Course: Master of Science in Quantitative Finance

Subject: QF 634 Applied Quantitative Research Methods

Project Title:

Optimizing Trading Strategies With Machine Learning:

A Case Study On HSI Index

Group Number: Group 8

Group Member:

HUANG Ziyan

WONG Keat Ye

XIANG Sihan

YU Lingfeng

Table of Contents

1 Introduction 3
2 Data Description 4
2.1 Data Pre-processing 4
2.2 Feature Engineering 4
2.2.1 Historical Data: 4
2.2.2 Self-correlation Features: Technical Indicators 4
2.2.3 Selective Correlation Features: Correlated Assets 4
2.2.4 Effective Correlation Features: Currency Exchange Rates & Indices: 4
3 Methodology 5
3.1 Regressor Algorithms 5
3.2 Classifier Algorithms 6
3.3 Batch DTW 6
4 Methodology Comparison 8
4.1 Result Comparison 8
4.1.1 Regressor Algorithms 8
4.1.2 Classifier Algorithms 9
4.1.3 KNN-DTW Algorithms 11
4.2 Final Model: Linear Regression 13
5 Discussion 15
5.1 Limitations 15
5.1.1 DTW 16
5.2 Improvements 17
6 Conclusion 17

1 Introduction

The Hang Seng Index (HSI) is the main stock index in the Hong Kong market. This index includes the top 50 companies with the highest market value among the stocks listed on the Hong Kong Stock Exchange, widely reflecting the overall performance of the Hong Kong stock market. So and Tse (2004) mentioned that the Hong Kong dollar maintained its peg to the U.S. dollar even in the currency crisis wave, implying that HSI is worthy of research.

Figure 1-1 Time Series of HSI

The following assumptions are proposed to make the model construction more reasonable:

- **Assumption 1**. Assume transactions without friction and the market is full of liquidity

  Reason 1. Trading frictions like trading fees are related to trading size. It is too early to consider in training set and testing set.

- **Assumption 2**. Assume there is no slippage during each transaction.

  Reason 2. Slippage is due to the time lag between the order time and actual trading time. Slippage is hard to quantify because it is highly related to the crowdedness of trading strategy and IT infrastructures.

- **Assumption 3**. Short selling is allowed in the model but ignored the borrowing costs.

  Reason 3. Borrow costs are related to investors' credit rating.

- **Assumption 4**. Returns are calculated daily, but reported in annualized results

  Reason 4. Convenient to report and compare across different models.

In this project, machine learning methods are divided into three categories, which are **regressor algorithms, classifier algorithms, and batch DTW,** then using these methods to forecast HSI's future trends and determine trading signals. Finally, pick the strategy with the best performance. This project uses a more advanced method to track market changes, which has practical significance for both quantitative trading and macro research fields.

2 Data Description

2.1 Data Pre-processing

The dataset used for training and prediction in DTW Batch strategy is a time series of daily percentage price changes with the normalization transformation implied by scaling its values between -1 and 1.

The dataset in the regressor algorithm is a time series of daily log returns.

The dataset in the classifier algorithm is the sign of daily log returns, with negative value is -1 and positive value is 1, representing sell and buy respectively.

2.2 Feature Engineering

2.2.1 Historical Data:

Retrieve historical price data for Hang Seng Index (HSI) from November 2001 to October 2023 from yahoo finance website. Obtain data for correlated assets, currencies, and relevant indices during the same time frame. Split the data into a training set (November 2001 - October 2011), testing set (November 2011 - October 2013), and deployment sets (November 2013 - October 2023).

Select features that cover various aspects, ranging from autocorrelation to full correlation, and encompass both domestic and international economic environments as well as monetary policies, aiming to capture various crucial influencing factors in the Hong Kong market.

2.2.2 Self-correlation Features: Technical Indicators

Lag features for the HSI prices to calculate daily lag log returns.

Moving averages of HSI in 21 days, 63 days, and 252 days.

Exponential moving averages of HSI in 10 days, 30 days, and 200 days.

2.2.3 Selective Correlation Features: Correlated Assets

Leading stocks representing different industries, respectively as **0005.HK for HSBC Holdings plc, 0001.HK for CK Hutchison Holdings Limited, 0293.HK for Cathay Pacific Airways Limited, 0003.HK for The Hong Kong and China Gas Company Limited,** provide a limited but crucial reflection of expectations for the Hong Kong economy. These companies typically hold leading positions in different industries, influencing overall market sentiment. These limited correlation features can to some extent reflect market heat, especially the heat associated with the performance of these representative companies.

2.2.4 Effective Correlation Features: Currency Exchange Rates & Indices:

Features related to the global economic environment, **VIX (CBOE Volatility Index), JP225 (Nikkei 225), T-bill (10-Year Treasury Constant Maturity Rate), SPX (S&P 500), Shanghai Composite Index**, offer a broader market context. These features reflect dynamics in the global economy and financial markets.

Exchange rates of **USD/JPY**, as well as **GBP/USD**, can more comprehensively capture the impact of global monetary policies and money supply. Due to the fixed exchange rate of the Hong Kong dollar to the US dollar, the USD exchange rate holds significant importance in the Hong Kong market.

3 Methodology

3.1 Regressor Algorithms

A regressor is a type of machine learning algorithm used for predicting a continuous outcome variable based on one or more predictor variables. In other words, a regressor is designed to establish a relationship between the input features and the target variable, allowing to make predictions about the target variable for new, unseen data. The process of building a regression model involves finding the best-fitting line or curve that represents the relationship between the input variables and the output variable. This line or curve is then used to make predictions for new data points.

The following regressor methods are employed in the project:

- Linear Regression
- LASSO Regression
- Ridge Regression
- Support Vector Machine Regressor
- Elastic Net
- Decision Tree (CART)
- K-Nearest Neighbors (KNN)
- Random Forest
- Extra Trees
- Gradient Boosting Regressor (GBR)
- AdaBoost Regressor (ABR)

Moreover, the following hyperparameters are considered to fine tune in the project in these three models:

Method	Hyperparameters
LASSO and Ridge	• α
Support Vector Machines Regressor	• Kernel Types • α • Degree (Degree of the polynomial kernel function) • C (Regularization parameter)

Table 3-1-1 Hyperparameters of Each Method

Strategy performances best when $\alpha = 0.3$ for LASSO and Ridge regression. However, all three types of kernels for SVM Regressors have the same bad performance because none of them beat the market asset return, which indicates that this method is not a very effective way to predict the index.

3.2 Classifier Algorithms

Classifier algorithms are a type of machine learning technology used for pattern identification and classification. They are crucial in supervised learning, in which the algorithm is trained on labelled datasets to generate predictions or assign labels to previously unknown data points. These algorithms learn from the inherent patterns and relationships in the training data, allowing them to generalize and make sound conclusions in new situations.

The following classifier methods are employed in the project:

- K-Nearest Neighbors (KNN),
- Support Vector Machines Classifier (SVC),
- Logistic Regression (LR),
- Decision Tree (DT) and
- Random Forests (RF)

Moreover, the following hyperparameters are considered to fine tune in the project in each method:

Method	Hyperparameters
K-Nearest Neighbors	• Number of Neighbors (k) • Distance Metrics • Weights • Algorithm Types
Support Vector Machines Classifier	• Kernel Types • α • Degree (Degree of the polynomial kernel function) • C (Regularization parameter)
Decision Tree	• Maximum Depth • Splitting Strategy • Maximum Number of Features • Minimum Number of Samples ○ Internal Node ○ Leaf Node
Random Forest	• Number of Estimators • Maximum Features • Maximum Depth • Random State

Table 3-2-1 Parameters of Each Method

In KNN, SVC, and Decision Tree models, although fine-tuning the hyperparameters is crucial for the project, due to machine constraints, only the highlighted hyperparameter will be fine-tuned. Alle of the hyperparameters of the Random Forest model can be fine-tuned.

3.3 Batch DTW

Dynamic time warping is employed in identifying the degree of similarity between two time series. The Euclidean distance may not facilitate a match between two similar sequences due to delays and other factors. In contrast, the dynamic time warping algorithm stretches or compresses the test sequences to ensure maximum alignment.

Consider two time series, sequence Q and sequence C, each of length n and m respectively. Construct an $n \times m$ matrix lattice with matrix elements (i, j) denoting the distances $d(Q_i, C_j)$ between the points Q_i and C_j. The distance measures the similarity between each point of sequence Q and C, where the smaller the distance, the higher the similarity.

The DTW algorithm involves identifying a path through a lattice of grid points to align two sequences computationally. The grid points where the path passes represent the points of alignment. This path is called the warping path regularization path and is designated as W. The k^{th} element of W is defined as $W_k = (i, j)_k$, which determines the mapping of sequences Q and C. In the condition of satisfying the constraints, the objective is to find the path that minimizes the cost of regularization. It is the principle of DTW, which can be expressed as,

$$DTW(Q, C) = min\left\{\frac{\sqrt{\sum_{k=1}^{K} W_k}}{K}\right\}$$

where the parameter K is used to normalize different warping paths with different lengths.

In this report, the K-Nearest-Neighbour-with-Dynamic-Time-Warping (KNN-DTW) algorithm is introduced to determine the appropriate cluster for a new time series. This algorithm measures the distances between the new time series and all the time series within the training dataset. The k nearest time series then is selected, and the new time series is assigned to the cluster that most of the k nearest neighbours belong to.

It must be noted that the computational magnitude of the DTW distance is $o(MN)$, when both time series are longer, the efficiency of the DTW algorithm is slower and cannot meet the demand. For this reason, we introduce the FastDTW method, which combines the use of both constraints and data abstraction to accelerate the computation of DTW to $o(N)$.

Restriction (Figure 3-1) is to reduce the search space of DTW within the sequence space.

Data abstraction (Figure 3-2), i.e., to statute the time series of previous length N into a length M (M<N) representation: when being compared.

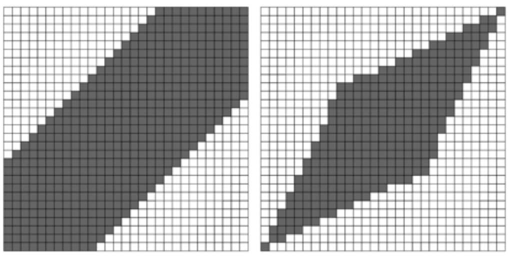

Figure 3-1 Restriction method in FastDTW

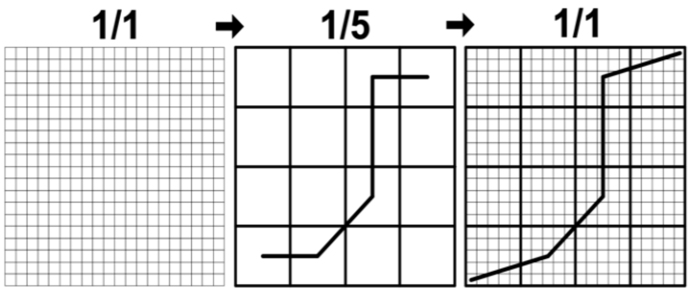

Figure 3-2 Data Abstraction method in FastDTW

4 Methodology Comparison

4.1 Result Comparison

In this session, we will compare our results mainly by evaluating the Sharpe ratio, followed by the compound annual growth rate (CAGR), Sortino ratio, maximum drawdown percentage and days, multiples of invested capital, and win rates.

4.1.1 Regressor Algorithms

A. Summary of Regressor Algorithms

To select effective models, compare the Mean Square Error (MSE) in the training set and testing set after regression. In Figure 4-1-1, the Decision Tree Regressor, Random Forest Regressor, and Extra Trees Regressor present overfitting because the errors in the training set are quite low, while a lot higher in the testing set. Therefore, these three kinds of methods are not effective and not practical, and they will be excluded when considering the model backtesting performances.

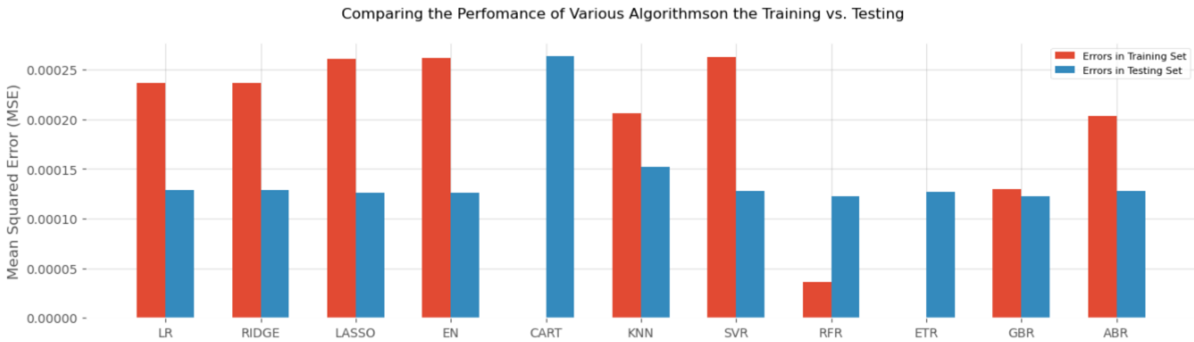

Figure 4-1-1 Bar Chart of Regressor Algorithms Result in Training and Testing Set

Method	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
LR	34.9200	1.9015	2.9782	-15.0839	161	1.8218	54.6748
GBR	40.5046	1.9921	3.1040	-14.5883	103	1.9760	57.9268
RIDGE	23.7315	1.3954	2.1120	-15.0839	157	1.5318	53.6585
LASSO	5.8528	0.3443	0.5031	-18.4068	303	1.1207	50.8130
KNN	19.6643	1.1994	1.8491	-11.6164	156	1.4327	51.6260

Table 4-1-1 Backtesting Results for Regressor Algorithm

Although the trading strategy based on GBR has the highest CAGR and Sharpe ratio, the results strongly depend on the selection of the random state factor, which brings significant randomness to the construction of strategies. Meanwhile, the Sharpe ratio of the strategy based on linear regression is only around 0.09 lower than the figure for the GBR method, and the result generated by linear regression will be more stable and reliable than the Gradient Boosting Regressor. Consequently, regard linear regression as the best regressor algorithm.

B. Deployment on Regressor Algorithms

To demonstrate the above viewpoints, these methods will be deployed from November 2013 to October 2023. The results obtained by the linear regression algorithm are far better than other regression algorithms in the long term.

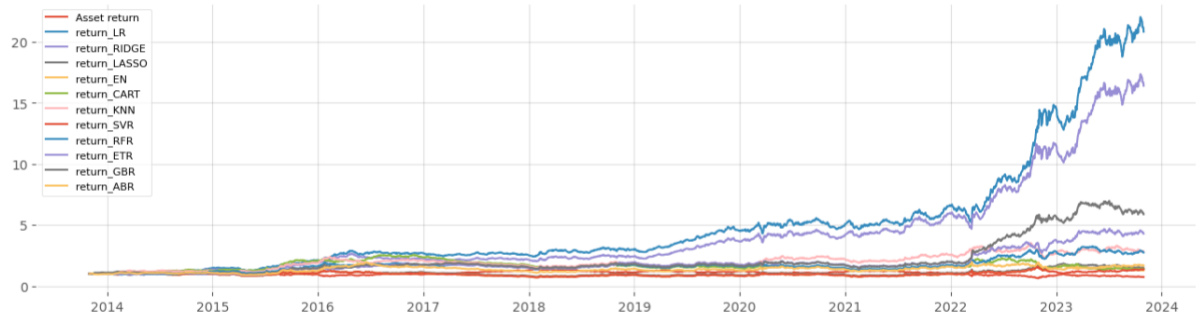

Figure 4-1-2 Line Graph Comparing Asset Return with Regressor Algorithm in Deploy Set

4.1.2 Classifier Algorithms

A. Summary of Classifier Algorithms

Based on the findings, it is discerned that in KNN, among various values of K, optimal performance is achieved when the number of K is set to 9.

Then, due to the similarity of SVC and LR, we compare SVC with logistic regression, utilizing their inherent commonalities to gain insights into their respective performances. Based on the presented results, it is determined that the LR model exhibits the best performance. Notably, LR outperforms SVC within this analytical framework, underscoring its greater efficacy in the given scenario.

Based on the information, it is evident that in DT, the configuration yielding optimal performance occurs when the maximum depth is set to 4.

The fine-tuning process for the Random Forest Classifier involves optimizing parameters such as the number of estimators, maximum features, maximum depth, and random state. Although the number of estimators was initially considered, its inclusion did not yield satisfactory results, leading to its omission from the table below, which showcases the top 5 best-performing models.

Model Number	Maximum Number of Features	Maximum Depth	Random State
4	5	None	1
7	None	3	1
6	None	2	1
8	None	4	1
0	None	None	1

Table 4-1-2 Random Forest Model Selection

Derived from the insights gleaned, it is evident that, within the RF model, optimal performance is attained in Model 4, characterized by a configuration where the maximum number of features are set to 5 and the random state is set to 1.

The summary of the classifier algorithms results are shown below.

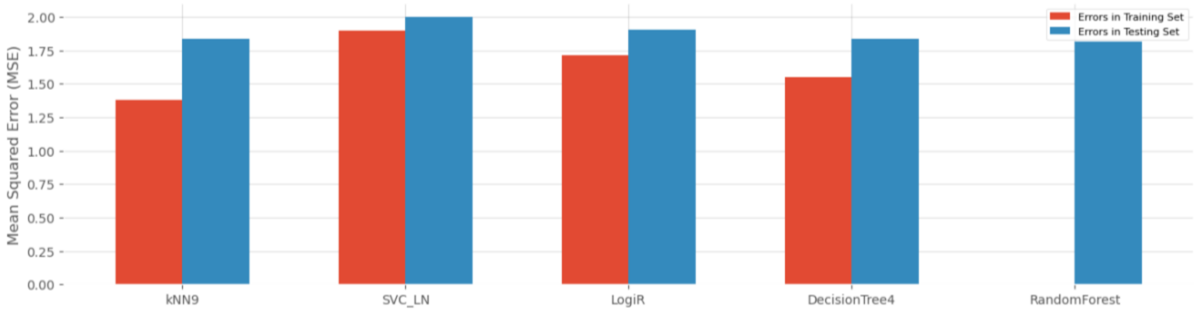

Figure 4-1-3 Bar Chart Comparing the Performance of Classifier Algorithms on the Training and Testing Sets

Machine Learning Model	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
Asset Return	**8.1733**	**0.6154**	**0.8970**	**-16.8251**	**274**	**1.1704**	**51.0163**
Decision Tree (Depth = 4)	42.9619	2.0901	3.4125	-14.8205	110	2.0458	54.0650
Random Forest (Model 4)	37.6403	1.8703	2.9395	-16.1001	155	1.8961	54.6748
KNN (k = 9)	25.2268	1.4645	2.2542	-11.8462	258	1.5691	54.0650
Logistic Regression	19.6925	1.0529	1.6592	-21.3615	256	1.4333	52.4390
SVC (Linear)	-7.0362	-0.4061	-0.5586	-30.1168	547	0.8641	50.0000

Table 4-1-3 Backtesting Results for Classifier Algorithms

Based on the graphical representation and tabular data presented above, it is discerned that the decision tree model attains the highest performance, boasting a Sharpe ratio of 2.0901. Nevertheless, upon closer examination of its performance on both training and testing sets, it becomes evident that the decision tree model falls short of the efficacy demonstrated by logistic regression. Consequently, through a meticulous comparative analysis encompassing mean squared error (MSE) metrics and comprehensive backtesting results, it is conclusively determined that logistic regression emerges as the preeminent model within the realm of classifier algorithms.

B. Deployment on Classifier Algorithms

Machine Learning Model	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
Asset Return	**-2.9994**	**-0.0504**	**-0.0713**	**-55.7008**	**2101**	**0.7374**	**51.3206**
Logistic Regression	23.7113	1.1679	1.7574	-24.6399	518	8.4013	52.8647
Random Forest (Model 4)	10.5054	0.5993	0.9071	-34.9102	954	2.7162	50.6705
KNN (k = 9)	8.5271	0.5111	0.7736	-35.9702	1517	2.2671	49.2076
Decision Tree (Depth = 4)	5.2265	0.3565	0.5019	-54.3402	1453	1.6646	52.6615
SVMs (Linear)	-3.2534	-0.0636	-0.0895	-57.0879	3344	0.7183	49.9797

Table 4-1-4 Backtesting Results for Classifier Algorithms (Deployment)

Following the deployment of the algorithms, it is evident that logistic regression manifests the most favourable performance.

4.1.3 KNN-DTW Algorithms

An analysis of HSI's historical performance is shown below. The process begins with the retrieval of daily closing prices from January 1, 2000, to November 30, 2023, using different window sizes (15, 20, and 30 days), to identify patterns in the historical data to gain insight into potential market trends. Following these historical events, the index is analysed for its performance in the days following them.

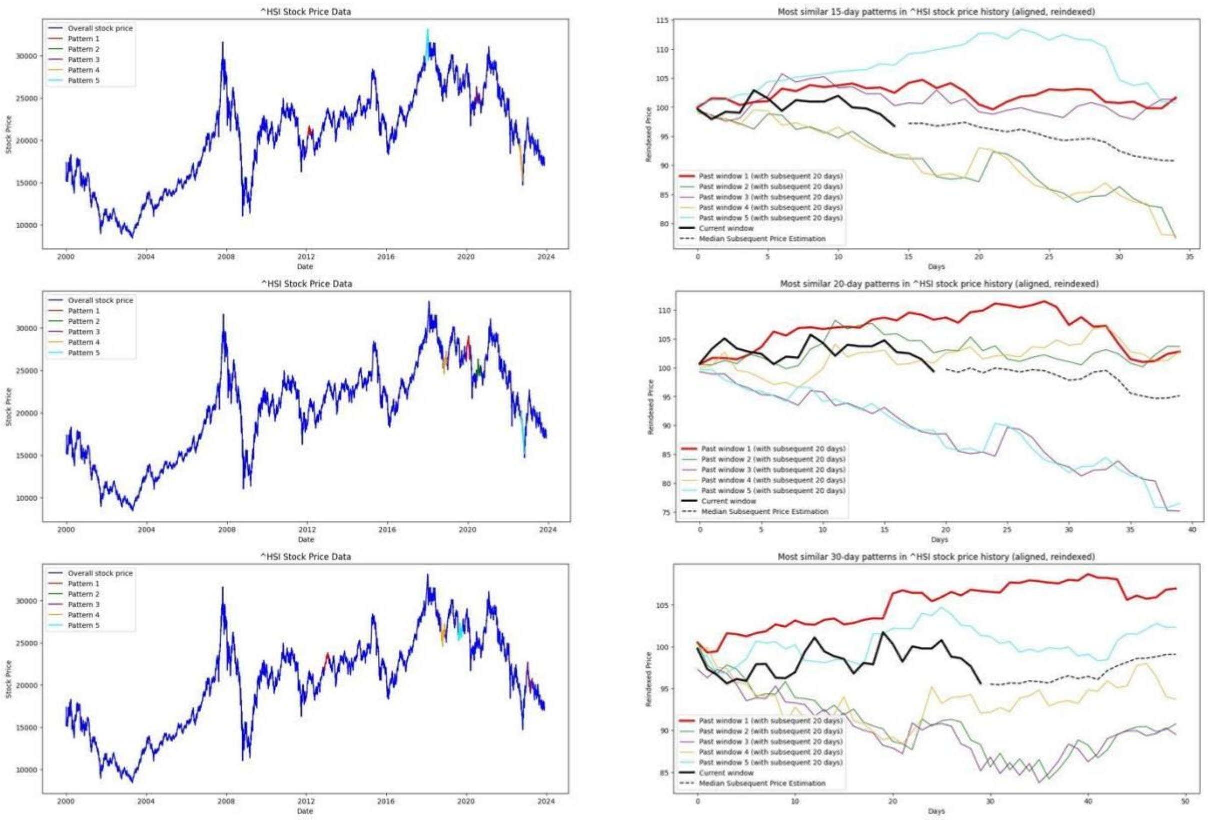

Figure 4-1-4 Performance of patterns using 15, 20 and 30 days as window size

In the above performance, it is evident that varying time periods have a considerable influence on selecting historical data patterns, subsequently impacting future condition predictions of the model. The model development for this project incorporated four critical hyperparameters to enhance profitability.

Hyperparameters	Denotation	Fine-tune Range
Days of Training Window	W	10, 13, 15, 17
K in K-Nearest-Neighbor	K	1, 2, 3
Days of Prediction	F	1, 2, 3
Threshold of prediction movements(10^{-4})	TH	0, 4, 8, 10, 12, 14

Table 4-1-5 Model Hyperparameters and Fine-Tune Ranges

The approach commences by using a set dataset of three months to create a model through experimenting with different hyperparameter combinations and identifying the most effective set of hyperparameters. First, try all combinations of 3 hyperparameters with threshold 0.

Hyperparameter Combination	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
Asset Return	9.6893	0.8831	1.3399	-11.6164	71	1.0236	50.8197
{W=17, K=2, F=1}	79.7368	3.1360	6.7218	-4.3174	12	1.1593	50.9804

Table 4-1-6 The return of optimal combination of 3 hyperparameters

The combination $\{W = 17, K = 2, F = 1\}$ exhibits the highest Sharpe ratio and the lowest maximum drawdown and can be selected as the best strategy.

Then consider the value of threshold, with fine tuning on the training dataset with different thresholds and compare the performance.

Best Combination with Different Threshold (TH)	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
0	79.7368	3.1360	6.7218	-4.3174	12	1.1593	56.8181
4	79.7854	3.2777	7.1956	-3.5414	11	1.1593	58.5366
8	79.7854	3.2777	7.1956	-3.5414	11	1.1593	58.5366
12	69.9684	2.9837	6.5239	-3.5414	11	1.1431	56.4103
14	69.9684	2.9837	6.5239	-3.5414	11	1.1431	56.4103

Table 4-1-7 Comparison of 3-Fixed-Hyperparameter Combination Different Thresholds

It is observed that threshold = 4×10^{-4} is the best value of return to take. Here, it is notable that only first 3 months of testing data has been used because of the time complexity, which is the main reason to upgrade the database in the latter research.

This optimal hyperparameter combination finalized as $\{W = 17, K = 2, F = 1, TH = 0.0004\}$, is then utilised on a ten-year test set, validated on a rolling basis at six-month intervals to determine the efficiency of the model. In the figure below, the Sharpe Ratio of the test data over six months is used to represent the performance of the strategy with the optimal combination.

Figure 4-1-5 Sharpe Ratio in 6-Month of Asset and Strategy

The strategy achieved a 50% success rate in the deployment test over a 10-year period, **not** delivering the anticipated high returns.

Given the limitations of the information provided by the database to inform future predictions, **upgrading** its quality yielded more acceptable results. Manually observe a strong up-trending interval from 5th June 2012 to 19th April 2013 to replace the database and use a wider range of hyperparameters repeating the above algorithm.

Hyperparameters	Fine-tune Range
Days of Training Window (W)	5-25
K in K-Nearest-Neighbor (K)	1, 2, 3
Days of Prediction (F)	1, 2, 3
Threshold of prediction movements (10^{-4}) (TH)	0, 10

Table 4-1-8 Hyperparameters in Deployment Test with Wider Range

The performance of part trials of hyperparameter combinations in testing data is shown in the Table 4-1-9, as founded the better combination $\{W = 10, F = 1\}$.

Hyperparameter trials under {W=10, F =1}		CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
K=2	TH=0	80.4754	3.7533	6.5130	-9.5732	191	3.2624	60.2510
	TH=10	80.1242	3.9536	7.0305	-8.4474	80	3.2497	61.8267
K=3	TH=0	43.2872	2.2837	3.6727	-19.0926	154	2.0551	56.9937
	TH=10	40.4861	2.3596	3.7574	-17.7180	164	1.9755	57.8692

Table 4-1-9 Performance of Combinations of Hyperparameters in Testing 2 Years

Apply the selected hyperparameters as a benchmark to testing the model's performance over the next 10 years.

Hyperparameter trials under {W=10, F =1}		CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
K=2	TH=0	6.9453	0.4388	0.6385	-48.7060	1436	1.9575	51.0910
	TH=10	6.6159	0.4450	0.6513	-46.6614	1425	1.8980	51.4908
K=3	TH=0	10.9988	0.6245	0.9247	-47.8253	1783	2.8399	51.7468
	TH=10	10.7215	0.6482	0.9522	-44.0784	1437	2.7698	52.6495

Table 4-1-10 Performance of Combinations of Hyperparameters in Deployment 10 Years

It is learned the model $\{W = 10, K = 2, F = 1, TH = 0.001\}$ that performed well in the testing period did **not** maintain their dominance in the deployment interval. In fact, the hyperparameter combination which displayed the best performance in the deployment is $\{W = 10, K = 3, F = 1, TH = 0.001\}$.

4.2 Final Model: Linear Regression

According to 4.1, linear regression can lead to the best result in the testing set after comparing all the methods. To get a better result, consider changing the features in the linear regression model. Different feature choices will cause different strategy performances.

Feature Group	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
ChoiceEff	31.9771	1.6251	2.4829	-17.2123	99	1.7431	54.4715
Choice4	34.9644	1.7559	2.7154	-7.8315	119	1.8230	54.6748
Choice3	34.4681	1.7341	2.6913	-8.2871	120	1.8096	53.4553
Choice5	28.9482	1.4887	2.2981	-12.8502	95	1.6639	53.6585
Choice2	28.8767	1.4858	2.2728	-8.3278	119	1.6621	53.4553

Table 4-2-1 Backtesting Results for Linear Regression with Different Features

In feature classifying, only Choice5 and ChoiceEff include exchange rate indicators, which is extremely significant in predicting HSI future return. From an economic perspective, many companies listed on the HSI have significant international operations and trade globally.

Changes in exchange rates can impact the competitiveness, revenue, and profitability of these companies. What's more, Hong Kong is a major international financial center and has close economic ties with various countries. Exchange rate movements can reflect changes in global trade dynamics, economic conditions, and geopolitical factors. If exchange rate indicators were ignored during regression, the prediction results would be biased for a relatively long period.

As a result, although Choice4 and Choice3 have better Sharpe ratios and CAGR, there is a high probability of prediction failure in the future because of neglecting exchange rate indicators. To verify the outcome, deploy these models in data from November 2013 to October 2023. Models with features from Choice5 and ChoiceEff perform much better. Specifically, both of their cumulative returns reach 20 times, with Sharpe ratios of 1.57 and 1.51 respectively.

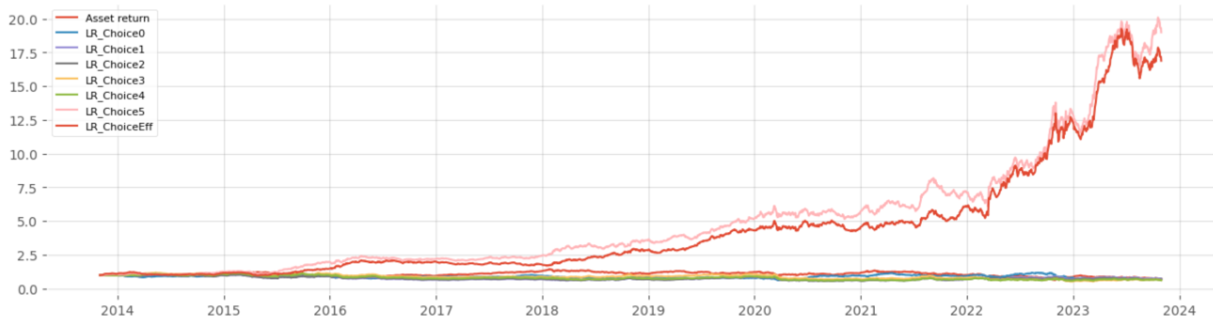

Figure 4-2-1 Linear Regression with Different Features in Deploy Set

Furthermore, considering that there may be slight changes in stock prices within a fluctuating range, such as from $100 to $100.1. In this case, although the stock price has risen, there is no obvious upward trend, which means that if investors buy at this time, there is only a very slim probability that it can bring positive returns to them. Set a threshold to optimize the strategy. Long the asset only when the increase in stock price is greater than the threshold. After fine tuning, the strategy beats the market well and performs better when setting thresholds as 0.12%, 0.13%, 0.14%, and 0.18%. Deploy the model in the deploy set, and the results under the features of Choice5 and ChoiceEff are shown below, and it is better than the original deployment.

Feature Group	Thres-hold (%)	CAGR (%)	Sharpe Ratio	Sortino Ratio	Max Drawdown (%)	Max Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
ChoiceEff	0.12	37.3668	1.9039	3.1759	-15.4786	687	23.9440	54.5241
	0.13	35.5867	1.8463	3.0647	-15.4786	687	21.0152	54.5894
	0.14	35.7875	1.8778	3.1141	-15.4786	687	21.3286	55.0995
	0.18	30.1927	1.7111	2.8363	-17.3076	692	14.0017	55.0683
Choice5	0.12	30.7481	1.6207	2.6135	-22.9378	647	14.6108	54.9941
	0.13	28.3977	1.5306	2.4670	-23.3552	653	12.1862	54.7472
	0.14	26.2745	1.4524	2.3321	-23.3552	653	10.3142	54.5629
	0.18	27.5912	1.5795	2.5729	-21.0715	425	11.4419	55.5311
Asset Return		-2.9994	-0.0504	-0.0713	-55.7008	2101	0.7374	51.3206

Table 4-2-2 Backtesting Results for Linear Regression with Different Features After Tuning

The final strategy uses linear regression to predict future returns, with Choice5 as input features and 12 as the threshold. The reason why neglecting ChoiceEff is that features in ChoiceEff conclude technical indicators, which may make strategy crowded in the market.

5 Discussion

5.1 Limitations

Despite assumptions made during testing, survivorship bias becomes a critical factor in the analysis. This bias is a challenge as we selectively pick important assets, making the study more complicated. It requires careful attention to limit its impact on the accuracy and reliability of the findings.

Equity market has changed rapidly due to the development of the past twenty years. It is not enough to use ten years of data to predict. Propose an assumption that when the training and testing periods are closer, the model fitted from the training set can predict accurately in the testing set. Therefore, based on the fundamental linear regression method, consider the rolling prediction method further to verify the assumption above. Derive the rolling prediction in two ways:

- Training set is fixed, but testing set is rolling

Use adjusted close prices of HSI between November 2001 and November 2011 as the fixed training set. Then, split the testing set into six groups, and the time horizon of each group is two years.

- Testing set is fixed, but training set is rolling

Starting from November 1, 2001, 2003, 2005, 2007, 2009, and 2011 respectively, the following 10 years of data were taken as 6 training sets for rolling training, while the testing set was from November 1, 2021, to November 1, 2023.

The result is not as good as expected. The rolling method just makes few improvements on strategy results. Nevertheless, if a fixed training and testing set is used, it is easy to generate survivorship bias in statistics because we will miss much information that could influence the predicted results. Thus, the rolling method is recommended in time series forecasting.

Survivorship bias occurs in the rolling method of linear regression. This is because, by assumption, the rolling model keeps updating the X features. Therefore, as it approaches the test window, the effectiveness of the train window is enhanced. However, in this project, the X features remain constant, the features selected for the model have remained consistent for 23 years. Therefore, survivorship bias is present in this scenario.

Yet, it's important to highlight the natural limitations in fine-tuning processes crucial to our methodology. Due to computer restrictions, adjusting Machine Learning (ML) parameters poses challenges in optimizing models for better performance. Similarly, refining features faces limitations imposed by computational resources, underscoring the need for careful decision-making in feature selection and engineering.

These limitations underscore the practical considerations within our methodology, acknowledging the delicate balance between our models' optimization goals and the practical constraints imposed by available computational resources. These factors play a crucial role in conducting a sophisticated and methodologically sound investigation aligned with the parameters of our research aims.

Balancing model optimization goals with the practical limits imposed by computing realities is crucial for a thorough and robust inquiry aligned with the research aims. In navigating these obstacles, the approach strives to find an informed balance, ensuring that the research remains methodologically robust and practically viable.

5.1.1 DTW

In DTW Batch strategy, it is noteworthy that in deployment testing, minor adjustments to hyperparameters can also lead to widely varying results. Strategies that have performed extremely well in the training dataset are also likely to show disadvantages in the future, whereas 'perfect strategies' implied in the future are difficult for DTW Batch strategy to uncover upfront.

For instance, we further discovered the hyperparameter combination with the highest return when deploying tests on data in the future decade, as indicated in Table 5-1-1.

Hyperparameter Combination	CAGR (%)	Sharpe Ratio	Sortino Ratio	Maximum Drawdown (%)	Maximum Drawdown (days)	Multiples of Invested Capital	Win Rates (%)
Asset Return	-2.9994	-0.0504	-0.0713	-55.7008	-2.9994	-0.0504	51.3206
{W=15, K=1, F=2, TH=10}	8.8413	0.5617	0.8225	-41.7549	986	8.8413	51.5196

Table 5-1-1 Performance of the Best Combination of Deployment Ten Years

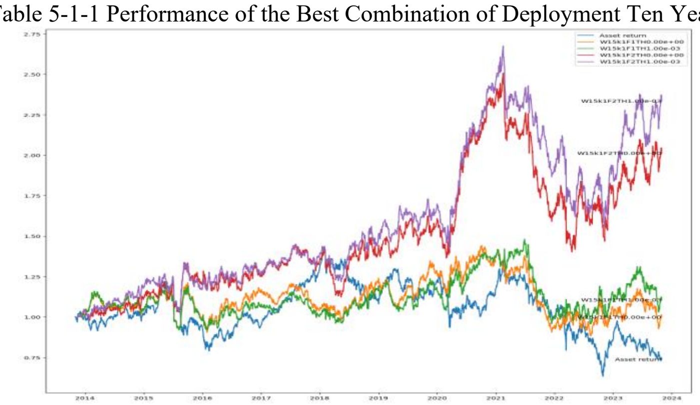

Figure 5-1-1 Comparison of More Combinations in Next Ten Years

The models in the Figure 5-1-1 all perform significantly better than those obtained during testing. This shows that when DTW is applied in a real environment, it is not able to dig deeper into the potential information buried in the database. Also, it is not possible to integrate and analyze external financial information, which is a limitation of the DTW algorithm. Some researchers have already developed optimization algorithms such as Derivative DTW, Indexing DTW based on DTW, which is the future research direction of this project.

Moreover, while FastDTW has been employed to diminish the time complexity, the computational workload remains a major predicament. In dealing with extensive datasets or frequent model runs, high computational complexity can result in inefficiencies that impede overall performance.

5.2 Improvements

In the ever-changing landscape of breakthroughs, rolling updates take centre stage, potentially shifting paradigms to enhance dynamism in our analytical framework. This method aligns models seamlessly with evolving data, fostering flexibility in response to changing information. Positioned strategically, rolling updates aim to elevate the quality and relevance of investigations, leading in temporal responsiveness.

An essential part of our improvement involves meticulous fine-tuning of thresholds in our models, emphasizing the delicate balance between precision and recall. The goal is to optimize overall performance and enhance the durability of our algorithms. Rigorous recalibration is expected to noticeably improve our models' ability to navigate different contexts with increased accuracy.

In methodological advancements, particularly in Dynamic Time Warping (DTW), we encourage additional parameter fine-tuning. Employing search techniques like grid or stochastic searches enhances model parameters, promoting robustness across varying time periods and market conditions.

Additionally, exploring variations in thresholds for Euclidean distance in DTW can influence model sensitivity. Achieving a balanced model that captures sufficient detail while avoiding noise over-sensitivity is crucial. Deepening past data analysis by partitioning into different trends and periods aids the model in comprehending and adjusting to diverse market states.

Moreover, combining DTW with Principal Component Analysis (PCA) for data mining enhances accuracy and uncovers nuanced insights in the stock market. This strategic evolution marks our methodology as intelligent, agile, and responsive to the needs of modern analytical challenges.

6 Conclusion

To sum up, machine learning or deep learning in predicting stock index prices has gained popularity in recent years. Although there are drawbacks in machine learning, such as its inability to forecast significant economic changes like the 2008 financial crisis and the 2019 pandemic, it remains useful for helping firms forecast prices.

Linear regression stands out as the most stable method, a conclusion derived from a thorough analysis of backtesting results. Notably, when Choice5 was used as input features with a threshold set at 0.12%, it yielded a significantly higher return of 14.6108 times the initial invested amount during the 10-year deployment test, compared to the HSI's return of only 0.7374. These results, highlighting the consistency and reliability of linear regression, demonstrate its minimal maximum drawdown compared to other strategies considered. This stability positions linear regression as a robust and preferred approach for predicting stock prices.

Selecting a stable trading strategy is crucial. Consequently, machine learning models that involve a random state, such as decision trees and random forests, are generally less stable compared to the reliability of linear regression. Additionally, non-parametric machine learning methods like KNN and support vector machines operate with a lazy training approach. Unlike linear regression, these models may struggle to capture a comprehensive set of features in machine learning.