Portfolio Management via Empirical Asset Pricing Powered by Machine Learning

Vittorio Carlei[1(✉)], Stefano Terzi[2], Federico Giordani[2], and Gaetano Adamo[2]

[1] University of Chieti-Pescara, Pescara 65122, Italy
carlei@unich.it
[2] Qi4M, Rome 00144, Italy
{terzi,giordani,adamo}@qi4m.com

Abstract. In this paper we propose an alternative machine learning forecasting technique for the canonical problem of predicting expected stock returns. The final goal is enhancing the financial performance of the investment product, which in our case refers to a portfolio of equities. We adopt a combination of algorithms, capable of hand-ling high-level abstraction, to study short- and long-term patterns emerging from the analysis of financial factors and market signals. The core of the model adopted to perform the prediction is composed of two independent entities, analyzing short-term dynamics and capturing long-term trends respectively. This adjustment helps us improve the predictive ability of the model in a dynamic environment, where high volatility and noise are intrinsic features. Lastly, we employ an ensemble algorithm that performs an intelligent weighting of each agent's output. This method allows us to identify the best stocks in terms of performance and to successfully implement quarter-long hold strategies that outperform the selected universe's equities return benchmark.

Keywords: Portfolio management · Asset pricing · Artificial intelligence · Machine learning

1 Introduction

In this paper we outline the methodology that guides the portfolio management process at Qi4M. We perform the analysis focusing on the problem of predicting expected returns. As a result, we obtain a model further used to develop an investment strategy that selects the equities to be included into a periodic portfolio. We subsequently present the reader with an empirical study, which is supported by historical pro-forma results.

The literature on empirical asset pricing maintains its stable influence on shaping asset managers' approach to investing. As a result, many continue to handpick the methodology that best fits their economic and market views, often

E. Bucciarelli et al. (Eds.): DECON 2020, SCI 990, pp. 121–129, 2021.
https://doi.org/10.1007/978-3-030-75583-6_12

prioritizing a restricted number of signals and factors for a set period of time prior to the examination of other possible approaches.

This became the essential reason behind our aim to derive a highly flexible Artificial Intelligence (AI) model that is adaptable to different market conditions. Besides, such model does not require any modifications of its inner workings and allows to avoid arbitrary factors selection at each future period. The nature of the problem we are tackling suggests the need for a Deep Learning algorithm employment to capture linear and non-linear relations in the data.

In view of modern financial theory, the undertaken approach falls under the Arbitrage Pricing Theory (APT), mathematically expressed by the following linear relation:

$$E(r_j) = r_f + \sum_{i=1}^{N} \beta_i R_j \tag{1}$$

where:

- $E(r_j)$ is the expected return of asset j,
- r_f is the risk-free rate of return,
- β_i is the view on asset j,
- R_i is the risk-premium by factor.

One of the key characteristics of the APT model is the accurate selection of financial factors that generate the risk-premia [1,2]. In order to optimize portfolio performance, it is necessary to switch to the best predictive factor, with regards to returns, at the right point in time. For instance, an investment strategy strongly focused on *momentum factors*, could produce higher returns if correctly altered by a *value factors* strategy, given that the strategy based on *value* outperforms the one based on *momentum* in the relevant observation period.

In addition, we assume a non-linear relation between returns, financial factors, market, and time in a point-based market scenario:

$$E(r_{j,t+1}|\mathcal{F}_t) = \psi(t, \beta_{i,j}, U) \tag{2}$$

where:

- $E(r_{j,t+1}|\mathcal{F}_t)$ is the expected return of asset$j$ at time *t+1* conditional on the information set (the filtration $\mathcal{F}_t$), formed by market participants, up until time t,
- t is the time,
- $\beta_{i,j}$ is the view on the i-th factor relative to asset j,
- U is the View on the market

Although it is initially important to perform an educated selection of factors, all successive factor switches are handled by the model itself, which eliminates the need for further arbitrary inputs. The reader will be later presented with a complete evolution of factor choices in our pro-forma portfolio.

1.1 Machine Learning Application to Asset Pricing

When it comes to asset pricing research, one can evidence the extensive application of Machine Learning methods. For instance, time series asset pricing theory revolves around the ability to explain a satisfactory level of variability in a stock's future returns. This is a fundamental problem of prediction that in large has always been characterized by the ability of a researcher to select the most relevant predictors for a specific asset or time period that is being subject to analysis.

Machine learning provides an optimal solution to such problem. One can avoid factor selection, in favour of adopting a reliable algorithm that is able to efficiently learn which factors have to be selected, and which ones have to be dropped. The prediction problem then becomes selecting the right algorithm, as opposed to the right predictors. However, it must be noted that our research and experience do not exhibit satisfactory results of such approach. Nonetheless, even at this naive level, it can be easily noted how the problem has moved from the asset pricing theory domain into the one of computational efficiency.

Efficiency is a key term to bear in mind when approaching problems in the machine learning domain, as it makes these methods well suited for asset pricing applications. In fact, classical or more intricate econometric models that analyse expected returns require an explicit presentation of a functional form that is often not exemplified by theory. On the other hand, neural networks handle this type of ambiguity well, as they extrapolate the form directly from the data provided.

Moreover, the theoretical literature on APT offers limited guidance for dealing with model specifications. It is therefore core to the analysis to perform an educated selection of the financial factors to be included in the data set. In our case, the factors selection process is deeply rooted in the financial economics research. We do, however, leave it to the algorithm to learn which of these factors must be included for producing the best prediction at any given point in time.

Three aspects of machine learning are extremely useful for handling such data. First, the flexibility, the ability to connect linear to non-linear models through ensemble functions, along with the diversity of such functions add a new level of control to the prediction problem. Second, neural networks are specifically designed to approximate complex nonlinear associations - non-linearities that often, through predictors interaction, end up characterizing the prediction altogether. Finally, they provide a high-level of control in avoiding overfit biases and false discovery.

1.2 Theoretical Foundation

This research relies heavily on two main theoretical foundations, one referring to machine learning, and the other to financial economics [3–5]. We will follow with a brief description of both to show the reader how the first comes to serve and enhance the second.

The main characteristic of machine learning is that the rules governing the algorithms' functioning do not need to be explicitly coded. The model discovers rules on its own, looking at the *training set*, to then generalize the rule that will later drive its output construction. To perform accurate predictions in the current financial market environment, we believe that the algorithm must be able to analyse both short-term and long-term trends.

The research by Fama-French shows that return performance based on the analysis of *financial factors* is period dependent [6]. We, therefore, classify factors into two sets; the first set is composed of factors that are related to stock return's behaviour, while the second set lists factors that can be adopted only in specific market scenarios and do not have much explanatory power otherwise. Here a feature selection problem emerges. The latter is not easily solvable, especially when it comes to selecting factors that are strongly scenario dependent.

The consequence of these findings is the adoption of two different *training* methodologies, supported by the creation of two different *training sets*. The ultimate result is the adoption of two prediction models.

2 Methodology

We begin by defining an investment universe that makes up the model's input. We consider highly liquid stocks that would allow us to interact on the market without price or volume frictions. However, it is important to note that the model is highly flexible with respect to which investment universe is selected, it can easily span across regions and industries without showing statistically significant differences in overall returns. As previously stated, the initial input selection of the model has been influenced by an arbitrary selection on our part, which is a product of conjunct research efforts. We will keep the description of such selection terse.

From this step, the model undergoes the training and testing phases, employing methodologies that will be presented in this section. The ultimate output of the model is an investment strategy in the form of a portfolio of equities, which repeatedly overperforms the universe's benchmark.

We will follow with a description of the financial theory that influenced the construction of the model. We avoid presenting the reader with a profound description of the relevant models. We invite the reader to refer back to the bibliography, to gain a more thorough understanding of the topics mentioned in the discussion.

2.1 Sample Splitting

The preliminary step in the formulation of our approach is to understand how to best design sub-samples for estimation and testing[1]. This process starts from

[1] Note that when constructing the *training set and target set*, you incur two phenomena: missing data and noisy data. To overcome the problem of missing data, we apply the method suggested by Beaver et al. [7]. At the same time, we adopt the approach proposed by Steege et al. [8] to handle the noisy data.

setting a time-frame for the characteristic *walk-forward* we employ in the creation of both models' training and testing.

The two methodologies cited before refer to two different methods we apply to perform the *walk-forward*. We essentially define the *steps* that constitute the *walk-forward*. Here, *steps* indicates the event of a new information becoming available to the market. In our case, the *steps* indicate the update of a given company's set of selected fundamentals. With respect to the aforementioned *steps*, we can differentiate the two methods that from now on will be referred to as 'Veloce' and 'Lento'. We then use this set to predict stocks' returns at a future point in time.

The first method, 'Veloce', creates a fixed *steps-size* train set using the last 'few' *n steps* available in the information set. Once new information becomes available it moves the rolling window forward to create a new train set that drops the first *step* and picks up the last. On the other hand, the second method, 'Lento', updates the training set by increasing its size to include new information available at the new *step* (Figs. 1 and 2).

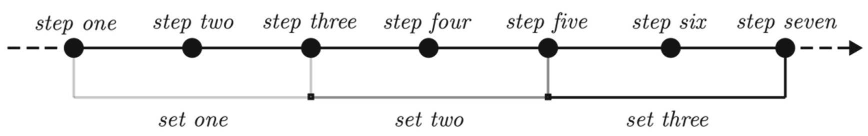

Fig. 1. Step size. The figure shows a fixed-length step in time. In this example the step length has been fixed to n = 2.

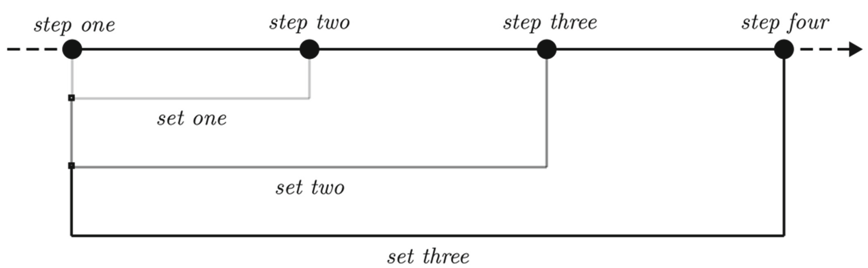

Fig. 2. Step incremental

2.2 Prediction

To perform the prediction of the stock returns we adopt two regressors: a feed-forward multi-layer perceptron, and a multivariate linear regressor, which are then used to minimize the same objective function, the mean squared prediction errors (MSE). The multi-layer perceptron's training phase handles the dynamics to be picked up by the 'Lento' method, while those of 'Veloce' are better handled by the multivariate regressor.

3 Empirical Study of MSCI World

The empirical study has been carried out by first constructing an indicative universe of investible equities. The equities have been selected according to a set of criteria. First of all, the company has to be registered in a region part of the MSCI World. Subsequently to the criteria satisfaction, we apply a selection flag to categorize equities according to their daily volatility features, such that the average daily liquidity for the past three months is at least five million. Our sample begins in Q1 2003 and ends Q4 2017. In addition, we build a large set of stock-level factors based on existing literature and proprietary research.

As inferred from the historical performance of this all-equity strategy, it outperforms the universe's benchmark even in periods of high volatility. The following graphs show the portfolio performance by period, along with the annualized return in comparison with the benchmark.

The yearly performance graph shows returns of Qi4M's factors-based strategy compared with the MSCI benchmark (Fig. 3).

Fig. 3. Yearly portfolio performance

The strategy repeatedly overperforms the MSCI, and does not employ stop-loss mechanisms, exposing the returns to market falls. However, the strategy is still able to outperform the benchmark by more than 12% on an annualized basis.

The following graphs give an interesting insight of the regions and sectors that the strategy is exposed to. Furthermore, they provide a broad understanding of the logic behind it. For instance, given the benchmark selection criteria - high liquidity - the most dominant region of exposure is North America, as expected. This level of high liquidity allows the algorithm to create a pro-forma portfolio with realistic historical returns.

Sectors breakdown indicates the strategy to be highly exposed to the consumer and industrials sectors on average, with a major reduction in the consumer sector position during the 2007-08 crisis, in favour of an increased exposure in the industrials sector (Fig. 4).

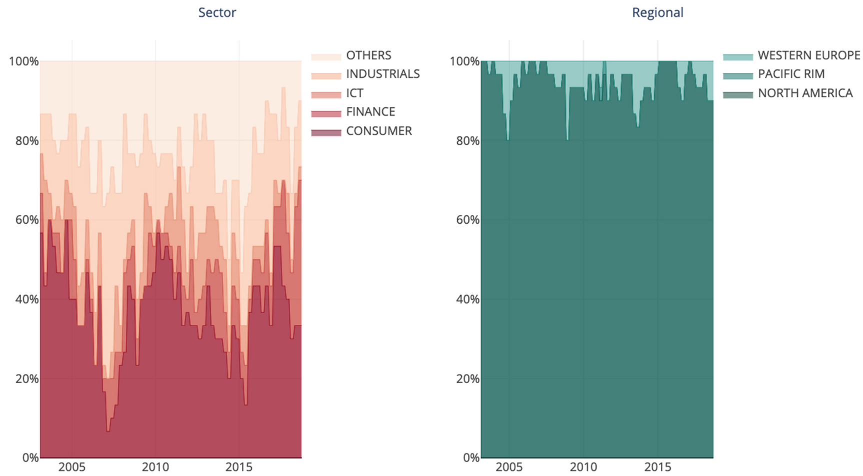

Fig. 4. Sector and region exposure

The following graph shows the pro-forma cumulated return generated from a $10.000 investment in Q1-2003, holding the position up to Q4-2017. The selection criteria is particularly important; in fact, we are able to select an equities benchmark that on its own is able to perform close to double the overall return from the universe. The algorithm further improves this result, overperforming both the benchmark and the MSCI universe. The annualized return generated by the algorithm is of 20.61%, compared to a 14.21% return from the selected benchmark, and 7.97% return from the universe (Fig. 5).

As shown in Fig. 6, the algorithm switches the weighting on the factors that it believes are able to best perform during the specific period. Exposure to each factor is often presents, however, in some periods the algorithm completely drops a few to make a clear strategic choice.

To assess the predictive performance for individual models and the ensemble of the two, we calculate the out-of-sample R^2 as:

$$AdjR^2_{oos} = 1 - \frac{\sum_{i,t} \mid |r_{i,t} - \hat{r_{i,t}}| \mid}{\sum_{i,t} \mid |r_{i,t}| \mid} \tag{3}$$

As opposed to:

$$R^2_{oos} = 1 - \frac{\sum_{i,t} \mid |r_{i,t} - \hat{r_{i,t}}| \mid}{\sum_{i,t} \mid |r_{i,t} - \bar{r_{i,t}}| \mid} \tag{4}$$

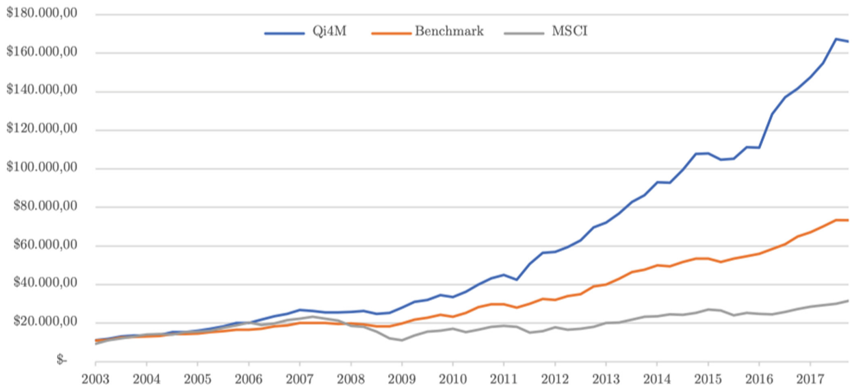

Fig. 5. Historical growth performance

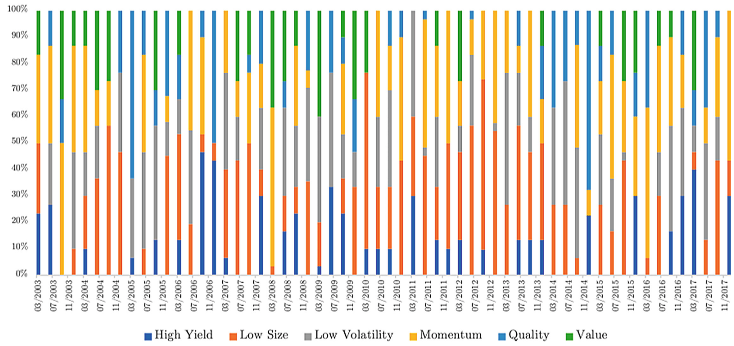

Fig. 6. Period by period factors weighting

For stocks $i = 1, ..., n$ and steps $t = 1, ...k$.

We applied this adjustment to the R^2_{oos} formula as it is often flawed when it comes to analysing individual stock returns. Historical mean returns are so noisy that they artificially lower the bar for 'good' performance. To avoid this pitfall, we benchmark the R^2_{oos} against a forecast value of zero (Kelly, 2017).

Table 1. Out-of sample R^2

	R^2_{oos}	$AdjR^2_{oos}$
Multi-layer perceptron	8.9%	41.7%
Multivariable regressor	4.8%	40.4%
Ensemble algorithm	5.4%	41.5%

Table 1 presents the comparison between the prediction methods adopted in terms of out-of-sample predictive R^2 Ultimately the values are compared to that obtained through the ensemble algorithm. The first row of Table 1 reports the R^2_{oos} for the multi-layer perceptron, showing it is able to explain 41.7% of the overall variability in the out-of-sample data. The second row presents the value for the multivariable regressor, that as expected is lower than the perceptron results, at 40.4%. The third and final row shows the R^2_{oos} for the ensemble algorithm, which unites the two method to return the final output of the model; the ensemble predictive ability is of 41.5%, showing it to be successful at explaining the variability in the data.

4 Conclusion

The results from the empirical study, supported by the existing literature, show that return prediction powered by machine learning and a multivariable regressor, if properly linked can provide an excellent equity-based investment strategy [9]. The model is able to produce a selection of stocks that outperforms the MSCI index, and furthermore outperforms the selected sub-universe benchmark. The ensemble algorithm ultimately connects the two methodologies to efficiently exploit both the two models' strengths and their temporal outlook differences. It is important to note that the strategy does not currently employ a stop- loss mechanism to protect the returns against negative market conditions. Research in the area is currently ongoing, and we expect to shortly update the strategy with the previously cited mechanism along with macroeconomic indicators that will improve the fine-tuning of such an implementation.

References

1. Bender, J., et al.: Foundations of Factor Investing. MSCI Index Research (2013)
2. Welch, I., Goyal, A.: A comprehensive look at the empirical performance of equity premium prediction. Rev. Financ. Stud. **21**, 1455–1508 (2008)
3. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
4. Eldan, R., Shamir, O.: The power of depth for feedforward neural networks. In: Feldman, V., Rakhlin, A., Shamir, O. (eds.) 29th Annual Conference on Learning Theory, Proceedings of Machine Learning Research, vol. 49, pp. 907–940 (2016)
5. Bishop, C.M.: Feed-forward network functions. In: Pattern Recognition and Machine Learning, pp. 226–231 (2016)
6. Fama, E.F., French, K.R.: A five-factor asset pricing model. J. Financ. Econ. **116**(1), 1–22 (2015)
7. Beaver, W.H., et al.: Delisting returns and their effect on accounting-based market anomalies. SSRN Electron. J. (2006)
8. Steege, F.-F., et al.: Effects of noise-reduction on neural function approximation. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2012)
9. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989)