Advances in Engineering Innovation Vol.17 Issue 4 EWA Publishing
Available Online: 26 March 2026 DOI: 10.54254/2977-3903/2026.32419

Machine learning approaches to stock return prediction: evidence from the JPX Tokyo stock exchange challenge

Diyi An

Brighton College, Brighton, UK

deanan276@gmail.com

Abstract. Predicting stock returns remains a central challenge in quantitative finance due to the noisy, non-stationary, and highly cross-sectional nature of financial markets. This paper investigates the use of machine learning methods for equity return prediction using data from the Japan Exchange Group (JPX) Tokyo Stock Exchange Prediction Challenge. The dataset comprises daily price, volume, trading, and financial statement information for approximately 2,000 Japanese equities over the period 2017–2021, with performance evaluated through a ranking-based portfolio construction framework. We develop a modelling pipeline that emphasises data integrity, parsimonious feature engineering, and economically meaningful evaluation. Following minimal preprocessing focused on alignment and deduplication, features are constructed from adjusted prices, lagged returns, rolling window statistics, trading volume measures, and daily cross-sectional ranks. A set of linear regression models is employed as interpretable baselines, against which a gradient boosting decision tree model implemented via LightGBM is benchmarked. Model performance is assessed using both pointwise prediction diagnostics and a portfolio-based spread metric consistent with the competition objective, which measures the risk-adjusted return of a daily long–short strategy formed from predicted rankings. Results indicate that linear models struggle to capture the non-linear and interaction-driven structure of equity returns, while gradient boosting provides materially stronger performance in terms of portfolio level outcomes. Overall, the findings highlight the importance of cross-sectional evaluation and demonstrate that relatively simple, carefully constructed features combined with flexible tree-based models can deliver meaningful improvements in stock ranking tasks.

Keywords: machine learning, kaggle, stock prediction, JPX Tokyo stock, deep learning

1. Introduction

Financial markets play a large role in the global economy, influencing investment decisions, capital allocation, and long-term economic growth [1]. Success in these markets often depends on the ability to identify under over-valued assets and to quickly act upon this information. Historically, such decisions were made manually by traders and analysts, but in recent years the increased availability of financial data, advances in computational power, and the integration of Machine Learning (ML) techniques have enabled the growth of quantitative trading strategies [2]. Despite advancements, financial prediction remains a difficult challenge due to the inherent volatility and non-stationarity of market data. Financial markets are characterized by high levels of noise—random fluctuations driven by factors such as economic events, investor sentiment, and market microstructure. This noise often obscures genuine signals, reducing the reliability of predictive models. As a result, even when exploitable patterns exist, they may be unstable or fragile under changing conditions [3]. Consequently, even when exploitable patterns exist, they are often overshadowed by this noise, leading to less reliable and riskier predictions [4]. ML offers promising avenues for financial prediction by capturing complex, non-linear relationships in data. However, their effectiveness is dependent on careful model evaluation and validation. Models must be robust to overfitting and capable of providing interpretable results to ensure their practical applicability in real-world financial decision-making. The JPX Tokyo Stock Exchange Prediction Competition on Kaggle [5] provides a unique opportunity to explore these challenges. Hosted by the Japan Exchange Group (JPX), which is the fifth-largest stock exchange operator [6] and operates the Tokyo Stock Exchange (TSE), Osaka Exchange (OSE), and Tokyo Commodity Exchange (TOCOM) [7]. The TSE alone lists 4,000 companies [8] and serves as a critical hub for both domestic and international investors, making it a valuable case study for quantitative modelling. The competition makes available detailed financial data for approximately 2,000 stocks and tasks participants with ranking these stocks by predicted returns. Performance is evaluated using the Sharpe ratio of daily spread returns, where the top 200 ranked stocks are assumed to be purchased and the bottom 200 shorted. This portfolio-based evaluation framework captures how a model's predictions translate into practical trading outcomes, emphasising risk-adjusted returns rather than raw predictive accuracy. Such a setup closely resembles real-world quantitative investment problems, where portfolio performance is more relevant than the accuracy of individual stock predictions. In this project, a range of ML approaches is systematically examined for the purpose of stock return prediction based on the JPX dataset. The analysis commences with exploratory data analysis and the implementation of baseline models, and subsequently extends to the application of more advanced methodologies. The aim is to benchmark predictive performance but also identify which financial variables contribute most to the model's predictions.

2. Related works

This section provides a review of the key literature on stock return forecasting, with particular emphasis on traditional econometric models, ML approaches, and deep neural network methods.

2.1. Traditional econometric and time series models

Autoregressive Integrated Moving Average (ARIMA) remains a foundational tool in financial time-series forecasting, used to capture linear dependencies in data [9]. It tends to perform well for short-range forecasting, especially when the data exhibits stability in mean, variance, and autocorrelation structure. If not, practitioners commonly apply differencing, which involves subtracting consecutive observations to make a non-stationary time series stationary, or run unit-root tests such as the Augmented Dickey–Fuller (ADF) [10] or KPSS [11, 12], which assess whether the series can be considered stationary after adjustments. Within stock price forecasting contexts, ARIMA reliably captures trend and seasonality components over short windows [13], especially when data are not excessively volatile [14, 15]. However, financial markets often display non-linear behaviors and structural breaks, which ARIMA models may fail to capture, limiting their predictive accuracy in such contexts [15, 16].

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) family of models, introduced by Bollerslev in 1986, addresses a different challenge of financial data: time-varying volatility. GARCH models assume volatility evolves over time based on past squared returns and previous volatility, thereby capturing volatility clustering—the tendency for large changes to cluster in time [17]. These models are widely used in risk management, derivative pricing, and financial econometrics because of their ability to model changing variance rather than assuming constant volatility. During times of market stress or turbulence, symmetric variants such as EGARCH often yield more accurate volatility forecasts than symmetric GARCH [18] or constant-volatility models. These extended models better capture large negative shocks and changing variance, making their volatility forecasts more accurate under extreme conditions [17, 19].

2.2. Machine learning models

Over the past decade, ML methods have gained increasing attention in financial time-series forecasting, as they are able to model non-linear relationships and adapt to changing market conditions. Studies have shown how ML (and hybrid models that combine ML with traditional methods) have improved upon classical econometric approaches in financial forecasting [11, 16].

For example, methods such as Random Forests and Gradient Boosting Machines have shown strong predictive accuracy in stock return forecasting, particularly when dealing with high-dimensional

feature spaces [20, 21]. Support Vector Machines (SVM) have also been widely applied, using their ability to construct non-linear decision boundaries in classification and regression tasks [22].

In addition, hybridisation with traditional econometric models has been investigated as a way to combine the strengths of both linear and non-linear approaches. For example, a recent study systematically integrates econometric methods such as ARIMA and ARFIMA with ML algorithms, including SVM, Extreme Gradient Boosting (XGBoost), and Deep Learning (DL) models such as Long Short-term Memory Networks (LSTM), showing that such hybrids can outperform individual components and baseline benchmarks across diverse datasets [23].

However, although ML models offer better accuracy in many cases, their performance depends on data quality, proper feature engineering, forecasting horizon, and generalisation to unseen market conditions [24]. Key challenges include the risk of overfitting—particularly when the feature space is high-dimensional relative to the sample size—along with sensitivity to hyperparameter choices, vulnerability to performance degradation under regime shifts, and limited interpretability compared to simpler models.

2.3. Deep learning and neural networks

DL methods have surged in popularity for financial time-series forecasting, largely because they can model long-range dependencies, handle many interacting features, and exploit heterogeneous data sources. Despite this power, these models come with tradeoffs: they require larger, cleaner datasets, careful tuning, and often sacrifice interpretability in exchange for improved performance.

For example, in 2019, Siami-Namini et al. [11] demonstrated that a Bidirectional LSTM (BiLSTM) model outperforms both a standard LSTM and ARIMA on multiple financial series, especially when longer forecasting horizons are used [11]. In 2024, a comparative analysis of LSTM, Gated Recurrent Unit (GRU), and Transformer architectures for stock price prediction shows that Transformer models are particularly effective at capturing long-range dependencies and complex temporal interactions, often outperforming recurrent approaches such as LSTM and GRU [25]. More recent work further suggests that hybrid and attention-enhanced architectures—including GRU, Convolutional Neural Networks (CNN) combined with LSTM (CNN-LSTM), and Transformers—tend to outperform conventional recurrent models, especially under volatile market conditions [26].

Altogether, the DL literature shows that architectures such as LSTM, GRU, Transformer, BiLSTM, and hybrid approaches can often deliver significant improvements over simpler methods; however, these gains remain conditioned on factors such as regime shifts, feature engineering, data size and quality, model complexity, and the capacity for generalisation.

3. Methodology

This section presents the methodological framework for developing and evaluating predictive models of Japanese equity returns. It begins with a description of the dataset obtained from the JPX Tokyo Stock Exchange Prediction competition, followed by exploratory data analysis to examine its statistical properties and market structure. The subsequent part details the data preprocessing procedures

applied to construct a clean and balanced panel suitable for modelling. Feature engineering techniques are then introduced, aimed at extracting economically meaningful and statistically informative signals. Finally, the modelling framework is outlined, encompassing both baseline approaches and modern ML methods that together form the foundation for the empirical evaluation.

Dataset

The dataset used in this study is sourced from the JPX Tokyo Stock Exchange Prediction competition on Kaggle [5]. It covers the Japanese equity market from 2017-01-04 to 2021-12-03 and includes multiple files containing market, trading, and financial information. The core file stock_prices.csv contains 2,332,531 observations across 12 columns, representing 2,000 unique equities over 1,202 trading days. The constituent files, including the primary file, are summarised below:

stock_prices.csv: daily stock-level information, including open, high, low, and close prices, trading volume, adjustment factors, and the competition target variable.

secondary_stock_prices.csv: price data for less actively traded securities outside the main set of 2,000 equities; not scored but useful for contextualising broader market dynamics.

trades.csv: aggregated weekly trading volume features, serving as proxies for market liquidity and activity.

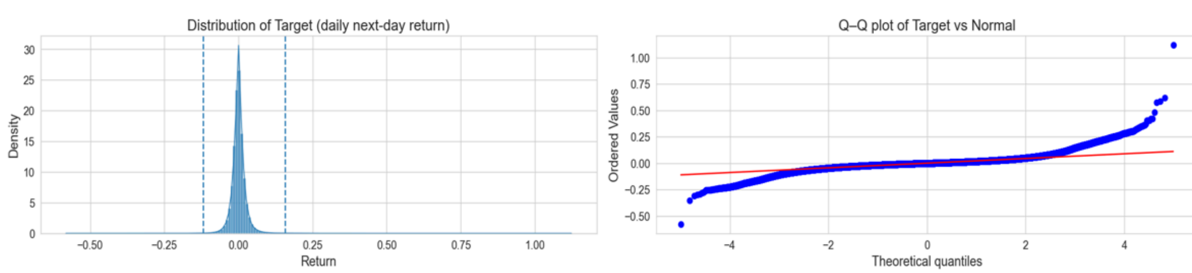
financials.csv: quarterly company fundamentals such as earnings and profits, reflecting aspects of financial health beyond price movements.

stock_list.csv: metadata linking stock codes to company names, sectors, and industry classifications.

options.csv: derivative market data, not directly scored, but informative about forward-looking market expectations.

The prediction target is defined as the daily return obtained by purchasing a stock at the close on day $t + 1$ and selling at the close on day $t + 2$:

$$C(k, t+2) - C(k, t+1) \tag{1}$$



(a) Distribution of returns

(b) Q–Q plot vs normal

Figure 1. Distributional properties of the target variable (target)

Distribution of Returns. Figure 1 shows the empirical distribution of the target variable (Target), centred close to zero with a sample mean of $9.5 \times 10{-}4$. The Q–Q plot against the normal distribution (Figure 1) highlights clear departures in the tails, confirming excess kurtosis. Extreme returns occur with non-negligible probability, with the 0.1% and 99.9% quantiles at approximately -0.09 and +0.09, respectively. These heavy tails are consistent with well-documented empirical properties of asset returns [3].

Time-series Patterns. The daily average of returns across equities (Figure 2) fluctuates around zero, with occasional spikes in either direction. This reflects the aggregate nature of stock-level variation and highlights episodes of heightened market stress where returns move sharply away from the mean.

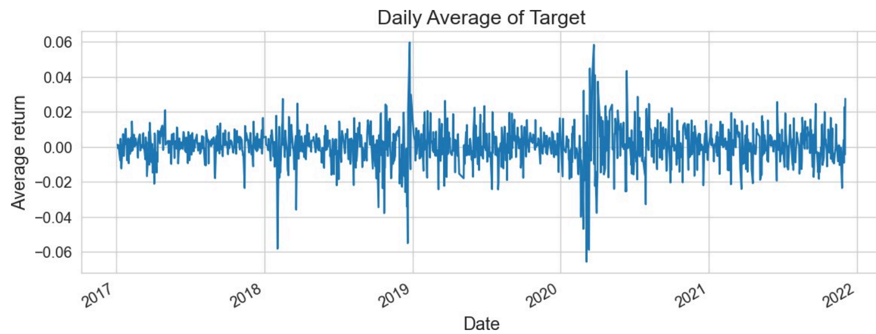


Figure 2. Daily average of returns across equities

Cross-sectional Dispersion. The daily cross-sectional standard deviation of returns (Figure 3) reveals strong time variation, withwhere $C(k, t{+}2)$ denotes the closing price of stock k on day $t + 2$.To better understand the statistical properties of the JPX datasetand guide subsequent preprocessing and model development, exploratory data analysis is conducted. Table 1 reports summary statistics for key stock price features.

$$r(k,t) = C(k,t+1) \tag{2}$$

Table 1. Summary statistics for stock price features

Feature	Mean	Std	Min	25%	75%
Open	2,594.5	3,577.2	14	1,022	3,030
Close	2,594.0	3,576.5	14	1,022	3,030
Volume	691,937	3,911,256	0	30,300	402,100
Target	0.00095	0.0715	-0.90	-0.0105	0.0110

The following analyses key statistical properties of the dataset. spikes during periods of market turbulence.

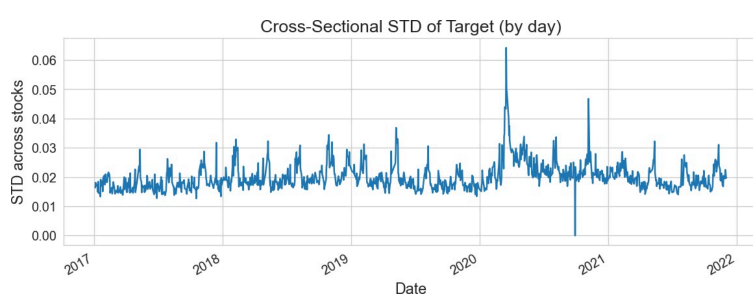


Figure 3. Cross-sectional standard deviation of daily returns

Sectoral Heterogeneity. By linking equities to their industry classification via stock_list.csv, we observe marked differences in return distributions across sectors (Figure 4). Cyclical industries such as IT services and machinery display wider boxes and longer tails, whereas defensive sectors such as foods show narrower distributions.

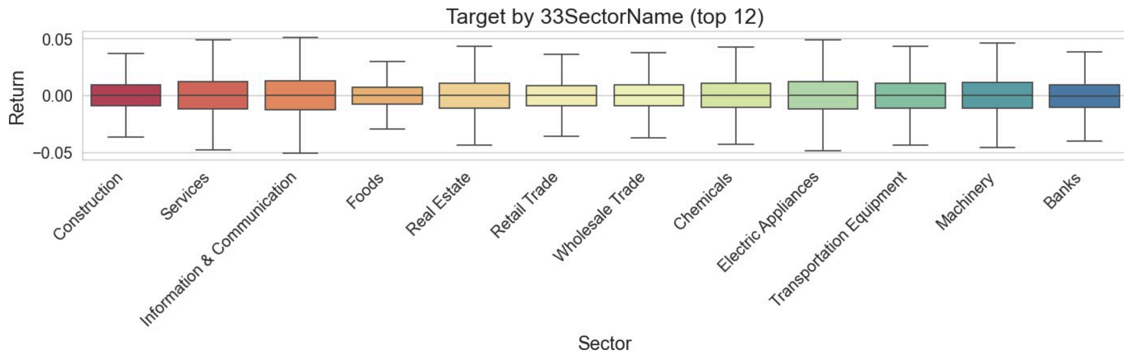


Figure 4. Distribution of returns across selected sectors

Feature Correlations. Price features (open, high, low, close) exhibit near-perfect correlation (Figure 5), confirming they provide largely redundant information when included simultaneously. In contrast, trading volume and adjustment factors are less correlated, offering complementary signals for prediction.

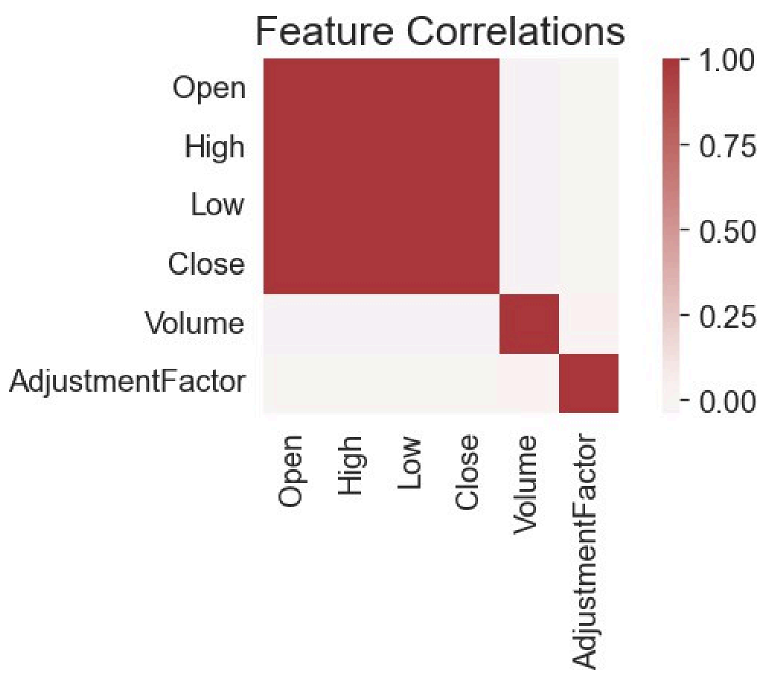


Figure 5. Correlation heatmap of price and volume features

Market Overview. To contextualise stock-level patterns, Figure 6 plots the time-series of market-wide averages: returns, closing prices, and trading volume. The market index shows gradual upward drift punctuated by sharp downturns, notably around early 2020, consistent with the COVID-19 shock. Trading activity also spikes during such turbulent periods, highlighting the link between volatility and liquidity.

Figure 6. JPX market average stock return, closing price, and shares traded

Yearly Sector Returns. Average returns vary substantially across sectors and years (Figure 7). Energy resources and transportationrelated industries often outperform in expansionary periods, while defensive sectors such as foods and pharmaceuticals show resilience during downturns. These temporal sectoral dynamics motivate incorporating year and sector interaction terms into the feature set.

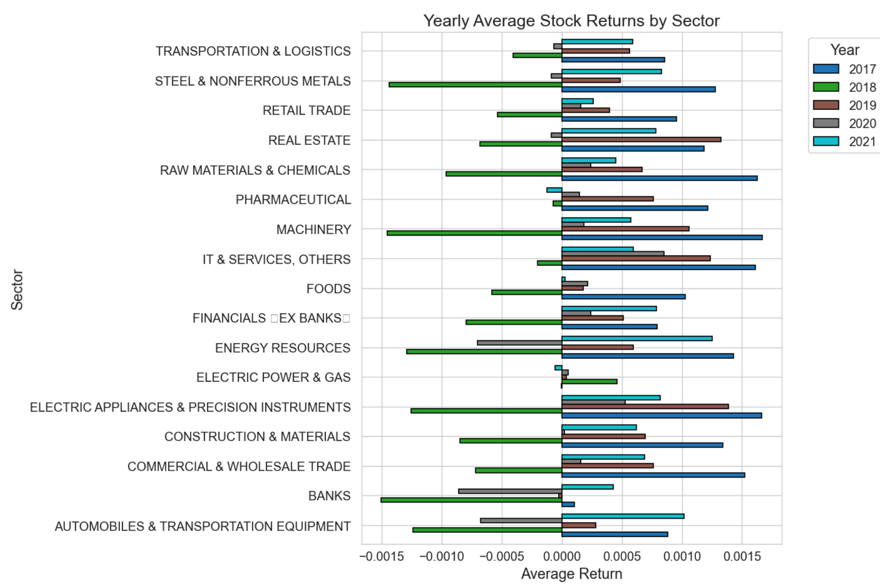


Figure 7. Yearly average stock returns by sector

Target Distribution by Sector. Figure 8 shows the distribution of returns grouped by sector. While most sectors cluster within a narrow band around zero, cyclicals such as real estate and construction display wider dispersions and more extreme outliers. This reinforces the view that industry classification contains predictive information about risk and return heterogeneity.

Figure 8. Distribution of returns across sectors

Sector Correlation Structure. Returns at the sector level exhibit strong co-movement, though the intensity varies across industries (Figure 9). For instance, cyclicals such as transportation and construction are highly correlated, while defensive or regulated sectors (e.g., utilities, pharmaceuticals) move more independently.

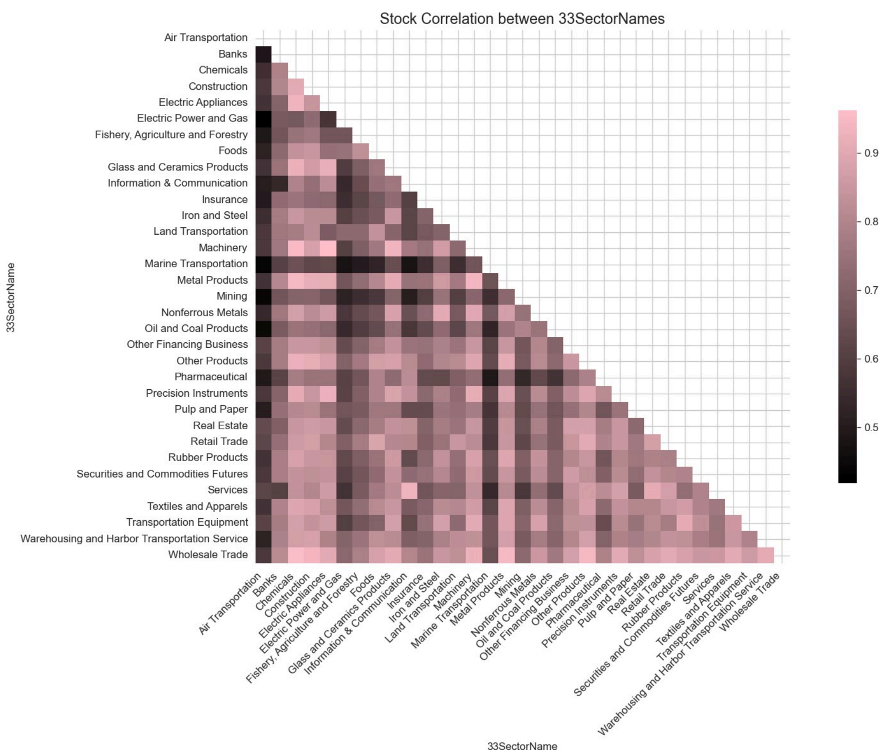


Figure 9. Correlation matrix of average sector-level returns

4. Data preprocessing

Prior to feature engineering, a set of light but essential preprocessing steps is applied to clean, align, and validate the raw datasets. The objective at this stage is not to transform variables, but to ensure internal consistency across data sources and to define a stable equity universe for subsequent modelling.

Data Sources and Initial Validation. The analysis draws on five core tables: daily stock prices, secondary stock prices, trading activity, firm financial statements, and a static stock list containing sector metadata. Each dataset is inspected for basic integrity, including date ranges, identifier consistency, duplicate observations, and missing values. Summary diagnostics confirm that the primary stock price dataset contains exactly 2,000 unique securities observed over 1,202 trading days, with no duplicate (Date, SecuritiesCode) pairs.

Identifier Cleaning and Universe Definition. Observations with missing SecuritiesCode identifiers are removed, resulting in the exclusion of two rows across the combined datasets. The core equity universe is then defined using the 2,000 securities present in the primary stock price table. This universe is used as the reference set for all subsequent merges to ensure a consistent cross-sectional dimension.

Financial Statement Deduplication. The financial statements dataset contains multiple records per firm-date due to repeated disclosures and revisions. To address this, observations are deduplicated at the (Date, SecuritiesCode) level, retaining a single record per firm per reporting date. This step reduces the financials table from its raw size to 45,062 observations and eliminates all remaining identifier-level duplicates.

Handling of Missing Values. At this stage, no global imputation strategy is applied. Missing values are preserved in their raw form to avoid introducing noise or implicit assumptions prior to feature construction. Columns with near-complete missingness are identified during exploratory analysis and deferred for removal or transformation during feature engineering, where their relevance can be assessed more appropriately.

Dataset Alignment. All datasets are temporally aligned using their date fields and restricted to the defined core universe of securities. Static information from the stock list is merged on SecuritiesCode, while time-varying tables are aligned on (Date, SecuritiesCode) keys. This produces a clean, internally consistent set of inputs suitable for downstream feature engineering and modelling.

Overall, the preprocessing performed at this stage is intentionally minimal, focusing on data integrity and structural consistency rather than statistical transformation. More involved operations, including feature construction, transformations, and normalization, are introduced in the subsequent feature engineering stage.

Following data cleaning and alignment, a set of derived features is constructed from prices, volumes, and financial statement variables. The feature engineering process is deliberately parsimonious, focusing on transformations that are both economically interpretable and empirically effective in cross-sectional return prediction.

Adjusted Price Construction. Raw closing prices are adjusted for corporate actions using the adjustment factors provided by the competition host. This produces an adjusted price series that is consistent across stock splits and reverse splits, ensuring that computed returns and rolling statistics reflect genuine price movements rather than mechanical discontinuities.

Return-Based Features. Daily returns are computed from adjusted prices and form the primary building block for subsequent features. To capture short-term dynamics, lagged returns over multiple horizons are included. These features allow the model to exploit shortrange autocorrelation and reversal effects commonly observed in equity returns.

Rolling Window Statistics. Rolling summary statistics of returns are computed over several window lengths, including rolling means and rolling standard deviations. These features capture local trends and time-varying volatility, allowing the model to adapt to changing market conditions. All rolling features are constructed using only past information to preserve causal validity.

Volume-Based Features. Trading volume is incorporated through rolling averages and transformations that capture persistent differences in trading activity across stocks. These features provide information about liquidity and investor attention without introducing high-frequency microstructure assumptions.

Cross-Sectional Normalization. To emphasize relative rather than absolute performance, several features are transformed into daily cross-sectional ranks across the equity universe. This ranking procedure reduces sensitivity to scale differences across stocks and aligns the feature representation with the cross-sectional nature of the prediction task.

Overall, the engineered feature set consists of return-, volatility-, and volume-based transformations that are simple, interpretable, and robust. More complex feature construction techniques are intentionally avoided in order to limit overfitting and maintain transparency in the modelling pipeline.

Together, these engineered features complement the raw inputs, enriching the dataset with both domain knowledge and statistical structure that are expected to enhance predictive performance.

Model Selection and Implementation

This study adopts a comparative modelling approach, using a small set of baseline models to contextualize the performance of a more expressive machine learning model. The objective is not to exhaustively survey all possible model classes, but to establish a clear performance hierarchy between simple benchmarks and a state-of-the-art non-linear method.

Baseline Models. As a point of reference, several linear regression

Model Configuration and Training. Model hyperparameters are selected using a combination of standard defaults and limited manual tuning, informed by prior empirical results and computational constraints. Extensive automated hyperparameter optimization is intentionally avoided to reduce the risk of overfitting and to maintain reproducibility.

All models are implemented using standard Python machine learning libraries, including scikit-learn for linear baselines and LightGBM for gradient boosting. Data manipulation and feature construction are performed using pandas and NumPy.

Computational Environment. Experiments are conducted in a Python-based environment on consumer-grade hardware. Model training and evaluation are performed on CPU, without reliance on specialized accelerators. This setup reflects a realistic research and prototyping environment and demonstrates that the proposed modelling pipeline is computationally tractable without bespoke infrastructure.

Overall, this modelling strategy balances interpretability, predictive performance, and practical feasibility, allowing meaningful comparisons between simple econometric baselines and a more expressive non-linear model.

5. Experiments

5.1. Evaluation metrics

Introduce metrics used (e.g., RMSE, MAE, R2, directional accuracy, Sharpe ratio if finance-specific);Why they are appropriate for the problem

To evaluate model performance, the project utilized a combination of error-based and portfolio-based metrics aligned with the competition objective to assess model performance.

Pointwise Prediction metrics. The following metrics were computed on the predicted vs true returns:

RMSE (Root Mean Squared Error):

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \tag{3}$$

captures the average magnitude of prediction error.

RAE (Relative Absolute Error):

$$\frac{\sum_{i=1}^{N}|(y_i-\hat{y}_i)|}{\sum_{i=1}^{N}|(y_i-\bar{y}_i)|} \tag{4}$$

measures absolute error relative to a naïve mean forecast.

Directional Accuracy (Hit Ratio)

$$\frac{1}{N}\sum_{i=1}^{N} 1(sign(y_i) = sign(\hat{y}_i) \tag{5}$$

measures how often the model correctly predicts the direction (positive or negative) of returns.

Portfolio Spread Construction. On each trading day t , we perform the following steps:

Rank all stocks by predicted return $r\hat{}(k, t)$.

Select the top 200 stocks (highest predictions) for the long leg, and the bottom 200 for the short leg.

Assign linear weights wi to each of those 200 positions: the highest rank receives weight 2.0, the lowest receives weight 1.0, and intermediate weights are spaced linearly. Let w denote the average weight.

Compute the weighted return for the long leg:

$$S_{up,t} = \frac{1}{\bar{w}}\sum_{i=1}^{200} w_i r(up_i, t) \tag{6}$$

and similarly for the short leg,

$$S_{down,t} = \frac{1}{\bar{w}}\sum_{i=1}^{200} w_i r(down_i, t) \tag{7}$$

Define the daily spread return:

$$R_{day,t} = S_{up,t} - S_{down,t} \tag{8}$$

Over T days, the competition score is:

$$Score = \frac{E[R_{day}]}{Std(R_{day})} \tag{9}$$

i.e. the mean of the daily spread return series divided by its standard deviation (i.e. annualized Sharpe without subtracting risk-free rate.

5.2. Validation strategy

A time-aware validation strategy is adopted to ensure that model evaluation reflects realistic forecasting conditions and avoids lookahead bias. Given the temporal structure of financial data and the cross-sectional nature of the prediction task, standard random cross-validation is not appropriate and is therefore avoided.

Train-Test Split. The dataset is split chronologically into training and evaluation periods. Models are trained exclusively on historical data and evaluated on a strictly later hold-out window. All feature construction follow this ordering, with rolling statistics and lagged variables computed using only information available prior to the prediction date.

This forward-chaining approach mirrors real-world deployment, where models are calibrated on past observations and applied to unseen future data. No information from the evaluation period is used during model fitting or feature construction.

Cross-Sectional Structure. Within each trading day, predictions are generated simultaneously for the full cross-section of equities in the defined universe. Evaluation is therefore conducted on a day-by-day basis across stocks, rather than treating observations as independent time-series samples. This structure is consistent with the competition objective, which assesses the quality of daily stock rankings rather than individual point forecasts.

Avoidance of Data Leakage. Several safeguards are implemented to prevent data leakage. First, all rolling and lagged features are computed using backward-looking windows only. Second, financial statement data are aligned by reporting date and merged without forward-filling future disclosures. Third, the train-test split is performed at the date level rather than the observation level, ensuring that no information from future trading days contaminates the training set.

As rolling-window cross-validation could be employed, a single, clearly defined out-of-sample evaluation window provides a transparent and conservative assessment of model performance. This choice aligns with both the structure of the JPX competition and the practical constraints faced in applied quantitative modelling.

6. Conclusion

This study examined the application of machine learning methods to cross-sectional stock return prediction using data from the JPX Tokyo Stock Exchange Prediction Challenge. Exploratory analysis confirmed key empirical properties of equity markets, including heavy-tailed returns, time-varying dispersion, and sectoral heterogeneity, motivating a modelling approach focused on relative, cross-sectional signals.

A comparative framework showed that linear models provide limited predictive power, while a gradient boosting decision tree model (LightGBM) achieved materially stronger performance, particularly when evaluated using a portfolio-based spread metric aligned with real trading objectives. These results indicate that non-linear tree-based models are better suited to capturing the complex interactions present in equity returns.

Overall, the findings demonstrate that carefully engineered but relatively simple features, when combined with flexible machine learning models and portfolio-oriented evaluation, can deliver meaningful improvements in stock ranking tasks. Future work may extend this approach by incorporating richer data sources, alternative model classes, and explicit trading constraints.

References

[1] Khan, M. S., & Abdelhak, S. S. (2000). *Financial Developmentand Economic Growth: An Overview*. International Monetary Fund, Working Paper.

[2] Alameer, A., Saleh, H., & Alshehri, K. (2022). *Reinforcement learning in quantitative trading: A survey.* Authorea Preprints.

[3] Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance, 1*(2), 223.

[4] Nabar, O., & Shroff, G. (2023, November). Conservative predictions on noisy financial data. In *Proceedings of the Fourth ACM International Conference on AI in Finance* (pp. 427-435).

[5] Kaggle. (2022). *JPX Tokyo Stock Exchange prediction*. https: //www.kaggle.com/competitions/jpx-tokyo-stock-exchange-prediction

[6] Capital.com. (n.d.). *The largest stock exchanges in the world.* https: //capital.com/en-int/markets/shares/largest-stock-exchanges

[7] Japan Exchange Group. (n.d.). *Japan Exchange Group*. https: //www.jpx.co.jp/english

[8] Japan Exchange Group. (n.d.). *Number of listed companies/shares.* https: //www.jpx.co.jp/english/listing/co/index.html

[9] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). *A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM*. arXiv preprint. https: //arxiv.org/abs/1911.09512

[10] Mushtaq, R. (2011). *Augmented Dickey–Fuller test.* SSRN. https: //doi.org/10.2139/ssrn.1911068

[11] Khan, M. S., & Senhadji, A. S. (2000). *Financial development and economic growth: An overview* (IMF Working Paper No. 00/209). International Monetary Fund. https://doi.org/10.5089/9781451874747.001

[12] Shin, Y., & Schmidt, P. (1992). The KPSS stationarity test as a unit root test. *Economics Letters, 38*(4), 387-392.

[13] Gonçalves, J. V. M., Alexandre, M., & Lima, G. T. (2023). *ARIMA and LSTM: A comparative analysis of financial time series forecasting* (Working Paper No. 2023_13). Department of Economics, University of São Paulo (FEA-USP).

[14] Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation* (pp. 106-112). IEEE.

[15] Stellwagen, E., & Tashman, L. (2013). ARIMA: The Models of Box and Jenkins. *Foresight: The International Journal of Applied Forecasting*, (30), 28-33.

[16] Alarbi, A., Khalifa, W., & Alzubi, A. (2025). A Hybrid AI Framework for Enhanced Stock Movement Prediction: Integrating ARIMA, RNN, and LightGBM Models. *Systems, 13*(3), 162.

[17] Schmidt, L. (2021). *Volatility forecasting performance of GARCH models: A study on Nordic indices during COVID-19* [Master's thesis, Umeå University]. Department of Economics, Umeå University.

[18] Ezzat, H. (2012). The application of GARCH and EGARCH in modeling the volatility of daily stock returns during massive shocks: The empirical case of Egypt. *International Research Journal of Finance and Economics, 96*, 143–154.

[19] Stempień, D., & Ślepaczuk, R. (2025). *Hybrid models for financial forecasting: Combining econometric, machine learning, and deep learning models*. arXiv preprint. https://arxiv.org/abs/2505.19617

[20] Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. B. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques. In *International Conference on Finance, Economics, Management and IT Business.* https://www.semanticscholar.org/paper/A-Survey-on-Machine-Learning-for-Stock-Price-and-Obthong-Tantisantiwong/7c4c6f3e9a2d8b5f1e4a7c9d6b8f2e3a5c7d9e1f

[21] Ryll, L., & Seidens, S. (2019). *Evaluating the performance of machine learning algorithms in financial market forecasting: A comprehensive survey.* arXiv preprint. https://arxiv.org/abs/1906.07786

[22] Jakkula, V. (2006). Tutorial on support vector machine (svm). *School of EECS, Washington State University, 37*(2.5), 3.

[23] Sjösten, L. (2022). *A Comparative Study of the KPSS and ADF Tests in terms of Size and Power.* https://hammockclusterstumble.com/e10qv7fj?key=54179c9d27afd2c4a6d90fb11c58910b

[24] Saberironaghi, M., Ren, J., & Saberironaghi, A. (2025). Stock Market Prediction Using Machine Learning and Deep Learning Techniques: A Review. *AppliedMath, 5*(3), 76.

[25] Sunki, A., SatyaKumar, C., Narayana, G. S., Koppera, V., & Hakeem, M. (2024). Time series forecasting of stock market using ARIMA, LSTM and FB prophet. In *MATEC Web of Conferences* (Vol. 392, p. 01163). EDP Sciences.

[26] Osman, E. G. A., & Otaibi, F. A. (2025). Integrating deep learning and econometrics for stock price prediction: A comprehensive comparison of lstm, transformers, and traditional time series models. *Machine Learning with Applications, 22*, 100730.