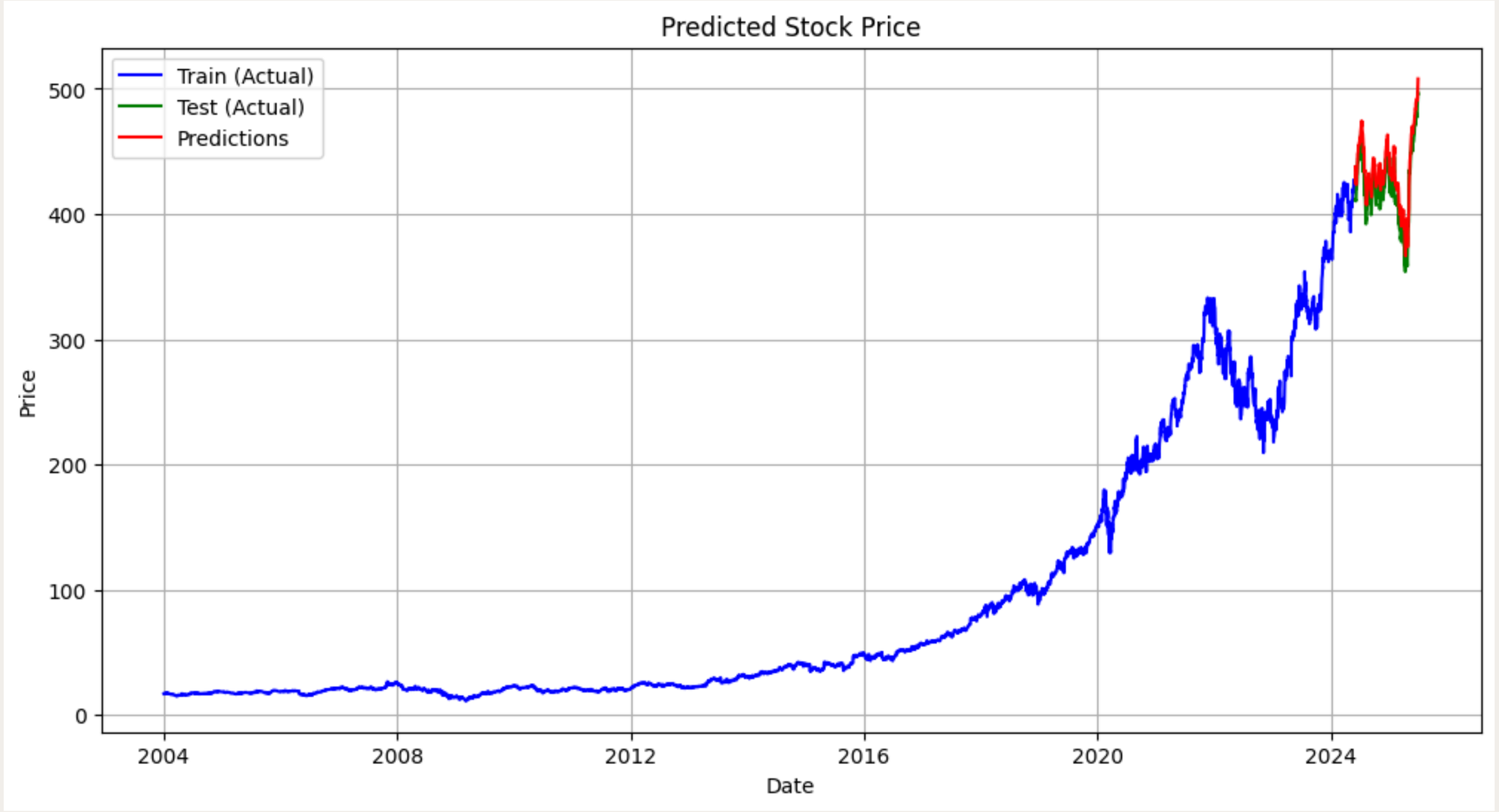
Data Source

We Used the Yahoo Finance Library (yfinance) on Python for getting of daily Microsoft Co. stock prices and volume data from 01-01-04 to 30-06-25



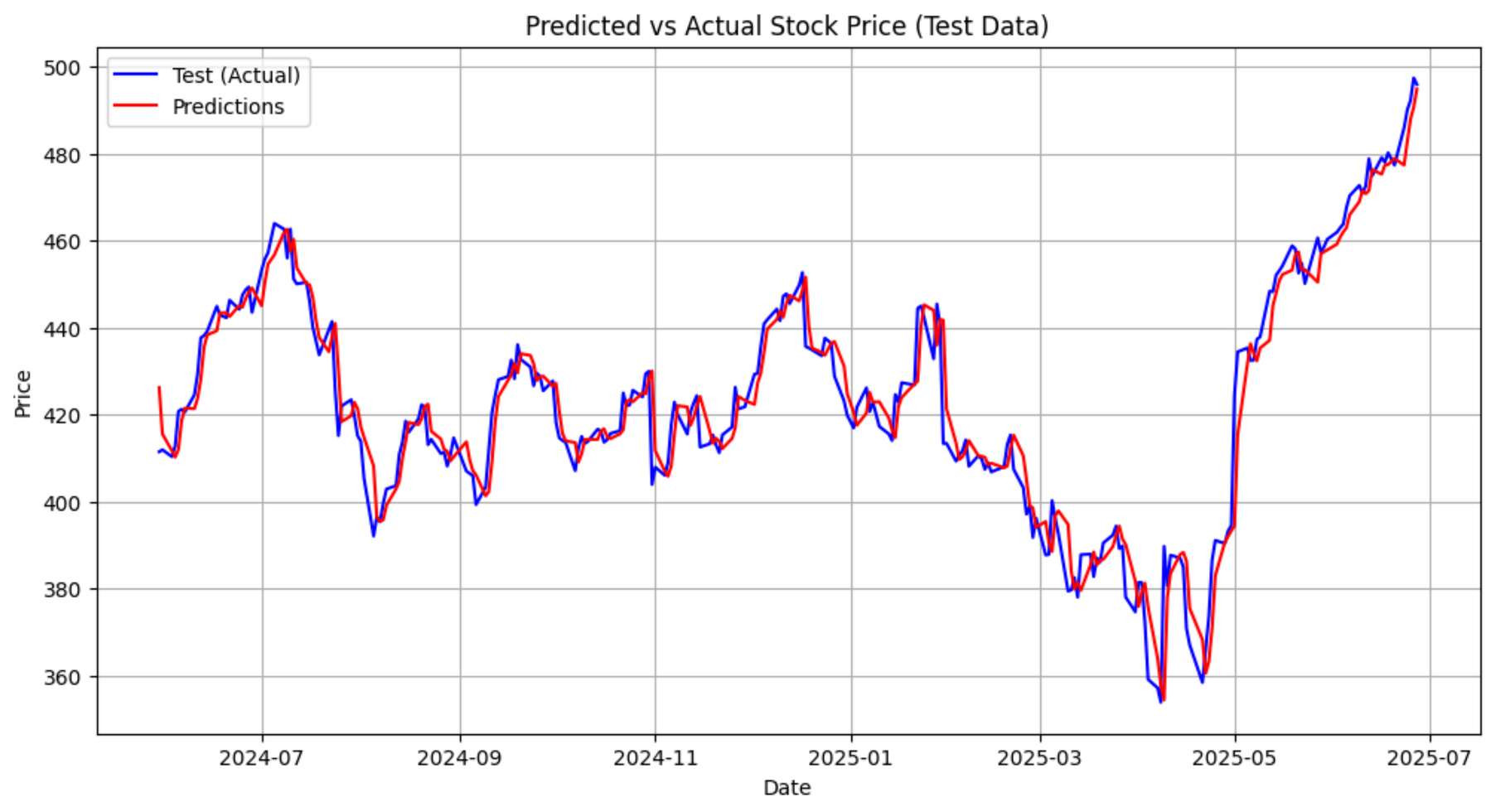
Graphical representation of data

PREDICTIONS USING THE BASIC 4 LAYERED LSTM MODEL



Graphical representation of data

PREDICTIONS USING LSTM MODEL



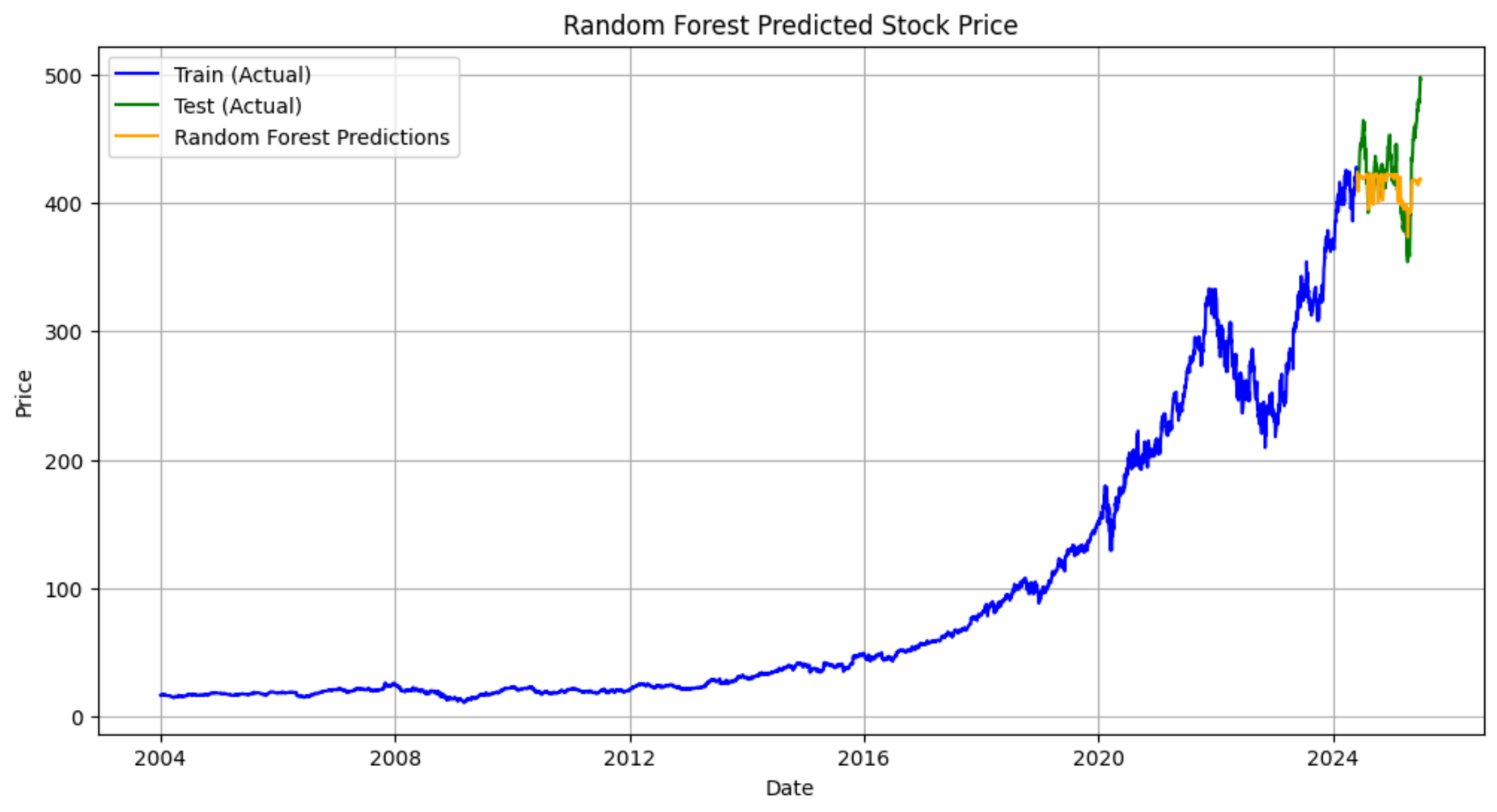
MEAN ABSOLUTE PERCENTAGE ERROR (MAPE): **0.01%**

R^2 SCORE: **0.9364**

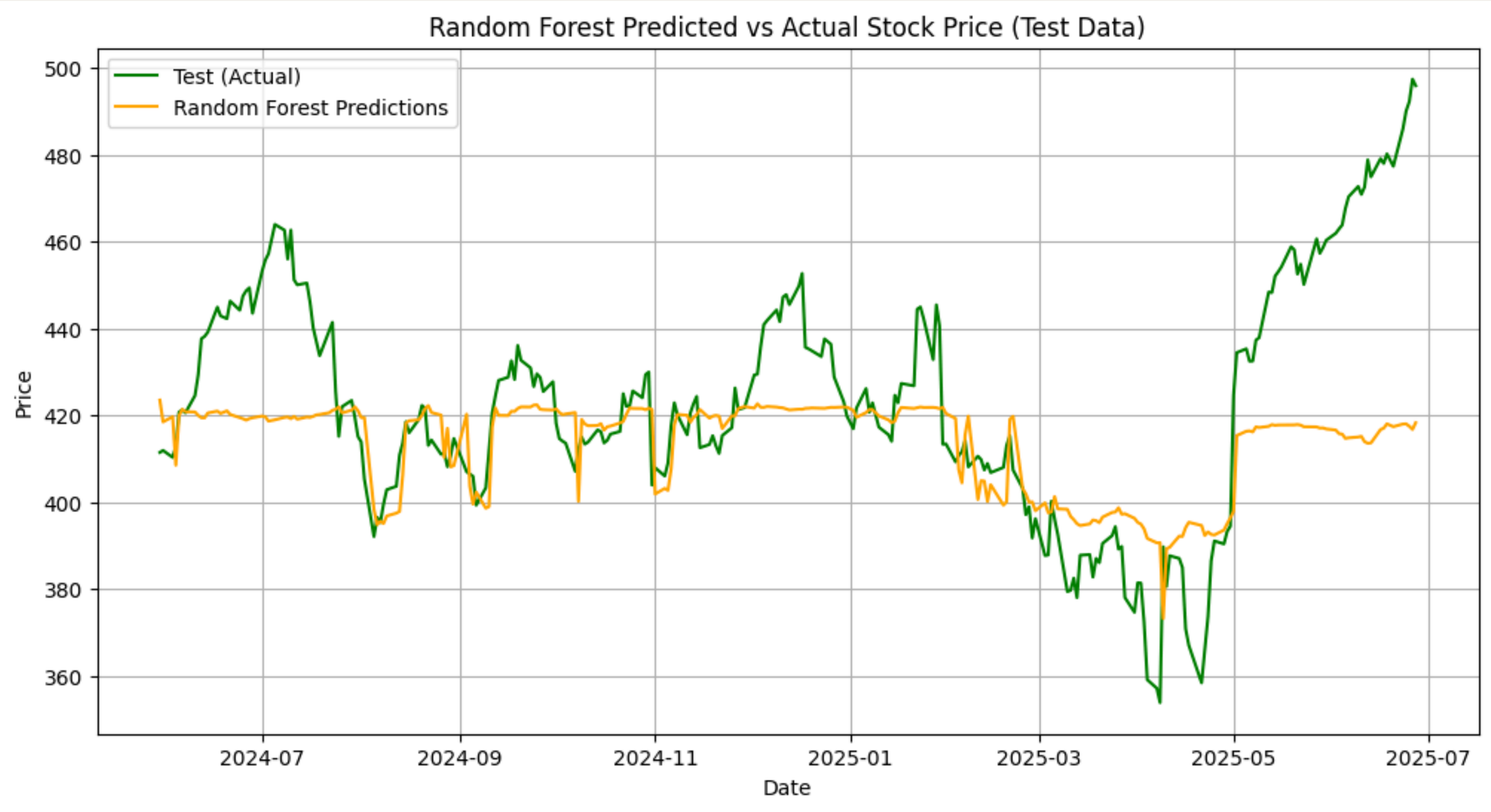
LSTM ROOT MEAN SQUARED ERROR (RMSE): **6.751001**

Graphical representation of data

PREDICTIONS AFTER USING RANDOM FOREST REGRESSION



Graphical representation of data

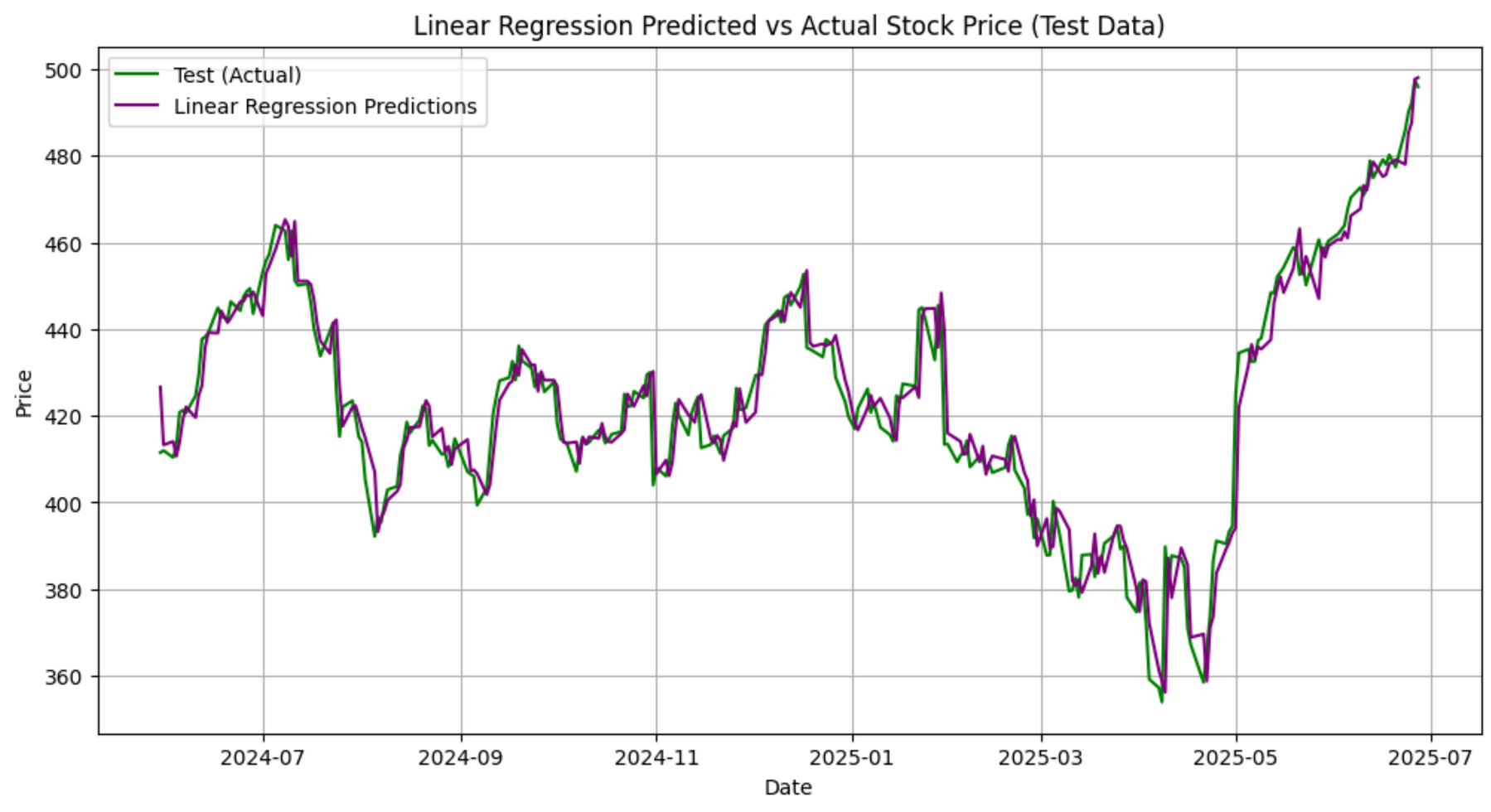


MEAN ABSOLUTE PERCENTAGE ERROR (MAPE): **3.66%**

LSTM ROOT MEAN SQUARED ERROR (RMSE): **23.23**

Graphical representation of data

PREDICTIONS AFTER INCORPORATING LINEAR REGRESSION



MEAN ABSOLUTE PERCENTAGE ERROR (MAPE): **0.01%**

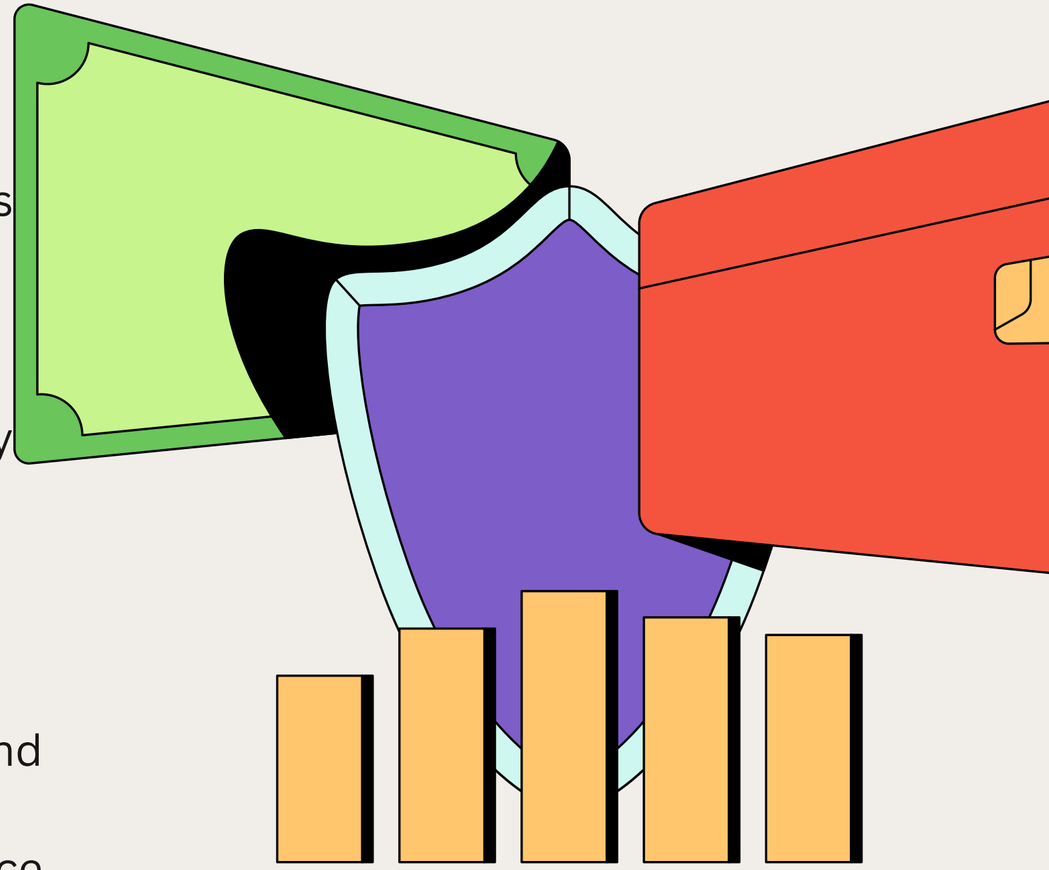
R^2 SCORE: **0.9381**

LSTM ROOT MEAN SQUARED ERROR (RMSE): **6.66**

Empirical Findings

Model comparision

- Linear regression showed superior performance as well as computation.
- LSTM performed reasonably but was computationally expensive.
- Random Forest had decent performance but was outperformed by Linear Regression.



Visual insights

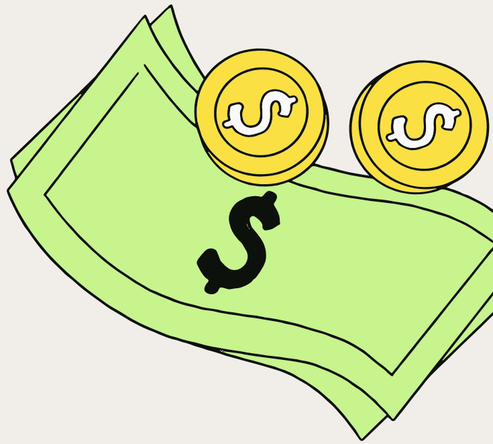
- Time-series plots of opening price insicate clear trends and seasonality.
- Log - returns distribution highlights volatality clustering in price movements.

Conclusions

- 🔍 Key Takeaways:
- Linear Regression is the most effective model for predicting stock returns in this context—simple yet accurate.
- Random Forest offers flexibility but does not outperform the baseline linear model.
- LSTM captures temporal dependencies, but its computational cost outweighs marginal gains in accuracy.

📈 Practical Implications:

- For real-time or low-latency prediction tasks, linear models are preferable.
- Deep learning models like LSTM may be better suited for high-frequency trading or long-term forecasting, where patterns are more complex.



Limitations

Data limitations

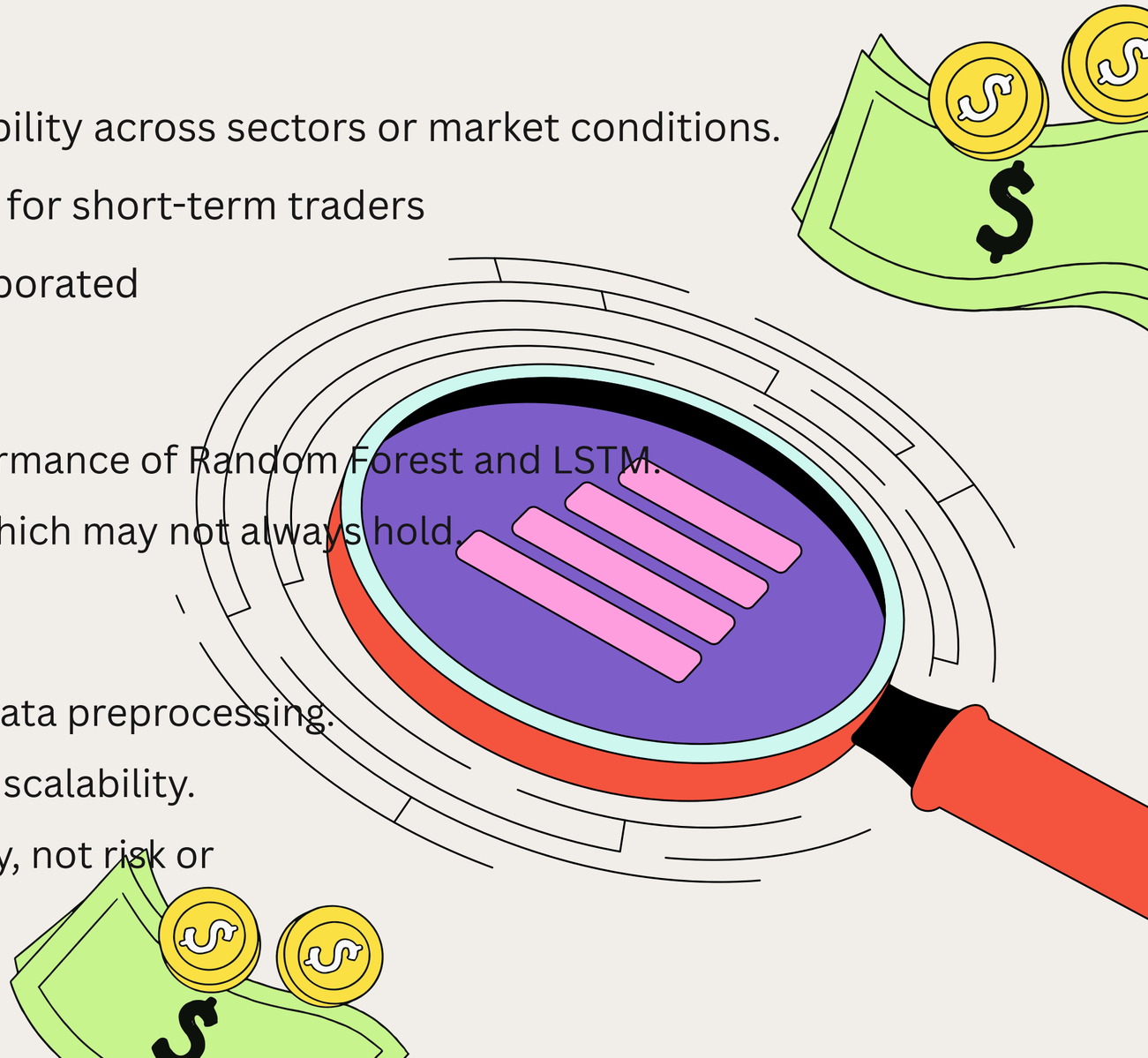
- Only one stock (MSFT) considered — limits generalizability across sectors or market conditions.
- Daily frequency ignores intraday dynamics important for short-term traders
- External macroeconomic or sentiment data not incorporated

Model limitations

- No hyperparameter tuning applied — could affect performance of Random Forest and LSTM.
- Assumes stationarity in returns for Linear Regression, which may not always hold.

Technical

- LSTM overfitting risk due to limited input features and data preprocessing.
- LSTM training was computationally expensive, affecting scalability.
- Evaluation metrics focused on point prediction accuracy, not risk or uncertainty.



Resources

Github link :https://github.com/premsanthosh23/Stock-Market-Prediction


THANK
you