Machine Learning for Finance

Assignment 2

Nakarin Kungvalchokchai
ID:2400952

CF969-7-SP/ZU: Machine Learning for Finance
Dr Panagiotis Kanellopoulos

1.Introduction

The prediction of stock market movements is one of the most challenging problems in financial forecasting. Stock prices are influenced by numerous unpredictable factors, including economic indicators, investor sentiment, and geopolitical events, leading to high volatility (Zouaghia et al., 2023). Traditional stock market prediction methods, such as autoregressive models and moving averages, struggle to effectively capture the nonlinear and dynamic nature of stock prices, highlighting the need for more advanced techniques (Kapgate & Chaturvedi, 2025).

However, recent advancements in machine learning have paved the way for more accurate and adaptable prediction models. By leveraging machine learning's capacity to model complex, nonlinear relationships among variables, researchers can achieve more robust risk assessment and forecasting (Gu et al., 2020).

This study will explore the application of different supervised machine learning models to predict stock returns using historical price data and technical indicators. Focusing on five major U.S. companies: Apple (AAPL), Amazon (AMZN), JPMorgan Chase (JPM), Procter & Gamble (PG), and UnitedHealth Group (UNH). We will evaluate and compare the performance of four models: linear regression, support vector machines (SVM), random forests, and neural networks. Performance will be assessed using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) to determine which model delivers the most accurate forecasts across different assets.

2.Methodology

2.1 Data Collection and Preprocessing

- Stock Selection

Historical stock price data is collected for five major stocks, balancing growth and defensive stocks, over a 5-year period using the yfinance library. The stocks selected are:
AAPL – Apple Inc.: Technology
AMZN – Amazon.com, Inc.: E-commerce & Cloud Computing
JPM – JPMorgan Chase & Co.: Financials
PG – Procter & Gamble Co.: Consumer Staples
UNH – UnitedHealth Group Incorporated: Healthcare

- Technical Indicators and Feature Engineering

The following technical indicators are calculated and used as features in our models:

1. Simple Moving Averages (SMA): 10-day and 50-day moving averages to identify trends
2. Relative Strength Index (RSI): 14-day RSI to identify overbought or oversold conditions
3. Moving Average Convergence Divergence (MACD): Trend-following momentum indicator
4. Bollinger Bands: Volatility indicators showing price channels
5. Average True Range (ATR): Measure of market volatility
6. Market Returns: Daily returns of the S&P 500 to capture broader market movements

- Train-Test Split: Data is partitioned chronologically (80% training, 20% testing) to preserve temporal order.
- Standardisation: Features are scaled using StandardScaler to normalise values (mean=0, variance=1).

2.2 Model Implementation

2.2.1 Linear Regression

Linear Regression served as a baseline model, assuming a linear relationship between features and target:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$$

Where:

- y represents the predicted returns
- $x_1, x_2, ..., x_n$ are the input features (technical indicators)
- $\beta_0, \beta_1, ..., \beta_n$ are the coefficients to be estimated
- ϵ is the error term

Implementation uses scikit-learn's LinearRegression class with default parameters. Ordinary Least Squares is used for coefficient estimation, and statistical significance of coefficients is assessed using statsmodels' OLS implementation to obtain p-values.

2.2.2 Support Vector Machines

Support Vector Regression (SVR) is implemented using the following formulation:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

Where:

- α_i and α_i^* are Lagrange multipliers
- $K(x_i, x)$ is the kernel function mapping inputs to feature space
- b is the bias term

Three kernel functions are evaluated:

1. Linear: $K(x_i, x_j) = x_i^T x_j$
2. Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$
3. Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$

Hyperparameter tuning is performed using GridSearchCV with 5-fold cross-validation. The hyperparameter grid include:

- C: [0.01,0.1, 1, 10]
- epsilon: [0.001,0.01, 0.1]
- gamma: [0.0001,0.001, 0.01, 0.1] (for RBF and polynomial kernels)
- degree: [2, 3] (for polynomial kernel)

2.2.3 Random Forest

Random Forest regression creates an ensemble of decision trees, with each tree trained on a bootstrap sample of the training data and a random subset of features:

$$f(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$

Where:

- B is the number of trees
- $T_b(x)$ is the prediction of the b-th tree

Hyperparameters are tuned using GridSearchCV with the following grid:

- n_estimators: [100, 200]
- max_depth: [None, 10, 20]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4]
- max_features: ['sqrt', 'log2']

2.2.4 Neural Networks

A feedforward neural network is implemented using TensorFlow/Keras with the following architecture:

- Input layer: Matching the number of features
- Hidden layers: Three layers with 64, 32, and 16 neurons respectively
- Output layer: Single neuron for regression prediction
- Activation functions: ReLU for hidden layers, linear for output layer
- Regularisation: Dropout (0.2) between layers and L2 regularisation (0.001)

The model is compiled with the Adam optimiser and Mean Squared Error loss function: Early stopping with patience = 10 is implemented to prevent overfitting. The model is trained for a maximum of 100 epochs with a batch size of 32.

2.3 Time Series Cross-Validation

To assess model robustness and prevent data leakage, TimeSeriesSplit from scikit-learn is employed with 5 splits. This approach ensures that only past data is used to predict future values, respecting the temporal nature of financial time series.

2.4 Model Evaluation

We employ three metrics to assess model performance:

- Mean Absolute Error (MAE)

$\mathrm{MAE} = \frac{1}{n} \sum |y_{\mathrm{true}} - y_{\mathrm{predicted}}|$ with lower MAE indicating better accuracy.

- Root Mean Squared Error (RMSE)

$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum (y_{\mathrm{true}} - y_{\mathrm{predicted}})^2}$ RMSE penalises large errors more than MAE.

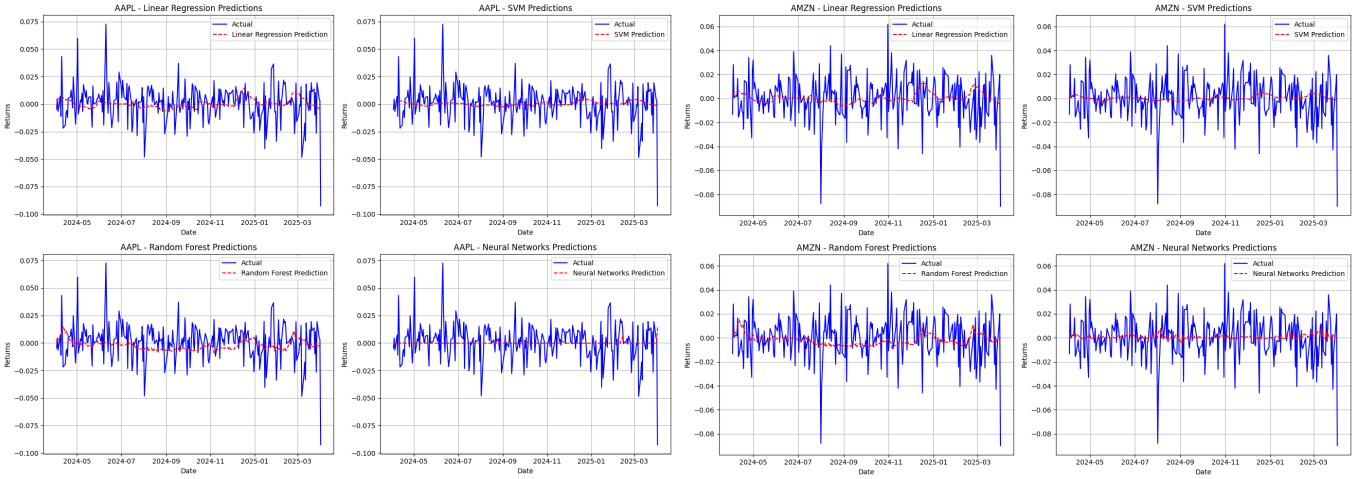
- Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

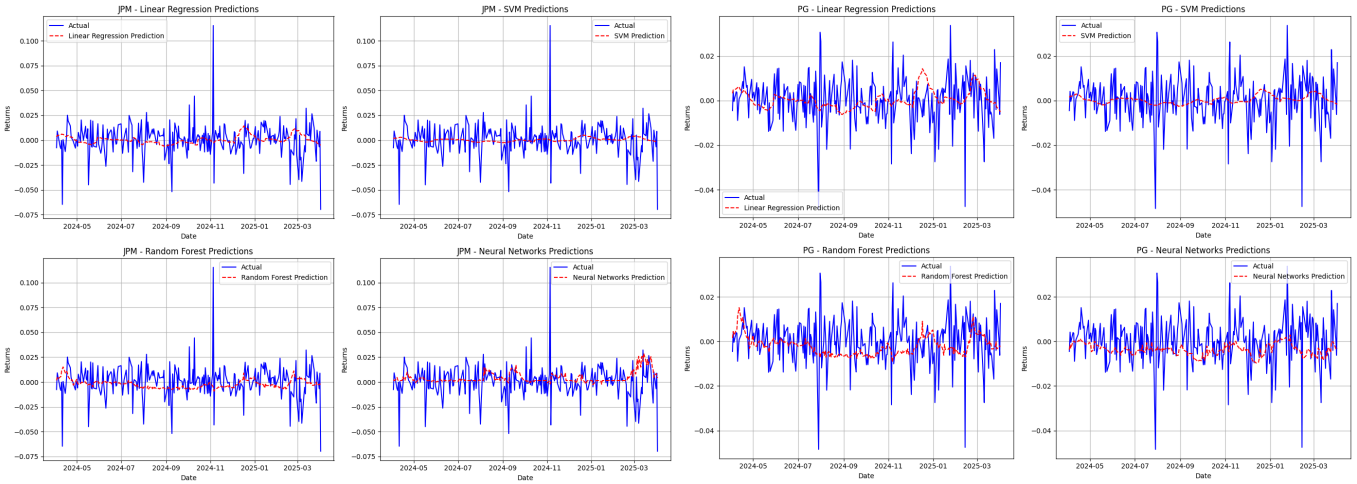
Represents proportion of variance explained where 1 indicates perfect fit.

3. Results and Analysis

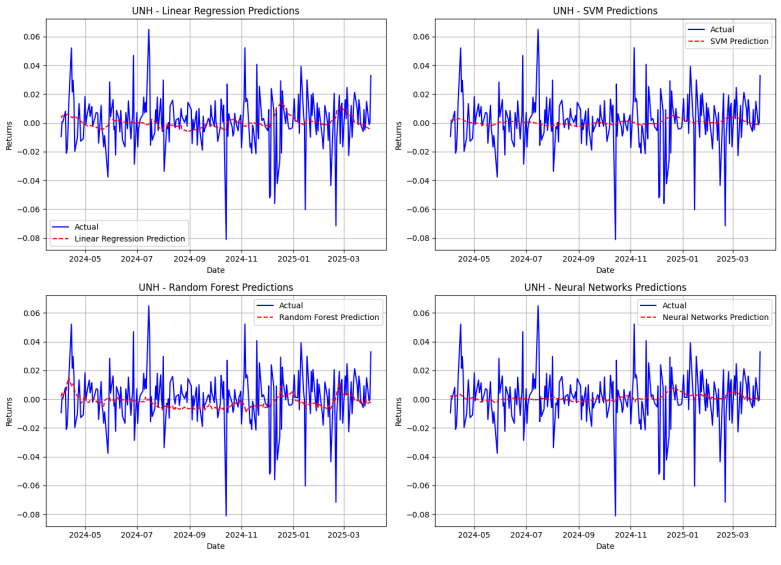
3.1 Stock Return Predictions Using Machine Learning Models



AAPL AMZN



JPM PG



UNH

Each chart compares actual returns (blue lines) to predicted returns (red dashed lines) from early 2024 to March 2025. All models have similar limits in predicting stock returns, which prediction lines seem considerably smoother than the highly volatile actual returns. While the models capture certain basic trends, they frequently fail to predict significant market spikes and drops, as evidenced by multiple significant outlier events in the data. Interestingly, the prediction patterns across the four different models look relatively similar for each stock, suggesting that model selection may not dramatically improve predictive performance in this context.

3.2 The cross-validation performance metrics

Stock	Model	MSE (Avg)	MAE (Avg)	R^2 (Avg)
AAPL	Linear	0.000342	0.014085	-0.234614
	SVM	**0.000302**	0.013165	**-0.059917**
	Random Forest	0.000311	0.013378	-0.131778
	Neuron Networks	0.000315	**0.013110**	-0.135362
AMZN	Linear	0.000530	0.016709	-0.078413
	SVM	**0.000498**	**0.016068**	**-0.005717**
	Random Forest	0.000595	0.018000	-0.227609
	Neuron Networks	0.000512	0.016458	-0.067407
JPM	Linear	0.000295	0.012841	-0.337661
	SVM	**0.000234**	**0.011051**	**-0.009707**
	Random Forest	0.000268	0.012342	-0.187865
	Neuron Networks	0.000252	0.011567	-0.087541
PG	Linear	**0.000121**	**0.008136**	**-0.062852**
	SVM	0.000147	0.008759	-0.188532
	Random Forest	0.000125	0.008390	-0.115549
	Neuron Networks	0.000153	0.009073	-0.299232
UNH	Linear	0.000289	0.012476	-0.486604
	SVM	**0.000221**	**0.010591**	**-0.003780**
	Random Forest	0.000236	0.011263	-0.091345
	Neuron Networks	0.000247	0.011299	-0.197128

The cross-validation results reveal performance variations among the four predictive models across different stocks. For AAPL, the Support Vector Machine (SVM) model achieves the lowest MSE (0.000302) along with the highest R^2 (-0.0599), indicating relatively better predictive capability than the others, although all models yield negative R^2 values, suggesting poor fit overall. Similarly, in AMZN, SVM outperforms the other models with lower MSE (0.000498), MAE (0.016068), and R^2 nearly to zero (-0.0057). In contrast, Random Forest has the lowest performance (MSE: 0.000595, MAE: 0.018000, R^2: -0.2276). In terms of JPM, SVM also outperforms with the lowest error metrics (MSE: 0.000234, MAE: 0.011051) and a nearly neutral R^2 value (-0.0097). For PG, linear regression showed slightly better metrics than other models, with the lowest MSE (0.000121) and MAE (0.008136) and a relatively higher R^2 (-0.0629), although differences were subtle. However, the neural network model performs the worst with a higher MSE (0.000153) and the lowest R^2 (-0.2992). Lastly, for UNH, the SVM model again delivered the best results with the lowest MSE (0.000221), MAE (0.010591), and R^2 (-0.0038), significantly outperforming linear regression (R^2: -0.4866) and neural networks (R^2: -0.1971).

Overall, SVM demonstrated the most consistent performance across all stocks, with the lowest error rates and R^2 values closest to zero. Although none of the models achieved a positive R^2, suggesting all struggled to generalise well to unseen data. This may point to the need for further tuning, feature engineering, or alternative modeling approaches to enhance prediction accuracy.

4. Discussion

4.1 Linear Regression

- P-values of Linear Regression Coefficients for Stock Return Prediction
  (Bold values indicate statistical significance at α=0.05)

Feature	AAPL	AMZN	JPM	PG	UNH
ATR	**0.007523**	0.517789	0.112132	0.576213	0.215182
BB_Lower	0.555366	0.080058	0.673766	**0.037615**	0.271691
BB_Upper	0.273214	0.513873	0.989023	0.317893	0.248568
MACD	0.350630	0.193894	0.533901	0.414613	0.537133
MarketReturns	**0.043681**	0.185186	0.296703	0.645754	0.482177
RSI_14	0.732234	0.260055	0.822236	0.553884	0.279260
SMA_10	0.649063	0.405847	0.786187	0.403843	0.827149
SMA_50	0.537571	0.326954	0.729730	0.516380	0.697127

The p-value analysis from the linear regression models reveals that only a few features exhibit statistically significant relationships with stock returns. For **AAPL**, the Average True Range (ATR) and MarketReturns are significant predictors, with p-values of 0.0075 and 0.0437 respectively, suggesting a strong relationship with AAPL's returns. For **PG**, BB_Lower (the lower Bollinger Band) is significant with a p-value of 0.0376, indicating it may play a predictive role in PG's returns. **AMZN** shows a slightly significant relationship with BB_Lower (p = 0.0801), which may require further investigation. In contrast, the remaining features across all stocks, including popular indicators such as MACD, RSI, and SMA, generally have high p-values, suggesting weak or no statistical significance in predicting stock returns within this model. Interestingly, **JPM** and **UNH** have no features below the 0.1 threshold, indicating no strong linear relationship between the selected features and their returns in this regression setup.

4.2 Support Vector Machines

- Optimal SVM Hyperparameters for Stock Return Prediction

Stock	C (Regularisation)	Gamma	Kernel	Epsilon	Degree
AAPL	0.1	0.01	poly	0.001	3
AMZN	0.01	0.01	rbf	0.01	–
JPM	0.01	0.001	poly	0.01	2
PG	1	0.01	poly	0.001	3
UNH	0.1	0.0001	rbf	0.01	–

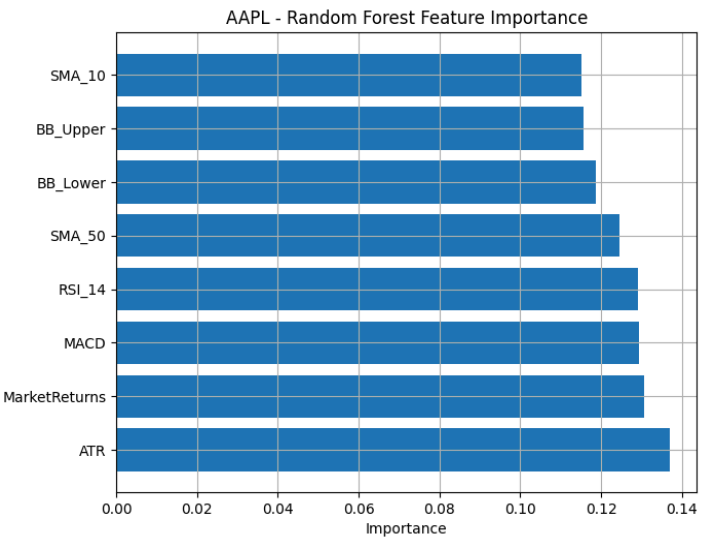
The hyperparameter tuning of the Support Vector Machines (SVM) models shows important differences in how each stock responded to the SVR configuration, which impacts the model's ability to fit the data and generalise effectively. When tuning the Support Vector Machine (SVM) models for different stocks, we noticed that each stock responds differently to the model settings. The kernel selection has a noticeable influence: **AAPL**, **JPM**, and **PG** perform best with a polynomial kernel, meaning their return trends follow smooth, curved patterns. In contrast, **AMZN** and **UNH** work better with the RBF kernel, which is more flexible and can handle more complex, irregular movements.

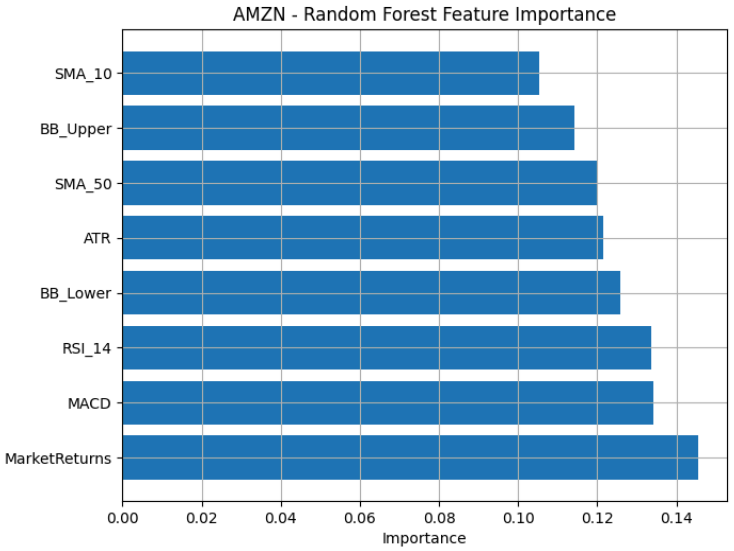
The regularisation parameter (C), which balances between fitting the data perfectly and avoiding overfitting, is set low (between 0.01 and 0.1) for most stocks, except **PG** (C=1). A small C helps prevent the model from memorising noise in the data, making it more reliable for unseen data. The relatively small C values across most models suggest a need to generalise better on unseen data rather than overfitting to training noise. Gamma, which determines how far each data point influences the model, is also kept small, even as low as 0.0001 for **UNH**. This means the model focuses on broader trends rather than noise, which is important for avoiding mistakes in stock predictions.

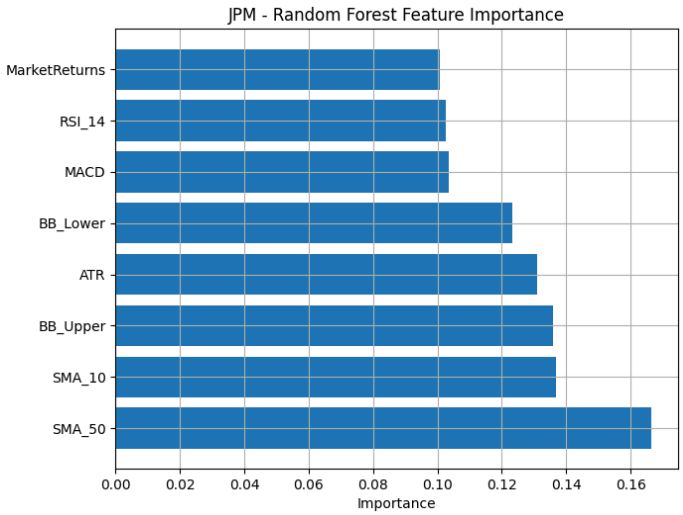
Additionally, the epsilon value, which sets how much prediction error the model allows, is minimal (0.001–0.01) across all stocks. This fine-tuning ensures the model is sensitive to even minor prediction errors, which is critical in the finance sector. For stocks using polynomial kernels, **AAPL** and **PG** use a degree 3 (cubic) polynomial to capture more complex curves, while **JPM** uses a simpler degree 2 (quadratic) polynomial. Overall, these settings help balance accuracy and reliability when predicting stock movements.

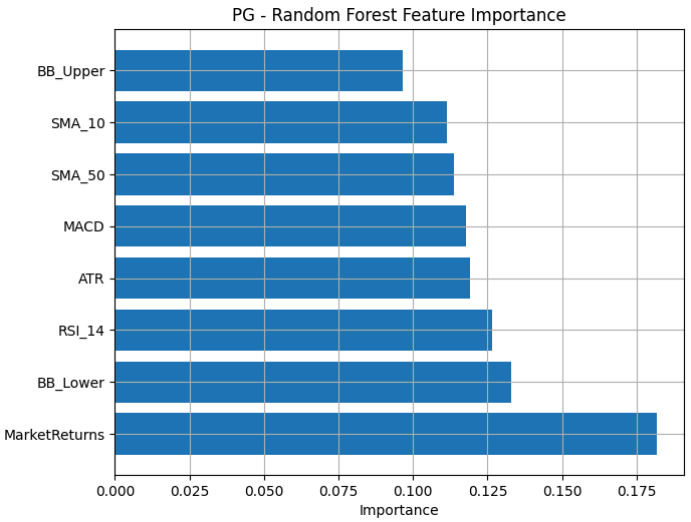
4.3 Random Forest

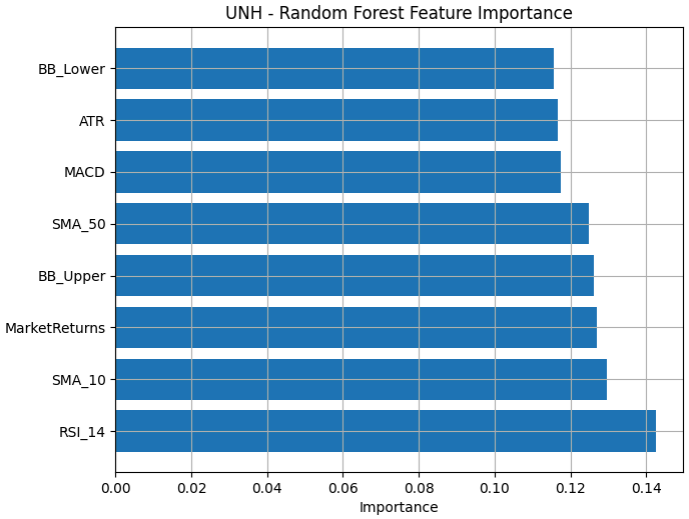
- Feature Importance











The random forest feature importance analysis across the five stocks reveals that no single technical indicator dominated the predictions. Instead, the models rely on a balanced mix of features, which has some common patterns seen. Interestingly, while market returns consistently rank among the top contributors, especially for **AAPL**, **AMZN**, and **PG**, suggesting that overall market sentiment is a key driver of price behavior in these stocks, the Average True Range (ATR) is also a significant feature for **AAPL**, indicating that volatility is the key factor of price pattern in this stock. The 50-day Simple Moving Average (SMA_50) stands out as the most influential feature for **JPM**, highlighting the importance of longer-term trends in this stock. However, the Relative Strength Index (RSI_14) is more significant for **UNH** and **PG**, indicating that momentum and overbought/oversold conditions play a larger role in those predictions. Indicators such as MACD and Bollinger Bands show moderate importance across most models, acting as supportive rather than primary predictors. Shorter-term indicators, such as the 10-day SMA (SMA_10), generally have lower influence, especially in stocks such as **AMZN** and **PG**, possibly due to the increased noise in short-term price movements.

Overall, the models benefit from the combination of different indicators, which each contribute differently depending on the stock, reinforcing the idea that technical analysis should be approached as a composite rather than a single-feature strategy.

4.4 Summary of Best Models per Stock

Stock	Lowest MSE	Lowest MAE	Highest R2	Suggested Best Model
AAPL	SVM (0.000302)	NN (0.01311)	SVM (-0.0599)	SVM / NN
AMZN	SVM (0.000498)	SVM (0.016068)	SVM (-0.0057)	SVM
JPM	SVM (0.000234)	SVM (0.011051)	SVM (-0.0097)	SVM
PG	Linear (0.000121)	Linear (0.008136)	Linear (-0.0628)	Linear
UNH	SVM (0.000221)	SVM (0.010591)	SVM (-0.00378)	SVM

The differences in model performance across stocks can be attributed to a combination of factors, including the complexity of stock price behavior, model architecture, feature interactions, and the quality of the data. The summary of best models per stock table shows that models such as Support Vector Machines (SVM) and Neural Networks (NN) are better suited for stocks with volatile and non-linear patterns such as **AAPL**, **AMZN**, and **UNH** because they can capture complex relationships between input features and return movements. Support Vector Machines (SVM), particularly when tuned with small gamma values, strike a balance between model flexibility and generalisation. This makes them robust against noise and less prone to overfitting. Neural networks can perform well when minimising mean absolute error (MAE), but they require large datasets, careful hyperparameter tuning (e.g., learning rate, network depth, dropout, and batch size), and regularisation techniques to prevent overfitting and ensure optimal performance.

In contrast, linear regression performs best when stock behavior follows a more stable and linear trend, as seen with **PG**. The simplicity of linear models becomes an advantage in these cases since it reduces the risk of overfitting, and it is particularly effective when the relationship between technical indicators and returns is straightforward.

5.Conclusion

This study highlights the potential of machine learning models in enhancing the accuracy of stock return predictions. Among the models evaluated, Support Vector Machines (SVM) consistently demonstrate strong performance across multiple stocks, particularly in handling non-linear relationships and minimising prediction errors. Neural networks also show promise in reducing average errors (MAE), although their performance is more sensitive to data volume and tuning. However, linear regression is most effective for more stable and linear stocks such as PG, indicating that model selection should align with the specific characteristics of the underlying asset. These findings reinforce the value of applying a diverse set of machine learning techniques, tailored to the behavior of individual stocks, to improve forecasting reliability.

To further advance financial prediction using machine learning, several improvements and future directions are recommended. To begin with, incorporating alternative data sources such as social media sentiment, macroeconomic indicators, or company-specific news can enhance predictive capability by capturing market context beyond price trends (Bollen, Mao, & Zeng, 2011). Additionally, leveraging advanced deep learning models such as Long Short-Term Memory (LSTM) networks or Transformers could better capture temporal dependencies in financial time series data (Fischer & Krauss, 2018). Furthermore, ensemble methods that combine multiple models may offer increased robustness by balancing their respective strengths (Gu et al., 2020). As the field progresses, integrating explainable AI (XAI) tools will also be crucial to ensure transparency and trust in model outputs, especially in high-stakes financial applications (Patil, 2025).

Overall, an effective approach that includes abundant and quality data, methodological flexibility across various modeling frameworks, and transparency in analytical interpretations offers great potential for the advancement of machine learning applications in the financial domain.

References

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science, 2(1)*, 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Gu, S., Kelly, B. T., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning. *Oxford Academic*. https://www.nber.org/system/files/working_papers/w25398/w25398.pdf

Kapgate, D., & Chaturvedi, S. (2025). Exploring Machine Learning Methods for Stock Market Prediction: A Review. 7(1), 276–279. https://doi.org/10.35629/5252-0701276279

Patil, D. (2025). *Explainable Artificial Intelligence (XAI): Enhancing Transparency And Trust In Machine Learning Models*. https://doi.org/10.2139/ssrn.5057400

Zouaghia, Z., Kodia, Z., & Ben Said, L. (2023). Stock Movement Prediction Based On Technical Indicators Applying Hybrid Machine Learning Models. 1–4. https://doi.org/10.1109/isncc58260.2023.10323971