Estimating Stock Market Betas via Machine Learning

Internet Appendix

Keywords: Beta estimation; machine learning; active trading strategy
JEL classification codes: G11, G12, C58, G17

Internet Appendix A: Benchmark estimators

In this appendix, we briefly present the established forecast models listed in Section IV.A of the main paper. We use them as our benchmarks to determine whether the machine learning-based approaches listed in Section IV.B of the main paper add incremental predictive power. We start with simple rolling-window estimators, continue with shrinkage-based approaches, and end with portfolio-based and long-memory models. We also note the major differences between the model families. Panel A of Table A1 summarizes the definitions and descriptions for each of the established beta estimators.

Rolling-window estimators

In the baseline rolling-window approach, we obtain historical betas by running time-series ordinary least squares (OLS) regressions:

$$r_{i,ts} = \alpha_{i,t}^{H} + \beta_{i,t}^{H} r_{M,ts} + \varepsilon_{i,ts}, \tag{A1}$$

where $r_{i,ts}$ and $r_{M,ts}$ are the excess returns on stock i and the market portfolio M, respectively. The subscript t indicates that we estimate historical alphas and betas ($\alpha_{i,t}^{H}$ and $\beta_{i,t}^{H}$, respectively) for each month t using a rolling window of daily or monthly excess returns. The subscript $s = 1, \dots, \tau$ denotes the returns before the end of month t, while τ refers to the length of the rolling window. The intercept $\alpha_{i,t}^{H}$ is the risk-adjusted return, while the slope $\beta_{i,t}^{H}$ is our parameter of interest. The error term $\varepsilon_{i,ts}$ is an idiosyncratic return shock.

Rolling-window beta estimators rely only on historical return information. Since there is no need to specify a set of conditioning variables (neither fundamental nor technical nor macroeconomic), these estimators are less prone to misspecification. However, they implicitly assume that betas are constant within the rolling window (and going forward), which leads to an important bias–variance trade-off between detecting short-term fluctuations in betas (conditionality) and accurately capturing long-term averages. Shorter rolling windows increase the ability to use short-term information, which reduces estimation bias. However, the resulting smaller rolling samples are more susceptible to microstructure noise, which increases both estimation variance and measurement error. Because of this trade-off, we consider two sets of rolling betas estimated from different window lengths and data frequencies. The first is obtained from rolling regressions using a five-year window of monthly returns (*ols_5y_m*), as proposed by Black, Jensen, and Scholes (1972) and Fama and MacBeth (1973). The second is obtained from rolling regressions using a one-year window of daily returns (*ols_1y_d*), as proposed by Andersen et al. (2006). Modifications to the basic rolling-window approach that help improve the bias–variance trade-off are numerous. In our empirical analysis, we focus on the two most commonly used approaches.

First, in the traditional OLS regression setting, observations entering the market model are equally weighted. However, from a conceptual perspective, changing the underlying weighting scheme allows more recent observations to be given greater weight. Thus, the estimates are "conditional", while still incorporating sufficiently large rolling samples. In line with Hollstein et al. (2019), we utilize rolling-window estimators with an exponentially-weighted moving average structure. To obtain *exponentially-weighted betas*, we run time-series weighted least squares (WLS) regressions using a one-year window of daily returns with exponential weights. These are defined by the time it takes for the weights to fall below half of their initial value, i.e., their half-life. A shorter (longer) time span thus reflects a faster (slower) exponential decay. To give an example, Figure A1 provides a visualization of exponentially decaying weights based on short and long half-lives. To be conservative, we consider two sets of rolling betas estimated from different horizons: one-third (*ewma_s*) and two-thirds (*ewma_l*) of the number of observations of the (initial) rolling window.

Second, the traditional OLS regression setting is particularly prone to outliers in a stock's return history, resulting in extreme and volatile beta estimates. To moderate the influence of single outlier returns, Welch (2022) assumes that market betas lie within the $(-2, +4)$ interval. In theory, this increases the signal-to-noise ratio, which translates into improved predictive power. To this end, stock-level returns must first be winsorized at market return-based bounds: $r_{i,ts} \in \left(-2r_{Mt,s}, +4r_{Mt,s}\right)$. A one-year window of these winsorized daily returns is then used within the baseline rolling-window approach to obtain *slope-winsorized betas* (*bsw*).

Shrinkage-based estimators

Another approach that aims to refine beta forecasts is to enhance rolling-beta estimates with additional cross-sectional information. The idea is that a stock's beta estimate should not be too different from those of other stocks with similar characteristics. Therefore, a prior on the true beta can be specified to which the sample beta estimate obtained from rolling regressions is shrunk. The established approach for obtaining *shrinkage betas* is to compute the weighted average of rolling-beta estimates $\beta_{i,t}^{H}$ and prior belief $\bar{\beta}_{i,t}$:

$$\tilde{\beta}_{i,t} = \varphi_{i,t}\beta_{i,t}^{H} + (1 - \varphi_{i,t})\bar{\beta}_{i,t}. \tag{A2}$$

The shrinkage weight is: $\varphi_{i,t} = \frac{\sigma_{\bar{\beta}_{i,t}}^{2}}{s_{\beta_{i,t}^{H}}^{2} + \sigma_{\bar{\beta}_{i,t}}^{2}}$, where $s_{\beta_{i,t}^{H}}^{2}$ and $\sigma_{\bar{\beta}_{i,t}}^{2}$are the variance of the sample estimates of beta and the prior, respectively. The degree of shrinkage is proportional to the relative precision of the rolling-beta estimates and the prior. The lower the relative precision of the sample estimates of beta (i.e., the larger $s_{\beta_{i,t}^{H}}^{2}$ is relative to $\sigma_{\bar{\beta}_{i,t}}^{2}$), the more weight is given to the prior. Conceptually, shrinking towards a well-defined prior reduces estimation noise, which helps improve the accuracy of the rolling-beta estimates. Prior beliefs thereby can be specified in several ways.

Vasicek (1973) suggests that, if no other information is known about a stock except that it comes from a broad universe, the optimal prior density for the true underlying beta is based on the cross-sectional distribution of the beta. Thus, the *value-weighted* mean and variance of rolling betas within the cross-section are used as prior information. Karolyi (1992) suggests grouping stocks into portfolios based on firm fundamentals and shrinking rolling-beta estimates towards their portfolio betas. Specifically, the *value-weighted* mean and variance of rolling betas within each industry portfolio are used as prior information.

However, Cosemans et al. (2016) argue that shrinkage based on Vasicek (1973) and Karolyi (1992) only dampens part of the noise in rolling-beta estimates. This is because the prior does not use the cross-sectional information embedded in firm fundamentals at all (Vasicek, 1973) or may be hampered by large intra-portfolio dispersion in betas (Karolyi, 1992). They suggest specifying priors unique to each firm that incorporate a comprehensive set of firm fundamentals as predictors. In particular, they outline a complex Bayesian framework (which they call a “hybrid model”) for computing firm-specific priors.[1]

In our empirical analysis, we implement the shrinkage approach as follows: We obtain rolling-beta estimates that contribute to each of the three shrinkage betas from rolling regressions using a one-year window of daily returns.[2] Prior information for the Vasicek (1973) and Karolyi (1992) beta estimates are obtained by considering the entire cross-section (*vasicek*) and, analogously to Cosemans et al. (2016), by creating 47 industry portfolios (*karolyi*) according to Fama and French’s (1997) industry

[1] The parameters of the hybrid model are estimated via Markov Chain Monte Carlo methods (Cosemans et al., 2016).

[2] While Cosemans et al. (2016) obtain the sample betas that contribute to the hybrid estimates of beta from a rolling regression using a *half-year* window of daily returns, we opt for a *one-year* window. First, although not the main objective of our empirical analysis, using the same rolling-window length for each (shrinkage) beta is the only way to compare their predictive performance consistently. It allows us to assess whether differences in predictive performance truly stem from differences in prior information, rather than from differences in rolling-window beta estimates. Second, and more importantly, in our empirical setting, we find that the predictive performance of the hybrid beta estimates is better for the one-year window (compared to the half-year window). This makes it an even more conservative benchmark for identifying the value-added of machine learning techniques.

classification. Consistent with Cosemans et al. (2016), the prior information for the hybrid beta estimates is based on the conditioning variables size, book-to-market ratio, financial leverage, operating leverage, momentum, and default spread (*hybrid*).[3]

Portfolio-based estimators

Fama and French (1992) take a different approach to estimating time-varying market betas. They first sort individual stocks into portfolios, then estimate rolling betas for each portfolio, and finally assign portfolio betas to individual stocks. In our empirical analysis, we implement their approach as follows: At the end of each month t, we sort the stocks into size deciles based on NYSE breakpoints. Each size decile is partitioned into ten portfolios based on sample estimates of beta obtained from rolling regressions using a one-year window of daily returns. Equal-weighted daily returns are computed for each of the resulting 100 size–beta portfolios over the next month. Finally, we obtain the *portfolio betas* from rolling regressions using a one-year window of daily post-ranking portfolio returns.[4] These beta estimates are assigned to the individual stocks in each of the 100 size–beta portfolios (*fama-french*).

Long-memory estimators

Rather than shrinking rolling-beta estimates to prior beliefs or assigning rolling portfolio betas to individual stocks, Becker et al. (2021) focus on the time-series properties of realized betas. They

[3] Our implementation differs slightly from the original Cosemans et al. (2016) shrinkage approach in two ways: First, we opt for the Novy-Marx (2011) definition of operating leverage (see Footnote 6 of the main paper), and for a one-year window of daily returns to compute sample estimates of beta obtained from rolling regressions.

[4] Fama and French (1992) estimate the pre- and post-ranking betas using monthly returns and construct the size–beta portfolios on an annual basis. However, Cosemans et al. (2016) and Hollstein et al. (2019) find that rolling-window beta estimates computed from daily returns are more accurate predictors of future betas than those computed from monthly returns. This is why we estimate the betas using a one-year window of daily returns. Furthermore, to always incorporate the most recent data, we construct the size–beta portfolios on a monthly basis.

find that the degree of memory within a beta time series, i.e., the order of integration d (typically with $0 \leq d \leq 1$), is the key determinant to modeling beta dynamics. A larger d thereby indicates a longer memory, and vice versa. Becker et al. (2021) find that beta time series clearly exhibit long-memory properties ($0 < d < 1$). In this case, the current value of a variable depends on past shocks, but the less so the further in the past these shocks are. In other words, past shocks neither die out quickly nor persist infinitely, but have a hyperbolical decaying effect. In our empirical analysis, we adapt their long-memory approach (*long-memo*) by implementing a $FI(0.4)$ model, i.e., a fractionally integrated time-series process with $d = 0.4$.

Table A1
Details on forecast models

This table summarizes the definitions and descriptions for each established beta estimator listed in Section IV.A of the main paper (Panel A) and provides the definitions and hyperparameter specifications for each machine learning-based beta estimator listed in Section IV.B of the main paper (Panel B).

Panel A: Benchmark estimators

Model	Description	Definition
ols_5y_m	Historical beta	Rolling regressions using a five-year window of monthly returns
ols_1y_d	Historical beta	Rolling regressions using a one-year window of daily returns
ewma_s	Exponentially-weighted beta	Rolling regressions using a one-year window of daily returns with exponentially decaying weights (short half-life)
ewma_l	Exponentially-weighted beta	Rolling regressions using a one-year window of daily returns with exponentially decaying weights (long half-life)
bsw	Slope-winsorized beta	Rolling regressions using a one-year window of *winsorized* daily returns
vasicek	Shrinkage beta	Shrinkage of *ols_1y_d* towards average beta within stock universe
karolyi	Shrinkage beta	Shrinkage of *ols_1y_d* towards average beta within industry portfolio
hybrid	Shrinkage beta	Shrinkage of *ols_1y_d* towards firm-specific beta prior
fama-french	Portfolio beta	Assignment of portfolio betas (rolling regressions using a one-year window of daily post-ranking portfolio returns) to individual stocks
long-memo	Long-memory beta	Application of fractionally integrated long-memory time-series process

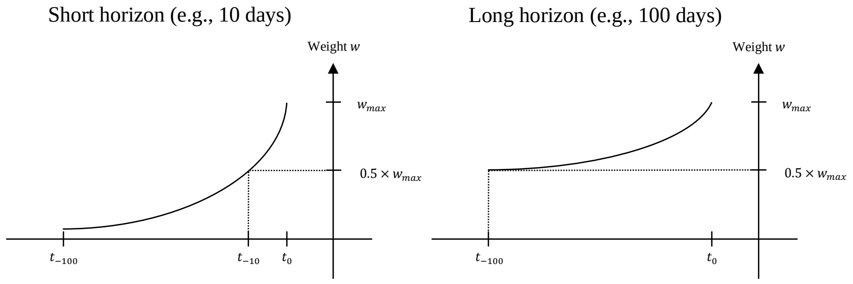
Continued on the next page

Panel B: Machine learning estimators

Model	Hyperparameter	Specification	Definition
lm			
	None		
elanet			
	λ	(0,1)	General strength of the penalization
	p	{0,0.5,1}	Weight on the lasso and ridge penalization
rf			
	L	(1,10)	Depth of the single regression trees
	M	{20,25,30,35,40}	Number of predictors randomly considered as potential split variables
	B	(10,500)	Number of trees added to the ensemble prediction
gbrt			
	L	(1,5)	Depth of the single regression trees
	ν	{0.01,0.05,0.1}	Weight for the learning rate shrinkage
	B	(10,500)	Number of trees added to the ensemble prediction
nn_1–nn_5			
	batch size	1000	Batch size
	n_{epochs}	100	Number of epochs
	patience	25	Number of iterations during which the value-weighted mean squared error is allowed to increase in the validation sample
	dropout rate	0.1	Fractional rate of input variables that are randomly set to zero at each iteration
	n_{seeds}	10	Number of independent seeds used for each specification family

Figure A1
Stylized visualization | Exponentially decaying weights

This figure depicts a stylized visualization that helps explain the concept of exponentially decaying weights based on short and long half-lives (all observations' weights sum to one).


Short horizon (e.g., 10 days)

Long horizon (e.g., 100 days)

Internet Appendix B: Machine learning estimators

In this appendix, we briefly introduce the representative set of machine learning techniques listed in Section IV.B of the main paper. We start with linear regressions, continue with tree-based models, and end up with neural networks, all of which aim to minimize the *value-weighted* mean squared error (MSE). We also note the major differences between the model families. Panel B of Table A1 provides the definitions and hyperparameter specifications for each machine learning-based beta estimator.

Linear regressions

Ordinary least squares regressions (*lm*) are the least complex approach in our empirical analysis. At each re-estimation date, we use the training sample to run pooled OLS regressions of future realized betas $\beta^R_{i,t+k}$ on the set of 81 predictors.[5,6] In line with Petkova and Zhang (2005), who argue that the relationship between firm characteristics and beta varies over the business cycle, we divide this set of predictors into firm characteristics $z_{i,t}$ (including industry dummies) and the default spread x_t, which we choose as an indicator of the state of the economy (Jagannathan and Wang, 1996). We follow Cosemans et al. (2016) and include interactions between firm characteristics and the default spread in the regression model:[7]

[5] While simple linear regressions do not require parameter tuning (based on the validation sample), we also estimate this model from only the training sample. This enhances the comparability with the machine learning-based forecast models (penalized linear regressions, tree-based models, and neural networks). Note that the main findings of our empirical analysis are qualitatively similar when pooling the training and validation samples together to estimate the simple linear regression model at each re-estimation date.

[6] Focusing on a similar type of forecast objective (stock-level expected returns), Lewellen (2015) and Drobetz et al. (2019) show that this approach is promising in capturing the cross-sectional variation in the dependent variable from both a statistical and an economic perspective. Although Drobetz et al. (2019) use cross-sectional Fama and MacBeth (1973) regressions (FM regressions) that are re-estimated on a monthly basis, Drobetz and Otto (2021) show that the OLS-based model provides nearly identical predictions in a sample-splitting and re-estimation setting similar to ours. To ensure comparability with the machine learning models that cannot be re-estimated on a monthly basis (due to computational limitations), we use pooled OLS regressions as a proxy for the linear FM regressions approach.

[7] For the sake of parsimony, we do not include interactions between industry dummies and the default spread.

$$\beta_{i,t+k}^{R} = \delta_0 + \delta_1 x_t + \delta_2' z_{i,t} + \delta_3' x_t z_{i,t} + \varepsilon_{i,t+k}. \quad \text{(B1)}$$

Incorporating such interactions may add additional explanatory power, but it also increases the number of predictors, which leads to a high-dimensionality problem when the number of predictors becomes very large relative to the number of observations. Particularly in a forecasting context, because the convexity of the OLS objective tends to emphasize heavy-tailed observations, as the number of predictors increases, simple linear regression models begin to overfit noise rather than extract signal, thereby undermining the stability of the predictions.[8]

The most common machine learning technique to overcome the overfitting problem in a high-dimensional regression setting is ***penalized least squares regression***. This approach helps identify which predictors are informative and omit those that are not. It modifies the OLS loss function by adding a penalty term $\Phi(\theta)$ to favor more parsimonious model specifications:

$$l(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(\beta_{i,t+k}^{R} - g\left(z_{i,t};\theta\right)\right)^2 + \Phi(\theta). \quad \text{(B2)}$$

We use the *elastic net* approach (*elanet*), which combines the lasso and ridge methods.[9] It computes the weighted sum of the lasso and ridge penalties to increase flexibility:

$$\Phi(\theta) = \lambda(1-p)\sum_{j=1}^{P}\left|\theta_j\right| + \lambda p \sum_{j=1}^{P}\left(\theta_j\right)^2. \quad \text{(B3)}$$

The tuning parameters in this forecast model are $\lambda \in (0,1)$ and $p \in (0,1)$. λ indicates the strength of the penalty (in particular, how strongly the regression coefficients are forced to zero); p indicates the relative weights of the lasso and ridge approaches. $p = 0$ corresponds to lasso; and $p = 1$ corresponds to ridge.

[8] Note that the main results of our empirical analysis are qualitatively similar when using the WLS loss function (based on market capitalization-based weights of the stocks).

[9] The lasso approach penalizes the sum of absolute coefficients, thereby setting the regression coefficients of a subset of predictors to exactly zero (variable selection). The ridge approach penalizes the sum of squared coefficients, thereby only pushing regression coefficients close to zero (variable shrinkage). We also test the lasso and ridge methodologies separately. We find no improvement in predictive performance relative to the elastic net approach, so we do not report results for these penalty functions.

Importantly, unless explicitly included as *predetermined* terms, pooled regressions cannot capture any nonlinear or interactive effects (neither simple nor penalized approaches). Thus, we use linear regressions as a benchmark to identify whether such effects, in addition to the two-way interaction between firm characteristics and the default spread, lead to incremental predictive power. Note that both tree-based models and neural networks inherently incorporate nonlinearity and multi-way interactions, without the need to add new predictors to capture these effects in advance.

Tree-based models

The idea behind ***tree-based models*** is that they adaptively partition the dataset into groups of observations that behave in a similar way. They follow an iteration process that is inspired by the growing behavior of real trees in nature (see Figure B1): First, the process starts with an initial node, the root, in order to find the optimal split variable and the optimal split value for it by minimizing the value-weighted MSE within each partition. This results in two nodes with minimized impurity. Second, to further disentangle the dataset, the algorithm determines optimal split variables and values on the subsamples left over from the previous step(s) to iteratively grow the regression tree. This results in multiple final nodes with minimized impurity, the leaves. The predicted beta for each leaf reflects the simple average of the historically realized betas for the firms sorted into that leaf. Regression trees are invariant to monotonic transformations of predictors, can incorporate categorical and numerical data in the same forecast models, and are designed to inherently capture nonlinearity and multi-way interactions. However, they are prone to overfitting and need to be strongly regularized. To accomplish this, the ensemble forecast approach aggregates forecasts from many different regression trees into a single one. According to Gu et al. (2020), there are two common methods: bagging and boosting.

Random forests (*rf*) modify Breiman's (2001) traditional bagging approach. The idea is to take multiple bootstrap samples of the original dataset, fit deep trees independently, and then average their predictions into an ensemble prediction to create a single strong learner. Because dominant predictors are always more likely to become split variables at low levels, which can lead to large correlations between bootstrap-replicated trees, random forests use the so-called "dropout" method. At each potential branch, they randomly drop out predictors, leaving only a subset of predictors to be selected as potential split variables. The tuning parameters in this forecast model are the depth of the trees L, the number of predictors M randomly considered as potential split variables, and the number of trees B added to the ensemble prediction.

In contrast, *gradient boosted regression trees* (*gbrt*) follow the boosting approach, which is based on the idea that combining multiple shallow trees creates a single strong learner, even stronger than a single deep tree. The iterative procedure is as follows: It computes a first shallow tree to fit the realized betas. This oversimplified tree exhibits a high forecast error. Next, it computes a second shallow tree, fitting the forecast residuals from the first tree. The forecasts from these two trees are then added together to form an ensemble prediction. To avoid overfitting the forecast residuals, the forecast component from the second tree must be shrunk by a factor $\nu \in (0,1)$. Each additional shallow tree fits the forecast residual from the preceding ensemble prediction, and its shrunk forecast component is added to the ensemble forecast accordingly. The tuning parameters in this forecast model are the depth of the trees L, the shrinkage weight ν, and the number of trees B added to ensemble prediction.

Neural networks

Neural networks (*nn*) are the most complex method in our empirical analysis. They are highly parameterized, which makes them suitable for solving complicated machine learning problems. However, they are opaque and can be difficult to interpret. In general, they map inputs (predictors) to outputs (realized betas). Inspired by the way the human brain works, they consist of many interconnected computational units, so-called “neurons”. Each neuron alone provides very little predictive power, but a network of multiple neurons functions cohesively and improves the predictive performance. We use feedforward neural networks, where each node has a connection to all nodes in the previous layer and the connections follow a one-way direction (from the input layer to the output layer). The input layer contains the predictor variables (e.g., lagged firm characteristics), while the output layer contains a prediction for the dependent variable (realized betas). The simplest neural network (with no hidden layers) corresponds to the OLS regression model. The addition of hidden layers leads from shallow to deep architectures that can capture nonlinear and interactive effects (see Drobetz and Otto, 2021, for stylized visualizations that help explain the structure, operation, and regularization of neural networks).

Neural networks predict the output y as the weighted average of inputs x. In the simplest model, the regression coefficients are used as weights. In more complex architectures, the weights must be computed iteratively by using the “backpropagation” algorithm. This algorithm initializes each connection with random weights. It also calculates the initial value-weighted MSE based on the predictions derived from the inputs of the (last) hidden layer. It then proceeds iteratively as follows: First, it recursively (from the output layer to the input layer) computes the gradient of the value-weighted MSE with respect to the weights. Second, it adjusts the weights slightly in the *opposite* direction of the computed gradients, since the goal is to *minimize* the value-weighted MSE. Third, it recalculates the value-weighted MSE based on the adjusted weights. The iteration process, called “gradient descent”, stops when the value-weighted MSE is finally minimized.

So far, it is assumed that each node in the hidden layer produces a signal (i.e., it is included in the weighted average computation). In the human brain, however, neural networks work somewhat differently. To avoid noise, a given node transforms each of the previous signals it transmits (if at all).

For example, it may amplify or condense the previous signals, or only generate a signal if the accumulated impulse is strong enough. Thus, at each node, the weighted average of the preceding signals (x, coming from either the input or the preceding layer) is subject to an activation function.[10] Following Gu et al. (2020), we choose the rectified linear unit (ReLU) activation function and apply it to each node in the hidden layers. To promote sparsity in the number of active neurons, it only provides a signal when the information from the previous layer accumulates beyond a given threshold: $ReLU(x) = \begin{cases} 0 \text{ if } x < 0 \\ x \text{ otherwise} \end{cases}$.

In our empirical analysis, we consider neural networks with up to five hidden layers ($HL = 5$) and thirty-two neurons ($N = 32$), which we choose according to the geometric pyramid rule (Masters, 1993). [11] Following Gu et al. (2020) and Drobetz and Otto (2021), we apply several different types of regularization simultaneously to ensure computational feasibility and to avoid overfitting.

First, in addition to a ReLU activation and a lasso-based penalization of the weights, we use the *stochastic gradient descent* (SGD) approach to train the neural networks. During the iteration process, the algorithm divides the training sample into small random subsamples, so-called "batches", and uses one at each iteration. This leads to strong improvements in computational speed. The algorithm still sees the entire training sample (sequentially, not concurrently, and at least once, but usually several times), which helps to incorporate all available information, and thus avoids degrading the predictive performance. Consequently, the number of iterations depends on the size of the batches and the number of epochs (i.e., the number of times the algorithm sees the entire training sample).

[10] While all weight transformations in the different nodes are purely linear, it is the activation function that allows neural networks to capture nonlinearity.

[11] The pre-specified neural network architectures are: *nn_1* ($HL = 1; N = \{32\}$), *nn_2* ($HL = 2; N = \{32,16\}$), *nn_3* ($HL = 3; N = \{32,16,8\}$), *nn_4* ($HL = 4; N = \{32,16,8,4\}$), and *nn_5* ($HL = 5; N = \{32,16,8,4,2\}$). Neural networks are computationally intensive and can be specified in a myriad of different architectures. Therefore, we refrain from tuning their parameters (e.g., batch size or number of epochs) and instead pre-specify five different models. We assume that our *nn_1–nn_5* architectures are a conservative lower bound for the predictive performance of neural network models.

Second, we employ the *batch normalization* algorithm introduced by Ioffe and Szegedy (2015). This aims to mitigate the internal covariate shift that occurs when the distribution of inputs to each hidden layer changes during training (as the parameters of the previous layers change), slowing down the learning process. To do this, it cross-normalizes the input to each hidden layer within each batch.

Third, we apply *learning rate shrinkage*. The learning rate determines the size of the incremental steps in the gradient, while iteratively minimizing the value-weighted MSE. There is a trade-off between finding the global minimum instead of its local counterpart (smaller learning rate) and computational speed (larger learning rate). This regularization procedure starts with a larger learning rate to speed up the computation. As the gradient approaches zero, it shrinks the learning rate toward zero to overcome a potential local minimum.

Fourth, we implement *early stopping*, since neural networks aim to minimize the value-weighted MSE in the training sample. This regularization stops the SGD iteration process when the value-weighted MSE in the validation sample increases for a pre-specified number of iterations, called "patience", which also speeds up the computation.

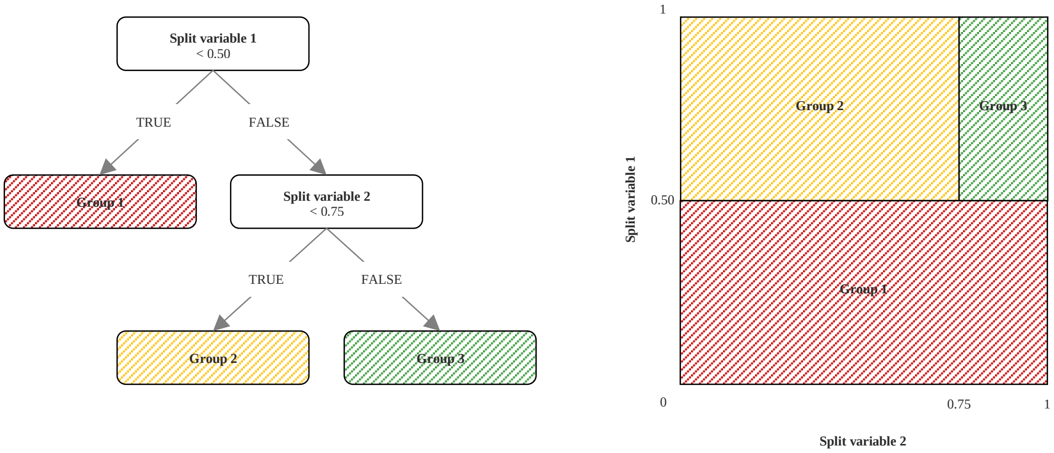
Fifth, we adopt the *ensemble* approach proposed by Hansen and Salamon (1990) and Dietterich (2000). We compute ten neural networks from the same specification family at each re-estimation date, using independent seeds.[12] We then average over the predictions to increase the signal-to-noise ratio, since the stochastic nature of the SGD approach leads to different forecasts for different seeds.

Finally, in addition to the regularization applied by Gu et al. (2020), we use the *dropout* method. This randomly sets a fraction of the input variables to exactly zero at each iteration, and is thus one of the most effective methods in the neural network framework to prevent overfitting.

[12] Seeds are numbers used to initialize random processes, which ensures different but reproducible predictions.

Figure B1
Stylized visualization | Regression trees

This figure depicts a stylized visualization that helps explain the structure and functioning of regression trees (as in Gu et al., 2020).



Internet Appendix C: Further analyses and robustness tests

Cross-Sectional and Time-Series Properties of Beta Estimates

First, we examine the properties of the beta estimates obtained from the different forecasting models. Panel A of Table C1 focuses on the cross-sectional summary statistics. The cross-sectional mean is close to one for all beta estimators, while the cross-sectional dispersion varies widely across the forecasting models. The rolling window estimators have the largest cross-sectional standard deviations. The other benchmark models produce lower cross-sectional dispersion. The machine learning-based beta estimators have the smallest cross-sectional standard deviations and lead to the least extreme beta estimates. The implied standard deviations are also lowest for the machine learning estimators, providing a first indication that they may provide more accurate estimates.[13]

Panel B of Table C1 focuses on the time-series summary statistics.[14] These are consistent with the cross-sectional metrics. The machine learning estimators yield the lowest time-series standard deviations, while the high volatility of the historical betas likely reflects measurement noise. Despite the inclusion of slow-moving firm fundamentals as predictors, the average time-series autocorrelations of the machine learning-based models are rather low compared to those of the benchmark approaches. Nevertheless, they are above 0.90 for all approaches.

Forecast errors

In this section, we present and discuss the results of further analyses and several robustness tests related to the forecast errors of the models. As a first step, we use the Giacomini and White (2006)

[13] The implied cross-sectional standard deviation of the true betas is computed following Pastor and Stambaugh (1999): $\widehat{Std}(\beta^R) = \left[\overline{Var(\beta^F)} - \overline{\widehat{Var}_{\beta_i^R}}\right]^{1/2}$. It is the square root of the difference between the time-series average of the monthly value-weighted cross-sectional variances and the value-weighted cross-sectional average of each firm's sampling variance. Small gaps between observed and implied standard deviations indicate small estimation errors.

[14] Following Becker et al. (2021), we omit firms with less than fifty beta estimates to allow for valid inference.

test to assess the relative *conditional* predictive performance of each model in pairwise comparisons. To do this, we use the mean squared forecast error differentials $d_t^{(j,l)} = \mathrm{M}SE_{t+k|t}^{(\mathrm{j})} - \mathrm{M}SE_{t+k|t}^{(l)}$ and regress them on a constant and the lagged forecast error differential, $\delta_t = \left(1\ d_t^{(j,l)}\right)'$. The test statistic is $GW = T\left(T^{-1}\sum_{t=1}^{T}\delta_{t-1}\,d_t^{(j,l)}\right)'\,\widehat{\Omega}^{-1}\left(T^{-1}\sum_{t=1}^{T}\delta_{t-1}\,d_t^{(j,l)}\right)$, where $\widehat{\Omega}$ is the Newey and West (1987) covariance matrix of $\delta_{t-1}d_t^{(j,l)}$. The GW test statistic is distributed as χ^2 with 2 degrees of freedom.

The results are reported in Table C2. Consistent with the results in the main paper, we find that the machine learning-based beta estimators also outperform all other methods in terms of their conditional predictive ability. The Giacomini and White (2006) GW test rejects its null hypothesis when comparing the random forests and neural networks with each benchmark model.

In Table C3, we examine the robustness of our main results to changes in the specifications of the machine learning estimators considered in the empirical analysis. First, we report additional results for the neural network architectures with two to five hidden layers (*nn_2–nn_5*). We find that these more complex models perform similarly to our baseline neural network architecture with only one hidden layer (*nn_1*). However, the average forecast errors are slightly higher, indicating that the less complex specification tends to perform slightly better. This is consistent with findings in Gu et al. (2020) and Drobetz and Otto (2021), and is likely a result of (moderate) overfitting.

Second, we test the robustness of our main results to the inclusion of additional macroeconomic variables in addition to the default spread (*dfy*). Following Welch and Goyal (2008), we add the U.S. T-bill rate (*tbl*), the U.S. T-bill rate volatility (*tbl_sd*), the term spread (*tms*), the stock variance (*svar*), the earnings-to-price ratio (*ep*) and the dividend payout ratio (*dp*), both at the market level, and the consumption–wealth ratio (*cay*). In addition, we include measures of industrial production, inflation, and unemployment as provided by the Federal Reserve Bank of St. Louis. Given our findings of the low variable importance for *dfy*, we expect that additional macroeconomic variables make little difference. Indeed, the results in Table C3 for random forests with additional macroeconomic variables (*rf_amv*) are qualitatively similar. While additional macroeconomic variables do not help improve the predictive performance, their inclusion does not hurt much either.

Third, we report the results for ensemble predictions, where we average the predictions of the *lm*, *elanet*, *rf*, *gbrt*, and *nn_1* models (*ens_1*), the *elanet*, *rf*, *gbrt*, and *nn_1* models (*ens_2*), and the *rf*, *gbrt*, and *nn_1* models (*ens_3*). The ensemble approaches perform quite well, with *ens_2* and *ens_3* producing lower average MSEs than *rf*. Including only those machine learning techniques that perform best in isolation, i.e., the *rf*, *gbrt*, and *nn_1* models, leads to the best ensemble prediction. One caveat is that applying this ensemble approach in practice is computationally intensive, as one must first estimate each of the three to five prediction models separately.

Next, we examine the robustness of our main results to changes in the forecast error measure. In particular, we use the *equal-weighted* mean squared error (MSE) in Table C4:

$$MSE_{t+k|t} = \frac{1}{N}\sum_{i=1}^{N_t}(\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F})^2, \text{ with } k = 12, \tag{C1}$$

and the *value-weighted* mean absolute error (MAE) in Table C5:

$$MAE_{t+k|t} = \sum_{i=1}^{N_t} w_{i,t}|\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F}|, \text{ with } k = 12, \tag{C2}$$

We find that the equal-weighted MSEs are notably higher for all approaches than for the value-weighted examination. This is consistent with our previous finding that it is considerably more difficult to estimate market betas for small stocks than it is for large stocks. Moreover, the machine learning-based approaches outperform the benchmark models even more when equally weighted, supporting our previous finding that they are particularly beneficial for small stocks (see Figure 3 of the main paper). In the MAE framework, all forecast errors are penalized in the same way. As a result, large forecast errors have less impact than in the MSE framework. Nevertheless, the machine learning-based approaches still outperform the benchmark models using the MAE, suggesting that the differences in predictive performance are not predominantly driven by large outliers in the forecast errors for just a few stocks.

We also use Mincer and Zarnowitz (1969) regressions in Table C6 to test for the unbiasedness of the different forecasting models. Following Fama and MacBeth (1973), we run either a weighted least squares (WLS) regression (using stock market capitalization-based weights) or an ordinary least squares (OLS) regression (using equal weights) of realized betas on the beta estimates obtained from the different forecasting models at the end of each month t: $\beta^R_{i,t+k} = a_t + b_t\beta^F_{i,t+k|t} + e_{i,t+k}$. Table C6 reports the time-series averages of the monthly intercepts (a), the slopes (b), and the *t*-statistics (in parentheses) testing the null hypotheses that $a = 0$ and $b = 1$, respectively. For the *t*-tests, we use Newey and West (1987) standard errors with eleven lags. Consistent with our previous results, we find that the best-performing machine learning models are also the least biased. For all machine learning techniques, the average intercept is closer to zero and the slope is closer to one (with mostly insignificant *t*-statistics). In contrast, in the vast majority of cases, the significant *t*-statistics indicate a rejection of the unbiasedness hypotheses for the benchmark models.

Finally, we examine different forecast horizons and sampling frequencies for the realized beta. In particular, we alternatively consider forecast horizons of three months and six months. For both, we continue to use daily data to compute the realized beta. We also consider a 12-month forecast horizon with weekly data and a 60-month forecast horizon with monthly data. The results are presented in Table C7.[15] For each combination of forecast horizon and valuation metric, we find that the

[15] Note that we skip the *hybrid* and *nn_1* models for this analysis. This is because both of them are prohibitively computationally expensive. Furthermore, as the main analysis shows, the *hybrid* model is generally not the best benchmark, and the *nn_1* model is generally not the best machine learning-based model. Thus, the loss of information from omitting these two models appears to be limited.

machine learning-based estimators perform best, with the *rf* model generally producing the lowest forecast or hedging errors.

Minimum variance portfolios

We examine the robustness of the MVP results to the use of subsample periods. Successful MVPs should perform well on a period-by-period basis. Therefore, we split the sample in two and examine the MVPs in the first and second halves separately. We present the results in Tables C8 and C9. Consistent with the main analysis, we find that the machine learning-based approaches clearly outperform the benchmarks for both halves of the sample period.

Anomaly performance

In the main paper, we show that machine learning techniques lead to betting-against-beta (BAB) portfolios that are truly market neutral ex-post. Novy-Marx and Velikov (2022) argue that in addition to *unconditional* market neutrality, *conditional* market neutrality is also important. To examine this, we follow their approach and run time-series regressions of monthly BAB portfolio returns on a constant, the current market excess return interacted with the log of the one-year to five-year market volatility ratio, the current market excess return, and the lagged market excess returns of the previous two months:

$$r_{BAB,t} = \alpha_{BAB} + \beta_1 r_{M,t} \ln\left(\sigma_1^{MKT}/\sigma_5^{MKT}\right) + \beta_2 r_{M,t} + \beta_3 r_{M,t-1} + \beta_4 r_{M,t-2} + \varepsilon_{i,t}, \tag{C3}$$

where σ_1^{MKT} and σ_5^{MKT} are the past one-year and five-year estimates of the volatility of the (daily) market excess return, lagged by one month. Following Novy-Marx and Velikov (2022), we standardize the volatility ratio to have a mean of zero and a standard deviation of one. All other variables are defined as before.

Novy-Marx and Velikov (2022) show that the original BAB portfolio of Frazzini and Pedersen (2014), but also their improved version of it, is significantly related to $\beta_1 r_{M,t}\ln(\sigma_1^{MKT}/\sigma_5^{MKT})$. Thus, the standard BAB strategy has a low exposure to the market during periods of high volatility and a high exposure during periods of low volatility, introducing a market-timing element that positively affects the performance.

The results are presented in Table C10. These results confirm that all machine learning-based betting-against-beta strategies are also conditionally market neutral. All coefficient estimates from Equation (C3) are insignificant at the 5% level. For the benchmark estimators, however, the β_2 coefficient estimates are all statistically significant.

Nonlinearity and interactions

The results in the main paper suggest that tree-based models and neural networks are superior to established beta estimators. Both machine learning-based model families are designed to capture nonlinearity and interactions in the relationship between predictors and future market betas. Importantly, they also outperform linear regressions that include the exact same set of covariates. As a result, much of this outperformance may be due to their ability to exploit nonlinear and interactive patterns in estimating future market betas. We therefore investigate whether the best-performing machine learning approach, random forests (*rf*), actually captures nonlinearity and interactions. For comparison, we contrast the results with beta estimates from simple linear regressions (*lm*).[16]

We first examine the marginal association between a single predictor and its beta estimates ($\beta_{i,t+k|t}^{F}$, with $k = 12$). To illustrate, we select a firm's sample beta estimate from rolling regressions

[16] Note that the patterns identified and their implications are qualitatively similar when comparing gradient boosted regression trees (*gbrt*) and neural networks (*nn_1*) to estimates obtained from penalized linear regressions (*elanet*). This underscores that the ability to exploit nonlinear and interactive patterns does indeed lead to the outperformance of tree-based models and neural networks over linear regressions.

using a one-year window of daily returns (*ols_1y_d*). It is the most influential predictor in our empirical analysis (see Figure 5, Panel B of the main paper) and helps to address the problem of underestimation and overestimation inherent in estimating time-varying market betas (see Figure 2 of the main paper).[17] To visualize the average effect of *ols_1y_d* on $\beta^F_{i,t+k|t}$, we set all predictors to their uninformative median values within the training sample at each re-estimation date, and the industry dummies to zero. We then vary *ols_1y_d* over the interval $(-1, +3)$ and compute the beta estimates. Finally, we average the beta estimates over all re-estimation dates.

Panel A of Figure C1 illustrates the marginal association between *ols_1y_d* and $\beta^F_{i,t+k|t}$. To this visualization we add a histogram showing the historical distribution of *ols_1y_d*. This allows us to assess the relevance of the differences in the predictions obtained from the *lm* and *rf* models to the overall forecast results. As expected, higher values of the one-year rolling betas lead to higher beta estimates for both model families. We observe an increasing linear relationship between *ols_1y_d* and $\beta^F_{i,t+k|t}$ for the *lm* model. In the center of the distribution, approximately in the interval $(+0.3, +1.5)$, the marginal association between *ols_1y_d* and the beta forecasts is also nearly linear for the *rf* model. Outside this interval, however, the *rf* model provides nearly constant predictions, resulting in an overall S-shaped relationship. In contrast, the *lm* model, by construction, must adhere to the increasing linear relationship. This results in less extreme beta estimates for random forests (compared to simple linear regressions) as *ols_1y_d* becomes small or large. Since a substantial fraction of the observations lies within these outer regions of the historical distribution, differences in predictions are highly relevant. This highlights the need to account for nonlinear effects of the predictor variables. We also observe such S-shaped relationships for other predictors (unreported), such as turnover (*to*) and size

[17] Note that the patterns identified and their implications are qualitatively similar for other predictor variables.

(*me*). Taken together, these visualizations provide an explanation for our previous findings that random forests generally provide less extreme beta forecasts while avoiding the systematic underestimation of low-beta stocks and the systematic overestimation of high-beta stocks (see Figure 2 of the main paper). These results also help to explain the outperformance of random forests over established and linear approaches in terms of lower forecast errors (see Tables 2 and 3 of the main paper) and their dominance over benchmark estimators in the construction of market-neutral portfolios (see Tables 4 and 5 of the main paper).

Next, we examine the interactions between predictors in estimating future market betas, again using *ols_1y_d* as our baseline covariate. We select *me*, another highly influential predictor in our empirical analysis (see Figure 5, Panel B of the main paper), as our interactive counterpart and repeat the procedure outlined above. In this case, however, we compute the beta estimates for different levels of *me* over the interval $(-1, +1)$. The interactive effect between *ols_1y_d* and *me* on $\beta_{i,t+k|t}^{F}$ is shown in Panel B of Figure C1. Low and high levels for *me* are marked with red and green lines, respectively. Conceptually, if there is no interaction, or if the model is unable to capture such interactions, calculating estimated betas for different levels of *me* simply shifts the lines up or down in parallel. In this case, the distance between the lines is identical for any given value of *ols_1y_d*. This pattern is apparent for simple linear regressions because no pre-specified interaction term, e.g., *ols_1y_d* × *me* for the interaction between *ols_1y_d* and *me*, is included as a predictor in the OLS-based framework. The lines are shifted upward as *me* increases, indicating that an increase in *me* also increases $\beta_{i,t+k|t}^{F}$, but independently of *ols_1y_d*. Unlike the *lm* model, the *rf* model uncovers the interactive effect between a firm's historical beta and size in estimating future betas.[18] While the lines are also shifted upward

[18] Although somewhat less pronounced, the *rf* model also reveals the interactive effects between other firm characteristics in estimating future betas, such as between a firm's historical beta (*ols_1y_d*) and turnover (*to*).

for higher levels of *me*, the strength of the shift is much more pronounced for larger values of *ols_1y_d*. Thus, the effect of a firm's size on its future beta estimate appears to be much stronger if the firm has historically been more sensitive to systematic market risk.

As this example shows, incorporating nonlinear effects of individual predictors and interactions between predictors is both essential and fundamental to the superior predictive performance of random forests. These effects explain the advantage of machine learning methods over established and linear benchmark models.

Size and value beta forecasting

Finally, we extend the beta forecasting analysis to the size and value factors of Fama and French (1993). That is, we adapt our estimation and evaluation procedure to a three-factor setup. We compute historical size (SMB) and value (HML) betas from multiple regressions including all three Fama and French (1993) factors:[19]

$$r_{i,ts} = \alpha_{i,t}^{H} + \beta_{i,t}^{H} r_{M,ts} + \beta_{i,t}^{S} r_{SMB,ts} + \beta_{i,t}^{V} r_{HML,ts} + \varepsilon_{i,ts}, \tag{C4}$$

where $r_{SMB,ts}$ and $r_{HML,ts}$ are the excess returns of the SMB and HML portfolios, respectively. We use the same set of covariates for predicting $\beta_{i,t}^{S}$ and $\beta_{i,t}^{V}$ as for predicting market betas, and separately forecast these factor betas for all stocks. We also use the same sample splitting scheme and hyperparameter sets. After computing the factor beta forecasts, we evaluate them against the realized factor betas over the next year. We compute these realized factor betas from a multiple regression of daily returns on a constant and the three factors. MSE and MAE are used to evaluate the forecast accuracy.

The results are shown in Table C11. We find that the machine learning-based models produce clearly smaller forecast errors than the benchmarks. This is true for both SMB and HML betas. For

[19] Not all benchmark estimators naturally extend to a multi-factor setup. We skip these benchmarks for the present analysis. In addition, we omit the *nn_1* model from the presentation because it is computationally very expensive.

example, for SMB betas, the *ols_1y_d* benchmark estimator produces an average value-weighted MSE of 20.68%. The *rf* model, on the other hand, produces an average value-weighted MSE that is more than 30% lower (14.37%). In 88% of the months, the value-weighted MSE of the *rf* model is significantly lower. Thus, the outperformance of the machine learning-based estimators is even more pronounced for SMB betas than for market betas. For HML betas, the value-weighted MSE is 19% lower for the *rf* model than for *ols_1y_d* (35.15% vs. 43.36%). In 53% of the months, the value-weighted MSE of the *rf* model is significantly lower when predicting HML betas. Alternative benchmark models perform similarly or worse than *ols_1y_d*. Thus, this section shows that machine learning methods are very promising for the prediction of factor betas.

Table C1
Cross-Sectional and Time-Series Properties of Beta Estimates

This table reports the properties of the beta estimates obtained from the forecasting models presented in Section IV of the main paper. Panel A focuses on the cross-sectional properties, reporting the time-series means of the monthly 1) value-weighted cross-sectional averages of the estimated betas, 2) value-weighted cross-sectional standard deviations, and 3) cross-sectional minimum, median, and maximum values. Following Pastor and Stambaugh (1999), the implied cross-sectional standard deviation of the true betas is also reported: $\widehat{Std}(\beta^R) = \left[\overline{Var(\beta^F)} - \overline{\widehat{Var}}_{\beta_i^R}\right]^{1/2}$. Panel B focuses on time-series properties and presents the value-weighted cross-sectional means of 1) time-series averages, 2) time-series standard deviations, 3) time-series minima, medians, and maxima, and 4) first-order autocorrelations of the estimated betas. Following Becker et al. (2021), firms with less than fifty beta estimates are omitted from the summary statistics. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020, while the first beta estimates are obtained in December 1979.

		Panel A: Cross-Sectional Properties						*Panel B: Time-Series Properties*					
	Model	Mean	Std	Min	Median	Max	Impl. Std	Mean	Std	Min	Median	Max	Auto-corr.
Benchmark Estimators	ols_5y_m	1.00	0.47	–1.09	0.97	4.68	0.31	1.09	0.32	0.49	1.06	1.82	0.96
	ols_1y_d	1.00	0.40	–1.39	0.78	3.22	0.26	1.05	0.29	0.44	1.02	1.84	0.95
	ewma_s	1.00	0.41	–1.66	0.78	3.44	0.26	1.05	0.31	0.38	1.03	1.92	0.93
	ewma_l	1.00	0.40	–1.47	0.78	3.27	0.26	1.05	0.30	0.43	1.02	1.86	0.95
	bsw	0.98	0.36	–0.19	0.80	2.26	0.25	1.03	0.26	0.49	1.01	1.67	0.96
	vasicek	0.99	0.36	–0.18	0.83	2.29	0.25	1.04	0.26	0.48	1.02	1.72	0.96
	karolyi	0.99	0.37	–0.19	0.84	2.43	0.26	1.04	0.26	0.49	1.01	1.74	0.96
	hybrid	0.99	0.36	–0.30	0.86	2.48	0.25	1.05	0.25	0.52	1.03	1.67	0.96
	fama-french	0.99	0.34	0.18	0.77	1.89	0.23	1.04	0.26	0.49	1.02	1.77	0.91
	long-memo	1.00	0.34	–0.54	0.79	2.39	0.28	1.06	0.19	0.68	1.05	1.51	0.92
ML Estimators	lm	1.03	0.28	–0.42	0.79	2.13	0.19	1.08	0.19	0.67	1.07	1.64	0.92
	elanet	1.03	0.26	–0.36	0.79	2.08	0.18	1.06	0.18	0.68	1.05	1.62	0.92
	rf	0.99	0.28	0.05	0.80	1.92	0.19	1.04	0.19	0.66	1.03	1.50	0.92
	gbrt	0.98	0.29	0.00	0.78	1.92	0.20	1.02	0.20	0.61	1.02	1.51	0.91
	nn_1	0.99	0.30	–0.09	0.78	2.12	0.21	1.03	0.20	0.65	1.01	1.54	0.91

Table C2
Conditional predictive ability

This table reports the results of conditional tests for equal predictive ability of the forecasting models presented in Section IV of the main paper. We report the test statistics of Giacomini and White (2006). To do so, we use the mean squared forecast error differentials $d_t^{(j,l)} = \mathrm{M}SE_{t+k|t}^{(\mathrm{j})} - \mathrm{M}SE_{t+k|t}^{(l)}$ between models j and l and regress them on a constant and the lagged forecast error differential, $\delta_t = \left(1\ d_t^{(j,l)}\right)'$. The test statistic is $GW = T\left(T^{-1}\sum_{t=1}^{T}\delta_{t-1}\, d_t^{(j,l)}\right)'\widehat{\Omega}^{-1}\left(T^{-1}\sum_{t=1}^{T}\delta_{t-1}\, d_t^{(j,l)}\right)$, where $\widehat{\Omega}$ is the Newey and West (1987) covariance of $\delta_{t-1}d_t^{(j,l)}$. The GW test statistic is distributed as $\chi^2_{\square}$ with 2 degrees of freedom. The critical values are 4.605 (at 10%), 5.991 (at 5%), and 9.210 (at 1%). *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Benchmark estimators										ML estimators				
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	hybrid	fama-french	long-memo	lm	elanet	rf	gbrt	nn_1
Benchmark estimators	vs. ols_5y_m		6.78**	10.10***	8.36**	8.24**	8.19**	7.56**	8.66**	7.64**	11.17***	22.27***	11.34***	11.80***	14.68***	17.36***
	vs. ols_1y_d			4.11	8.87**	16.34***	11.14***	13.59***	13.83***	5.87*	6.63**	5.36*	5.78*	12.70***	9.57***	13.21***
	vs. ewma_s				9.10**	11.96***	10.59***	11.10***	14.25***	8.75**	13.24***	7.64**	7.74**	17.10***	12.97***	16.64***
	vs. ewma_l					10.58***	8.71**	10.72***	12.15***	5.79*	8.74**	5.33*	5.73*	13.44***	9.79***	13.24***
	vs. bsw						4.64*	1.96	2.33	3.07	3.42	1.50	1.70	7.46**	4.64*	6.41**
	vs. vasicek							0.24	3.84	4.01	4.17	1.95	2.39	8.41**	5.47*	7.58**
	vs. karolyi								5.11*	2.42	4.29	2.17	2.55	8.19**	5.76*	8.30**
	vs. hybrid									4.16	3.46	1.06	0.93	5.49*	3.78	6.97**
	vs. fama-french										3.28	2.33	2.62	11.45***	8.35**	9.97***
	vs. long-memo											2.37	1.09	11.34***	3.83	12.34***
ML estimators	vs. lm												1.47	7.45**	4.71*	7.53**
	vs. elanet													8.78**	3.28	4.62*
	vs. rf														3.19	3.84
	vs. gbrt															5.52*
	vs. nn_1															

Table C3
Forecast errors (additional forecast models)

This table shows the differences in forecast errors obtained from the forecasting models presented in Section IV of the main paper and Internet Appendix C. Panel A reports the time-series averages of the monthly value-weighted MSEs: $MSE_{t+k|t}^{(j)} = \sum_{i=1}^{N_t} w_{i,t}(\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F,(j)})^2$, with $k = 12$, where N_t is the number of stocks in the sample at the end of month t and $w_{i,t}$ is the market capitalization-based weight of stock i. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen et al. (2011) model confidence set (MCS) and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test statistics). The DM tests of equal predictive ability examine the differences in stock-level squared forecast errors (SEs): $SE_{i,t+k|t}^{(j)} = (\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F,(j)})^2$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_t^{(j,l)} = \frac{\bar{d}_t^{(j,l)}}{\hat{\sigma}_{\bar{d}_t^{(j,l)}}}$, where $d_{i,t}^{(j,l)} = SE_{i,t+k|t}^{(j)} - SE_{i,t+k|t}^{(l)}$ is the difference in SEs, $\bar{d}_t^{(j,l)} = \sum_{i=1}^{N_t} w_{i,t} d_{i,t}^{(j,l)}$ the value-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_t^{(j,l)}}$ the Newey and West (1987) standard error of $\bar{d}_t^{(j,l)}$ (with four lags to account for possible heteroscedasticity and autocorrelation). Statistical tests are based on the 10% significance level. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Baseline specifications							Alternative specifications							
		ols_1y_d	long-memo	lm	elanet	rf	gbrt	nn_1	nn_2	nn_3	nn_4	nn_5	rf_amv	ens_1	ens_2	ens_3
Panel A: Average forecast errors																
	MSE, v.w. [%]	9.70	8.29	9.15	8.89	7.77	8.04	7.79	7.98	8.02	8.08	7.99	7.90	7.84	7.71	7.64
Panel B: Forecast errors over time																
	In MCS	39.92	60.91	39.92	45.95	81.08	65.70	77.55	69.85	70.69	62.99	66.94	74.01	76.09	80.87	83.58
Baseline specifications	vs. ols_1y_d		60.29	46.78	50.73	68.61	61.75	62.99	59.46	59.04	57.59	57.59	62.79	61.54	64.86	68.81
	vs. long-memo	14.97		21.62	27.23	43.04	40.33	43.04	39.50	40.75	40.33	39.71	42.83	47.82	52.39	52.18
	vs. lm	27.65	40.75		37.63	65.28	54.47	63.83	58.00	54.05	53.64	58.00	55.30	82.95	77.55	72.56
	vs. elanet	25.78	34.10	18.92		56.13	49.69	54.26	51.14	48.44	49.48	54.47	50.52	74.64	71.31	61.54
	vs. rf	9.77	10.81	5.82	8.94		18.30	20.17	16.63	15.18	14.76	17.46	18.09	32.43	40.75	46.15
	vs. gbrt	16.22	19.54	9.77	18.71	39.92		34.93	27.23	22.87	21.21	25.57	33.26	43.87	54.26	62.58
	vs. nn_1	12.89	11.02	6.65	13.93	23.70	16.22		16.22	15.18	9.77	17.26	23.70	29.94	38.88	32.64
Alternative specifications	vs. nn_2	14.55	15.18	8.52	15.80	32.85	21.62	45.11		27.65	21.62	24.95	29.52	35.34	44.28	45.74
	vs. nn_3	14.97	15.38	7.28	14.35	32.43	23.28	40.96	32.02		24.53	28.90	28.07	39.71	45.11	48.65
	vs. nn_4	16.22	17.67	8.73	17.05	36.59	25.16	49.48	36.17	37.63		32.43	32.02	37.21	44.70	51.14
	vs. nn_5	13.51	17.46	6.24	15.38	34.51	22.04	42.00	29.31	26.82	19.54		29.52	40.75	45.74	51.98
	vs. rf_amv	10.60	16.84	8.32	10.60	34.10	23.08	31.39	22.87	20.17	21.62	22.25		38.67	44.28	47.82
	vs. ens_1	11.02	14.76	2.29	5.20	29.11	17.26	22.87	18.50	15.38	14.35	14.76	26.82		48.44	39.71
	vs. ens_2	11.23	12.68	2.49	3.74	21.62	9.56	18.92	14.14	11.02	9.98	12.06	18.92	11.85		36.17
	vs. ens_3	10.60	8.32	3.74	6.86	17.67	7.90	11.23	12.06	8.52	7.07	10.40	16.01	21.41	32.02	
	T	481	481	481	481	481	481	481	481	481	481	481	481	481	481	481

Table C4
Forecast errors (equal-weighted mean squared errors)

This table shows the differences in forecast errors obtained from the forecasting models presented in Section IV of the main paper. Panel A reports the time-series averages of the monthly equal-weighted MSEs: $MSE_{t+k|t}^{(j)} = \frac{1}{N}\sum_{i=1}^{N_t}(\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F,(j)})^2$, with $k = 12$, where N_t is the number of stocks in the sample at the end of month t. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen et al. (2011) model confidence set (MCS) and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) (DM) test statistics). The DM tests of equal predictive ability examine the differences in stock-level squared forecast errors (SEs): $SE_{i,t+k|t}^{(\mathrm{j})} = (\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F,(j)})^2$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_t^{(j,l)} = \frac{\bar{d}_t^{(j,l)}}{\hat{\sigma}_{\bar{d}_t^{(j,l)}}}$, where $d_{i,t}^{(j,l)} = SE_{i,t+k|t}^{(\mathrm{j})} - SE_{i,t+k|t}^{(l)}$ is the difference in SEs, $\bar{d}_t^{(j,l)} = \frac{1}{\mathrm{N_t}}\sum_{i=1}^{N_t} d_{i,t}^{(j,l)}$ the equal-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_t^{(j,l)}}$ the Newey and West (1987) standard error of $\bar{d}_t^{(j,l)}$ (with four lags to account for possible heteroscedasticity and autocorrelation). Statistical tests are based on the 10% significance level. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Benchmark estimators										ML estimators				
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	hybrid	fama-french	long-memo	lm	elanet	rf	gbrt	nn_1
Panel A: Average forecast errors																
	MSE, e.w. [%]	48.60	21.73	22.91	21.73	16.86	17.63	18.45	18.60	16.73	16.39	16.52	16.42	15.42	15.84	15.16
Panel B: Forecast errors over time																
	In MCS	0.00	2.08	5.20	3.95	19.54	9.56	8.94	8.73	24.32	24.95	20.58	23.28	44.91	38.67	69.65
Benchmark estimators	vs. ols_5y_m		96.88	93.56	96.67	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.38	100.00
	vs. ols_1y_d	0.00		22.25	37.21	93.35	87.73	74.01	59.46	87.32	86.07	79.00	79.83	84.82	82.95	84.82
	vs. ewma_s	0.42	49.27		66.94	87.53	84.20	72.14	58.63	81.91	84.41	77.55	78.17	83.99	82.74	84.20
	vs. ewma_l	0.00	29.31	11.85		90.02	86.49	67.36	56.55	83.16	83.37	76.92	78.79	83.16	81.70	83.99
	vs. bsw	0.00	1.25	5.20	3.95		9.36	6.86	18.50	33.47	41.58	55.09	55.51	68.61	66.53	72.14
	vs. vasicek	0.00	1.25	5.20	4.16	58.00		7.69	29.11	51.14	55.30	60.29	61.33	75.05	75.05	77.75
	vs. karolyi	0.00	4.16	6.24	5.20	73.60	66.53		34.51	61.12	66.11	66.11	69.23	77.34	76.72	79.00
	vs. hybrid	0.00	16.01	15.80	18.30	63.41	55.93	42.41		64.03	62.79	71.93	72.14	79.83	79.00	83.37
	vs. fama-french	0.00	2.91	5.82	5.41	34.93	23.08	14.55	21.62		41.79	50.52	50.31	66.11	65.28	71.52
	vs. long-memo	0.00	3.53	3.33	3.95	22.66	17.46	13.31	11.64	27.86		42.41	42.83	57.59	58.84	68.81
ML estimators	vs. lm	0.00	13.31	14.76	14.97	27.03	22.25	20.17	16.22	30.35	31.60		45.11	61.75	58.42	79.63
	vs. elanet	0.00	13.72	12.89	13.51	27.65	22.66	19.33	17.05	29.94	31.19	24.95		61.33	59.04	76.92
	vs. rf	0.00	8.73	8.52	9.56	18.50	15.59	12.89	10.60	18.92	19.75	9.15	8.52		21.83	51.56
	vs. gbrt	0.00	9.98	9.98	10.60	20.79	17.05	15.59	12.68	19.96	23.28	17.05	17.05	39.09		59.88
	vs. nn_1	0.00	8.52	8.94	9.15	17.26	15.59	13.72	10.19	16.84	17.88	4.99	6.65	21.62	14.76	
	T	481	481	481	481	481	481	481	481	481	481	481	481	481	481	481

Table C5

Forecast errors (value-weighted mean absolute errors)

This table shows the differences in forecast errors obtained from the forecasting models presented in Section IV of the main paper. Panel A reports the time-series averages of the monthly value-weighted mean absolute errors (MAEs): $MAE_{t+k|t}^{(j)} = \sum_{i=1}^{N_t} w_{i,t} |\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F,(j)}|$, with $k = 12$, where N_t is the number of stocks in the sample at the end of month t and $w_{i,t}$ is the market capitalization-based weight of stock i. Panel B reports the fraction of months during the out-of-sample period for which the column model is 1) in the Hansen et al. (2011) model confidence set (MCS) and 2) significantly better than the row model in a pairwise comparison (according to the Diebold and Mariano (1995) test statistics). The DM tests of equal predictive ability examine the differences in stock-level absolute forecast errors (AEs): $AE_{i,t+k|t}^{(\mathrm{j})} = |\beta_{i,t+k}^{R} - \beta_{i,t+k|t}^{F,(j)}|$, with $k = 12$. The DM test statistic in month t for comparing the model under investigation j with a competing model i is $DM_t^{(j,l)} = \frac{\bar{d}_t^{(j,l)}}{\hat{\sigma}_{\bar{d}_t^{(j,l)}}}$, where $d_{i,t}^{(j,l)} = AE_{i,t+k|t}^{(\mathrm{j})} - AE_{i,t+k|t}^{(\mathrm{l})}$ is the difference in SEs, $\bar{d}_t^{(j,l)} = \sum_{i=1}^{N_t} w_{i,t} d_{i,t}^{(j,l)}$ the value-weighted cross-sectional average of these differences, and $\hat{\sigma}_{\bar{d}_t^{(j,l)}}$ the Newey and West (1987) standard error of $\bar{d}_t^{(j,l)}$ (with four lags to account for possible heteroscedasticity and autocorrelation). Statistical tests are based on the 10% significance level. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Benchmark estimators										ML estimators				
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	hybrid	fama-french	long-memo	lm	elanet	rf	gbrt	nn_1
Panel A: Average forecast errors																
	MAE, v.w. [%]	32.37	23.05	22.86	22.73	22.03	22.21	22.23	21.75	22.34	21.37	22.74	22.40	20.64	20.94	20.64
Panel B: Forecast errors over time																
	In MCS	3.74	36.38	45.95	44.91	55.30	53.01	53.85	59.46	49.90	72.77	54.47	61.54	85.03	77.55	82.12
Benchmark estimators	vs. ols_5y_m		83.16	83.37	84.82	88.36	86.69	86.07	93.14	87.53	96.88	87.94	87.53	96.26	95.22	95.63
	vs. ols_1y_d	2.70		30.15	44.70	75.26	80.67	85.65	70.48	46.36	54.47	43.45	48.02	63.41	57.59	62.16
	vs. ewma_s	2.91	22.87		35.14	46.36	45.74	44.49	49.69	33.26	50.94	38.88	41.79	58.00	57.59	58.21
	vs. ewma_l	2.49	15.38	20.37		50.31	47.40	48.86	54.47	34.30	50.73	39.71	42.20	57.80	55.30	57.17
	vs. bsw	1.87	4.99	13.93	10.60		18.92	20.17	39.71	15.38	39.50	30.98	36.59	53.01	49.48	51.35
	vs. vasicek	2.08	7.69	15.18	13.31	29.52		19.33	47.19	17.05	39.09	32.85	37.01	56.34	52.39	53.85
	vs. karolyi	2.08	4.16	15.80	12.27	22.87	25.36		45.32	20.17	38.88	32.85	38.05	56.13	50.52	53.85
	vs. hybrid	1.66	4.99	13.31	13.51	18.30	19.13	21.00		12.89	29.11	26.82	29.73	47.19	43.66	46.99
	vs. fama-french	1.66	13.93	16.63	17.05	27.65	26.61	28.07	37.84		44.07	28.27	32.02	58.42	56.34	55.93
	vs. long-memo	0.00	14.97	16.01	16.84	18.71	18.30	19.13	18.92	16.84		18.30	23.08	37.42	35.34	42.41
ML estimators	vs. lm	3.95	29.94	29.73	32.43	35.76	34.51	36.59	36.80	30.77	40.12		35.55	61.12	55.93	63.62
	vs. elanet	5.20	29.94	29.94	31.81	35.97	35.34	36.59	34.72	31.39	34.51	17.26		52.60	51.77	55.72
	vs. rf	0.00	9.56	8.73	10.40	14.35	13.51	13.31	13.51	6.86	12.47	3.33	8.94		19.33	23.70
	vs. gbrt	0.00	13.72	14.97	16.84	19.13	18.92	20.58	18.50	11.85	17.46	7.48	11.43	30.77		28.48
	vs. nn_1	0.00	12.47	12.47	13.31	17.05	16.63	16.84	14.76	12.27	13.31	4.37	10.19	22.25	16.42	
	T	481	481	481	481	481	481	481	481	481	481	481	481	481	481	481

Table C6
Unbiasedness test

This table reports the results of Mincer and Zarnowitz (1969) regressions to test the unbiasedness of the forecasting models presented in Section IV of the main paper. Following Fama and MacBeth (1973), at the end of each month t, either a weighted least squares (WLS) regression (using the stock market capitalization-based weights) or an ordinary least squares (OLS) regression (using equal weights) of realized betas is run on the beta estimates obtained from the different forecasting models: $\beta^{R}_{i,t+k} = a_t + b_t\beta^{F,(j)}_{i,t+k|t} + e_{i,t+k}$. In particular, this table reports the time-series averages of the monthly intercepts (a) and slopes (b), and the t-statistics (in parentheses) testing the null hypotheses that $a = 0$ and $b = 1$, respectively. The t-tests are based on Newey and West (1987) standard errors (with eleven lags to account for possible heteroscedasticity and autocorrelation). Panel A shows the value-weighted results (based on WLS regressions), while Panel B adds the results for equal weights (based on OLS regressions). The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Panel A: Value-weighted results		*Panel B: Equal-weighted results*	
	Model	α	β	α	β
Benchmark estimators	ols_5y_m	0.50	0.49	0.42	0.36
		(3.76)	(–4.72)	(1.54)	(–9.11)
	ols_1y_d	0.25	0.74	0.28	0.65
		(2.88)	(–3.04)	(6.78)	(–5.29)
	ewma_s	0.26	0.73	0.30	0.62
		(4.43)	(–4.64)	(8.42)	(–6.55)
	ewma_l	0.25	0.74	0.28	0.64
		(3.43)	(–3.62)	(7.34)	(–5.77)
	bsw	0.18	0.83	0.12	0.83
		(1.87)	(–1.86)	(2.85)	(–2.90)
	vasicek	0.17	0.83	0.08	0.85
		(1.79)	(–1.84)	(1.53)	(–2.02)
	karolyi	0.18	0.82	0.06	0.86
		(1.91)	(–1.98)	(1.01)	(–1.62)
	hybrid	0.15	0.85	0.10	0.80
		(1.46)	(–1.64)	(1.23)	(–3.22)
	fama-french	0.12	0.87	0.10	0.89
		(1.33)	(–1.40)	(2.68)	(–2.25)
	long-memo	0.12	0.86	0.14	0.82
		(1.19)	(–1.59)	(3.89)	(–3.72)
ML estimators	lm	–0.10	1.06	0.06	0.94
		(–0.57)	(0.35)	(1.28)	(–1.44)
	elanet	–0.18	1.14	0.01	1.00
		(–1.10)	(1.04)	(0.15)	(–0.05)
	rf	–0.09	1.08	0.00	1.03
		(–0.75)	(0.78)	(–0.02)	(0.34)
	gbrt	–0.05	1.06	0.03	1.01
		(–0.45)	(0.54)	(0.38)	(0.07)
	nn_1	–0.02	1.02	0.03	1.00
		(–0.20)	(0.23)	(0.44)	(–0.04)

Table C7
Forecast errors (different realized beta horizons and sampling frequencies)

This table shows the differences in forecast errors for realized betas over different horizons. The forecasting models are presented in Section IV of the main paper. We report the time-series averages of the monthly value-weighted MSEs, equal-weighted MSEs, value-weighted MAEs, and value-weighted mean squared hedging errors. We report results for the following realized beta specifications (period, sampling frequency):

(1) three months, daily returns

(2) six months, daily returns

(3) one year, weekly returns

(4) five years, monthly returns

The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Benchmark estimators									ML estimators			
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	bsw	vasicek	karolyi	fama-french	long-memo	lm	elanet	rf	gbrt
(1)	MSE, v.w. [%]	25.13	14.28	13.83	13.87	13.75	13.82	13.83	14.20	13.85	14.59	14.44	13.22	13.44
	MSE, e.w. [%]	70.59	41.90	42.52	41.61	37.83	38.66	39.63	37.79	37.79	38.00	37.84	36.59	36.91
	MAE, v.w. [%]	37.12	27.58	27.06	27.13	27.05	27.14	27.14	27.49	27.24	28.07	27.88	26.41	26.61
	MSHE, v.w. [%]	7.74	7.62	7.61	7.61	7.55	7.56	7.57	7.55	7.52	7.51	7.53	7.49	7.50
(2)	MSE, v.w. [%]	21.04	10.77	10.49	10.44	10.07	10.18	10.21	10.48	9.97	10.77	10.46	9.39	9.61
	MSE, e.w. [%]	55.58	27.76	28.59	27.58	23.36	24.16	25.05	23.32	23.12	23.39	23.28	22.11	22.56
	MAE, v.w. [%]	34.04	24.19	23.81	23.79	23.42	23.56	23.58	23.84	23.30	24.42	24.02	22.52	22.76
	MSHE, v.w. [%]	7.74	7.62	7.61	7.61	7.55	7.56	7.57	7.55	7.52	7.52	7.51	7.48	7.50
(3)	MSE, v.w. [%]	23.37	17.42	17.27	17.15	16.32	16.50	16.56	16.83	15.84	16.16	15.69	14.64	14.77
	MSE, e.w. [%]	62.62	45.75	46.81	45.68	40.39	40.45	40.65	41.44	40.04	38.55	38.45	37.47	38.04
	MAE, v.w. [%]	35.25	30.82	30.68	30.58	29.92	30.05	30.06	30.30	29.29	30.36	29.72	28.32	28.43
	MSHE, v.w. [%]	7.74	7.62	7.61	7.61	7.55	7.56	7.57	7.55	7.52	7.54	7.50	7.45	7.47
(4)	MSE, v.w. [%]	24.66	21.63	22.08	21.67	20.41	20.46	20.50	20.58	18.75	18.70	17.85	17.15	16.83
	MSE, e.w. [%]	46.91	44.47	46.13	44.78	38.54	37.39	36.37	40.62	38.77	32.96	31.94	29.85	30.22
	MAE, v.w. [%]	35.48	33.83	34.24	33.87	32.74	32.84	32.81	32.91	31.22	31.40	30.37	29.42	29.02
	MSHE, v.w. [%]	7.55	7.43	7.42	7.42	7.37	7.37	7.38	7.36	7.33	7.28	7.24	7.21	7.21

Table C8

Minimum variance portfolios (first half of the sample period)

This table reports the properties of the minimum variance portfolios for the first half of the sample period. The portfolios are constructed based on beta estimates obtained from the forecasting models introduced in Section IV of the main paper. For the portfolio optimization, we impose a single-factor structure on the covariance matrix of stock returns. Thus, the market betas are the primary determinants of the stock weights in the minimum variance portfolio. The approach is described in detail in Section VI.C of the main paper. Each month, we compute the weights that minimize the expected portfolio variance, subject to the constraints that the weights are positive, that each individual weight is less than 5%, and that the weights sum to 1. The forecasts for the market and idiosyncratic variances are based on daily returns over the previous year. Panel A presents the annualized risk and return measures of the resulting minimum variance portfolios. Std reports the ex-post time-series standard deviation and Dwnd the ex-post downside standard deviation (of negative returns). Min is the lowest monthly excess return and MaxDD is the maximum drawdown of the minimum variance portfolio from peak to trough over multi-month periods. TV is the terminal value in November 1999 of a $1 investment in the minimum variance portfolio in December 1979. Mean is the average portfolio return, and SR is the Sharpe ratio. Panel B reports the ex-post market betas of the minimum variance portfolios (β_{pv}) as well as the beta of a market-neutral minimum variance portfolio that hedges the expected market risk (depending on the portfolio beta forecast) each month using an additional investment in the market portfolio (β_{mn}). The *t*-statistics in parentheses are based on Newey and West (1987) robust standard errors with 11 lags. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the first half of our sample period from March 1970 to November 1999 and have a market capitalization above the 20^{th} percentile of NYSE stocks. The first beta estimates are obtained in December 1979.

		Panel A: Minimum variance							*Panel B: Market neutrality*	
	Model	Std [%]	Dwnd [%]	Min [%]	MaxDD [%]	TV [%]	Mean [%]	SR	β_{pv}	β_{mn}
Benchmark estimators	ols_1y_d	12.96	11.08	–24.24	25.88	6.19	10.01	0.77	0.26	–0.21
									(11.55)	(–8.00)
	bsw	12.56	10.09	–21.91	22.8	6.07	9.85	0.78	0.24	–0.13
									(10.18)	(–6.44)
	hybrid	12.76	10.11	–21.88	24.08	5.19	9.08	0.71	0.24	–0.10
									(6.26)	(–2.17)
	fama-french	12.47	10.02	–23.04	23.04	6.22	9.97	0.8	0.29	–0.05
									(16.26)	(–1.90)
	long-memo	12.83	9.93	–20.35	26.35	5.14	9.04	0.7	0.22	–0.11
									(8.78)	(–4.18)
ML estimators	lm	11.94	8.36	–12.98	18.54	5.9	9.62	0.81	0.24	–0.08
									(10.64)	(–3.50)
	elanet	12.44	10.2	–22.34	22.34	5.31	9.17	0.74	0.22	–0.05
									(10.54)	(–1.75)
	rf	12.19	9.17	–19.32	24.84	5.09	8.91	0.73	0.21	0.04
									(7.22)	(1.20)
	gbrt	11.16	8.84	–18.82	19.1	6.51	10.03	0.9	0.22	0.03
									(9.36)	(1.25)
	nn_1	11.81	8.79	–16.28	20.71	6.11	9.78	0.83	0.21	0.00
									(7.83)	(–0.18)

Table C9
Minimum variance portfolios (second half of the sample period)

This table reports the properties of the minimum variance portfolios for the second half of the sample period. The portfolios are constructed based on beta estimates obtained from the forecasting models introduced in Section IV of the main paper. For the portfolio optimization, we impose a single-factor structure on the covariance matrix of stock returns. Thus, the market betas are the primary determinants of the stock weights in the minimum variance portfolio. The approach is described in detail in Section VI.C of the main paper. Each month, we compute the weights that minimize the expected portfolio variance, subject to the constraints that the weights are positive, that each individual weight is less than 5%, and that the weights sum to 1. The forecasts for the market and idiosyncratic variances are based on daily returns over the previous year. Panel A presents the annualized risk and return measures of the resulting minimum variance portfolios. Std reports the ex-post time-series standard deviation and Dwnd the ex-post downside standard deviation (of negative returns). Min is the lowest monthly excess return and MaxDD is the maximum drawdown of the minimum variance portfolio from peak to trough over multi-month periods. TV is the terminal value in December 2019 of a $1 investment in the minimum variance portfolio in December 1999. Mean is the average portfolio return, and SR is the Sharpe ratio. Panel B reports the ex-post market betas of the minimum variance portfolios (β_{pv}) as well as the beta of a market-neutral minimum variance portfolio that hedges the expected market risk (depending on the portfolio beta forecast) each month using an additional investment in the market portfolio (β_{mn}). The *t*-statistics in parentheses are based on Newey and West (1987) robust standard errors with 11 lags. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the second half of our sample period from December 1999 to December 2020 and have a market capitalization above the 20^{th} percentile of NYSE stocks. The first beta estimates are obtained in December 1999.

		Panel A: Minimum variance							*Panel B: Market neutrality*	
	Model	Std [%]	Dwnd [%]	Min [%]	MaxDD [%]	TV [%]	Mean [%]	SR	β_{pv}	β_{mn}
Benchmark estimators	ols_1y_d	11.84	8.73	–14.37	43.55	4.52	8.24	0.7	0.52	–0.24
									(7.67)	(–3.55)
	bsw	11.55	8.42	–14.53	43.07	4.50	8.18	0.71	0.49	–0.18
									(6.69)	(–2.23)
	hybrid	11.65	8.51	–14.32	41.12	5.06	8.78	0.75	0.49	–0.19
									(6.92)	(–2.57)
	fama-french	11.57	8.99	–13.21	37.08	4.12	7.74	0.67	0.52	–0.09
									(9.76)	(–1.36)
	long-memo	11.00	8.04	–10.84	34.73	5.64	9.25	0.84	0.50	–0.14
									(7.94)	(–1.84)
ML estimators	lm	11.50	9.17	–12.11	50.15	5.99	9.62	0.84	0.59	–0.18
									(8.75)	(–1.84)
	elanet	11.39	9.80	–13.94	49.34	5.83	9.46	0.83	0.59	–0.15
									(9.45)	(–1.58)
	rf	10.62	7.36	–10.82	39.12	6.50	9.92	0.93	0.49	–0.02
									(7.52)	(–0.12)
	gbrt	11.06	8.06	–11.74	38.85	6.51	9.98	0.90	0.50	–0.03
									(6.83)	(–0.25)
	nn_1	10.50	7.42	–10.95	35.08	6.14	9.63	0.92	0.49	–0.06
									(6.81)	(–0.50)

Table C10
Conditional market neutrality of betting-against-beta portfolios

This table analyzes the conditional market neutrality of the betting-against-beta (BAB) portfolios. Each month, we construct decile portfolios by sorting the stocks by their beta estimates, using the predicted beta of each forecasting model. The portfolios go long and short the extreme deciles, buying the stocks in decile one and shorting those in decile ten. Finally, the portfolios are hedged each month with a position in the market portfolio equal to the negative of the portfolio beta predicted by the forecasting models. We then regress the annualized BAB returns on a constant, the current market excess return interacted with the log of the one-year to five-year market volatility ratio, the current market excess return, and the lagged market excess returns of the previous two months. The regression equation is $r_{BAB,t} = \alpha_{BAB} + \beta_1 r_{M,t} \ln{(\sigma_1^{MKT}/\sigma_5^{MKT})} + \beta_2 r_{M,t} + \beta_3 r_{M,t-1} + \beta_4 r_{M,t-2} + \varepsilon_{i,t}$, where $r_{BAB,t}$ is the annualized monthly BAB long–short portfolio return in month t, $r_{M,t}$ is the market return in month t, and σ_1^{MKT} and σ_5^{MKT} are the one-year and five-year estimates of (daily) market excess return volatility, respectively, lagged by one month. Following Novy-Marx and Velikov (2022), we standardize this volatility ratio to have a mean of zero and a standard deviation of one. The t-statistics in parentheses are based on Newey and West (1987) robust standard errors with four lags. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020 and have a market capitalization above the 20^{th} percentile of NYSE stocks. The first beta estimates are obtained in December 1979.

	Model	α [%]	β₁	β₂	β₃	β₄	Adj. R^2 [%]
Benchmark estimators	ols_1y_d	7.26	–0.14	0.44	0.06	0.03	9.97
		(2.25)	(–1.61)	(4.48)	(1.19)	(0.44)	
	bsw	6.02	–0.12	0.30	0.06	0.02	4.97
		(1.90)	(–1.51)	(3.22)	(1.32)	(0.34)	
	hybrid	9.17	–0.08	0.22	0.03	0.00	2.05
		(2.91)	(–1.05)	(2.70)	(0.56)	(0.03)	
	fama-french	7.27	–0.09	0.26	0.07	0.00	4.02
		(2.49)	(–1.15)	(3.18)	(1.46)	(0.01)	
	long-memo	8.60	–0.11	0.21	0.02	0.00	2.41
		(2.84)	(–1.76)	(2.76)	(0.31)	(0.05)	
ML estimators	lm	8.61	0.00	0.07	0.07	0.01	0.06
		(2.61)	(0.04)	(0.91)	(1.41)	(0.11)	
	elanet	8.42	–0.03	0.05	0.08	0.00	–0.21
		(2.66)	(–0.39)	(0.51)	(1.75)	(0.08)	
	rf	8.64	–0.14	–0.03	0.06	–0.02	1.32
		(2.54)	(–1.73)	(–0.30)	(1.14)	(–0.43)	
	gbrt	9.86	–0.14	0.00	0.05	–0.03	0.76
		(2.92)	(–1.49)	(0.04)	(0.89)	(–0.66)	
	nn_1	9.51	–0.12	0.05	0.05	–0.03	0.18
		(2.88)	(–1.30)	(0.59)	(0.91)	(–0.46)	

Table C11
Forecast errors of size and value betas

This table shows the differences in forecast errors for realized factor betas. We consider forecasts for betas of the Fama and French (1993) three-factor model. Specifically, we forecast realized size (small-minus-big; SMB) and value (high-minus-low; HML) betas. The realized betas are computed from a multiple regression of daily returns over the next year on a constant and the three Fama and French (1993) factors. The forecasting models are adapted versions of those presented in Section IV of the main paper. We report the time-series averages of the monthly value-weighted MSEs, equal-weighted MSEs, and value-weighted MAEs. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Benchmark estimators				ML estimators			
		ols_5y_m	ols_1y_d	ewma_s	ewma_l	lm	elanet	rf	gbrt
SMB	MSE, v.w. [%]	37.87	20.68	21.68	20.67	24.30	23.46	14.37	14.51
	MSE, e.w. [%]	127.47	62.40	68.92	63.74	43.16	42.60	38.46	38.78
	MAE, v.w. [%]	44.97	33.23	34.06	33.26	37.32	36.87	27.70	27.88
HML	MSE, v.w. [%]	65.43	43.36	43.80	42.63	43.30	41.41	35.15	35.44
	MSE, e.w. [%]	134.25	96.61	107.57	99.08	61.02	59.86	57.27	58.42
	MAE, v.w. [%]	59.41	47.29	47.70	46.98	47.86	46.70	42.94	43.16

Table C12
Statistical Significance of Alpha Differences

This table shows the *t*-statistics for the alpha differences presented in Table 5 of the main paper. Each month, we construct decile portfolios by sorting the stocks by their momentum (MOM), idiosyncratic volatility (IVOL), and beta estimates (BAB). For the latter, we use the predicted beta of each forecasting model. The anomaly portfolios go long and short in the extreme deciles. For momentum, the resulting portfolio goes long in decile ten and short in decile one, while those for the other two anomalies go long in decile one and short in decile ten. Finally, the portfolios are hedged each month with a position in the market portfolio equal to the negative of the portfolio beta predicted by the forecasting models. We report the alphas of the returns over the next month of these strategies with respect to the CAPM and the Fama and French (2015) 5-factor model (FF5). The *t*-statistics are based on Newey and West (1987) robust standard errors with four lags. We print in **bold** all *t*-statistics that are significant at the 10% level. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020. The first beta estimates are obtained in December 1979.

		Benchmark estimators				ML estimators				
		bsw	hybrid	fama-french	long-memo	lm	elanet	rf	gbrt	nn_1
MOM α_{CAPM}	ols_1y_d	1.24	0.47	0.17	1.07	1.21	0.39	1.63	1.48	1.40
	bsw		-0.63	-0.77	0.82	0.97	0.08	1.71	1.49	1.30
	hybrid			-0.14	1.00	1.20	0.26	1.64	1.44	1.45
	fama-french				1.47	0.97	0.29	2.03	1.80	1.44
	long-memo					0.24	-0.35	1.01	0.80	0.35
	lm						-1.30	0.31	0.20	-0.11
	elanet							1.11	1.01	0.82
	rf								-0.64	-1.10
	gbrt									-0.72
MOM α_{FF5}	ols_1y_d	1.29	0.23	0.40	1.25	0.98	0.05	1.75	1.66	1.50
	bsw		-0.93	-0.53	1.04	0.66	-0.29	1.87	1.72	1.43
	hybrid			0.23	1.35	1.00	-0.01	1.87	1.73	1.71
	fama-french				1.62	0.66	-0.10	2.21	2.04	1.49
	long-memo					-0.12	-0.69	0.84	0.71	0.11
	lm						-1.51	0.62	0.57	0.30
	elanet							1.35	1.30	1.11
	rf								-0.37	-1.22
	gbrt									-0.96
IVOL α_{CAPM}	ols_1y_d	-0.02	-0.39	-0.15	1.15	0.95	1.00	1.03	1.19	1.41
	bsw		-0.53	-0.10	1.54	1.25	1.36	1.56	1.67	1.99
	hybrid			0.36	1.82	1.39	1.31	1.56	1.68	2.02
	fama-french				1.24	1.03	1.10	1.14	1.30	1.51
	long-memo					0.45	0.26	0.27	0.78	1.09
	lm						-0.15	-0.45	0.18	0.42
	elanet							-0.18	0.30	0.45
	rf								1.37	1.97
	gbrt									0.33

Continued on the next page

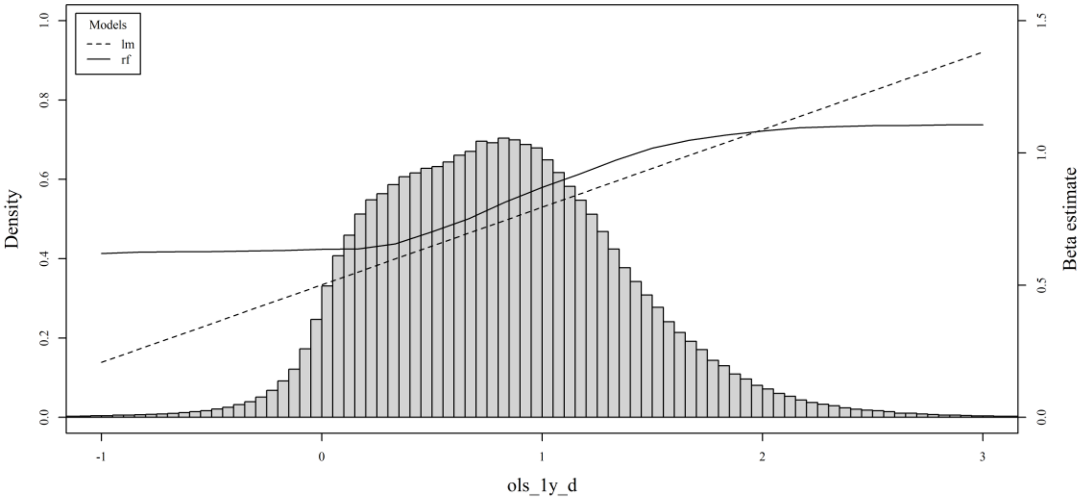
Table C12 continued
Statistical Significance of Alpha Differences

		Benchmark estimators				ML estimators				
		bsw	hybrid	fama-french	long-memo	lm	elanet	rf	gbrt	nn_1
IVOL α_{FF5}	ols_1y_d	-0.26	-0.47	-0.43	1.14	0.82	0.71	0.97	1.15	1.32
	bsw		-0.36	-0.07	1.56	1.14	1.08	1.57	1.67	1.91
	hybrid			0.25	1.78	1.26	1.04	1.55	1.66	1.95
	fama-french				1.30	0.95	0.86	1.15	1.31	1.47
	long-memo					0.19	-0.19	0.02	0.59	0.81
	lm						-0.37	-0.27	0.35	0.58
	elanet							0.33	0.74	0.84
	rf								1.31	1.83
	gbrt									0.27
BAB α_{CAPM}	ols_1y_d	-2.10	1.42	-0.08	0.64	0.48	0.53	1.01	1.54	1.35
	bsw		3.05	1.24	1.71	1.27	1.39	2.29	2.68	2.44
	hybrid			-1.39	-0.47	-0.34	-0.34	-0.19	0.47	0.38
	fama-french				0.63	0.53	0.58	0.90	1.40	1.26
	long-memo					0.03	0.05	0.30	0.86	0.76
	lm						0.05	0.23	0.82	0.75
	elanet							0.21	0.84	0.80
	rf								1.11	0.84
	gbrt									-0.15
BAB α_{FF5}	ols_1y_d	-2.18	1.04	-0.21	0.52	0.03	0.13	0.72	1.33	1.06
	bsw		2.84	1.22	1.65	0.77	1.00	2.08	2.56	2.19
	hybrid			-1.07	-0.21	-0.50	-0.45	-0.06	0.66	0.46
	fama-french				0.63	0.15	0.27	0.77	1.32	1.09
	long-memo					-0.35	-0.26	0.15	0.78	0.57
	lm						0.26	0.52	1.07	0.89
	elanet							0.45	1.08	0.90
	rf								1.17	0.74
	gbrt									-0.27

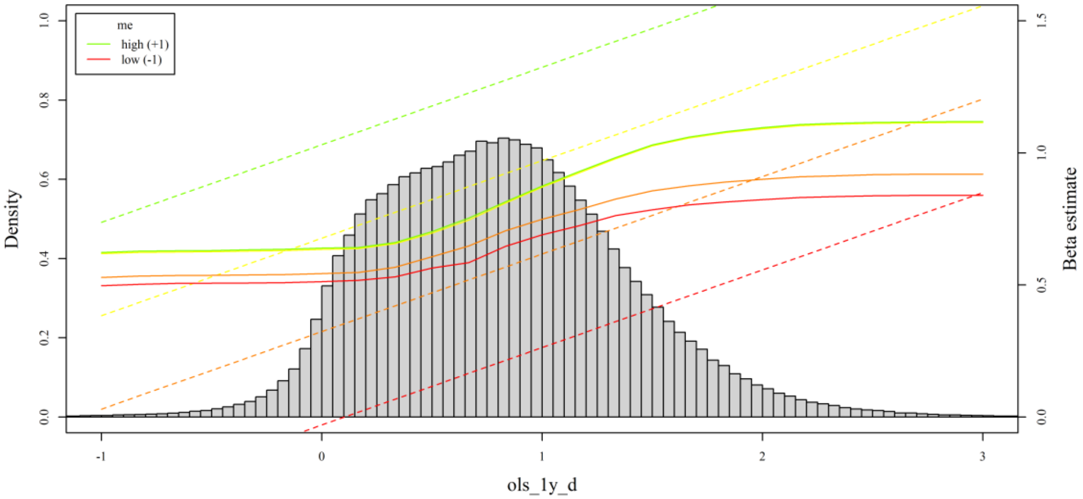
Figure C1
Nonlinear and interactive effects in estimating future market betas

This figure illustrates the ability to capture nonlinear and interactive effects in estimating future market betas for both random forests and simple linear regressions (*rf* and *lm*, introduced in Section IV.B of the main paper). Panel A shows the marginal association between a firm's sample beta estimate from rolling regressions using a one-year window of daily returns (*ols_1y_d*) and its beta estimates ($\beta^{F}_{it+k|t}$, with $k = 12$). To visualize the average effect of *ols_1y_d* on $\beta^{F}_{it+k|t}$, all predictors are set to their uninformative median values within the training sample at each re-estimation date, and the industry dummies are set to zero. *ols_1y_d* is then varied over the interval $(-1, +3)$ and the beta estimates are computed. Finally, the beta estimates are averaged over all re-estimation dates. This visualization is accompanied by a histogram showing the historical distribution of *ols_1y_d*. Panel B shows the interactive effect of *ols_1y_d* and firm size (*me*) on $\beta^{F}_{it+k|t}$. For this purpose, the procedure described above is repeated. In this case, however, the beta estimates are computed for different levels of *me* over the interval $(-1, +1)$. Low and high levels of *me* are marked with red and green lines, respectively. The sample includes all firms that were or are listed on the NYSE, AMEX, or NASDAQ in any month during the sample period from March 1970 to December 2020, while the first beta estimates are obtained in December 1979.

Panel A



Panel B



References

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Wu, G. (2006). Realized Beta: Persistence and Predictability. In T. Fomby, and D. Terrel, *Advances in Econometrics: Econometric Analysis of Economic and Financial Time Series.* Amsterdam, Netherlands: Elsevier.

Becker, J., Hollstein, F., Prokopczuk, M., and Sibbertsen, P. (2021). The Memory of Beta. *Journal of Banking and Finance, 124*(1), 106026.

Black, F., Jensen, M. C., and Scholes, M. S. (1972). The Capital Asset Pricing Model: Some Tests. In M. C. Jensen, *Studies in the Theory of Capital Markets.* New York (NY), U.S.: Praeger.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32.

Cosemans, M., Frehen, R., Schotman, P. C., and Bauer, R. (2016). Estimating Market Betas Using Prior Information Based on Firm Fundamentals. *The Review of Financial Studies, 29*(4), 1072–1112.

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

Dietterich, T. (2000). Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science: Multiple Classifier Systems*. Berlin, Germany: Springer.

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

Drobetz, W., and Otto, T. (2021). Empirical Asset Pricing via Machine Learning: Evidence from the European Stock Market. *Journal of Asset Management, 22*(7), 507–538.

Fama, E. F., and French, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance, 47*(2), 427–465.

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.

Fama, E. F., and French, K. R. (1997). Industry Costs of Equity. *Journal of Financial Economics, 43*(2), 153–193.

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

Frazzini, A., and Pedersen, L. H. (2014). Betting Against Beta. *Journal of Financial Economics, 111*(1), 1–25.

Giacomini, R., and White, H. (2006). Tests of Conditional Predictive Ability. *Econometrica, 74*(6), 1545–1578.

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

Hansen, L. K., and Salamon, P. (1990). Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(10), 993–1001.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The Model Confidence Set. *Econometrica*, 79(2), 453–497.

Hollstein, F., Prokopczuk, M., and Wese Simen, C. (2019). Estimating Beta: Forecast Adjustments and the Impact of Stock Characteristics for a Broad Cross-Section. *Journal of Financial Markets, 44*(1), 91–118.

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

Jagannathan, R., and Wang, Z. (1996). The Conditional CAPM and the Cross-Section of Expected Returns. *The Journal of Finance, 51*(1), 3–53.

Karolyi, G. A. (1992). Predicting Risk: Some New Generalizations. *Management Science, 38*(1), 57–74.

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

Masters, T. (1993). *Practical Neural Network Recipes in C++*. Burlington (MA), U.S.: Morgan Kaufmann Publishers.

Mincer, J. A., and Zarnowitz, V. (1969). The Evaluation of Economic Forecasts. In J. A. Mincer, *Economic Forecasts and Expectations: Analysis of Forecasting Behavior and Performance.* Cambridge (MA), U.S.: NBER.

Newey, W. K., and West, K. D. (1987). A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica 55*(3), 703–708.

Novy-Marx, R. (2011). Operating Leverage. *Review of Finance, 15*(1), 103–134.

Novy-Marx, R., and Velikov, M. (2022). Betting Against Betting Against Beta. *Journal of Financial Economics, 143*(1), 80–106.

Pastor, L., and Stambaugh, R. F. (1999). Costs of Equity Capital and Model Mispricing. *The Journal of Finance, 54*(1), 67–121.

Petkova, R., and Zhang, L. (2005). Is Value Riskier than Growth? *Journal of Financial Economics, 78*(1), 187–202.

Vasicek, O. (1973). A Note on Using Cross-Sectional Information in Bayesian Estimation of Market Betas. *The Journal of Finance, 28*(5), 1233–1239.

Welch, I. (2022), Simply Better Market Betas, *Critical Finance Review, 11*(1), 37–64.

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