Short-Term Stock Price Direction Prediction Using Machine Learning Models

Hang Sun

Faculty of Science, The University of Sydney, Sydney, NSW 2006, Australia
sunh19491001@gmail.com

Abstract. This study looks at a simple question in stock prediction. Can basic machine learning models give useful signals about whether a stock return will go up or down in the next period? The experiment does not try to predict the exact stock price. It uses relative returns, because returns are easier to compare across stocks and across time than absolute prices. The dataset contains five large technology stocks, and the features are built from lagged returns, moving averages and recent volatility. Four classifiers are tested under the same time-based training and testing setup, including logistic regression, random forest, gradient boosting and K-nearest neighbours. Random forest gives the best test accuracy in this experiment, reaching 0.630. Gradient boosting reaches 0.580, K-nearest neighbours reach 0.550 and logistic regression reaches 0.540. These results are slightly better than random guessing, but the improvement is still limited. The main message is that simple return-based features can provide some information, while short-term stock direction prediction remains difficult and would need richer data in future work.

Keywords: Stock direction prediction; machine learning; return-based features; random forest; financial forecasting

1 Introduction

Stock market prediction is a useful but difficult problem in financial analytics. If a model can give even a small improvement in predicting the direction of returns, it may help with trading signals, portfolio decisions or risk control. At the same time, financial data are noisy and change across market conditions. Prices react not only to past prices but also to news, investor expectations and broader economic factors. Recent review papers show that machine learning has been widely used for stock forecasting, but the reported results depend heavily on the market, features, time horizon and validation design [1-3]. This makes stock direction prediction a suitable task for testing a clear machine learning workflow.

Another reason for choosing this topic is that a high prediction accuracy does not automatically mean that a model is practically useful. A financial model should define its target clearly, avoid using future information, and be evaluated on data that were not used for training [4]. For that reason, this study keeps the task narrow. Instead of predicting the future stock price itself, it predicts whether the next-period return is

© The Author(s) 2026
L. C. Yoong (ed.), *Proceedings of the 2026 2nd International Conference on Financial Innovation and Marketing Management (FIMM 2026)*, Advances in Economics, Business and Management Research 408,
https://doi.org/10.2991/978-94-6239-801-6_59

positive or not. This setup is close to a very simple trading decision, because a positive signal may suggest taking exposure, while a negative signal may suggest avoiding or reducing exposure.

The experiment uses returns rather than absolute prices. Absolute prices can move to very different levels over time because of long-term growth, corporate actions and changes in market regimes. They are also not easy to com-pare across different stocks. Returns are more suitable for this project because they describe relative price changes. Therefore, the numerical inputs are based on past returns, rolling average returns and rolling volatility.

This study makes three small but clear contributions. It first sets up short-term stock direction prediction as a bi-nary classification problem based on next-period returns. It then builds a simple return-based feature set for a multi-stock panel dataset. Finally, it compares logistic regression, random forest, gradient boosting and K-nearest neigh-bours, abbreviated as KNN after first use, under the same time-series evaluation design. The study does not propose a new algorithm. Its contribution is to provide a clean baseline experiment and to explain honestly what the results can and cannot show.

2 Related Work

Stock market prediction has been studied for many years because it is both practically important and statistically difficult. Review studies suggest that the choice of input variables, the construction of the dataset and the validation design can strongly affect the reported performance of stock forecasting models [1]. Many financial prediction stud-ies combine price history, technical indicators and supervised learning methods [2]. Recent systematic reviews also show that support vector machines, long short-term memory networks and artificial neural networks are common methods in this area [3]. A broader review of artificial intelligence and machine learning in finance further shows that financial prediction has become an active interdisciplinary topic [4].

Several empirical studies are especially relevant to this project. Stock movement direction is often treated as a binary prediction target, and support vector machines have been used for this kind of classification task [5]. Long short-term memory networks have also been applied to financial market prediction, showing the potential of deep learning for time-series data [6]. More recent work has tested convolutional neural networks and transformers for financial time-series forecasting, which suggests that newer models may capture both local and longer-term patterns [7]. These studies are useful for understanding possible extensions. In this paper, however, simpler baseline models are used because the dataset is small and the main aim is comparison rather than model complexity.

Classical machine learning models are still useful as baselines. Random forest is one important example because it combines many decision trees and can model nonlinear relationships while reducing the variance of a single tree [8]. Standard statistical learning texts also stress several points that are important for this project, including the difference between classification and regression, the need to compare several models, and the importance of test performance rather than training performance alone [9,10].

This study follows these ideas by applying the same feature set and the same time-ordered train-test split to all classifiers.

3 Methodology

3.1 Data and Target Construction

The experiment uses the Plotly Express built-in stocks dataset, which contains normalized stock price series for five large technology stocks, namely AAPL, AMZN, GOOG, MSFT and NFLX [11]. The original data are in wide format, so they are reshaped into a long panel format with one observation for each ticker and date. After creating lag and rolling-window variables and removing missing values, the final modelling dataset contains 500 observations.

Let r denote the relative return of stock i at time t. It is calculated from normalized prices as follows.

$$r_{i,t} = (P_{i,t} / P_{i,t-1}) - 1 \quad (1)$$

The target variable is defined according to the sign of the next-period return.

$$y_{i,t} = 1 \; if \; r_{i,t+1} > 0, and \; y_{i,t} = 0 \; otherwise \quad (2)$$

The target variable has only two possible values. A value of one means that the next-period return is positive, while a value of zero means that the next-period return is zero or negative. For this reason, the task is treated as a classification problem rather than a regression problem.

3.2 Feature Engineering

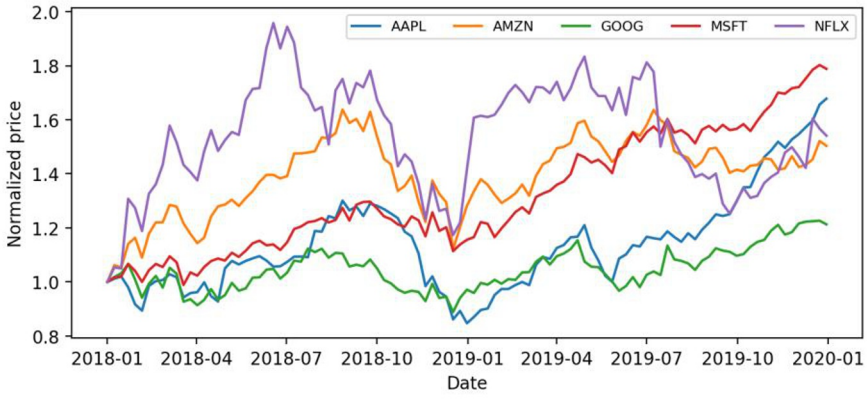


Fig. 1. Normalized price series for selected technology stocks (Picture credit: Original).

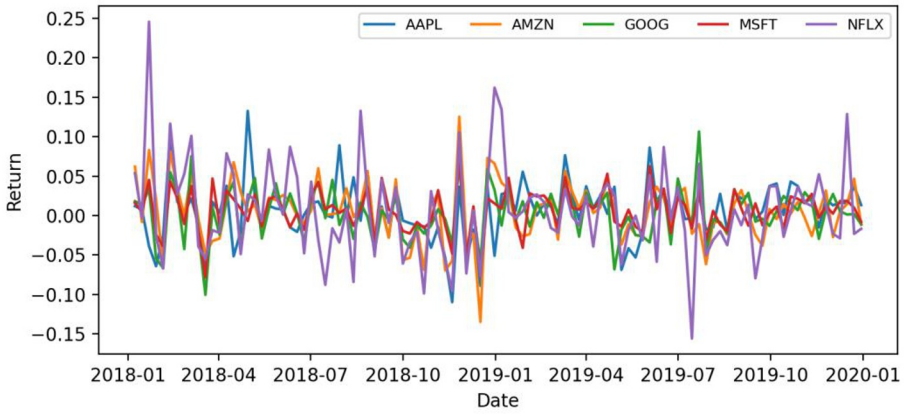


Fig. 2. Relative returns used for modelling (Picture credit: Original).

The input variables are constructed from past returns rather than from absolute prices. The main features are four lagged returns, a four-period moving average of returns and a four-period rolling volatility measure. A ticker indicator is also added so that the models can distinguish between different stocks. These features are simple by design. The aim is to build an interpretable baseline experiment, not a highly optimized trading system. As shown in Fig. 1, the normalized price paths differ across stocks. Fig. 2 shows the return series, which fluctuate around zero and are more comparable across time.

3.3 Models

Four supervised learning models are used in the experiment. Logistic regression is included as a simple and interpretable baseline for binary classification. Random forest and gradient boosting are included because tree-based ensemble methods can capture nonlinear patterns. KNN is included as a distance-based comparison method. Before fitting the models, numerical variables are standardized, and ticker names are converted into one-hot encoded variables.

3.4 Evaluation Design

The dataset is ordered by date and then split into two parts. The earlier 80 percent of observations are used for training, and the later 20 percent are used for testing. This is more appropriate for time-series data than a random split, because it avoids using later observations to predict earlier ones. Time-series cross-validation is also applied within the training period to estimate validation accuracy.

The evaluation uses several metrics. Training accuracy shows how well the model fits the data it has already seen. Test accuracy is more important because it measures performance on later observations that were not used for training. Cross-validation accuracy gives another validation result within the training data. Precision is the share of predicted positive cases that are actually positive, while recall is the share of actual positive cases that are correctly identified. The F1-score combines precision and recall.

The area under the receiver operating characteristic curve, abbreviated as AUC after first use, measures how well the model ranks positive cases above negative cases.

4 Results

Table 1 reports the model results. Random forest gives the strongest test accuracy, reaching 0.630. Gradient boosting reaches 0.580, KNN reaches 0.550 and logistic regression reaches 0.540. The results are better than, or close to, a simple random-guessing benchmark. However, the improvement is not large, which is consistent with the fact that short-term stock returns are hard to predict.

Table 1. Model comparison results.

Model	Training accuracy	CV accuracy	Test accuracy	Precision	Recall	F1-score	AUC
Logistic Regression	0.565	0.464	0.540	0.667	0.647	0.657	0.473
Random Forest	0.762	0.503	0.630	0.687	0.838	0.755	0.509
Gradient Boosting	0.958	0.470	0.580	0.697	0.676	0.687	0.524
KNN	0.638	0.448	0.550	0.658	0.706	0.681	0.496

The gap between training accuracy and test accuracy also needs attention. Gradient boosting reaches a training ac-curacy of 0.958, but its test accuracy is much lower. This suggests that the model may be fitting noise in the training data. Random forest also has higher training accuracy than test accuracy, but it still gives the best out-of-sample result among the four models. This supports using ensemble tree models as useful baselines, while also showing why validation is necessary.

The AUC values are all close to 0.5 in this experiment. This means that the probability ranking is weak even though the classification accuracy is slightly above random guessing. One possible reason is that the test set is relatively small and the features only contain simple historical return information. Therefore, the results should be understood as a baseline learning experiment, not as evidence that the model can produce a profitable trading strategy.

5 Conclusion

This study tested whether simple machine learning models can predict the short-term direction of stock returns. The target variable is the sign of the next-period return, so

the problem is set up as classification. The experiment also follows the financial modelling idea of using returns instead of absolute prices.

Among the four models, random forest performs best in terms of test accuracy. Even so, the overall predictive power is limited. This result is reasonable because stock returns are noisy and simple historical return features may not contain enough information for reliable prediction. The main value of the study is therefore methodological. It shows how to define the target, build return-based features, compare classifiers and evaluate out-of-sample results in a cautious way.

There are still clear limitations. The dataset is small and comes from normalized sample data rather than a long daily market dataset. The features are also basic and do not include macro variables, firm fundamentals, news sentiment or transaction costs. The experiment evaluates prediction accuracy, but it does not test a full trading strategy. Future work could use Yahoo Finance daily data, add market index and volatility index variables, test support vector machines and long short-term memory networks, and check whether the prediction signals still help after trading costs are considered.

References

1. Kumbure, M.M., Lohrmann, C., Luukka, P., Porras, J.: Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications 197, 116659 (2022). https://doi.org/10.1016/j.eswa.2022.116659
2. Henrique, B.M., Sobreiro, V.A., Kimura, H.: Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications 124, 226-251 (2019). https://doi.org/10.1016/j.eswa.2019.01.012
3. Lin, C.Y., Marques, J.A.L.: Stock market prediction using artificial intelligence: A systematic review of systematic reviews. Social Sciences & Humanities Open 9, 100864 (2024). https://doi.org/10.1016/j.ssaho.2024.100864
4. Ahmed, S., Alshater, M.M., Ammari, A.E., Hammami, H.: Artificial intelligence and machine learning in finance: A bibliometric review. Research in International Business and Finance 61, 101646 (2022). https://doi.org/10.1016/j.ribaf.2022.101646
5. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Computers & Operations Research 32(10), 2513-2522 (2005). https://doi.org/10.1016/j.cor.2004.03.016
6. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270(2), 654-669 (2018). https://doi.org/10.1016/j.ejor.2017.11.054
7. Zeng, Z., Kaur, R., Siddagangappa, S., Rahimi, S., Balch, T., Veloso, M.: Financial time series forecasting using CNN and transformer. arXiv preprint arXiv:2304.04912 (2023)
8. Breiman, L.: Random forests. Machine Learning 45, 5-32 (2001). https://doi.org/10.1023/A:1010933404324
9. James, G., Witten, D., Hastie, T., Tibshirani, R.: An introduction to statistical learning with applications in R. Springer, New York (2021)
10. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: Data mining, inference, and prediction. Springer, New York (2009)
11. Plotly Technologies Inc.: Plotly Express built-in stocks dataset (2024)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

