Portfolio Allocation with Time Series and Machine Learning Models

Xinyue Liao

Leonard N. Stern School of Business, New York University, New York, USA

XL2894@nyu.edu

Abstract. The historical average stock return is commonly used to predict expected stock return in portfolio management for its better prediction and more straightforward calculations. However, most regressive models empirically proved to underperform the historical average return are explanatory models using predictor variables. This article takes the challenge to further explore the regressive power in predicting stock return but focuses on time series (TS) forecasting models and machine learning (ML) models. The results of this article show that many of these TS and ML regression models beat the historical average return, delivering higher portfolio Sharpe ratios and realized returns in backtesting. Although TS and ML models have inherently weaker explanatory power for their predictions, these results are still meaningful for retail and institutional mean-variance investors, opening up a new angle to forecast expected returns in portfolio allocation.

Keywords: Equity Portfolio Management, Machine Learning, Gradient-Boosting, KNN, Walk-forward Analysis.

1. Introduction

The question of how to construct an equity portfolio that offers the maximum Sharpe ratio is at the heart of many mean-variance investors. Mathematically, maximizing the portfolio Sharpe ratio is easily doable given the expected return of each stock and their covariance matrix. However, whether and how expected returns can be predicted remains a mystery.

In fact, the predictability of stock return has been controversial in the financial world for years. In the 1980s, researchers started using valuation multiples such as price-earnings and price-dividend ratios to predict stock return. For example, around 1984 to 1988, authors including Fama and French, Rozeff, and Campbell and Shiller, proposed that many valuation multiples and stock returns are positively correlated, implying the predictive power of multiples on stock returns [1-4]. In the 1990s and 2000s, research continued to explore more possible predictor variables, with some articles suggesting that macroeconomic factors such as treasury bond rate and consumption level would help predicting returns. Since 2000, many of these formal findings have been criticized to the point that these predictors perform poorly in out-of-sample predictions [5-7]. Several articles have compared explanatory regression models with the historical average return model in out-of-sample prediction and found that historical average return almost persistently outperforms regression models. Recently, with the increasing abundance of data and the improving computing power, it seems that merely using averages doesn't fully exploit the information hidden in the data set and potentially leaves money on the table. Researchers have started to study the power of machine learning in the financial industry and introduced a number of machine learning-based prediction methods [8-9].

Given this trend, this article attempts to use time series and machine learning models to predict stock returns, even in the post-Covid world, where the market is volatile, and the predictability of expected stock return is relatively more elusive [10]. This article is structured as follows. Section 2 reviews data used in stock return prediction and portfolio performance backtesting. Section 3 explains predicting expected stock return, portfolio construction, and backtesting methods. Section 4 presents portfolios' backtested results including Sharpe ratio and realized return using each forecasting method. Section 5 extends the discussion and closes the article with the conclusion.

2. Data

2.1. Selected stocks

This article selects the 10 largest components of the S&P 500 index by weight. The S&P 500 is a market-capitalization weighted index of 500 large-cap U.S. listed stocks, accounting for 70% to 80% of total U.S. equity market value, making it one of the most representative U.S. equity market indices. Meanwhile, the top 10 components of the S&P 500 accounted for 27.8% of the index's market capitalization as of Aug 31. 2022, and thus they are strong drivers in the market. The details of selected stocks are shown in the following table 1.

Table 1. Selected Stock Information (as of Sep 5th, 2022)

Symbol	Company Name	Index Weighting (%)	Price
AAPL	Apple Inc.	7.260	155.81
MSFT	Microsoft Corporation	5.830	256.06
AMZN	Amazon.com Inc.	3.350	127.51
TSLA	Tesla Inc	2.090	270.21
GOOGL	Alphabet Inc. Class A	1.970	107.85
GOOG	Alphabet Inc. Class C	1.820	108.68
BRK.B	Berkshire Hathaway Inc. Class B	1.530	277.67
UNH	UnitedHealth Group Incorporated	1.470	516.35
JNJ	Johnson & Johnson	1.300	162.74
XOM	Exxon Mobil Corporation	1.180	95.59

2.2. Raw data collection and pre-processing

The raw data used in this article is the daily adjusted closing prices of selected stocks from September 1st, 2017, to August 31st, 2022, collected from Yahoo Finance. Essentially, prices of stocks are non-stationary time series, but stationarity is required for time series analysis. To overcome this problem, daily linear returns of selected stocks are calculated, converting a non-stationary to a stationary time series. This article focuses on return data instead of price data for forecasting and portfolio construction [11]. The table 2 below shows the descriptive statistics of selected stocks' returns during the period of interest.

Table 2. Stock Daily Return Descriptive Statistics (from Sep. 1st, 2017, to Aug. 31st, 2022)

	mean	std	min	max	skewness	kurtosis
AAPL	0.13%	2.03%	-12.86%	11.98%	-0.098413	5.08556
AMZN	0.10%	2.17%	-14.05%	13.54%	0.107254	5.189365
BRK-B	0.05%	1.42%	-9.59%	11.61%	-0.079667	11.104401
GOOG	0.09%	1.88%	-11.10%	10.45%	-0.000685	4.178626
GOOGL	0.08%	1.89%	-11.63%	9.62%	-0.04339	4.05498
JNJ	0.04%	1.30%	-10.04%	8.00%	-0.327079	9.518065
MSFT	0.12%	1.88%	-14.74%	14.22%	-0.031172	7.757234
TSLA	0.28%	4.03%	-21.06%	19.89%	0.230222	4.003095
UNH	0.10%	1.86%	-17.28%	12.80%	-0.05692	12.811167
XOM	0.06%	2.08%	-12.22%	12.69%	0.033267	5.711068

3. Method

3.1. Expected value of stock return

Statistically speaking, the expected value of an event is its outcome on average if repeated infinitely many times. If the return r_i on a stock i is equal to $r_i(s)$ with probability $p(s)$ for an

event $s = 1,..,S$ to occur, the expected return of stock i can be expressed as the probability-weighted average of all possible outcomes:

$$E[r_i] = \sum_{s=1}^{S} p(s) r_i(s) \tag{1}$$

3.2. Historical average return

The historical average return model assumes that a stock's future performance is the same as its past performance on average, and each historical return is equally likely to re-occur next period:

$$p(s) = \frac{1}{S} \text{for all } s = 1,..,S \tag{2}$$

While this assumption is usually too strong to be true, the historical average return model is widely used due to its simplicity and stability. Moreover, historical average returns often provide better predictions than other sophisticated forecasting models, so it is usually considered a default benchmark. Building onto the expected value model, the historical average return is expressed as below:

$$E[r] = \bar{r} = \frac{1}{T} \sum_{t=1}^{T} r_t \tag{3}$$

Where r_t stands for the historical return of the stock at time t.

3.3. Autoregressive model (AR)

An Autoregressive model (AR) is a classical time series model in forecasting. An AR model relies on the autocorrelation of the stochastic process, capturing the effect and predictive power of past values on current values.

An AR model is a special kind of multiple linear regression, where the predictors are lagged values of the time series of interest itself. Thus, an AR model of order p, denoted as $AR(p)$, can be written as:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t \tag{4}$$

Where $\phi_1, \ldots, \phi_p$ are coefficients and ε_t is white noise.

In theory, an AR model can include an infinite number of explanatory variables, but a high-order AR model is often inefficient and misleading because the ACF (Auto Correlation Function) of a stationary time series usually deprecates to zero quickly, meaning that there is little correlation between time series y_t and y_{t-p} if p is large. As a result, a high-order AR model usually overfits the training set and delivers very low predictability for the testing set. Therefore, this article only applies AR(1) model and AR(2) model to forecast expected stock returns:

$$\text{AR}(1): y_t = \phi_0 + \phi_1 y_{t-1} + \varepsilon_t \tag{5}$$

And

$$\text{AR}(2): y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t \tag{6}$$

3.4. Modified cross-sectional regression

This article uses a modified cross-sectional regression, in which one stock's return is regressed against one-period-lagged returns of all selected stocks, including itself, and this model can be expressed in the form of a multivariable linear regression:

$$y_{i,t} = \alpha + \beta_1 x_{1,t-1} + \beta_2 x_{2,t-1} + \ldots + \beta_n x_{n,t-1} + \varepsilon_t \tag{7}$$

Where y_i is the stock return of interest, $x_{1,t-1}, \ldots, x_{n,t-1}$ are one-period-lagged returns of all selected stocks, $\beta_1, \ldots, \beta_n$ are coefficients and ε_t is white noise. This model uses OLS to estimate parameters. The cost function and optimization objective for OLS is:

$$J(\vec{\beta},\alpha)=\frac{1}{2m}\sum_{i=0}^{m}(f(\vec{x}_{i,t-1})-y_{i,t})^2 \tag{8}$$

$$\min_{\vec{\beta},\alpha} J(\vec{\beta},\alpha) \tag{9}$$

Where m is the size of the dataset used in regression.

Loosely speaking, the modified cross-sectional regression model can be viewed as an extension of the AR(1) model, using past returns of all selected stocks to explain the current return of one stock. In theory, the modified cross-sectional regression model would have a higher R-square than the AR(1) model for explaining the same variable of interest.

3.5. Lasso regression

LASSO, standing for Least Absolute Shrinkage and Selection Operator, is a regularization technique in statistics and machine learning. It reduces overfitting in a complex model with many predictors and helps mitigate the problem of multicollinearity. LASSO regression automates the variable selection process in multiple linear models by penalizing large coefficients in absolute values [12].

In the modified cross-sectional regression model, predictors are exposed to the same market risk and correlated. Thus, applying LASSO regularization to the modified cross-sectional regression model would reduce the probability of overfitting and improve the model's performance. Under LASSO regression, the forecasting model remains:

$$y_{i,t}=\alpha+\beta_1 x_{1,t-1}+\beta_2 x_{2,t-1}+\ldots+\beta_n x_{n,t-1}+\varepsilon_t \tag{10}$$

But the cost function and optimization objective changes to:

$$J(\vec{\beta},\alpha)=\frac{1}{2m}\sum_{i=0}^{m}(f(\vec{x}_{i,t-1})-y_{i,t})^2+\lambda\sum_{j=1}^{n}|\beta_j| \tag{11}$$

$$\min_{\vec{\beta},\alpha} J(\vec{\beta},\alpha) \tag{12}$$

Where λ is the penalty factor.

3.6. Gradient boosting

Gradient Boosting (GB) is an ensemble learning model in which multiple weak prediction models or learners are built sequentially. Each subsequent model tries to explain the residuals left over by the previous model and is combined to form a robust prediction model. This article wants to use GB to explain the return of stock i at time t using returns of selected stocks at time $t-1$. The first weak learner of the GB is a linear regression model such that:

$$F_1: y_i=\alpha_1+\beta_1 x_1+\varepsilon_1 \tag{13}$$

Where y_i is stock i's return, x_1 is one-period lagged stock returns whose absolute value of correlation with y_i is the highest among all selected stocks and ε_1 is the residual. Next, the second weak learner of the GB is another similar linear regression model such that:

$$F_2: \varepsilon_1=\alpha_2+\beta_2 x_2+\varepsilon_2 \tag{14}$$

Where ε_1 is the residuals of first learner, x_2 is one-period lagged stock returns whose absolute value of correlation with ε_1 is the highest among all selected stocks and ε_2 is the residual.

In general, the $m_{th}(m>1)$ weak learner of the GB is a linear regression model in the form:

$$F_m: \varepsilon_{m-1}=\alpha_m+\beta_m x_m+\varepsilon_m \tag{15}$$

Where ε_{m-1} is the residuals of $(m-1)_{th}$ learner, x_m is one-period lagged stock returns whose absolute value of correlation with ε_{m-1} is the highest among all selected stocks and ε_m is the residual. As a result, a GB model with M stages is:

$$y_i = \alpha_1 + \alpha_2 + \ldots + \alpha_M + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_M x_M + \varepsilon_M \quad (16)$$

3.7. Self-implemented KNN

KNN, or K-Nearest Neighbors, is a supervised machine learning algorithm that uses K nearest neighbors for prediction, assuming that similar things exist within close distance. To predict $r_{i,t}$, the expected stock i return on Day t, KNN computes the average performance of that stock on the second day of the day when the performance of selected stocks is similar to that on Day $t-1$. The steps of KNN algorithm are illustrated as below:

(1) Calculate distance between $[r_{1,t-1}, r_{2,t-1}, \ldots, r_{10,t-1}]$, the performance of selected stocks on Day $t-1$, and $[r_{1,n}, r_{2,n}, \ldots, r_{10,n}]$, the historical performance of selected stocks on Day n, where $t-1-w \leq n \leq t-2$, and w is the number of historical data considered;

(2) Sort $[r_{1,n}, r_{2,n}, \ldots, r_{10,n}], t-1-w \leq n \leq t-2$ based on their distance from $[r_{1,t-1}, r_{2,t-1}, \ldots, r_{10,t-1}]$ from the nearest to the furthest;

(3) Select the first K nearest $[r_{1,n}, r_{2,n}, \ldots, r_{10,n}], t-1-w \leq n \leq t-2$ neighbors of $[r_{1,t-1}, r_{2,t-1}, \ldots, r_{10,t-1}]$, and label them as $V_k = [r_{1,k}, r_{2,k}, \ldots, r_{10,k}], 1 \leq k \leq K$;

(4) On Day t, the expected return of stock i is calculated as: $\mathrm{E}(r_{i,t}) = \frac{\sum_{k=1}^{K} r_{i,k+1}}{K}$

In this article, KNN algorithm uses two types of distance metrics to measure similarity between two vectors $X = [x_1, x_2, \ldots x_n]$ and $Y = [y_1, y_2, \ldots y_n]$:

(1) Pearson Correlation Coefficient:

$$Corr\ Coef. = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (17)$$

(2) Cosine distance:

$$Cosine\ dis. = cos(\theta) = \frac{X \cdot Y}{|X||Y|} = \frac{\sum x_i y_i}{\sqrt{\sum {x_i}^2}\sqrt{\sum {y_i}^2}} \quad (18)$$

$X\ and\ Y$ are closer with either higher correlation coefficient or higher cosine distance.

3.8. Maximum sharpe ratio portfolio

This article considers the maximum raw Sharpe ratio portfolio of selected stocks. The portfolio raw Sharpe ratio equals to the portfolio return divided by volatility:

$$raw\ Sharpe\ ratio\ = \frac{r_p}{\sigma_p} \quad (19)$$

Suppose the expected returns and the weight allocation of selected stocks are

$$R = [r_1, r_2, \ldots, r_{10}] \quad (20)$$

$$W = [w_1, w_2, \ldots, w_{10}] \quad (21)$$

Then the objective function in finding the weight allocation of a maximum Sharpe ratio portfolio is:

$$max\ \frac{\sum w_i r_i}{\sqrt{\sum_i \sum_j w_i w_j \sigma_{ij}}} \quad (22)$$

Subject to

$$\sum w_i = 1 \quad (23)$$

$$-1 \leq w_i \leq 1 \quad (24)$$

3.9. Walk-forward analysis

In walk-forward analysis, the historical dataset is divided into two portions dynamically: in-sample data, or training data, and out-of-sample data, or testing data, where the latter is a period after the in-sample data. The in-sample data rolls forward to find optimal trading systems at different points in time, while the out-of-sample data is used to test systems. In this article, the first out-of-sample data starts on September 1st, 2021, and ends on August 31st, 2022, and the in-sample data is always the five-year data before the out-of-sample data (Details are shown in the following Figure 1).

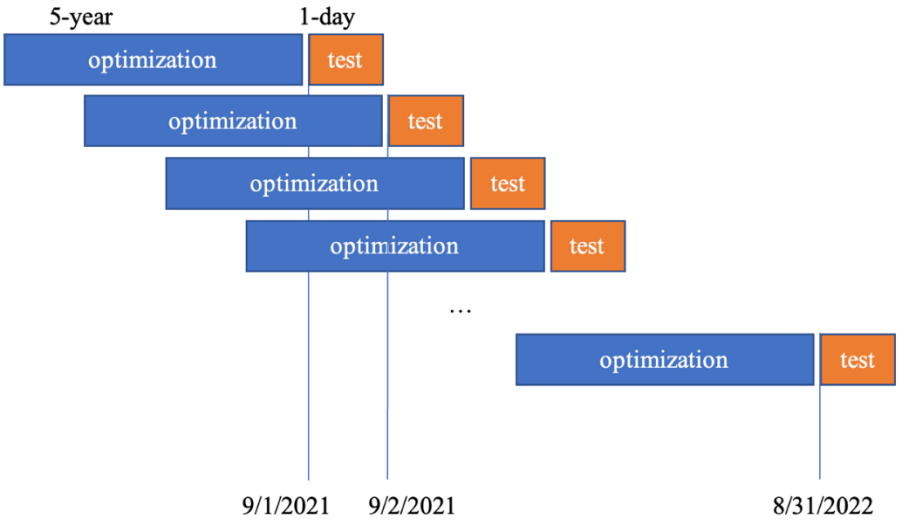

Figure 1. Illustration of the Walk-Forward Analysis

In general, this article uses daily returns, but expected stock returns, covariances, and Sharpe ratios are annualized, assuming 252 trading days a year.

4. Results

For each expected stock return predicting model, the backtested results are as follows in Table 3.

Table 3. Backtested Sharpe ratios and Realized Returns Using Different Predictive Methods

Predictive Model	Raw Sharpe Ratio (annualized)	*Realized Return (annualized)
Historical avg. return	-0.306	-18.578%
AR(1)	0.141	-2.493%
AR(2)	-0.118	-12.369%
Cross-sectional regression	1.073	47.523%
LASSO regression ($\lambda = 1 \times 10^{-5}$)	0.758	27.448%
LASSO regression ($\lambda = 1 \times 10^{-6}$)	1.116	50.293%
LASSO regression ($\lambda = 1 \times 10^{-7}$)	1.153	53.094%
GBM (M=2)	0.440	10.173%
GBM (M=3)	0.741	26.353%
GBM (M=4)	0.718	25.697%
KNN (corr coef.)	-0.720	-33.267%
KNN (cosin dis.)	0.384	7.462%
KNN (Euclidean dis.)	0.001	-7.018%

*Realized Return: calculated by chaining all daily returns together

Among all strategies, LASSO regression with $\lambda = 1 \times 10^{-7}$ yields the highest Sharpe ratio and realized returns. However, since both LASSO regressions with $\lambda = 1 \times 10^{-7}$ and $\lambda = 1 \times 10^{-6}$ show decent performance with tiny discrepancy, it is inconclusive that $\lambda = 1 \times 10^{-7}$ is better than $\lambda = 1 \times 10^{-6}$ in the LASSO regression in this case. It is also worth noting that the Sharpe ratio and

realized return obtained with the modified cross-sectional regression model are close to those obtained with LASSO regression, especially when λ is small. This corresponds with the fact that LASSO regression is the same as Cross-sectional regression when $\lambda = 0$ (no regularization applied).

Considering AR models, AR(1) fits fewer explanatory variables but outperforms AR(2), implying that AR(2) might be overfitting, better-fitting the training data but delivering higher variance and lower predictability for the testing data set. Likewise, the GB ($M = 3$) outperforms the GB ($M = 4$) indicating the GB ($M = 4$) may be overfitting. On the other hand, however, the GB ($M = 3$) does outperform the GB ($M = 2$), demonstrating that GB ($M = 2$) might be underfitting.

Using different distance metrics in KNN models leads to different performance results, inferring the characteristics of the collected financial data itself. For example, KNN using the cosine distance metric is better than KNN using the correlation coefficient distance metric, implying that the average return of selected stocks is important in predicting one stock's return. A simple example illustrating this idea might be that the correlation coefficient between $[1, 2]$ and $[2, 4]$ is the same as the correlation coefficient between $[1, 2]$ and $[3, 6]$, but with different cosine distances. It makes sense since selected stocks account for about 27.8% of the value of S&P 500 which makes up more than 70% of the total U.S. equity market value. Thus, the average return of selected stocks should largely drag the average return of the total market, thus revealing the returns of an individual stock.

As a benchmark, holding S&P 500 index over the same testing period (September 1st, 2021 to August 31st, 2022) yields the following result in Table 4:

Table 4. Sharpe ratios and Realized Returns of Holding S&P 500

	Sharpe Ratio (annualized)	Realized Return (annualized)
Holding 100% S&P 500 index fund	-0.497	-11.863%

In conclusion, most regression-based strategies yield a higher Sharpe ratio and a higher realized return than the passive strategy of holding a 100% S&P 500 index fund and significantly beating the historical average return.

5. Conclusion

In summary, this article investigates the potential of the time series forecasting model and machine learning model in predicting expected stock returns and compares their predictability with the historical average stock return. First, this article selects the top 10 stocks in S&P 500 index by weight and uses autoregression, modified cross-sectional regression, Lasso regression, KNN, and Gradient-Boosting models to predict expected stock returns. Then, this article constructs maximum Sharpe ratio portfolios and backtests their performance using walk-forward analysis. As the result shows, the time series and machine learning models outperform the historical average stock return and beat the market in the selected testing time frame. Mean-variance investors who currently use historical average stock return to predict expected stock return for stock trading and investment may benefit from this result.

While the time series and machine learning models perform decently, investors should recognize that they provide less explanatory power and are more complex to implement than the historical average return or explanatory regressions. However, the predictive power of time series forecasting and machine learning models on expected stock return shown in this article is likely to be systemic across time and deserves attention from investors and researchers.

References

[1] Fama, E. F., K. R. French. Dividend Yields and Expected Stock Returns. Journal of Financial Economics 1988, 22: 3-25.

[2] Rozeff, M. S. Dividend Yields are Equity Risk Premiums. Journal of Portfolio Management 1984, 11(1): 68-75.

[3] Campbell, J. Y., R. J. Shiller. The Dividend-Price Ratio and Expectations of Future Dividends and Discount Factors. The Review of Financial Studies 1988, 1: 195-228.

[4] Campbell, J. Y., R.J. Shiller. Stock Prices, Earnings, and Expected Dividends, Journal of Finance 1988, 43: 661-76.

[5] Goyal, A., I. Welch. Predicting the Equity Premium with Dividend Ratios. Management Science, 2003, 49: 639-54.

[6] Goyal, A., I. Welch. A Comprehensive Look at the Empirical Performance of Equity Premium Predication. The Review of Financial Studies, 2007.

[7] Bulter, A. W., G. Grullon, J. P. Weston. Can Managers Forecast Aggregate Market Returns? Journal of Finance, 2005, 6-: 963-86.

[8] Ke, Zheng Kelly, Bryan T., Xiu, Dacheng, Predicting Returns with Text Data, 2020.

[9] Rasekhschaffe, Keywan Jones, Robert. Financial Analysts Journal. Machine Learning for Stock Selection, 2019.

[10] Chowdhury, Emon Kalyan, Chowdhury, Emon Kalyan, Abedin, Mohammad Zoynul, COVID-19 Effects on the US Stock Index Returns: An Event Study Approach, 2020.

[11] Meucci, Attilio, Exercises in Advanced Risk and Portfolio Management (ARPM) with Solutions and Code, 2010.

[12] Liu, Yang, Zhou, Guofu, Zhu, Yingzi, Maximizing the Sharpe Ratio: A Genetic Programming Approach, 2020.