Machine Learning - The Future of Equity Premium Prediction

A Comparative Study of Machine Learning Methods for Predicting the Equity Premium

Abstract

Predictions of the equity premium have historically been made by using traditional predictive regressions. Despite the great promise of machine learning applications for prediction tasks, it has largely been overlooked in the financial literature. We predict the monthly equity premium, defined as the monthly excess return on the S&P 500, using three linear and five non-linear machine learning models. The models are evaluated against a benchmark consisting of the historical average return. Six of our models outperform the benchmark, with the three most successful being non-linear models. We perform a statistical evaluation of the machine learning forecasts, finding that three of the outperforming models are significantly different from the benchmark. Additionally, the models are evaluated economically by calculating an implied Sharpe ratio using the predictive results, showing meaningful economic gains even for models with only a slight increase in predictability. Successively, we translate our forecasts to their inherent directional prediction, finding that four models beat the benchmark. By formulating a naïve investment strategy, we show that also the directional predictability can be exploited to generate a higher Sharpe ratio than that of the market.

ALFRED HEDLUND

IBRAHIM METE BACAK

Master Thesis in Finance

Stockholm School of Economics

2020

Keywords
Machine learning

Equity premium

Prediction

Penalized linear models

Non-linear models

Ridge regression

Lasso regression

Elastic net regression

Regression trees

Random forests

Light gradient boosting machines

K-nearest neighbors

Artificial neural networks

Authors
Alfred Hedlund

Ibrahim Mete Bacak

Tutor
Tobias Sichert

Examiner
Jungsuk Han

Acknowledgements
We would like to thank our tutor Tobias Sichert for providing valuable input and guidance during the process of writing this thesis.

Table of Contents

1. INTRODUCTION

Trying to predict the direction and magnitude of movements in equity markets has a long-standing history in the finance literature. Some of the first known publications date back to the early 20th century and were a series of articles by C.H. Dow, later published in book format under the title *Scientific Stock Speculation* (1920). Despite being extensively researched, most attempts at predicting stock market movements have had problems to outperform the use of historical averages as a prediction, leaving the time-series variation in excess return largely unexplained. For instance, Welch & Goyal (2008) argue that previously suggested models for predicting the equity premium[1] do not seem robust. Until recently, the literature has mainly been focused on finding traditional linear models which could explain equity returns. Out-of-sample predictability is debated but found to improve with, for instance, time-varying coefficients, as opposed to constant coefficients (Dangl and Halling, 2012). Lately, a new sub-field trying to predict movements with both advanced linear and non-linear methods (i.e. machine-learning applications) has been growing rapidly. The concept of machine learning is not new itself; it was pioneered in its simplest form already in the 1950s by, for instance, Alan Turing (1950). Yet, it has taken more than half a century for these applications to gain traction in the literature related to predicting the equity premium. Generally, the literature that does exist has been focusing either on predicting the direction of index movement or predicting the actual return. While both mechanisms are meaningful from an economic standpoint, this paper will focus mainly on trying to predict the magnitude of excess returns. However, we do translate these predictions to their inherent directional predictions to get an understanding of the predictive performance in this aspect too. While there is not a myriad of existing literature on the subject, some studies suggest increased predictability when using machine learning. For instance, Feng et al. (2018) show that applying deep learning algorithms to a set of eight predictors can improve forecasts enough to outperform a benchmark consisting of the historical average return. In this paper, we evaluate the predictive power of 8 models using 48 predictors between January 2000 to December 2019. The number of predictors exceeds that of most equity premium prediction literature, but the models are selected partly on their ability to handle a vast predictor set.

[1] Throughout this paper, we use the terms "equity premium" and "excess return" interchangeably when we refer to the return in excess of the risk-free rate.

We are able to conclude that the light gradient boosting machine is the most accurate model in the context of this study. The model produces a forecast significantly different from the historical average and shows a meaningful improvement in the Sharpe ratio. Additionally, we show that various measures of change in the moving average price level have an important role in prediction for many of the applied models.

The remainder of this paper is structured as follows. Section 2 surveys the existing literature on equity premium prediction and machine learning applications in finance. Section 3 describes the data. Section 4 covers the methodology of our study and provides a brief introduction to the machine learning models used. Section 5 shows our results and provides some comparisons to the existing literature. Finally, Section 6 contains a short discussion based on the results, as well as a few suggestions for future research within the field.

2. PREVIOUS PREDICTION LITERATURE

Campbell (2008) states that the excess return, or equity premium, was often considered as a constant in the 1960s and 1970s. He explains that this consensus stemmed from the prevailing interpretation of the efficient market hypothesis, and the historical average excess return was considered to be the closest estimate of the actual premium. As years went by, the financial literature suggested that the use of predictive regressions, i.e. regressing lagged variables[2] on the excess market returns, could predict the equity premium. Among the most popular predictors were valuation ratios, like the dividend yield and earnings-price ratio, which were shown to predict returns on a long-run basis (see, for example, Rozeff, 1984; Fama and French, 1988). The predictability of the equity premium faced criticism based on spurious correlation bias in combination with data mining issues (Ferson et al., 2003). However, predictability was later defended by Cochrane (2008), who states that the observed variation in dividend yield requires that either returns or dividend growth must be predictable. Hence, the author argues that the absence of dividend growth predictability should be seen as strong evidence of return predictability. The performance of a number of predictive regressions was re-evaluated on equal terms by Welch & Goyal (2008), who found that out-of-sample performance had been poor for over thirty years and that most investors would have been better off using the historical

[2] Throughout this paper, we use the terms "predictor" and "variable" interchangeably when referring to variables used for predicting the return.

average equity premium as a forecast. Overall, the topic has been discussed widely and predictive regressions have divided the research community for a long time. While using time-varying coefficients (Dangl and Halling, 2012), and including effects of time-varying volatility and estimation risk (Johannes et al., 2014), seem to improve the performance of predictive regressions, some researchers are abandoning traditional regression methods in favor of more complex machine learning applications in the pursuit of higher prediction accuracy.

Our review of the machine learning literature within finance shows that the studies are fewer and to a greater extent published in data science journals, rather than recognized financial journals. Despite this, machine learning should have great potential when it comes to predicting excess equity returns, or equivalently measuring the equity premium, as prediction is a task where machine learning algorithms are particularly well suited (Gu et al., 2020). As Gu et al. point out, the use of predictive regressions carries some problems which could possibly be severe, most importantly, it is problematic that these regressions are generally not well equipped to handle the numerous predictor variables that the literature has compiled over many decades. While machine learning models are typically designed to work well with a large predictor set and often improve predictions, Gu et al. (2020) stress that it is important to understand that the predictions themselves do not discover economic mechanisms or equilibria. For machine learning to be useful for such purposes, it requires that the individual intentionally evaluates certain pre-specified structures that can be applied via the models (Gu et al., 2020).

Machine learning applications for prediction have been used and published more frequently in two sub-fields of financial literature, namely, prediction of the magnitude of returns and predicting the directional movement of the market. Directional movements refer to whether the return will be positive or negative over the coming period. In this paper, our main focus is to apply and compare different machine learning techniques in their ability to predict the magnitude of realized returns. However, this implies that we inherently also predict the directional movement for each month, allowing a comparison of the tested algorithms in this aspect as well. This renders both sub-fields relevant for us, although predicting the magnitude of the equity premium is the focal point of this study.

Fischer and Krauss (2018) were able to exploit predictability to get a Sharpe ratio of 5.8 before transaction costs, which is high compared to the market's 0.34 over the same period. The predictions were carried out using a specific deep learning model called long short-term memory networks, using daily return data from the S&P 500 between 1992 and 2015. However, the authors note that the ability to make excess returns from directional predictability dropped

sharply from 2010 onwards. Kara et al. (2011) were able to show an average accuracy of above 75% when predicting the direction of return on the Istanbul Stock Exchange index, using artificial neural networks. While this is not translated into excess returns by the authors, they note that they outperform certain previous papers using a similar methodology but that their model has its lowest accuracy during the financial market turmoil of 2001.

Gu et al. (2020) predict the magnitude of asset risk premiums using a range of machine learning methods and find that there are large economic gains for an investor when using machine learning forecasts. The authors also discover that the best performing prediction models are random forests and neural networks and trace the most successful predictors to variations in momentum, liquidity, and volatility. An investor using the authors' neural network model to time the S&P 500 has an annualized Sharpe ratio of 0.77, compared to 0.51 of a buy-and-hold investor, showing clear economic gains even for simple index investments. Feng et al. (2018) apply deep learning and vary the number of hidden layers in an artificial neural network to predict asset returns, using the same data as Welch and Goyal (2008). They claim to find nonlinear factors explaining asset returns, and these factors are shown to have the most impact at the extremes of the characteristic space. The authors apply both a rolling window and an expanding window to estimate the prediction models, with the highest accuracy achieved with the latter. Their belief is that the models more easily identify nonlinear structures when provided with more training data, hence the approach of an expanding window is more suitable. Feng et al. evaluate their models based on the out-of-sample R^2, just like Welch and Goyal (2008) did for predictive regressions. Feng et al. find positive out-of-sample R^2 for several models, meaning that they are able to outperform the benchmark prediction consisting of the unconditional mean return.

In summary, the debate surrounding the use of traditional predictive regressions for equity premium prediction seems to continue in parallel with researchers' increasing curiosity regarding the prospects of machine learning for this purpose. So far, machine learning has shown promising signs for predicting both directional movements and the magnitude of excess returns. The predictability is often shown to generate returns in excess of the market, leading to an increase in the Sharpe ratio, which for a mean-variance investor implies utility gains. However, some studies, like Fischer and Krauss (2018), argue that the ability to make abnormal returns with their trading strategy almost vanished after 2010. Others, like Kara et al. (2011), exhibit substantially worse performance in times of turmoil in financial markets. Consequently,

many questions remain unanswered concerning the use of machine learning applications for making predictions in financial markets.

3. DATA

The monthly total return on the S&P 500 was obtained from the Center for Research in Security Prices ("CRSP") for the sample period, beginning in January 2000 and ending in December 2019. The reason for not going back further in time is the lack of historical data available for some of our predictor variables. While an increased time horizon is generally beneficial for machine learning applications, we decided not to drop the predictors lacking historical data before 2000, in favor of evaluating the models using this extended set of predictors. The 48 predictor variables can be divided into three subcategories: successful variables from previous equity premium prediction literature, successful variables from machine learning literature on returns or directional movements, and variables affecting the constituents of S&P 500.

The process of selecting the aforementioned predictors was based on several criteria. Firstly, variables proven to predict returns in equity premium literature were considered. For instance, the dividend yield as shown by Rozeff (1984), albeit with stronger prediction power for longer time horizons than the monthly predictions in our study (Fama and French, 1988). These variables are generally collected from researcher websites or Thomson Reuters Eikon. Secondly, we included variables from machine learning literature on return prediction. Kara et al. (2011) show that moving averages can be important in explaining directional movements, which was the precedent for including, for instance, changes in moving averages and exponential moving averages in our paper. These variables mostly consist of technical indicators, which can be derived from the S&P 500 total return index itself. Finally, we wanted to test a number of variables impacting the constituents of the S&P 500. While these do not have the same type of history in the literature, they could still prove to have an effect on the excess return. These variables are predominately made up of interest rates, macro-economic indicators, and currency exchange rates, and can generally be collected from the Federal Reserve Economic Data (FRED). Please refer to Appendix 1 for a full list of variables, explanation of variables, and sources of data.

Throughout the predictor selection process, we exclusively considered variables for which data was available on a monthly basis to match our prediction frequency, without having

missing data for some time periods. It is important to note that a substantial number of predictors were collected in the form of levels or nominal values. However, to match our dependent variable and avoid any bias toward higher values in an upward trending predictor, all such variables have been converted to monthly changes in percent. The main goal of this study is to compare a set of machine learning models and find the best one for predicting the excess return on the S&P 500. Applying traditional regression methods would limit the possibility to include many predictors successfully. However, the models studied in this paper were selected partly based on their ability to handle vast predictor sets, without encountering major problems even when faced with high pairwise correlation or multicollinearity among the predictors. The model selection allows us to include more variables without manually having to adjust the model setup, but we do understand that the ability to effectively manage the vast predictor set varies between the selected models. The model selection and their respective properties are discussed further in Section 4.

4. METHODOLOGY

This section describes the overarching framework applied for each of the models, as well as the specific models themselves. The presentation and description of models aim to give the reader an overview of how each model is applied and what makes that model different from the others. Essentially, we provide only a high-level introduction to the models and their history, as opposed to describing the statistical and computational mechanisms in detail. This approach allows a greater focus on the core contribution of this paper, which is to evaluate and compare the prospects of a set of machine learning techniques for predicting the excess equity return.

4.1 Overarching Framework

We focus on predicting the monthly equity premium, in this paper defined as the monthly simple excess return on the S&P 500. We use Python to apply machine learning models using an expanding window of observations to estimate model parameters. To find the excess return we deduct the equivalent of one month's return on the risk-free rate. The yield on the 3-month treasury bill is used as a proxy for the risk-free rate. On a high level, we assume that the excess return on the S&P 500 can be thought of as an additive model:

$$r_t = E_{t-1}(r_t) + \epsilon_t \,, \tag{1}$$

where ϵ_t is the error term and expected excess return, $E_{t-1}(r_t)$, is represented by:

$$E_{t-1}(r_t) = g^*(z_{t-1}) \,. \tag{2}$$

In this paper, we try to predict r_t by testing different models to approximate the function $g^*()$ in Equation 2. The predictions are driven by a set of predictor variables, denoted in vectorized form as z_{t-1}. In general, we assume no specific functional form of $g^*(z_{t-1})$. Instead, we apply a wide range of models with a combined ability to detect both linear and non-linear relationships with the goal of minimizing the sum of squared prediction errors and, hence, maximizing the out-of-sample prediction power for the realized return on the S&P 500, r_t. While the framework is highly flexible, it does impose the important restriction of using only predictor values available at *t*-1 for predicting the return in month *t*.

Deciding on the split between data used for estimation and out-of-sample testing is a dubious task for traditional regressions. This task is no easier for machine learning models, where data has to be used for three purposes: estimation of parameters, optimization of hyperparameters, and testing. This ambiguity is discussed by Welch and Goyal (2008), who stress that it is important that the first estimation is done over an adequate number of observations to get a reliable estimate of model parameters for the first prediction. While their statement concerns predictive regressions, we apply the same reasoning for the machine learning models applied in this study. Thus, we use one-fourth of the data collected, corresponding to 60 months, to estimate the algorithms' parameters *and* hyperparameters before predicting the excess return in month 61. Next, we successively add one month of data and re-estimate the models' parameters before predicting the return for the next month, implying that each prediction is truly out-of-sample. This method allows us to utilize the collected data more efficiently, both for model estimation and true out-of-sample prediction, as opposed to if we would have split the data into separate parts for estimation, cross-validation, and testing, respectively. Instead, the cross-validation and simultaneous tuning of model hyperparameters are done solely based on data from the first 60 months.

Hyperparameters are model inputs used to control the learning process of a machine learning algorithm. By definition, these are not decided when estimating the model but rather

set by the researchers themselves. Parameters[3], on the other hand, are estimated automatically by fitting a model to the data points. As mentioned, the optimal values for the hyperparameters are sought after by applying *K*-fold cross-validation. Our implementation of the cross-validation procedure splits the data used for cross-validation into two random sets, 80% for training with certain hyperparameters, and 20% held out for testing. This process is repeated *K* times, after which the scores from each test are evaluated automatically. Cross-validation can be used for multiple purposes, but in our study, it is used solely to optimize hyperparameters with the objective of balancing the bias-variance tradeoff[4]. In essence, our goal is to introduce a reasonable amount of bias into the models while trying to limit the variance of out-of-sample predictions. Depending on the number of folds, cross-validation can be computationally intense even with today's modern computers, hence, the extent to which these parameters can be optimized is somewhat limited. We use 64-fold cross-validation for the penalized linear models and eight-fold for the non-linear models, based on data from the first 60 months. Another approach would be to re-estimate the hyperparameters using a rolling cross-validation window, moving simultaneously with the expanding window of observations used for parameter observation. This method has the benefit of maintaining the temporal ordering of the data. However, it is computationally complicated and reduces the interpretability of models even further. Additionally, our time-horizon is limited compared to other equity premium prediction literature, and the *K*-fold cross-validation allows us to more effectively use the data held out from making predictions.

4.1.1 Prediction Performance Evaluation

We evaluate the models based on out-of-sample R^2, just like Campbell and Thompson (2008) did for predictive regressions. The same evaluation approach is used by both Feng et al. (2018) and, with minor adjustments, by Gu et al. (2020) for machine learning prediction models. The out-of-sample R^2 is defined as:

[3] For example, the parameters in an OLS regression are the coefficients. However, an OLS regression has no hyperparameters, as it is unbiasedly fitted to the data points.

[4] The bias-variance tradeoff relates to the total prediction error of the model across samples. The variance stems from too high model complexity, also called overfitting. Bias, on the other hand, is an error that arises from erroneous assumptions in fitting the model, which results in underfitting. The tradeoff itself is related to minimizing the total error stemming from these two individual sources of errors. For a more extensive explanation, see for instance Section 2.9 in Hastie et al. (2017).

$$R_{OS}^2 = 1 - \frac{\sum_{t=1}^{T}(r_t - \hat{r}_t)^2}{\sum_{t=1}^{T}(r_t - \overline{r}_t)^2} \ , \tag{3}$$

where r_t is the one-month realized excess return on the S&P 500 at time *t*, $\hat{r}_t$ is the predicted one-month excess return on the S&P 500 based on information available up until time *t*-1, and $\overline{r}_t$ is the historical average monthly return up until time *t*-1. By definition, if $R_{OS}^2 > 0$ the predictive model is producing a lower mean squared prediction error than the historical average return, meaning that it has higher accuracy in its predictions. Hence, this metric inherently compares the models to a no-predictability benchmark[5] which, as discussed in Dangl and Halling (2012), is equivalent to an unconditional model neglecting the predictive power of all our collected predictive variables. Important, however, is that even though this is a no-predictability benchmark, it is still updated successively as also the benchmark is estimated over an expanding window of observations.

We also assess the applied models based on the root mean squared prediction error ("RMSE"), a measure which is computed by taking the average of the square root of the sum of squared prediction errors ("SSE"):

$$SSE = \sum_{t=1}^{T}(r_t - \hat{r}_t)^2 \tag{4.1}$$

$$RMSE = \frac{\sqrt{SSE}}{T} \ . \tag{4.2}$$

To get an understanding of how the models perform in different time periods, we calculate a twelve-month rolling RMSE. Our belief is that models should learn by having access to an increasing amount of data for estimating parameters. In that case, we would expect to see a decrease in the rolling RMSE over time.

We conduct a statistical evaluation of the predictions from all of the machine learning models using the Diebold and Mariano (1995) test for differences in out-of-sample forecasts. The forecast produced by each machine learning model is compared to the forecast based on the unconditional mean return. The test is also adjusted according to the suggestion by Harvey

[5] Throughout this paper, we use the terms "historical average", "unconditional mean", and "benchmark" interchangeably when referring to our benchmark return consisting of the average historical return on the S&P 500. This benchmark is updated on a monthly basis, as more data is in-sample for parameter estimation.

et al. (1997). Their corrections have the most impact when testing a small sample of predictions, yet it has been shown to generate better results than the original test in larger samples as well (Mariano, 2004). The modified test statistic (*DM**) is defined as:

$$DM^* = \sqrt{\frac{T + 1 - 2h + T + h(h-1)}{T}}\, DM \sim T_{dist}(T-1)\ , \tag{5}$$

where *T* is the number of predictions, *h* is steps forecasted (in our case equal to 1), T_{dist} is the T distribution, and *DM* is the original Diebold and Mariano test statistic defined as $DM = \bar{d}/\hat{\sigma}_{\bar{d}}$. In the DM test statistic, $\bar{d}$ is the average of the squared error differential between the two forecasts, and $\hat{\sigma}_{\bar{d}}$ is a consistent estimate of the standard deviation of $\bar{d}$.

$$\bar{d} = \frac{1}{T}\sum_{t=1}^{T}(\hat{e}_{t,i}^2 - \hat{e}_{t,avg}^2) \tag{6}$$

In our case, the squared error differential will be between one of the machine learning models and that of the historical average, in Equation 6 indexed as *i* and *avg*, respectively.

We also undertake an economic evaluation of the prediction results by calculating the implied Sharpe ratio of improved predictability over the benchmark. Campbell and Thompson (2008) showed that the Sharpe ratio obtained by an active investor making use of predictive information, summarized as $R^2_{OS,}$ can be adjusted using the Sharpe ratio of a passive investor ("*SR*"). This adjusted Sharpe ratio ("*SR**") was later adopted in the machine learning literature by Gu et al. (2020).

$$SR^* = \sqrt{\frac{SR^2 + R_{OS}^2}{1 - R_{OS}^2}}\ . \tag{7}$$

The adjustment of the Sharpe ratio shown in Equation 7, is one way of showing that even small improvements in predictability can be of economic importance, as pointed out by Campbell and Thompson (2008). Furthermore, we calculate the implied Sharpe ratio improvement of utilizing predictive information as $SR^* - SR$.

4.1.2 Directional Prediction

All models applied to predict the return will also inherently predict whether the return on the S&P 500 will be positive or negative over the following month. We are aware that the existing literature on directional movements largely focuses on daily changes (see, for example, Kara et al., 2011; Patel et al., 2015) rather than monthly, and that our models will be tuned to predict the magnitude of returns and not direction. Yet, evaluating to what extent they are right in their directional prediction is interesting as it serves as an easily interpretable indication of the models' prediction accuracy. The models will be assessed based on the share of positive and negative directional predictions that are true and false, respectively. Successively, we use these directional predictions to construct a naïve investment strategy in which we are 100% invested in the market whenever the equity premium is predicted to be greater than 0, and stepping out entirely whenever the premium is expected to be less than or equal to 0. While the focus of this study is to construct and evaluate prediction models, not forming novel investment strategies, this exercise allows us to get an initial understanding of whether the models could be applied in practice to exploit any predictability found. To measure the performance of the naïve investment strategy, we calculate the annualized Sharpe ratio ("SR"), as discussed by Sharpe (1994):

$$SR_{Annulized} = \frac{T\overline{r_i}}{\sqrt{T}\,\sigma_{r_i}} = \sqrt{T}\frac{\overline{r_i}}{\sigma_{r_i}} \quad , \tag{8}$$

where T is the frequency over which the returns are measured, in our study equal to 12, $\overline{r_i}$ is the average monthly excess returns of the strategy based on algorithm i, and σ_{r_i} is the monthly standard deviation of the excess return. While Sharpe points out that this annualization method is not ideal, as it builds on some dubious assumptions, he still stresses that it could provide at least a somewhat meaningful comparison between different strategies. It also provides us with Sharpe ratios in the same magnitude as other studies, although any comparisons will still be questionable and treated with caution as the time horizon and method are likely to be different. The performance of the naïve investment strategy will be compared to a benchmark consisting of a buy-and-hold investor in the S&P 500 index.

4.2 Machine Learning Models

Machine learning is often defined vaguely, and different authors refer to different techniques when using the term. When the term is used in this paper, we refer to models that are more advanced in their underlying statistical framework compared to a traditional ordinary least square ("OLS") regression. In the following subsections, we will provide a high-level overview of the models used in this paper. Besides serving as an introduction for anyone with limited experience with these types of models, it also further clarifies what is defined as machine learning in the context of this paper. The model selection is based on what has successfully been used in previous literature (see, for example, Gu et al., 2020; Feng et al., 2018). Additionally, the models should be capable of effectively making use of vast predictor sets, or having the ability to efficiently select which variables to use as predictors.

4.2.1 Penalized Linear Models: Ridge, Lasso, and Elastic Net

An OLS regression becomes inefficient for prediction when the number of predictors increases, particularly when the number of predictors is approaching the number of observations. Gu et al. (2020) go as far as saying that simple linear models are bound to fail when many predictors are included in the model, with the clarification that an OLS regression begins to overfit the noise rather than find the true patterns in the data. When approaching financial prediction tasks similar to those in this paper, overfitting is particularly problematic as the signal-to-noise ratio[6] often tends to be quite low (Gu et al., 2020). A commonly applied technique for tackling this issue is adding a regularization term, commonly also referred to as a penalty term, to the original loss function of the OLS regression. By applying this penalty, the in-sample fit is systematically deteriorated in an attempt to increase out-of-sample performance. The concept is motivated by the notion that an improvement will occur if the penalty reduces the extent to which the model is fitted to noise versus actual signals. Adding the penalty term to the original OLS regression loss function $\mathcal{L}(\beta)$ generates the following general loss function for penalized linear models:

$$\mathcal{L}(\beta; \cdot) = \mathcal{L}(\beta) + \phi(\beta; \cdot), \tag{9}$$

[6] The signal-to-noise ratio is the amount of true signal value compared to the noise in the data. A low ratio indicates that the data is noisy compared to how much true signal value exists, and can be a problem as only signals can be modeled and predicted.

where the traditional OLS loss function $\mathcal{L}(\beta)$ in our case can be written as:

$$\mathcal{L}(\beta) = (g^*(z_{t-1}) - r_t)^2 \ . \tag{10}$$

The penalty term $\phi(\beta; \cdot)$ in Equation 9 can take many functional forms and in this study, three advanced regression methods with different types of loss functions are evaluated: ridge regression, lasso regression, and elastic net regression. All hyperparameters of the penalized linear models applied in our study are optimized using 64-fold cross-validation and can be found in Appendix 6.

The ridge regression, as made popular by Hoerl and Kennard (1970), applies a penalty in the form of:

$$\phi(\beta; \lambda) = \lambda \sum_{j=1}^{P} \beta_j^2, \tag{11}$$

meaning that the model is penalized for the sum of squared coefficients. This implies that some coefficients are suppressed when minimizing the residual sum of squares. Important to note is that in a ridge regression the coefficients do not converge exactly to zero as a result of the regularization process. This means that the penalty term in ridge regressions helps in the bias-variance tradeoff of prediction tasks but does not inherently help with predictor selection, although some high-value coefficients are suppressed to a value closer to zero. When predictors are many, and potentially highly correlated, a traditional OLS regression can estimate one of these to a giant positive value, only to be counterbalanced by a similar but negative value on the correlated variable. Imposing a penalty on coefficient size, which is effectively what the ridge regularization process does, alleviates the problem by punishing large coefficients and as such results in predictors having smaller coefficients.

The least absolute shrinkage and selection operator ("lasso") regression, as presented by Robert Tibshirani (1996), has a penalty term which is similar to ridge:

$$\phi(\beta; \lambda) = \lambda \sum_{j=1}^{P} |\beta_j|. \tag{12}$$

However, the fact that it applies a penalty to the sum of the absolute value of the estimated parameters β_j has some important implications. Firstly, the nature of how the penalty term is constructed makes it both possible and likely that it will produce some coefficients exactly equal to zero. The implication of coefficients set to zero is that the model not only alleviates problems related to multicollinearity, like the ridge loss function, but also automatically assists in the model specification by dropping variables.

The elastic net regularization (Zou and Hastie, 2005) is essentially a compromise between the ridge and lasso penalties, taking the form:

$$\phi(\beta; \lambda_1, \lambda_2) = \sum_{j=1}^{P} (\lambda_1 |\beta_j| + \lambda_2 \beta_j^2). \tag{13.1}$$

From Equation 13.1 we define the hyperparameter α as a function of λ_1 and λ_2:

$$\alpha = \frac{\lambda_2}{(\lambda_1 + \lambda_2)}\ , \tag{13.2}$$

allowing us to rewrite the loss function in Equation 13.1 to:

$$\phi(\beta; \lambda, \alpha) = \lambda_{1+2} \sum_{j=1}^{P} ((1-\alpha)|\beta_j| + \alpha \beta_j^2). \tag{13.3}$$

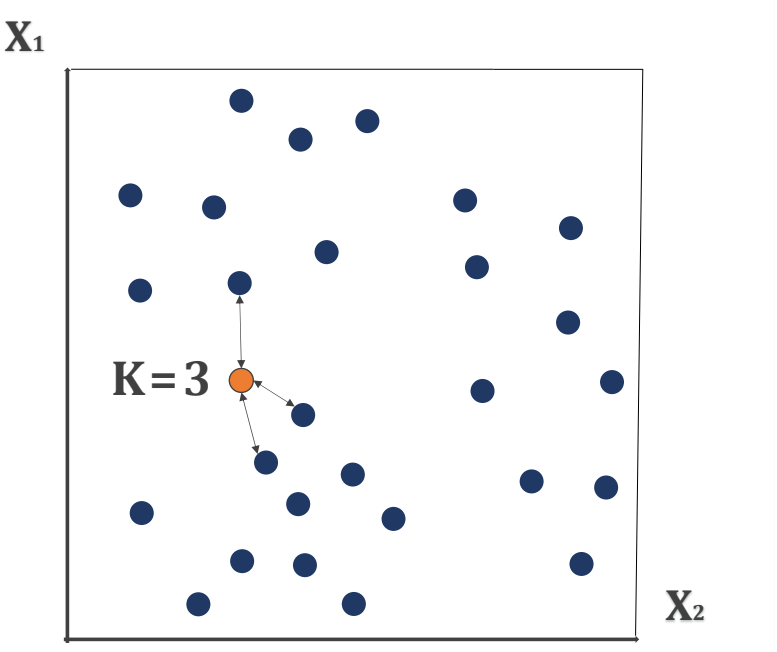
As can be seen from Equation 13.3, the two previously defined penalized linear models are at the extremes of the hyperparameter α, and setting it to exactly 0 or 1 yields a lasso or ridge penalty term, respectively. Simplified, this means that the hyperparameter α defines how much the model behaves like a ridge regression versus a lasso regression. In practice, the elastic net regression mitigates problems with correlated predictors through two mechanisms, as specified by Hastie et al. (2017). The ridge part of the error term tends to average highly correlated coefficients and the lasso part favors a sparse solution with respect to the number of coefficients, meaning that some of them are likely to be exactly zero (Hastie et al., 2017). Conclusively, the elastic net is often highly effective when the number of predictors exceeds

the number of observations, but its features are also well suited to handle many, possibly correlated, predictors in general (Zou and Hastie, 2005).

4.2.2 K-Nearest-Neighbors ("KNN")

KNN's history goes back at least to Fix and Hodges (1952) and is based on the rather simple idea of making predictions based on how close the new observation is to the historical data. Despite having been around for some time and being based on a rather simple idea, it has been successful for prediction tasks, at the very least when it comes to classification problems (Hastie et al., 2017). The model can also handle continuous data and is then usually referred to as a KNN regression, even though it is a non-linear model. As the model is non-parametric and quite intuitive visually at the basic level, its basic properties are best described via imagery.

Figure 1. Illustration of K-Nearest Neighbors Regression Properties



Notes: Illustration of KNN's basic properties. When new data is introduced to the model the prediction is based on the average of the K nearest neighbors in the predictor space. In this illustration, there are only two dimensions, whereas our model has 48 predictors. The prediction of the out-of-sample data, illustrated by the orange dot, is typically based on the average of the K nearest neighbors. X_n represents predictors.

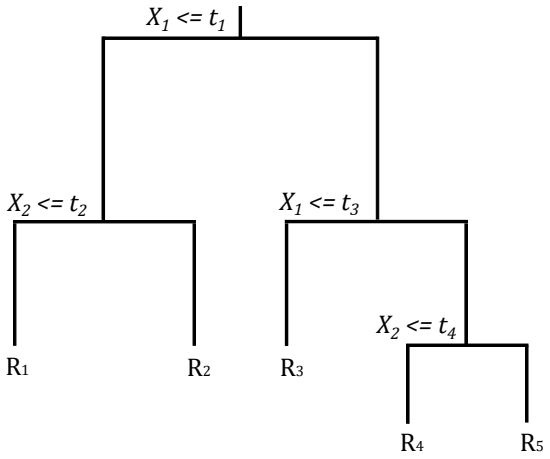
KNN makes predictions of new out-of-sample data by taking the average of the excess return on the S&P 500 for the K nearest neighbors. There is also an option to make the model slightly more complex by weighting the closer neighbors higher in the prediction, as compared to a straight average. Compared to the two-dimensional version presented in Figure 1 we have significantly more predictors, but the intuition is still the same, the model's output is based on

the average of the K nearest neighbors. Despite being more successful for classification problems, its simplicity is appealing when trying to build an intuitive model for predicting excess returns. When applying eightfold cross-validation it turns out that the optimal value for K is 12, and that weighting the distances for prediction is sub-optimal in our case.

4.2.3 Classification and Regression Trees ("CART")

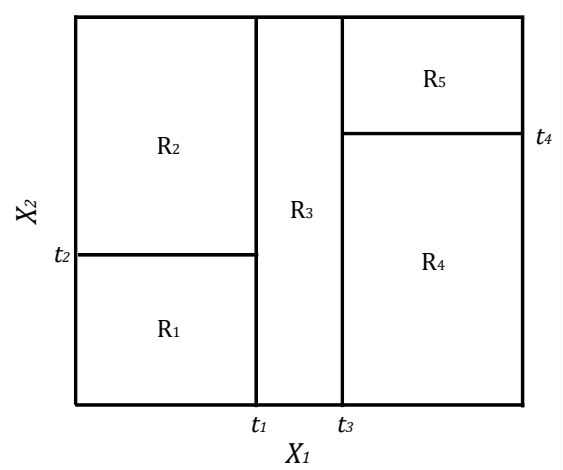
CART was formally introduced in a book by Breiman et al. (1984), building on the logic of decision trees dating back several decades earlier (see, for example, Morgan and Sonquist, 1963). Just like KNN, it can handle both classification and regression data, but as we perform no classification, only the regression tree is relevant for us. Again, it is important to note that although the name of the method includes the word "regression", it is a non-linear method which imposes no assumptions of the distribution of data. A regression tree is fitted to the training data by recursive binary splitting, which means that the data is partitioned repeatedly at different threshold values for the predictors in that specific model. This process generates several groups of data that have similar characteristics.

Figure 2.1 Decision Tree Partitions



Notes: Illustration of a decision tree partitioned into 5 terminal nodes (R_n) based on two predictors (X_n), t_n is the threshold value for each split.

Figure 2.2 Partitioned Predictor Space



Notes: Illustration of the predictor space of a decision tree with 5 nodes (R_n) based on two predictors (X_n), t_n is the threshold value for each split.

As illustrated in Figure 2.1, the first split is at threshold value t_1 for predictor X_1, giving two new branches with data being sorted in each one. Based on the data in each leg, new threshold values (t_2 and t_3) for predictors X_2 and X_1 are found, respectively, based on what best splits the data into disparate groups. This sequence is repeated and divides the predictor space into several regions (denoted as R_n in Figures 2.1 and 2.2) and approximates the value of the

independent variable as the mean of the values in that region when producing predictions. As with the KNN, the relatively simple intuition of a regression tree is appealing for excess return prediction. Since the regression tree makes no assumption regarding the relationship between predictors, and splits the data based on what forms the most distinct difference between nodes, the model is unlikely to make a second split on a correlated predictor. This suits our dataset, as some of the variables show high pairwise correlation (see Appendix 4 for a correlation matrix). The hyperparameters are optimized through eightfold cross-validation.

4.2.4 Random Forests

Random forests (Breiman, 2001) is fundamentally an extension of decision trees, in which many trees are combined with the purpose of reducing the out-of-sample error in prediction. Hastie et al. (2017) point out that regression trees *generally* produce low-bias predictions, which often results in a high variance in their out-of-sample predictions. Random forests use a modified version of bootstrap aggregation, commonly referred to as bagging, to reduce the variance. For our return data, the algorithm repeatedly draws a bootstrap sample from the training data and fits a tree using a random selection of predictors to the data, until a certain pre-specified number of trees are fitted. As mentioned in Gu et al. (2020), considering a lower number of predictors for each tree than the total available lowers the correlation between the trees. This is important as the total variance of the model has a negative relationship to the correlation between trees (Hastie et al., 2017). The prediction on out-of-sample data consists of the average prediction of all of the trees in the random forest. Hyperparameters consist of the depth of the trees, the number of predictors considered in each of the splits, as well as the number of bootstrap samples, and are all optimized using eightfold cross-validation.

4.2.5 Light Gradient Boosting Machines (“LGBM”)

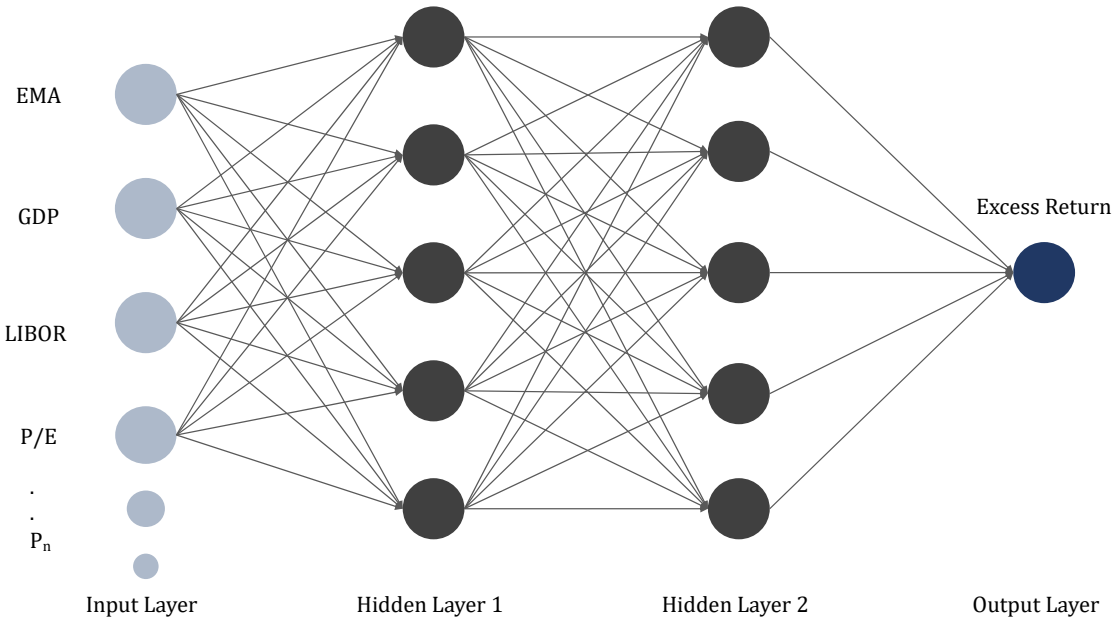
Gradient boosting decision trees is another popular extension for improving the predictions of decision trees. Like the random forests, this model is also considered an ensemble learning method, meaning that it fits multiple trees on the data with the purpose of achieving better predictions when aggregated than those of individual trees. Instead of fitting each model on a bootstrapped sample of the data, it sequentially fits a tree on a modified version of the initial data set (James et al., 2013). Simplified, it fits a number of trees B, on the prediction residuals from the previous trees. This procedure allows each fitted tree to learn from the mistakes of those prior. LGBM (Ke et al., 2017) is a modification of gradient boosting, requiring less

computational power than its predecessors. What makes it unique in practice is that the trees are grown node by node, instead of level by level as other boosting algorithms. Simplified, this means that the algorithm considers each individual node when deciding on the next split, instead of growing all terminal nodes simultaneously. Boosting algorithms also tend to perform well with noisy data and be efficient in finding non-linear patterns. One of the reasons for performing well is believed to be the slow learning rate of the model, allowing each tree to learn from the previous ones and improve predictions where it is not performing well (James et al., 2013). Unlike random forests, the trees are not de-correlated as each tree learns from the previous trees. This means that an LGBM model is more easily overfitted, requiring caution when optimizing the hyperparameters. The learning rate, the number of trees, and the number of splits in each tree are controlled by the hyperparameters, all of which are tuned in eightfold cross-validation in our study.

4.2.6 Artificial Neural Networks ("ANN")

Artificial neural networks are non-linear models with a mild similarity to how a biological brain supposedly works. Input variables, in our case predictors of the equity premium, are fed to the model. At each neuron, depicted as grey dots in Figure 3, the variables can interact or be transformed after which they are fed forward to the next hidden layer. A feed-forward neural network can in theory have anywhere between 1 to N hidden layers where the data is transformed. The final layer is the output layer, in which the output from the aforementioned hidden layers is provided, in our case prediction of the excess return. We apply a specific neural network algorithm commonly referred to as ADAM, because of its suitability in handling a vast number of predictors (Kingma and Ba, 2014). Our neural network has two hidden layers, which is decided through our eightfold cross-validation along with the other hyperparameters. While having a record of being fruitful in complex prediction tasks involving non-linear patterns, ANN is known to be notoriously hard to interpret and having a lack of transparency, as pointed out by Gu et al. (2020). Despite its relative lack of interpretability, it has been successfully implemented for predicting equity returns by, for example, Gu et al (2020) and Feng et al. (2018).

Figure 3. Illustration of the Basic Properties of a Feed-Forward Artificial Neural Network



Notes: The figure depicts the basic properties of how an artificial neural network works. Simplified, predictor data is fed to the model and the neurons (depicted by grey dots in the hidden layers) which are arranged in hidden layers in the model. At each neuron, they are transformed and interactions between variables can take place, and signals are sent from neuron to neuron until it reaches the final layer which is the output. P_n is the n:th predictor in the dataset.

4.3 Predictor Importance

With the ambition of understanding which predictors have the most impact on our decision-tree based predictions of the equity premium, we extract the measure of relative variable importance for the final month of prediction. As a result of using an expanding window of observations, this is the month in which the models have been trained on all of the data except for the very last month in the sample. As such, we hope that it can shed some light on which variables have the most impact on predictions. The relative importance score is based on the Gini importance score[7] as first suggested in the original CART paper by Breiman et al. (1984). For a single regression tree, it is exactly equal to the Gini importance score. For additive tree models, like the random forest applied in our study, it scales by averaging the score over the number of trees. However, when the individual trees become non-additive the computation gets more complicated, and explaining it in detail is outside the scope of this paper. As the measure is relative, we scale all results to a value between 0 and 100 to make comparisons across the tree-based models.

[7] For a more comprehensive yet high-level explanation of the Gini impurity score see section 10.13.1 in Hastie et al. (2017).

The coefficients of the final month's penalized linear regressions are also reported, mostly to illustrate which predictors' coefficients are penalized to converge to zero. The coefficients are to be interpreted with care as they are updated each month when observations are added to the training set.

For KNN and ANN, we do not provide any measures of variable importance. One of the reasons is that the discussion regarding how to measure the importance of predictors in a feed-forward ANN model is still active in the computer science literature (see, for example, de Sa', 2019). Instead, we focus on the more well-established measurement methods, as our core focus is comparing the performance of predictive models, and not measures for variable importance.

5 EMPIRICAL RESULTS

In the following sections, we present empirical results from the outlined methodology. First, we compare models based on predictive performance. Second, we present relative predictor importance values for selected models. Finally, we translate predictions made by the models to their inherent directional movement and present the performance of the naïve investment strategy. Where applicable, we offer comparisons to findings presented in previous financial literature. The existing research typically differs in several aspects, making it difficult to find adequate benchmark studies. Hence, the comparisons made in the following sub-sections will be supplemented by further comments on differences in methodology.

5.1 Out-of-Sample Performance

Of the eight models applied to predict the equity premium, a total of six outperform the benchmark forecast, illustrated by the positive values of R^2_{OS}. Compared to the traditional predictive regressions evaluated by Welch and Goyal (2008), these six models show higher accuracy. It should be noted that their study re-evaluates previously suggested predictors, rather than attempting to find a new predictive model. The best performing model in our study is LGBM, with an out-of-sample R^2 of 0.261, followed by regression trees and random forests. Interestingly, all of these are based on the non-linear algorithmic procedure in decision trees and produce significantly different forecasts compared to the benchmark, as shown by the

DM*-test statistic. In terms of ensemble learning methods, boosting was more successful than bagging in our context, as seen by the outperformance of LGBM relative to random forests.

All of the penalized linear methods outperform the historical average, albeit with a narrow margin. The ranking of performance based on RMSE shows that the penalized linear models are separable from each other in terms of performance, despite showing identical results down to the fourth decimal. While the forecasts of the penalized linear models are not significantly different from the historical average in the modified Diebold-Mariano test, it can be translated into a Sharpe ratio improvement of 0.15. The narrow but consistent outperformance of the benchmark using penalized linear models aligns with the results of Feng et al. (2018). The authors find that both lasso and elastic net regressions outperform narrowly when using an expanding window to estimate models. Also, all of Feng et al.'s penalized linear models demonstrate similar predictive performance, which is comparable to what we find. Feng et al.'s study differs from ours in many aspects; the time horizon; the number of variables, variable type, considering interactions between variables; and predicting logarithmic returns instead of simple. Importantly, their main focus is to evaluate ANN models, most of which outperform the historical average in their study, in contrast to our implementation of ANN.

The results of our models' predictive performance are summarized in Table 1, alongside the DM test statistic and the implied Sharpe ratio improvement.

Table 1. Out-of-Sample Performance – Full Prediction Period

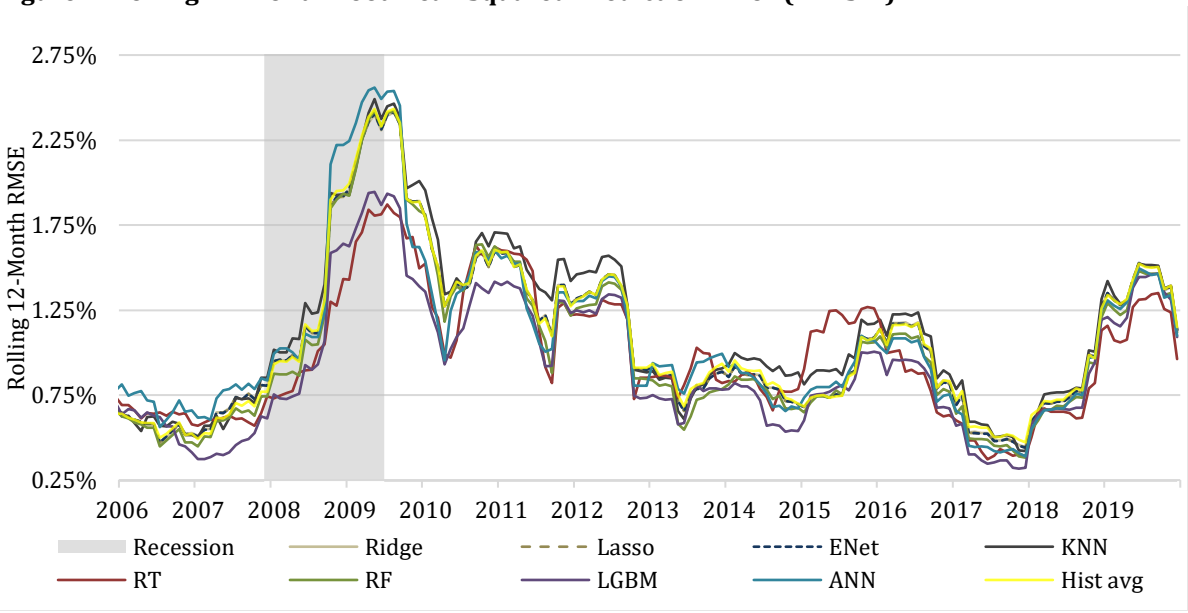
	Penalized Linear			**Non-Linear**					
	Ridge	**Lasso**	**ENet**	**KNN**	**RT**	**RF**	**LGBM**	**ANN**	**Hist. Avg.**
R^2_{OS}	0.011	0.011	0.011	-0.092	0.192	0.062	0.261	-0.009	N/A
RMSE	3.98%	3.98%	3.98%	4.18%	3.60%	3.88%	3.44%	4.02%	4.00%
RMSE Rank	6	5	4	9	2	3	1	8	7
*DM**	1.49	1.55	1.55	-2.57**	1.76*	2.16**	4.79***	-0.14	N/A
*SR**	0.53	0.53	0.53	N/A	1.74	0.98	2.11	N/A	N/A
SR-SR*	0.15	0.15	0.15	N/A	1.36	0.59	1.72	N/A	N/A

Notes: This table presents the full prediction period performance of our predictive models in terms of out-of-sample R^2, root mean squared error("RMSE"), and a rank based on the root mean squared error ("RMSE rank"). The modified Diebold-Mariano test statistic is shown (DM*), with standard significance thresholds shown as "*", "**", "***", for a significant difference in forecast compared to that of the historical average at the 10%, 5%, and 1% level, respectively. The adjusted Sharpe ratio (SR*) and improvement in Sharpe ratio (SR*-SR) vis-à-vis a passive investor is also reported. N/A is used for values not applicable to that specific model. Ridge, lasso and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied.

The machine learning models perform well in predicting returns in the early part of our out-of-sample data, but only some of them beat the historical average in terms of rolling 12-month RMSE. During the financial crisis of 2008 to 2009, all models show increasing errors

as the market becomes more unstable. LGBM and regression trees seem to have lower errors in those years, yet also shows large increases in rolling RMSE. From mid-2009 to the end of 2017, all models are on a downward trend in the error measure, with a few seemingly simultaneous disruptions to the trend. ANN is the worst performing in years classified as a period of recession by the National Bureau of Economic Research ("NBER"). As the model gets more data via the expanding window and the market stabilizes, it recovers and actually has errors comparable to the best models in some periods. This aligns with the findings of Feng et al (2018) to some extent, as they point out that the ANN models perform better the more data they are allowed to train on. As such, using a longer horizon and being able to reoptimize hyperparameters periodically like in Gu et al. (2020) could have the potential to improve the model's predictions significantly. We present a graph showing the development of the 12-month rolling RMSE for all models in Figure 4.

Figure 4. Rolling 12-Month Root Mean Squared Prediction Error ("RMSE")

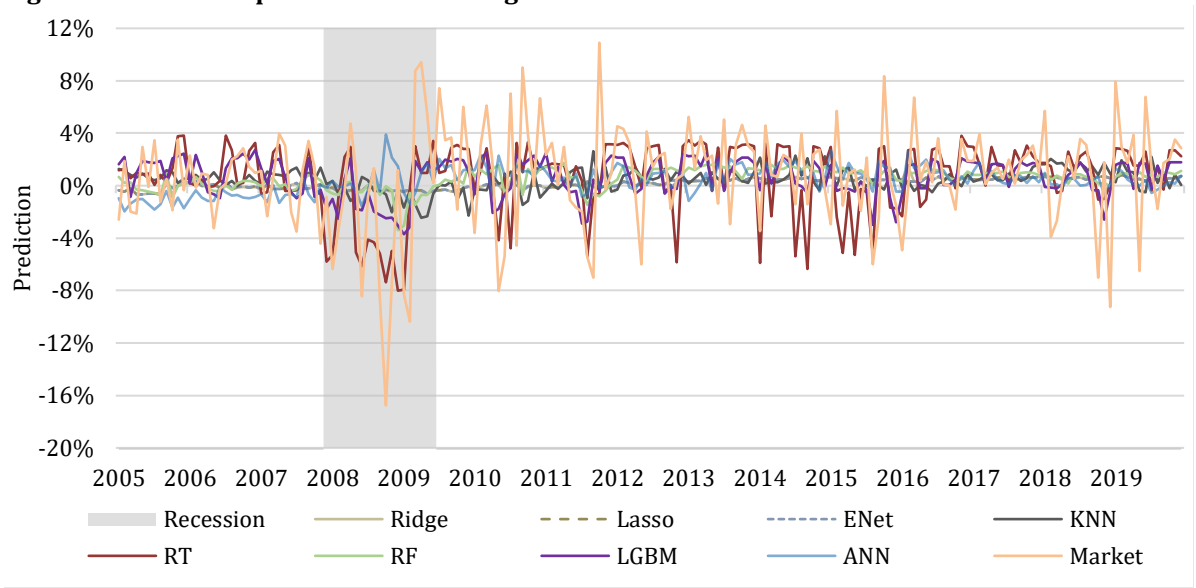


Notes: The chart displays the rolling RMSE for the last twelve months. Ridge, lasso and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied. The shaded area represents the only NBER recession in our out-of-sample data.

The models seem to behave quite differently in how they predict excess returns. All of our penalized regression methods produce very stable predictions compared to the other models. In fact, looking at Figure 5 they are too close to be separable visually. We believe that these models are under-fitted as the predictions seem to be roughly around 0, however, their predictions are still better than the benchmark, as seen in Table 1. LGBM shows more variation in its predictions and seems to track the market realized excess returns more accurately,

explaining why it is the best performing model in terms of out-of-sample R^2. The regression tree also follows market movements quite well, although it has some large deviations in its predictions between 2014 and 2015. It also becomes evident from Figure 5 why the models' rolling RMSE increases so significantly during the financial crises, as the large deviations from the realized return of the market can be seen clearly in the shaded area. The magnitude of variation in realized market returns also sheds some light on the deviations from the downward trend in Figure 4. In periods where market movements are large, the models generally have higher errors in their predictions.

Figure 5. Out-of-Sample Machine Learning Prediction and Realized Excess Return



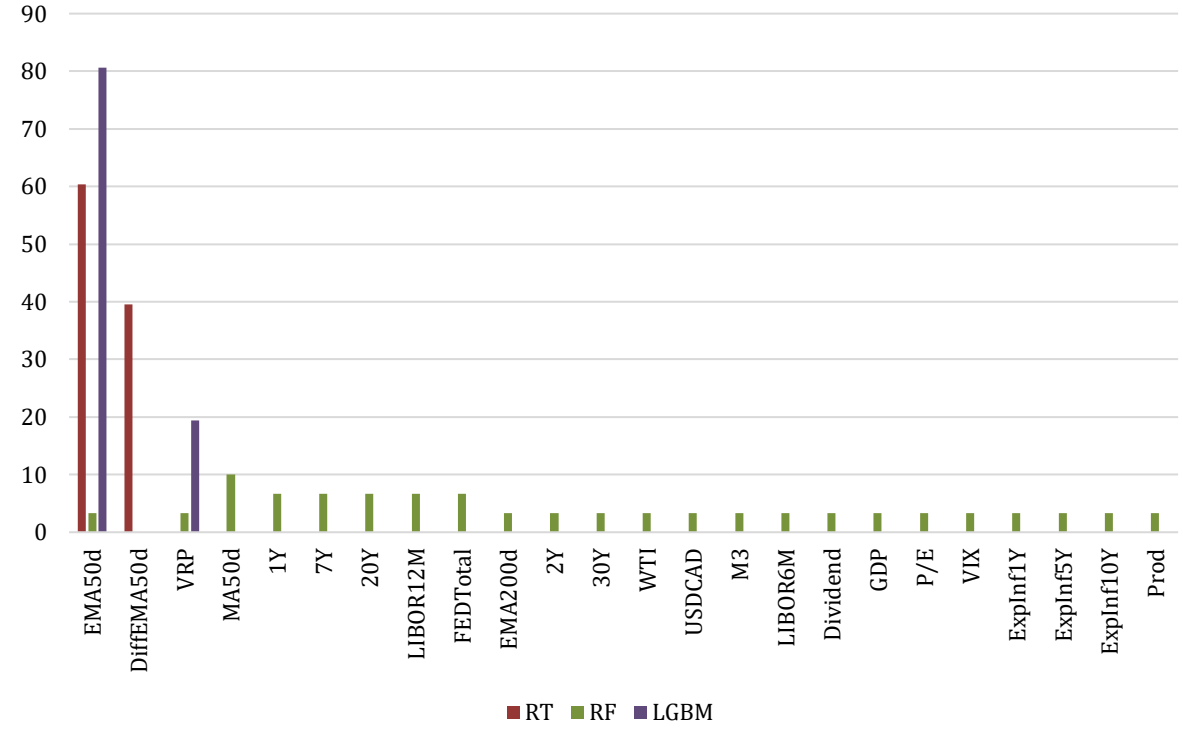
Notes: The chart shows the monthly prediction of each machine learning model, and the realized excess return on S&P 500 ("Market"). For readability purposes the colors have been altered slightly vis-à-vis the other charts presented in this paper. Ridge, lasso and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied. The shaded area represents the only NBER recession in our out-of-sample data.

5.2 Predictor Importance

In the final prediction period, the variable importance of the tree-based methods shows that among the top-ranking predictors are changes in moving average and exponential moving average, as well as the change in the moving average compared to the current index level of S&P 500. It can arguably be said that these could serve as proxies for two known important effects in financial markets, namely momentum and mean reversion. While momentum has previously been found to generate abnormal returns both for both cross-sectional (Jegadeesh and Titman, 1993) and time-series implementation (Moskowitz et al., 2012), any effect on our predictions would be stemming from time-series momentum, as we are predicting only the

S&P 500. Gu et al. (2020) present the one-month momentum as the overall most important predictor across their applied models, although their study is based on data on individual stocks and ranges over a longer time horizon. Both LGBM and regression trees assign importance to only a select few variables. This is likely to be a result of the hyperparameter tuning process, where the attempts of reducing out-of-sample variance have excluded the majority of the predictors. Looking back at Table 1, LGBM is our best performing model and forecasts the equity premium far more accurately than the benchmark in our sample. LGBM's forecast is also different from the benchmark with high significance. It is the only model in our selection with relatively high importance assigned to the Variance Risk Premium ("VRP"), which we find somewhat surprising as the VRP has been shown to be a strong predictor of returns, albeit with a peak in predictability at a four-month horizon (Bollerslev et al., 2014). As expected, the random forest differs from the other tree-based methods by assigning weights more evenly across several predictors, resulting from the de-correlated tree growing process. In Figure 6, we display a chart of final month predictor importance for all tree-based models.

Figure 6. Predictor Importance for Tree Based Models



Notes: The chart shows the relative predictor importance values in the final prediction month, on a scale of 0-100. The values are available for regression trees. ("RT"), random forests ("RF"), and LGBM. Refer to Appendix 1 for a list of variable descriptions.

In Appendix 5, we present the coefficients of our penalized regression models for the prediction of the final month. These are to be interpreted with caution as the models' coefficients are updated each month, as new data is added to the training data successively. However, after inspecting them, we are reasonably confident that our hyperparameters, selected via cross-validation, resulted in underfitting the models as many coefficients are zero or extremely close to zero. Lasso and elastic net are expected to force some coefficients to zero, but the non-zero coefficients are surprisingly small. Furthermore, an underfit of the models would explain why the predictions of excess returns given by the penalized linear models are hovering around zero.

5.3 Directional Prediction and Naïve Investment Strategy

After having converted the models' predictions to their inherent prediction of directional movement, it is evident that they are generally better in predicting the months with a realized positive return than the negative ones. LGBM is the most successful in this aspect as well, with a total accuracy of 71.7%. It performs well in predicting the positive returns, but especially outperforms other models in negative months. The penalized linear models have a high similarity also in directional prediction and the lasso and elastic net regressions output identical results. In Table 2, the full directional prediction results are presented, divided into realized positive, realized negative, and total accuracy.

Table 2. Directional Prediction Accuracy

		Penalized Linear			Non-Linear					
		Ridge	**Lasso**	**ENet**	**KNN**	**RT**	**RF**	**LGBM**	**ANN**	**Hist. Avg.**
Realized Positive	*True Positive*	80	82	82	87	92	104	92	87	81
	False Negative	39	37	37	32	27	15	27	32	38
	Total Real. Pos.	119	119	119	119	119	119	119	119	119
	Accuracy Pos.	67.2%	68.9%	68.9%	73.1%	77.3%	87.4%	77.3%	73.1%	68.1%
Realized Negative	*True Negative*	25	23	23	12	32	23	37	31	26
	False Positive	36	38	38	49	29	38	24	30	35
	Total Real. Neg.	61	61	61	61	61	61	61	61	61
	Accuracy Neg.	41.0%	37.7%	37.7%	19.7%	52.5%	37.7%	60.7%	50.8%	42.6%
Total	*Total True*	105	105	105	99	124	127	129	118	107
	Total False	75	75	75	81	56	53	51	62	73
	N. Total Periods	180	180	180	180	180	180	180	180	180
	Total Accuracy	58.3%	58.3%	58.3%	55.0%	68.9%	70.6%	71.7%	65.6%	59.4%

Notes: The table presents the accuracy of the inherent directional predictions of our models, versus the prediction of the historical average return. We report the results both in terms of the number of instances and the accuracy in percent. The panes divide the results into the months where the realized return was positive or negative and provides an aggregation of these in the total-pane. True/False describes whether the prediction was right or wrong, respectively. Positive/Negative refers to if the realized return was positive or negative, respectively. As an example, "True Positive" represents when a model predicted a positive return and the realized return turned out to be positive. Ridge, lasso, and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forests ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied.

When using the directional results in our naïve investment strategy, all models but KNN produce a higher Sharpe ratio than the benchmark portfolio of a buy-and-hold index investor ("Market"), as shown in Table 3. Despite underperforming the benchmark in predicting the magnitude of returns, the naïve strategy using ANN's directional forecast shows a greater Sharpe ratio than the market. ANN's relative success in this regard is a consequence of performing well in directional predictions, compared to the other models. Unsurprisingly, all of the respective models' strategies have lower volatility, which is likely a result of not being invested in the market when the forecasted excess return is negative.

Table 3. Naïve Investment Strategy - Annualized Return, Standard Deviation, and Sharpe Ratio

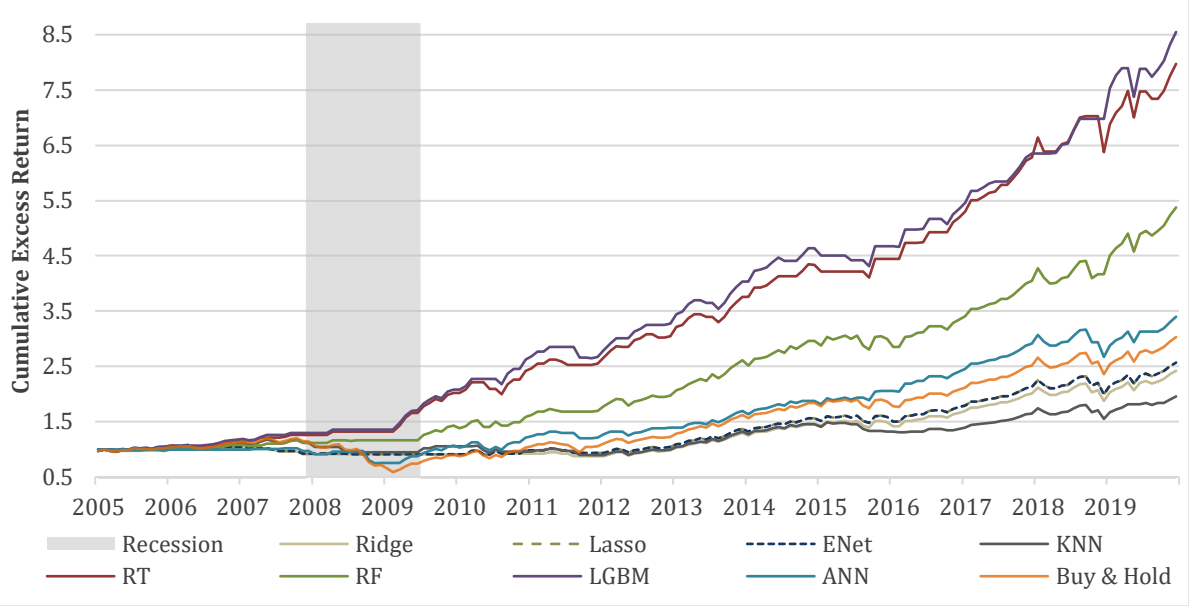
	Penalized Linear			Non-Linear					
	Ridge	**Lasso**	**ENet**	**KNN**	**RT**	**RF**	**LGBM**	**ANN**	**Market**
Ann. Return	6.3%	6.8%	6.8%	4.9%	14.3%	11.7%	14.8%	8.8%	8.4%
Ann. StDev	9.3%	9.5%	9.5%	9.6%	9.2%	9.5%	8.7%	11.4%	13.7%
Sharpe Ratio	0.681	0.714	0.714	0.517	1.561	1.236	1.700	0.774	0.610

Notes: The table shows the annualized returns ("Ann. Return"), standard deviations ("Ann. StDev"), and Sharpe Ratio of the naïve investment strategy, compared to that of a buy-and-hold investor ("Market"). Ridge, lasso and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied.

As seen in Figure 7, LGBM and regression trees undoubtedly benefit from being the best in predicting the negative return months. As a result, they lose a lot less value in the financial

market turmoil of 2008 to 2009, and in other market downturns. While we find the high returns and Sharpe ratios rather impressive, they are actually lower than the Sharpe ratio of 5.8 that Fischer and Krauss (2018) present. Their study differs from ours in many ways, yet the magnitude still puts our results in perspective and shows what could possibly be achieved.

Figure 7. Cumulative Return of Naïve Investment Strategy vs. Buy-and-Hold Market



Notes: The chart shows the cumulative return as if $1 was invested in January 2005, then timed the market based on our naïve investment strategy. It also shows the cumulative return of a buy-and-hold investment ("Buy & Hold"). Ridge, lasso and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied. The shaded area represents the only NBER recession in our out-of-sample data.

6 DISCUSSION

By applying machine learning methods to predict the equity premium, we are able to outperform the benchmark. The enhanced predictability can be exploited to earn higher returns with lower volatility, leading to a considerable improvement in Sharpe ratio. Considering that our models' hyperparameters were only optimized once, we believe that it should be possible to increase accuracy by allowing for re-optimization at regular intervals. While prediction results from a short time-period should be treated with caution, we believe that our study emphasizes the great promise of machine learning for equity premium prediction. Furthermore, our results suggest that non-linear models better describe the monthly movements in equity

premium than penalized linear models. We interpret this as an indication of non-linearity in the time-series variation of the equity premium.

The relative complexity of machine learning models often results in a loss of interpretability, especially regarding what drives the output of models. Yet, we are able to present an indication of which of the variables applied in our study that are more successful in predicting the excess returns on the S&P 500. The variable importance aligns reasonably well with previous literature, despite the introduction of many non-proven predictors in our study. Hence, we interpret the selection and importance of predictors as a sign of robustness in our models, even though comparable studies differ in methodology and time horizon.

To some extent, it seems like we have succeeded in selecting machine learning models that can handle a vast predictor set. A majority of the models consider only a selected set of important variables. Some, like random forest, make use of more predictors but assign higher importance to the predictors which are also used in the sparser models. This relative success in model selection also shows that our inclusion of variables with impact on the constituents of the S&P 500 had limited prediction power in our study. We were unable to report any noteworthy results relating to the importance of, for instance, currency exchange rates or interest rates. Instead, the most successful predictors came from existing literature.

Converting the excess return predictions to predictions of the directional movement produced some encouraging results. Despite optimizing our models to predict the magnitude, and not the direction of the excess return, the naïve investment strategy generates an improvement of the Sharpe ratio compared to a buy-and-hold investor. Predicting directional movement is essentially a classification problem rather than a regression problem, meaning that re-optimizing our models for that purpose could possibly have generated even higher accuracy.

We propose that future studies of machine learning applications for equity premium prediction focus on two issues. First, increasing the interpretability of models. This can possibly be achieved by quantifying predictor importance in a structured way across algorithms. Second, testing machine learning applications within the context of a theoretical framework relevant to financial markets. Combining the predictive ability of machine learning algorithms with existing, or new, financial theory has the potential to immensely enhance our understanding of the equity premium.

References

Bollerslev, Tim, James Marrone, Lai Xu, and Hao Zhou, 2014, Stock Return Predictability and Variance Risk Premia: Statistical Inference and International Evidence, *The Journal of Financial and Quantitative Analysis* 49, 633-661.

Breiman, Leo, 2001, Random Forests, *Machine Learning* 45, 5-32.

Breiman, Leo, H. J. Friedman, A. R. Olshen, and J. C. Stone, 1984, *Classification and Regression Trees* (Routledge, New York).

Campbell, John Y., 2008, Viewpoint: Estimating the equity premium, *The Canadian Journal of Economics* 41, 1-21.

Campbell, John Y., and Samuel B. Thompson, 2008, Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average? *The Review of Financial Studies* 21, pp. 1509-1531.

Carhart, Mark M., 1997, On Persistence in Mutual Fund Performance, *The Journal of Finance (New York)* 52, 57-82.

Cochrane, John H., 2008, The Dog That Did Not Bark: A Defense of Return Predictability, *The Review of Financial Studies* 21, 1533-1575.

Dangl, Thomas, and Michael Halling, 2012, Predictive regressions with time-varying coefficients, *Journal of Financial Economics* 106, 157-181.

de Sa', C. R., 2019, Variance-Based Feature Importance in Neural Networks, *LNCS, Volume 11828*, 306-315.

Diebold, Francis X., and Robert S. Mariano, 1995, Comparing Predictive Accuracy, *Journal of Business & Economic Statistics* 13, 134-144.

Dow, Charles H., 1851-1902, 1920, *Scientific stock speculation*, Magazine of Wall Street, c1920, New York (State).

Fama, Eugene F., and Kenneth R. French, 1988, Dividend yields and expected stock returns, *Journal of Financial Economics* 22, 3-25.

Feng, He, and Polson, 2018, Deep Learning for Predicting Asset Returns, *Working Paper*, 1-23.

Ferson, Wayne E., Sergei Sarkissian, and Timothy T. Simin, 2003, Spurious Regressions in Financial Economics? *The Journal of Finance (New York)* 58, 1393-1413.

Fischer, Thomas, and Christopher Krauss, 2018, Deep learning with long short-term memory networks for financial market predictions, *European Journal of Operational Research* 270, 654-669.

Fix, Evelyn, and Jr Hodges J L, 1952, Discriminatory Analysis - Nonparametric Discrimination: Small Sample Performance, *U.S. Air Force, School of Aviation Medicine*.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2020, Empirical Asset Pricing via Machine Learning, *The Review of Financial Studies* 33, 2223-2273.

Harvey, David, Stephen Leybourne, and Paul Newbold, 1997, Testing the equality of prediction mean squared errors, *International Journal of Forecasting* 13, 281-291.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman, 2017, *The Elements of Statistical Learning* (Springer New York, New York, NY).

Hoerl, Arthur E., and Robert W. Kennard, 1970, Ridge Regression: Biased Estimation for Nonorthogonal Problems, *Technometrics* 12, 55-67.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani, 2013, *An introduction to statistical learning* (Springer, New York).

Jegadeesh, Narasimhan, and Sheridan Titman, 1993, Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency, *The Journal of Finance* 48, 65-91.

Johannes, Michael, Arthur Korteweg, and Nicholas Polson, 2014, Sequential learning, predictability, and optimal portfolio returns, *The Journal of Finance* 69, 611-644.

Kara, Yakup, Melek Acar Boyacioglu, and Ömer K. Baykan, 2011, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange, *Expert Systems with Applications* 38, 5311-5319.

Ke, Meng Qi, Thomas Finley, Wang Taifeng, Chen Wei, Ma Weidong, Ye Qiwei, and Liu Tie-Yan, 2017, LightGBM: a highly efficient gradient boosting decision tree, *In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, 3149-3157.

Kingma, Diederik P., and Jimmy Ba, 2014, Adam: A Method for Stochastic Optimization, *arXiv*, 1-15.

Mariano, Roberto S., 2004, Testing Forecast Accuracy, in Anonymous *A Companion to Economic Forecasting* (John Wiley & Sons, Ltd).

Morgan, James N., and John A. Sonquist, 1963, Problems in the Analysis of Survey Data, and a Proposal, *Journal of the American Statistical Association* 58, 415-434.

Moskowitz, Tobias J., Yao H. Ooi, and Lasse H. Pedersen, 2012, Time Series Momentum, *Journal of Financial Economics* 104, 228-250.

Patel, Jigar, Sahil Shah, Priyank Thakkar, and K. Kotecha, 2015, Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques, *Expert Systems with Applications* 42, 259-268.

Rozeff, Michael S., 1984, Dividend yields are equity risk premiums, *Journal of Portfolio Management* 11, 68-75.

Sharpe, William F., 1994, The Sharpe Ratio, *Journal of Portfolio Management* 21, 49-58.
Tibshirani, Robert, 1996, Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society. Series B (Methodologial)* 58, 267-288.

Turing, A. M., 1950, Computing Machinery and Intelligence, *Mind* 49, 433-460.

Welch, Ivo, and Amit Goyal, 2008, A Comprehensive Look at The Empirical Performance of Equity Premium Prediction, *The Review of Financial Studies* 21, 1455-1508.

Zou, Hui, and Trevor Hastie, 2005, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 301-320.

APPENDIX

Appendix 1. Variables, Descriptions, and Sources

Name	Source	Explanation
S&P 500 Total Return	CRSP	The total monthly return on the S&P 500 (incl. distributions) (%)
S&P 500 Total Index (TI)	Thomson Reuters Eikon	The total monthly return index level of the S&P 500 (incl. distributions)
MA50d	Derived from S&P 500 TI	50-day moving average (monthly change in %)
MA200d	Derived from S&P 500 TI	200-day moving average (monthly change in %)
EMA50d	Derived from S&P 500 TI	50-day exponential moving average (monthly change in %)
EMA200d	Derived from S&P 500 TI	200-day exponential moving average (monthly change in %)
DiffMA50d	Derived from S&P 500 TI	The distance between the index level and 50-day MA (monthly change in %)
DiffMA200d	Derived from S&P 500 TI	The distance between the index level 200-day MA (monthly change in %)
DiffEMA50d	Derived from S&P 500 TI	The distance between the index level 50-day exponential MA (monthly change in %)
DiffEMA200d	Derived from S&P 500 TI	The distance between the index level 200-day exponential MA (monthly change in %)
3M	FRED	3-month treasury constant maturity rate (%)
6M	FRED	6-month treasury constant maturity rate (%)
1Y	FRED	1-year treasury constant maturity rate (%)
2Y	FRED	2-year treasury constant maturity rate (%)
3Y	FRED	3-year treasury constant maturity rate (%)
5Y	FRED	5-year treasury constant maturity rate (%)
7Y	FRED	7-year treasury constant maturity rate (%)
10Y	FRED	10-year treasury constant maturity rate (%)
20Y	FRED	20-year treasury constant maturity rate (%)
30Y	FRED	30-year treasury constant maturity rate (%)
Diff10Y2Y	FRED	10-year treasury minus 2-year treasury constant maturity rate (%)
Gold	FRED	Fixing price 10:30 A.M. (London time) in London Bullion Market, USD (monthly change in %)
Silver	FRED	Fixing price 12:00 noon (London time) in London Bullion Market, USD (monthly change in %)
Brent	FRED	Benchmark prices which are representative of the global market. (monthly change in %)
WTI	FRED	Benchmark prices which are representative of the global market. (monthly change in %)
USDEUR	FRED	Foreign Exchange Rate, averages of daily noon buying rates (monthly change in %)
USDGBP	FRED	Foreign Exchange Rate, averages of daily noon buying rates (monthly change in %)
USDCHF	FRED	Foreign Exchange Rate, averages of daily noon buying rates (monthly change in %)
USDJPY	FRED	Foreign Exchange Rate, averages of daily noon buying rates (monthly change in %)
USDCAD	FRED	Foreign Exchange Rate, averages of daily noon buying rates (monthly change in %)
RealEffR	FRED	Weighted avg. of bilateral exchange rates adjusted by relative consumer prices. (monthly change in %)

M3	FRED	A measure of money supply including all physical currency and deposits in checking accounts, deposits in savings accounts, certificates of deposit, institutional money market accounts, repurchase agreements, and other large liquid assets (monthly change in %)
LIBOR1M	FRED	1-month Overnight London Interbank Offered Rate, based on USD (%)
LIBOR3M	FRED	3-month Overnight London Interbank Offered Rate, based on USD (%)
LIBOR6M	FRED	6-month Overnight London Interbank Offered Rate, based on USD (%)
LIBOR12M	FRED	12-month Overnight London Interbank Offered Rate, based on USD (%)
ConsSent	FRED	According to the surveys covering three broad areas of consumer sentiment: personal finance, business and buying (index)
CPI	FRED	Consumer Price Index: all items for the United States (monthly change in %)
DiscRate	FRED	The interest rate at which the central bank lends to commercial banks to meet their liquidity needs (%)
Dividend	Thomson Reuters Eikon	Twelve-month S&P 500 price-to-earnings ratio
FEDTotal	FRED	Total Assets on Fed Balance Sheet (Less Elim. from Consolidation) (monthly change in %)
GDP	FRED/OECD	Leading Indicator of Gross Domestic Product for United States. (2012 = 100)
P/E	Thomson Reuters Eikon	Trailing twelve-month price-to-earnings ratio of the S&P500
VIX	Thomson Reuters Eikon	Volatility Index. A measure of expected price fluctuations in S&P 500 Index options over the next 30 days, calculated by the CBOE.
VRP	Hao Zhou's Website	Variance Risk Premium as defined by Bollerslev et al. (2014)
EVRP	Hao Zhou's Website	Expected Variance Risk Premium as defined by Bollerslev et al. (2014)
ExpInf1Y	Cleveland Fed	Estimates of the annual expected rate of 1-year inflation (%)
ExpInf5Y	Cleveland Fed	Estimates of the annual expected rate of 5-year inflation (%)
ExpInf10Y	Cleveland Fed	Estimates of the annual expected rate of 10-year inflation (%)
ExpInf30Y	Cleveland Fed	Estimates of the annual expected rate of 30-year inflation (%)
Prod	FRED	Industrial Production: Total Index (Real output for all facilities located in the US manufacturing, mining, electric, and gas utilities. 2012 = 100) (monthly change in %)

Notes: The table describes the data used in the study and the respective source of data collection for each variable. S&P 500 Total Return is the dependent variable in our analysis, the S&P 500 Total Index is not used as a predictor variable, rather as the basis for calculating the various moving average predictor variables. CRSP is the Center for Research in Security Prices. FRED is the Federal Reserve Economic Data, managed by St. Louis Fed. Cleveland Fed is the website of the Federal Reserve Bank of Cleveland. Hao Zhou is the website of one of the original inventors of the Variance Risk Premium measure. Thomson Reuters Eikon is the financial analysis software provided by Thomson Reuters and Refinitiv.

Appendix 2. Descriptive Statistics of Variables

Name	Mean	Median	St. Dev.	Min	Max
S&P 500 Total Return (%)	0.483	0.957	4.167	-16.755	10.900
MA50d (%)	0.537	0.970	2.624	-12.615	8.476
MA200d (%)	0.506	0.901	1.530	-6.131	4.499
EMA50d (%)	0.538	1.105	2.507	-13.864	5.452
EMA200d (%)	0.510	0.883	1.442	-5.525	2.485
DiffMA50d (%)	10.499	-0.037	92.183	-1.000	1092.245
DiffMA200d (%)	1.766	0.038	16.752	-0.996	255.072
DiffEMA50d (%)	12.266	-0.032	163.490	-0.998	2536.087
DiffEMA200d (%)	0.840	0.019	4.780	-0.985	57.245
6M (%)	1.779	1.170	1.843	0.040	6.390
1Y (%)	1.877	1.325	1.786	0.100	6.330
2Y (%)	2.118	1.625	1.707	0.210	6.810
3Y (%)	2.342	1.895	1.616	0.330	6.770
5Y (%)	2.781	2.485	1.456	0.620	6.690
7Y (%)	3.131	2.925	1.349	0.980	6.720
10Y (%)	3.432	3.395	1.234	1.500	6.660
20Y (%)	3.977	4.230	1.233	1.820	6.860
30Y (%)	4.082	4.275	1.081	2.120	6.630
Diff10Y2Y (%)	1.314	1.435	0.910	-0.410	2.830
Gold (%)	0.806	0.643	4.769	-18.785	13.153
Silver (%)	0.857	-0.107	8.435	-27.897	27.490
Brent (%)	0.781	1.776	8.603	-26.909	21.562
WTI (%)	0.712	1.742	8.385	-28.875	24.467
USDEUR (%)	-0.013	-0.076	2.293	-6.006	8.111
USDGBP (%)	0.110	-0.027	2.162	-5.810	10.015
USDCHF (%)	-0.171	-0.013	2.344	-7.016	12.397
USDJPY (%)	0.051	0.027	2.267	-6.201	7.658
USDCAD (%)	-0.028	-0.060	1.924	-5.832	11.954
RealEffR (%)	0.014	-0.009	1.199	-3.566	5.640
M3 (%)	0.500	0.470	0.365	-0.462	2.298
LIBOR1M (%)	1.928	1.310	1.925	0.151	6.804
LIBOR3M (%)	2.053	1.337	1.901	0.223	6.863
LIBOR6M (%)	2.195	1.570	1.850	0.322	7.105
LIBOR12M (%)	2.424	1.816	1.764	0.534	7.501
ConsSent	85.782	88.600	12.232	55.300	112.000
CPI (%)	0.018	0.020	0.029	-1.770	1.380
DiscRate (%)	2.295	1.750	1.842	0.500	6.250
Dividend (%)	1.890	1.900	0.371	1.110	3.600
FEDTotal (%)	0.915	0.121	4.870	-13.949	62.489
GDP	100.030	99.939	0.922	97.716	101.846
P/E	25.867	21.724	17.819	13.008	122.413
VIX	19.491	17.170	7.868	9.510	59.890
VRP	12.528	10.416	21.199	-218.564	80.611

EVRP	14.884	9.160	21.757	-48.195	201.397
ExpInf1Y (%)	1.986	1.954	0.627	-0.481	3.760
ExpInf5Y (%)	2.014	1.908	0.485	1.207	3.400
ExpInf10Y (%)	2.108	2.023	0.433	1.412	3.349
ExpInf30Y (%)	2.384	2.333	0.298	1.915	3.218
Prod (%)	0.066	0.109	0.655	-4.337	1.517

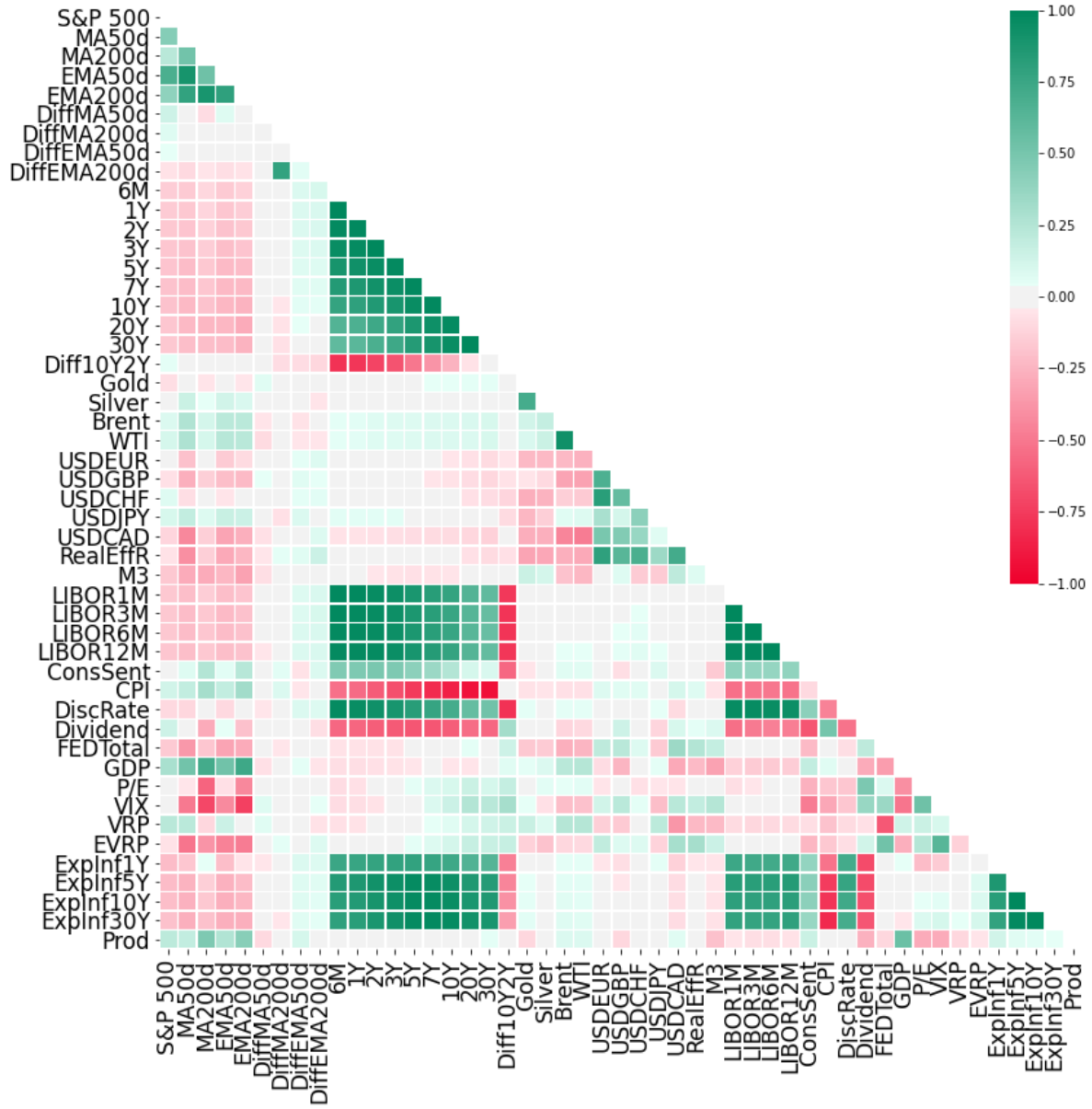
Notes: This table provides descriptive statistics of all variables used in the analysis, both for the dependent variable (S&P 500 Total Return %), and for the predictor variables. We present the mean, median, standard deviation, as well as the minimum and maximum value.

Appendix 3. Descriptive Statistics of Predictions Per Model

		Mean (%)	Median (%)	St. Dev. (%)	Min (%)	Max (%)
Penalized linear	*Ridge*	0.17	0.25	0.39	-0.66	1.44
	Lasso	0.18	0.29	0.38	-0.66	0.83
	Elastic Net	0.18	0.29	0.38	-0.66	0.83
Non-linear	*KNN*	0.45	0.46	0.86	-2.45	2.63
	RT	0.63	1.36	2.72	-8.01	3.81
	RF	0.55	0.67	0.79	-3.39	1.99
	LGBM	0.65	1.17	1.47	-3.71	2.73
	ANN	0.38	0.40	1.03	-1.96	3.90
	Hist. Avg.	0.12	0.10	0.25	-0.55	0.47

Notes: This table presents descriptive statistics of the predictions of excess return per model, as well as for the historical average ("Hist. Avg"). Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied.

Appendix 4. Correlation Matrix for All Predictors



Notes: This figure depicts the correlation matrix for all predictor variables, as well as the dependent variable. Please refer to Appendix 1 for a full description of predictors.

Appendix 5. Coefficients of Penalized Regression Methods, Final Prediction

Predictor	Ridge	Lasso	ENet
MA50d	2.19241E-10	0	0
MA200d	8.7079E-11	0	0
EMA50d	4.11153E-10	0	0
EMA200d	1.372E-10	0	0
DiffMA50d	7.56379E-07	2.24529E-05	2.07191E-05
DiffMA200d	-3.76496E-09	0	0
DiffEMA50d	-9.83555E-08	0	0
DiffEMA200d	-4.72966E-08	0	0
6M	-5.28338E-09	0	0
1Y	-4.85688E-09	0	0
2Y	-4.84896E-09	0	0
3Y	-4.62298E-09	0	0
5Y	-3.86231E-09	0	0
7Y	-3.53498E-09	0	0
10Y	-2.87937E-09	0	0
20Y	-2.27079E-09	0	0
30Y	-1.81517E-09	0	0
Diff10Y2Y	1.96959E-09	0	0
Gold	2.86045E-11	0	0
Silver	1.53101E-11	0	0
Brent	-1.18453E-10	0	0
WTI	-1.94343E-10	0	0
USDEUR	1.03529E-10	0	0
USDGBP	6.48482E-11	0	0
USDCHF	8.18497E-11	0	0
USDJPY	1.29045E-10	0	0
USDCAD	-2.53748E-11	0	0
RealEffR	2.54124E-11	0	0
M3	-1.48107E-11	0	0
LIBOR1M	-5.49118E-09	0	0
LIBOR3M	-5.28111E-09	0	0
LIBOR6M	-5.20652E-09	0	0
LIBOR12M	-5.25291E-09	0	0
ConsSent	-2.02635E-08	0	0
CPI	2.17786E-08	0	0
DiscRate	-1.18091E-09	0	0
Dividend	1.289E-09	0	0
FEDTotal	3.69353E-11	0	0
GDP	7.9524E-12	0	0
P/E	-5.44031E-08	0	0
VIX	1.16258E-08	0	0
VRP	7.9867E-08	0	0
EVRP	8.53145E-08	0	0
ExpInf1Y	-1.64426E-11	0	0
ExpInf5Y	-1.45452E-11	0	0
ExpInf10Y	-1.24185E-11	0	0
ExpInf30Y	-7.70065E-12	0	0
Prod	4.76577E-12	0	0

Notes: The table presents the coefficients of the penalized regression models for the final prediction period. Please note that these coefficients update each month and should be interpreted with care.

Appendix 6. Cross-Validated Hyperparameters

Model	Hyperparameter	Value	Description
Ridge	Lambda*, λ	5.5	Constant that multiplies the penalty term*.
Lasso	Lambda, λ	1.25	Constant that multiplies the penalty term.
Elastic Net	Lambda, $\lambda_1 + \lambda_2$	3.1	Constant that multiplies the penalty terms.
	Alpha, α	0.41	The elastic net mixing parameter, with 0 <= α <= 1. For α = 0 the penalty is a ridge penalty. For α = 1 it is a lasso penalty. For 0 < α < 1, the penalty is a combination of lasso and ridge.
K-Nearest Neighbors	N-Neighbors	12	Number of neighbors to use for k-neighbors queries.
	Weights	Uniform	All points in each neighborhood are weighted equally.
Regression Trees	Splitter	Best	The strategy used to choose the split at each node. The 'Best' strategy chooses the best split in terms of MSE.
	Max Leaf Nodes	3	Max leaf nodes that a tree can reach by choosing the best split.
Random Forests	Max Depth	2	The maximum depth of each tree.
	Max Features	3	The number of features to consider when looking for the best split.
	N-Estimators	30	The number of trees in the random forest.
	Min Samples Split	0.0001	The minimum number of samples required to split an internal node.
Light GBM	Learning Rate	0.001	Controls the learning rate in the boosting process.
	Max Depth	3	Maximum tree depth for base learners.
	N-Estimators	160	Number of boosted trees to fit.
	N-Leaves	2	Maximum tree leaves for base learners.
ANN	Hidden Layer Size	(200, 200)	The number of neurons in the hidden layers respectively.
	Learning Rate	0.001	The initial learning rate. It controls the step-size in adjusting the weights.
	Tol	1.00E-04	Tolerance for the optimization
	Max Iterations	88	Maximum number of iterations that solver iterates until. This determines the number of epochs (how many times each data point will be used).
	N-Iterations to change	25	Maximum number of epochs.

Notes: The table presents the values of the optimized hyperparameters for each model. Ridge, lasso and elastic net ("ENet") are the penalized linear regressions applied. Light gradient boosting machines ("LGBM"), K-nearest-neighbors ("KNN"), regression trees ("RT"), random forest ("RF"), and artificial neural networks ("ANN") are the non-linear methods applied. * This is an alternative way of stating the value of λ, not to be interpreted as the constant described in Equation 11 of this paper.