 https://doi.org/10.26219/heal.aueb.6658

SCHOOL OF INFORMATION SCENCES & TECHNOLOGY

Department of Statistics

Postgraduate Program

Thesis Title: Industry Return predictability via Machine Learning

Author: Stamatopoulos Charis

Supervisor: Vrontos Ioannis

Athens, Greece

February 2021

https://doi.org/10.26219/heal.aueb.6658

Σχολή Επιστημών & Τεχνολογίας της Πληροφορίας

Τμήμα Στατιστικής

Μεταπτυχιακό Πρόγραμμα

Τίτλος Διπλωματικής: Προβλεπτική Ικανότητα βιομηχανικών αποδόσεων μέσω Machine Learning

Φοιτητής: Σταματόπουλος Χάρης

Επιβλέπων Καθηγητής: Ιωάννης Βρόντος

Αθήνα, Ελλάδα

Φεβρουάριος 2021

https://doi.org/10.26219/heal.aueb.6658

ABSTRACT

In this particular project we conducted a comparative analysis of several machine learning techniques to detect whether lagged industry returns of thirty (30) major sectors of the economy can benefit us in predicting industry returns across the entire economy. We used a general linear regression predictive framework as our benchmark. We employed methodologies such as LASSO, elastic-net, and neural networks to select the model with the most accurate forecasts. After controlling for multiple testing and downside biases in magnitude, we found significant in sample evidence of return predictability. Lagged returns for finance, commodity and material producing industries are the most frequently selected significant predictors by our penalization methods, and thus exhibit a widespread predictive ability. We created zero-net industry-rotation portfolios (one for each methodology) which goes long the industries with the highest forecasted returns. For this purpose, we used walk-forward cross validation and backtesting simulation. We concluded that elastic net generated the higher annualized returns and Sharpe ratios, whereas at the same time the lowest MSE across predictions and annualized drawdown across all models. Finally, neural networks weren’t capable of outperforming clearly the elastic net procedure, probably due to the lack of high frequency data, the insufficient parametrization of the network or just the absence of non-linear relationships among the predictor variables and the dependent variables.

https://doi.org/10.26219/heal.aueb.6658

ΠΕΡΙΛΗΨΗ

Στην προκειμένη εργασία πραγματοποιήσαμε μια συγκριτική ανάλυση διαφόρων machine learning τεχνικών για να εντοπίσουμε εάν οι παρελθοντικές αποδόσεις τριάντα (30) βασικών τομέων της οικονομίας μπορούν να συνεισφέρουν στην πρόβλεψη της μελλοντικής απόδοσης των κύριων τομέων της οικονομίας. Ένα γενικό μοντέλο γραμμικής παλινδρόμησης για πρόβλεψη χρησιμοποιήθηκε ως σημείο αναφοράς (benchmark). Με το benchmark μοντέλο συγκρίναμε διάφορες machine learning τεχνικές όπως το LASSO, το elastic net και τα τεχνητά νευρωνικά δίκτυα. Αφού ελέγξαμε για multiple testing και μεροληψία στις εκτιμήσεις μας, προέκυψε στατιστικά σημαντική ένδειξη προβλεπτικής ικανότητας στις παρελθοντικές αποδόσεις εντός του δείγματος. Παρελθοντικές αποδόσεις τομέων όπως το finance, τα εμπορεύματα και τα υλικά αποδείχθηκε ότι επηρεάζουν τις αποδόσεις των περισσότερων βιομηχανιών. Δημιουργήσαμε τεχνητά industry-rotation χαρτοφυλάκια για την κάθε μεθοδολογία τα οποία αποτελούνταν από τις βιομηχανίες με την υψηλότερη προβλεπόμενη απόδοση. Αυτό επιτεύχθηκε μέσω walk forward cross validation και backtesting προσομοίωσης. Σαν συμπέρασμα καταλήξαμε ότι το elastic net παράγει τις πιο ακριβείς προβλέψεις ανάμεσα στα υποψήφια μοντέλα ως προς τις αποδόσεις σε ετήσια βάση και την ελαχιστοποίηση του σφάλματος MSE. Τέλος, τα νευρωνικά δίκτυα δεν κατάφεραν να βελτιώσουν την απόδοση του χαρτοφυλακίου που στηρίχθηκε στο elastic net. Οι λόγοι για αυτό ίσως ήταν η χαμηλή συχνότητα των δεδομένων που είχαμε στη διάθεση μας, η ελλιπής παραμετροποίηση του δικτύου ή η απουσία μη γραμμικών σχέσεων ανάμεσα στις εξαρτημένες και τις ανεξάρτητες μεταβλητές μας.

https://doi.org/10.26219/heal.aueb.6658

ACKNOWLEDGMENTS

I would like to thank my professor and supervisor Dr. Ioannis Vrontos. His critical help, his guidance and his support were extremely valuable and necessary for the completion of my work. I would also like to express my gratitude to the whole academic faculty of the MSc in Statistics for the high-level knowledge and expertise they offered me. Last but not least, i would like to thank my family for supporting me in every possible way.

https://doi.org/10.26219/heal.aueb.6658

List of Tables

https://doi.org/10.26219/heal.aueb.6658

List of Figures

https://doi.org/10.26219/heal.aueb.6658

Table of Contents

 https://doi.org/10.26219/heal.aueb.6658

1. Introduction

In this project we perform a comparative analysis of machine learning (ML) techniques for empirical finance. Although a huge part of the finance literature investigates the aggregate stock market return predictability, also called as equity risk premiums, we will examine stock return predictability along industry lines. Typical studies of industry return predictability use the most widely known predictors such as dividend yields, nominal yields and yield spreads (Avramov 2014, Ferson and Harvey 1991, 1999 and others). As Goyal (2008) argues most readers believe that prediction works with variables such as dividend-price ratios, earnings-price ratios, dividend-earnings ratios, and an assortment of other financial indicators. Here, we use a different set of predictor variables, which we name as lagged industry returns across the entire economy.

The two main strands of this project are the industry returns predictability and machine learning. Conventional forecasting is conducted either using time series predictions or cross-sectional predictions. Cross sectional models interpret differences in expected returns across stocks based on stock-level characteristics, namely as firm characteristics, such as value, momentum and size (Fama and French 2015, Lewellen, 2015). They most frequently use the Fama-McBeth (1973) regressions[1]. On the other hand, conventional time series models use macroeconomic factors such as interest rate spreads, book to market ratio, price to earnings ratio and others in order to predict aggregate portfolio returns. In this application we will use a time series framework, since our goal is to explain and forecast individual industry returns based on lagged industry returns from the entire economy. We use a general linear predictive framework, which enables each industry's return to be affected of the lagged returns of other industries. We selected to use this kind of data since firm characteristics data are difficult to attend and we wanted to present a more public and fee free investment strategy.

[1] More information about Fama- McBeth regressions can be found here http://didattica.unibocconi.it/mypage/dwload.php?nomefile=fama-macbeth20141115121157.pdf

https://doi.org/10.26219/heal.aueb.6658

This study is one of the first to directly forecast industry return predictability using this kind of predictors (only Rapach and Strauss, 2018 has done a similar one). Our study was inspired by the model of Hong (2007), who propose information frictions into an economy with multiple linked industries (Rapach, 2018). For example, cash flows shift in one industry can affect expected cash flows to all the industries, which are related to this one. Moreover, when a particular industry faces a cash flow shock, investors specialized to this industry are not capable of fully working out the implication of this shock. As a result, the delayed adjustment in equity prices highlights the benefits of industry return predictability based on lagged returns.

The lack of previous research is maybe contributed to the vast number of predictors used and so to the statistical challenges that arise for the estimation of the regression models. As the dimension of the predictors rises, simple linear models using the least squares objective function are susceptible to overfitting, and thus undermining the robustness and stability of the predictions. In addition, it is impossible to figure out in advance which of the predictors are the most important. Furthermore, since financial data are always noisy, nonlinear relationships between the excess returns of each industry and the lagged industry returns maybe exist. Last but not least, measuring the conditional expectation of a future realized excess return (risk premium) is a prediction task. Machine learning is perfectly suited for prediction tasks and so is ideal for our application. For these reasons, we combine the empirical asset pricing literature with the machine learning field.

Many researchers, see for example, Kelly (2020), Freyberger (2018), Feng (2017), Otto (2020), Wenbu and Wu (2019), have used ML to identify the relevant determinants of cross-sectional returns. Although our goal is not to identify the most important predictors of the cross-sectional excess returns, we share an identical approach of the ML algorithms used by the aforementioned authors in order to defend against the high-dimensionality of our data.

Our empirical analysis follows closely that of Rapach (2018), with several main differences. Firstly, we use more data since our data extent from 12/1959-12/2017. Secondly, we employ some additional ML algorithms, such as Elastic Net and Neural Networks. Thirdly, we shed more light on the bactesting simulation and the walk forward cross validation. Finally, we don't follow the conventional

 https://doi.org/10.26219/heal.aueb.6658

procedure of standardizing our predictor variables, since the predictors are expressed in identical units and we observe that unstandardized data perform better.

More specifically we use a simple linear regression which uses all predictors and estimates their coefficients using the OLS objective function as our benchmark model. Then, we use the least absolute shrinkage operator (LASSO) introduced by Tibshirani (1996). In contrast to ridge regression, which uses an l_2 parameter penalization, LASSO uses an absolute value (or l_1 penalization) and thus drives coefficients on a subset of covariates to exactly zero. So, LASSO performs variable selection whereas Ridge variable shrinkage. The advantage of LASSO is that leads to sparse models. Sparsity is desirable, since it prevents overfitting and identifies the most statistically significant predictors. The drawback of LASSO is that tends to overshrink the coefficients of the remaining predictors, which causes downward biases in the estimated coefficients. For this reason, researchers (Efron et al., 2004; Meinshausen, 2007; Belloni and Chernozhukov, 2011, 2013) suggest firstly to select the most important variables and then to execute an OLS regression to re-estimate the final coefficients in order to reduce the biases. The above procedure is named after on as OLS post-LASSO. In the same context we also perform the elastic net penalty. For complicity reasons, we check whether the OLS-post LASSO (OLS post-elastic net) performs better than pure LASSO (elastic net). We estimate 30 regression models at each time step, each one for every industry, using the lagged returns for all the 30 industries respectively.

Both in-sample and out of sample analysis is conducted in order to emphasize that lagged returns can help in the prediction of the excess return of each industry. Regarding the in-sample analysis, since we test a large number of individual null hypothesis, we face the problem of multiple testing. In order to deal with this, we apply the Benjamini and Hochberg (2000) procedure to examine the false discovery rate.

Our empirical analysis investigates the excess returns of 30 industry portfolios provided by the Kenneth French’s Data Library and ranging from December 1959 to December 2017. We assign each NYSE, AMEX, and NASDAQ stock to a value weighted industry portfolio at the end of June of year t based on its four-digit SIC code at that time. (We use Compustat SIC codes for the fiscal year ending in calendar

https://doi.org/10.26219/heal.aueb.6658

year t-1. Whenever Compustat SIC codes are not available, we use CRSP SIC codes for June of year t.) We then compute returns from July of t to June of t+1. We show that at least one lagged industry return is chosen by LASSO as a predictor for 29 out of 30 industries and more than once for 22 out of 30. Secondly, OLS post-LASSO proves that the predictors selected by the LASSO are most of the times statistically significant even after checking for multiple testing. The results of our analysis indicate that when we aggregate our data with 4 popular macroeconomic variables proposed by Ferson and Harvey (1991, 1999), Ferson and Korajczyk (1995), and Avramov (2004), LASSO selects exactly the same predictors as before and so the lagged returns maintain their predictive ability. In conclusion, apart from the statistical significance we also evaluate the economic significance of the predictors mainly in terms of annualized Sharpe Ratio.

Interpreting the results for the in-sample analysis, we can argue that lagged returns of financial, commodity and material-producing sectors are selected most often across the industries. This result was expected at some extent. When the financial sector experiences a positive shock, they obtain larger capital and so can provide bigger credit to companies belonging in other industries. In contrary, when commodities experience a positive shock, we expect a raise in the prices and the returns of industries belonging in the earlier stages of the production process. As a result, profits and returns of the later stage industries are decreased. As expected, the coefficients of lagged commodities-materials returns are negatively correlated with the returns of the later stages industries.

Out of sample analysis corresponds to the selection of an economically meaningful portfolio of industries for investment in real time. In order to build our investment portfolio, we use the forecast of monthly excess industry returns calculated by the aforementioned models. We then sort the 30 industries into 5 quantiles (each quantile contains 6 industries) according to the forecasted returns over the next month. After that, a zero-net investment portfolio is constructed, which goes long (short) for the top (bottom) 20% quantile of sorted industries. For this purpose, we use the walk forward cross validation procedure because of the time series type of our data. The training data set extends from 12/1959-12/1969. So, in each time step we compute the forecasted returns for one month ahead. The simple linear model generates an annualized mean of 2.73% and annualized Sharpe Ratio of 0.50%, the

 https://doi.org/10.26219/heal.aueb.6658

OLS post-LASSO 3.603% and 0.674%, and the Elastic Net 4.534% and 0.707%, respectively. We also figure out that the portfolio constructed using the penalized methods perform efficiently during recession times of the whole economy and so these portfolios could be used as a hedge against downswings of the macroeconomy.

As the last part of our project we expand our models in order to accommodate nonlinear relationships via neural networks. As mention by Kelly (2020), allowing for nonlinearities and potentially complex interactions among the predictors substantially improves predictions. We employ shallow learning instead of deep learning since we have a narrow range of data. We consider several different architectures of networks and insert many different regularization techniques such as learning rate shrinkage via the ADAM algorithm (a version of Stochastic Gradient Descent), batch normalization and early stopping. The tuning of the hyperparameters was conducted via cross validation. Although we structured our network in the most optimized way in our opinion, we didn't derive greater returns in terms of the financial indicators mentioned above. As a result, the portfolios created by the neural networks predictions didn't add any value in comparison with the penalized methods. This maybe occurred due to the low frequency of our data (maybe daily or weekly data instead of monthly could have performed better) or due to the absence of nonlinearities between the lagged industry returns. Even when we added the squared values of the lagged returns and some interactions between the predictors in our data set, we didn't receive better results. Perhaps neural networks are just not the most suitable model for this particular application.

The rest of the thesis is structured as follows. In Section 1.2 we present a brief literature review about asset pricing in empirical finance via machine learning. In Section 2 we describe the predictive framework of the models mentioned above and we explain the LASSO and Elastic Net regularization methods. In Section 3 we demonstrate the in-sample results, whereas in Section 4 the out-of-sample results and the methodology followed to build our bottom up portfolios. In Section 5 we focus on the Neural Networks and the procedure followed to optimize them. Section 6 provides some concluding remarks.

https://doi.org/10.26219/heal.aueb.6658

1.2. Literature Review

We follow pretty closely the paper of Rapach, Strauss, Tu, and Zhou (2018). They demonstrate that LASSO helps to identify the most significant lagged industry excess returns and create an economically meaningful long short portfolio based on the forecasted returns estimated by OLS using the predictors that LASSO selected. Freyberger, Neuhierl, and Weber (2020) apply a non-parametric adaptive group LASSO version to determine the most valuable characteristics in terms of predictive power for the cross-section of expected returns. Rapach, Strauss, and Zhou (2013) also use the LASSO framework to investigate the lead-lag relationships among monthly country-level stock returns. We also gained insight by the work of Gu, Kelly and Xiu (2020). They illustrate wonderfully the most important and up to date ML techniques (linear model, penalized linear model, PCA, Boosted Regression Trees, Neural Networks) in order to measure asset risk premiums. They compare all these forecast models and conclude that shallow neural network model is the most effective. Similar to the later work, is the research of Drobertz and Otto (2019), which was also very helpful to my study.

Apart from the penalized linear models, Giglio and Xiu (2016) conduct a principal component analysis (PCA) in order to deal with the high dimensionality of their data. Kelly, Pruit and Su (2019) apply also a version of PCA to re-estimate and test factor pricing models. Moritz and Zimmermann (2016) introduce tree-based models to portfolio sorting.

Several authors apply neural networks to predict derivative prices (Yao, Li, and Tan, 2020 among others). Messmer (2017) explores the predictability of firm characteristics on stock returns via neural networks. Heaton, Polson, and Witte (2016) use neural networks for portfolio selection. All of the above authors used feed forward neural networks. Gu, Kelly, and Xiu (2019) include different architectures such as recurrent long short-term memory (LTSM) networks, generative artificial networks (GAN) and autoencoder networks.

Finally, some researchers study the sign rather than the stock return. Because of the binary nature of this problem, classification methods (linear discriminant analysis, logit, probit) are applied to predict the direction (or the signal) of the stock

 https://doi.org/10.26219/heal.aueb.6658

return. In the same spirit, Huerta, Corbacho, and Elkan (2013) build a support vector machine (SVM) to distinguish stocks that overperform form those which underperform.

2. Linear regression framework and regularization

This section explains the basic predictive regression framework that we follow in our analysis and the general functional form for estimating each industry's excess return. First of all, the simple return of an asset (e.g. stock, bond, ETF, etc), assuming no dividends, is the percentage change over two time periods (in our case between two months). Mathematically, it can be written as:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1,$$

where P_t denotes the price of the asset at month t and the time between t and t-1 is called the holding period. The continuously compounded monthly return (which we use later on) is defined as:

$$r_t = ln(\frac{P_t}{P_{t-1}}),$$

We refer to excess return, as the return of each industry when the one-month U.S.A Treasury bill return (the risk-free rate in general) is subtracted. Most of the times in order to conduct comparisons between different investment models, we use the annualized returns instead of the simple net return. In the simplest case that our investment horizon is one month and we receive the same return each month, the annualized return corresponds to the monthly compounded return for 12 months and can be expressed as:

$$r_A = (1 + r)^{12} - 1.$$

If the return we receive each month is not constant and n is the number of years the investment took place, then the annualized rate of return each year over a given time period can be written as:

$$r_A = [(1 + r_1)(1 + r_2)(1 + r_3)\dots(1 + r_n)]^{12/n} - 1.$$

 https://doi.org/10.26219/heal.aueb.6658

Another term we used in our analysis is the annualized volatility. It is the standard deviation of the industry portfolio excess returns multiplied by the square root of 12 (number of months in each year). In addition, one of our main criterions for the model selection is the Sharpe Ratio. Sharpe (1996) proposed the measure as:

$$S_i = \frac{E[R_i] - r_f}{\sigma_i} = \frac{\overline{R_\iota} - r_f}{\sigma_i},$$

where σ_i is the standard deviation of the investment, r_f is the risk-free rate and $\overline{R_\iota}$ is the expected return of an investment. In general, Sharpe Ratio is an investment measurement, which calculates the average return of the investment beyond the risk-free rate per unit of volatility. So, the higher the ratio, the more profitable is the investment. Investments with a Sharpe Ration beyond one are considered extremely efficient. Lastly, we observe the maximum drawdown of each method. Maximum drawdown expresses the maximum observed loss from a peak to a trough of the portfolio, before a new peak is attained. It basically defines the downside risk over a specific time period. We define the maximum drawdown of a strategy as:

$$MaxDD = \max_{0 \le t_1 \le t_2 \le T} Y_{t_1} - Y_{t_2},$$

where Y_t is the cumulative log return from date 0 through t.

In the most general form, an industry's excess return is given by an additive regression model as:

$$r_{i,t+1} = E(r_{i,t+1}) + \varepsilon_{i,t+1},$$

where

$$E(r_{i,t+1}) = g^*(z_{i,t}).$$

We assume that $g^*(\cdot)$ is a flexible (either linear or nonlinear) function of the predictors, i=1,2,..,N is the number of industry portfolios, t=1,2,…,T is the number of monthly observations we use to estimate the parameters of our model and $r_{i,t}$ is the excess return of portfolio i at time t. The linear model specification is explained below:

$$\mathbf{y}_i = a_i \boldsymbol{\iota}_T + \mathbf{X}\mathbf{b}_i + \boldsymbol{\varepsilon}_i, \quad (2.1)$$

where

 https://doi.org/10.26219/heal.aueb.6658

$$\mathbf{y}_i = [r_{i,1} \dots r_{i,T}]', \quad (2.2)$$

$$\mathbf{X} = [x_1 \dots x_N]', \quad (2.3)$$

$$x_j = [r_{j,0} \dots r_{j,T-1}]', \text{ for } j = 1,..,N, \quad (2.4)$$

$$b_j = [b_{i,1} \dots b_{i,N}]', \quad (2.5)$$

$$\varepsilon_j = [\varepsilon_{i,1} \dots \varepsilon_{i,T}]', \quad (2.6)$$

$\boldsymbol{\iota_T}$ is a T-vector of ones and $\boldsymbol{\varepsilon}_i$ is an error term following the Standard Normal distribution. In equation (2.1) excess lagged returns of all industries across the economy can interfere with the excess return of any given industry. In a way, we can think (2.1) as a first-order vector autoregression for all N industry portfolios excess returns. We observe that as each time we estimate 30 different excess return (one of each economy sector) and each one of these has 30 predictors. So, the problem of high dimensionality in our application is obvious. As we mentioned before, a traditional linear predictive regression model via OLS underperforms in this kind of applications and tends to overfit. Nevertheless, we use it as our benchmark model. The linear model approximates the conditional expectation $g^*(\cdot)$ by a linear function of raw predictors. The conventional objective function (or l_2 penalization) used by the linear regression which we want to be minimize can be expressed as:

$$\underset{ai\in\mathbb{R}}{\operatorname{argmin}}\left(\frac{1}{2T} || \mathbf{y}_i - a_i \boldsymbol{\iota_T} - \mathbf{X}\mathbf{b}_i ||_2^2\right) \quad (2.7)$$

The advantage of equation (2.7) is that it avoids complicated optimization and the estimated coefficients come with low computational cost. On the other hand, when the number of predictors approaches the number of observations, the simple linear model starts to overfit noise rather than extracting signal. In order to deal with this, we have to reduce the number of predictors and this can happen if we add a penalty term to the objective function in order to seek for more parsimonious predictors. In equation (2.8) we present the same equation as (2.7) accompanied by the famous elastic net penalty term.

$$\underset{a_i, b_i\in\mathbb{R}}{\operatorname{argmin}}\left(\frac{1}{2T} || \mathbf{y}_i - a_i \boldsymbol{\iota_T} - \mathbf{X}\mathbf{b}_i ||_2^2 + \lambda(1-\rho)\sum_{i=1}^{N} |b_{i,j}| + 0.5\lambda\rho\sum_{i=1}^{N} b_{i,j}^2\right) \quad (2.8)$$

https://doi.org/10.26219/heal.aueb.6658

The elastic net equation contains two hyperparameters λ and ρ. When ρ=0 equation (2.8) transforms to the popular LASSO or l_1 penalization. The geometry of LASSO forces the unimportant covariates to exactly zero and thus improves the sparsity of the model. When ρ=1 equation (2.8) transforms to ridge regression or an l_2 penalization. Ridge doesn't perform variable selection but variable shrinkage. It leads the unimportant coefficients towards zero, but not exactly to zero. Ridge prevents unimportant coefficient to affect largely in magnitude the dependent variable. Since one of the primary goals of this project is to reduce the dimension of the predictor data set, we will execute only the LASSO and elastic net penalizations.

In general LASSO works well in terms of choosing the most important predictors. One disadvantage of LASSO is that tends to overshrink the parameters in terms of magnitude. To avoid this, several researchers (Effron, 2014; Meinhausen, 2007) suggest to re-estimate the coefficients for the remaining post-LASSO predictors via OLS.

Another problem is that when we test numerous individual null hypotheses, typical p-values can be misleading for the actual statistical significance. This issue is called multiple testing and we face it using the familiar Bonferroni and Hochberg (2000) procedure. They try to control the false discovery rate (FDR). In practice, this procedure puts the m adjusted p-values and their m null-hypothesis into an ascending order. In our application m=167, since 167 is the number of predictors chosen by LASSO. Then, we reject the first k hypothesis in respect to:

$$k = \max_{j}\{p_{(j)}^{BH} \leq q\}$$

where q is the significance level and $p_{(j)}^{BH} = \frac{m}{j}\, p_{(j)},\ \ p_{(j)}$ is the unadjusted p-value of null hypothesis j. We performed the Bonferroni and Hochberg (2020) correction using the "multipletests" module in python.

In order to execute LASSO, we have to optimize the hyperparameter λ. This can be accomplished either by using k-fold cross validation or by using the AIC, BIC or AICc criterion. Most of the times when we have high-dimensional data sets with many collinear features we use the LassoCV module. Because of the serial structure of our data and their relatively small size, we decided to use AIC criterion via the python sklearn LassoLarsIC module. AIC generally selects the most efficient model

 https://doi.org/10.26219/heal.aueb.6658

in terms of the l_2 loss criterion for predictive performance. This method is computationally cheaper, because the regularization path for the optimization of λ is computed only once instead of k+1 times as in k-fold cross validation. The mathematical type of AIC under a multiple linear regression framework via OLS using LASSO regularization is given as:

$$AIC_{OLS} = N ln(\frac{\frac{1}{2T}|| \mathbf{y}_i - a_i \boldsymbol{\iota}_T - \mathbf{X}\mathbf{b}_i ||_2^2 + \lambda \sum_{i=1}^{N} |\mathrm{b}_{i,j}|}{N}) + 2(p+1),$$

where N=30 and p is the total number of non-zero predictors under each λ value.

Even though in this application we perform a special case of the conventional cross validation (named as walk forward cross validation), we would like to explain in more depth the general notion of sample splitting and hyperparameter tuning via validation for complicity reasons. The general idea is to perform the hyperparameter tuning based on a validation sample of our data. We divide the data set into three subsets, while maintain the temporal order of the data. The first sample, called training sample, is used for the estimation of the parameters of each model for some specific values of the tuning parameters. The second subsample, is called validation sample, and is used in order to choose the most efficient values of the tuning parameters based on the estimated model by the training data set. To be more specific, we make forecasts based on the estimated model using a set of the hypermeter. Then we measure the value of an objective function and iteratively change the set of hyperparameters until we minimize our objective function (at each iteration we re-estimate the model from the training sample using the next set of hyperparameters). The third subsample, called testing sample, evaluates the predictive ability of our model. The test sample is used neither for the estimation of the model parameters nor for tuning the hyperparameters and as a result is really out of sample.

3. Neural Networks

As the last part of this research, we included in our study the most popular and powerful nonlinear method in the machine learning toolkit, the artificial neural

 https://doi.org/10.26219/heal.aueb.6658

networks (ANN). They are currently applied mainly in applications such as computer vision, speech recognition and automated game playing. They have earned the synonym deep learning, because they impose nonlinear interactions between the predictor by entwining several telescoping layers. On the other hand, they are really complicated since many parameters needs to be optimized, the computation cost is very large and the interpretation of their findings is not straightforward, since the undermining processes occurring in the network cannot be fully explained and thus can be characterized as a black-box procedure. In this empirical study, we employ some neural networks in order to check the presence of nonlinear relationships between the lagged industry excess returns, aiming to increase the economic value of the constructed long industry rotation portfolios.

Our analysis concentrates on “feed forward” neural networks. Following the functionality of a human brain, artificial neural networks are made of many interacting computational units, called “neurons”. A group of neurons represents a hidden layer in machine learning terminology.

In feed forward neural networks, the output of each node (neuron) in each hidden layer is connected to all the nodes of the next layer and these connections follow a one-way path, from input to output. In our study, the input layer consists of 30 predictors, which are the lagged industry returns. We then entail one or more hidden layers, which interact and nonlinearly transform the predictors. The final layer, which is called the output layer, aggregates hidden layers into an ultimate outcome prediction. The outcome prediction in our case is the dependent variable, which is the forecasted excess return for each one of the industries. Since we want to predict the returns for all the 30 industries in each time step, we run simultaneously 30 regressions and thus the output layer consists of 30 outputs (one forecast for each industry). In comparison with the human brain the hidden layers represent groups of “neurons”. Each layer is connected with the next one by “synapses”, which transmit the signal.

Figure 1 shows the simplest neural network possible. It contains no hidden layers and input layer represents the lagged industry returns. Each ones of the inputs are multiplied by a weight parameter w_i. We sum all the weighted inputs and add an intercept, which is called bias. Then we insert this sum into a function f, which is called activation function. If we use a linear activation function, the network depicted

https://doi.org/10.26219/heal.aueb.6658

in Figure 2 represents the simple linear regression. We run the backtest simulation using the simplest neural network and we confirmed that the results matched pretty closely the results derived from OLS model mentioned above.

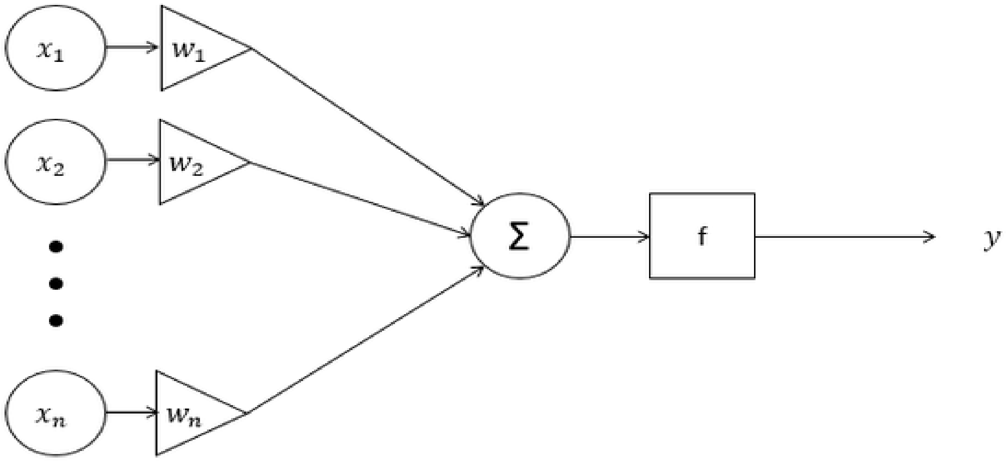


Figure 1: Simplest neural network, equal to linear regression.

In Figure 2 we present an example of a network with one hidden layer and 3 ultimate outcome predictions. The output layer doesn't consist of one output (equivalent to one prediction), but of three outputs and so three predictions. This case resembles better our application, since our outcome layer consists of thirty predictions.

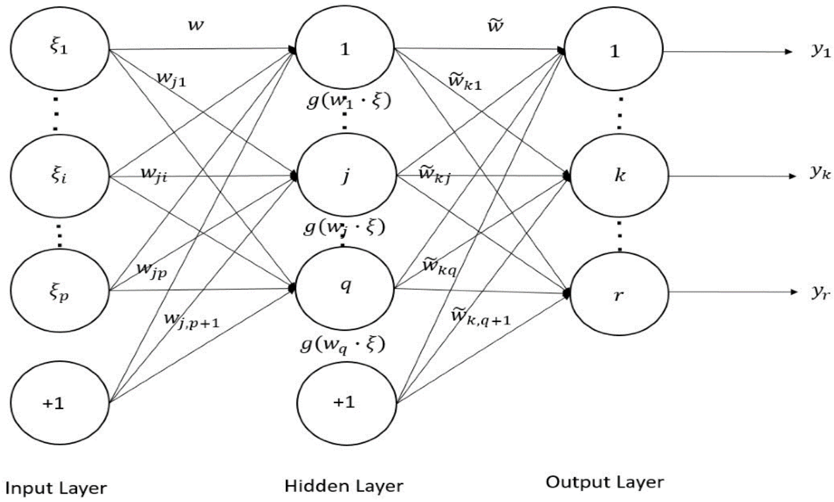


Figure 2: Neural network with one hidden layer and an output layer of 3 predictions.

 https://doi.org/10.26219/heal.aueb.6658

In more complex architectures (with one or more hidden layers) we have to define many parameters such as the number of hidden layers, the number of neurons in each hidden layer and which unit we want to interconnect. Recent research in empirical finance applications has shown that is preferable to train simple networks, with a limited number of parameters instead of deep networks. Deep networks require a large number of parameters and more importantly millions of data points, which is infeasible in financial applications.

Selecting the optimum architecture for a network is impossible because the number of possible combinations is immense. Thus, we set a fix number of parameters in advance and select the best performing network by cross validation among these parameters. The ultimate goal is to construct an even more profitable portfolio than the portfolio created above using the elastic net forecasts.

We choose to examine networks with up to 3 hidden layers. The number of neurons in each layer is decreased according to the geometric rule (see, Masters, 1993). For example, we denote the network with one hidden layer and 32 neurons as NN1. NN2 denotes the network with two hidden layers, 32 and 16 neurons in the first and second layer respectively. NN3 is a network with 3 hidden layers and 32, 16, and 8 neurons in each layer respectively. By comparing the different architectures, we define the trade of between the network's depths and its performance. Our networks are fully connected and thus every output of a neuron is inserted in all the neurons of the next layer. In terms of the activation function, we use the well-known rectified linear unit (ReLU) function for all nodes. ReLU can be defined as:

$$ReLU(x) = \begin{cases} 0 & if\ x < 0 \\ x & otherwise \end{cases}$$

ReLU activation function is linear for values greater than zero, meaning it allows for faster derivative calculation in the backpropagation algorithm. Yet, it is still a nonlinear function as negative values are always output as zero. The loss function we minimize in order to compile the network is MSE.

The algorithm used for weights estimation is called backpropagation. In backpropagation, we first initialize the weights by randomly generating values from a uniform distribution. We also measure the mean square forecast error (MSFE) based on the forecasts coming from the outputs of the last hidden layer. Then, following the

https://doi.org/10.26219/heal.aueb.6658

opposite direction (from output to input) the algorithm computes the gradient of MSFE with regard to the weights. Then, the weights are adjusted in the opposite direction of the computed gradients in order to minimize the MSFE. Finally, using the adjusted weights, the MSFE is recalculated. When the MSFE is minimized this process (known as gradient descent) stops.

We used the stochastic gradient descent algorithm (SGD) to estimate our weights. SGD evaluates the gradient from a small random subset of the data at each iteration, instead of the whole training sample as conventional gradient descent does. Each one of the random subsets of the data is called a batch. We set the number of batches to 128. The total number of times SGD "sees" the entire training data set is called epochs. We set the number of epochs to 1000. Strangely, when we tried to increase the number of epochs in order to improve the convergence of the algorithm, we obtained both worse results and extra computational cost.

Apart from the weight penalization, we employed several other regularization techniques to enhance the efficiency of the network. These methods are:

- Learning rate shrinkage: In SGD it is really important to control the step size of the descent. When gradient approaches zero, learning rate should be close to zero and reversely. Although we need most of the times small values of the learning rate, we want big values in the first iterations to speed up the process. For this reason, we use the famous ADAM (adaptive moment estimation) algorithm (Kingma and Ba, 2014) under the default values for the learning rate and momentum parameters used by the python module "keras". The mathematical approach of the ADAM algorithm is fully explained in Algorithm 1 below.

- Early Stopping: In each step of the optimization process, the parameters are updated to reduce MSFE in training sample. At each new guess, predictions are made for the validation set. When the validation error starts to increase, we stop the optimization routine. This is achieved before prediction error in training set is minimized. By terminating the process earlier, parameters are led towards the initial guess. In practice, we don't stop the iteration process exactly when the MSE in the validation sample start to increase, but after a

 https://doi.org/10.26219/heal.aueb.6658

prespecified number of iterations (so-called "patience"). We set the patience=5. The mathematical approach of the early stopping algorithm is fully clarified in Algorithm 2 below. For the neural networks, we decided to standardize the inputs (the predictors) in order to follow the "traditional" methodology for neural networks. In order to achieve comparability among the results from the neural networks and the results derived from the other techniques we also standardized the predictors for the linear methods.

- Batch normalization (Ioffe and Szegedy, 2015): It is possible that the inputs to the hidden layers across our network might change after the weights are updated. This modification in the distribution of the inputs between the hidden layers is named as "internal covariate shift". Batch Normalization controls the "internal covariate shift" (the variability of predictors across different regions of the network). For each hidden layer (including the input layer) in each training step (a "batch"), the algorithm cross-sectionally normalizes (re-centers and re-scales) the batch inputs to restore the representation power of the unit. Batch Normalization accelerates training and reduces generalization error. The detailed mathematical approach of the batch normalization algorithm is fully described in Algorithm 3 below.

- Dropout: This is the last regularization tool we use. It probabilistically removes inputs to a layer, which can be input variables from the initial data set or activations from a previous layer. It creates a large number of networks with different architectures and as a result makes the nodes of the network more robust to the inputs. The Dropout layer is added to a model between existing layers and applies to outputs of the prior layer that are fed to the subsequent layer. This helps us to prevent overfitting. We set the dropout rate to 0.2.

Now, we present the mathematical illustration of the algorithms used for the neural network construction. Algorithm 1 corresponds to ADAM (in the same manner as Kingba and Ma, 2014), Algorithm 2 corresponds to the early stopping and

 https://doi.org/10.26219/heal.aueb.6658

Algorithm 3 corresponds to the Batch Normalization (in the same manner as Ioffe and Szegedy, 2015).

As we mentioned before, a widely used algorithm to fit a neural network is the stochastic gradient descent (SGD). In our application we used a specific version of the SGD called "adaptive moment estimation algorithm or ADAM". ADAM calculates adaptive learning rates by taking advantage of the first and second order moments of the gradients. Below we present the mathematic illustration of the algorithm. The loss function is denoted as $L(\theta;\cdot)$ and $L(\theta;\cdot) = \frac{1}{T}\sum_{t=1}^{T} L(\theta;\cdot)$, where $L(\theta;\cdot)$ is the penalized cross-sectional average prediction error at month t. Algorithm 1 explains the ADAM algorithm in the same manner as Kingma and Ba (2014) do in their analysis. During each step of the training, a randomly selected batch is inserted to the algorithm. In Algorithm 2 is presented the Early Stopping algorithm, which is used in combination with ADAM. Algorithm 3 performs the algorithm of Batch Normalization presented by Ioffe and Szegedy (2015), which is applied to each activation function after ReLU transformation.

Algorithm 1: ADAM for stochastic optimization. Default parameters are $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. By writing β_1^t, β_2^t we denote β_1, β_2 to the power of t.

Require: $\beta_1, \beta_2 \epsilon [0,1)$: Exponential decay rates for the moment estimates

Require $L(\theta;\cdot)$: Stochastic objective function with parameters θ

Require: θ_0: Initial parameter vector

$m_0 \leftarrow 0$ (Initialize 1st moment vector)

$v_0 \leftarrow 0$ (Initialize 2nd moment vector)

$t \leftarrow 0$ (Initialize timestep)

while θ_t not converged do

$t \leftarrow t+1$

$g_t \leftarrow \nabla_\theta L(\theta;\cdot)|_{\theta=\theta_{t-1}}$.

https://doi.org/10.26219/heal.aueb.6658

$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$.

$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t \odot g_t$.

$\widehat{m_t} \leftarrow m_t / (1-(\beta_1)^t)$.

$\widehat{v_t} \leftarrow v_t / (1-(\beta_2)^t)$.

$\theta_t \leftarrow \theta_{t-1} - \alpha \widehat{m_t} / (\sqrt{\widehat{v_t}} + \varepsilon)$.

end while

return θ_t (the final resulting parameter)

Algorithm 2: Early Stopping algorithm. Each neuron that received as a signal a batch of x, now receives $BN_{\gamma,\beta}(x)$ instead. Parameters γ and β need also to be tuned.

Initialize $j = 0$ $\varepsilon = \infty$ and select the patience parameter p.

While $j < p$ do

Update θ using the training algorithm (e.g., the steps inside the while loop of Algorithm 5 for h steps).

Calculate the prediction error from the validation sample, denoted as 0.

If $\varepsilon' < \varepsilon$ then

$j \leftarrow 0$

$\varepsilon' \leftarrow \varepsilon$

$\theta' \leftarrow \theta$

else

$j \leftarrow j + 1$

end

end

https://doi.org/10.26219/heal.aueb.6658

Result: The final parameter estimated is θ'.

Algorithm 3: Batch Normalization (for one Activation over one Batch)

Input: Values of x for each activation over a batch $B = \{x_1, x_2, x_3, \ldots, x_N\}$

$$\mu_B \leftarrow \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma_B \leftarrow \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu_B)^2$$

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{{\sigma_B}^2 + \varepsilon}}$$

Result: $\{y_i \leftarrow \gamma\widehat{x_i} + \beta := BN_{\gamma,\beta}(x) : i - 1,2,\ldots,N\}$.

4. Empirical Application

4.1. Data and in sample modeling

We obtain monthly excess industry returns from Kenneth French's Data Library for thirty value-weighted industry portfolios. The classification of the industries is based on the Stand Industrial Classification (SIC) system and our data range from 12/1959-12/2017. Using data from such a wide time span, makes the training of our algorithms easier and reduces the illiquidity and trading worries.

In Table 1 we perform some descriptive statistics for industry excess returns such as annualized returns, annualized volatilities, annualized Sharpe ratios, skewness, kurtosis and the maximum and minimum returns throughout these years. We observe that the smoke industry has the biggest annualized returns and corresponding Sharpe ratio with 10.12% and 0.49 respectively. In contrast, steel industry has the lowest annualized return and Sharpe ratio with 0.59% and 0.02,

https://doi.org/10.26219/heal.aueb.6658

respectively. The maximum return is observed for Textiles (59.03%) and the minimum for smoke (-38.09%). Kurtosis is around 4 and 6 for all industries, except Textiles (around 12). Since the mean kurtosis is greater than 3 (corresponds to the normal distribution), the industry portfolios have a leptokurtic distribution and so greater extremity of deviations (or outliers) than the normal distribution. The skewness of almost all the returns is negative, but really close to zero, so the tails on both sides of the distribution of the industry returns balance out.

In Table 2 we report the coefficient estimators after we employed the OLS post-LASSO procedure. Since we take into consideration the order one lagged predictors our data set expands from 01/1960-12/2017. We use a bold (italicized) mark to indicate the statistically significant coefficients at the 10% (5%) level using a simple t-statistic. We generally observe that LASSO reduces the number of lagged predictors from a total of 900 to 167. It also selects at least one lagged predictor for the vast majority of individual industries (29 out of 30). For 22 out of 30 industries LASSO selects more than one lagged predictor. According to t-statistic, 82 (53) out of 167 selected coefficients are statistically significant at 10% (5%) level. In addition, an industry's own lagged return is chosen only for seven industries, and so we can argue that autocorrelation has a modest impact.

Some easily noticed conclusions from Table 2 is that the financial (Fin) industry (Banking, Insurance, Real Estate and Trading firms) lagged excess return is a statistically significant predictor for most of the other industries (for 18 out of 30 industries to be exact). It is also worth-mentioning that the coefficient is always positive, which mean that when the financial sector performs well in terms of profit, all the other industries are expected to perform also well. Financial firms give credit to firms belong in other industries and so the whole economy gains an economic "boost". On the other side, when financial firms have negative returns, the whole economy lacks in financing and as a result excess returns of firms across the entire economy are expected to fall. The largest in magnitude coefficient estimate for lagged Fin comes for Textiles (Txtls) sector, and so this is the most severely affected sector from financial companies.

https://doi.org/10.26219/heal.aueb.6658

	Ann. Return%	Ann. Volatility%	Skewness	Kurtosis	Sharpe Ratio	Min%	Max%
Food	7.39	14.92	-0.026	5.27	0.49	-18.15	19.89
Beer	7.43	17.52	-0.06	5.50	0.42	-20.19	25.51
Smoke	10.12	20.87	-0.06	5.48	0.48	-25.32	32.38
Games	5.83	24.69	-0.27	5.45	0.23	-33.40	34.52
Books	4.46	20.02	-0.025	5.50	0.22	-26.56	33.13
Hshld (Goods)	5.57	16.37	-0.32	4.71	0.34	-22.24	18.22
Clothes	5.97	22.02	-0.076	5.60	0.27	-31.50	31.79
Healh	6.75	16.96	0.007	5.34	0.39	-21.06	29.01
Chems	4.92	18.99	-0.12	5.28	0.25	-28.60	21.68
Textiles	5.76	24.19	0.50	12.76	0.23	-33.11	59.03
Cnstr	4.66	20.64	-0.21	5.37	0.22	-28.74	25.02
Steel	0.59	25.16	-0.15	5.22	0.02	-32.99	30.30
FabrPr	5.09	21.05	-0.36	5.46	0.24	-31.63	22.91
ElcEq	6.50	21.36	-0.18	4.62	0.31	-32.80	23.20
Autos	3.08	23.00	0.24	9.12	0.13	-36.49	49.56
Carry	6.94	21.66	-0.32	4.44	0.32	-31.10	23.39
Mines	3.51	25.72	-0.15	4.73	0.13	-34.55	35.15
Coal	2.71	35.17	0.18	5.12	0.07	-38.09	45.55
Oil	6.16	18.48	0.03	4.27	0.33	-18.97	23.70
Util	5.03	13.75	-0.14	4.02	0.36	-12.94	18.26
Telecom.	5.03	15.96	-0.17	4.15	0.31	-16.44	21.22
Services	6.20	22.44	-0.15	4.32	0.27	-28.67	23.38
BusEq	4.78	23.19	-0.31	4.84	0.20	-32.07	24.66
Paper	4.95	17.38	-0.14	5.55	0.28	-27.74	21.00
Trans	5.38	19.78	-0.24	4.25	0.27	-28.50	18.50
Wholesale	5.73	19.28	-0.35	5.20	0.29	-29.25	17.53
Retail	6.71	18.42	-0.22	5.21	0.36	-29.74	26.49
Meals	6.67	21.01	-0.49	5.51	0.32	-31.89	27.38
Finance	5.95	18.62	-0.37	4.66	0.32	-22.53	20.59
Other	2.55	20.01	-0.38	4.65	0.13	-28.09	19.96

Table 1: Descriptive statistics of individual industries excess returns from 12/1959-12/2017 for the 30 value weighted industry portfolios. The shortcuts for the dependent variable correspond to: Food = Food Products; Beer = Beer and Liquor; Smoke = Tobacco Products; Games = Recreation; Books = Printing and Publishing; Hshld = Consumer Goods; Clths = Apparel; Hlth = Healthcare, Medical Equipment, and Pharmaceutical Products; Chems = Chemicals; Txtls = Textiles; Cnstr = Construction and Construction Materials; Steel = Steel Works, Etc.; FabPr = Fabricated Products and Machinery; ElcEq = Electrical Equipment; Autos = Automobiles and Trucks; Carry = Aircraft, Ships, and Railroad Equipment; Mines = Precious Metals, Non-Metallic, and Industrial Metal Mining; Coal = Coal; Oil = Petroleum and Natural Gas; Util = Utilities; Telcm = Communication; Servs = Personal and Business Services; BusEq = Business Equipment; Paper = Business Supplies and Shipping Containers; Trans = Transportation; Whlsl = Wholesale; Rtail = Retail; Meals = Restaurants, Hotels, and Motels; Fin = Banking, Insurance, Real Estate, and Trading; Other = Everything Else.

Estimated Regression Coefficients

Dependent	***Fo***	***Beer***	***Smok***	***Game***	***Book***	***Hshld***	***Clths***	***Hlth***	***Chem***	***Txtls***
Food							0.04			
Beer	***0.1***						0.05			
Smoke										0.07
Games					***0.18***		0.05			
Books				0.03	0.04					
Hshld							**0.1**			
Clths					0.08		0.08		**0.16**	
Hlth					***0.1***					
Chems							0.08			
Txtls							0.08			
Cnstr							0.04			
Steel										
FabPr										
ElcEq										
Autos						***-0.3***	0.04			
Carry										
Mines										
Coal		***-0.27***	-0.09		0.13					
Oil		-0.08						***-0.13***		
Util	0.1	***-0.1***	0.02			-0.08		-0.08		
Telcm		-0.06	-0.03		0.09	-0.07				
Servs			**-0.09**		0.1					
BusEq			***-0.14***		***0.12***					
Paper							0.06			
Trans										
Whlsl	-0.1	-0.05	-0.05		***0.14***			-0.06		
Rtail										
Meals			-0.06		0.06		0.1			
Fin										
Other							0.07			
R^2(%)	2.2	2.51	6.58	5.05	6.3	2.95	7.86	2.69	0.76	7.92

Table 2: This table reports the OLS estimated coefficients in terms of magnitude and statistical significance after we have implied LASSO to our full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables for the OLS regression by LASSO. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Estimated Regression Coefficients

Dependent	***Cnstr***	***Steel***	***FabPr***	***ElcE***	***Autos***	***Carry***	***Mines***	***Coal***	***Oil***	***Util***
Food										**0.09**
Beer										-
Smoke						***0.17***	-0.02	-0.03	-0.1	**0.27**
Games										-
Books										**0.13**
Hshld										
Clths						0.06				**0.17**
Hlth							-0.06			**0.11**
Chems										
Txtls					**0.11**					
Cnstr										***0.15***
Steel										
FabPr										
ElcEq										
Autos										***0.17***
Carry										
Mines										
Coal								**0.08**	***-0.2***	
Oil						***0.17***				
Util	***-0.18***		**0.12**			**0.08**	-0.04			**0.09**
Telcm	-0.07				-0.04	-0.04	-0.01			***0.16***
Servs		**-0.08**								0.12
BusEq										***0.18***
Paper										
Trans										
Whlsl						0.07				***0.25***
Rtail										
Meals						0.04				0.12
Fin										
Other										
R^2(%)	5.13	1.3	1.56	0.8	6.13	2.27	-	2.85	2.52	7.85

Table 2 (continued) This table reports the OLS estimated coefficients in terms of magnitude and statistical significance after we have implied LASSO to our full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables for the OLS regression by LASSO. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

Estimated Regression Coefficients

Dependent	***Tlcm***	***Servs***	***BusEq***	***Paper***	***Trans***	***Whlsl***	***Rtail***	***Meals***	***Fin***	***Other***
Food							0.02			
Beer										
Smoke	-0.11	**-0.15**		**-0.19**	0.1				**0.11**	
Games									0.1	
Books		0.05	0.06	**0.18**	0.04		0.03		0.08	
Hshld							0.05			
Clths	**-0.13**	**0.12**	**0.12**				0.07			
Hlth										
Chems										
Txtls							0.07		***0.19***	
Cnstr					0.06		-		**0.15**	
Steel							-		***0.15***	
FabPr					0.06				0.09	
ElcEq									***0.1***	
Autos			***0.12***				**0.18**		0.14	
Carry					***0.16***					
Mines										
Coal				0.19			0.1			
Oil										
Util	0.07		0.03			***-0.14***			**0.13**	**0.09**
Telcm		0.02		0.05			0.1	**-0.1**	***0.16***	
Servs									**0.16**	
BusEq										
Paper							**0.3**		**0.1**	
Trans									***0.12***	
Whlsl	-0.12	0.06	0.03						0.11	0.05
Rtail							***0.13***			
Meals		0.07	0.06					0.05	0.05	
Fin									***0.13***	
Other									**0.1**	
R^2	5.18	2.88	2.73	3.23	1.29	7.45	1.61	7.90	1.7	2.69

Table 2 (continued) This table reports the OLS estimated coefficients in terms of magnitude and statistical significance after we have implied LASSO to our full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables for the OLS regression by LASSO. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Another interesting finding is that industries located in the earlier stages of production process are negatively correlated. For instance, Coal (selected for 16 out of 30 industries) and Oil (selected fir 13 out of 30 industries) lagged returns have negative coefficients when are chosen for industries such as Txtl, Whlsl (Wholesale), Smoke and Books, which belong to the later stages. This negative in magnitude correlations arise from the fact that a supply shock will increase returns and profit margins for companies belonging in the earlier stages such as Coal and Oil following the supply and demand law. In contrast, this rise will affect badly industries in the latter stages in terms of profits and returns. Apart from these conclusions, there are relationships arising from Table 2 that we are not capable of evaluating. For example, we don't understand why lagged Beer is connected to future returns of Coal.

In Table 2 we also demonstrate the related R^2 for the industries which had at least one significant lagged predictor. The highest R^2 is around 8% for Txtls, whereas the lowest is 0.76% for Chemicals. The average R^2 is around 4%. As we will see in the next Section, although in sample R^2 is pretty low, meaning that our model cannot explain much of the variance of the data (just 4% on average), the performance of our long short portfolios is pretty good. Furthermore, R^2 shouldn't be our milestone, since bigger R^2 doesn't improve the predictive ability of our model. If we had used the whole predictor set for each industry, we would have got a much higher in sample R^2 than with the LASSO subset. But then the model would not be able to predict that well out of sample.

As we mentioned earlier in the introduction, LASSO tends to overshrink the coefficients. We found that if we run an OLS regression model with the full set of covariates, the magnitude of the estimated coefficients would be twice the size of the LASSO coefficients. This conclusion is presented in Tables A1 and A2 of the Appendix. In Table A1 of the Appendix we denote the estimated coefficients for the OLS model using the full set of predictors. In the same notion as in Table 2, the in-sample R^2 and the significant coefficients in the 10% (5%) level under the conventional t-statistic are shown. In Table A2 of the Appendix we examine the coefficients estimated by the pure LASSO model, where we can spot the aforementioned conclusion. When we execute the OLS post-LASSO model though, we found that the coefficients are only 15% smaller in magnitude than the full OLS model (after comparing Tables 2 and A1).

https://doi.org/10.26219/heal.aueb.6658

As the last part of our in-sample analysis we augmented in our data set four popular macroeconomic variables in order to check whether LASSO identifies the same lagged industry returns with and without the extra variables. We use the same macroeconomic variables as Avramov (2004), which are the S&P 500 dividend yield, the three-month Treasury bill yield, the difference between yields on a ten-year Treasury bond and three-month Treasury bill (term spread), and the difference between yields on BAA- and AAA-rated corporate bonds (credit spread). Using the OLS post-LASSO model we found that the estimated coefficients and the selected lagged industry returns are the same. Conclusively, the popular return predictors that we added, doesn't interfere with the predictive ability of the lagged industry excess returns. So, after taking into consideration all of the above in-sample analysis, we can argue that historically there were patterns in industry rotation that could have been exploited.

In order to introduce the reader more smoothly to the next Section, which refers to the out of sample analysis, we would like to describe a bit the elastic net procedure. LASSO in general deals with multicollinearity and chooses only one predictor from a potential group of correlated predictors. Elastic net differs by selecting correlated predictor variables as a group. In the following part of the project, we prove that the portfolios created under the OLS post-elastic net model derive the most profit among all other models tested, including OLS-post -LASSO. In Table A3 of the Appendix we present the estimated coefficients for OLS-post Elastic Net model for complicity reasons using ρ=0.5 (ρ=1 corresponds to LASSO). We observe that more coefficients are selected, whereas the in sample R^2 is not enhanced in relation to OLS post-LASSO model. In general, we get similar results both in predictors (elastic net picks the same predictors and some additional than the LASSO) and in magnitude of the predictors. So, LASSO seems to be sufficient in order to point out the relevant lagged predictors and thus is preferred for the in-sample analysis.

4.2. Out of sample analysis

In this section, we evaluate the out of sample predictive ability of our suggested models. The models tested is the vanilla OLS regression, the pure LASSO

https://doi.org/10.26219/heal.aueb.6658

model, the OLS post-LASSO model, the pure elastic net model and the OLS post-elastic net model.

In order to extract valuable out of sample conclusions we should never look our testing data set. To accomplish this, we should never decide according to the in-sample training metrics. We should use cross validation in order to tune the hyperparameters of the model and select the model with the best performance. Moreover, we never make a decision based on the test data. We leave test data set aside and use it only when the final-best performing model is selected. Following this procedure, we should be able to choose the best model in terms of bias-variance trade off. The bias-variance curve in Figure 3 illustrates that as we increase the complexity of our model (degree of non-linearity, number of hyperparameters etc.), the accuracy also increases. On the downside as we keep increasing complexity, variance also tends to increase. The number of parameters increases while there is a lack of data, resulting in overfitting and less robust estimated parameters. So, targeting the optimum model in terms of model complexity and minimization of the prediction error is crucial.

In order to attend the economic value of lagged industry returns predictability we construct a monthly equally weighted long-short industry rotation portfolio. We used data from 01/1960-12/1969 (for the first 120 months) for the training of our model. We used the resulting model, in order to forecast the industry returns for 01/1970. We sorted these predictions in ascending order and we picked the top 6 industries (20%). In practice we sorted the industries into 5 quantiles and just go long for the top quantile, whereas going short for the bottom quantile. After that, we train for 01/1960-01/1970 and use the selected model to predict the updated industry returns for 02/1970, and then choose the top quantile portfolio to go long. We keep on the same spirit, until we reach the end of our data. At the end of the day, we have constructed an equally weighted portfolio with the highest 20% industries at each time step for the out of sample period from 01/1970 to 12/2017. By this way we eliminate look-ahead bias, since the portfolio for each month of this period only uses data available at the time of forecast and as a result it represents a real time investment strategy.

 https://doi.org/10.26219/heal.aueb.6658

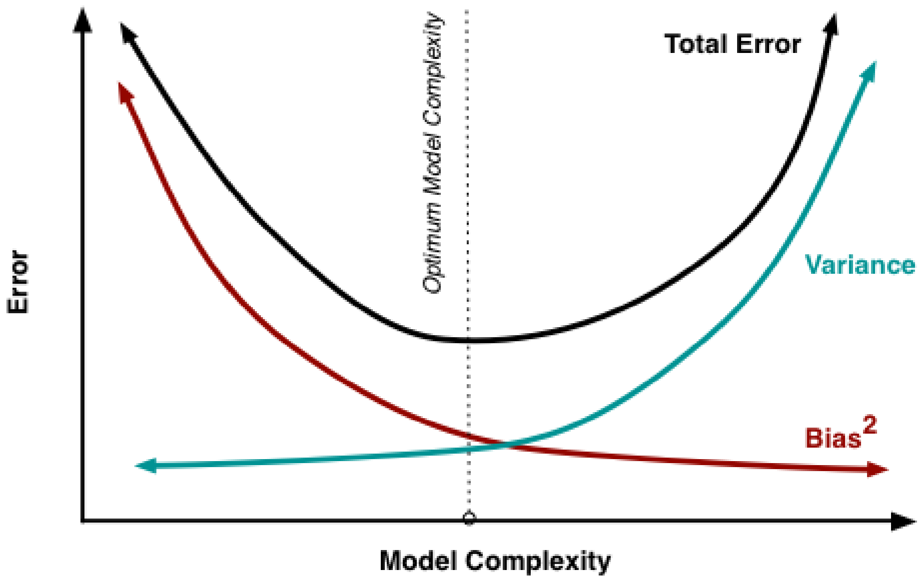


Figure 3: Bias-Variance curve.

In order to examine whether ML techniques help improve the predictability of the OLS model and so the economic value of the formatted portfolios, we set the OLS model as our benchmark for comparison reasons. We could also use a prevailing mean model as our benchmark (Rapach and Straus, 2018; Goyal and Welch, 2008), which is a really popular one in out of sample predictability literature. In the prevailing mean model, the monthly portfolios formed are based on the mean of all the monthly returns from the beginning of the sample till the date of the forecast. This model corresponds to the assumption of no industry excess return predictability. We found that the prevailing mean model performs really poorly (negative cumulative average returns), so we didn't consider it is as a trustworthy model to use as benchmark.

4.2.1. Backtesting and walk forward cross validation

In order this methodology to become clear to the reader, we would like to introduce the concept of walk forward cross validation and backtest simulation. First

https://doi.org/10.26219/heal.aueb.6658

of all, backtesting refers to applying a model to historical data to verify how the model would have worked during the specified time period and is useful if a system was not profitable in the past. In the field of time series forecasting, typical train-test splits and k-fold cross validation don't work because they don't take into consideration the temporal components and the serial correlation inherent in the problem. In order to respect the temporal order of our data, we will use walk forward cross validation to backtest our models in time series data. In walk forward cross validation, the model is updated each time new data are received, and so it can mimic a real time investing procedure. It can also deal with the problem of less and less accurate predictions as time passes, which is a typical concern in time series modelling. There are a few constraints we have to set before applying walk forward cross validation:

1. Set the minimum number of observations for model training. This could be thought as the width of the sliding window. In our application, the width is ten year (120 months).
2. Typically, we also have to choose either a sliding or an expanding window. In our application we use an expanding window. The method is also called as Rolling Window Analysis or Rolling Forecast due to the window used.
3. We have to choose in advance between the benefit of more robust estimators/improved performance and the computational cost of creating so many models (one model at each time step).

Generally, walk forward cross validation is the analogue of k-fold cross validation in time series. In a more technical level, we follow the next steps to apply this specialized form of CV:

1. Specify the folds in which we split our training data set (in this project the number of folds is set to five) without randomizing or re-ordering
2. Use fold 1 to train the model, and evaluate the chosen model using MSE in fold 2
3. Use folds 1 and 2 to train the model, and evaluate based on MSE in fold 3
4. Use folds 1, 2, and 3 to train the model and evaluate based on MSE in fold 4
5. Use folds 1, 2, 3, and 4 to train the model and evaluate based on MSE in fold 5

 https://doi.org/10.26219/heal.aueb.6658

6. Average the 4 scores. Since we have calculated the average score, we tune our model by iteratively evaluating different set of values for the hyperparameters. We select the best performing model based on MSE in cross validation.
7. Finally, the backtest simulation is performed with the most effective model in each time step.

4.2.2. Metrics of the generated portfolios

Table 3 summarizes the economic measures and some financial indicators for the constructed industry rotation portfolios based on the aforementioned models. As we suggested previously, we don't report metrics for the prevailing mean model since it underperforms. The vanilia OLS model, which uses the whole data set of candidate predictors for the forecasts, has an annualized mean return of 2.731%, annualized volatily of 5.791% and so annualized Sharpe Ratio of 0.501. It also shows a mediocre annualized maximum drawdown of 16.158%, whereas the MSE across predictions is 43.865. The OLS-post LASSO model, which is the model that Rapach, Strauss 2018 suggest, improves our benchmark with an annualized mean return of 3.603% and annualized SR=0.674. Furthermore, it reduces the value of annualized maximum drawdown to 12.748% and the MSE across prediction to 41.483. The pure LASSO model, which doesn't re-estimate the significant coefficients identified by LASSO, outperforms the benchmark but underperforms the OLS post-LASSO. The detailed results are presented in Table 3. A quick interpretation of these results is that even without imposing ML techniques, OLS model outperforms the prevailing mean model and so the lagged industry returns produce more accurate forecasts than the prevailing mean. Moreover, applying the OLS-post LASSO outperforms both the pure LASSO and the benchmark OLS model, suggesting that the OLS model suffers from overfitting. In addition, the predictors selected by the LASSO and the re-estimation of their coefficients using OLS, produces the most accurate predictions and consequently the most economically valuable industry rotation portfolio.

As the last additive linear model, we used the Elastic Net regularization which includes a combination of L1 and L2 regularization. We checked both a pure Elastic Net and an OLS-post Elastic Net model. We perform cross validation to tune the hyperparameters λ and ρ of equation 2.7. We tested various set of values for the

 https://doi.org/10.26219/heal.aueb.6658

hyperparameters and we derived the conclusion that for λ=10 and ρ=0.1, we obtained better results than the OLS post-LASSO model. In Table A4 of the Appendix, we present a whole table of the scores for the different set of λ and ρ. We predefined λ=1, 3, 10, 30, 100 and ρ= 0.01, 0.05, 0.1, 0.5, 0.667, 0.75, 0.9, 0.95, 0.99, 1. Through cross validation for all the possible combinations we found the optimal regularization parameters for the elastic net model. So, we found that the annualized mean return for the portfolio formed based on the elastic net model forecasts is 4.354%, the annualized volatility is 6.491% (a bit higher than the volatility of the OLS post-LASSO) and the Sharpe Ratio is improved to 0.707%. The annualized maximum drawdown is pretty much alike at 12.624% and the MSE across prediction is 39.727. In Table 3, we present also the results for the OLS post-Elastic Net model and we observe that the forecasts of the pure Elastic Net outperform them.

Table 3: Performance measures for long industry rotation portfolios.

Model	Annualized mean return (%)	Annualized Volatility (%)	Annualized Sharpe Ratio (%)	Annualized Maximum Drawdown (%)	MSE across predictions
OLS	2.731	5.791	0.501	16.158	43.865
OLS post-LASSO	3.603	5.579	0.674	12.748	41.483
LASSO	2.888	5.849	0.534	21.969	40.005
Elastic Net	**4.534**	**6.409**	**0.707**	**12.624**	**39.727**
OLS post-Elastic Net	3.959	6.284	0.629	13.088	39.684

The table shows some financial metrics for top 20% decile long industry portfolio constructed under the out of sample predictions of the OLS model, the OLS pos-LASSO model, the pure LASSO model without re-estimation of the selected slope coefficients, the pure elastic net model and the OLS post-elastic net model. The benchmark model is the OLS model, and we show in bold the metrics for the portfolio selected by the best performing model, which is the Elastic Net model. Each long portfolio is a zero-investment portfolio that picks the top 20% industries in terms of forecasted excess returns. All the portfolios are equally weighted and the backtesting is done simultaneously with walk forward cross validation.

As a part of further research, in order to exploit nonlinear relationships, we added squared returns of individual industries and some interaction terms between the industries in our Elastic Net model. The results were not better, so we reported only the results from our initial data set.

https://doi.org/10.26219/heal.aueb.6658

The results of Table 3 are visually summarized in Figure 4, which reports the cumulative return of portfolios sorted on out-of-sample machine learning industry excess returns forecasts. The lines represent the long (top decile) positions and all the portfolios are equally weighted. The industry rotation portfolio based on elastic net gains consistently and reaches a terminal value of around 750% at the end of our sample size (12/2017).

Moreover, from Figure 4 we can detect the performance of our portfolios during the NBER (National Bureau of Economic Research) recession dates. For example, during the Great Recession between 2008 and 2010 the annualized averaged portfolio returns of the elastic net model continue to rise even with a greater rate than during expansions of the economy. For this reason, we can argue that the elastic net model can provide us with a portfolio which can be used as a hedge for the bad times of the macroeconomy.

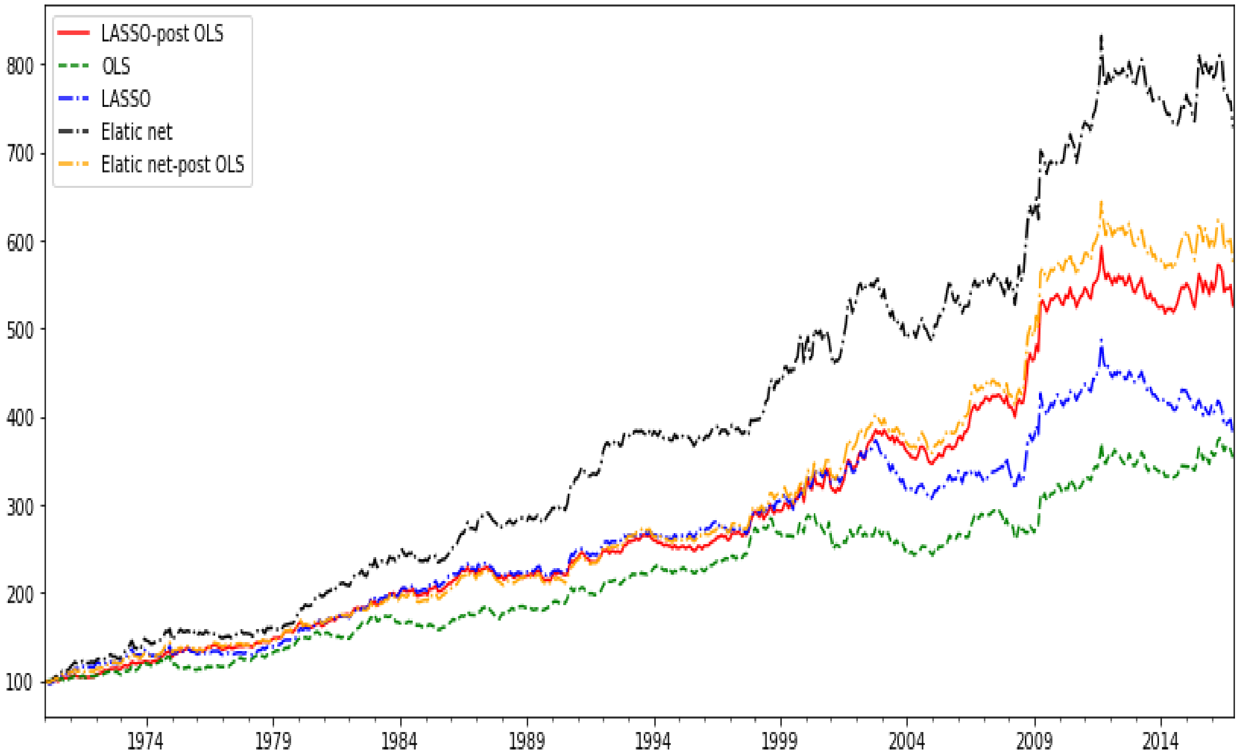
Last but not least, the out of sample analysis for the selected NN is depicted in Figure 5. In our project, the weights are penalized using the penalized l1 and l2 objective function of prediction errors (similar to elastic net). In Table 4 we perform the results for the cross-validation procedure regarding the number of hidden layers, the number of hidden units in each layer and the different values for the regularization penalties. We detected that for higher regularization penalty values the MSE's group around 39, meaning we have high-bias models. We also spotted, that in many occasions we get a lower MSE than in the linear methods, but the respective networks resulted in also lower Sharpe Ratios. The higher Sharpe Ratios are around 0.55 for some of the NN1 and NN2 networks. These networks are the closest counterparts of linear regression. As a result, we could conclude that the relationships underlying the predictors and the dependent variables are linear. We could choose the NN1 network with regularization penalty=0.001 or the NN2 network with regularization penalty=0.001. Since we don't improve in comparison with at least LASSO, NN should not be a preferred strategy. The aforementioned conclusion can be shown graphically in Figure 5. NN2 doesn't improve in terms of cumulative excess return. Elastic Net still provides the higher returns.

It is obvious that neural networks need many parameters in order to be completely specified. It is easy to understand why we preset a specific range of

 https://doi.org/10.26219/heal.aueb.6658

architectures we examine. In our opinion, these architectures are representative for the predictive performance under any architecture.

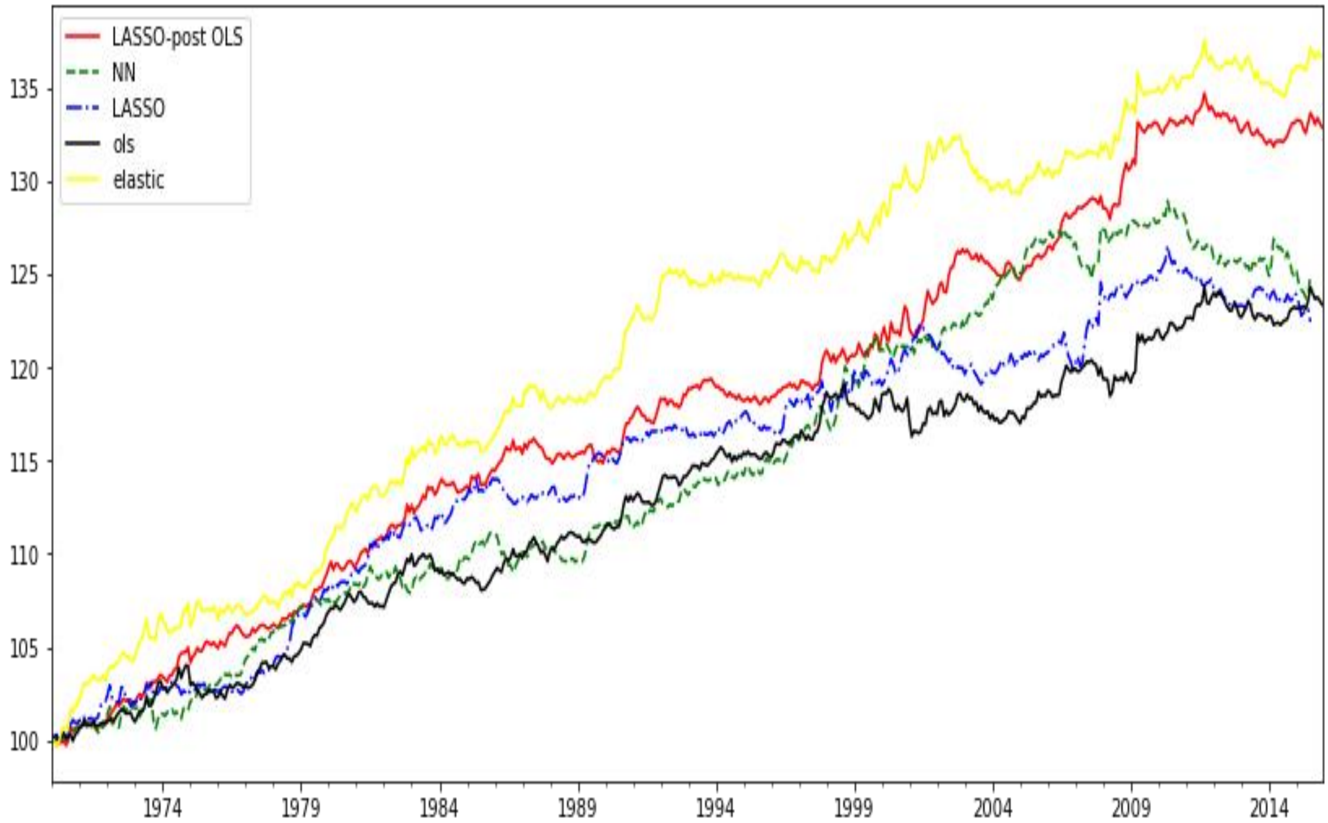
Figure 4: Cumulative excess returns for the long industry rotation portfolios based on each one of the proposed models.



This Figure presents the cumulative excess returns for long industry portfolio that goes long the six industries with the highest out-of-sample forecasted excess returns. The green line (OLS) corresponds to our benchmark model, while the black line (elastic net model) correspond to the long portfolio constructed under the best performing model. During the great Recession of 2008-2010, the portfolio based on elastic net keeps gaining profits.

 https://doi.org/10.26219/heal.aueb.6658

Figure 5: Cumulative excess returns for the long industry rotation portfolios based on each one of the proposed models including the NN2.



This Figure presents the cumulative excess returns for long industry portfolio that goes long the six industries with the highest out-of-sample forecasted excess returns. All the predictors used in order to forecast the returns under each one of the models are standardized. The NN2 model doesn't improve the elastic net model. The results of the NN2 are pretty similar with these of the linear regression and the pure LASSO models.

https://doi.org/10.26219/heal.aueb.6658

Table 4: Metrics for the neural networks under different architectures.

Hidden Layers	Hidden units	Regularization Penalty	Sharpe Ratio	Ann. return(%)	MSE
1	**32**	**0.00001**	**0.533**	**0.459**	**60.78**
1	**32**	**0.001**	**0.558**	**0.494**	**65.23**
1	32	0.1	0.509	0.412	70.94
1	**32**	**1**	**0.533**	**0.446**	**51.5**
1	32	3	0.311	0.276	44.73
1	32	10	0.259	0.234	39.98
2	32,16	0.00001	0.420	0.387	63.28
2	**32,16**	**0.001**	**0.513**	**0.438**	**55.24**
2	32,16	0.1	0.139	0.112	62.25
2	32,16	1	0.305	0.263	68.9
2	32,16	3	-0.296	-0.217	39.54
2	32,16	10	-0.245	-0.18	39.54
3	32, 16, 8	0.00001	0.293	0.245	55.98
3	32, 16, 8	0.001	0.062	0.802	50.5
3	32, 16, 8	0.1	0.297	0.829	60.7
3	32, 16, 8	1	0.160	0.127	41.36
3	32, 16, 8	3	-0.303	-0.209	39.55
3	32, 16, 8	10	-0.182	-0.131	39.55

5. Conclusion

In this project we used context from the return predictability literature combined with the machine learning framework. The goal is to prove the existence of links between the industry returns and their lagged values. We worked on a widely used and public dataset of excess industry returns provided by Kenneth French's Data Library.

 https://doi.org/10.26219/heal.aueb.6658

Firstly, in Section 2 we present a general predictive regression model, which regresses simultaneously each one of the thirty industry returns with the thirty lagged returns counterparts. The results are pretty interesting, since we have proven that the long industry rotation portfolio based on the forecasts of the multidimensional linear regression outperforms in terms of annualized returns and Sharpe Ratio the typical and widely used prevailing mean portfolio.

In terms of in sample analysis we show in Section 4.1 that a simple linear regression can provide us with better in sample R^2 since it uses more predictors. On the other hand, it lucks predictive effectiveness since we overfit the data. We tried to expand these results by using the LASSO and elastic net penalization methods. LASSO and elastic net based-estimators correspond to lower in sample R^2, but to a better out of sample performance. In order to further enhance our results, we used OLS to estimate the coefficients derived from the LASSO and elastic net. By this way, we succeed in defending against the downside biases in magnitude existing in the pure LASSO coefficient estimators. Moreover, we faced the problem of multiple testing for the in-sample analysis. We picked the most significant predictors using the Bonferroni and Hochberg (2020) methodology.

For the out of sample analysis, in Section 4.2 we created a zero-investment long industry rotation portfolio, which invests in the top 20% industries depending on their excess returns. We did that by employing the walk-forward validation approach, which is the counterpart of conventional cross-validation for time series data. So, we have chosen the optimal hyperparameters by walk forward validation, and then performed backtesting through sklearn python packages. The portfolios constructed based on the forecasts of the two machine learning methods provide better results than the typical linear regression using OLS as the objective function. The reason behind this improvement is the dimensionality reduction that these methods are created to achieve. The elastic net-based portfolio delivers the highest annualized excess return of 4,354%, the highest Sharpe Ratio of 0,707, the lowest MSE across predictions and the lowest annualized maximum drawdown of 12,624%.

The last part of the project includes the most powerful machine learning method in the recent literature, the artificial neural networks. We so created a long industry rotation portfolio based on the forecasts of many differently structured neural

https://doi.org/10.26219/heal.aueb.6658

networks. We observe that the results for the best-performing network, which contained two hidden layers with 32 and 16 neurons respectively, was not better than these derived by the elastic net. Although we gained better MSE results and so seemingly better predictions, we weren't able to generate high enough Sharpe Ratios. Since neural networks are a lot more sophisticated, computationally expensive and complicated regarding their parametrization we determined that they shouldn't be the selected technique.

Complex machine learning methods require high frequency data apart from the correct tuning and general structuring. The bad results maybe link to the fact that non-linear relationships between lagged returns and the dependent variable, or non-linear interactions among the predictors are non-existent. It is also quite possible that we either didn't manage to tune our network optimally, or our training data set is not representative of the out of sample data or it is overly correlated with the validation set. An ensemble methodology of many networks with identical architecture could have proven valuable

There are huge margins for improvement and further research. A good idea would be to use cross entropy as our objective function, which is more directly connected with performance. Another alternative would be to maximize directly excess returns or Sharpe Ratio, and not a proxy such as MSE. Moreover, picking the top 20% industries is not convex everywhere. So, another strategy for portfolio construction maybe could generate even better results.

References

Avramov, D. (2004). Stock Return Predictability and Asset Pricing Models. Review of Financial Studies 17:3, 699–738.

Avramov, D., Barras, L., and Kosowski, R. (2013). Hedge fund return predictability under the magnifying glass. Journal of Financial and Quantitative Analysis, 48(4):1057–1083.

 https://doi.org/10.26219/heal.aueb.6658

Avramov, D., Kosowski, R., Naik, N. Y., and Teo, M. (2011). Hedge funds, managerial skill, and macroeconomic variables. Journal of Financial Economics, 99(3):672–692.

Bai, J. and Ng, S. (2008). Forecasting economic time series using targeted predictors. Journal of Econometrics, 146(2):304–317

Benjamini, Y. and Y. Hochberg (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society. Series B (Methodological) 57:1, 289–300.

Benjamini, Y. and Y. Hochberg (2000). On the Adaptive Control of the False Discovery Rate in Multiple Testing with Independent Statistics. Journal of Educational and Behavioral Statistics 25:1, 60–83.

Bollerslev, T., Hood, B., Huss, J., & Pedersen, L. H. (2018). Risk Everywhere: Modeling and Managing Volatility. Review of Financial Studies, 31(7), 2729-2773.

Cao, C., Chen, Y., Liang, B., and Lo, A. W. (2013). Can hedge funds time market liquidity? Journal of Financial Economics, 109(2):493–516.

Campbell, J. Y. and S. B. Thompson (2008). Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average? Review of Financial Studies 21:4, 1509–1531.

Carhart, M. M. (1997). On Persistence in Mutual Fund Performance. Journal of Finance 52:1, 57–82.

Chen, Y. and Liang, B. (2007). Do market timing hedge funds time the market? Journal of Financial and Quantitative Analysis, 42:827–856.

Diebold, F., & Mariano, R. (1995). Comparing Predictive Accuracy. Journal of Business & Economic Statistics, 13(3), 253-263.

Drobetz, W., Haller, R., Jasperneite, C., & Otto, T. (2019). Predictability and the Cross Section of Expected Returns: Evidence from the European Stock Market. Journal of Asset Management, 20(7), 508- 533.

Drobetz, Wolfgang and Otto, Tizian, Empirical Asset Pricing via Machine Learning: Evidence from the European Stock Market (July 1, 2020).

https://doi.org/10.26219/heal.aueb.6658

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least Angle Regression. Annals of Statistics 32:2, 407–451.

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. Journal of Finance, 47(2):427–465.

Fama, E. F., & MacBeth, J. D. (1973). Risk, Return, and Equilibrium: Empirical Tests. Journal of Political Economy, 81(3), 607-636.

Fama, E. F. and K. R. French (1993). Common Risk Factors in the Returns on Stocks and Bonds. Journal of Financial Economics 33:1, 3–56.

Fama, E. F. and K. R. French (2015). A Five-Factor Asset Pricing Model. Journal of Financial Economics 116:1, 1–22.

Fama, E. F., & French, K. (2019). Comparing Cross-Section and Time-Series Factor Models. Chicago Booth Research Paper No. 18-08, 1-43.

Feng, G., S. Giglio, and D. Xiu (2017). Taming the Factor Zoo. Unpublished Working Paper, City University of Hong Kong, Yale University, University of Chicago.

Ferson, W. E. and C. R. Harvey (1991). The Variation of Economic Risk Premiums. Journal of Political Economy 99:2, 385–415. Ferson, W. E. and C. R. Harvey (1999). Conditioning Variables and the Cross Section of Stock Returns. Journal of Finance 54:4, 1325–1360.

Ferson, W. E. and R. A. Korajczyk (1995). Do Arbitrage Pricing Models Explain the Predictability of Stock Returns? Journal of Business 68:3, 309–349.

Freyberger, J., Neuhierl, A., & Weber, M. (2020). Dissecting Characteristics Nonparametrically. The Review of Financial Studies, 33(5), 2326-2377.

Friedman, J. H. (2002). Stochastic gradient boosting. Comp. Stat. & Data Analy., 38(4):367– 378

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software 33:1, 1–22.

Galakis, John and Vrontos, Ioannis D. and Xidonas, Panos, On Tree-Structured Linear and Quantile Regression Based Asset Pricing (July 1, 2019).

 https://doi.org/10.26219/heal.aueb.6658

Giomouridis, D. and Vrontos, I. (2007). Hedge fund portfolio construction: A comparison of static and dynamic approaches. Journal of Banking and Finance, 31(1):199–217.

Goyal, A. and I. Welch (2008). A Comprehensive Look at the Empirical Performance of Equity Premium Prediction. Review of Financial Studies 21:4, 1455–1508.

Green, J., J. R. M. Hand, and X. F. Zhang (2017). The Characteristics that Provide Independent Information about Average U.S. Monthly Stock Returns. Review of Financial Studies 30:12, 4389–4436.

Gu, S., Kelly, B., & Xiu, D. (2020a). Empirical Asset Pricing via Machine Learning. The Review of Financial Studies, 33(5), 2223-2273.

Gu, S., Kelly, B., & Xiu, D. (2020b). Autoencoder Asset Pricing Models. Journal of Econometrics, Forthcoming.

Han, Y., A. He, D. E. Rapach, and G. Zhou (2018). What Firm Characteristics Drive US Stock Returns? Unpublished Working Paper, University of North Carolina at Charlotte, Emory University, Saint Louis University, Washington University in St. Louis.

Harvey, C. R., Y. Liu, and H. Zhu (2016). ... and the Cross-Section of Expected Returns. Review of Financial Studies 29:1, 5–68.

Heaton, J., Polson, N., & Witte, J. H. (2016). Deep Learning for Finance. Working Paper.

Hochberg, Y. and Y. Benjamini (1990). More Powerful Procedures for Multiple Significance Testing. Statistics in Medicine 9:7, 811–818.

Hong, H., W. Torous, and R. Valkanov (2007). Do Industries Lead Stock Markets? Journal of Financial Economics 83:2, 367–396.

Huerta, Ramon and Elkan, Charles and Corbacho, Fernando, Nonlinear Support Vector Machines Can Systematically Identify Stocks with High and Low Future Returns (September 6, 2012). Algorithmic Finance (2013), 2:1, 45-58

Hurvich, C. M. and C.-L. Tsai (1989). Regression and Time Series Model Selection in Small Samples. Biometrika 76:2, 297–307.

 https://doi.org/10.26219/heal.aueb.6658

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning, 448-456.

Jiaqi Chen & Michael Tindall & Wenbo Wu, 2016. "Hedge Fund Return Prediction and Fund Selection: A Machine-Learning Approach," Occasional Papers 16-4, Federal Reserve Bank of Dallas.

Kelly, B. T., Pruitt, S., & Su, Y. (2019). Characteristics are Covariances: A Unified Model of Risk and Return. Journal of Financial Economics.

Kingma, D., and J. Ba. 2014. Adam: A method for stochastic optimization. Preprint, https:// arxiv.org/abs/1412.6980.

Lewellen, J. (2015). The Cross-section of Expected Stock Returns. Critical Finance Review, 4(1), 1-44.

Linnainmaa, J. T. and M. R. Roberts (2018). The History of the Cross Section of Stock Returns. Review of Financial Studies 31:7, 2606–2649.

Meinshausen, N. (2007). Relaxed Lasso. Computational Statistics and Data Analysis 52:1, 374–393.

Meligkotsidou, L., Panopoulou, E., Vrontos, I.D. and Vrontos, S. 2014. A quantile regression approach to equity premium prediction. Journal of Forecasting. 33 (7), pp. 558-576.

Merton, R. C. (1987). A Simple Model of Capital Market Equilibrium with Incomplete Information. Journal of Finance 42:3, 483–510.

Moritz, B., and T. Zimmermann. 2016. Tree-based conditional portfolio sorts: The relation between past and future stock returns. Working Paper, Ludwig Maximilian University of Munich.

Moskowitz, T. J. and M. Grinblatt (1999). Do Industries Explain Momentum? Journal of Finance 54:4, 1249–1290.

Rapach, D. E. and G. Zhou (2013). Forecasting Stock Returns. In Handbook of Economic Forecasting. Ed. by G. Elliott and A. Timmermann. Vol. 2A. Amsterdam: Elsevier, pp. 328–383.

 https://doi.org/10.26219/heal.aueb.6658

Rapach, D., Strauss, J., & Zhou, G. (2013). International Stock Return Predictability: What Is the Role of the United States? The Journal of Finance, 68(4), 1633-1662.

Rapach D., Jack K. Strauss, Jun Tu and Guofu Zhou (2019). Industry Return Predictability: A Machine Learning *Approach*.The Journal of Financial Data Science Summer 2019, 1 (3) 9-28;

Sortino, F. A. and Price, L. N. (1994). Performance measurement in a downside risk framework. The Journal of Investing, 3(3):59–64.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. Journal of the Royal Statistical Society. Series B (Methodological) 58:1, 267–288.

Vrontos, S., Vrontos, I.D. and Giamouridis, D. 2008. Hedge fund pricing and model uncertainty. Journal of Banking and Finance. 32 (5), pp. 741-753.

Welsch, R. E. and Zhou, X. (2007). Application of robust statistics to asset allocation models. Statistical Journal, 5(1):97–114.

Vrontos, I.D., Meligkotsidou, L. and Vrontos, S. 2011. Performance evaluation of mutual fund investments: the impact of non-normality and time-varying volatility. Journal of Asset Management. 12 (4), pp. 292-307.

Zou, H. and T. Hastie (2005). Regularization and Variable Selection via the Elastic Net. Journal of the Royal Statistical Society. Series B (Statistical Methodology) 67:2, 301–320.

Website and Books

https://alphaarchitect.com/2017/09/27/machine-learning-investors-primer/

https://scikit-learn.org/stable/index.html

https://keras.io/api/layers/

https://machinelearningmastery.com/

https://towardsdatascience.com/machine-learning/home

https://doi.org/10.26219/heal.aueb.6658

https://faculty.washington.edu/ezivot/econ424/returnCalculations.pdf

"Introduction to Portfolio Construction and Analysis with Python" by EDHEC Business School 2020, Coursera

https://en.wikipedia.org/wiki/Walk_forward_optimization

http://amunategui.github.io/wallstreet/

https://christophj.github.io/replicating/r/replicating-goyal-welch-2008/

http://didattica.unibocconi.it/mypage/dwload.php?nomefile=fama-macbeth20141115121157.pdf

Emmanuel Jurczenko, "Machine Learning for Asset Management, New Developments and Financial Applications", Wiley Editions

https://doi.org/10.26219/heal.aueb.6658

Appendix

Regressors

Dependent	*Food*	*Beer*	*Smok*	*Game*	*Book*	*Hshld*	*Clths*	*Hlth*	*Chem*	*Txtls*
Food	0.004	-0.09	0.02	0.02	0.04	0.00	0.02	0.06	-0.07	0.01
Beer	**0.2**	**-0.11**	-0.00	-0.07	0.08	0.05	0.09	-0.02	0.05	-0.06
Smoke	-0.00	-0.01	001	-0.05	0.03	-0.07	0.06	0.04	0.07	0.05
Games	-0.1	0.01	-0.03	-0.04	**0.26**	-0.01	0.12	-0.03	0.03	0.01
Books	-0.1	-0.02	0.04	0.04	0.1	-0.02	0.01	-0.04	0.08	-0.01
Hshld	0.00	0.01	-0.01	-0.02	0.09	-0.05	**0.11**	0.01	0.01	-0.01
Clths	-0.07	-.05	-0.01	-0.09	0.08	**-0.14**	0.07	0.08	0.17	0.01
Hlth	-0.04	0.02	0.00	-0.01	**0.18**	-0.03	0.07	-0.08	0.08	-0.06
Chems	-0.04	0.04	-0.01	-0.12	0.11	-0.12	0.09	-0.01	0.1	-0.03
Txtls	-0.16	0.04	0.02	-0.05	-0.00	-0.16	0.12	0.02	0.15	-0.07
Cnstr	-0.00	-0.05	-0.02	**-0.14**	-0.08	-0.11	0.05	-0.01	0.15	-0.02
Steel	-0.01	0.01	-0.06	-0.11	0.04	-0.17	-0.04	-0.06	0.09	-0.04
FabPr	-0.03	-0.01	-0.05	**-0.13**	0.13	-0.13	0.01	-0.06	0.14	-0.06
ElcEq	0.03	0.01	-0.05	-0.07	0.08	-0.14	-0.00	-0.01	0.14	-0.06
Autos	-0.06	0.04	-0.02	**-0.15**	0.13	-0.28	0.03	-0.08	0.16	0.01
Carry	-0.02	-0.03	0.00	-0.06	0.1	-0.08	0.02	-0.1	0.00	-0.01
Mines	-0.13	-0.03	0.07	**-0.14**	0.07	-0.12	-0.04	-0.04	-0.04	-0.08
Coal	0.09	**-0.25**	-0.1	0.00	0.19	-0.14	-0.01	-0.04	0.07	0.00
Oil	0.00	-0.06	-0.02	-0.07	0.04	-0.14	-0.03	-0.1	0.05	0.01
Util	0.1	**-0.1**	0.02	-0.04	0.01	-0.09	-0.01	-0.08	-0.08	-0.02
Telcm	0.06	-0.06	-0.03	0.01	0.1	-0.07	-0.04	-0.04	0.02	-0.00
Servs	-0.07	0.05	-0.09	-0.07	0.15	0.01	0.01	-0.01	0.09	-0.02
BusEq	-0.03	0.07	**-0.15**	-0.11	**0.24**	-0.04	-0.01	0.04	0.13	-0.04
Paper	0.04	-0.01	0.00	**-0.09**	0.11	0.01	0.07	-0.03	0.07	0.02
Trans	0.08	-0.08	-0.03	-0.1	**0.15**	-0.08	0.02	-0.1	0.05	-0.01
Whlsl	-0.1	-0.04	-0.04	-0.08	**0.18**	-0.02	-0.02	-0.06	0.11	-0.00
Rtail	0.00	-0.02	-0.00	-0.06	0.06	-0.06	0.01	0.05	0.14	-0.05
Meals	0.05	-0.04	-0.04	-0.02	0.12	-0.04	0.1	-0.09	0.08	-0.05
Fin	-0.00	-0.05	-0.02	-0.07	**0.13**	-0.08	0.09	-0.03	0.01	-0.01
Other	-0.04	-0.00	-0.06	-0.05	0.07	-0.06	0.11	-0.02	0.08	-0.01
R^2(%)	5.12	5.01	7.53	7.57	8.30	4.97	9.25	4.56	5.64	11.64

Table A1: This table reports the OLS estimated coefficients in terms of magnitude and statistical significance using the full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Regressors

Dependent	***Cnstr***	***Steel***	***FabP***	***ElcEq***	***Autos***	***Carry***	***Mines***	***Coal***	***Oil***	***Util***
Food	-0.09	0.00	**0.16**	-0.09	0.01	0.04	-0.02	**-0.04**	-0.08	**0.15**
Beer	-0.09	0.00	0.04	-0.08	0.02	0.06	-0.01	**-0.05**	-0.07	0.04
Smoke	0.03	0.02	0.01	-0.08	-0.00	**0.16**	-0.04	-0.04	**-0.13**	**0.28**
Games	-0.02	-0.13	**0.25**	-0.06	0.01	-0.03	0.03	**-0.06**	**-0.16**	0.15
Books	-0.12	**-0.12**	0.11	-0.12	0.01	0.03	0.05	-0.03	**-0.19**	**0.18**
Hshld	-0.08	-0.05	0.06	-0.04	0.01	0.02	-0.01	-0.03	-0.03	0.07
Clths	0.09	-0.14	0.08	**-0.27**	0.01	0.06	0.02	**-0.06**	**-0.18**	**0.21**
Hlth	-0.07	-0.05	0.03	0.01	-0.04	0.02	-0.04	-0.04	-0.06	**0.17**
Chems	0.02	-0.08	**0.17**	-0.11	0.04	0.08	0.04	-0.04	**-0.17**	**0.15**
Txtls	0.01	-0.08	**0.24**	**-0.24**	0.11	-0.05	0.05	**-0.09**	**-0.26**	**0.19**
Cnstr	-0.06	**-0.13**	0.13	-0.12	0.01	0.08	0.05	**-0.06**	**-0.2**	**0.21**
Steel	**-0.14**	0.08	-0.04	0.02	0.02	0.06	-0.01	-0.15	**-0.15**	**0.19**
FabPr	-0.03	-0.12	0.14	-0.07	-0.02	0.02	0.05	-0.02	**-0.17**	**0.16**
ElcEq	-0.05	-0.06	**0.23**	**-0.15**	0.02	0.01	0.02	-0.04	**-0.23**	**0.24**
Autos	-0.01	-0.1	0.18	-0.14	-0.06	-0.01	0.04	-0.04	**-0.2**	**0.21**
Carry	-0.07	-0.08	**0.21**	-0.1	0.03	0.05	-0.00	**-0.06**	**-0.13**	**0.18**
Mines	0.11	-0.12	0.21	-0.06	0.1	**0.14**	-0.01	-0.06	-0.04	**0.27**
Coal	-0.15	-0.01	-0.05	0.07	0.04	0.1	0.04	**0.1**	**-0.25**	0.11
Oil	-0.03	-0.04	0.07	0.1	0.01	**0.15**	0.02	-0.03	-0.08	0.05
Util	**-0.19**	0.00	**0.15**	-0.01	0.01	**0.07**	-0,04	0,00	**-0.08**	0.09
Telcm	-0.09	-0.01	0.12	-0.04	-0.05	-.03	-0.01	-0.03	**-0.11**	**0.18**
Servs	-0.12	**-0.12**	0.17	-0.05	-0.05	0.01	0.01	0.00	**-0.17**	**0.16**
BusEq	-0.16	-0.09	0.09	-0.05	-0.05	-0.04	0.02	0.00	**-0.14**	**0.23**
Paper	-0.03	-0.05	0.12	**-0.18**	-0.03	0.08	0.01	-0.04	**-0.14**	0.08
Trans	-0.07	**-0.13**	**0.25**	**-0.15**	-0.03	0.03	0.04	-0.02	**-0.19**	**0.2**
Whlsl	-0.00	-0.02	0.05	-0.09	-0.01	0.08	0.01	-0.04	**-0.19**	**0.25**
Rtail	0.03	**-0.15**	0.12	**-0.16**	0.05	-0.07	-0.01	-0.02	**-0.14**	0.11
Meals	0.03	**-0.15**	0.07	-0.04	-0.02	0.06	0.02	**-0.07**	**-0.19**	0.14
Fin	-0.06	-0.09	0.16	-0.04	-0.02	0,02	-0.01	-0.03	**-0.14**	**0.18**
Other	0.01	-0.07	0.12	-0.09	-0.03	0.05	-0.00	-0.03	**-0.15**	**0.17**
R^2(%)	8.46	4.11	5.41	5.67	8.71	6.23	5.63	4.28	5.16	8.38

Table A1 (continued): This table reports the OLS estimated coefficients in terms of magnitude and statistical significance using the full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

Regressors

Dependent	***Tlcm***	***Servs***	***BusE***	***Paper***	***Trans***	***Whlsl***	***Rtail***	***Meals***	***Fin***	***Other***
Food	-0.08	-0.02	-0.01	-0.04	0.03	-0.03	0.06	0.02	0.05	0.02
Beer	0.00	0.03	0.03	-0.09	0.03	-0.01	-0.09	0.07	0.08	0.01
Smoke	-0.11	**-0.2**	0.11	**-0.22**	0.1	0.01	-0.11	0.02	0.12	0.03
Games	-0.01	0.13	0.01	-0.07	0.07	-0.12	-0.15	-0.01	**0.2**	-0.04
Books	-0.03	0.08	0.1	-0.04	0.03	-0.08	0.07	0.02	**0.15**	0.00
Hshld	-0.03	0.04	-0.03	-0.06	0.05	-0.11	0.04	0.02	0.09	0.01
Clths	**-0.14**	0.14	**0.13**	-0.02	0.02	-0.07	0.09	0.05	0.09	0.01
Hlth	-0.04	0.06	-0.00	-0.06	0.04	-0.01	-0.02	0.00	0.01	0.01
Chems	-0.03	0.04	0.05	-0.12	0.04	**-0.18**	0.05	0.04	0.05	-0.01
Txtls	-0.01	0.05	0.06	-0.01	0.07	-0.17	0.16	0.02	**0.24**	0.04
Cnstr	0.00	0.09	0.07	-0.05	0.09	-0.11	0.03	0.03	**0.22**	-0.01
Steel	0.01	**0.16**	**0.17**	0.06	-0.08	-0.08	0.01	0.03	**0.3**	0.5
FabPr	-0.06	0.04	0.12	-0.03	0.1	0.01	0.03	-0.01	**0.19**	-0.02
ElcEq	-0.11	0.07	0.1	-0.12	-0.01	-0.06	0.04	0.08	**0.21**	-0.04
Autos	-0.04	0.07	**0.17**	-0.02	-0.01	-0.14	0.16	0.11	**0.18**	0.02
Carry	-0.01	0.08	0.08	-0.06	**0.15**	-0.00	-0.07	0.00	**0.12**	-0.01
Mines	**-0.19**	0.1	0.1	0.09	-0.01	**-0.2**	0.07	**0.16**	-0.05	-0.04
Coal	0.00	0.2	-0.15	**0.34**	-0.16	-0.18	0.17	-0.02	-0.04	0.01
Oil	-0.07	-0.04	0.02	-0.06	0.03	-0.09	0.05	-0.02	0.12	0.08
Util	0.06	0.01	0.03	0.06	0.01	**-0.15**	0.03	0.03	0.12	**0.1**
Telcm	-0.06	0,04	0.04	-0.03	-0.01	-0.02	0.12	**-0.11**	**018**	-0.01
Servs	-0.07	0.02	0.06	-0.09	-0.04	0.04	0.03	0.01	**0.21**	0.03
BusEq	-0.04	0.03	0.08	-0.05	-0.06	0.15	0.06	-0.07	**0.18**	-0.00
Paper	-0.04	0.08	0.07	-0.14	0.08	**-0.14**	0.03	0.01	**0.16**	-0.06
Trans	-0.08	0.1	0.04	-0.06	0.07	**-0.16**	0.04	0.03	**0.18**	0.1
Whlsl	**-0.12**	0.07	0.07	-0.08	-0.00	-0.03	0.01	0.05	0.13	0.07
Rtail	-0.07	0.07	0.1	-0.08	-0.04	-0.04	0.08	0.07	0.09	0.09
Meals	-0.02	0.07	0.1	-0.12	0.02	-0.01	-0.04	0.09	0.1	0.01
Fin	-0.05	0.07	0.06	0.01	0.01	**-0.17**	-0.04	-0.04	**0.17**	0.12
Other	-0.12	0.13	0.03	0.01	0.06	-0.08	-0.06	-0.05	**0.2**	-0.02
R^2(%)	5.82	4.26	6.07	7.02	7.48	8.27	6.97	8.87	6.61	6.26

Table A1 (continued): This table reports the OLS estimated coefficients in terms of magnitude and statistical significance using the full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

Regressors

Dependent	***Food***	***Beer***	***Smok***	***Game***	***Book***	***Hshld***	***Clths***	***Hlth***	***Chemm***	***Txtls***
Food							0.01			
Beer	0.09						0.03			
Smoke										0.03
Games					**0.15**		0.03			
Books				0.02	0.02					
Hshld							0.06			
Clths					0.03		0.08		0.03	
Hlth					0.05					
Chems							0.02			
Txtls							0.07			
Cnstr							0.02			
Steel										
FabPr										
ElcEq										
Autos						-0.11	0.01			
Carry										
Mines										
Coal		**-0.18**	-0.05		0.11					
Oil		-0.01						-0.04		
Util	0.03	**-0.08**				-0.03		-0.05		
Telcm		-0.05	-0.02		0.05	-0.04				
Servs			-0.03		0.05					
BusEq			-0.05		0.07					
Paper							0.04			
Trans										
Whlsl	-0.06	-0.02	-0.04		**0.12**			-0.02		
Rtail										
Meals			-0.02		0.05		0.09			
Fin										
Other							0.05			
R^2(%)	1.18	2.1	5.46	4.5	5.56	2.22	6.35	2.02	0.42	7.31

Table A2: This table reports the pure LASSO estimated coefficients in terms of magnitude and statistical significance. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables by LASSO. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

	Regressors									
Dependent	***Cnstr***	***Steel***	***FabP***	***ElcE***	***Autos***	***Carry***	***Mine***	***Coal***	***Oil***	***Util***
Food								-0.02		0.03
Beer								-0.03		
Smoke						***0.12***	-0.02	-0.02	-0.07	***0.21***
Games								-0.02		
Books								-0.02	-0.09	0.05
Hshld								-0.02		
Clths		-0.02	-0.11		0.02			-0.04	-0.07	0.07
Hlth							-0.03	-0.03		0.06
Chems										
Txtls					0.08			-0.05	-0.08	
Cnstr								-0.04	-0.05	0.07
Steel										
FabPr										
ElcEq										
Autos										
Carry										
Mines										
Coal										
Oil						0.05				
Util	-0.11		0.05	-0.03		-0.04	0.06	0.08		0.01
Telcm	-0.03				-0.02	-0.02	-0.01	-0.02	-0.07	***0.12***
Servs		-0.02							-0.05	0.03
BusEq										0.08
Paper								-0.02	-0.05	
Trans										
Whlsl						0.05		-0.03	***-0.11***	***0.17***
Rtail										
Meals		-0.06				0.01		-0.05	**-0.11**	0.06
Fin										
Other										
R^2(%)	4.37	1.12	1.21	0.38	4.77	1.37	-	2.47	1.2	6.75

Table A2 (continued): This table reports the pure LASSO estimated coefficients in terms of magnitude and statistical significance. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables by LASSO. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

 https://doi.org/10.26219/heal.aueb.6658

Regressors

Dependent	***Tlcm***	***Servs***	***BusEq***	***Paper***	***Trans***	***Whlsl***	***Rtail***	***Meals***	***Fin***	***Other***
Food							0.01			
Beer										
Smoke	-0.05	**-0.1**		-0.05	0.04				0.02	
Games									0.06	
Books		0.04	0.04				0.05		0.06	
Hshld							0.02			
Clths	-0.03	0.07	0.05				0.07		0.04	
Hlth										
Chems										
Txtls							0.08		0.11	
Cnstr					0.03		0.01		0.12	
Steel									**0.1**	
FabPr					0.02				0.06	
ElcEq									0.03	
Autos			0.05				0.15		0.06	
Carry					0.06					
Mines										
Coal				0.11			0.05			
Oil										
Util						-0.06			0.08	0.07
Telcm		0.01	0.03				0.06	-0.08	**0.12**	
Servs									0.08	
BusEq							0.03		0.04	
Paper							0.03		0.04	
Trans									0.03	
Whlsl	-0.05	0.03	0.01						0.06	0.04
Rtail							**0.06**			
Meals		0.06	0.03				0.05		0.03	
Fin									0.06	
Other									0.07	
R^2(%)	4.7	1.89	1.9	2.52	0.62	6.81	1.21	7.29	1.04	2.4

Table A2 (continued): This table reports the pure LASSO estimated coefficients in terms of magnitude and statistical significance. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables by LASSO. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Regressors

Dependent	***Foo***	***Beer***	***Smok***	***Games***	***Book***	***Hshld***	***Clths***	***Hlth***	***Chem***	***Txtls***
Food				0.01	0.07		0.03			0.00
Beer	**0.1**						0.04			
Smoke										0.05
Games			-0.07		***0.15***		0.03			
Books				0.03	0.04		0.02			
Hshld					0.04		**0.09**			
Clths					0.05		0.08			0.02
Hlth					***0.09***		0.03			
Chems				-0.11			***0.12***			
Txtls						-0.15	0.06		0.11	
Cnstr				-0.14			0.05			
Steel			-0.09	-0.11						
FabPr			-0.06	-0.14	0.07					
ElcEq			-0.07							
Autos			-0.03	***-0.14***	0.07	***-0.3***	0.04			
Carry		-0.08			0.03			***-0.14***		
Mines				-0.16						-0.09
Coal		***-0.27***	-0.08		0.16					
Oil		-0.05	-0.02			-0.12		-0.09		0.02
Util		***-0.1***								
Telcm		-0.08			0.06					
Servs			-0.09		0.07					
BusEq			***-0.14***	-0.1	***0.21***			0.03		-0.04
Paper					0.05		0.05			
Trans		-0.01			0.08		0.02	-0.01		
Whlsl			-0.08		***0.1***					
Rtail										
Meals			-0.06	0.05			0.09			
Fin					0.03		0.02			
Other			-0.07				0.06			
R^2(%)	2.61	2.70	5.79	5.68	6.59	3.76	7.18	2.94	2.79	9.96

Table A3: This table reports the OLS post-elastic net estimated coefficients in terms of magnitude and statistical significance after we have implied elastic net penalization to our full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables for the OLS regression by elastic net. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Regressors

Dependent	***Cnstr***	***Steel***	***FabP***	***ElcEq***	***Autos***	***Carry***	***Mine***	***Coal***	***Oil***	***Util***
Food						0.02	-0.04	**-0.05**		0.08
Beer						0.03	-0.03	***-0.05***		
Smoke						***0.14***	-0.03	-0.03	**-0.11**	***0.28***
Games			0.06					-0.06	-0.1	
Books		-0.07						-0.03	***-0.14***	**0.12**
Hshld		-0.07					-0.02	***-0.03***		
Clths		-0.07		***-0.25***		0.07		***-0.04***	-0.11	0.11
Hlth		-0.05					-0.04	***-0.03***		**0.1**
Chems					0.06	0.07		-0.03	-0.09	
Txtls			0.15	***-0.25***	**0.08**			***-0.08***	***-0.19***	
Cnstr						0.04		***-0.05***	***-0.15***	***0.14***
Steel		-0.05							-0.08	0.13
FabPr								-0.01	-0.08	
ElcEq				***-0.15***				-0.02	***-0.15***	***0.18***
Autos								-0.04	***-0.13***	***0.15***
Carry						0.04		-0.05	-0.1	***0.17***
Mines		-0.11	0.16		0.13	0.08		-0.06		***0.18***
Coal					0.02			**0.09**	***-0.19***	
Oil				0.08	0.02	***0.14***		-0.03		
Util	***-0.12***					**0.07**	-0.05			
Telcm							-0.02	-0.02	-0.1	**0.14**
Servs		0.11							-0.11	**0.13**
BusEq		-0.06			-0.06				-0.06	-0.09
Paper				-0.18				**-0.03**	-0.09	
Trans		-0.07		-0.11				-0.01	-0.12	***0.17***
Whlsl						0.05		-0.03	***-0.14***	***0.18***
Rtail		-0.11		-0.14					-0.07	
Meals		***-0.12***				0.04		**-0.05**	***-0.14***	0.11
Fin							-0.03	-0.03	***-0.1***	**0.11**
Other								-0.02	-0.08	
R^2(%)	6.07	2.87	3.28	3.65	6.96	5.04	4.09	3.51	3.54	4.62

Table A3 (continued): This table reports the OLS post-elastic net estimated coefficients in terms of magnitude and statistical significance after we have implied elastic net penalization to our full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables for the OLS regression by elastic net. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Regressors

Dependent	***Tlcm***	***Servs***	***BusE***	***Paper***	***Trans***	***Whlsl***	***Rtail***	***Meals***	***Fin***	***Other***
Food								0.01		
Beer								0.01		
Smoke	-0.1	***-0.14***			0.07					
Games		0.02							0.13	
Books		0.04	**0.1**					0.01	0.09	
Hshld							0.02		0.06	
Clths		0.09	0.11				0.05	-0.00	0.07	
Hlth										
Chems										
Txtls			0.05				**0.14**		***0.25***	
Cnstr		0.08			0.07				**0.17**	
Steel			0.11						***0.25***	
FabPr			0.07		0.08				0.16	
ElcEq		0.04	0.09					0.04	***0.15***	
Autos		0.02	***0.13***				**0.14**	0.1	***0.14***	
Carry		0.04	0.06		0.11				0.08	
Mines	***-0.18***		0.11					0.1		
Coal		0.18	-0.16	***0.26***		-0.02	0.11			
Oil										
Util	***0.12***									0.09
Telcm			0.04					-0.1	***0.13***	
Servs		0.02	0.05						0.13	
BusEq		0.02	0.07						0.12	
Paper		0.05			0.06			0.03	0.11	
Trans		0.08			0.08				0.13	0.06
Whlsl		0.03							0.04	0.04
Rtail		0.04	0.1				0.03		0.11	0.05
Meals		0.07	0.06					0.05	0.05	
Fin			0.03						0.09	0.04
Other		0.03			0.35				**0.15**	
R^2(%)	4.06	3.12	4.84	4.61	5.12	6.21	5.28	7.9	3.69	4.08

Table A3 (continued): This table reports the OLS post-elastic net estimated coefficients in terms of magnitude and statistical significance after we have implied elastic net penalization to our full predictor data set. In sample R^2 for each industry is also shown. Dependent corresponds to the dependent variable in each regression or the specific industry excess return we want to analyze. Regressor expresses the selected variables for the OLS regression by elastic net. Bold (italicized) represents the significant coefficient at the 10% (5%) level based on a conventional t-statistic.

https://doi.org/10.26219/heal.aueb.6658

Table A4: Table with all the possible combination of the hyperparameters for the Elastic Net model and their corresponding metrics

λ	P	MSE	Sharpe Ratio	Direction score	Kendalltau
1	0.01	42.58601	0.505243	0.465127	0.027792
1	0.05	42.38339	0.506767	0.465838	0.028903
1	0.1	42.14816	0.550575	0.465601	0.030578
1	0.5	40.82391	0.542905	0.462285	0.03324
1	0.667	40.50835	0.552524	0.465305	0.034857
1	0.75	40.38608	0.606174	0.465068	0.035413
1	0.9	40.20863	0.591598	0.46566	0.035159
1	0.95	40.15928	0.632497	0.465956	0.035314
1	0.99	40.12296	0.642982	0.466134	0.035796
1	1	40.11451	0.630688	0.465601	0.035935
3	0.01	41.45134	0.54078	0.465838	0.033877
3	0.05	41.11621	0.522352	0.46566	0.034188
3	0.1	40.76593	0.513852	0.461634	0.03605
3	0.5	39.76472	0.637196	0.464298	0.036873
3	0.667	39.69188	0.573119	0.462049	0.03345
3	0.75	39.68147	0.538382	0.464298	0.030729
3	0.9	39.67261	0.478649	0.463943	0.028948
3	0.95	39.66916	0.479178	0.463766	0.027338
3	0.99	39.66604	0.470328	0.462285	0.02603
3	1	39.66514	0.462993	0.462522	0.025695
10	0.01	40.26402	0.581564	0.464476	0.039113
10	0.05	39.91919	0.612711	0.465601	0.041578
10	**0.1**	**39.7274**	**0.707494**	**0.465838**	**0.042053**
10	0.5	39.69809	0.320724	0.464417	0.021209
10	0.667	39.76187	0.158754	0.467318	0.012014
10	0.75	39.79056	0.050685	0.465897	0.006852
10	0.9	39.82381	-0.07793	0.461989	0.000443
10	0.95	39.83124	-0.11157	0.461042	-0.00087
10	0.99	39.83479	-0.143	0.461516	-0.00136
10	1	39.83533	-0.13779	0.461338	-0.00141
30	0.01	39.66438	0.661428	0.468147	0.042167
30	0.05	39.59691	0.573101	0.465956	0.034799
30	0.1	39.63898	0.496526	0.462285	0.028242
30	0.5	39.84481	-0.17682	0.458141	-0.00306
30	0.667	39.84481	-0.17682	0.458141	-0.00306

https://doi.org/10.26219/heal.aueb.6658

30	0.75	39.84481	-0.17682	0.458141	-0.00306
30	0.9	39.84481	-0.17682	0.458141	-0.00306
30	0.95	39.84481	-0.17682	0.458141	-0.00306
30	0.99	39.84481	-0.17682	0.458141	-0.00306
30	1	39.84481	-0.17682	0.458141	-0.00306
100	0.01	39.565	0.526888	0.466844	0.0334
100	0.05	39.73922	0.255819	0.467614	0.015162
100	0.1	39.84009	-0.16248	0.458911	-0.00193
100	0.5	39.84481	-0.17682	0.458141	-0.00306
100	0.667	39.84481	-0.17682	0.458141	-0.00306
100	0.75	39.84481	-0.17682	0.458141	-0.00306
100	0.9	39.84481	-0.17682	0.458141	-0.00306
100	0.95	39.84481	-0.17682	0.458141	-0.00306
100	0.99	39.84481	-0.17682	0.458141	-0.00306
100	1	39.84481	-0.17682	0.458141	-0.00306

In this Table we perform cross validation for the preselected values of λ and ρ in order to attain the optimal values of the elastic net tuning parameters. For λ=10 and ρ=0.1 we get the maximum Sharpe Ratio=0.707 and a relatively small MSE= 39.727. Kendall's Tau is a measure of rank correlation. If it is 1, we always predict the ranking correctly. If it is -1, always wrong. If it is 0 predictive ranking gives no information.