QF 2103 GROUP PROJECT REPORT
(Group 14)

National University of Singapore

Trading Project for QF2103 AY 2024/25, Semester 2

April 2025

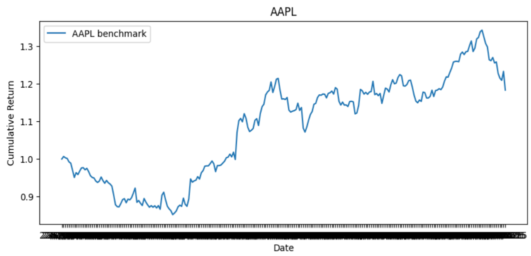
Project Report

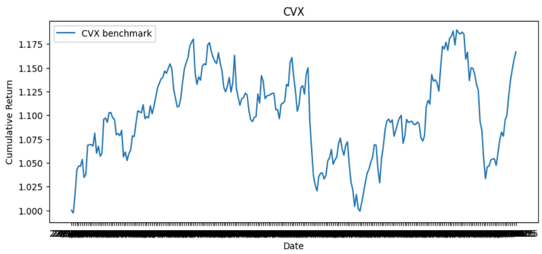
1. Introduction

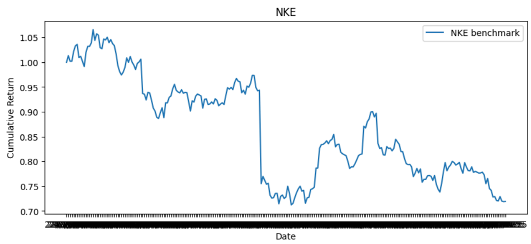
The project required selection of only 10 out of the 20 stocks to create a profitable trading strategy. Hence exploratory data analysis (EDA) was conducted with 2 goals in mind: stock selection and evaluating the performance of different models on the stocks to maximise returns.

EDA

EDA on the list of stocks was conducted with the idea of splitting the stocks by industries, with an idea that similar industries are likely to have similar performance. However we discovered that this is not the case and classified the stocks by general trends, with general uptrend, ranges and down trends. Some examples are shown below:







We decided that the training data is too little for accurate prediction so we used extra stock data from Yahoo Finance from 2023-01-01 to 2024-02-29. The overlapped period with the provided dataset was not merged as we noticed a difference in quoted prices from the source and decided to keep only Yahoo Finance data for training since price trends should be similar.

2. Stock Selection & Justification

Data

Before stock selection, we must first determine a suitable range of dates to train and test the models on. We used the yfinance module to import Close prices of the 20 stocks from Yahoo finance, and the range of dates selected depends on the size of our model's window, stretching from before 2024-01-01, to 2024-03-01.

While the data for 2024-01 to 2024-03 was already provided, we chose to only use Yahoo finance's data for training and testing our models as the 2 datasets differ slightly in their stock prices. To prevent mixing the 2 datasets when we train our models with a rolling window, we decided to only use Yahoo finance's data.

Methods - SMA

The first model tested is the Simple moving average (SMA) model. The SMA model utilizes both long-run trends and short-run fluctuations to determine an appropriate position to take. In our model, we used a 50-day average as an indicator of long-run trends while the short-run trend uses a 30-day average. We chose to not go above 50 days for the long-run as we do not have data before 2024-01 available in the 2024-2025 dataset that we will be applying our strategies on.

For our entry/exit rules, we follow the conventional applications of an SMA model. By comparing the short (ST) and long-term (LT) SMA prices, our strategy is too short when ST-SMA passes LT-SMA from above, which indicates a drop in short term prices. Vice-versa, we will long the stock when ST-SMA passes LT-SMA from below, indicating a rise in the short term.

We first choose 10 stocks to perform SMA with , train the model, then test it over the period 2024-01-01 to 2024-02-29. After that, we then plot the cumulative returns and compare it against the benchmark model. For all our testing, the benchmark model will be a long position in the stock over the same period. Finally, we select the stocks that performed better than the benchmark to include them in our portfolio for the trading period beyond 2024-03-01.

Results - SMA

By comparing the cumulative returns against the benchmark, we can determine which stocks outperformed their benchmark counterpart and determine which stocks included in the portfolio should be using the SMA model.

From the graphs above in Fig. X, we determine that JNJ, NKE, UNH, BA, CVX and PFE outperformed the benchmarks while the other stocks did not. Going forward, these 6 stocks will be included in the portfolio.

Methods - Machine Learning

The second method used in choosing the stocks would be machine learning techniques. Machine learning techniques make use of algorithms that can discover relationships or patterns in data on their own. In our case, we will be using supervised machine learning, which makes use of data that is already labelled to form relationships between them.

Specifically, the models that we will be testing in this section will be Linear Regression, Decision Trees and Support Vector Machines. We believe that including both regression and classification models will give us a better attempt at capturing a wider range of relationships or patterns in the data.

Linear Regression is performed using lags of log returns to predict future log returns. The result is further converted into either +1 to indicate a long position or -0.5 to indicate a short position as a signal for trading (We use 0.5 since only 50% of the stock value is allowed to be shorted). Since Decision Tree and SVM are binary methods, we change the dependent variable to the sign of its corresponding log return, with +1 indicating a positive return and vice versa. Similar to linear regression, we also accounted for the 50% short rule by further converting the binary -1 signals to -0.5 for trading.

For the parameters of the models, Linear used the default parameters while SVM used a linear kernel. For Decision tree, we restricted the maximum depth to 3. Since our actual rolling window for trading beyond 2024-03-01 only has 28 points of data (using the given dataset), we decided to set the max depth to 3 to prevent an issue of overfitting as a depth of 3 already allows for a maximum of 8 leaf nodes. Increasing the depth by 1 would allow situations where most of the leaf nodes only have 1 data point, which could result in overfitting.

For our entry/exit rules, we follow the predicted signals where a positive signal indicates a long position while a negative signal indicates a short position in the stock.

For the testing period, similarly to SMA,we first select 8 stocks, test the models from 2024-01-01 to 2024-02-29 and then calculate their corresponding total returns over the 2 month period. To allow more options in the selection of stocks, we trained the models using both 2 and 3 lags of log returns. To determine the selection of stocks, we implement the models for all the remaining stocks, and select 4 stocks that are able to beat the long benchmark's total returns over the same period.

Results - Machine Learning

Ticker	Benchmark_Roll_2	LinReg_Roll_2	DecisionTree_Roll_2	SVC_Roll_2	LinReg_Roll_3	DecisionTree_Roll_3	SVC_Roll_3
GOOGL	0.983858	0.970043	0.934689	0.995066	0.962089	0.967590	0.995066
KO	1.010258	1.032185	1.032166	1.002233	1.033774	1.020995	1.002233
MCD	1.006442	1.044234	0.979942	0.987366	1.019745	1.067258	0.987366
MSFT	1.023295	1.076504	1.088919	1.114391	1.054632	1.078962	1.114391

After comparing the models, we selected 4 stocks, GOOGL, MSFT, KO and MCD and their corresponding best strategy. Since both SVM 3 lags and 2 lags performed similarly, we went with 3 lags as having more features could potentially fit the data better since we are unsure of the future data points' relationship and patterns.

Summary of Stock selection

Stock	Model Used
PFE	SMA
UNH	SMA

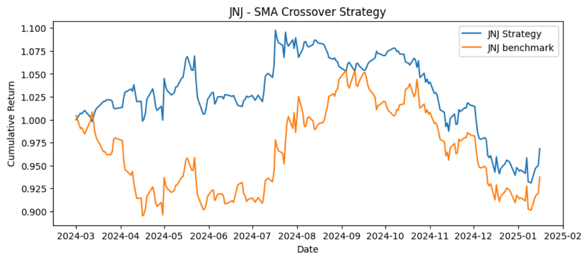
NKE	SMA
BA	SMA
CVX	SMA
JNJ	SMA
MCD	Decision Tree
KO	Linear Regression
MSFT	SVM
GOOGL	SVM

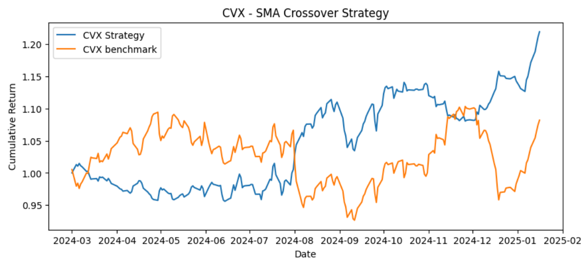
Going forward, these 10 stock and model combinations will be the strategies that we will be using in our actual trading period from 2024-03-01 to 2025-01-16.

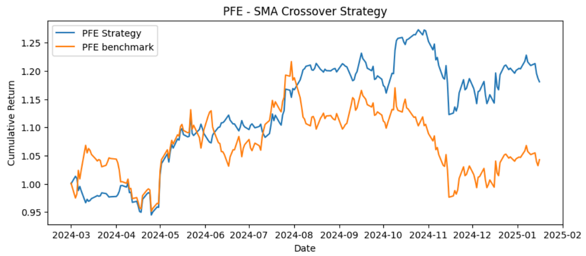
3. Trading Decisions and Documentation

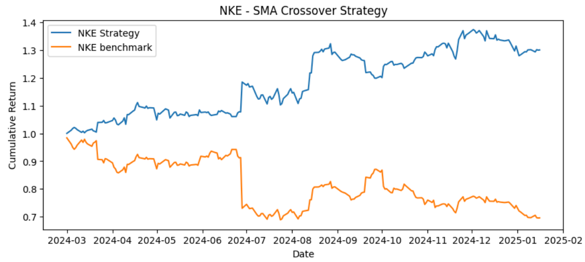
<u>Trading Results - SMA</u>

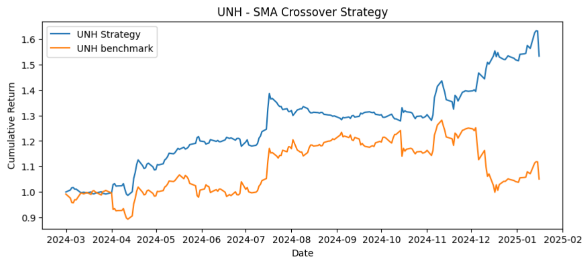
After implementing the strategies in section 2 on the chosen stocks, we obtain the cumulative returns graphs for each of them.

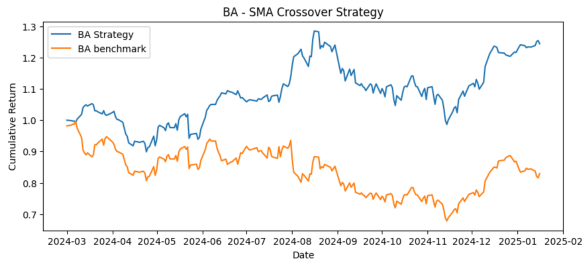












Looking at the results from SMA, we see that all 6 stocks chosen have performed very well over the period of 2024-03-01 to 2025-01-19, with all 6 of them out-performing the benchmarks. The 2 highest returns seems to be NKE and BA, which the model has performed extremely well at predicting opportunities for shorting the stock during times when the stock prices experienced large drops

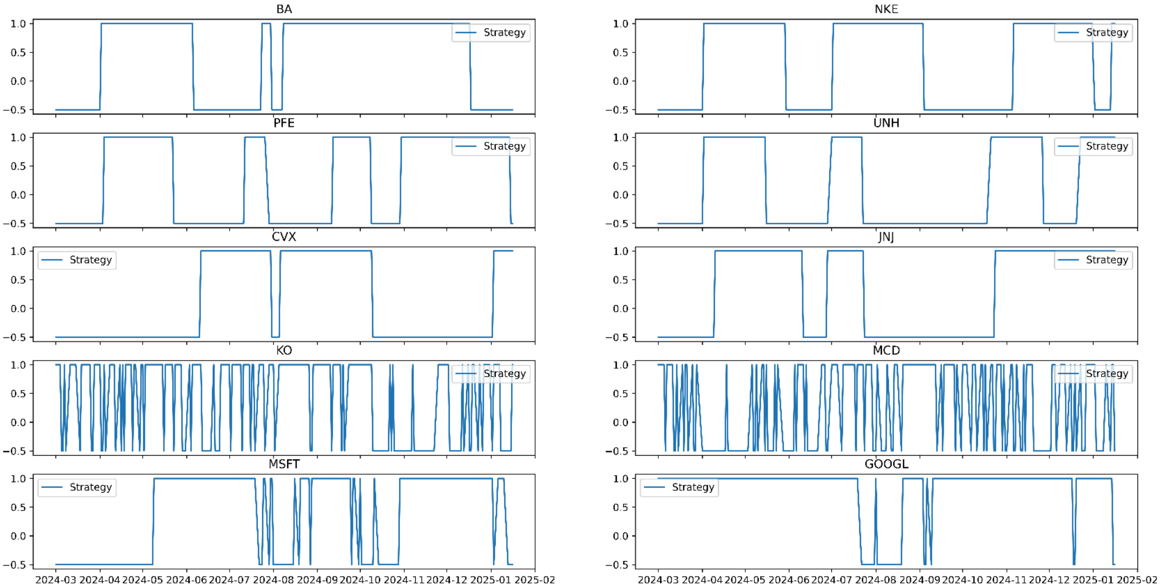
Trading Results - Machine Learning

For machine learning models, the results are abit mixed. While MCD and GOOGL outperformed the benchmarks, MSFT and KO both failed to beat the benchmark, with both having around 7% lesser returns.

KO in particular, has failed to take advantage of the large shifts in its stock prices, resulting in very low returns overall. While its linear regression model has performed well in its model selection period, that performance did not carry over to the 2024-2025 dataset, and this could be due to the the model oversimplifying the stock price dynamics, coupled with having only 3 lags, the noise in its price may have dominated any actual signals from the features.

While the other 3 stocks did not have the best performances as compared to the SMA models above, the use of classification models seems to be better at capturing signals from the data as the 3 models are able to follow closely along with the benchmark, with MCD and GOOGL outperforming it.

Trading Signals



For the trading signals, SMA's signals are very consistent, with minimal changes in positions. This is due to the model relying averages over many days/weeks, hence any immediate shifts in prices will not result in an immediate change in position.

For the machine learning models, since they are very sensitive due to having only 3 lags as features, any change in stock prices and subsequently log returns will very likely cause a shift in its predicted signals. This is especially the case for decision trees which rely on partitioning the input data into many discrete regions.

Trading Logs

Included in separate CSV

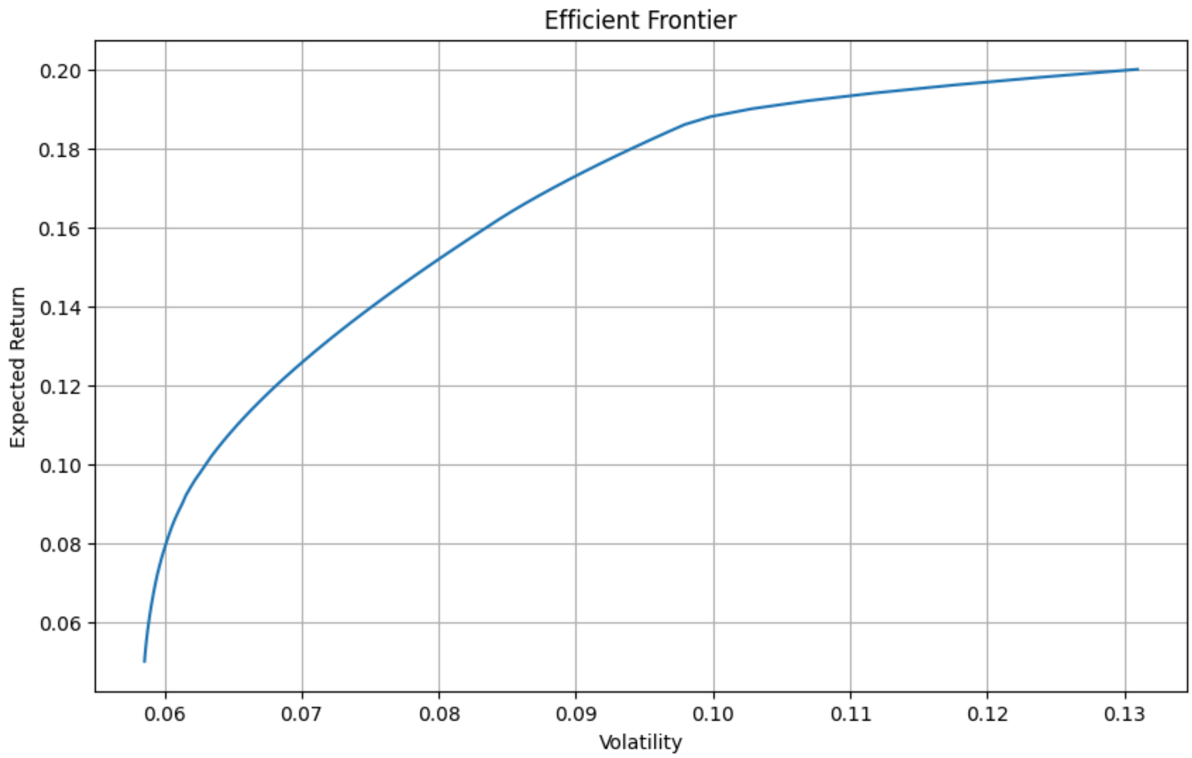
4. Portfolio Optimisation

<u>*Sharpe Ratio Optimization*</u>

After seeing the result of the trading strategies in the training period, we now what to optimise how we allocate our fund, namely, portfolio optimisation.

With a maximum sharpe ratio allocation, we are able to obtain a sharpe ratio of 1.92, as well as expected log return as 0.168 in the training phase, and therefore an expected simple annual return of 0.1833.

Having seen the maximised sharpe ratio case, we would like to explore other possibilities as well, for example, the efficient frontier. Here, we have plotted the efficient frontier based on the returns of our trading strategies in the training phase.



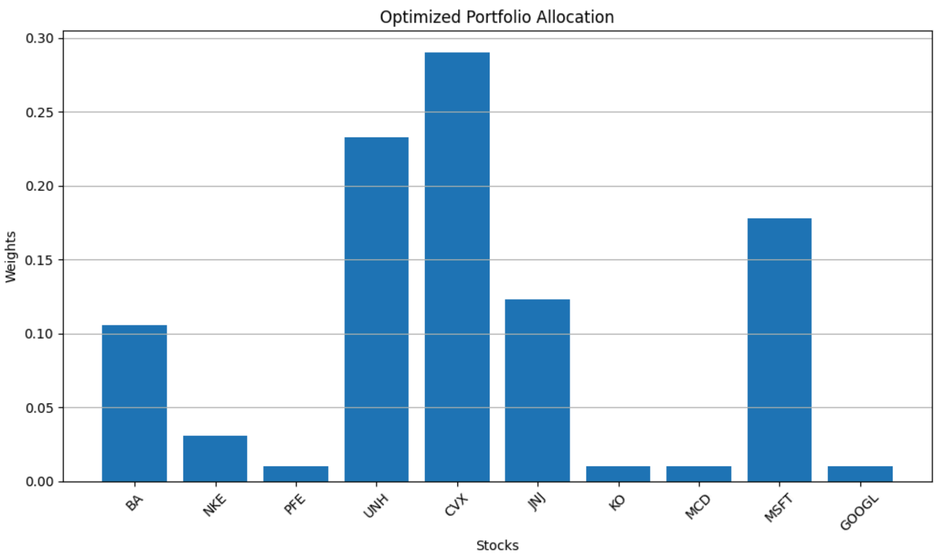
As we can see, the volatility was not too high at the point of maximum sharpe ratio, and we believe that we could accept this volatility below 0.09. As the volatility further increases, the expected return tapers off, thus we would rather like to just choose the point with the optimal sharpe ratio as our final portfolio weight allocation.

Weights

Here is a table for the respective stocks' weight.

```
Ticker
BA       0.105345
NKE      0.030863
PFE      0.010000
UNH      0.232652
CVX      0.290275
JNJ      0.123169
KO       0.010000
MCD      0.010000
MSFT     0.177697
GOOGL    0.010000
dtype: float64
```

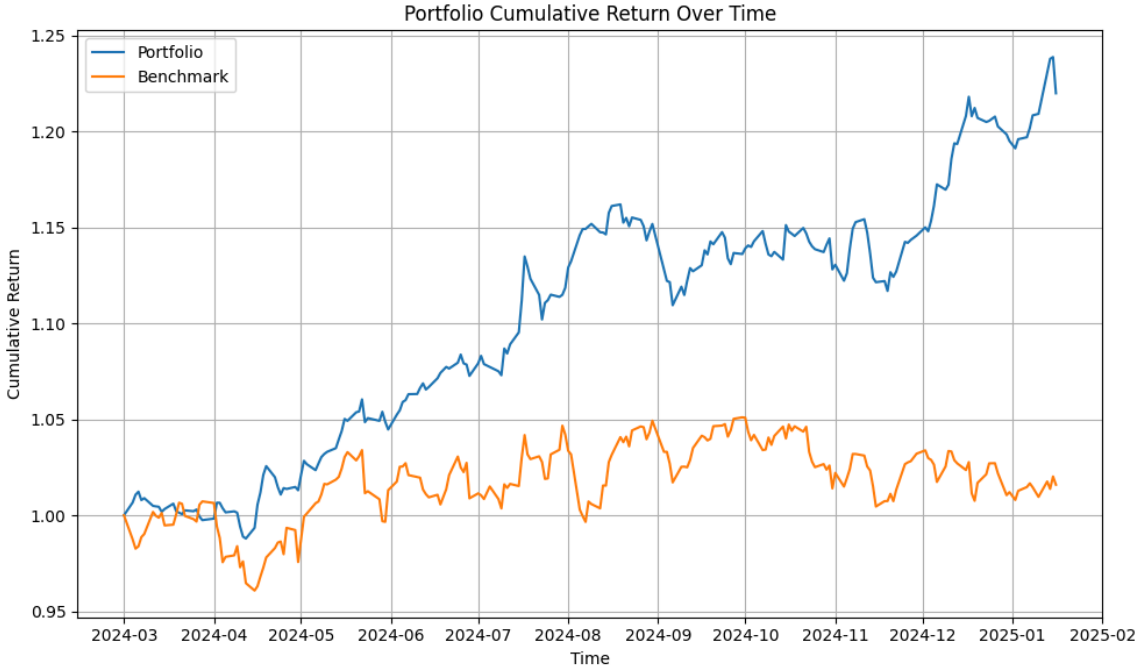
As we can see, there are four stocks with 0.01 as their weightage, namely PFE, KO,MCD and GOOGL. The reason why they have a weightage of 0.01 is that we have limited the bounds to be (0.01,1), in order to diversify the risks and make sure that we invest in all 10 stocks, since we have already chosen them based on our strategies. The following chart gives a direct visualisation of all the weights given to different stocks.



Cumulative Portfolio Returns

Having seen the predicted results in the training period, we are curious to see what actually happens when these weights and strategies are put into practice in the trading period.

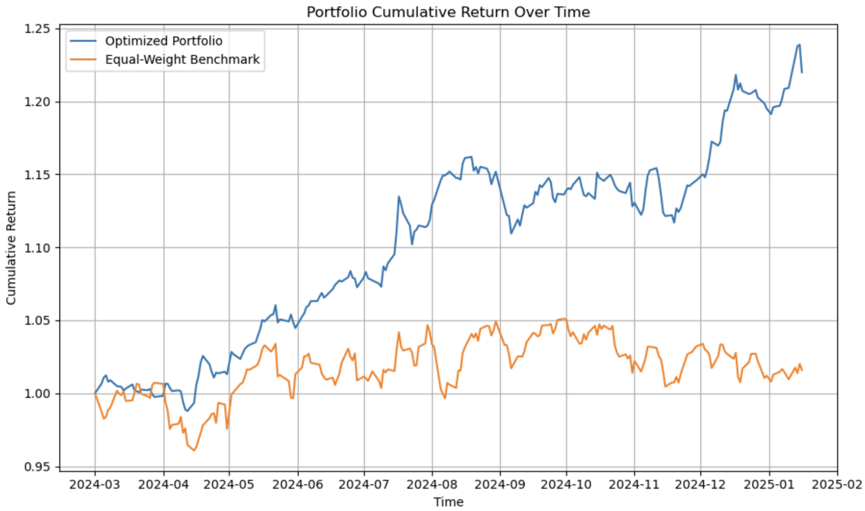
Therefore, the following is a graph plotted to show the cumulative return of our strategies + optimised weights, against the benchmark, who has equal weights and simple long strategy.



As we can see from the graph, our strategy far outperforms the benchmark. The final Portfolio Return is 1.219828180624119 at the end of the trading period, and final Benchmark Return is 1.0159257315394803 at the end of the trading period. By simple calculation, we can see that our strategy outperforms the benchmark by 20.070605828209076%, which is more than 20%, signalling an overall success of the strategies and optimised portfolios.

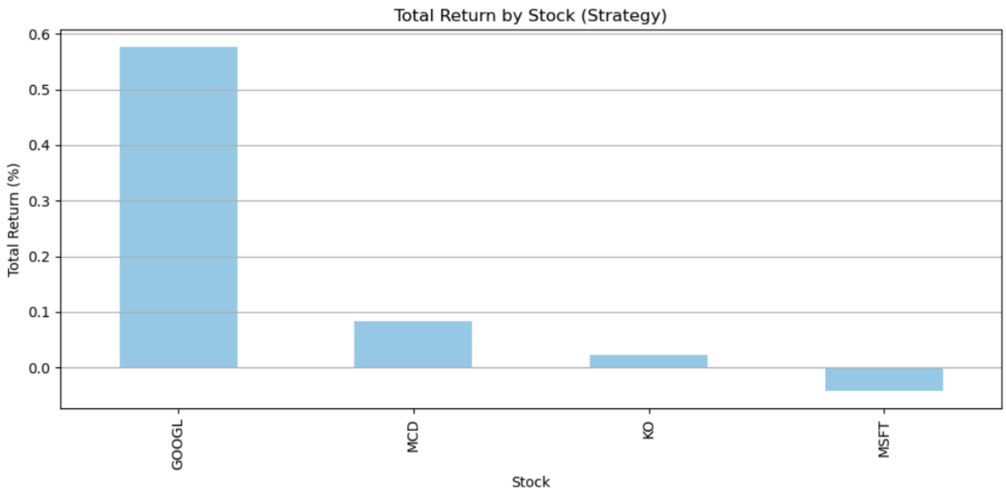
5.Results, Analysis & Conclusion

Portfolio



Our strategy achieved a final cumulative return of **21.99%**, significantly outperforming the passive benchmark of **1.59%** as shown in the cumulative returns over time plot. This outperformance, which widened over time, demonstrated the compounded effectiveness of dynamic signal-driven strategies combined with risk-aware portfolio optimisation. These results validate our overall workflow - from data-driven model selection top optimal weight allocation.

Returns by stock



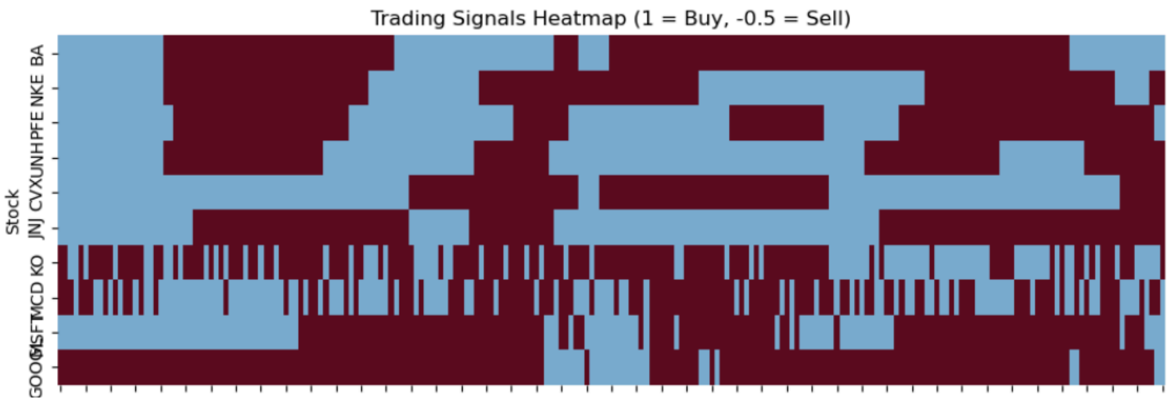
As seen from the bar plot showing total returns per stock, returns varied substantially across the 10 selected stocks. Notably:

- GOOGL (+57.6%), modeled using an SVC, and UNH (+53.3%), modeled using SMA, were the top contributors.
- NKE (+30%) and BA (+24.5%) also generated strong returns under the SMA strategy.

- In contrast, MSFT (-4.2%) and JNJ (-3.2%) underperformed, despite MSFT receiving a high allocation of 17.8% based on its favourable Sharpe ratio during the training phase.

These results highlight a critical nuance: while our Sharpe ratio optimisation prioritised low-volatility performers (e.g. CVX at 29% weight), it also underweighted high-return stocks like GOOGL (just 1%) due to their higher volatility. This led to stable portfolio growth, but potentially left additional alpha on the table. Such trade-offs suggest the future value of return-volatility balanced rebalancing strategies.

Heatmap



The signal heatmap visualises how frequently each strategy triggered positions changes:

- MCD, MSFT and KO experienced frequent signal shifts, driven by classifier sensitivity to short-term volatility.
- In contrast, NKE, BA and CVX showed sustained position holding, reflecting successful long-trend capture through SMA.
- These patterns demonstrate that our models were not passive, instead they adapted daily, cutting exposure during noisy phases and capitalising on directional trends.

Future Improvements

The final portfolio achieved outperformance of over +20%, combining simple rule based models with machine learning to generate diversified, risk-adjusted returns. The variation in results among stocks, however, suggests room for refinement:

Future strategies may benefit from adaptive rebalancing to capture emerging alpha from underweighted high-return stocks
Incorporating model selection based on historical volatility patterns or ensemble blending may enhance model-stock fit and reduce downside risk.

Conclusion

In conclusion, the project validated the value of a hybrid strategy framework - one that combines intuitive rule-based methods with more sophisticated predictive tools and highlights the importance of flexibility, adaptability and continuous model evaluation in active portfolio management.

6. Team Roles and responsibilities

Member	Roles and responsibilities
Ng Jun Hao	Training of models for stock selection using ML models (DT,Linear, SVM), Portfolio returns & visualization, visualizing trading signals
Vernon-Evans George Jeremy	Evaluation and analysis of trading results, including benchmarking against a passive strategy. Plots of cumulative returns, stock-level performance, and model signal responsiveness. In-depth analysis on model effectiveness, asset allocation, and return attribution across the portfolio.
Ren Guanpeng	EDA, training of models for stock selection and trading, implementing overall code pipeline and documentation.
Lyu Zhengxiang	Portfolio optimisation, preliminary training models for stock selection and trading, determination of portfolio weights and a portion of result graph plotting.