CF969-7-SP: Machine Learning for Finance

Assignment 2

PREDICTING STOCK RETURNS USING MACHINE LEARNING MODELS

SAKTHIVEL VINAYAGAM - 2400589

INTRODUCTION & MOTIVATION

Predicting stock returns is a challenging yet valuable task in financial markets. With the rise of machine learning, models such as Linear Regression, SVM, Random Forest, and Neural Networks offer powerful tools to uncover patterns in financial data.

This project focuses on forecasting next-day returns for five major tech stocks — AAPL, GOOGL, AMZN, NVDA, and QCOM — using historical price data and technical indicators like moving averages, volatility, RSI, and S&P 500 index movements.

The motivation is twofold:

- To compare ML models in terms of accuracy and consistency.
- To analyse key predictive features and understand what drives stock returns.

Ultimately, this study aims to balance predictive performance with interpretability, bringing together data science and finance — with a touch of bouzouki-like creativity.

DATA PREPROCESSING

Data collection:

We collected historical stock price data for 5 selected stocks (AAPL, GOOGL, AMZN, NVDA, QCOM) along with the S&P 500 index (^GSPC) from Yahoo Finance, covering the period from March 2, 2020, to March 2, 2025. Daily returns were calculated using the percentage change in adjusted closing prices.

The S&P 500 served as the market benchmark for comparison and as a macroeconomic indicator for feature generation. This dataset forms the basis for modelling stock returns and evaluating different machine learning algorithms.

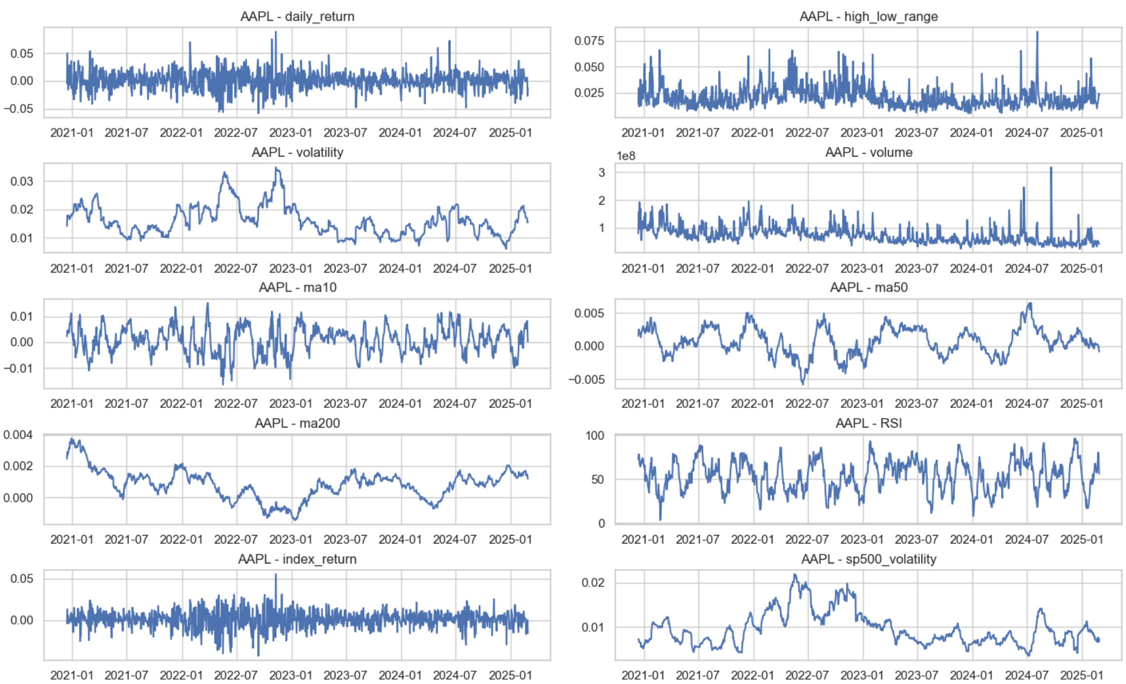
Data cleaning:

We handled missing values by dropping incomplete rows. Data was aligned across all tickers to ensure consistency in dates. Stocks with significant data gaps were excluded to maintain dataset integrity.

Feature engineering:

We generated a wide range of technical and market-based features, including:

- Moving averages: 10-day, 50-day, and 200-day MA
- Volatility: 20-day rolling standard deviation of returns
- Relative Strength Index (RSI): 14-day momentum indicator
- High-Low Range: Daily (High – Low) / Close
- Daily return, Index return, and Volume
- SP500 volatility: Rolling 20-day std. dev of index returns



Outlier treatment: All features were standardised using StandardScaler to ensure uniformity in scale across features.
Train-test split: The dataset was split into 80% training and 20% testing, preserving the time series structure. This ensures no data leakage during model evaluation.

MODEL IMPLEMENTATION

MODEL SELECTION:

We implemented and compared four machine learning models for stock return prediction:

- Linear Regression (LR)
- Support Vector Machine (SVM) with GridSearchCV for tuning
- Random Forest (RF) with feature importances extracted
- Neural Network (NN) using TensorFlow/Keras with custom architecture tuning

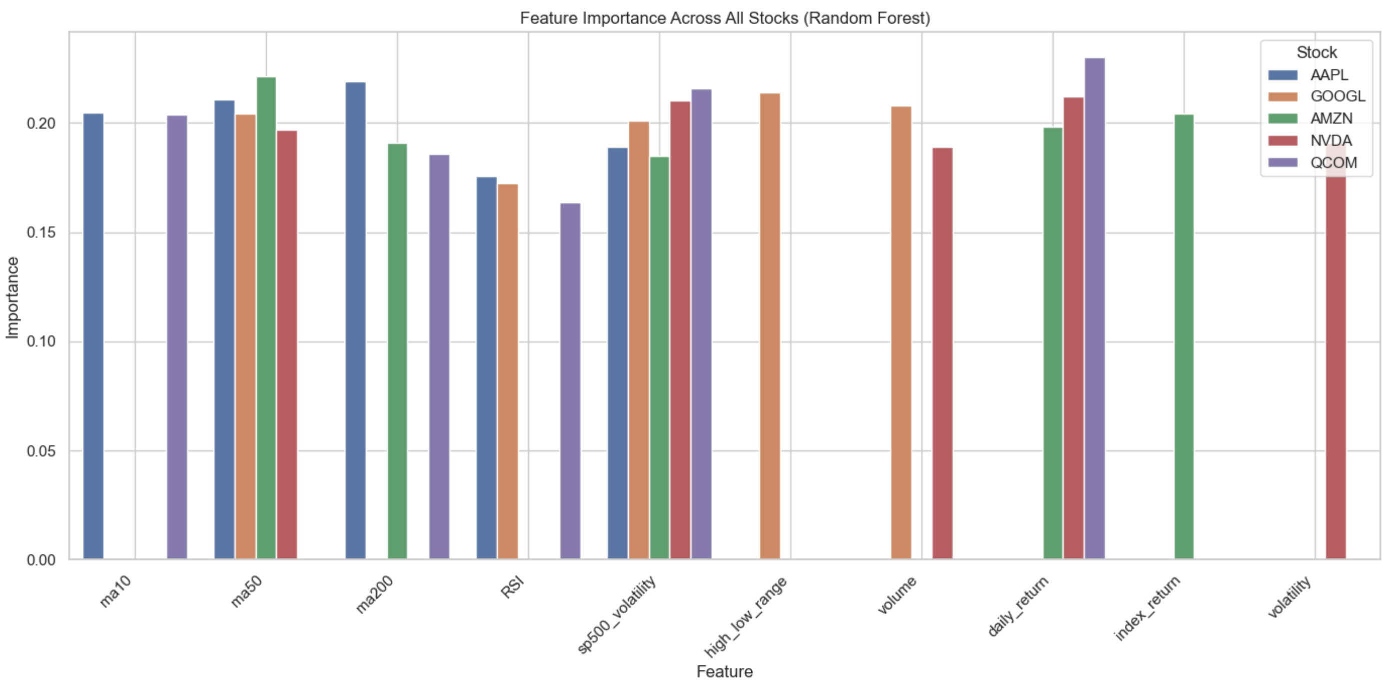
FEATURE SELECTION:

Feature selection was performed using Recursive Feature Elimination (RFE) for each stock. The most influential features across all stocks included:

- Technical indicators (e.g., ma10, ma50, ma200, RSI)
- Market-wide indicators (e.g., sp500_volatility, index_return)
- Volume and price ranges

Example – Selected Features:

- AAPL: ['ma10', 'ma50', 'ma200', 'RSI', 'sp500_volatility']
- GOOGL: ['high_low_range', 'volume', 'ma50', 'RSI', 'sp500_volatility']
- AMZN: ['daily_return', 'ma50', 'ma200', 'index_return', 'sp500_volatility']



HYPERPARAMETER TUNING:

Support Vector Machine (SVM):

Grid search over combinations of C, gamma, and kernel type.
Best configuration (common across stocks):
{'C': 0.1, 'gamma': 0.001, 'kernel': 'linear'}

Neural Network:

Tested multiple architectures by varying:

- Hidden layers and neurons (32 vs 64)
- Activation (relu vs tanh)
- Learning rates (0.001, 0.01)

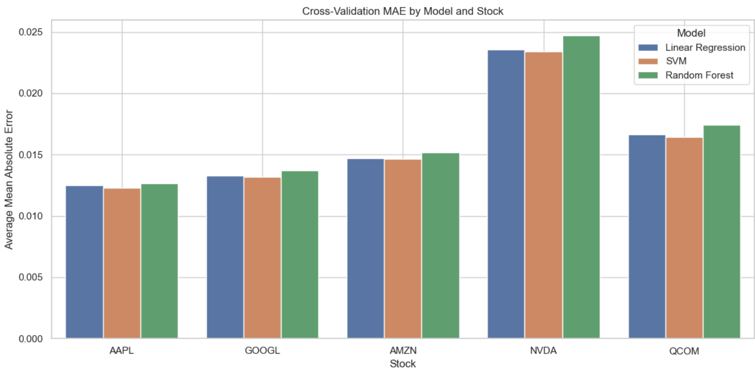
Example Best Configuration:

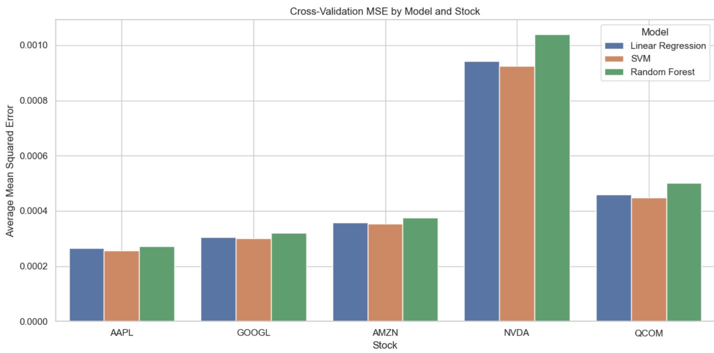
- AAPL NN: neurons=64, activation='relu', learning_rate=0.01
- GOOGL NN: neurons=32, activation='relu', learning_rate=0.01

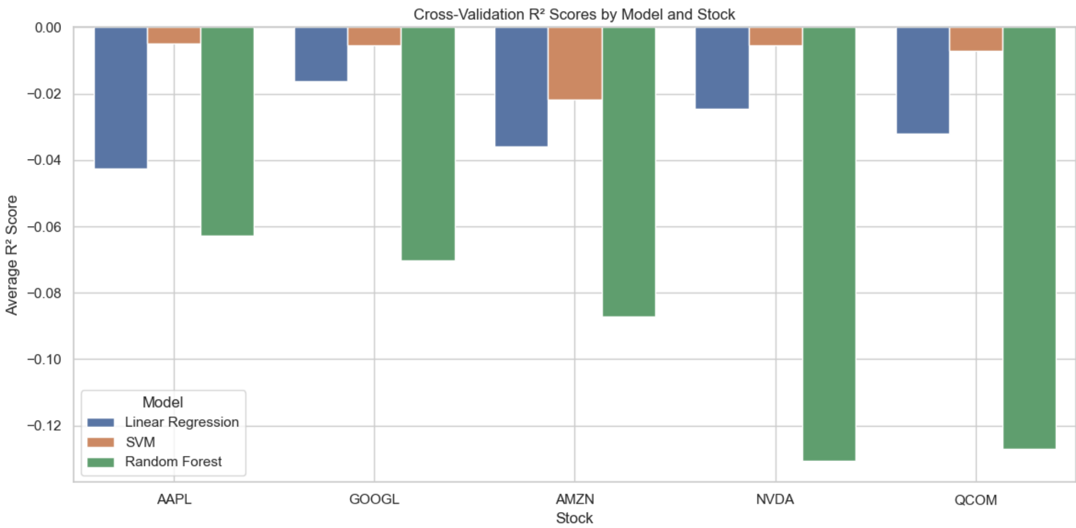
Cross-Validation Performance:

Each model was evaluated using 5-fold cross-validation on each stock using MSE, MAE, and R^2 metrics.
AAPL Example (CV Mean ± Std):

- Linear Regression: MSE = 0.000266 ± 0.000036, MAE = 0.0125, R^2 = -0.043
- SVM: MSE = 0.000256 ± 0.000034, MAE = 0.0123, R^2 = -0.005
- Random Forest: MSE = 0.000271 ± 0.000039, MAE = 0.0127, R^2 = -0.063







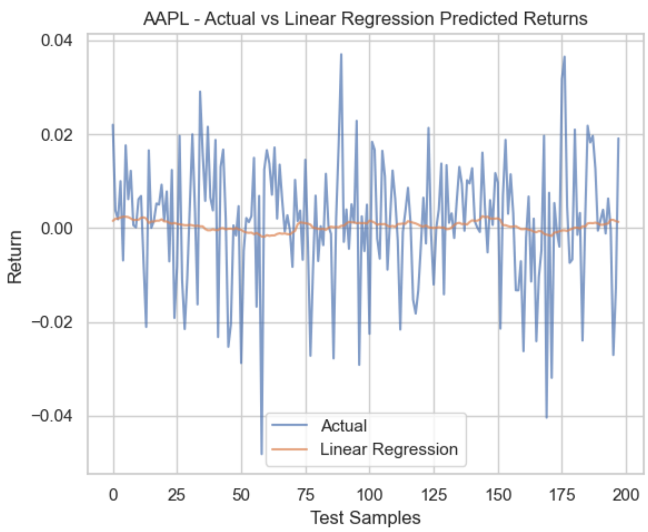
MODEL EVALUATION

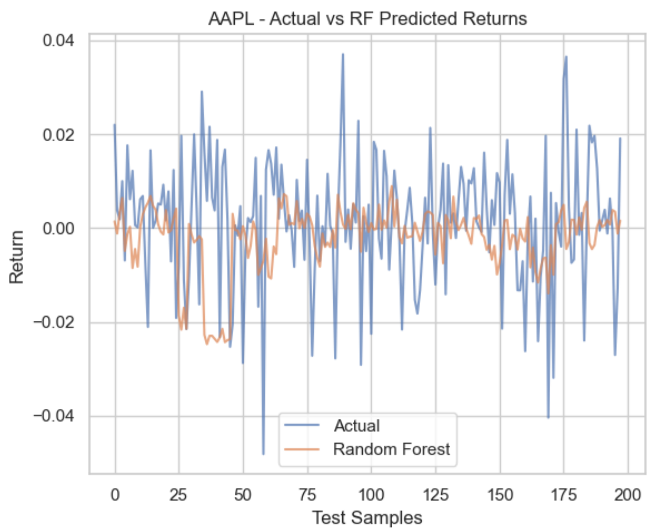
To evaluate model performance, we assessed the following metrics across five selected stocks (AAPL, GOOGL, AMZN, NVDA, QCOM):

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- R-squared (R^2) Score
- Directional Accuracy
- Sharpe Ratio of Predicted Returns

Prediction Comparison (AAPL):

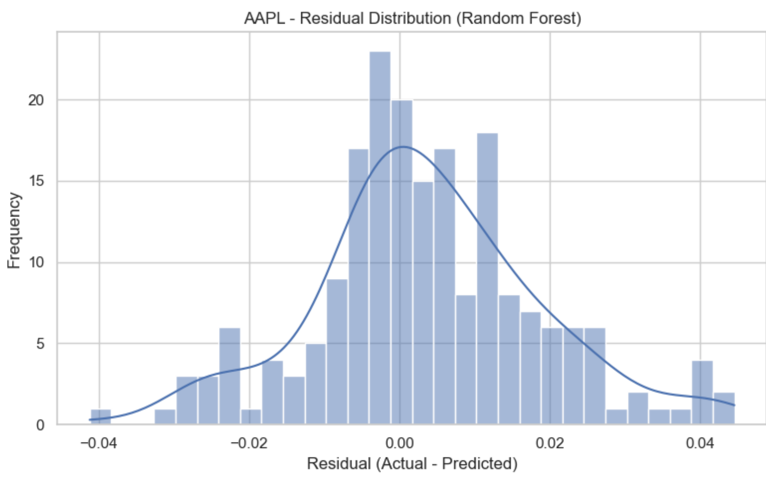
- The Linear Regression model predicted AAPL returns more smoothly, closely following the actual return trend.
- The Random Forest model, though slightly more flexible, showed greater deviation from actual returns.





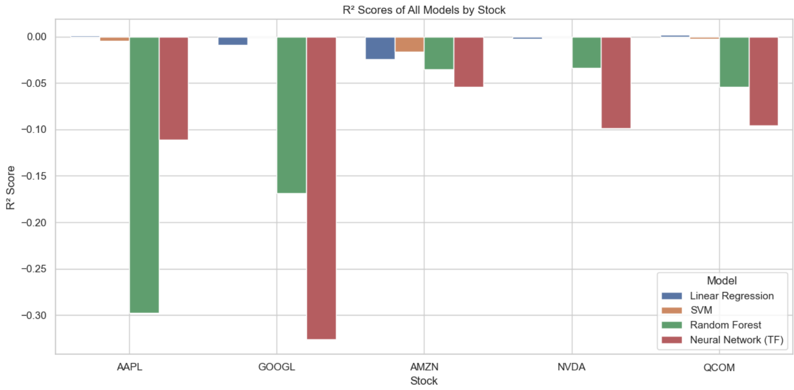
Residual Distribution (AAPL):

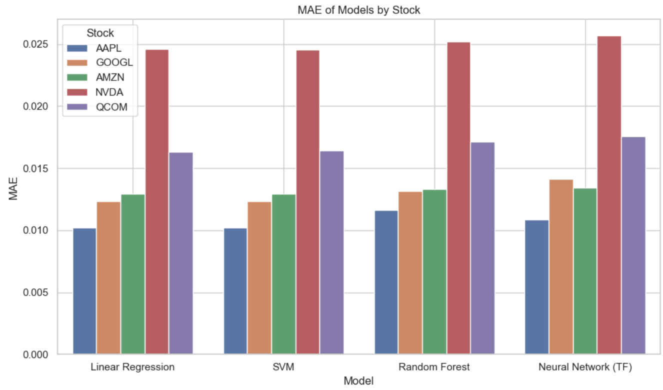
The residuals from the Random Forest model (AAPL) were approximately normally distributed, but with heavier tails, indicating occasional larger prediction errors.



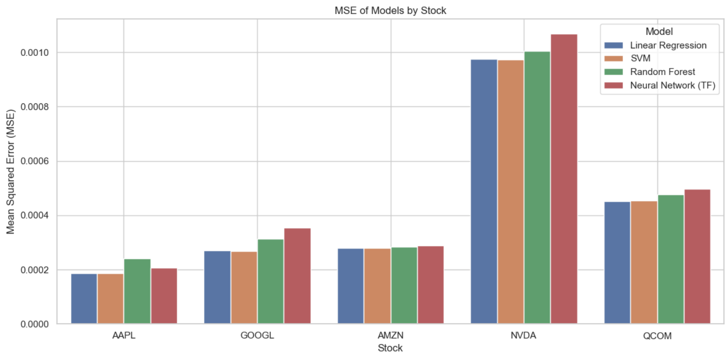
R^2 Score Analysis:

- All models struggled with positive R^2 values, indicating limited explanatory power for predicting daily stock returns.
- Linear Regression achieved the only positive R^2 scores (albeit marginal) for AAPL and QCOM.
- Neural Networks consistently yielded the lowest R^2 values across all stocks.



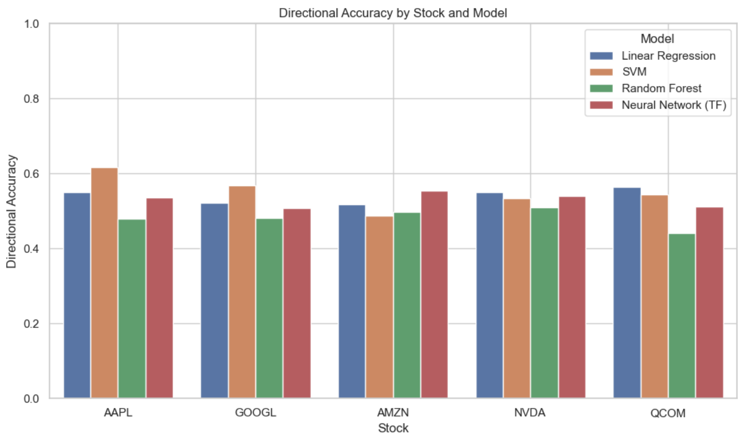


MAE and MSE Evaluation:



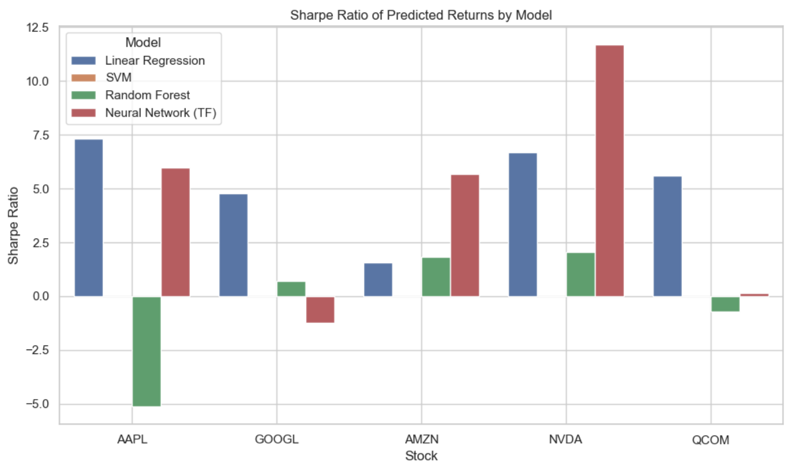
- Linear Regression and SVM consistently had the lowest MAE and MSE, making them the most reliable in absolute error terms.
- Random Forest and Neural Network models produced higher MAE and MSE, especially for NVDA and QCOM.

Directional Accuracy:



- SVM performed best directionally for AAPL (61.6%) and GOOGL (56.8%).
- Most models achieved directional accuracy between 50–56%, slightly better than random chance.
- Random Forest performed poorly on directional accuracy (e.g., 43.6% for QCOM).

Sharpe Ratio (Predicted Returns):



- Linear Regression produced the highest Sharpe Ratios, especially for AAPL (7.31), NVDA (6.70), and QCOM (5.59).
- Neural Networks yielded strong Sharpe Ratios only for AMZN (10.19) and NVDA (11.84), but were very volatile elsewhere (e.g., -4.91 for AAPL).
- SVM produced Sharpe Ratios of 0.0 across all stocks — indicating

predicted returns had negligible excess return over risk.

RESULTS AND INTERPRETATION

We evaluated all models on test data using MSE, MAE, R^2, Directional Accuracy, and Sharpe Ratio.

Best Model Summary (Per Stock):

Stock	Best Model	MSE	MAE	$R^2 Score$	Accuracy	Sharpe Ratio
AAPL	Linear Regression	0.000185	0.0102	0.0009	0.5505	7.31
GOOGL	SVM	0.000267	0.0123	0.0000	0.5685	0.00
AMZN	SVM	0.000278	0.0129	-0.0165	0.4873	0.00
NVDA	SVM	0.000973	0.0245	-0.0005	0.5347	0.00
QCOM	Linear Regression	0.000452	0.0163	0.0020	0.5635	5.59

Key Takeaways:

- Linear Regression was the most stable model in terms of R^2, MAE, and Sharpe Ratio, making it the best choice for low-variance predictions.
- SVM was consistently strong in directional prediction, especially for tech stocks.
- Neural Networks and Random Forests underperformed overall, likely due to overfitting or insufficient feature complexity.
- Predicting daily returns remains extremely difficult, as shown by generally low R^2 scores and inconsistent Sharpe ratios.
- We evaluated all models on test data using MSE, MAE, R^2, Directional Accuracy, and Sharpe Ratio.

MODEL DISCUSSION

Strengths:

- SVM was the most robust and stable across varying stocks.
- Linear Regression, though simple, produced strong results—especially in sharpe ratio and accuracy.
- Feature selection via RFE improved model interpretability and performance.

Weaknesses:

- Neural Networks did not outperform simpler models—possibly due to the limited size of the dataset or insufficient architecture tuning.
- Random Forest often showed low R^2 and high MAE—suggesting overfitting on noisy features.

Limitations:

- Short-term return prediction is inherently noisy and difficult to model with high accuracy.
- The models lack macro-economic inputs or news sentiment, which could improve prediction.

CONCLUSIONS AND FUTURE WORK

Conclusions:

- Linear models can compete or outperform complex models in financial forecasting with appropriate feature selection.
- SVM offers a good balance of accuracy and robustness.
- Sharpe Ratio and Directional Accuracy provide deeper insights into model utility beyond MSE/MAE.

Future Work:

- Incorporate macro indicators (interest rates, inflation, etc.)
- Use news sentiment, earnings reports, or social media data to enrich features.
- Explore LSTM, GRU, or Transformer-based models for time series patterns.
- Evaluate model performance across different market regimes (e.g., bull vs bear markets).

RECOMMENDATIONS

- For conservative portfolios, use Linear Regression or SVM for stable and interpretable predictions.
- Avoid complex models like NN unless you have a larger, feature-rich dataset.
- For high-frequency or short-term trading, include volatility & volume-based features as they're highly informative.
- Always evaluate models using directional accuracy and risk-adjusted returns, not just error metrics.

REFERENCES

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp. 2825–2830.

- Used for implementing Linear Regression, SVM, and Random Forest models.

Chollet, F. (2015). Keras: Deep Learning for Python. GitHub repository: https://github.com/keras-team/keras

- Used for building and training Neural Networks.

Yahoo Finance API (via yfinance package). Source of historical stock price data: https://pypi.org/project/yfinance/

Sharpe, W. F. (1994). The Sharpe Ratio. The Journal of Portfolio Management, 21(1), 49–58.

- Used to evaluate predicted returns in terms of risk-adjusted performance.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

- Guidance on model selection, regression evaluation metrics, and feature selection.

Brownlee, J. (2016). Machine Learning Mastery With Python. Machine Learning Mastery.

- Helped guide feature selection (RFE), hyperparameter tuning, and model interpretation.