Stock Price Prediction on NASDAQ Data: A Comparison of Linear Regression and Deep Neural Networks

Adam Johnson
University of Houston
Houston, TX, USA
adam.johnson@example.edu

ABSTRACT

This report investigates the effectiveness of two machine-learning approaches — linear regression and a two-layer deep neural network — for forecasting future stock returns on NASDAQ equities. Using fifteen-day and sixty-day historical windows, we train the models to predict five- and twenty-day returns, respectively. We benchmark predictive accuracy (RMSE, MAPE, R^2, directional accuracy) and simulated trading performance against a passive buy-and-hold baseline and discuss the impact of under- and over-fitting on real-world profitability.

ACM Reference Format:
Adam Johnson. 2025. Stock Price Prediction on NASDAQ Data: A Comparison of Linear Regression and Deep Neural Networks. In . ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 GROUP MEMBERS AND INDIVIDUAL CONTRIBUTIONS

I completed every part of the project individually. No additional group members contributed.

2 INTRODUCTION AND PROBLEM DESCRIPTION

Accurately forecasting stock prices is notoriously difficult because market behaviour is driven by a tangled mix of sentiment, macro-economic factors, breaking news and human psychology. Yet investors, analysts and trading algorithms all seek models that can reliably anticipate price movements.

Our objective is to apply machine-learning (ML) techniques to predict **future closing prices** (expressed as returns) for NASDAQ-listed equities. Historical price data — retrieved with the `yfinance` package — provide input features (*Open, High, Low, Close, Volume*) over rolling windows of length 15 or 60 days. Corresponding target returns are measured 5 or 20 days ahead. Supplementary metadata (`symbols_valid_meta.csv`) supply company names, exchange details and market categories that may prove useful for future feature engineering.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 LITERATURE REVIEW

Support Vector Machines [1]: Huang *et al.* formulated the prediction task as a binary classification ("up" vs "down") and fed engineered indicators such as moving averages into an SVM. Their work showed that even simple ML classifiers can outperform naïve baselines when supplied with informative features.

Decision Trees and Random Forests [2]: Patel *et al.* compared individual decision trees with ensemble random forests. The ensemble's ability to average many diverse trees yielded more stable forecasts in volatile regimes.

Long Short-Term Memory (LSTM) Networks [3]: Recognising that price series possess temporal dependencies, Hiransha *et al.* employed LSTMs to learn long-range patterns directly from raw sequences. Their deep-learning model achieved lower error than linear benchmarks in multiple markets.

These studies highlight the trade-offs between interpretability, non-linearity and sequence modelling that motivate our own comparison of linear regression and a compact deep neural network (DNN).

4 MACHINE LEARNING MODELS, METHODS, OR ALGORITHMS

4.1 Model Definitions

Linear Regression fits a single set of weights to map the scaled feature vector at time t to the return at $t+\Delta$. The model is fast, transparent and provides a strong baseline but is limited to linear relationships.

Deep Neural Network (DNN) comprises two fully-connected hidden layers (sizes 64 and 32) with ReLU activations. We train with the Adam optimiser, batch size 256 and a maximum of 500 epochs. Although expressive, the network risks memorising noise in small training sets.

4.2 Data & Training Setup

- **Input windows:** 15 or 60 consecutive trading days of (Open, High, Low, Close, Volume), standardised to zero mean and unit variance.
- **Forecast horizons:** 5-day or 20-day future returns.
- **Split:** 70% training, 15% validation, 15% test (chronologically ordered to prevent look-ahead bias).

4.3 Evaluation Metrics

- Root Mean Squared Error (RMSE)
- Mean Absolute Percentage Error (MAPE)
- Coefficient of Determination (R^2)

- Directional Accuracy (percentage of correctly predicted up/down moves)
- Simulated trading outcomes starting from $10,000:
  - *Buy-and-Hold*: passive benchmark.
  - *Model-Driven*: each day invests (predicted return)×$10,000 long (positive) or short (negative) for the forecast horizon.

5 EXPERIMENT RESULTS

5.1 Linear Regression

15-day / 5-day horizon

- Train — RMSE 0.0267, MAPE 233.5%, R^2 −0.112, DirAcc 53%
- Val — RMSE 0.0201, MAPE 273.3%, R^2 −0.233, DirAcc 48%
- Test — RMSE 0.0263, MAPE 330.6%, R^2 −0.366, DirAcc 43%
- Cash — Buy-and-Hold $12,401; Model-Driven $9,863

60-day / 20-day horizon

- Train — RMSE 0.0451, MAPE 213.9%, R^2 0.071, DirAcc 62%
- Val — RMSE 0.0361, MAPE 237.8%, R^2 −0.280, DirAcc 47%
- Test — RMSE 0.0463, MAPE 183.4%, R^2 −0.137, DirAcc 45%
- Cash — Buy-and-Hold $16,268; Model-Driven $9,815

5.2 Deep Neural Network

15-day / 5-day horizon

- Train — RMSE 0.0293, MAPE 369.1%, R^2 −0.336, DirAcc 57%
- Val — RMSE 0.0600, MAPE 1,407%, R^2 −10.05, DirAcc 53%
- Test — RMSE 0.2031, MAPE 7,533%, R^2 −80.56, DirAcc 63%
- Cash — Buy-and-Hold $12,401; Model-Driven $10,963

60-day / 20-day horizon

- Train — RMSE 0.0440, MAPE 267.9%, R^2 0.116, DirAcc 65%
- Val — RMSE 0.0675, MAPE 755%, R^2 −3.49, DirAcc 51%
- Test — RMSE 0.1275, MAPE 1,133%, R^2 −7.63, DirAcc 69%
- Cash — Buy-and-Hold $16,268; Model-Driven $14,757

5.3 Analysis

Linear regression delivers the lowest numeric errors but underfits, producing trading returns that trail a passive benchmark. The DNN captures direction better (up to 69% accuracy) and, despite noisy point forecasts, generates higher trading profits. However, the widening gap between training and validation/test errors signals severe over-fitting.

6 CONCLUSION

Our study reinforces two lessons for financial ML. First, simple linear baselines are hard to beat on classical accuracy scores yet often fail to translate that accuracy into profit. Second, powerful non-linear models may uncover richer signals, but careful regularisation (dropout, early stopping) and additional data are essential to curb over-fitting.

Future work will explore richer feature sets (news sentiment, technical indicators) and hybrid ensembles that blend a regularised linear model for position sizing with a well-regularised DNN for directional calls.

REFERENCES

[1] W. Huang, Y. Nakamori, and S. Wang. 2005. Forecasting Stock Market Movement Direction with Support Vector Machines. *Computers & Operations Research* 32, 10 (2005), 2513–2522.

[2] J. Patel, S. Shah, P. Thakkar, and K. Kotecha. 2015. Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. *Expert Systems with Applications* 42, 1 (2015), 259–268.

[3] M. Hiransha, E. Gopalakrishnan, V. Menon, and K. Soman. 2018. NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science* 132 (2018), 1351–1362.