risks

MDPI

Article

Automated Machine Learning and Asset Pricing [†]

Jerome V. Healy [1], Andros Gregoriou [1,*] and Robert Hudson [2]

[1] Liverpool Business School, Liverpool John Moores University, Liverpool L3 5UG, UK; j.v.healy@ljmu.ac.uk
[2] Business School, University of Hull, Hull HU6 7RX, UK; robert.hudson@hull.ac.uk
[*] Correspondence: a.gregoriou@ljmu.ac.uk

[†] G12—Asset Pricing. Any errors in the work remain the responsibility of the authors.

Abstract: We evaluate whether machine learning methods can better model excess portfolio returns compared to the standard regression-based strategies generally used in the finance and econometric literature. We examine 17 benchmark factor model specifications based on Expected Utility Theory and theory drawn from behavioural finance. We assess whether machine learning can identify features of the data-generating process undetected by standard methods and rank the best-performing algorithms. Our tests use 95 years of CRSP data, from 1926 to 2021, encompassing the price history of the broad US stock market. Our findings suggest that machine learning methods provide more accurate models of stock returns based on risk factors than standard regression-based methods of estimation. They also indicate that certain risk factors and combinations of risk factors may be more attractive when more appropriate account is taken of the non-linear properties of the underlying assets.

Keywords: machine learning; asset pricing; risk factors; prospect theory; Peak-End rule

JEL Classification: G10; G11; G12; C52; C55; C58; G0; G1; G17

1. Introduction

The application of machine learning in finance has grown along with increased computing power speed, memory capacity, and the vast amounts of data generated by modern financial markets. The development of "data science" as a distinct field has led to the recent development of numerous ML algorithms and their applications to tasks such as portfolio optimisation, risk modelling, trend analysis, and sentiment analysis of news, amongst others. However, regulators and many finance academics perceive ML methods as "black-box" procedures, and are sceptical of "empirical" or "engineering" techniques. This is particularly true for asset pricing, where the price and hence the future return of financial assets is estimated from a variety of factors.

There are two broad established approaches to asset pricing in the finance literature, namely, work based on Expected Utility Theory and that drawing on "behavioural finance". The first approach assumes that investors are rational and will make investment decisions with the objective of maximising their expected utility. These decisions will involve making appropriate "trade-offs" between risk and expected return. This approach is exemplified in the work of (Treynor 1961), and (Sharpe 1963, 1964) who developed the Capital Asset Pricing Model (CAPM) linking risk, as quantified by the standard deviation of market returns, and return in a way consistent with Expected Utility Theory. Subsequently, Mossin (1966), Lintner (1965, 1969), Black (1972), Merton (1973), Ross (1976), Fama and French (1993, 2015), Carhart (1997), and others extended and generalised the CAPM. The later work introduced a number of additional factors which empirically explain observed returns. Given that, by definition, expected returns should be driven by risk, in this research the extra factors are often referred to as risk factors.

check for updates

Citation: Healy, Jerome V., Andros Gregoriou, and Robert Hudson. 2024. Automated Machine Learning and Asset Pricing. *Risks* 12: 148. https://doi.org/10.3390/risks12090148

Academic Editors: Hong Liu and Angelos Dassios

Received: 25 July 2024
Revised: 21 August 2024
Accepted: 9 September 2024
Published: 14 September 2024

Although the Expected Utility Theory approach to asset pricing remains the dominant paradigm for academics and many market practitioners, the second approach, popularly referred to as "behavioural finance", is an important alternative and a more recent development. It rests on behavioural or cognitive models of decision-making under risk, and builds on insights from psychology and neuroscience. These insights can be used to develop factors which can be used to price assets without necessarily assuming rationality on the part of market participants.

It is, of course, empirically feasible to combine factors drawn from both the behavioural and Expected Utility Approaches to pragmatically derive the most effective asset pricing models. To a large extent, this is an empirical exercise and thus it is important to consider the most effective empirical methods, and this is the main issue addressed in this paper.

In our paper, we evaluate a number of appropriate asset pricing models. A number of models have become standard and well established in studies drawing on the Expected Utility approach. There is the original CAPM, which has been supplemented and partially replaced first by the Fama French three-factor model, and subsequently the Fama French five-factor model. There are many models associated with behavioural finance, so the choice of factors is less obvious. However, arguably, the most seminal work in the behavioural area is that of (Kahneman and Tversky 1979), who developed Prospect Theory and the related concept of reference dependence. One of the best-known examples of reference dependence is the Peak-End rule (Fredrickson and Kahneman 1993). Thus, it is reasonable to consider combining the Peak-End rule with factor models based on Expected Utility theory as a benchmark for our empirical work.

Our selection of factors follows the work of (Gregoriou et al. 2019), who tested the asset pricing performance of the Peak-End rule, and thus of Prospect Theory. Their results confirmed that Peak-End behaviour by investors occurs and is not captured by factor models based on Expected Utility Theory. Their proposed seven-factor pricing model, incorporating the insights of both Expected Utility theory and Prospect Theory, outperforms other popular factor models in explaining portfolio returns.[1]

However, their tests of asset pricing models were based on the use of standard regression techniques utilising OLS, as indeed is standard in the asset pricing literature. In this paper, we investigate whether recently developed machine learning algorithms can identify features of the data generating process undetected by standard methods. A major reason for applying ML techniques to financial tasks is that ML methods are able to model non-linear relationships in the data. Non-linear techniques are required when outputs are not directly proportional to the inputs. Traditional analytical methods (e.g., OLS) assume that a linear relationship exists, or utilise non-linear functions that can be simplified to a linear model. There is considerable general evidence that non-linear relationships are prevalent in the financial markets (Amini et al. 2021). Moreno and Olmeda (2007) give a summary of inter-temporal work on non-linear modelling. Kolm et al. (2014) and Carroll et al. (2017) give summaries of cross-sectional work on non-linear modelling in financial markets. Despite this, there is very limited academic work considering non-linear effects in asset pricing. Similarly, to our knowledge, modern machine learning techniques have not been used in asset pricing.[2]

In our paper, we specifically test whether Automated Machine Learning (AutoML) can better model excess portfolio returns compared to the standard regression-based strategies used in the econometric literature. Our results support (Gu et al. 2020) by showing that machine learning methods provide a superior fit of the data given the non-linearities involved. This is very important when it comes to how stock markets react to news. An important element of this is that reactions to news (under- and over-reactions) are possibly not estimated correctly with the use of OLS. This clearly implies that machine learning should be applied in the estimation of not only asset pricing models but also to the efficient market hypothesis. Machine learning could provide a solution to the joint hypothesis problem when testing for market efficiency.

The remainder of the paper is organized as follows. Section 2 discusses the reasons for applying machine learning and particularly AutoML and describes the specific instantiation of the technology which we use. Section 3 describes our data sources and the testing methodology we use. Section 4 discusses our findings. Finally, Section 5 terminates the article with our summary, suggestions for further work, and conclusions.

2. Machine Learning and AutoML

Machine Learning is a sub field of Artificial Intelligence (AI), and encompasses a large and varied set of algorithms suited to different tasks. Broadly, these can be classified into three categories:

(1) Unsupervised machine learning: a data mining technique for partitioning and reducing the dimensionality of data. Unsupervised learning generalizes statistical approaches to data reduction, such as principal component analysis. An example of unsupervised learning is K-means clustering for portfolio selection.

(2) Supervised machine learning: either a parametric or non-parametric, algorithmic or probabilistic method of learning the relationship between response and explanatory variables. Supervised machine learning generalizes statistical techniques such as ordinary least squares (OLS) regression, or time series methods such as auto-regressive models.

(3) Reinforcement learning: a method of stochastic control, with feedback, which learns a policy based on decisions which change the state of inputs. Reinforcement learning generalizes stochastic dynamic programming. Example applications include derivative pricing, optimal hedging, and optimal trade execution.

In our study, we investigated Supervised Learning (task 2) through the use of regression analysis. For our empirical analysis, we used Automated Machine Learning and we elected to use Microsoft H2O Automl (https://azure.microsoft.com/en-us/solutions/automated-machine-learning accessed on 21 August 2024). This is an open-source code, which avoids the problems associated with "black-box" systems. Also, it offers a selection of up-to-date ML algorithms, and is among the industry leaders in the field. We will now describe H2O AutoML and the ML algorithms it offers.

2.1. H2O AutoML

H2O AutoML (LeDell and Poirier 2020) provides an interface which automates the process of training a large selection of candidate models by performing a number of modelling-related tasks. These include the automatic training and tuning of many models within a user-specified number or time-limit. Stacked Ensembles—one based on all previously trained models, another one on the best model of each family—are automatically trained on collections of individual models to produce ensemble models which, in most cases, will be the top performing models in theAutoML Leaderboard. A number of model explanatory methods are provided. These apply to AutoML objects (groups of models) as well as individual models. Explanations can be generated automatically, with a single function call providing an interface for exploring and explaining the AutoMLmodels. The H2O AutoML interface is designed to have as few parameters as possible so that all the user needs to do is point to their data set, identify the response column, and specify a time constraint or limit on the number of total models trained.

2.2. H2O AutoML Machine Learning Algorithms

The following algorithms are currently supported: Distributed Random Forest (DRF), Extremely Randomised Trees (XRT), General Linearised Models (GLM), Gradient Boosting Machine (GBM), Deep Learning (Neural Networks), and Stacked Ensembles. There follows a brief description of each algorithm.

2.2.1. DRF

Distributed Random Forest (DRF) (Assunçao et al. 2013; Breiman 2001) is a tree-based classification and regression tool. When given a set of data, DRF generates a "forest" of classification or regression trees, rather than a single classification or regression tree. Increasing the number of trees will reduce the variance, without increasing the bias. Both classification and regression take the average prediction over all of the trees generated to make a final prediction. For a classification task this will be a category, and for a regression task it will take a numeric value.

2.2.2. XRT

In random forests, a randomly selected subset of data features (variables) is used to decide on the splitting rule for each branching. In extremely randomized trees (XRT), a random subset of candidate features is also used. However, thresholds are also drawn at random for each data feature, and the best is picked as the splitting rule. This allows a further reduction in the variance of the model, but at the cost of a small increase in bias (Geurts et al. 2006).

2.2.3. GLM

A generalized linear model (GLM), as its name suggests, is a generalization of ordinary linear regression (OLS) that allows the dependent (response) variable to have a non-normal error distribution. In a GLM, the linear model is related to the response variable by a link function and by permitting the variance of each measurement to be a function of its predicted value. In a GLM, each value Y of the response variables is assumed to follow an exponential distribution, which could be, e.g., a normal, binomial, Poisson, gamma, or other exponential distribution. The mean, μ, of the distribution depends on the independent variables, X, through

$$E(Y|X) = \mu = g^{-1}(X\beta)$$

where *E(Y | X)* is the expected value of Y given X, and β is a vector of unknown parameters. g is the link function. The variance of each measurement is given by

$$Var(Y|X) = V\left(g^{-1}(X\beta)\right)$$

Any of Maximum Likelihood, Maximum Quasi-Likelihood, or Bayesian techniques can be used to estimate the parameters β, and the V may be from any exponential distribution (Nelder and Wedderburn 1972; Lee et al. 2006; Friedman et al. 2010). GLMs can be used for prediction or classification.

2.2.4. GBM

Gradient Boosting Machines can be used for both regression and classification tasks. They are an ensemble modelling technique based, usually, on decision trees, which produce an ensemble of weakly predictive models. Gradient boosting combines these weakly predictive models into one strongly predictive model by an iterative process. Suppose we run the algorithm for m iterations. Each run produces an imperfect model

$$F_m(x) = \hat{y}_m$$

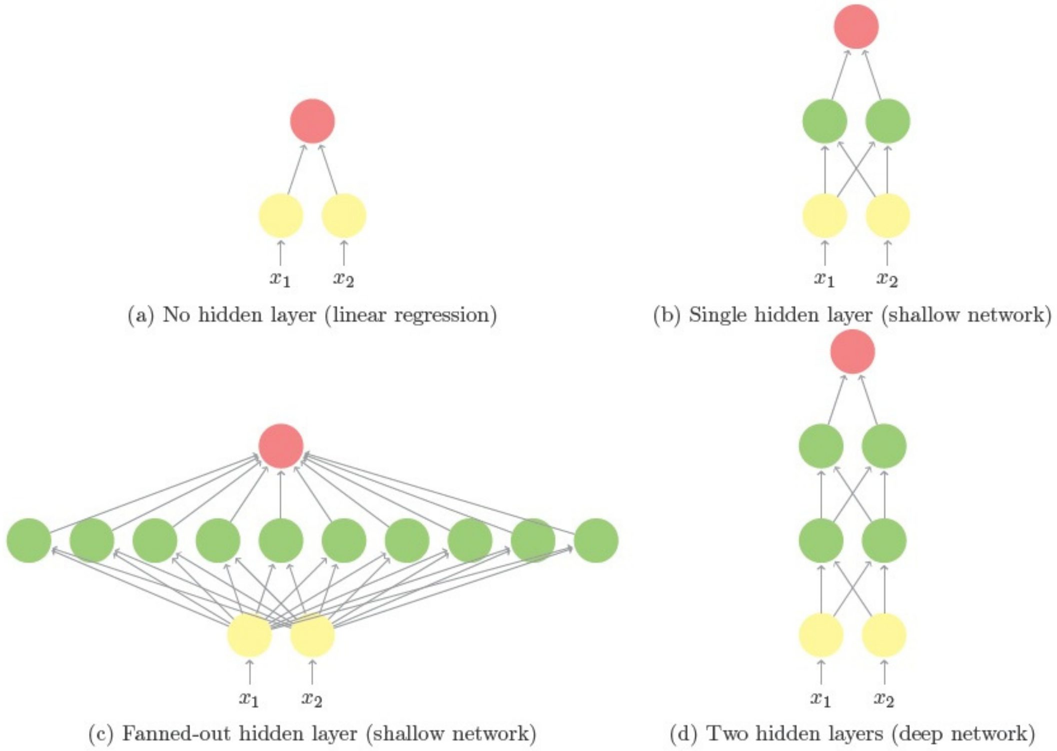
The next iteration will improve this estimate by appending another estimator, thus

$$F_{m+1}(x) = F_m(x) + h_m(x) = \hat{y}_{m+1}$$

where $h_m(x) = y - F_m(x)$. Thus, gradient boosting fits h to the residuals $y - F_m(x)$. Each iteration F_{m+1} therefore improves on the estimate of the previous iteration F_{m+1}, by minimising the loss function (Friedman et al. 2000; Hastie et al. 2009).

2.3. Deep Learning

Deep Learning networks are a form of Artificial Neural Network (ANN) containing multiple hidden layers. In recent years, the term "Deep Learning" has become synonymous with "neural network". Kolmogorov (1957), in his representation theorem, showed that an ANN with a single hidden layer can approximate any Borel measurable function. In their seminal papers, (Hornik et al. 1989, 1990) showed that an ANN with a single hidden layer is capable of arbitrarily accurate approximation to any function and its derivatives, to any desired degree of accuracy, provided that sufficient hidden units are available. This class of network is now referred to as a "shallow network" (Figure 1c), whereas the term "deep network" refers to ANNs with multiple hidden layers.



(a) No hidden layer (linear regression)

(b) Single hidden layer (shallow network)

(c) Fanned-out hidden layer (shallow network)

(d) Two hidden layers (deep network)

Figure 1. Neural network architectures. Source: Dixon and Halperin (2019).

Deep Learning has become popular in finance because it can handle high dimensional inputs. The principal advantage of Deep Learning, however, is computational speed and efficiency.

Figure 2 shows a deep learning network applied to the task of forecasting the S&P500 index. The inputs are the 10 largest stocks in the index and there are two hidden layers. Mathematically, the network is represented as follows:

$$\hat{\mu}_y(x;\hat{\Omega} = \Theta\left(\sum_{h=1}^{H} w_{lh}\Phi_h(\sum_{k=1}^{K} w_{hk}x_k + \omega_h) + \omega_l\right)$$

Here, the left-hand side is the output. The DL network consists of one layer of K input nodes $x_1, \ldots, x_K$, a layer of l output nodes, and H hidden layer nodes. In this case, $K = 10$, $l = 1$, and $H = 2$. The functions Θ and Φ are termed activation functions. These can be sigmoidal $1/(1+e^{-x})$, cosh(x), tanh(x), heavyside gate functions, or rectified linear units (ReLU), Max(x, 0). The latter function has proved particularly effective for dimension reduction. For a continuously valued target variable, the output functions Θ are usually linear and may be the identity. The w are referred to as the weights and the ω are constant intercept terms known as biases. The set of estimated weights and biases is denoted by $\{\hat{\Omega}, (w_1, \ldots, w_{KH+Hl}, \omega_1, \ldots, \omega_{H+l}) \in \hat{\Omega}\}$. These weights and biases cannot be interpreted, as coefficients in OLS regression are interpreted. However, the Jacobian matrix of these weights and biases can be used to calculate the relative importance of each input to the network.

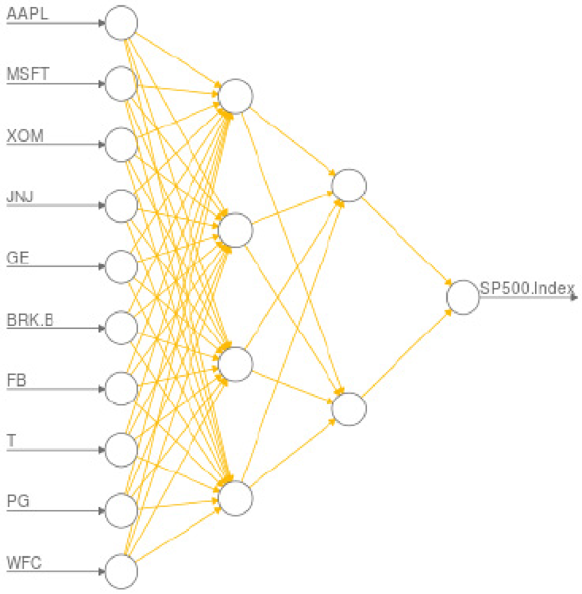


Figure 2. Deep learning applied to S&P500 index forecasting. Source: Heaton et al. (2017).

H2O's Deep Learning is based on a multi-layer feed-forward ANN that is trained with stochastic gradient descent using back propagation. The network can contain a large number of hidden layers consisting of neurons with tanh, rectifier, and maxout activation functions. Advanced features such as adaptive learning rate, rate annealing, momentum training, dropout, L1 or L2 regularization, checkpointing, and grid search are provided.[3]

2.4. Stacked Ensembles

Ensemble ML methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any one of the constituent ML algorithms. Many popular modern machine learning algorithms are actually ensembles. For example, the above discussed Random Forest and Gradient Boosting Machine (GBM) are both ensemble learners. Both bagging (e.g., Random Forest) and boosting (e.g., GBM) are methods for ensembling that take a collection of weak learners (e.g., decision tree) and form a single, strong learner. Stacking (Wolpert 1992), also referred to as stacked generalization, is another method of combining the predictions of several individual ML algorithms. The ML algorithms are first trained on the available data. A combiner algorithm is then trained to make a final prediction using the individual predictions of the ML algorithms as inputs. A logistic regression model is commonly used as the combiner. The resulting stacked ensemble frequently out-performs any of the constituent individual ML algorithms.

H2O's Stacked Ensemble method is a supervised ensemble machine learning algorithm that finds the optimal combination of a collection of prediction algorithms using stacking. Like all supervised models in H2O, Stacked Ensemble supports regression, binary classification, and multi-class classification.[4]

3. Data and Methodology

Following Fama and French (2015), and Gregoriou et al. (2019), we tested all our specifications on a common data set. We initially used the U.S. Research Returns Data 25 value-weighted portfolios (daily and monthly), sorted by size and momentum, together with the associated momentum risk factors. In addition, we used the Fama–French three-factor model and five-factor model risk factors, together with the applicable one-month T-bill rates,[5] which were obtained from the Kenneth R. French Data Library. These data provide the most complete history of returns possible for the broad US stock market from 1926 to 2021. Using these data facilitates comparability with the results of Fama and French (1993, 2015), Carhart (1997), and others who have used the same data. The full sample period we used was 1927:01 to 2021:06, giving 1133 monthly observations. We used the full sample period for our analysis. We tested thirteen different model specifications on the full sample period. Our test assets (dependent variables) were the excess returns generated by the 25 portfolios. We also added a 26th asset, the excess return of Portfolio 25—Portfolio 1 (big minus small). We computed the excess returns by subtracting the T-bill rate from each portfolio return. We examined how well each of our specifications and each ML algorithm priced the excess returns of these portfolios, compared to a conventional analysis using OLS. Additionally, we calculated four further models which included the RMW and CMA profitability and investment risk factors from the Fama and French (2015) five-factor model for the 693 months for which these factors are available. In all, we estimated 4420 individual models for 17 specifications and 26 portfolios, and generated 10 different ML models for each.

3.1. The Factor Model Specifications Estimated

In order to assess how well ML algorithms measure risk-adjusted returns, we first analysed model specifications (1)–(13) in Table 1 on the full data set, and then specifications (14) to (17) on the smaller 693 month data set.

Table 1. Specifications tested.

$r_{it} - r_{ft} = \alpha_i + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(1)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \varepsilon_{it}$	(2)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \varepsilon_{it}$	(3)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(4)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(5)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \varepsilon_{it}$	(6)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \varepsilon_{it}$	(7)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(8)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(9)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,MOM_t} MOM_t + \varepsilon_{it}$	(10)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,MOM_t} MOM_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \varepsilon_{it}$	(11)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,MOM_t} MOM_t + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(12)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB3_i} SMB3_t + \beta_{i,HML_i} HML_t + \beta_{i,MOM_t} MOM_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(13)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB5_i} SMB5_t + \beta_{i,HML_i} HML_t + \beta_{i,RMW_t} RMW_t + \beta_{i,CMA_t} CMA_t + \varepsilon_{it}$	(14)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB5_i} SMB5_t + \beta_{i,HML_i} HML_t + \beta_{i,RMW_t} RMW_t + \beta_{i,CMA_t} CMA_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \varepsilon_{it}$	(15)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB5_i} SMB5_t + \beta_{i,HML_i} HML_t + \beta_{i,RMW_t} RMW_t + \beta_{i,CMA_t} CMA_t + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(16)
$r_{it} - r_{ft} = \alpha_i + \beta_{i,MKT_t} MKT_t + \beta_{i,SMB5_i} SMB5_t + \beta_{i,HML_i} HML_t + \beta_{i,RMW_t} RMW_t + \beta_{i,CMA_t} CMA_t + \beta_{i,MaxPi_{t-1}} MaxPi_{t-1} + \beta_{i,Pi_{t-1}} Pi_{t-1} + \varepsilon_{it}$	(17)

Specification (1) is the Peak-End hypothesis in isolation. In Specification (1), $r_{it} - r_{ft}$ is the portfolio i alpha. $MaxPi_{t-1}$ is the monthly excess return for portfolio i, implied by the highest daily return occurring in the previous month. This is the PEAK variable. Pi_{t-1} is the monthly excess return for portfolio i for the previous month; this is the END variable. ε_{it} is a zero-mean residual. Specification (2) is the single-factor CAPM. The variable MKT_t in Specification (2) is ($r_{MKTt} - r_{ft}$), and represents the excess return on the market at time t. Specifications (3), (4), and (5), respectively, add first the peak factor $MaxPi_{t-1}$, then the end factor Pi_{t-1}, and finally both together to the single CAPM risk factor. Specification (6) is the Fama and French (1993) three-factor model, which adds firm size and value factors to the single CAPM factor. Specifications (7), (8), and (9), respectively, add first the peak factor $MaxPi_{t-1}$, then the end factor Pi_{t-1}, and finally both together to the three (Fama and French 1993) risk factors. Specification (10) is the Carhart (1997) model, which extends the Fama and French (1993) three-factor model by additionally including a momentum factor. As before, specifications (11), (12), and (13), respectively, add first the peak factor $MaxPi_{t-1}$, then the end factor Pi_{t-1}, and finally both together to the four Carhart (1997) factors.

Specification (14) in Table 1 is the Fama and French (2015) five-factor model. This includes two additional factors, RMW_t and CMA_t, intended to capture profitability and investment effects, respectively. As before, we added first the peak factor $MaxPi_{t-1}$, then the end factor Pi_{t-1}, and finally both together to Specification (14), giving us Specifications (15)–(17).

If, in any of the specifications in Table 1, ML algorithms provide more accurate measurements of excess returns, then we should expect to find significantly smaller residuals in out-of-sample tests than we observe for conventional estimation methods.

3.2. Testing Procedure

Firstly, we used our software to perform data scaling. ML algorithms generally perform better if the data are scaled between 0 and 1. AutoML does not currently support this data pre-processing feature. However, the required data pre-processing (scaling) and post-processing (inverse scaling) is provided by our Python code. Next, we utilised AutoML to generate 10 models, using its supported algorithms for each of our 17 specifications on each of the 26 portfolios. We generated out-of-sample performance statistics for each model. Following inverse scaling, the out-of-sample actual and estimated values of excess returns for each model were saved. Two-sample *t*-tests assuming unequal variances were performed on each of these pairs.

For comparison, we next performed OLS regressions for each of our 17 specifications for each of the 26 portfolios. The resulting models were again used to generate out-of-sample performance statistics, on the same-sized withholding data sets as used for our ML models. The out-of-sample actual and estimated excess returns were again saved and paired *t*-tests performed.

4. Empirical Findings

Our findings are presented in Table 2. The table compares the results from the ML algorithms with those from OLS regressions for the same data for portfolio 26 which represents the difference between portfolio's P25 and P1. In particular, we compare the two approaches on one month's, out-of-sample data. We measure the out-of-sample root mean squared error (RMSE), mean absolute error (MAE) and excess returns for each of the methods. We initially see that various different H2O AutoML algorithms performed best depending on the particular Asset Pricing model. It can be seen that the out-of-sample root mean squared error (RMSE) are smaller for the ML algorithms in almost all cases, the only exceptions being for specifications (7) and (14) where the OLS results are marginally smaller. The results are also strong for the mean absolute error (MAE) figures with the ML being superior for all except 5 specifications and in these cases the difference is marginal. Thus, generally, for any given model the ML algorithms will give more accurate projections.

As another comparison between the ML algorithms and the OLS estimates, we examined the out-of-sample excess returns (alphas). If the risk factors in an asset pricing model explain all the variation in expected returns, then alpha should be 0 (Gregoriou et al. 2019). Thus, a larger value of alpha in absolute terms indicates a less effective asset pricing model. Given this, we can evaluate how well the risk factors in the model explain the market performance out of sample and whether different risk factors or combinations of risk factors might perform better. We see that whether the absolute values of alpha from the ML algorithms or OLS are larger varies substantially across the asset pricing models, with neither approach being systematically dominant. For the simple Peak + End model (model 1), the OLS performs significantly better. For the single factor CAPM (model 2), there is no significant difference between the alpha estimates. For the models extending the CAPM by adding in the peak and/or end variables (models 3 to 5), the OLS tends to perform better although not always to a significant extent. For the models based on the FF three-factor model (models 6 to 9), the ML performs substantially better for models 6 and 8 but not for the other two models. For the models based on the Carhart model (models 10 to 13), the ML always performs substantially and significantly better. For the models based on the FF five-factor model (models 14 to 17), there are not significant differences in performance except for in model 17, where OLS performs significantly better.

Table 2. Portfolio P26—out-of-sample results, H2O AutoML vs. OLS estimation. This table shows the estimates of abnormal returns (Jensen's Alpha) for portfolio 26 (the difference between portfolio P25 and portfolio P1), estimated by AutoML and OLS, respectively, for each of 17 popular factor models from the asset pricing literature. Portfolio P25 contains the largest capitalisation firms with the largest momentum. Portfolio P1 contains the smallest capitalisation firms with the smallest momentum. In the table, the first column gives the specification tested. The second column indicates which H2O AutoML algorithm performed the best for each specification for portfolio P26. The next two columns give the RMSE and MAE achieved by the algorithm in question on out-of-sample data. The following column gives the corresponding out-of-sample estimate of annualised excess returns. The following three columns give corresponding out-of-sample statistics for the same specifications resulting from OLS regressions. The final column gives the t-statistic for a paired *t*-test of the AutoML vs. OLS estimates.

Specification	**H2O AutoML**				**OLS**			**T-stat**
Results for Portfolios P26	**Algorithm**	**RMSE**	**MAE**	**Alpha**	**RMSE**	**MAE**	**Alpha**	**H2O Automl vs. OLS**
(1) PEAK + END alpha	SE Best of Family	7.31	5.84	−0.36	7.46	5.92	−0.11	−2.51 ***
(2) SINGLE FACTOR CAPM alpha	GBM	6.46	5.42	0.22	6.50	5.44	−0.02	1.27
(3) SINGLE FACTOR CAPM + PEAK alpha	GLM	6.48	5.36	−0.59	6.48	5.37	−0.49	−5.35 ***
(4) SINGLE FACTOR CAPM + END alpha	GBM	6.32	5.10	−0.32	6.51	5.43	−0.04	−1.02
(5) SINGLE FACTOR CAPM +PEAK + END alpha	GBM	6.46	5.24	−1.01	6.49	5.38	−0.47	−1.89 *
(6) FF 3 FACTOR MODEL alpha	SE Best of Family	5.90	4.84	0.75	5.94	4.74	1.83	−4.65 ***
(7) FF 3 FACTOR MODEL + PEAK alpha	SE Best of Family	5.75	4.67	1.28	5.69	4.63	1.17	3.91 ***
(8) FF 3 FACTOR MODEL+ END alpha	SE All Models	5.67	4.66	0.71	6.00	4.78	1.84	−3.56 ***
(9) FF 3 FACTOR MODEL + PEAK + END alpha	SE Best of Family	5.72	4.66	1.22	5.72	4.65	1.22	−0.19
(10) CARHART MODEL alpha	SE All Models	4.39	3.39	0.26	4.65	3.51	1.01	−6.55 ***
(11) CARHART MODEL + PEAK alpha	SE All Models	4.38	3.39	−0.01	4.39	3.31	0.48	−4.08 ***
(12) CARHART MODEL + END alpha	SE Best of Family	4.49	3.48	0.14	4.65	3.49	0.97	−5.55 ***
(13) CARHART MODEL + PEAK + END alpha	SE Best of Family	4.41	3.38	−0.30	4.46	3.34	0.60	−5.53

Table 2. *Cont.*

Specification	H2O AutoML				OLS			T-stat
Results for Portfolios P26	**Algorithm**	**RMSE**	**MAE**	**Alpha**	**RMSE**	**MAE**	**Alpha**	**H2O Automl vs. OLS**
(14) FF 5 FACTOR MODEL alpha	SE Best of Family	4.76	3.67	1.17	4.72	3.74	1.21	−0.14
(15) FF 5 FACTOR MODEL + PEAK alpha	SE Best of Family	4.53	3.49	0.60	4.54	3.58	0.40	1.28
(16) FF 5 FACTOR MODEL+ END alpha	SE Best of Family	4.62	3.61	1.30	4.75	3.75	1.21	0.62
(17) FF 5 FACTOR MODEL + PEAK + END alpha	SE All Models	4.13	3.38	1.02	4.51	3.57	0.37	2.70 ***

Significant at 1% level ***. Significant at 5% level **. Significant at 10% level *.

Given the foregoing, we can see that if ML algorithms are used we might arrive at rather different optimal factor models from those determined using OLS. For example, taking into account the accuracy of predictions and the support for the risk factors chosen, the Carhart-based models seem much more attractive when ML methods are used. The Carhart-based models generally perform better than the FF five-factor models which are coming into prominence in the asset pricing literature.

5. Conclusions

We have evaluated whether machine learning methods can better model excess portfolio returns compared to standard regression-based strategies from the econometric literature. We have examined 17 benchmark factor model specifications which are based on well-known works in the finance literature. The factors are based on Expected Utility Theory and Prospect Theory. We used a variety of machine learning methods and ranked the best-performing algorithms. Our tests used 95 years of CRSP data from 1926 to 2021, encompassing the price history of the broad US stock market. Our findings suggest that machine learning methods provide more accurate models of stock returns based on risk factors than standard regression-based methods of estimation. They also indicate that certain risk factors and combinations of risk factors may be more attractive when more appropriate account is taken of the non-linear properties of the underlying assets. We believe our research has provided three fundamental contributions to the literature, which are the following. First, asset pricing models are greatly improved when they encapsulate behavioural factors, supporting the research of (Gregoriou et al. 2019). Second, we established that machine learning provides a superior fit of the data and should be used to estimate asset pricing models in the future, instead of standard OLS. Finally, we witnessed the power of Automated Machine Learning, where the optimal non-linear model is determined empirically by the data.

Future work might usefully consider whether machine learning can be of assistance in selecting possible new factors for use in asset pricing. Also, more empirical evidence for different financial markets and over various portfolios would add value to the research in this field.

Author Contributions: Conceptualization, J.V.H.; methodology, A.G.; investigation, R.H. and A.G.; writing, R.H.; validation, R.H.; writing—review and editing, A.G.; supervision, A.G.; software, J.V.H.; formal analysis, J.V.H.; resources, J.V.H.; data curation, J.V.H.; visualization, J.V.H.; project administration, J.V.H.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Notes

[1] Gregoriou et al. (2019) was recently cited by (Saona et al. 2023; Le and Gregoriou 2022).

[2] There has been modest use of artificial intelligence techniques in Finance mainly for prediction see, for example, (Manahov et al. 2019) for a summary of work using Genetic Programming.

[3] https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html accessed on 21 August 2024.

[4] https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html accessed on 21 August 2024.

[5] For more information readers are referred to the following web resources http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/25_Portfolios_ME_Prior_12_2_CSV.zip accessed on 21 August 2024. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/25_Portfolios_ME_Prior_12_2_Daily_CSV.zip accessed on 21 August 2024. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/F-F_Momentum_Factor_CSV.zip accessed on 21 August 2024. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/F-F_Momentum_Factor_daily_CSV.zip accessed on 21 August 2024.

References

Amini, Shima, Robert Hudson, Andrew Urquhart, and Jian Wang. 2021. Nonlinearity everywhere: Implications for empirical finance, technical analysis and value at risk. *The European Journal of Finance* 27: 1326–49. [CrossRef]

Assunçao, Joaquim, Paulo Fernandes, Lucelene Lopes, and Silvio Normey. 2013. Distributed Stochastic Aware Random Forests—Efficient Data Mining for Big Data. Paper presented at the 2013 IEEE International Congress on Big Data, Santa Clara, CA, USA, June 27–July 2; pp. 425–26.

Black, Fischer. 1972. Capital market equilibrium with restricted borrowing. *The Journal of Business* 45: 444–55. [CrossRef]

Breiman, Leo. 2001. Random forests. *Machine Learning* 45: 5–32. [CrossRef]

Carhart, Mark M. 1997. On persistence in mutual fund performance. *The Journal of Finance* 52: 57–82. [CrossRef]

Carroll, Rachael, Thomas Conlon, John Cotter, and Enrique Salvador. 2017. Asset allocation with correlation: A composite trade-off. *European Journal of Operational Research* 262: 1164–80. [CrossRef]

Dixon, Matthew Francis, and Igor Halperin. 2019. The four horsemen of machine learning in finance. *Social Science Research Network Electronic Journal*. [CrossRef]

Fama, Eugene F., and Kenneth R. French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33: 3–56. [CrossRef]

Fama, Eugene F., and Kenneth R. French. 2015. A five-factor asset pricing model. *Journal of Financial Economics* 116: 1–22. [CrossRef]

Fredrickson, Barbara L., and Danielm Kahneman. 1993. Duration neglect in retrospective evaluations of affective episodes. *Journal of Personality and Social Psychology* 65: 45. [CrossRef]

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2000. Special invited paper. Additive Logistic regression: A statistical view of boosting. *Annals of Statistics*, 337–407. [CrossRef]

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* 33: 1–12. [CrossRef] [PubMed]

Geurts, Pierre, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning* 63: 3–42. [CrossRef]

Gregoriou, Andros, Jerome V. Healy, and Huong Le. 2019. Prospect theory and stock returns: A seven factor pricing model. *Journal of Business Research* 101: 315–22. [CrossRef]

Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33: 2223–73. [CrossRef]

Hastie, Trevor, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. Boosting and additive trees. In *The Elements of Statistical Learning*. New York: Springer, pp. 337–87.

Heaton, James B., Nick G. Polson, and Jan Hendrik Witte. 2017. Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry* 33: 3–12. [CrossRef]

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2: 359–66. [CrossRef]

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. 1990. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks* 3: 551–60. [CrossRef]

Kahneman, Daniel, and Amos Tversky. 1979. Prospect Theory: An Analysis of Decision Under Risk. *Econometrica* 47: 263–91. [CrossRef]

Kolmogorov, Andrei Nikolaevich. 1957. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk*. Moscow: Russian Academy of Sciences, vol. 114, pp. 953–56.

Kolm, Petter N., Reha Tütüncü, and Frank J. Fabozzi. 2014. 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research* 234: 356–71. [CrossRef]

LeDell, Erin, and Sebastien Poirier. 2020. H2o automl: Scalable automatic machine learning. Paper presented at the AutoML Workshop at ICML, San Diego, CA, USA, July 18. San Diego: ICML.

Lee, Youngjo, John A. Nelder, and Yudi Pawitan. 2006. *Generalized Linear Models with Random Effects*. Boca Raton: Chapman & Hall/CRC.

Le, Huong, and Andros Gregoriou. 2022. Liquidity and asset pricing: Evidence from a new free-float-adjusted price impact ratio. *Journal of Economic Studies* 49: 751–71. [CrossRef]

Lintner, John. 1965. Security prices, risk, and maximal gains from diversification. *The Journal of Finance* 20: 587–615.

Lintner, John. 1969. The aggregation of investor's diverse judgments and preferences in purely competitive security markets. *Journal of Financial and Quantitative Analysis* 4: 347–400. [CrossRef]

Manahov, Viktor, Robert Hudson, and Andrew Urquhart. 2019. High-frequency trading from an evolutionary perspective: Financial markets as adaptive systems. *International Journal of Finance & Economics* 24: 943–62.

Merton, Robert C. 1973. An intertemporal capital asset pricing model. *Econometrica: Journal of the Econometric Society*, 867–87.

Moreno, David, and Ignacio Olmeda. 2007. Is the predictability of emerging and developed stock markets really exploitable? *European Journal of Operational Research* 182: 436–54. [CrossRef]

Mossin, Jan. 1966. Equilibrium in a capital asset market. *Econometrica: Journal of the Econometric Society* 34: 768–83. [CrossRef]

Nelder, John Ashworth, and Robert W. M. Wedderburn. 1972. Generalized Linear Models. *Journal of the Royal Statistical Society Series A (General)* 135: 370–84. [CrossRef]

Ross, Stephen A. 1976. The Arbitrage Theory of Capital Asset Pricing. *Journal of Economic Theory* 13: 343–62. [CrossRef]

Saona, Paolo, Laura Muro, and Andros Gregoriou. 2023. The phenomenon of zero-leverage policy: Literature review. *Research in International Business and Finance* 66: 102012. [CrossRef]

Sharpe, William F. 1963. A Simplified Model for Portfolio Analysis. *Management Science* 9: 277–93. [CrossRef]

Sharpe, William. F. 1964. Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *Journal of Finance* 19: 425–42.

Treynor, Jack L. 1961. Market Value, Time, and Risk. Available online: https://ssrn.com/abstract=2600356 (accessed on 21 August 2024).

Wolpert, David H. 1992. Stacked generalization. *Neural Networks* 5: 241–59. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.