Machine learning portfolios with equal risk contributions

This version: November 2019

Alexandre Rubesam[a,*]

[a] *Department of Finance, IESEG School of Management (LEM-CNRS 9221), Paris, France*

Abstract

We use machine learning methods to forecast individual stock returns in the Brazilian stock market, using a unique data set including technical and fundamental predictors. We find that portfolios formed on the highest quintile of predicted returns significantly outperform market benchmarks. However, portfolios formed on the lowest quintile of predicted returns earn positive returns and have high volatilities, making traditional long-short strategies unnatractive. To resolve this problem, we propose an equal risk contribution (ERC) ensemble approach to build a portfolio combining long-short portfolios obtained with various machine learning methods such that (i) the risk contributions of all individual long-short portfolios are equal, and (ii) the aggregate risk contribution of all long positions equals that of all short positions. The ERC ensemble portfolio outperforms, on an after-cost, risk-adjusted basis, all individual machine learning long-short portfolios, as well as equally-weighted ensembles of these portfolios.

Keywords: Finance, machine learning, stock market prediction, portfolio optimization, equal risk contribution

JEL: C53, G11, G15.

[☆] We would like to thank participants in the Finance Hub research seminar at INSPER in São Paulo, Brazil, for useful suggestions.

[*] Corresponding author. IESEG School of Management, 1 Parvis de La Défense - F-92044 Paris La Défense, France. Tel: +33 (0)1435 882622.

Email address: a.rubesam@ieseg.fr (Alexandre Rubesam)

1. Introduction

The literature on stock return predictability and empirical asset pricing has identified hundreds of characteristics that appear to predict stock returns (Jacobs, 2015; Harvey et al., 2016; Green et al., 2017). Historically, most studies in the finance literature have relied on linear econometric techniques, sorts on firm characteristics, or rankings based on ad-hoc combinations of fundamental variables. These methods are not well-suited to deal with the high dimensionality of modern financial applications, or potential nonlinearities. Although machine learning (henceforth, ML) methods have long been used in financial applications, most such studies have been published in the operations research and machine learning literature. Recently, there has been a renewed interest in the topic within the finance community, including both academics and practitioners.

Several studies using U.S. data make it clear that ML methods and techniques can be extremely useful to understand the drivers of stocks returns (Gu et al., 2018*a*; Kelly et al., 2019; Kozak et al., 2019) and to develop profitable, sophisticated statistical arbitrage strategies (Krauss et al., 2017; Fischer and Krauss, 2018; Huck, 2019). However, fewer studies have examined the use of ML to predict stock returns in emerging markets, and the existing studies often have limitations regarding the number and type of predictors used, and the ML methods tested. Emerging markets tend to be extremely volatile (Bekaert and Harvey, 1997; Bekaert et al., 1998; Hwang and Satchell, 1999*a*), and therefore it is not reasonable to assume that the results obtained in the U.S. market apply directly to emerging markets. Moreover, the abundance of ML methods creates a dilemma for portfolio managers seeking to leverage these technologies: which method(s) should be used? Should forecasts or portfolios obtained using different ML methods be combined, and if so, how?

Given this context, the present paper has two main objectives. The first one is to investigate the use of different ML methods to predict stock returns and create portfolios in an emerging market, namely the Brazilian equities market. We use a comprehensive dataset of technical and fundamental predictors, provided by a local asset manager, that is comparable to those used in studies with U.S. data. The second objective is to develop an approach to combine multiple long-short portfolios based on ML forecasts, without itself relying on predictions of which ML method will perform better in the future. To this end, we propose an equal risk contribution (ERC) approach to construct an ensemble portfolio that seeks to balance the risk contributions of different long and short portfolios obtained via ML.

Our dataset includes 62 technical and fundamental indicators. When dummy variables are included to represent different equity sectors, the total number of predictors is 86. We train over thirteen ML methods including simple linear regression, linear regression models with regularization via lasso and ridge, linear models with dimension reduction via principal components regression and partial least squares, linear models using Bayesian variable selection, random forests, gradient boosting, and neural networks with different numbers of hidden layers. Models are trained using an expanding training window approach; hyperparameters are selected via a separate validation window, and optimal models are used to recursively predict stock returns out of sample. Our results suggest that portfolios of stocks in the top quintile of predicted returns using different ML methods significantly outperform market benchmarks, with the best results being obtained with neural networks with three or four hidden layers. However, all ML portfolios suffer from large maximum drawdowns and high turnover. Additionally, portfolios of stocks in the lowest quintile of predicted returns have high volatility and earn low, but positive returns, making traditional long-short strategies unattractive. The ERC ensemble of ML portfolios we propose dynamically adjusts allocations to the long and short ML portfolios, mitigating the large differences in risk between them. It outperforms, on a risk-adjusted basis, all individual ML methods, as well as equally-weighted ensembles of these methods, while having a maximum drawdown that is a fraction of theirs. The results remain economically significant after accounting for transaction costs.

The rest of this paper is organized as follows. Section 2 is a short literature review on stock return prediction and the recent use of ML in financial applications. Section 3 presents the data set used in the study. Section 4 explains the methodology used to train the various models used to forecast returns. Section 5 presents the empirical results. Our conclusions are presented in section 6.

2. Literature review

There have been many studies on stock return predictability in the finance and accounting literature, using a variety of methods and different sets of predictors. Historically, these studies have focused on linear models (e.g. Haugen and Baker, 1996; Campbell and Thompson, 2007; Lewellen, 2014; Green et al., 2017), procedures to sort stocks into portfolios based on firms' characteristics (e.g. studies following the approach introduced by Fama and French, 1993), or the creation of ad hoc measures combining different fundamental variables (Piotroski, 2000; Mohanram, 2005). The focus in this literature is often the identification and testing of whether one or more variables represent priced risk factors about which investors care. Although there is no consensus on the exact set of priced factors, firm characteristics that are useful to predict future returns in a way that cannot be explained by risk loadings in widely accepted risk factors are often termed “anomalies”. There are now hundreds of such anomalies, see for example Jacobs (2015), Harvey et al. (2016), and Green et al. (2017).[1]

The approaches developed in the finance and accounting literature are not well-suited to deal with the large and increasing number of predictors, vast amounts of data, or the potential for non-linear dynamics between returns and the predictors. ML methods, which have been developed precisely to deal with these issues, have long been used in financial applications. Given financial economists' preference for linear econometric methods, however, much of this literature is published in operational research and machine learning journals, as highlighted by Huck (2019). For example, Atsalakis and Valavanis (2009) provide a survey of over 100 articles focusing on the use of ML to forecast stock markets. Hsu et al. (2016) contrast studies using ML with those using regular (linear) econometric techniques, concluding that the best ML methods are superior to the best econometric methods in terms of accuracy. Sermpinis et al. (2013) provides further references focusing on the use of ML to predict changes in exchange rates. Huck (2009, 2010) apply neural networks and multiple-criteria decision methods to forecast returns and select pairs in a pairs trading strategy. Kaucic (2010) considers a genetic algorithm coupled with ML methods to develop a trading system using common technical indicators.

Recently, there has been a surge of interest in ML methods within the finance and economics fields.[2] Recent studies in empirical asset pricing, for example, focus on the application of ML techniques like regularization or dimension reduction via principal component analysis (PCA) to identify the most relevant drivers of asset returns (see for example Feng et al., 2017; Freyberger et al., 2017; Kelly et al., 2019; Kozak et al., 2019; Lettau and Pelger, 2018). Gu et al. (2018*b*) examine a number of ML methods to forecast monthly individual stock returns in the U.S. market using about 100 fundamental and technical features, and their interactions with macroeconomic variables. Their results suggest that nonlinear models such as neural networks can significantly improve predictions relative to simple linear regression approaches, and that long-short strategies based on several ML methods appear to remain profitable in recent periods.

Another fertile area of research is the application of ML methods and concepts to portfolio formation. Ban et al. (2016) apply the concepts of regularization and cross-validation to portfolio optimization. DeMiguel et al. (2019) use a LASSO technique to select characteristics in a parametric portfolio problem. Heaton et al. (2017) explore deep learning to form portfolios, and provide an application of this approach to create portfolios to track or outperform an index. Kolm and Ritter (2019) provide an overview of applications of reinforcement learning in finance, including

[1] Despite the large number of predictors, recently published papers in top finance journals still rely on linear regression via ordinary least squares, see for example Green et al. (2017).

[2] See Varian (2014) and Mullainathan and Spiess (2017) for discussions on the role of big data and ML in the econometrics toolbox.

mean-reversion trading strategies, derivatives pricing and optimal hedging. De Spiegeleer et al. (2018) apply ML to various problems in quantitative finance including calculation of option prices and greeks (sensitivities of option prices to model inputs). Kyriakou et al. (2019) apply simple ML methods to forecast annual stock returns to use as benchmarks for pension planning. Chen et al. (2019) develop a sparse-group lasso methodology for portfolio selection that allows investors to express preferences over equity sectors, and show its connection to robust portfolio selection (Kim et al., 2018).

Whereas the focus on asset pricing is in identifying a small set of priced risk factors that determine asset returns in lower frequencies (monthly or annual), practitioners are usually interested in exploiting ML to develop profitable trading strategies, which often operate in a higher frequency (measured from days to fractions of a second). Some recent studies suggest ML methods can be used to build profitable long-short statistical arbitrage strategies, although the profitability seems to be decreasing or even negative in recent periods, consistent with arbitrageurs increasingly exploiting the market inefficiencies uncovered by these methods. Examples of these studies include Krauss et al. (2017), Fischer and Krauss (2018), and Huck (2019), who apply ML models including deep learning, gradient boosted trees, and random forests to forecast daily stock returns using different lags of individual stock returns.

The decline in the profitability of ML strategies in the U.S. market is in line with that reported by studies such as Green et al. (2017), who document a decrease in the profitability of a long-short portfolio using linear regression forecasts since the early 2000s. The authors link this to changes in the regulatory and trading environments, which have made it cheaper and easier to implement quantitative long-short trading strategies exploiting a large number of signals. This is also consistent with the results of McLean and Pontiff (2016), who document significant out-of-sample and post-publication declines in the returns of predictors published in academic journals, suggesting arbitrageurs actively exploit new predictors as they become known.

The situation in emerging markets is less clear, due to the much smaller number of studies in these markets, the fact that most studies focus either on the prediction of stock market indices, or use a small number of predictors. For example, out of more than 100 studies surveyed by Atsalakis and Valavanis (2009) and Hsu et al. (2016), only a handful investigate emerging markets. In an earlier work, Campbell (2000) investigated the use of neural networks to predict emerging market stock indices using lagged returns, concluding that an active strategy based on neural network forecasts beats a passive strategy or an active strategy based on linear regression forecasts. A few studies have investigate the use of ML methods to predict individual stock returns in emerging markets. Cao et al. (2005, 2011) uses neural networks to forecast individual stock returns in China, however, the number of predictors is limited to the factors in the Fama and French (1993) models. Raposo and Cruz (2002) used fuzzy neural networks to predict individual stock returns in the Brazilian market using fundamental indicators. However, their data is limited to five fundamental indicators, and the analysis is focused on stocks belonging only to one sector. To our knowledge, ours is the first study to systematically compare multiple ML methods to predict individual stock returns in an emerging market, using a large number of technical and fundamental predictors.

3. Data

Our data set includes 572 Brazilian stocks over the period from January 2003 to December 2018. In order to have a universe of reasonably tradable stocks, a few minimum requirements are imposed. First, a minimum of one year of trading data and two years of accounting data is required. Second, minimum market capitalization and liquidity requirements are imposed, to eliminate stocks which are too small or illiquid to use in a realistic trading strategy. These restrictions are applied at each month, and thus result in a variable set of eligible stocks.

For each stock and each month, we have data on 62 predictors. We also add 24 dummy variables to represent the firm sectors, bringing the total number of variables to 86. These can be broadly classified into five categories, shown in Table 1.[3]

Table 2 reports summary statistics on all stocks which are eligible for the whole of each year. The number of eligible stocks varies from 54 in 2003 to 198 in 2010 and 2011. Individual stocks in the Brazilian market are very volatile: the average standard deviation of monthly returns is above 10% in most years. This is a feature of emerging markets which has been reported in several studies (Bekaert and Harvey, 1997; Bekaert et al., 1998; Hwang and Satchell, 1999*b*). Individual stock returns are also positively skewed in general.

4. Methodology

4.1. Training, validation, and test windows

Our objective is to build regression models to forecast stocks returns at time $t + 1$, based on the value of predictors at time t. We apply the usual approach of dividing the data into training, validation, and test sets. The training set is used to fit the models. The validation set is used for hyperparameter tuning within a class of models (for example, to choose the optimal penalty parameter in LASSO). We do so by calculating the mean squared error

[3] Due to the proprietary nature of the data, we do not disclose the exact constructions of all variables. However, we note that these variables are comparable to those commonly used in factor investing (e.g., Ang, 2014) and large-scale stock predictability studies such as McLean and Pontiff (2016) and Green et al. (2017).

Table 1: Summary of predictors used in this study

Category	Number of predictors	Examples
Growth	12	Historical growth of EBIT (Earnings Before Interest), Historical growth of EPS (Earnings per share); Trend of profit margin
Quality	27	Volatility of sales; volatility of earnings; measures of stability of return on equity and return on invested capital
Risk	6	Volatility, downside volatility
Technical	10	Previous returns over various windows, technical analysis indicators
Value	7	Price-to-Book, Price-to-Earnings, Enterprise Value to Sales
Sector	24	Sector dummy variables

Table 2: Summary statistics of available stocks per year

Every month, the monthly returns of all available individual stocks are collected. We then calculate the cross-sectional average, standard deviation, skewness, and kurtosis of the returns for that month. The table reports, for each year, the average number of available stocks in each month, the average of the cross-sectional statistics, and the 5-th and 95-th percentiles of all returns during that year.

Year	# stocks	Average return	Average Std dev	Average skewness	Average kurtosis	Percentile 5%	Percentile 95%
2003	54	6.88%	2.85%	1.20	7.07	-10.17%	26.89%
2004	67	3.26%	10.16%	0.39	4.18	-13.15%	23.21%
2005	75	1.67%	11.14%	0.90	6.64	-16.15%	21.86%
2006	87	3.73%	10.31%	1.09	6.69	-12.43%	23.38%
2007	124	2.93%	11.29%	1.26	7.89	-14.34%	20.12%
2008	163	-4.30%	12.49%	0.39	5.33	-30.26%	20.84%
2009	178	7.30%	12.74%	1.11	7.10	-11.25%	32.27%
2010	198	1.60%	10.14%	1.03	9.92	-13.07%	18.21%
2011	198	-1.27%	10.45%	1.24	17.39	-15.76%	13.69%
2012	192	1.53%	10.51%	0.22	7.27	-16.05%	18.23%
2013	186	-0.92%	10.54%	0.43	11.06	-17.06%	14.06%
2014	187	-1.46%	10.75%	0.58	10.52	-19.18%	15.48%
2015	179	-1.69%	13.84%	0.94	10.04	-23.53%	20.53%
2016	172	2.71%	15.99%	1.17	12.96	-19.59%	30.79%
2017	179	3.22%	11.72%	1.24	8.95	-13.76%	23.47%
2018	189	0.84%	12.45%	1.14	12.65	-19.02%	23.09%

(MSE) of forecasts in the validation set, and choosing the value of the hyperparameter(s) with the lowest MSE. Finally, out-of-sample predictions are made for observations in the test set. We start with a training window of 24 months, a validation window of 12 months, and a test set of 6 months. We then expand the training window by six months, and move the validation and testing windows six months forward. This process is repeated until the end of the sample.

We use a pooled-data approach, stacking individual stock returns over all months of each training window into a single vector, and their predictors into a single matrix.[4] Let $y_{i,t+1}$ denote the return on stock i at month $t + 1$, and let n_t be the number of eligible stocks in month t. Let τ_k contain the indices corresponding to the months in the k-th training window. Pooling all returns yields a vector $\mathbf{y}_{Tr,k}$ of dimension $\left(\sum_{t \in \tau_k} n_t\right) \times 1$. Likewise, assume that there are $p_{Tr,k}$ predictors for which there is data available for the whole training window. Let x_{it} be the column vector of the $p_{Tr,k}$ predictors for stock i, at month t. Stacking all predictors produces a matrix $\mathbf{X}_{Tr,k}$ of size $\left(\sum_{t \in \tau_k} n_t\right) \times p_{Tr,k}$. The regression models for the k-th training window thus have the following specification:

$$\mathbf{y}_{Tr,k} = f(\mathbf{X}_{Tr,k}) + \boldsymbol{\varepsilon}_{Tr,k}, \tag{1}$$

where $f(\cdot)$ represents a functional form and $\boldsymbol{\varepsilon}$ is an error term. We next describe specific choices for f.

4.2. Classes of models for f

We briefly review the types of models employed in our study. Most models are explained in standard textbooks such as Friedman et al. (2001). For simplicity, we drop the subscripts and use $\mathbf{y}$ and $\mathbf{X}$ for the response variable and matrix of predictors, respectively, assuming that $\mathbf{y} = (y_1, \ldots, y_n)'$ has n elements, and $\mathbf{X}$ is an $n \times p$ matrix.

4.2.1. Linear models via OLS

The usual regression model estimated via ordinary least squares (OLS) corresponds to $f(\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$. There are no hyperparameters.

4.2.2. Ridge regression and LASSO

Both ridge regression and LASSO shrink regression coefficients by imposing a penalty on their size, but ridge regression imposes an L_2 penalty, whereas LASSO uses an L_1 penalty. The coefficients are obtained as the solution to the following problem:

[4] The data are transformed to the $(-1, 1)$ interval at each month. This transformation is also applied to the response variable. This removes outliers and eliminates the necessity of applying robust loss functions, simplifying the hyperparameter tuning.

$$\hat{\boldsymbol{\beta}}^{ridge} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda J(\boldsymbol{\beta}) \right\}, \tag{2}$$

where the penalty term is $J(\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j^2$ for ridge regression or $J(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|$ for LASSO. The model is estimated for a grid of values of the penalty parameter λ. The optimal parameter is the one that minimizes the MSE in the validation set.

4.2.3. Principal Components Regression (PCR)

Principal Components Regression (PCR) is a dimension reduction technique that uses linear transformations of the data (the principal components) as the predictors. Let $Z_1, \ldots, Z_M$ represent $M < p$ linear combinations of the original variables:

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j, \quad m = 1, \ldots, M. \tag{3}$$

A linear regression of $\mathbf{y}$ on the transformed variables $\mathbf{Z}$ can be represented as a linear combination of the original variables:

$$\begin{aligned} y_i &= \theta_0 + \sum_{m=1}^{M} \theta_m z_{im}, i = 1, \ldots, n \\ &= \theta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \end{aligned} \tag{4}$$

where $\beta_j = \sum_{m=1}^{M} \theta_m \phi_{mj}$. For PCR, the Z_m are the principal components of the data. The number of principal components (M) in the regression model (4) is chosen based on the validation set MSE.

4.2.4. Partial Least Squares (PLS)

Partial Least Squares (PLS), like PCR, considers linear combinations of the inputs for the regression, but it also makes use of the response variable $\mathbf{y}$. PLS starts by setting $\phi_{j1}, j = 1, \ldots, p$ in equation (3) as the linear regression coefficient of $\mathbf{y}$ onto each X_j. Once Z_1, the first partial least squares direction, is obtained in this way, the first set of coefficients $\hat{\theta}_1$ is obtained as the regression coefficient of $\mathbf{y}$ on Z_1. Then, $X_1, \ldots, X_p$ are orthogonalized with respect to Z_1. This process is continued until $M \leq p$ directions are obtained. The number of partial least squares dimenions (M) in the regression model (4) is chosen based on the validation set MSE.

4.2.5. Bayesian Variable Selection

There are many methods for Bayesian variable selection in regression models, see e.g. George and McCulloch (1997) and O'Hara and Sillanpää (2009). We focus on the method proposed by Smith and Kohn (1996) for the linear regression model, due to its simplicity and efficiency. Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{5}$$

where $\mathrm{Var}(\varepsilon) = \sigma^2$. The variable selection method works by introducing a vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)'$ of latent dummy variables, where $\gamma_j = 1$ if the $j - th$ variable is included in the model, and zero otherwise. For a given value of $\boldsymbol{\gamma}$, let $\mathbf{X}_\gamma$ represent the matrix of regressors corresponding to those elements of $\boldsymbol{\gamma}$ that are equal to one, and let $\boldsymbol{\beta}_\gamma$ contain the corresponding elements of $\boldsymbol{\beta}$. A hierarchical prior is assumed for $\boldsymbol{\beta}_\gamma | \boldsymbol{\gamma}, \sigma^2$, of the form $\boldsymbol{\beta}_\gamma | \boldsymbol{\gamma} \sim N(0, c\sigma^2(\mathbf{X}'_\gamma \mathbf{X}_\gamma)^{-1})$, where c is a hyperparameter which Smith and Kohn (1996) suggest be set between 10 and 1000. The prior for σ^2 given γ is $p(\sigma^2 | \boldsymbol{\gamma}) \propto 1/\sigma^2$.

Given these choices, it is possible to simulate the relevant conditional posterior distributions using the Gibbs sampler. We consider three values for c: 100, 500, and 1000. For each one, we simulate the distribution 10,000 times, discarding the first 5,000 simulations as a "burn-in" period. The forecast from this method is obtained as the average of the forecasts across each simulated value of $\boldsymbol{\gamma}$.

4.2.6. Random Forests and Boosting

We apply two ensemble methods that make use of regression trees: random forests and boosting. Tree-based methods work by partitioning the feature space into a set of distinct and non-overlapping rectangular regions $R_1, \ldots, R_M$, by creating splits in the predictors. The prediction of a tree is constant for all observations in each region. For a regression tree, this is simply the average of the observations in that region.

Trees are estimated using a recursive binary splitting algorithm which determines, at each step, a combination of a variable and a split point that minimize the forecast error at that stage. The complexity of a tree is thus a function of the number of splits and regions in the tree. Several strategies have been developed to decide when to stop growing a tree, or to prune a large tree in order to avoid overfitting, see for instance Breiman (1984) and Quinlan (1993).

Despite their advantages and interpretability, trees are methods with high variance: small perturbations to the data usually lead to very different trees. Bootstrap aggregation or bagging (Breiman, 1996) is a variance-reduction technique for methods like trees, which combines many trees grown on bootstrapped versions of the data. For a regression problem, the bagged model amounts to averaging the prediction of each bootstrapped tree. Random forests (Breiman, 2001) are a modification of bagging that attempts to build less correlated trees by randomly selecting a subset of the predictors in each bootstrapped sample. Besides the parameters controlling the growth of individual trees, the hyperparameters of random forests are the number of bootstrap samples or trees and the number of variables to sample. Because random forest are quite robust to overfitting regarding the number of trees, we set the number of trees to a large value (1000), and choose the number of variables to sample using the validation sample.

Boosting was originally developed for classification problems (Schapire, 1990), and relies on the idea of combining many weak classifiers (models whose error rates are just slightly better than random guessing), trained on sequentially modified versions of the data, to obtain a powerful ensemble with better performance. At each iteration of the algorithm, observations in the training set are multiplied by a weight, and a model (often a tree) is fitted to the modified data. Observations that were misclassified in the previous iteration have their weights increased, while the opposite is true for correctly classified observations. Many boosting algorithms have been developed, see e.g. Friedman et al. (2000) and Freund and Schapire (1997). Friedman (2001) proposed a paradigm for function approximation based on additive expansions using steepest descent called gradient boosting, which can be used for classification and regression problems. We use the LS_Boost algorithm proposed in that paper for regression problems with a MSE criterion. The hyperparameters of the methods are the number of splits of the tree used in each iteration, the number of iterations, and a shrinkage or learning rate, which controls the contribution of each tree added to the ensemble. These parameters are chosen based on the MSE in the validation sample.

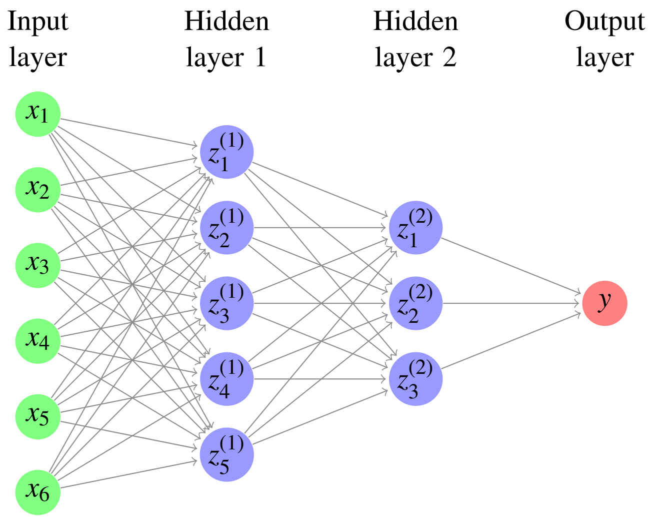
4.2.7. Neural Networks

Neural networks are a class of very flexible non-linear models that were developed in the artificial intelligence and statistics literature. We focus on feed-forward neural networks for regression problems.[5] These models are typically represented using a network diagram, such as the one on Figure 1, which represents a multi-layer neural network with six inputs (x_1 to x_6), two hidden layers with five and three neurons, whose outputs are represented by $z_1^{(1)}$ to $z_5^{(1)}$ and $z_1^{(2)}$ to $z_3^{(2)}$, respectively, and a single output y. The output of a neuron of a given layer is obtained by applying an *activation function* g to a linear combination of the values reaching that neuron from the previous layer. A commonly used activation function for regression problems is the sigmoid function, $g(x) = 1/(1 + e^{-x})$. The

[5] We provide only a brief overview of the type of neural network used in this study. More details on neural networks can be found, e.g. in Bishop (1995) and Ripley (1996).

complexity of a neural network is thus a function of the number of hidden layers, the number of neurons in each layer, and the activation functions used. It can be shown that neural networks are *universal approximators*: they can approximate arbitrarily well any continuous function if the number of neurons or layers is allowed to increase.[6]

Figure 1: Diagram of a feedforward neural network with two hidden layers



The figure shows a neural network with two hidden layers, and a single output. The input layer has six inputs, labelled x_1 to x_6. The first and second hidden layers have five and three neurons, respectively. The outputs from hidden layer neurons are represented by $z_j^{(l)}$, where $l \in 1, 2$ represents the hidden layer, and j identifies the neuron. They are calculated by applying an activation function to a linear combination of the preceding connections.

Consider a neural network with L hidden layers, where the number of neurons in layer l is $M^{(l)}$. Let $z_j^{(l)}$ be the output of neuron j of the hidden layer l. For the first hidden layer, these are simply the original inputs, i.e. $z_j^{(0)} = x_j, j = 1, \dots, p$. Assuming that $x_1 = 1$, a linear combination of the xs includes an intercept term, and so we can write $z_j^{(l)} = g(\boldsymbol{\beta}^{(l-1)\prime} z^{(l-1)})$, where $\boldsymbol{\beta}^{(l-1)}$ is a vector of weights to be estimated, and $z^{(l-1)} = (z_1^{(l-1)}, \dots, z_{M^{(l-1)}}^{(l-1)})$. Fitting a neural network consists in finding the weights $\boldsymbol{\beta}^{(l-1)}$ in each layer by minimizing a criterion such as MSE. This is done using a gradient descent method and what is known as the *backpropagation* algorithm. Because neural networks can be extremely flexible, regularization techniques such as adding a penalty term to the optimization criterion or early stopping are usually applied, to avoid overfitting. In this paper, we apply the latter, using the MSE in the validation sample to decide when to stop the training.

[6] This has been proved by Cybenko (1989) for single-layer networks with sigmoid activation functions. See Lu et al. (2017) for deep networks.

We consider neural networks similar to those in Gu et al. (2018*b*), i.e. networks with one (NN1) to five (NN5) hidden layers, with a pyramidal structure for the number of neurons in each hidden layer. The number of neurons in the first hidden layer is 32, and the number of neurons in each subsequent layer (if any) is half the number of neurons in the previous layer. Thus, NN1 has 32 neurons in its single hidden layer, NN2 has 32 neurons in the first layer and 16 in the second layer, and so on. We use a sigmoid activation function. In addition to early stopping, we also employ an ensemble approach. For each topology of neural network, 50 independent neural networks are trained, and their results are averaged to form a combined forecast. This approach reduces the variance of the individual models due to the fact that the weights are randomly initialized.

4.3. Portfolio formation, performance and cost calculations

4.3.1. Long-short portfolios based on individual ML methods

We form equally-weighted portfolios based on quintiles of the predicted returns obtained with different ML methods.[7] These portfolios are referred to as the baseline ML portfolios, and labelled P_1 (lowest quintile of predicted returns) to P_5 (highest quintile). The choice of using quintiles is based on the much smaller number of stocks in the Brazilian market, compared with the U.S. market.[8] We form different long-short portfolios using these baseline portfolios, which differ in the amount of leverage used and their total net exposure.

First, a traditional long-short strategy for a given ML method is obtained by going long the stocks in the top quintile of predicted returns (P_5) and short the stocks in the lower quintile (P_1). The returns of this strategy are simply the difference between the returns on the P_5 and P_1 portfolios. These long-short portfolios have a net exposure of zero and a leverage ratio of 2.

An alternative to a traditional long-short strategy is a 130/30 portfolio, which holds 130% of its capital in long positions and 30% in short positions. Unlike traditional long-short strategies, 130/30 portfolios retain a net exposure of 100%, while adding a degree of leverage and attempting to benefit from short positions.[9] For each ML method, we form 130/30 portfolios that are long 1.3 times the P_5 and short 0.3 times the P_1 portfolio. The leverage ratio of these portfolios is equal to 1.6.

Finally, considering that stocks in the lower quintile are expected to be more volatile, we consider a simple way to balance the risk contributions of the long and short components of a long-short strategy. We dynamically

[7] We note that, although we obtain different forecasts of individual stock returns, we do not attempt to solve a traditional mean-variance problem using these forecasts, because these estimates are very noisy and mean-variance portfolio optimization is extremely sensitive to these inputs, i.e. Best and Grauer (1991); Michaud (1989).

[8] The results obtained using different percentiles to define the long and short portfolios are qualitatively similar.

[9] See for example Lo and Patel (2008) and Johnson et al. (2007).

adjust the weight of the short portfolio by using an Equal Risk Contribution (ERC) criterion with a volatility risk measure.[10] Consider a portfolio which is long asset 1 with a weight w_1, and short asset 2 with a weight equal to $w_2 = -\kappa w_1$. Then it can be easily verified (see Appendix A) that the value of κ that achieves equal risk contributions between the long and short positions, for any value of w_1, is $\kappa = \sigma_1/\sigma_2$, where σ_1 and σ_2 are the volatilities of the two assets. We consider portfolios with $w_1 = 1$ and, to simplify calculations, we use a simple average of the volatilities of the assets in P_5 and P_1 to estimate the volatilities of each portfolio.

4.3.2. Long-short portfolios combining multiple ML methods

In practice, it is not possible to know in advance which ML method will perform best out-of-sample. A portfolio manager hoping to use ML methods to create a single long-short portfolio thus needs to deal with the issue of whether and how to combine forecasts or portfolios obtained using different ML methods. In principle, combining portfolios obtained with different ML methods seems desirable, due to the potential diversification benefits. We consider two approaches to do so that do not rely on any forecast of which method will perform better. The first approach is to create equally-weighted combinations of the long-short portfolios obtained with each method, an approach that seems naïve, but tends to work well in practice, e.g. DeMiguel et al. (2009).[11] The equally-weighted combinations of the traditional long-short and 130/30 portfolios are referred to as EW^{LS} and $\text{EW}^{130/30}$, respectively. If m is the number of ML methods, the weights of an equally-weighted combination of individual methods are obtained as:

$$w_{it}^{\text{EW}} = \frac{1}{m} \sum_{s=1}^{m} w_{it}^{s}, \tag{6}$$

where w_{it}^{s} is the weight of stock i in strategy s at time t.

For traditional long-short portfolios and 130/30 portfolios, the equally-weighted approach produces long-short portfolios with the same net exposure and leverage as the individual long-short strategies. For ERC long-short portfolios, there is no guarantee that the average of the long-short ERC portfolios using each method will generate a portfolio with equal risk contributions in the long and short legs. Instead, we propose an approach to ensure that (i) the risk contributions of the long-short portfolios under each ML method are balanced, and (ii) that the overall risk contributions of the long and short positions are equal. We form these portfolios based on defining allocations

[10] For a description of ERC or "risk parity" methods, see for example Qian (2005), Maillard et al. (2010) and Roncalli (2016). See Bertrand and Lapointe (2018) for an application of other risk-based strategies for portfolio construction using socially responsible investments.

[11] Other studies use equally-weighted ensembles of the forecasts of each ML method, see Krauss et al. (2017). For a review of different approaches to aggregate long-only portfolios, including a utility-based approach, see Bonaccolto and Paterlini (2019).

to the long and short portfolios associated with each strategy.

In order to formalize this idea, let $\mathbf{r}_t$ be a $2m \times 1$ vector of returns, such that the first (last) m columns contain the returns on the long (short) portfolios of each method at time t, and let $\mathbf{\Sigma}_t$ be its covariance matrix at time t. We drop time subscripts to simplify the notation in what follows. A long-short combination of the $2m$ portfolios can be represented by the $2m \times 1$ vector of weights $\boldsymbol{\omega} = (\omega_1^L, \ldots, \omega_m^L, \omega_1^S, \ldots, \omega_m^S)'$. The first m elements of $\boldsymbol{\omega}$ are positive, while the remaining m elements are negative. We can think of the combination as a portfolio of pair trades, where each pair trade is long the P_5 portfolio and short the corresponding P_1 portfolio for a given ML method. The return on the combination portfolio is $r^C = \boldsymbol{\omega}'\mathbf{r}$, and its volatility is given by $\sigma^C = \sqrt{\boldsymbol{\omega}'\mathbf{\Sigma}\boldsymbol{\omega}}$. We consider forming portfolios with the following characteristics. First, the portfolio has an overall long allocation of 100%, i.e. $\sum_{s=1}^m \omega_s^L = 1$. Second, the allocation to the short leg of each method is a fraction κ of the long allocation: $\omega_s^S = -\kappa\omega_s^L$. As a result, the overall short allocation is equal to κ.[12] Third, let RC_s^L and RC_s^S be the risk contributions of the long and short legs of method s to the combined portfolio.[13] The total risk contribution of the long-short portfolio associated with method s is $RC_s^{LS} = RC_s^L + RC_s^S$, and the total risk contribution of all long and short positions are $RC^L = \sum_{s=1}^m RC_s^L$ and $RC^S = \sum_{s=1}^m RC_s^S$, respectively. We then impose the following conditions:

$$RC_s^{LS} = RC_{s'}^{LS}, \quad s, s' \in \{1, \ldots, m\} \tag{7}$$

$$RC^L = RC^S \tag{8}$$

Condition (7) states that the contribution of each long-short portfolio should be equal for all ML methods. Condition (8) states that the overall risk contributions of all long and short positions are the same. In order to solve this non-standard ERC problem, we define the following quantities. Let $\boldsymbol{\eta} = (\omega_1^L, \ldots, \omega_{m,t}^L, \kappa)'$ denote a vector containing the weights of the long portfolios and the short multiplier κ. Since the weights on the short portfolios are defined as $\omega_s^S = -\kappa\omega_s^L$, the vector $\boldsymbol{\eta}$ determines the full allocation to the long and short portfolios. We collect the risk contributions in a $(m+2) \times 1$ vector $\boldsymbol{RC} = (RC_1^{LS}, \ldots, RC_m^{LS}, RC^L, RC^S)'$. Next, we define the risk budgets as follows. The risk budget for each long-short portfolio is $b_s^{LS} = 1/m, \quad s = 1, \ldots, m$. The risk budget for all

[12] In principle, it could be possible to find a solution that does not impose this restriction. We attempted such a solution in this study and found that, in general, it is not feasible, due to the high volatilities of the P_1 portfolios.

[13] The risk contributions of the long and short legs of method s are calculated as

$$RC_s^L = \omega_s^L \frac{\partial\sigma^C}{\partial\omega_s^L} = \omega_s^L \frac{(\Sigma\omega)_s}{\sqrt{\omega'\Sigma\omega}} \text{ and } RC_s^S = \omega_s^S \frac{\partial\sigma^C}{\partial\omega_s^S} = \omega_s^S \frac{(\Sigma\omega)_s}{\sqrt{\omega'\Sigma\omega}}.$$

long positions is equal to the risk budget of all short positions: $b^L = b^S = 0.5$. The risk budgets are collected in a $(m+2) \times 1$ vector $\boldsymbol{b} = (b_1^{LS}, b_2^{LS}, \ldots, b_m^{LS}, b^L, b^S)'$. We then solve the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\eta}}{\text{minimize}} \quad & f(\boldsymbol{\eta}, \boldsymbol{b}) \\ \text{subject to} \quad & \sum_{s=1}^{m} \omega_s^L = 1 \\ & 0 \leq \omega_s^L \leq 1, \;\; s = 1, \ldots, m \\ & 0 \leq \kappa \leq 2, \end{aligned}$$

where

$$f(\boldsymbol{\eta}, \boldsymbol{b}) = \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\frac{RC_i^{LS}}{b_i^{LS}} - \frac{RC_j^{LS}}{b_j^{LS}} \right)^2 + \left(\frac{RC^L}{b^L} - \frac{RC^S}{b^S} \right)^2 .$$

The function $f(\boldsymbol{\eta}, \boldsymbol{b})$ is minimized when the risk contributions of all long-short portfolios are equal, and when the risk contributions of the long positions equals that of the short positions. The short multiplier κ is allowed to vary between 0 and 2. In general, if the volatility of the short legs is substantially higher than that of the long legs, we should expect $0 \leq \kappa \leq 1$.

4.3.3. Portfolio calculations

Let N be the total number of stocks in the universe, T be the total number of months, and w_{it}^P denote the weights of a generic portfolio P. We calculate the average monthly turnover of a portfolio P over T months as

$$\text{Turnover}^P = \frac{1}{T} \sum_{t=2}^{T} \sum_{i=1}^{N} |w_{it}^P - w_{i,t-1}^P|. \tag{9}$$

The turnover in each month is used to estimate the transaction costs. Higher turnovers imply higher transaction costs and therefore negatively impact the net results of a given portfolio or strategy. We consider a fixed cost of 15 basis points (bps) for all trades, comprising 10 bps of bid-ask spread and 5 bps of brokerage costs, as well as a fixed annual borrowing cost of 4.5%, applied on the total amount of short positions for long-short portfolios.[14]

We build portfolios with different degrees of leverage, including portfolios whose leverage is time varying. In order to compare these portfolios, we calculate the average leverage ratio of a long-short portfolio as

[14] This corresponds to a monthly borrowing cost of approximately 0.37% for a traditional long-short portfolio.

$$\text{Leverage}^P = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}|w_{it}^P|. \tag{10}$$

Note that for traditional long-short portfolios, the leverage ratio is constant and equal to 2. A 130/30 portfolio has a constant leverage ratio of 1.6.

5. Empirical Results

5.1. Performance of baseline ML portfolios

We start by looking at the performance of the baseline ML portfolios, which are reported in Table 3. All numbers are on a monthly basis, except for Sharpe ratios, which are annualized. For each method, we report average gross and net returns, the standard deviation of returns, gross and net Sharpe ratio, maximum drawdowns, and monthly portfolio turnover.[15] The table is based on returns from the period from January 2006 to December 2018.

Returns increase monotonically across quintile portfolios for all ML methods, and the portfolios formed on the highest quintile (P_5) deliver average monthly returns which are in most cases close to 2%. For comparison, an equally-weighted portfolio of all stocks achieved a monthly return of 1.04%, and the IBOVESPA index, the main index for the Brazilian stock market, delivered 0.80% per month over this period. On the other hand, the returns of all P_1 portfolios are on average positive, and all P_1 portfolios have significantly higher volatility than the corresponding P_5 portfolios (the average volatility of P_1 portfolios is 58% higher than that of P_5 portfolios). This limits the potential profitability of long-short portfolios, especially on a risk-adjusted basis. Additionally, all portfolios have very high maximum drawdowns, typically higher than 50% and much higher for the P_1 portfolios.

The performance of long-short portfolios ($P_5 - P_1$) on a net basis are highly dependent the portfolio turnovers. For example, in terms of gross long-short returns, only five methods outperform OLS (PLS, RF, NN3, NN4 and NN5), but if we consider net monthly average returns, most methods outperform OLS, because the turnover of the long-short portfolio using OLS is among the highest, at 178.54%. However, the volatility of the long-short portfolio using OLS is one of the lowest, at 5.08% per month, which results in OLS outperforming most ML methods on a risk-adjusted basis using the net Sharpe ratio (SR). The net SR of the long-short OLS portfolio is 0.68, which is outperformed only by NN3 (0.84) and NN4 (0.82).

[15] To calculate Sharpe ratios, we use as the risk-free rate the Brazilian interbank certificate of deposit rate, or CDI, which represents the average rate of all interbank overnight transactions in Brazil. When calculating Sharpe ratios, we assume the investor deploys any excess cash in an investment that yields the risk-free rate.

Table 3: Performance of Baseline Machine Learning Portfolios

The table reports out-of-sample performance metrics for equally-weighted quintile portfolios formed on predicted returns using machine learning methods. P_1 (P_5) is the portfolio formed on the lowest (highest) quintile of predicted returns. $P_5 - P_1$ is a hedge portfolio long the stocks in P_5 and short the stocks in P_1. The table reports average monthly return before (Ave) and after costs (Ave (net)), the monthly standard deviation (Std), the annualized Sharpe Ratio before (SR) and after costs (SR (net)), the maximum drawdown (Max. DD), the monthly turnover.

	OLS						LASSO						Ridge					
	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$
Ave	0.28	0.84	1.00	1.42	1.92	1.63	0.32	0.56	0.96	1.66	1.92	1.59	0.30	0.63	0.95	1.69	1.90	1.60
Ave (net)	0.16	0.64	0.79	1.21	1.78	1.37	0.23	0.41	0.80	1.51	1.82	1.40	0.22	0.48	0.79	1.54	1.80	1.41
Std	8.64	6.54	6.08	5.91	6.32	5.08	9.33	6.34	6.09	6.10	6.03	6.23	9.29	6.34	6.19	6.07	6.06	6.20
SR	-0.23	-0.01	0.08	0.33	0.58	1.11	-0.20	-0.17	0.06	0.46	0.61	0.89	-0.21	-0.13	0.05	0.47	0.59	0.89
SR (net)	-0.28	-0.12	-0.04	0.21	0.50	0.68	-0.23	-0.25	-0.03	0.37	0.55	0.58	-0.24	-0.21	-0.04	0.38	0.54	0.58
Max.DD	83.17	62.19	50.85	48.35	50.47	51.59	84.68	67.57	51.68	46.42	52.13	63.96	84.81	63.73	52.40	45.64	52.81	63.33
Turnover	86.36	132.35	140.42	137.29	92.18	178.54	60.15	99.27	106.49	102.28	66.62	126.77	58.52	96.87	103.96	100.87	64.48	123.00

	PLS						PCR						Bayes					
	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$
Ave	0.10	0.82	1.14	1.48	1.92	1.82	0.28	1.18	0.99	1.30	1.71	1.43	0.22	0.74	1.30	1.36	1.84	1.62
Ave (net)	0.03	0.69	1.00	1.34	1.84	1.66	0.20	1.05	0.85	1.17	1.62	1.27	0.10	0.56	1.10	1.18	1.71	1.37
Std	9.81	7.12	6.45	5.79	5.06	7.19	9.75	7.42	6.10	5.94	4.75	7.19	9.00	6.65	6.18	6.15	5.90	5.93
SR	-0.27	-0.02	0.15	0.37	0.72	0.88	-0.21	0.15	0.07	0.26	0.62	0.69	-0.25	-0.06	0.25	0.28	0.57	0.95
SR (net)	-0.30	-0.08	0.07	0.29	0.67	0.62	-0.24	0.09	-0.01	0.18	0.56	0.43	-0.29	-0.16	0.13	0.18	0.50	0.59
Max.DD	87.72	72.89	53.85	47.39	43.08	61.07	88.19	65.44	50.67	50.38	41.20	64.57	83.52	66.23	47.30	49.13	51.97	60.97
Turnover	49.63	87.87	97.21	93.23	53.97	103.60	52.01	88.16	96.46	91.26	54.59	106.60	79.95	124.29	131.72	125.47	86.93	166.88

	Boost						RF						NN1					
	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$
Ave	0.27	0.73	1.12	1.22	1.84	1.58	0.18	0.90	1.16	1.36	1.86	1.68	0.31	0.97	1.28	1.24	1.67	1.36
Ave (net)	0.17	0.58	0.96	1.06	1.73	1.37	0.05	0.70	0.95	1.16	1.72	1.41	0.17	0.77	1.06	1.03	1.53	1.08
Std	9.33	7.13	6.25	5.87	5.51	6.59	9.31	6.23	6.27	6.23	5.85	6.06	7.85	6.56	6.69	6.23	6.45	4.86
SR	-0.22	-0.06	0.14	0.21	0.62	0.83	-0.25	0.02	0.16	0.28	0.59	0.96	-0.25	0.06	0.21	0.21	0.43	0.97
SR (net)	-0.26	-0.14	0.05	0.12	0.55	0.53	-0.30	-0.09	0.05	0.17	0.51	0.60	-0.30	-0.05	0.10	0.10	0.36	0.51
Max.DD	90.36	66.88	58.97	56.27	40.78	65.05	88.96	57.42	55.51	49.36	45.87	55.93	83.56	53.24	53.44	51.34	54.67	36.92
Turnover	62.58	100.72	110.99	103.81	74.77	137.35	85.84	133.06	139.44	135.92	94.90	180.73	90.07	135.91	143.02	136.23	93.56	183.63

	NN2						NN3						NN4					
	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$
Ave	0.18	1.20	1.12	1.27	1.69	1.51	0.02	1.06	1.12	1.31	1.96	1.94	0.11	0.86	0.97	1.42	2.10	1.99
Ave (net)	0.05	1.00	0.91	1.07	1.55	1.24	-0.10	0.87	0.92	1.11	1.83	1.69	0.01	0.69	0.78	1.23	1.98	1.77
Std	8.34	6.74	6.16	6.21	6.36	5.22	8.57	6.79	6.74	5.68	6.09	5.45	9.22	6.94	5.95	5.94	5.77	5.93
SR	-0.28	0.17	0.14	0.23	0.45	1.00	-0.34	0.10	0.13	0.27	0.62	1.23	-0.28	0.00	0.07	0.32	0.74	1.16
SR (net)	-0.34	0.07	0.03	0.11	0.38	0.58	-0.39	0.00	0.03	0.15	0.55	0.84	-0.32	-0.09	-0.05	0.22	0.67	0.82
Max.DD	84.84	52.89	51.60	58.51	50.10	50.13	87.23	54.83	55.34	50.13	49.14	53.11	87.82	66.72	50.22	52.40	47.46	53.46
Turnover	85.40	133.44	139.77	136.79	93.89	179.29	77.42	126.81	134.64	130.58	86.53	163.95	67.37	115.02	126.21	124.87	76.89	144.26

	NN5					
	P_1	P_2	P_3	P_4	P_5	$P_5 - P_1$
Ave	0.20	0.82	1.14	1.39	1.92	1.72
Ave (net)	0.10	0.65	0.96	1.22	1.81	1.52
Std	9.45	6.96	6.18	5.73	5.51	6.29
SR	-0.24	-0.02	0.16	0.32	0.66	0.95
SR (net)	-0.28	-0.10	0.06	0.21	0.60	0.64
Max.DD	87.75	65.70	52.76	53.34	46.82	60.57
Turnover	61.26	109.43	122.37	117.85	70.65	131.91

Our results bear some similarities with those of Gu et al. (2018*b*), which is perhaps the closest paper in terms of the ML methods explored and the type and frequency of the data used. Like them, we also find that neural networks are the best performers among ML methods, with performance peaking at three to four hidden layers. However, in our application, the performance of long-short portfolios using other nonlinear ML methods, such as boosting and random forests, is quite disappointing. It is possible that other choices for parameter optimization for these models might produce better results. However, the fact that only neural networks were able to uncover patterns that produce long-short portfolios that beat linear models suggests non-linearities do play a strong role in this market. In comparison with other studies that use daily stock return data, our results show important differences. For example, Krauss et al. (2017) reports highest Sharpe ratios with random forests, followed by gradient boosting and deep networks. Huck (2019) also finds that random forests outperform other methods such as deep belief networks and the elastic net. Comparison with these studies is challenging, however, due to differences in data and methodology. Particularly, these studies use a classification approach to predict the direction of daily price changes, while we use a regression approach to predict monthly returns.

Overall, our results show that ML is not a panacea for portfolio management, and highlight the importance of the portfolio construction and risk management processes, as well as the context of the market in which it is applied. Additionally, since many ML methods produce portfolios with high turnover, taking into account transaction costs when comparing ML portfolios is essential.

5.2. Performance of different long-short ML portfolios

The results in Table 3 show that traditional long-short portfolios based on ML methods in the Brazilian equity market have some drawbacks. First, shorting stocks does not produce additional returns, since none of the P_1 portfolios have negative returns. Second, the high volatility of the P_1 portfolios increases the volatility of the long-short portfolios. Third, the long-short portfolios have high drawdowns. In this subsection, we explore alternative portfolio construction methods, namely 130/30 portfolios and ERC portfolios based on the P_5 and P_1 portfolios for each method.

Table 4 reports the results. Panel A shows results for the traditional long-short portfolios, i.e. the $P_5 - P_1$ portfolios in Table 3. Panel B reports results for 130/30 portfolios. The average returns of the 130/30 portfolios are higher than those of the traditional long-short portfolios for all methods. But because of differences in average returns and volatilities of the P_1 portfolios, some methods that were not attractive in terms of traditional long-short portfolios produce 130/30 portfolios with much better risk-adjusted returns. For example, the traditional long-short portfolio using PLS has a net SR of only 0.62, worse than OLS, because it suffers from the very high volatility of the P_1 portfolio. The lower short weight in the 130/30 portfolio reduces volatility, while increasing the return from the long leg, producing the highest net SR among all methods (0.95, the same as NN4). The maximum drawdown from this portfolio (39.27%) is also much lower than that of the traditional long-short portfolio (62.91%). This reduction in maximum drawdown, however, is not observed for all 130/30 portfolios; for some methods, maximum drawdown is higher for 130/30 portfolios compared to traditional long-short portfolios.

Finally, in Panel C we consider ERC long-short portfolios for each ML method, as described in subsection 4.3.1. The ERC portfolios have variable leverage, depending on the evolution of the volatilities of the baseline portfolios. The average leverage ratios of these portfolios varies from 1.50 for the PCR method to 1.74 for the Bayes method. The volatilities and maximum drawdowns of the ERC portfolios are greatly reduced, compared to traditional long-short portfolios. The average reduction in volatility is 43%, while maximum drawdowns are reduced by over 70% on average. For most methods, the average returns are higher for ERC portfolios compared with traditional long-short portfolios, since the short legs are under-weighted to balance their risk contributions. In terms of net SR, all ERC portfolios outperform the traditional long-short portfolios, as well as the 130/30 portfolios. Similarly to the traditional long-short portfolios, the best net SR is achieved by the NN3 and NN4 models, whose ERC portfolios both have net SR of 1.35. Figure 2 plots the average (net) monthly return and monthly standard deviation of the different long-short portfolios for each method.

5.3. Performance of ensembles of ML portfolios

The previous results suggest that the performance of different long-short portfolios formed using ML forecasts varies according to the portfolio formation procedure, and the level of turnover and trading costs. Since it is not possible to know in advance which method will perform better, one possibility is to aggregate long-short portfolios obtained with multiple ML methods into one ensemble portfolio. As described before, we consider three ways to create these ensembles: simple equal-weighted averages of the traditional long-short (EW^{LS}) and 130/30 ($EW^{130/30}$) portfolios obtained with each ML method, and an ERC ensemble (ERC^{LS}) that balances the risk of the long and short portfolios for each method, as well as the overall risk contribution of all the long and short positions. To obtain the ERC long-short portfolio, we require estimates of the covariance matrix of the matrix of returns of all P_1 and P_5 portfolios. We use an exponentially-weighted moving average estimator with a decay factor of $\lambda = 0.96$ to estimate this covariance matrix each month, starting with one year of daily returns.[16]

[16] Our results are not dependent on this choice. Results with other values of λ, rolling-window sample covariance estimates, or the shrinkage estimator of Ledoit and Wolf (2004) are qualitatively similar.

Table 4: Long-Short Machine Learning Portfolios

The table reports out-of-sample performance metrics for different long-short portfolios formed on predicted returns using machine learning methods. Panel A shows results for traditional long-short portfolios which are long (short) the highest (lowest) decile of predicted returns. Panel B shows results for 130/30 portfolios, which are long (short) 130% (30%) of the highest (lowest) quintile. Panel B shows results for Equal Risk Contribution (ERC) portfolios, which attempt to equalize the risk contributions of the long and short legs. It is calculated using the average volatility of the stocks in each leg. The table reports average monthly return before (Ave) and after costs (Ave (net)), the monthly standard deviation (Std), the annualized Sharpe Ratio before (SR) and after costs (SR (net)), the maximum drawdown (Max. DD), the monthly turnover, and the average leverage.

Panel A: Traditional long-short strategies

	OLS	LASSO	Ridge	PLS	PCR	Bayes	Boost	RF	NN1	NN2	NN3	NN4	NN5
Ave	1.63	1.59	1.60	1.82	1.43	1.62	1.58	1.68	1.36	1.51	1.94	1.99	1.72
Ave (net)	1.00	1.04	1.04	1.30	0.90	1.00	1.00	1.04	0.72	0.88	1.32	1.41	1.15
Std	5.08	6.23	6.20	7.19	7.19	5.93	6.59	6.06	4.86	5.22	5.45	5.93	6.29
SR	1.11	0.89	0.89	0.88	0.69	0.95	0.83	0.96	0.97	1.00	1.23	1.16	0.95
SR (net)	0.68	0.58	0.58	0.62	0.43	0.59	0.53	0.60	0.51	0.58	0.84	0.82	0.64
Max.DD	53.83	65.67	65.07	62.91	66.25	62.80	67.67	58.29	39.77	52.43	55.28	55.61	62.42
Turnover	178.54	126.77	123.00	103.60	106.60	166.88	137.35	180.73	183.63	179.29	163.95	144.26	131.91
Leverage	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00

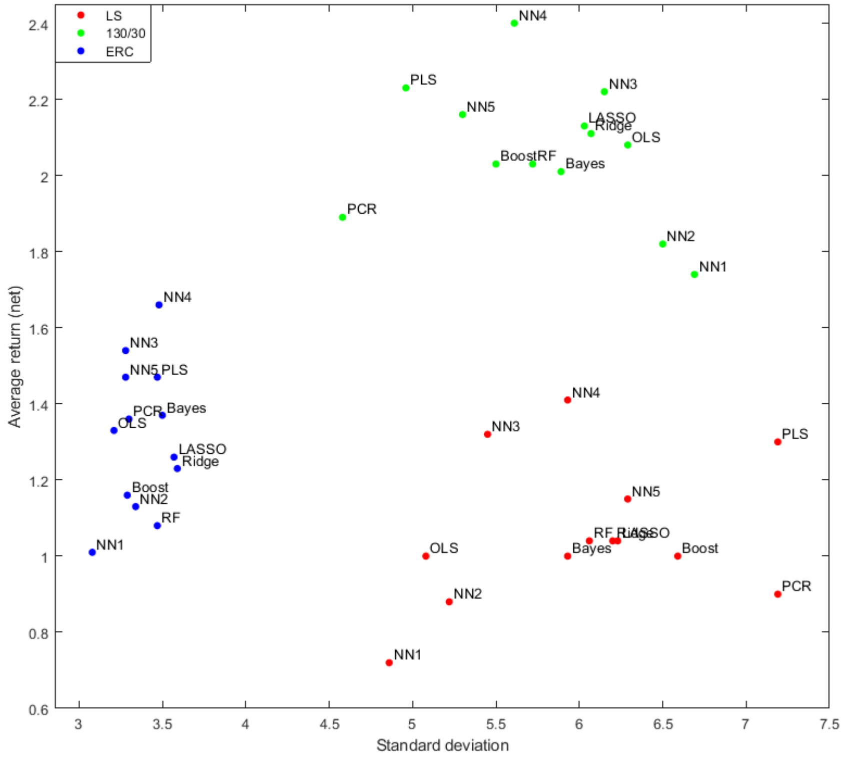
Panel B: 130/30 long-short strategies

	OLS	LASSO	Ridge	PLS	PCR	Bayes	Boost	RF	NN1	NN2	NN3	NN4	NN5
Ave	2.41	2.40	2.38	2.46	2.13	2.32	2.31	2.36	2.07	2.15	2.54	2.70	2.43
Ave (net)	2.08	2.13	2.11	2.23	1.89	2.01	2.03	2.03	1.74	1.82	2.22	2.40	2.16
Std	6.29	6.03	6.07	4.96	4.58	5.89	5.50	5.72	6.69	6.50	6.15	5.61	5.30
SR	0.85	0.88	0.86	1.12	0.96	0.86	0.91	0.91	0.63	0.69	0.94	1.13	1.03
SR (net)	0.67	0.73	0.72	0.95	0.78	0.67	0.74	0.71	0.46	0.51	0.77	0.95	0.85
Max.DD	50.29	52.28	53.15	39.27	36.14	52.76	38.73	44.85	58.69	52.54	50.30	46.66	44.89
Turnover	145.75	104.65	101.38	85.05	86.57	136.99	115.97	149.12	148.65	147.68	135.72	120.16	110.23
Leverage	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60	1.60

Panel C: Equal Risk Contribution long-short strategies

	OLS	LASSO	Ridge	PLS	PCR	Bayes	Boost	RF	NN1	NN2	NN3	NN4	NN5
Ave	1.84	1.68	1.65	1.80	1.68	1.87	1.54	1.55	1.56	1.64	2.00	2.07	1.85
Ave (net)	1.33	1.26	1.23	1.47	1.36	1.37	1.16	1.08	1.01	1.13	1.54	1.66	1.47
Std	3.21	3.57	3.59	3.47	3.30	3.50	3.29	3.47	3.08	3.34	3.28	3.48	3.28
SR	1.74	1.39	1.36	1.43	1.32	1.65	1.26	1.26	1.57	1.46	1.83	1.75	1.60
SR (net)	1.19	0.98	0.95	1.10	0.99	1.16	0.85	0.79	0.95	0.93	1.35	1.35	1.20
Max.DD	15.83	19.02	19.47	12.47	20.29	15.67	21.78	27.83	11.95	14.67	12.78	17.86	13.37
Turnover	162.63	114.26	110.85	87.54	86.12	153.63	117.91	156.91	173.20	162.45	145.04	124.20	111.99
Leverage	1.73	1.70	1.70	1.55	1.50	1.74	1.58	1.65	1.80	1.73	1.67	1.62	1.59

Figure 2: Monthly standard deviation and average return of different long-short portfolios



The results are shown in Table 5. As expected, the equally-weighted ensembles, EW^{LS} and $EW^{130/30}$, underperform the best individual long-short and 130/30 portfolios. Both ensembles achieve a net SR of 0.70. Similarly to the individual long-short portfolios, the equally-weighted ensembles have large maximum drawdowns, of over 60% and 47%, respectively. The ERC^{LS} ensemble, on the other hand, has a net SR of 1.53, which is over twice the Sharpe ratio of the equally weighted ensembles and higher than that of all individual long-short ERC portfolios, while still keeping a maximum drawdown that is a fraction of that of traditional long-short portfolios. The leverage ratio of the ERC^{LS} ensemble is 1.66, implying that the average short weight multiplier (i.e. the average value of κ_t) is 0.66. The evolution of κ_t is shown in Figure 3. Most of the time, κ_t is lower than 1, in order to balance the risk contributions of the long and short legs, due to the higher volatility of the P_1 portfolios. The minimum value is achieved in mid-2011, when the short exposure is less than 40% of the long exposure.

These results strongly support the idea of combining portfolios obtained using various ML methods into one ensemble with balanced risk contributions. The ERC^{LS} ensemble outperforms, on a risk-adjusted basis, all in-

Table 5: Ensemble Machine Learning Portfolios

The table reports performance metrics for three different ensembles of machine learning portfolios. EW^{LS} is a strategy that invests equally in each of the 13 different machine learning long-short portfolios. Similary, $EW^{130/30}$ is a strategy that invests equally in each 130/30 strategy. ERC^{LS} is an Equal Risk Contribution strategy that assigns the same risk contributions to the long and short leg of each machine learning portfolio, while keeping an equal risk contribution across portfolios. The table reports average monthly return before (Ave) and after costs (Ave (net)), the monthly standard deviation (Std), the annualized Sharpe Ratio before (SR) and after costs (SR (net)), the maximum drawdown (Max. DD), the monthly turnover, and the average leverage.

	EW^{LS}	$EW^{130/30}$	ERC^{LS}
Ave	1.76	2.19	2.01
Ave (net)	1.19	1.91	1.59
Std	5.88	5.36	2.99
SR	1.03	0.87	2.01
SR (net)	0.70	0.70	1.53
Max.DD	60.51	47.49	18.46
Turnover	132.94	108.44	117.00
Leverage	2	1.6	1.66

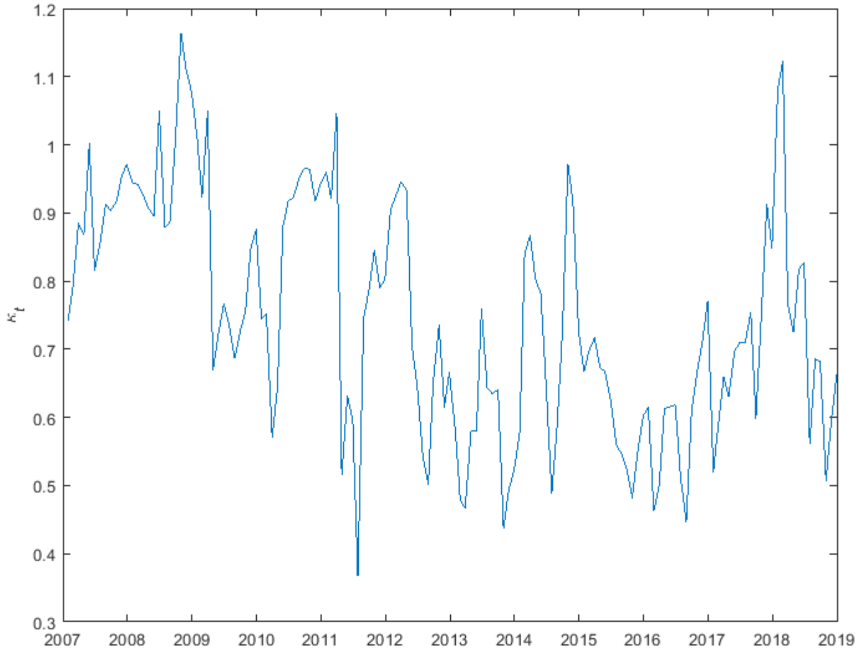
dividual ML portfolios, as well as the equally-weighted ensembles, while keeping the maximum drawdown at a more acceptable level. Other studies that apply ensembles of forecasts obtained with different ML methods, such as Krauss et al. (2017), also find benefits to aggregating different forecasts, however the underlying characteristics of the resulting portfolio, such as the maximum drawdown, do not change as much. Figure 4 plots the cumulative gross returns of the ensembles. In Panel A, we show the cumulative returns of the original ensembles. Despite its much lower volatility, the ERC^{LS} ensemble achieves almost the same total return as the $EW^{130/30}$ ensemble. If we scale all ensembles to have the same volatility as ERC^{LS} (Panel B), the risk-adjusted outperformance of the ERC approach is even clearer.

6. Concluding remarks

In this paper, we have explored the use of machine learning (ML) methods and a rich dataset with many technical and fundamental indicators to forecast stock returns in an emerging market, namely the Brazilian equity market, and have proposed an Equal Risk Contribution (ERC) algorithm to combine portfolios obtained with various ML methods. Our paper contributes to the literature in the intersection of machine learning, operations research, and finance. Specifically, the paper makes the following contributions.

First, we compare the use of many different ML methods to predict individual stocks returns in an emerging market, using a unique database containing many technical and fundamental signals actually used by practitioners. To our knowledge, this is the first such large scale investigation in an emerging market. Second, for each ML

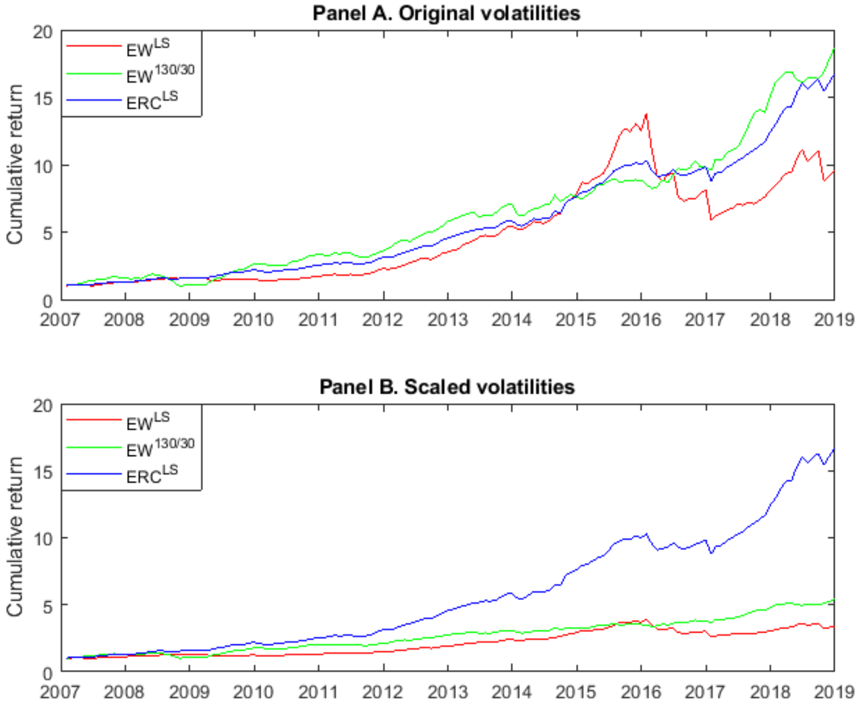
Figure 3: Evolution of ERC short weight multiplier (κ_t)



method, we explore portfolios with different long and short exposures, including traditional long-short portfolios, 130% long/30% short portfolios, and ERC portfolios, that balance the risk contributions of the long and short legs. Third, we investigate equally-weighted ensembles of portfolios obtained with each ML method. Finally, we propose an approach to combine ML portfolios that (i) does not rely on forecasts of which method will outperform in the future, and (ii) balances the risk contributions of each ML portfolio, as well as the risk contributions from the long and short positions.

Our results show that all ML methods methods can produce portfolios that easily outperform local market benchmarks, even after transaction costs, despite suffering from large maximum drawdowns, high volatility and turnover. However, portfolios investing in stocks in the lowest quintile of predicted returns tend to earn low, but positive returns, and have high volatility, making traditional long-short strategies obtained with ML methods relatively unattractive when compared to results reported in developed markets such as the U.S. (Gu et al., 2018*b*; Fischer and Krauss, 2018; Krauss et al., 2017). Overall, neural networks clearly outperform all other ML methods in terms of either net (after cost) average returns or risk-adjusted returns for all types of long and long-short

Figure 4: Cumulative returns of ensembles of machine learning portfolios



portfolios. However, among the other methods, the picture is not so clear, as the extreme volatility of portfolios in the bottom quintile of predicted returns, as well as differences in portfolio turnover, strongly influence the results. These results are in contrast with those reported in the U.S., which typically show a clearer outperformance of ML methods over OLS, especially for nonlinear methods like boosting and random forests.

Our work has important managerial implications related to investment management, especially for portfolio managers seeking to incorporate ML into their investment process. Specifically, the ERC ensemble approach we propose to combine ML portfolios delivers a solution to two problems faced in practice by a portfolio manager applying ML methods. First, it provides a formal way to define allocations to each ML portfolio, without requiring a forecast of which ML portfolio will outperform in the future. In this sense, each ML method can be thought of as a trading desk or strategy, to which the portfolio manager allocates an equal risk budget. Second, the ERC ensemble balances the risk contributions of the overall long and short positions, by allowing a variable degree of leverage in the short positions. We have shown this to be highly relevant and effective in the Brazilian market, due to the much higher risk of the short portfolios. Our empirical results show that the ERC ensemble substantially outperforms, on a risk-adjusted basis, all individual ML strategies, as well as equally-weighted ensembles of ML portfolios, while drastically reducing maximum drawdowns. Although we apply the approach to combinations of portfolios obtained using ML methods, it is general, and be applied to any combination of long and short portfolios.

References

Ang, A. (2014), *Asset management: A systematic approach to factor investing*, Oxford University Press.

Atsalakis, G. S. and Valavanis, K. P. (2009), ‘Surveying stock market forecasting techniques–part ii: Soft computing methods’, *Expert Systems with Applications* **36**(3), 5932–5941.

Ban, G.-Y., El Karoui, N. and Lim, A. E. (2016), ‘Machine learning and portfolio optimization’, *Management Science* **64**(3), 1136–1154.

Bekaert, G., Erb, C. B., Harvey, C. R. and Viskanta, T. E. (1998), ‘Distributional characteristics of emerging market returns and asset allocation’, *Journal of Portfolio Management* **24**(2), 102–+.

Bekaert, G. and Harvey, C. R. (1997), ‘Emerging equity market volatility’, *Journal of Financial economics* **43**(1), 29–77.

Bertrand, P. and Lapointe, V. (2018), ‘Risk-based strategies: the social responsibility of investment universes does matter’, *Annals of Operations Research* **262**(2), 413–429.

Best, M. J. and Grauer, R. R. (1991), ‘On the sensitivity of mean-variance efficient portfolios to changes in asset means: Some analytical and computational results’, *Review of Financial Studies* **4**, 315–342.

Bishop, C. M. (1995), *Neural networks for pattern recognition*, Oxford university press.

Bonaccolto, G. and Paterlini, S. (2019), ‘Developing new portfolio strategies by aggregation’, *Annals of Operations Research* pp. 1–39.

Breiman, L. (1984), *Classification and regression trees*, Wadsworth.

Breiman, L. (1996), ‘Bagging predictors’, *Machine learning* **24**(2), 123–140.

Breiman, L. (2001), ‘Random forests’, *Machine learning* **45**(1), 5–32.

Campbell, J. Y. (2000), ‘Asset Pricing at the Millennium’, *The Journal of Finance* .

Campbell, J. Y. and Thompson, S. B. (2007), ‘Predicting excess stock returns out of sample: Can anything beat the historical average?’, *The Review of Financial Studies* **21**(4), 1509–1531.

Cao, Q., Leggio, K. B. and Schniederjans, M. J. (2005), ‘A comparison between fama and french’s model and artificial neural networks in predicting the chinese stock market’, *Computers & Operations Research* **32**(10), 2499–2512.

Cao, Q., Parry, M. E. and Leggio, K. B. (2011), ‘The three-factor model and artificial neural networks: predicting stock price movement in china’, *Annals of Operations Research* **185**(1), 25–44.

Chen, J., Dai, G. and Zhang, N. (2019), ‘An application of sparse-group lasso regularization to equity portfolio optimization and sector selection’, *Annals of Operations Research* pp. 1–20.

Cybenko, G. (1989), ‘Approximation by superpositions of a sigmoidal function’, *Mathematics of control, signals and systems* **2**(4), 303–314.

De Spiegeleer, J., Madan, D. B., Reyners, S. and Schoutens, W. (2018), ‘Machine learning for quantitative finance: fast derivative pricing, hedging and fitting’, *Quantitative Finance* **18**(10), 1635–1643.

DeMiguel, V., Garlappi, L., Nogales, F. J. and Uppal, R. (2009), ‘A Generalized Approach to Portfolio Optimization: Improving Performance by Constraining Portfolio Norms’, *Management Science* **55**, 798–812.

DeMiguel, V., Martin-Utrera, A., Nogales, F. J. and Uppal, R. (2019), A portfolio perspective on the multitude of firm characteristics.

Fama, E. F. and French, K. R. (1993), ‘Common risk factors in the returns on stocks and bonds’, *Journal of financial economics* **33**(1), 3–56.

Feng, G., Giglio, S. and Xiu, D. (2017), Taming the factor zoo.
URL: *https://ssrn.com/abstract=2934020*

Fischer, T. and Krauss, C. (2018), ‘Deep learning with long short-term memory networks for financial market predictions’, *European Journal of Operational Research* **270**(2), 654–669.

Freund, Y. and Schapire, R. E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of computer and system sciences* **55**(1), 119–139.

Freyberger, J., Neuhierl, A. and Weber, M. (2017), Dissecting characteristics nonparametrically, Technical report, National Bureau of Economic Research.

Friedman, J. H. (2001), 'Greedy function approximation: a gradient boosting machine', *Annals of statistics* pp. 1189–1232.

Friedman, J., Hastie, T. and Tibshirani, R. (2001), *The elements of statistical learning*, Vol. 1, Springer series in statistics New York.

Friedman, J., Hastie, T., Tibshirani, R. et al. (2000), 'Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)', *The annals of statistics* **28**(2), 337–407.

George, E. I. and McCulloch, R. E. (1997), 'Approaches for Bayesian Variable Selection', *Statistica Sinica* **7**, 339–373.

Green, J., Hand, J. R. and Zhang, X. F. (2017), 'The characteristics that provide independent information about average us monthly stock returns', *The Review of Financial Studies* p. hhx019.

Gu, S., Kelly, B. T. and Xiu, D. (2018*a*), Empirical asset pricing via machine learning.
URL: *https://papers.ssrn.com/sol3/papers.cfm?abstract$_i$d* = 3159577

Gu, S., Kelly, B. and Xiu, D. (2018*b*), Empirical asset pricing via machine learning, Technical report, National Bureau of Economic Research.

Harvey, C. R., Liu, Y. and Zhu, H. (2016), '… and the cross-section of expected returns', *The Review of Financial Studies* **29**(1), 5–68.

Haugen, R. A. and Baker, N. L. (1996), 'Commonality in the determinants of expected stock returns', *Journal of Financial Economics* **41**(3), 401–439.

Heaton, J., Polson, N. and Witte, J. H. (2017), 'Deep learning for finance: deep portfolios', *Applied Stochastic Models in Business and Industry* **33**(1), 3–12.

Hsu, M.-W., Lessmann, S., Sung, M.-C., Ma, T. and Johnson, J. E. (2016), 'Bridging the divide in financial market forecasting: machine learners vs. financial economists', *Expert Systems with Applications* **61**, 215–234.

Huck, N. (2009), 'Pairs selection and outranking: An application to the s&p 100 index', *European Journal of Operational Research* **196**(2), 819–825.

Huck, N. (2010), 'Pairs trading and outranking: The multi-step-ahead forecasting case', *European Journal of Operational Research* **207**(3), 1702–1716.

Huck, N. (2019), 'Large data sets and machine learning: Applications to statistical arbitrage', *European Journal of Operational Research* **278**(1), 330–342.

Hwang, S. and Satchell, S. E. (1999*a*), 'Modelling Emerging Market Risk Premia Using Higher Moments', *Politics* **296**, 271 –296.

Hwang, S. and Satchell, S. E. (1999*b*), 'Modelling emerging market risk premia using higher moments', *International Journal of Finance & Economics* **4**(4), 271–296.

Jacobs, H. (2015), 'What explains the dynamics of 100 anomalies?', *Journal of Banking & Finance* **57**, 65–85.

Johnson, G., Ericson, S. and Srimurthy, V. (2007), 'An empirical analysis of 130/30 strategies: Domestic and international 130/30 strategies add value over long-only strategies', *The journal of alternative investments* **10**(2), 31.

Kaucic, M. (2010), 'Investment using evolutionary learning methods and technical rules', *European Journal of Operational Research* **207**(3), 1717–1727.

Kelly, B. T., Pruitt, S. and Su, Y. (2019), 'Characteristics are covariances: A unified model of risk and return', *Journal of Financial Economics* .

Kim, J. H., Kim, W. C. and Fabozzi, F. J. (2018), 'Recent advancements in robust optimization for investment management', *Annals of Operations Research* **266**(1-2), 183–198.

Kolm, P. N. and Ritter, G. (2019), ‘Modern perspectives on reinforcement learning in finance’, *The Journal of Machine Learning in Finance* **1**(1).

Kozak, S., Nagel, S. and Santosh, S. (2019), ‘Shrinking the cross-section’, *Journal of Financial Economics* .

Krauss, C., Do, X. A. and Huck, N. (2017), ‘Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500’, *European Journal of Operational Research* **259**(2), 689–702.

Kyriakou, I., Mousavi, P., Nielsen, J. P. and Scholz, M. (2019), ‘Forecasting benchmarks of long-term stock returns via machine learning’, *Annals of Operations Research* .

Ledoit, O. and Wolf, M. (2004), ‘Honey, I Shrunk the Sample Covariance Matrix’, *Journal of Portfolio Management* **30**, 110–119.

Lettau, M. and Pelger, M. (2018), Factors that fit the time series and cross-section of stock returns, Technical report, National Bureau of Economic Research.

Lewellen, J. (2014), ‘The cross section of expected stock returns’, *Forthcoming in Critical Finance Review* .

Lo, A. W. and Patel, P. N. (2008), ‘130/30: The new long-only’, *Institutional Investor* **42**(5), 186.

Lu, Z., Pu, H., Wang, F., Hu, Z. and Wang, L. (2017), The expressive power of neural networks: A view from the width, *in* ‘Advances in Neural Information Processing Systems’, pp. 6231–6239.

Maillard, S., Roncalli, T. and Teiletche, J. (2010), ‘The properties of equally weighted risk contribution portfolios’, *The Journal of Portfolio Management* **36**(4), 60–70.

McLean, R. D. and Pontiff, J. (2016), ‘Does academic research destroy stock return predictability?’, *The Journal of Finance* **71**(1), 5–32.

Michaud, R. O. . (1989), ‘The Markowitz optimization enigma: Is optimized optimal’, *Financial Analysts Journal* **45**, 31–42.

Mohanram, P. S. (2005), ‘Separating winners from losers among lowbook-to-market stocks using financial statement analysis’, *Review of accounting studies* **10**(2), 133–170.

Mullainathan, S. and Spiess, J. (2017), ‘Machine learning: an applied econometric approach’, *Journal of Economic Perspectives* **31**(2), 87–106.

O’Hara, R. B. and Sillanpää, M. J. (2009), ‘A Review of Bayesian Variable Selection Methods: What, How and Which’, *Bayesian Analysis* **4**(1), 85?118.

Piotroski, J. D. (2000), ‘Value investing: The use of historical financial statement information to separate winners from losers’, *Journal of Accounting Research* pp. 1–41.

Qian, E. (2005), ‘Risk parity portfolios: Efficient frontiers through true diversification’, *PanAgora Asset Management White Paper* .

Quinlan, J. R. (1993), *C4. 5: Programs for Machine Learning*, Morgan Kaufmann.

Raposo, R. and Cruz, A. D. O. (2002), Stock market prediction based on fundamentalist analysis with fuzzy-neural networks, *in* ‘Proceedings of 3rd WSES International Conference on Fuzzy Sets’.

Ripley, B. D. (1996), *Pattern recognition and neural networks*, Cambridge university press.

Roncalli, T. (2016), *Introduction to risk parity and budgeting*, Chapman and Hall/CRC.

Schapire, R. E. (1990), ‘The strength of weak learnability’, *Machine learning* **5**(2), 197–227.

Sermpinis, G., Theofilatos, K., Karathanasopoulos, A., Georgopoulos, E. F. and Dunis, C. (2013), ‘Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization’, *European Journal of Operational Research* **225**(3), 528–540.

Smith, M. and Kohn, R. (1996), ‘Nonparametric regression using bayesian variable selection’, *Journal of Econometrics* **75**(2), 317–343.

Varian, H. R. (2014), ‘Big data: New tricks for econometrics’, *Journal of Economic Perspectives* **28**(2), 3–28.

Appendix A. Equal Risk Contributions for a long-short portfolio

Consider two assets 1 and 2 whose covariance matrix is given by

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix},$$

and a long-short portfolio investing $w_1 > 0$ in asset 1, and $w_2 = -kw_1$ in asset 2. Let $w = (w_1 \quad w_2)'$. The volatility of the long-short portfolio is $\sigma_{LS} = \sqrt{w'\Sigma w}$. The marginal risk contributions are defined by the derivative of σ_{LS} with respect to w:

$$\frac{\partial \sigma_{LS}}{\partial w} = \frac{\Sigma w}{\sqrt{w'\Sigma w}} = \frac{1}{\sigma_{LS}} \begin{pmatrix} w_1^2\sigma_1^2 - \kappa w_1^2\sigma_{12} \\ -\kappa w_1^2\sigma_{12} + \kappa^2 w_1^2 \sigma_2^2 \end{pmatrix} \tag{A.1}$$

The risk contributions of the long and short positions are calculated as the product of each weight and the corresponding element of (A.1). Therefore, requiring equal risk contributions amounts to:

$$\begin{aligned} \frac{1}{\sigma_{LS}}(w_1^2\sigma_1^2 - \kappa w_1^2\sigma_{12}) &= \frac{1}{\sigma_{LS}}(-\kappa w_1^2\sigma_{12} + \kappa^2 w_1^2\sigma_2^2) \\ &\Rightarrow \kappa = \frac{\sigma_1}{\sigma_2}. \end{aligned}$$

Thus, for any arbitrary $w_1 > 0$, risk parity between the long and short legs is achieved when the weight of the short position is equal to the ratio of the volatilities, multiplied by the weight of the long position. This result is independent of the correlation between the assets.