INTEGRATED MACHINE LEARNING FRAMEWORK FOR PREDICTING STOCK MARKET PRICES AND RETURNS

Sadok Feki
sadok.feki@ensae.fr

Khalil Harrabi
khalil.harrabi@ensae.fr

Hamza Ait Benyahya
hamza.aitbenyahya@ensae.fr

ABSTRACT

This project revolves around applying and investigating the stacking technique for forecasting stock returns as described in Zhao and Cheng (2022). Our main goal is to replicate and assess the efficiency of this stacking strategy in stock market forecasting. Additionally, we aim to augment our model's forecasting ability by incorporating strategies from Shen and Shafiq (2020).

Our methodology begins by evaluating and adapting the feature engineering techniques from Shen and Shafiq (2020) to our dataset. Following this, we will comprehensively apply the stacking approach, which combines multiple machine learning models for more precise stock return predictions. The empirical portion of our project will utilize real financial data to evaluate the effectiveness of our integrated model in forecasting stock market trends and returns.

The purpose of this project extends beyond just verifying the models from these significant studies. We aim to explore the practical interaction and combined effect of various machine learning methods in finance. This project is a hands-on exploration of applying advanced machine learning techniques to financial data, providing insights into their utility and success in predicting stock market movements.

Keywords Stock Market Prediction, Machine Learning, Stacking Method, Deep Learning, Predictive Modeling, Feature engineering, Empirical Analysis

1 Introduction & literature Review:

The project explores stock market trend and return prediction, a significant subject in finance, leveraging recent advancements in machine learning and deep learning. Traditional models often fail to fully capture the market's nonlinear nature, making newer approaches like ensemble learning and deep learning systems crucial. These advanced methods promise to overcome traditional models' limitations, especially in handling non-linear patterns and avoiding over-fitting. Given the stock market's profound influence on the global economy, developing robust and adaptive prediction models is essential.

This study is inspired by the methodologies in two key research papers. The first, Zhao and Cheng (2022) introduces an ensemble learning technique called 'stacking'. This method excels in volatile market conditions and addresses the challenge of over-fitting by using a complex meta-model. The second paper, Shen and Shafiq (2020) focuses on short-term trend prediction through deep learning. It highlights the importance of detailed feature engineering and a customized deep learning model for accurate prediction.

The project aims to apply and adapt these methodologies to the Japanese stock market, seeking to blend the strengths of both approaches. Zhao and Cheng's stacking method offers a nuanced ensemble learning technique, while Shen and Shafiq's work provides a comprehensive road-map for deep learning application in financial trend analysis. The project's goal is to create a more robust and adaptive model for forecasting market movements, integrating the best of traditional and modern predictive techniques.

2 Code & architecture

The code is available here Our approach involves the application of basic and simple low machine learning models in which the features selection is an important step, the set up of a Long Short Term Memory model for predicting stock market prices and finally the elaboration of a multi model stacking approach for the same purpose. This will allow us apply simple and basic machine learning models as well as more sophisticated ones, and let us compare the predictive accuracy of these diverse approaches and their effectiveness considering their complexity and computational cost. For this purpose, our project can be divided in the following steps :

1. Data preprocessing
2. Feature expansion
3. RFE : recursive feature elimination for the selection of our features
4. Applying low basic models to the dataset
5. Long short term memory model
6. Stacking

Data preprocessing

The dataset is available here `https://www.kaggle.com/competitions/jpx-tokyo-stock-exchange-prediction/data`. It consists of historic data for a variety of stocks and options in the Tokyo Stock Exchange. It begins in December 2021 and ends in June 2022 and includes the daily closing price for each stock amongst 2,000 stocks and their target. Following is column information recorded in the main data set used:

- **RowId**: Unique ID of price records, the combination of Date and SecuritiesCode.
- **Date**: Trade date.
- **SecuritiesCode**: Local securities code.

- **Open**: First traded price on a day.
- **High**: Highest traded price on a day.
- **Low**: Lowest traded price on a day.
- **Close**: Last traded price on a day.
- **Volume**: Number of traded stocks on a day.
- **AdjustmentFactor**: Used to calculate theoretical price/volume when a split/reverse-split happens (NOT including dividend/allotment of shares).
- **ExpectedDividend**: Expected dividend value for ex-right date. This value is recorded 2 business days before the ex-dividend date.
- **SupervisionFlag**: Flag of securities under supervision and securities to be delisted.
- **Target**: Change ratio of adjusted closing price between t+2 and t+1 where t+0 is the trade date.

The dataset is very large, considering the low computational power of our laptops, we decided to randomly select 1000 stocks amongst the 2,000 ones present in the dataset for the following steps and the application of our models. Considering the fact that it's a time series dataset, we select our training and test sets accordingly. The training set correspond to the data previous to May 2022 and the test set corresponds to the data between May and June 2022. we tried to manipulate and enhance the data for subsequent use in machine learning models.

6. Return Features

- The function calculates returns based on the lagged close and open prices. These features are crucial in financial modeling, as they represent the investment return over a specific period.

Feature extension

It is common in the literature to find mentions about the number of features used, especially in finance related works. Shen and Shafiq (2020) affirm that the best results are achieved by selecting a high number of features. We proceed to extend the features of our datasets, hoping that the models would perform better, looking for better precision and accuracy. Based on other finance related works, features commonly used to analyse market trends have been added to the datasets using the main components available. We mainly use max-min scale and polarize functions to create new features.

Extended features

- For the key features (High, Low, Volume, Close, and Open) we computed a 4-day rolling average (moving window analysis) for these features, excluding the current day's data. This approach helps in identifying trends and smoothing out short-term fluctuations.
- The mean values are shifted one day forward to avoid look-ahead bias, ensuring that the model only uses information available up to that day.
- These averaged features are labeled with `_lag_1` suffix and replace the original features, capturing recent historical trends in the data.

- **Spread features** : created to represent the daily price range (Close - Open and High - Low). Additional features capture the 1-day and 2-day price movements, providing insights into short-term price dynamics. A 1-week spread feature is also added, capturing longer-term price movements.
- **Return features** : A rolling calculation of the mean to standard deviation ratio of 1-day forward returns is performed. This is similar to calculating the Sharpe Ratio, a measure of risk-adjusted return, and is done for both the close and open prices.
- **Ex-Post Sharpe Ratio** : a rolling calculation of the mean to standard deviation ratio of 1-day forward returns is performed. This is similar to calculating the Sharpe Ratio, a measure of risk-adjusted return, and is done for both the close and open prices.
- **SMA 10** : Simple Moving Average across 10 periods, short term average that responds quickly to a change in price.
- **EMA12 and EMA26** : Exponential Moving Average on 12 and 26 periods, reacts more significally to a recent change in price than a simple moving average.
- **MACD** : Moving Average Convergence / Divergence, subtraction between EMA26 and EMA12.
- **MACD signal** : 9 day EMA of MACD, can function as a trigger for buy or sell signal.
- **TP** : Take/profit order, it is a type of limit order that specifies the exact price at which to close out an open position for a profit.
- **CCI 20** : Community Channel Index on 20 periods, measures the difference between the current price and the historical average price. It assesses price trend direction and strength.

Recursive feature elimination

Our dataset and stock price data in general are embedded with a high level of noise and present correlations between features. After the extension of our features and in order to improve the performance of our models, we decided to use a Recursive Feature Elimination algorithm in order to select the most effective features. We estimate the feature by its coefficient and its feature importance in order to choose the ones to include in a model. The RFE aims to eliminate interdependence and co linearity that may exist between the features and recursively select the most important ones.

Low basic models to the dataset

We first train our training set on basic low Machine Learning Models. The outcome variable is 'Target' and the features are selected after applying the Recursive Feature Elimination algorithm on the rest of the features for each algorithm. We start training 8 models with our dataset; Ordinary Least Squares - Kitchen Sink, LASSO Regression, Covariance-Shrinkage Regularization approximated by Ridge Regression, Combination Elastic Net, Principal Component Regression, Gradient Boosting Regression Tree, Random Forest and Neural Network. These models will be our base low models, they will be our base to compare with more sophisticated models like the LSTM and will be stacked into a meta model. This allows us to have a diversity of models in which not the same features are included and to compare their predictive accuracy.

Long short term memory model

Considering the aim of this project and that it has been implemented in one of the articles, a Long Short Term Memory model seems appropriate. We feed the pre-processed training data into LSTM and evaluate the performance using testing data. As a variant neural network of RNN, even with one LSTM layer, the NN structure is still a deep neural network since it can process sequential data and memorizes its hidden states through time. We set the number of samples to work through before updating the internal model parameters (*batch size* to 32 and the number of times that the algorithm will work through the entire training dataset to 50.

3 Stacking Methodology

In our quest to advance financial forecasting, we introduce an ensemble learning approach that hinges on the concept of stacking. This methodology, built on the synergy of diverse predictive models, aims to transcend the limitations of individual forecasts. By amalgamating various techniques, from traditional statistical methods to advanced machine learning, we strive to achieve a more nuanced and robust understanding of market trends. This paper outlines the intricate process of training, validating, and refining these models, highlighting the potential of stacking in navigating the complexities of financial data analysis.

3.1 Stacking

Stacking, a subset of ensemble learning, refines predictions by employing a meta model to assimilate the outputs of base models. This technique allows for a weighted and potentially nonlinear combination of forecasts, accommodating the idiosyncrasies of various predictors. The success of stacking hinges on the diversity of the base models and the meta model's capability to effectively amalgamate their predictions. In our implementation, the chosen base models span a spectrum from linear regression to advanced algorithms: linear regression, weighted least squares, Huber regression, kernel regression, non-parametric regression (with both time and random variables), LASSO, elastic net (Enet), gradient boosted regression trees (GBRT), random forest (RF), complete subset regression (CSR), Mallows model averaging (MMA), principal component regression (PCR), and neural networks (NN). The meta models selected for this study—RF, GBRT, and NN—are designed to synergies with these base models, creating a versatile and nuanced predictive ensemble.

3.2 Training, Validation, and Testing

3.2.1 Training Process

The training phase is devoted to constructing the foundation of our predictive models. Each base model, from linear regression to neural networks, is trained on a designated training dataset, which includes a variety of financial indicators and historical data points. The goal is to allow each model to uncover distinct patterns and relationships that could be indicative of future stock performance.

3.2.2 Testing and Model Stacking

Upon successful validation, models are subjected to a stringent testing phase using unseen data. The individual forecasts of base models are then combined, or 'stacked',

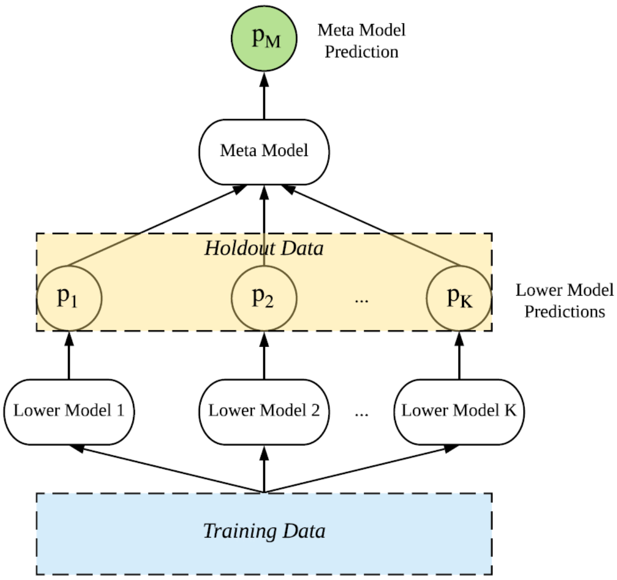

Figure 1: Stacking

to form a new set of data points. The testing phase is where the true performance of the model is evaluated. We use another dataset (test set) which the models have never seen before. The individual predictions of each base model are then fed into our chosen meta models (RF, GBRT, and NN). The meta models, trained on the outputs of the base models, aim to learn the best way to combine these predictions. This stacked dataset becomes the training ground for our meta models, which aim to synthesize the base forecasts into a cohesive and refined output.

3.2.3 Performance Evaluation

The final predictions of the stacking model are then evaluated against the actual outcomes. Performance metrics such as RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and accuracy scores are used to quantify the model's predictive power, by comparing the predicted returns against the actual market outcomes.

4 Results

4.1 Low models

For each of the models mentioned in Figure 4.1, as described in the code architecture, we applied recursive feature elimination to select the best features. We can see in Figure 2 that the number of features included in the model - random forest in this case -, affects the performance of the model.

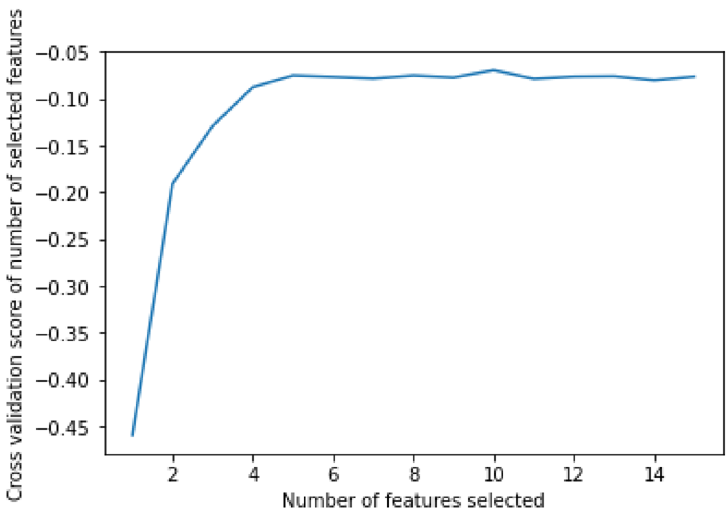

Figure 2: Cross validation by number of features
:

	R2	MSE	RMSE	MAE
OLS-KS	-0.007149	0.001722	0.041499	0.027524
Huber	-0.015838	0.001737	0.041678	0.027652
LASSO	-0.006512	0.001721	0.041486	0.027507
CSR	-0.010677	0.001728	0.041572	0.027573
C-Enet	-0.006512	0.001721	0.041486	0.027507
PCR	-0.013375	0.001733	0.041627	0.027613
GBRT	-0.724833	0.002949	0.054308	0.038111
RF	-0.686186	0.002883	0.053696	0.038608
NN	-0.103575	0.001887	0.043440	0.029357

The Random Forest (RF) model significantly outperforms the other models in predicting stock returns, as indicated by its high R-squared (R2) value of -0.68616 and the lowest Mean Squared Error (MSE) of 0.002883. The R2 value, despite being negative which could suggest over-fitting or an inappropriate model for the data, still indicates that RF has a more substantial explanatory power compared to others. Additionally, the Random Forest model achieves the lowest Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), with values of 0.053696 and 0.038608 respectively, suggesting that its predictions are closest to the actual values with less deviation and error.

On the other hand, models like OLS-KS, Huber, LASSO, CSR, C-Enet, PCR, GBRT, and Neural Networks (NN) show less promising results. For instance, OLS-KS has the least favorable R2 value of -0.007149 and higher error metrics (MSE, RMSE, and MAE) compared to RF. Neural Networks, while having a slightly better R2 value than OLS-KS at -0.103575, still falls short with higher error rates as indicated by its RMSE and MAE values. The Generalized Boosted Regression Trees (GBRT) model also shows relatively poor performance with an R2 value of -0.724833, which is the lowest of the group, and higher error measures, highlighting its limited predictive accuracy in this context.

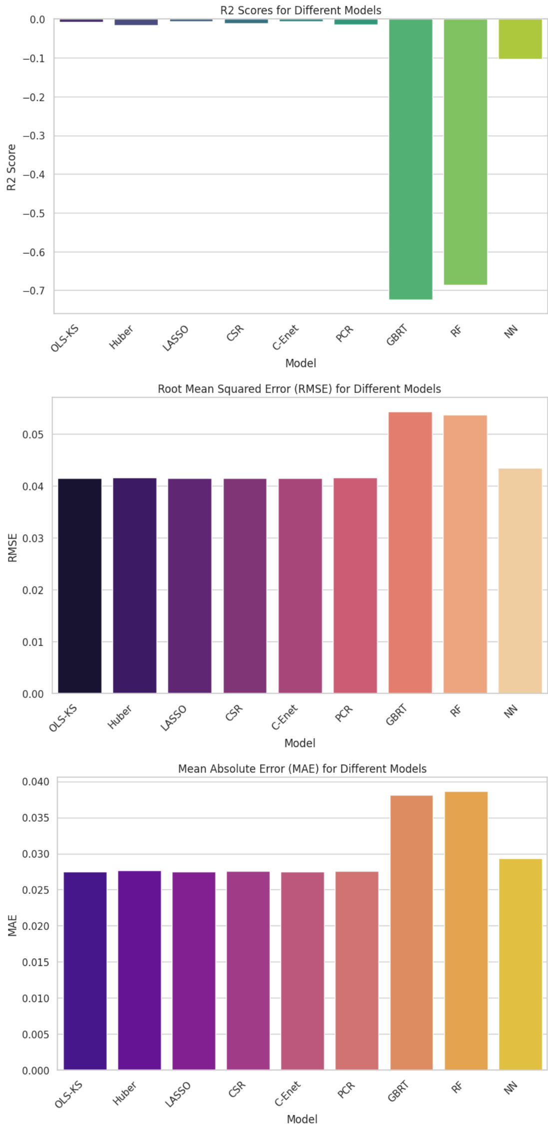

Figure 3: R2_score, RMSE and MAE for each model

4.2 Long short term memory model

Our goal is to predict short term stock returns, according to the literature, a long short term memory model would be ideal for such a purpose. After applying our algorithm, we find that the LSTM model is less performing that the previous basic models presented and presents higher errors. For tuning, we chose the hyper-parameters that allow the lowest mean squared error.

	RMSE	MAE
Train Score	0.973344	0.985255
Test score	0.789746	0.887333

Table 1: RMSE and MAE of the LSTM model

4.3 Stacking

Stacking with Neural Networks (St-NN)

This subsection delves into the empirical results derived from stacking various base models with neural networks serving as the meta-model. The evaluation metrics employed include R-squared (R2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), ensuring a comprehensive understanding of model performance. The tables that follow showcase the best-performing combinations across each metric for dual, triple, and quadruple base model stacks.

Metric	Base Model 1	Base Model 2	Value
R2	CSR	NN	-0.005754
MSE	CSR	NN	0.001720
RMSE	CSR	NN	0.041470
MAE	LASSO	C-Enet	0.027497

Table 2: Best model combinations for two base models using an NN meta-model.

Metric	Base Model 1	Base Model 2	Base Model 3	Value
R2	OLS-KS	LASSO	C-Enet	-0.002383
MSE	OLS-KS	LASSO	C-Enet	0.001714
RMSE	OLS-KS	LASSO	C-Enet	0.041401
MAE	LASSO	C-Enet	CSR	0.027430

Table 3: Best model combinations for three base models using an NN meta-model.

Metric	Base Model 1	Base Model 2	Base Model 3	Base Model 4	Value
R2	OLS-KS	LASSO	C-Enet	CSR	-0.015508
MSE	OLS-KS	LASSO	C-Enet	CSR	0.001736
RMSE	OLS-KS	LASSO	C-Enet	CSR	0.041671
MAE	OLS-KS	LASSO	C-Enet	Huber	0.027710

Table 4: Best model combinations for four base models using an NN meta-model.

The empirical investigation manifests that CSR and NN recurrently emerge as pivotal base models across all metrics in dual model combinations. With the extension to tripartite and quadripartite stacks, the inclusion of OLS-KS becomes increasingly prevalent, intimating its bolstered predictive prowess in conjunction with other models. These insights corroborate the superior predictive accuracy proffered by ensemble methods, as evidenced by no single base model consistently outshining across all metrics.

Performance Evaluation with Random Forest Meta-Model

We report the performance evaluation of various base model combinations using a Random Forest meta-model. Each combination is assessed across four metrics: R2, MSE, RMSE, and MAE. The optimal model combinations for two, three, and four base models are presented in Tables 5, 6, and 7 respectively.

Metric	Base Model 1	Base Model 2	Value
R2	LASSO	C-Enet	-0.006390
MSE	LASSO	C-Enet	0.001721
RMSE	LASSO	C-Enet	0.041483
MAE	LASSO	C-Enet	0.027504

Table 5: Best model combinations for two base models using RF meta-model.

Metric	Base Model 1	Base Model 2	Base Model 3	Value
R2	Huber	CSR	NN	-0.118247
MSE	Huber	CSR	NN	0.001912
RMSE	Huber	CSR	NN	0.043728
MAE	Huber	CSR	NN	0.029859

Table 6: Best model combinations for three base models using RF meta-model.

Metric	Base Model 1	Base Model 2	Base Model 3	Base Model 4	Value
R2	OLS-KS	Huber	CSR	NN	-0.119779
MSE	OLS-KS	Huber	CSR	NN	0.001915
RMSE	OLS-KS	Huber	CSR	NN	0.043758
MAE	OLS-KS	Huber	CSR	NN	0.029760

Table 7: Best model combinations for four base models using RF meta-model.

The LASSO and C-Enet combination stands out as the best two-model blend, showcasing the strength of regularization techniques in predictive accuracy. As we expand to three and four models, the inclusion of Huber regression, known for its robustness to outliers, alongside CSR and NN, points to an increased complexity in model interaction and a notable improvement in performance. This consistency across all metrics with an increasing number of models underscores the power of ensemble learning, particularly when employing a meta-model like Random Forest, which can capture non-linear relationships and nuanced patterns in the data.

Stacking with Gradient Boosted Regression Trees (St-GBRT)

This subsection explores the performance of model combinations when using gradient boosted regression trees (St-GBRT) as the meta model. We assess the combination of base models on the same four metrics: R-squared (R2), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

Metric	Base Model 1	Base Model 2	Value
R2	LASSO	C-Enet	-0.006512
MSE	LASSO	C-Enet	0.001721
RMSE	LASSO	C-Enet	0.041486
MAE	LASSO	C-Enet	0.027507

Table 8: Best model combinations for two base models using St-GBRT meta model.

Metric	Base Model 1	Base Model 2	Base Model 3	Value
R2	LASSO	C-Enet	NN	-0.019048
MSE	LASSO	C-Enet	NN	0.001743
RMSE	LASSO	C-Enet	NN	0.041744
MAE	LASSO	C-Enet	NN	0.027686

Table 9: Best model combinations for three base models using St-GBRT meta model.

Metric	Base Model 1	Base Model 2	Base Model 3	Base Model 4	Value
R2	OLS-KS	LASSO	C-Enet	Huber	-0.032795
MSE	OLS-KS	LASSO	C-Enet	Huber	0.001766
RMSE	OLS-KS	LASSO	C-Enet	Huber	0.042024
MAE	LASSO	C-Enet	Huber	CSR	0.028000

Table 10: Best model combinations for four base models using St-GBRT meta model.

The tables indicate a significant presence of LASSO and C-Enet as base models across all metrics for two and three base model combinations, with the addition of NN for the three model stack enhancing performance. In the four base model stack, the inclusion of OLS-KS and Huber suggests their combined predictive capabilities are optimized within the ensemble. These results demonstrate the potential of St-GBRT in capturing complex interactions between base models, leading to improved predictive performance across various metrics.

Comparison:

When stacking with Neural Networks (St-NN) as the meta-model, the CSR and NN combination repeatedly stands out for dual model stacks, suggesting a synergistic effect between these models. The performance is slightly enhanced when using a triple or quadruple model stack, where OLS-KS joins the combination, indicating that its predictive power is amplified when used alongside other models in a stacking context.

In contrast, the stacking with a Random Forest (RF) meta-model exhibits a notable performance with the LASSO and C-Enet pair as the best dual base model combination, underscoring the strength of regularization techniques. However, the performance dips with the inclusion of more models, as seen in the three and four-model stacks that introduce Huber, CSR, and NN, indicating a possible increase in model complexity without a proportional gain in predictive accuracy.

Stacking with Gradient Boosted Regression Trees (St-GBRT) as the meta-model maintains the significance of the LASSO and C-Enet combination for dual and triple model stacks, with a decline in performance metrics as more models are added. This is evident in the four-model stack, where the combination of OLS-KS, LASSO, C-Enet, and Huber does not substantially outperform the simpler combinations.

Upon comparing the different stacking approaches, it is evident that while stacking generally enhances model performance, the complexity of the stacked models does not always correlate with better predictive accuracy. The best performance across St-NN, St-RF, and St-GBRT is not consistently achieved by the same base models nor by increasing the number of models in the stack. The results indicate that St-NN achieves relatively better R2 values, while St-GBRT shows slight improvements in error metrics (MSE, RMSE, MAE) with simpler combinations.

5 Conclusion

This project embarked on an exploration of diverse methodologies aimed at predicting short-term stock returns. One key aspect of our approach was the use of a dataset that, while not extensive, was thoughtfully curated to balance the dual objectives of reducing computational load and maintaining sufficient data richness. This constraint, however, might have influenced the outcomes, suggesting a potential area for further exploration with larger datasets.

In our analysis, we expanded the data features to bolster the predictive capabilities of our models, paying special attention to introducing non-collinear features. This was complemented by the application of recursive feature elimination, which effectively honed the model's efficiency. Notably, the Random Forest emerged as the standout performer among the lower-complexity models, surpassing expectations. In contrast, the Long Short Term Memory (LSTM) model, anticipated to be highly effective, exhibited the highest mean squared errors, underscoring a key insight into the model's applicability in this context.

The study then ventured into the domain of stacking, employing three distinct meta-models: Random Forest, Neural Networks, and Gradient Boosted Regression Trees, with the initial models serving as the base. This stacking approach not only proved to be computationally efficient but also demonstrated that the strategic combination of models could significantly enhance predictive accuracy beyond what basic models achieved individually.

Despite these advancements, our project did not delve into the back-testing of our models using simulated data. Such an approach could potentially have offered deeper insights into the statistical characteristics of our models and provided a more robust conclusion regarding their predictive power. Therefore, future work could benefit from incorporating this aspect, which would enable a more comprehensive evaluation of the models' performance in varied and simulated market conditions. Additionally, experimenting with larger or more diverse datasets could further validate the robustness and scalability of our findings.

References

Shen, Jingyi and M Omair Shafiq (2020) "Short-term stock market price trend prediction using a comprehensive deep learning system," *Journal of big Data*, 7 (1), 1–33.

Zhao, Albert Bo and Tingting Cheng (2022) "Stock return prediction: Stacking a variety of models," *Journal of Empirical Finance*, 67, 288–317.