The Anatomy of Out-of-Sample Forecasting Accuracy

Daniel Borup
Aarhus University and CREATES
dborup@econ.au.dk

Philippe Goulet Coulombe
Université du Québec à Montréal
p.gouletcoulombe@gmail.com

David E. Rapach*
Federal Reserve Bank of Atlanta
dave.rapach@gmail.com

Erik Christian Montes Schütte
Aarhus University, CREATES, and DFI
christianms@econ.au.dk

Sander Schwenk-Nebbe
Aarhus University
sandersn@econ.au.dk

May 31, 2023

*Corresponding author. Send correspondence to David Rapach, Research Department, Federal Reserve Bank of Atlanta, 1000 Peachtree Street NE, Atlanta, GA 30309; e-mail: dave.rapach@gmail.com. The views expressed here are those of the authors and not necessarily those of the Federal Reserve Bank of Atlanta or the Federal Reserve System. Any remaining errors are the authors' responsibility. We thank seminar and conference participants at the European Commission Joint Research Center: Online Seminar, 2022 International Symposium on Forecasting, Workshop on Advances in Alternative Data and Machine Learning for Macroeconomics and Finance, and Federal Reserve Bank of Atlanta, as well as Daniele Bianchi, Giulio Caperna, Todd Clark, Marco Colagrossi, Claudia Foroni (Workshop on Advances in Alternative Data and Machine Learning discussant), Nikolay Gospodinov, Andreas Joseph, Juri Marcucci, Michael McCracken, Marcelo Medeiros, Stig Møller, and Mirco Rubin, for insightful comments. We created the Python package anatomy to compute the metrics for interpreting fitted prediction models developed in this paper.

The Anatomy of Out-of-Sample Forecasting Accuracy

Abstract

We develop metrics based on Shapley values for interpreting time-series forecasting models in macroeconomics and finance, including "black-box" models from machine learning. Our metrics are model agnostic, so they are applicable to any model (linear or nonlinear, parametric or nonparametric). Two of the metrics, iShapley-VI and oShapley-VI, measure the importance of individual predictors for explaining the in-sample and out-of-sample predicted target values, respectively, in the sequence of fitted models that generates the time-series forecasts. The third metric is the performance-based Shapley value (PBSV), our main methodological contribution. PBSV measures the contributions of individual predictors to the out-of-sample loss corresponding to the time-series forecasts generated by the sequence of fitted models. In essence, PBSV anatomizes out-of-sample forecasting accuracy. In an empirical application forecasting US inflation with a large dataset and machine learning models, leading predictors for improving out-of-sample forecasting accuracy according to PBSV include the price of oil at short horizons and the medical services component of the consumer price index at longer horizons. We also find a number of discrepancies between an individual predictor's relevance according to the in-sample iShapley-VI and the out-of-sample PBSV, so a predictor's in-sample importance does not necessarily capture its relevance for out-of-sample forecasting accuracy. We created the `Python` package `anatomy` for computing the out-of-sample PBSV.

Keywords: Model interpretation, Shapley value, Loss function, Machine learning, Inflation

JEL classifications: C22, C45, C52, C53, E31, E37

1. Introduction

The use of large datasets (i.e., "big data") and machine learning for out-of-sample time-series forecasting in macroeconomics and finance is burgeoning. Indeed, there is growing evidence that the combination of large datasets and machine learning significantly improves out-of-sample performance. Macroeconomic applications include forecasting inflation, output and employment growth, the unemployment rate, unemployment insurance initial claims, and recessions.[1] Applications in finance often involve forecasting stock returns.[2] Large datasets allow researchers to draw on a wealth of information, thereby increasing the capacity of prediction models to incorporate relevant signals. Machine learning offers a variety of tools for guarding against overfitting, which is vital for improving out-of-sample performance in the presence of a large number of predictors.[3] Some classes of machine learning models (e.g., random forests, boosted trees, and neural networks) also accommodate general forms of nonlinearities in predictive relations, further increasing the scope for improving out-of-sample performance when nonlinearities are an important attribute of the data-generating process.[4]

While researchers are certainly concerned with improving out-of-sample forecasting accuracy, they are also keenly interested in interpreting fitted prediction models. For example,

[1]See, for example, Li and Chen (2014), Exterkate et al. (2016), Medeiros and Mendes (2016), Döpke, Fritsche, and Pierdzioch (2017), Kim and Swanson (2018), Smeekes and Wijler (2018), Medeiros et al. (2021), Vrontos, Galakis, and Vrontos (2021), Yousuf and Ng (2021), Borup and Schütte (2022), Goulet Coulombe et al. (2022), Hauzenberger, Huber, and Klieber (2023), and Borup, Rapach, and Schütte (forthcoming).

[2]See, for example, Chinco, Clark-Joseph, and Ye (2019), Rapach et al. (2019), Freyberger, Neuhierl, and Weber (2020), Gu, Kelly, and Xiu (2020), Kozak, Nagel, and Santosh (2020), Bryzgalova, Pelger, and Zhu (2021), Cong et al. (2022), Dong et al. (2022), Avramov, Cheng, and Metzker (forthcoming), and Chen, Pelger, and Zhu (forthcoming).

[3]Stock and Watson (2002a,b) spurred a literature that uses large datasets for macroeconomic forecasting based on principal component regression (e.g., Stock and Watson 1999b; Bernanke and Boivin 2003; Banerjee and Marcellino 2006). Applications in finance that forecast stock and bond returns based on large datasets and principal component regression include Ludvigson and Ng (2007, 2009), Neely et al. (2014), Çakmaklı and van Dijk (2016), and Dong et al. (2022).

[4]Earlier studies that investigate nonlinear approaches to macroeconomic modeling and forecasting include Lee, White, and Granger (1993), Kuan and White (1994), Swanson and White (1997), Stock and Watson (1999a), Trapletti, Leisch, and Hornik (2000), Nakamura (2005), Medeiros, Teräsvirta, and Rech (2006), and Marcellino (2008).

especially with a large number of predictors, it is important to identify which predictors are the most important for determining the forecasts generated by fitted models. It is also valuable to know how the predictors contribute to out-of-sample forecasting accuracy. Such knowledge helps users of forecasting models to wrap their minds around the models so that they are not simply "black boxes" that opaquely transform predictors into forecasts. By identifying the most relevant predictors in fitted models that perform well out of sample, researchers gain insight into empirically important economic mechanisms that can help to guide the assessment and development of theoretical models. In a similar vein, researchers involved in policy need to be able to interpret forecasting models to provide more comprehensible advice to policymakers.

An array of tools have been developed for interpreting fitted prediction models. Many are model agnostic, so they can be applied to any model. One set of tools analyzes how the predictions generated by fitted models vary with the individual predictors. Such methods include partial dependence plots (Friedman 2001), Shapley values (Shapley 1953; Štrumbelj and Kononenko 2010, 2014; Lundberg and Lee 2017), individual conditional expectation plots (Goldstein et al. 2015), locally interpretable model-agnostic explanations (Ribeiro, Singh, and Guestrin 2016), and accumulated local effects (Apley and Zhu 2020). A related set of tools measures variable importance, namely, how important individual predictors are in accounting for the predictions produced by fitted models. Variable importance metrics include those based on partial dependence plots (Greenwell, Boehmke, and McCarthy 2018), permutations (Fisher, Rudin, and Dominici 2019), and Shapley values (Lundberg and Lee 2017; Casalicchio, Molnar, and Bischl 2018).

Tools for interpreting fitted forecasting models are typically applied in a manner appropriate for cross-sectional data. Specifically, a researcher divides the total sample of observations into training and test samples. The researcher then fits a prediction model using data from the training sample and uses the fitted model to generate predictions for the test sample observations. To interpret the model that generates the forecasts, the researcher computes,

for example, the variable importance for each predictor based on the fitted model and training data used to estimate the model. This conventional approach is eminently reasonable, especially in a cross-sectional context.[5] However, it is not necessarily appropriate in a time-series setting. In such a setting, a researcher typically re-estimates the prediction model each period using an expanding or rolling window of data, as they generate a sequence of out-of-sample forecasts. Thus, instead of a single model, there is a sequence of estimated models to interpret. The importance of the predictors in explaining the sequence of out-of-sample forecasts is also likely to be of interest. Moreover, because researchers are concerned with out-of-sample performance, they will be interested in understanding how the individual predictors contribute to out-of-sample forecasting accuracy for the sequence of forecasts.

In this paper, we propose metrics for interpreting time-series forecasting models. The metrics are based on Shapley values. Using insights from coalitional game theory, Shapley values fairly allocate contributions among predictors and have attractive properties for analyzing predictor relevance (as discussed in Section 2). The first metric is iShapley-VI$_p$, an in-sample variable importance measure for predictor p. This is an aggregate measure of an individual predictor's importance across the entire set of fitted models that generate the sequence of out-of-sample time-series forecasts. The next metric is oShapley-VI$_p$, which measures the importance of predictor p for the sequence of out-of-sample forecasts.

The final metric is the *performance-based Shapley value* (PBSV$_p$), our main methodological contribution. The iShapley-VI$_p$ and oShapley-VI$_p$ metrics do not take into account the realized target value; in contrast, PBSV$_p$ measures the contribution of predictor p to the out-of-sample loss for the forecast evaluation period (although it can also be computed for any subsample of the forecast evaluation period, including for a single observation), thereby taking into account the realized target value. In essence, PBSV$_p$ anatomizes out-of-sample forecasting accuracy. PBSV$_p$ applies to any loss function, including the popular mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE)

[5]For example, this approach is used on numerous occasions for the applications in the insightful textbook by Molnar (2022).

criteria. All of our metrics are model agnostic, so they can be applied to any forecasting model (linear or nonlinear, parametric or nonparametric). In sum, our metrics provide an informative set of tools for interpreting time-series forecasting models in macroeconomics and finance. To facilitate their implementation, we develop computationally efficient algorithms for computing oShapley-VI$_p$ and PBSV$_p$.

We illustrate the use of iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ in an empirical application forecasting US inflation. Inflation forecasting is the subject of a sizable literature (for a survey, see Faust and Wright 2013) and an important topic in many contexts, including for central banks when setting monetary policy. A spate of recent studies finds that nonlinear machine learning models, including random forests and neural networks, significantly improve inflation forecasts (e.g., Medeiros et al. 2021; Goulet Coulombe 2022; Goulet Coulombe et al. 2022; Hauzenberger, Huber, and Klieber 2023). We generate inflation forecasts using a set of approximately 120 predictors—primarily from the FRED-MD database (McCracken and Ng 2016)—and a variety of models based on principal component regression (PCR, Stock and Watson 2002a,b), elastic net (ENet, Zou and Hastie 2005) estimation of a linear model, random forests (Breiman 2001), XGBoost (Chen and Guestrin 2016), and neural networks. We also consider ensembles of individual forecasts generated by different models. Our forecasting models consistently outperform a standard autoregressive (AR) benchmark at horizons ranging from one to twelve months, in line with the recent literature.

Applying our metrics to the fitted forecasting models, we make a number of findings. First, there is considerable overlap between the importance of the individual predictors based on iShapley-VI$_p$ and oShapley-VI$_p$. This is perhaps not surprising, as the fitted models used in determining the importance of individual predictors for the in-sample and out-of-sample predicted target values are the same. Similarly, for numerous predictors, we find a relatively close correspondence between the in-sample iShapley-VI$_p$ and the out-of-sample PBSV$_p$, so predictors that are important in the fitted prediction model often also improve the accuracy of the out-of-sample forecasts, which we expect for a model that forecasts well.

However, in a number of cases, we find substantive discrepancies between the relevance of individual predictors according to iShapley-VI$_p$ and PBSV$_p$. Specifically, some predictors that are among the most important according to iShapley-VI$_p$ contribute adversely to out-of-sample forecasting accuracy according to PBSV$_p$. The discrepancies between iShapley-VI$_p$ and PBSV$_p$ in our empirical application serve as a warning: the in-sample importance of a predictor in determining the predicted target values does not necessarily align with the predictor's role in determining out-of-sample forecasting accuracy, even when a forecasting model performs well.

The remainder of the paper is organized as follows. Section 2 describes the iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ metrics for analyzing predictor relevance in a time-series context. Section 3 presents the empirical application forecasting US inflation. Section 4 concludes. We created the `Python` package `anatomy` to implement algorithms for computing oShapley-VI$_p$ and PBSV$_p$.

2. Methodology

This section describes our methodology for measuring the relevance of individual predictors in time-series forecasting models. We begin with a discussion of Shapley values (Shapley 1953) in a time-series context, as they form the foundation for our approach. We then define in-sample and out-of-sample variable importance measures based on Shapley values. Finally, we propose PBSV$_p$ for analyzing the contributions of predictors to out-of-sample forecasting accuracy.

We use the following notation in our time-series context. We index individual predictors by p and collect the predictors in the index set $S = \{1, \ldots, P\}$. The period-t P-dimensional vector of predictor observations is denoted by $\boldsymbol{x}_t = [\ x_{1,t} \quad \cdots \quad x_{P,t}\ ]'$. The prediction model is given by

$$y_{t+1:t+h} = f(\boldsymbol{x}_t) + \varepsilon_{t+1:t+h}, \tag{1}$$

where $y_{t+1:t+h} = (1/h)\sum_{k=1}^{h} y_{t+k}$ is the target, h is the forecast horizon, f is the conditional mean (i.e., prediction) function, and $\varepsilon_{t+1:t+h}$ is a zero-mean disturbance term.[6] We denote the fitted prediction model by $\hat{f}$, while $W_i = \{t_{i,\text{start}}, \ldots, t_{i,\text{end}}\}$ denotes the set of observations used to train the model based on window W_i. The fitted prediction model evaluated at $\boldsymbol{x}_t$ and trained using W_i for horizon h is denoted by $\hat{f}(\boldsymbol{x}_t\,; W_i, h)$.

2.1. Shapley Values in a Time-Series Context

Shapley values draw on coalitional game theory to utilize the analogy between the predictors (or features) in a model and players in a cooperative game earning payoffs, where an individual predictor's payoff corresponds to its contribution to the model's prediction. In a time-series setting, the aim of a Shapley value is to quantify the marginal contribution of predictor $x_{p,t}$ to the prediction $\hat{f}(\boldsymbol{x}_t\,; W_i, h)$, given the presence of all of the other predictors $(S \setminus \{p\})$. Viewed through the lens of coalitional game theory, Shapley values provide a means for fairly allocating the contributions among predictors (even in the presence of correlated predictors and interactions between them in the fitted model).

Adapting Štrumbelj and Kononenko (2010, 2014) to our time-series context, the Shapley value for predictor p and instance $\boldsymbol{x}_t$ for a model trained using window W_i for horizon h is given by

$$\phi_p(\boldsymbol{x}_t\,; W_i, h) = \sum_{Q \subseteq S \setminus \{p\}} \frac{|Q|!(P - |Q| - 1)!}{P!} \big[\xi_{Q \cup \{p\}}(\boldsymbol{x}_t\,; W_i, h) - \xi_Q(\boldsymbol{x}_t\,; W_i, h)\big] \tag{2}$$

for $p \in S$ and $t \in W_i$, where Q is a subset of predictors (i.e., a coalition), $Q \subseteq S \setminus \{p\}$ is the set of all possible coalitions of $P - 1$ predictors in S that exclude predictor p, $|Q|$ is the cardinality of Q, $|Q|!(P - |Q| - 1)!/P!$ is a combinatorial weight,

$$\xi_Q(\boldsymbol{x}_t\,; W_i, h) = \mathbb{E}\Big[\hat{f} \,\Big|\, X_{j,t} = x_{j,t}\, \forall\, j \in Q\,; W_i, h\Big] \tag{3}$$

[6] It is straightforward to extend the notation to allow for the conditional mean function in Equation (1) to include additional lags of $\boldsymbol{x}_t$.

is the value function, and $\mathbb{E}$ is the expectation operator. The value function $\xi_Q(\boldsymbol{x}_t\,; W_i, h)$ in Equation (3) is the prediction of the fitted model conditional on the predictors in coalition Q, so $\xi_{Q\cup\{p\}}(\boldsymbol{x}_t\,; W_i, h) - \xi_Q(\boldsymbol{x}_t\,; W_i, h)$ in Equation (2) measures the change in the prediction, conditional on the predictors in coalition Q, when the predictor p is included in the conditioning information set. The difference $\xi_{Q\cup\{p\}}(\boldsymbol{x}_t\,; W_i, h) - \xi_Q(\boldsymbol{x}_t\,; W_i, h)$ is computed for all possible coalitions of $P-1$ predictors that exclude predictor p, with each quantity receiving the weight $|Q|!(P-|Q|-1)!/P!$ in the summation in Equation (2) (the weights sum to one). In essence, the Shapley value uses coalitions to control for the other predictors when measuring the contribution of predictor p to the prediction corresponding to instance $\boldsymbol{x}_t$.

The Shapley value in Equation (2) has a number of attractive properties in our time-series context. The first is efficiency, also known as local accuracy:

$$\sum_{p\in S}\phi_p(\boldsymbol{x}_t\,; W_i, h) = \hat{f}(\boldsymbol{x}_t\,; W_i, h) - \mathbb{E}\Big[\hat{f}\,; W_i, h\Big], \tag{4}$$

where $\mathbb{E}[\hat{f}\,; W_i, h]$ is the baseline prediction, which corresponds to the unconditional expectation of $\hat{f}$ (i.e., the prediction based on the empty coalition set). Equation (4) says that we can exactly decompose the model prediction corresponding to instance $\boldsymbol{x}_t$ (in terms of the deviation from the baseline prediction) into the sum of the Shapley values for the individual predictors for that instance. Two additional properties, missingness and symmetry, are intuitively appealing. Missingness is given by

$$\forall\, R\subseteq S\setminus\{p\} : \xi_{R\cup\{p\}}(\boldsymbol{x}_t\,; W_i, h) = \xi_R(\boldsymbol{x}_t\,; W_i, h) \Rightarrow \phi_p(\boldsymbol{x}_t\,; W_i, h) = 0, \tag{5}$$

while symmetry is given by

$$\forall\, R\subseteq S\setminus\{p,q\} : \xi_{R\cup\{p\}}(\boldsymbol{x}_t\,; W_i, h) = \xi_{R\cup\{q\}}(\boldsymbol{x}_t\,; W_i, h) \Rightarrow \phi_p(\boldsymbol{x}_t\,; W_i, h) = \phi_q(\boldsymbol{x}_t\,; W_i, h). \tag{6}$$

Finally, linearity says that for any real numbers c_1 and c_2 and models $\hat{f}(\boldsymbol{x}_t\,; W_i, h)$ and $\hat{f}'(\boldsymbol{x}_t\,; W_i, h)$,

$$\phi_p\Big(c_1\Big[\hat{f}(\boldsymbol{x}_t\,; W_i, h) + c_2\hat{f}'(\boldsymbol{x}_t\,; W_i, h)\Big]\Big) = c_1\phi_p\Big(\hat{f}(\boldsymbol{x}_t\,; W_i, h)\Big) + c_1c_2\phi_p\Big(\hat{f}'(\boldsymbol{x}_t\,; W_i, h)\Big). \quad (7)$$

Linearity is useful for computing Shapley values for ensembles of prediction models.

It is practically infeasible to compute the exact Shapley value in Equation (2) for even a moderate number of predictors, as the prediction function has to be evaluated for all possible coalitions both with and without predictor p. Building on the sampling-based approach of Castro, Gómez, and Tejada (2009), Štrumbelj and Kononenko (2014) develop an algorithm for estimating the Shapley value. We use a refined version of their algorithm. We first express Equation (2) in the equivalent form:

$$\phi_p(\boldsymbol{x}_t\,; W_i, h) = \frac{1}{P!}\sum_{\mathcal{O}\,\in\,\pi(P)}\Big[\xi_{\mathrm{Pre}_p(\mathcal{O})\,\cup\,\{p\}}(\boldsymbol{x}_t\,; W_i, h) - \xi_{\mathrm{Pre}_p(\mathcal{O})}(\boldsymbol{x}_t\,; W_i, h)\Big] \quad (8)$$

for $p \in S$ and $t \in W_i$, where $\mathcal{O}$ is an ordered permutation for the predictor indices in S, $\pi(P)$ is the set of all ordered permutations for S, and $\mathrm{Pre}_p(\mathcal{O})$ is the set of indices that precede p in $\mathcal{O}$.

The algorithm is based on making a random draw m with replacement for an ordered permutation from $\pi(P)$, which we denote by $\mathcal{O}_m$. Using $\mathcal{O}_m$, we compute the following:

$$\begin{aligned}\theta_{p,m}(\boldsymbol{x}_t\,; W_i, h) = \frac{1}{|W_i|}\sum_{s\,\in\,W_i}\Big[&\hat{f}(\boldsymbol{x}_{j,t} : j \in \mathrm{Pre}_p(\mathcal{O}_m)\cup\{p\},\, \boldsymbol{x}_{k,s} : k \in \mathrm{Post}_p(\mathcal{O}_m)\,; W_i, h) - \\ &\hat{f}(\boldsymbol{x}_{j,t} : j \in \mathrm{Pre}_p(\mathcal{O}_m),\, \boldsymbol{x}_{k,s} : k \in \mathrm{Post}_p(\mathcal{O}_m)\cup\{p\}\,; W_i, h)\Big]\end{aligned} \quad (9)$$

for $p \in S$ and $t \in W_i$, where $\mathrm{Post}_p(\mathcal{O})$ is the set of indices that follow p in $\mathcal{O}$. Equation (9) approximates the effect of removing predictors not in the coalition by replacing them with background data from the training sample (Štrumbelj and Kononenko 2014; Lundberg and Lee 2017). "Background data" refer to the data used to integrate out the predictors not in

the coalition when estimating the conditional expectation in Equation (3).[7] The estimate of $\phi_p(\boldsymbol{x}_t\,; W_i, h)$ in Equation (8) is then given by

$$\hat{\phi}_p(\boldsymbol{x}_t\,; W_i, h) = \frac{1}{2M}\sum_{m=1}^{2M}\theta_{p,m}(\boldsymbol{x}_t\,; W_i, h) \tag{10}$$

for $p \in S$ and $t \in W_i$, where M is the number of random draws. To increase computational efficiency, we follow Castro, Gómez, and Tejada (2009) and compute Shapley values for each predictor $p \in S$ for a randomly drawn ordered permutation from $\pi(P)$. In addition, we implement antithetic sampling as a variance-reduction technique by computing $\theta_{p,m}(\boldsymbol{x}_t\,; W_i, h)$ in Equation (9) for the original order of a randomly drawn ordered permutation as well as when the order is reversed (Mitchell et al. 2022). Equation (10) retains the properties in Section 2.1, including efficiency:

$$\sum_{p\,\in\,S}\hat{\phi}_p(\boldsymbol{x}_t\,; W_i, h) = \hat{f}(\boldsymbol{x}_t\,; W_i, h) - \underbrace{\bar{\hat{f}}(W_i, h)}_{\hat{\phi}_\emptyset(W_i,h)} \tag{11}$$

for $t \in W_i$, where $\bar{\hat{f}}(W_i, h) = (1/|W_i|)\sum_{t\,\in\,W_i}\hat{f}(\boldsymbol{x}_t\,; W_i, h)$ is the average in-sample prediction for the model trained using sample W_i, which corresponds to the baseline or unconditional forecast (i.e., the forecast based on the empty coalition set, which we denote by $\hat{\phi}_\emptyset(W_i, h)$).

Suppose that the prediction model is linear in the predictors: $f(\boldsymbol{x}_t) = \alpha + \sum_{p=1}^{P}\beta_p x_{p,t}$; the fitted prediction model is given by $\hat{f}(\boldsymbol{x}_t) = \hat{\alpha} + \sum_{p=1}^{P}\hat{\beta}_p x_{p,t}$, where $\hat{\alpha}, \hat{\beta}_1, \ldots, \hat{\beta}_P$ are estimates of $\alpha, \beta_1, \ldots, \beta_P$, respectively. In this case, the Shapley value in Equation (8) is

[7] Equation (9) effectively samples from the empirical marginal distribution based on the training sample for the predictors not in the coalition, which implicitly assumes that the predictors not in the coalition are distributed independently of those in the coalition. Because this assumption is not likely to hold in practice, Lundberg and Lee (2017) propose sampling from the empirical conditional distribution for the predictors not in the coalition. Using insights from Pearl (2009), however, Janzing, Minorics, and Blöbaum (2020) argue that, to fairly allocate the contributions across the individual predictors, it is more appropriate to use the empirical marginal distribution, as in Equation (9).

given by

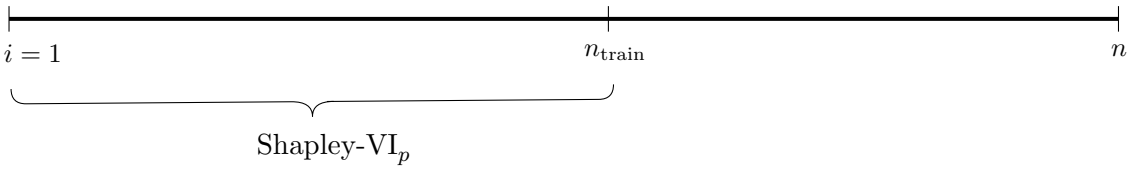
$$\hat{\phi}_p(\boldsymbol{x}_t\,; W_i, h) = \hat{\beta}_p(x_{p,t} - \bar{x}_p) \tag{12}$$

for $p \in S$ and $t \in W_i$, where $\bar{x}_p$ is the sample mean of $x_{p,t}$ for the training sample. Because there are no interactions for a linear model, it is straightforward to compute Shapley values via Equation (12).

The Shapley value $\hat{\phi}_p(\boldsymbol{x}_t\,; W_i, h)$ provides a local measure of the contribution of predictor p to the prediction corresponding to instance $\boldsymbol{x}_t$ in the training sample. A global measure of the importance of predictor p for the training sample can be computed by taking the average of the absolute values of the Shapley values for predictor p across the training sample observations:

$$\text{Shapley-VI}_p(W_i, h) = \frac{1}{|W_i|} \sum_{t \in W_i} \left| \hat{\phi}_p(\boldsymbol{x}_t\,; W_i, h) \right| \tag{13}$$

for $p \in S$. The variable importance measure in Equation (13) is a popular metric for assessing predictor importance in machine learning applications (e.g., Molnar 2022, Chapter 9.6). Equation (13) is based on a single training sample. Tools for interpreting fitted models are typically applied in this manner, which is appropriate for cross-sectional data (or time-series data if a researcher only estimates the prediction model once). The following diagram illustrates the conventional case for cross-sectional data indexed by $i = 1, \ldots, n$, where the first n_{train} observations comprise the training sample.



In a time-series context, however, researchers typically re-estimate the model on a regular basis over time as additional data become available, so there are multiple training samples. In Section 2.2, we develop a variable importance metric more suited to this practice.

2.2. In-Sample Shapley Values for Time Series

When forecasting time-series variables in macroeconomics and finance, it is common to regularly retrain the prediction model using data available at the time of forecast formation. For example, if we are forecasting a monthly variable at horizon h, we re-estimate the prediction model each month as additional data become available, which is typically done using either an expanding or rolling window, where the estimation sample becomes longer (remains the same size) for the former (latter). Suppose that there are $t = 1, \ldots, T$ total observations available. The initial in-sample period ends in $t = T_{\text{in}}$, while the remaining $T - T_{\text{in}} = D$ observations constitute the out-of-sample period.

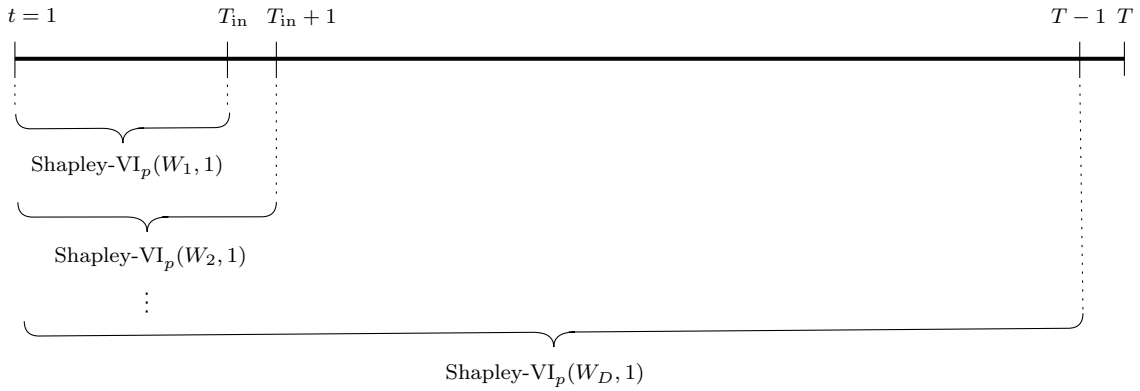
Mimicking the situation of a forecaster in real time, we proceed as follows. We first use data from $t = 1$ through $t = T_{\text{in}}$ to fit the prediction model and generate an out-of-sample forecast of $y_{T_{\text{in}}+1:T_{\text{in}}+h}$. After accounting for the forecast horizon and lag in Equation (1), there are $T_{\text{in}} - (h-1) - 1$ usable observations for training the prediction model for the first out-of-sample forecast. For an expanding (rolling) window, we then use data from $t = 1$ ($t = 2$) through $T_{\text{in}} + 1$ to fit the prediction model and generate a forecast of $y_{T_{\text{in}}+2:T_{\text{in}}+h+1}$. Continuing in this manner, we generate a sequence of $D - (h-1)$ out-of-sample forecasts, where, for the final forecast, we use data from the first period (period $T - D - (h-1)$) through $T - h$ for an expanding (rolling) window to fit the prediction model and generate a forecast of $y_{T-(h-1):T}$. Note that we only use data available at the time of forecast formation to train the model so that there is no "look-ahead" bias in the out-of-sample forecasts. We denote the sequence of time-series forecasts by $\hat{y}_{T_{\text{in}}+1:T_{\text{in}}+h}$, $\hat{y}_{T_{\text{in}}+2:T_{\text{in}}+h+1}, \ldots,$ $\hat{y}_{T-(h-1):T}$.

The Shapley-based variable importance in Equation (13) corresponds to a prediction model trained once using the observations in W_i. To accommodate the sequence of $D-(h-1)$ time-series forecasts for models regularly retrained with an expanding or rolling window, we denote the set of training samples by $W = \{W_1, \ldots, W_{D-(h-1)}\}$. In this context, we define

the *in-sample Shapley-based variable importance* as

$$\text{iShapley-VI}_p(W, h) = \frac{1}{|W|} \sum_{i \in W} \text{Shapley-VI}_p(W_i, h) \tag{14}$$

for $p \in S$, which is the average of the variable importance measures for predictor p across all of the training samples used to generate the sequence of time-series forecasts. To help make the temporal dimension of Equation (14) clear, the following diagram shows how $\text{iShapley-VI}_p(W, h)$ is computed in terms of the time-series observations for an expanding window and $h = 1$.



$$\text{iShapley-VI}_p(W, 1) = \frac{1}{D} \sum_{i=1}^{D} \text{Shapley-VI}_p(W_i, 1)$$

2.3. Out-of-Sample Shapley Values for Time Series

We are also interested in measuring variable importance for the sequence of out-of-sample forecasts. We begin by defining the Shapley value for the fitted model and vector of predictors used to generate an out-of-sample forecast, which corresponds to an out-of-sample version of Equation (8):

$$\phi_p^{\text{out}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right) = \frac{1}{P!} \sum_{\mathcal{O} \in \pi(P)} \left[\xi_{\text{Pre}_p(\mathcal{O}) \cup \{p\}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right) - \xi_{\text{Pre}_p(\mathcal{O})}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right)\right] \tag{15}$$

for $p \in S$ and $i = 1, \ldots, D-(h-1)$, where $\boldsymbol{x}_{T_{\text{in}}+(i-1)}$ is the vector of predictors plugged into the fitted prediction model trained with W_i used to generate the ith out-of-sample forecast, which is given by $\hat{y}_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} = \hat{f}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big)$. To estimate Equation (15), we use a suitably modified version of the algorithm in Section 2.1. For a random draw m of an ordered permutation $\mathcal{O}_m$, we modify Equation (9) to

$$\begin{aligned}\theta^{\text{out}}_{p,m}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big) = \\ \frac{1}{|W_i|}\sum_{s\,\in\, W_i}\Big[\hat{f}\big(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m)\cup\{p\},\, \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m)\,; W_i, h\big) - \\ \hat{f}\big(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m),\, \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m)\cup\{p\}\,; W_i, h\big)\Big],\end{aligned} \tag{16}$$

while Equation (10) becomes

$$\hat{\phi}^{\text{out}}_p\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big) = \frac{1}{2M}\sum_{m=1}^{2M}\theta^{\text{out}}_{p,m}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big) \tag{17}$$

for $p \in S$ and $i = 1, \ldots, D-(h-1)$. Equation (16) continues to approximate the effect of removing predictors not in the coalition by replacing them with background data from W_i, as this is the sample used to train the prediction model that generates the out-of-sample forecast; in this sense, we remain "true to the model" used for forecasting.[8]

The $\hat{\phi}^{\text{out}}_p\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big)$ estimate in Equation (17) continues to be characterized by efficiency, so we can decompose the out-of-sample forecast corresponding to $\boldsymbol{x}_{T_{\text{in}}+(i-1)}$ as follows:

$$\sum_{p\,\in\, S}\hat{\phi}^{\text{out}}_p\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big) = \hat{f}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h\big) - \hat{\phi}_{\emptyset}(W_i, h) \tag{18}$$

[8] "True to the model" means that we use parameter estimates from the fitted prediction model and background data from the training sample used to fit the prediction model. In other words, we retain the basic elements of the fitted model when estimating the Shapley value in Equation (17).

for $i = 1, \ldots, D - (h-1)$. For a model that is linear in the predictors, the Shapley value in Equation (15) is given by

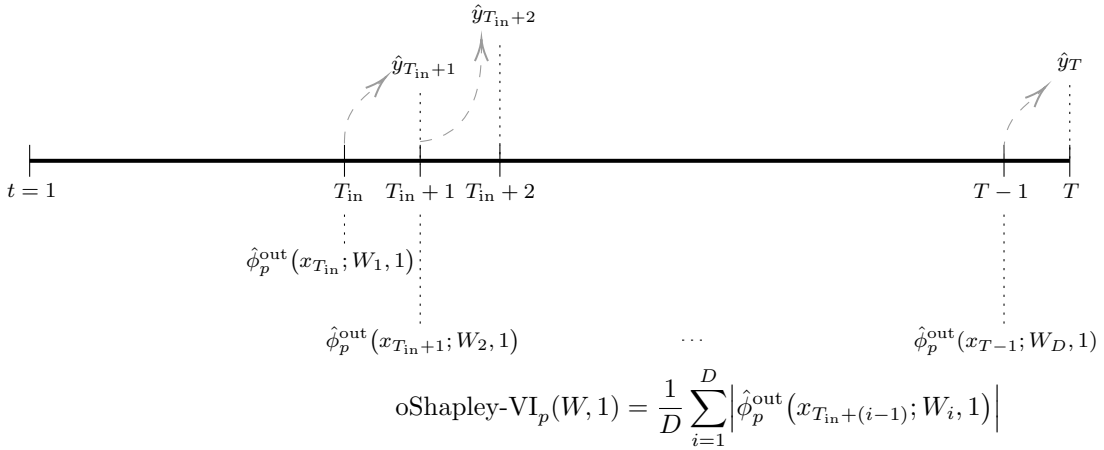
$$\hat{\phi}_p^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\big) = \hat{\beta}_p\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big) \tag{19}$$

for $p \in S$ and $i = 1, \ldots, D - (h-1)$, where $\hat{\beta}_p$ and $\bar{x}_p$ are again the estimate of β_p and sample mean of $x_{p,t}$, respectively, based on the training sample.

Taking the absolute value of $\hat{\phi}_p^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\big)$ in Equation (17) produces a Shapley-based variable importance measure for predictor p and a particular out-of-sample forecast. To compute the variable importance for p for the entire sequence of out-of-sample forecasts, we proceed analogously to the in-sample Shapley-based variable importance in Equation (14) and define the *out-of-sample Shapley-based variable importance* by taking the average of the absolute values of Equation (17) across the out-of-sample forecasts:

$$\text{oShapley-VI}_p(W, h) = \frac{1}{|W|} \sum_{i \in W} \Big| \hat{\phi}_p^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\big) \Big| \tag{20}$$

for $p \in S$. The following diagram depicts how the time-series observations are incorporated into Equation (20) for an expanding window and $h = 1$.



2.4. Performance-Based Shapley Values

Out-of-sample forecasts are typically assessed using a loss function. Accordingly, we propose PBSV_p to decompose the loss over the out-of-sample period into the components attributable to the individual predictors $p \in S$.

The key insight for computing PBSV_p is to wrap a loss function around the predictions in Equation (16). We denote a generic loss function by

$$L\Big(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \hat{f}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\big)\Big) \tag{21}$$

for $i = 1, \ldots, D-(h-1)$. To incorporate the loss function, we further modify the algorithm. For a random draw m of an ordered permutation $\mathcal{O}_m$, we adjust Equation (16) as follows:

$$\begin{aligned}
&\theta^{\text{out}}_{p,m}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h, L\big) = \\
&L\left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \frac{1}{|W_i|}\sum_{s\in W_i} \hat{f}\big(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m)\cup\{p\}, \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m); W_i, h\big)\right) - \\
&L\left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \frac{1}{|W_i|}\sum_{s\in W_i} \hat{f}\big(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m), \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m)\cup\{p\}; W_i, h\big)\right)
\end{aligned} \tag{22}$$

for $p \in S$ and $i = 1, \ldots, D-(h-1)$. Equation (17) becomes

$$\hat{\phi}^{\text{out}}_p\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h, L\big) = \frac{1}{2M}\sum_{m=1}^{2M} \theta^{\text{out}}_{p,m}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h, L\big) \tag{23}$$

for $p \in S$ and $i = 1, \ldots, D-(h-1)$. The local PBSV_p in Equation (23) measures the contribution of predictor p to the loss incurred by the ith out-of-sample forecast. Like Equation (16), Equation (22) approximates the effect of removing predictors not in the coalition by replacing them with background data from the training sample W_i so that we continue to remain true to the model that generates the out-of-sample forecast. Based on the logic of Shapley values, the local PBSV_p in Equation (23) fairly allocates the loss among

the predictors for the ith out-of-sample forecast. Equation (23) is characterized by efficiency:

$$\sum_{p \in S} \hat{\phi}_p^{\text{out}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h, L\right) = L\Big(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \hat{f}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right)\Big) - \hat{\phi}_{\emptyset}^{\text{out}}(W_i, h, L) \quad (24)$$

for $i = 1, \ldots, D - (h-1)$, where $\hat{\phi}_{\emptyset}^{\text{out}}(W_i, h, L)$ corresponds to the loss for the baseline or unconditional prediction based on the empty coalition set.

Because the loss function can be nonlinear, for a prediction model that is linear in the predictors, we do not have a simple expression analogous to Equation (12) or Equation (19) for the local PBSV$_p$. Nevertheless, in the special case of a linear model, we can derive an analytical expression for the local PBSV$_p$ for a specific loss function. For example, consider the squared error loss for the ith out-of-sample forecast:

$$L\left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \hat{y}_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}\right) = \left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} - \hat{y}_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}\right)^2. \quad (25)$$

For a linear model and Equation (25), the local PBSV$_p$ can be expressed as

$$\begin{aligned}\hat{\phi}_p^{\text{out}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h, \text{SE}\right) = \underbrace{\hat{\beta}_p\left(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\right)}_{\hat{\phi}_p^{\text{out}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right)} \Big[&\left(\hat{y}_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} - y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}\right) \\ &- \left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} - \hat{\phi}_{\emptyset}(W_i, h)\right)\Big],\end{aligned} \quad (26)$$

where $\hat{\phi}_p^{\text{out}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right) = \hat{\beta}_p\left(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\right)$ is from Equation (19). We can view $\hat{\phi}_{\emptyset}(W_i, h)$ in Equation (26) as a naïve forecast that ignores the information in the predictors and simply uses the sample mean of the target for the training sample as the prediction. For squared error loss, the local PBSV$_p$ measures the contribution of predictor p to the squared error for the forecast that incorporates the information in the predictors relative to the squared error for the naïve forecast that ignores the information. In the special case of a linear model, Equation (26) says that $\hat{\phi}_p^{\text{out}}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h, \text{SE}\right)$ is proportional to the error for the forecast based on the set of predictors—after adjusting for the naïve forecast error—where the factor of proportionality is given by $\hat{\beta}_p\left(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\right)$ (i.e., the Shapley

value for predictor p and instance $\boldsymbol{x}_{T_{\text{in}}+(i-1)}$ for a linear model). Furthermore, the sign of $\hat{\phi}_p^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h, \text{SE}\big)$ in Equation (26) depends on the signs of $\hat{\beta}_p\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big)$ and the term in brackets.

To gain some intuition for Equation (26), suppose that the linear model forecast is perfect ($\hat{y}_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} = y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}$); in addition, assume that the realized target value is greater than the naïve forecast ($y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} > \hat{\phi}_{\emptyset}(W_i, h)$), so the term in brackets in Equation (26) is negative. If $\hat{\beta}_p\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big) > 0$, then $\hat{\phi}_p^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h, \text{SE}\big) < 0$. In this case, predictor p contributes to the forecast being higher than the naïve forecast—since $\hat{\beta}_p\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big) > 0$—which is in line with the realized target value being greater than the naïve forecast; accordingly, the local PBSV_p in Equation (26) deems that predictor p contributes to lowering the squared error vis-à-vis the naïve forecast.[9]

We are primarily interested in the performance of the entire sequence of out-of-sample forecasts, so we also define a global PBSV_p. To obtain the global PBSV_p, we again modify the algorithm. Specifically, we expand Equation (22) to reflect the average loss for the out-of-sample period:

$$\theta_{p,m}^{\text{out}}(W, h, L) =$$
$$\frac{1}{|W|}\sum_{i\,\in\,W} L\left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \frac{1}{|W_i|}\sum_{s\,\in\,W_i}\hat{f}\big(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m)\cup\{p\},\, \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m)\,; W_i, h\big)\right) -$$
$$\frac{1}{|W|}\sum_{i\,\in\,W} L\left(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \frac{1}{|W_i|}\sum_{s\,\in\,W_i}\hat{f}\big(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m),\, \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m)\cup\{p\}\,; W_i, h\big)\right) \tag{27}$$

for $p \in S$. To remain true to the model, Equation (27) continues to approximate the effect of removing predictors not in the coalition by replacing them with background data from

[9] Conversely, if $\hat{\beta}_p\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big) < 0$, then $\hat{\phi}_p^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h, \text{SE}\big) > 0$. In this case, although the linear model forecast is perfect, the local PBSV_p deems that predictor p increases the squared error vis-à-vis the naïve forecast, as p contributes to the forecast being below the naïve forecast, while the realized target value is above the naïve forecast. A perfect forecast together with $y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} > \hat{\phi}_{\emptyset}(W_i, h)$ and $\hat{\beta}_p\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big) < 0$ imply that there are one or more other predictors $q \neq p$ for which $\hat{\beta}_q\big(x_{p,T_{\text{in}}+(i-1)} - \bar{x}_p\big) > 0$ and $\hat{\phi}_q^{\text{out}}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)}\,; W_i, h, \text{SE}\big) < 0$, as the other predictors contribute to the forecast being higher than the naïve forecast, ultimately producing the perfect forecast.

the training sample. Equation (23) is now given by

$$\hat{\phi}_p^{\text{out}}(W,h,L) = \frac{1}{2M}\sum_{m=1}^{2M}\theta_{p,m}^{\text{out}}(W,h,L) \tag{28}$$

for $p \in S$. The global PBSV_p in Equation (28) allows us to decompose the average loss for a sequence of out-of-sample forecasts into the contributions of each of the P predictors. In this way, we anatomize out-of-sample performance by fairly assessing how the individual predictors contribute to out-of-sample forecasting accuracy. Equation (28) is again characterized by efficiency:

$$\sum_{p\in S}\hat{\phi}_p^{\text{out}}(W,h,L) = \frac{1}{|W|}\sum_{i\in W}L\Big(y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)}, \hat{f}\big(\boldsymbol{x}_{T_{\text{in}}+(i-1)};W_i,h\big)\Big) - \hat{\phi}_{\emptyset}^{\text{out}}(W,h,L), \tag{29}$$

where $\hat{\phi}_{\emptyset}^{\text{out}}(W,h,L)$ corresponds to the average loss for the sequence of baseline forecasts based on the empty coalition set.[10]

Our PBSV_p bears some resemblance to the Shapley feature importance (SFIMP) metric in Casalicchio, Molnar, and Bischl (2018), as both measures are computed using a loss function for the out-of-sample observations. However, there are important differences between PBSV_p and SFIMP. SFIMP assumes that the prediction model is estimated only once, which is more appropriate for cross-sectional data, while PBSV_p is explicitly designed for time-series data when the out-of-sample forecasts are generated by a sequence of fitted models based on an expanding or rolling window. Furthermore, there are substantive differences in the algorithms used to compute PBSV_p and SFIMP (beyond the fact that the former is based on a sequence of fitted models, while the latter is not). For example, SFIMP uses background data from the test sample to control for predictors not in the coalition when computing Shapley values; in contrast, Equation (27) always uses background data from the training sample so that

[10] In addition to the entire out-of-sample period, PBSV_p in Equation (28) can be computed for any subsample of the forecast evaluation period; for an example, see Figure 4 for the empirical application in Section 3.

we remain true to the fitted models that generate the out-of-sample forecasts.[11] In sum, PBSV_p provides a means for fairly allocating the out-of-sample loss for a sequence of time-series forecasts across the individual predictors, thereby shedding light on the anatomy of out-of-sample forecasting accuracy.

As an example of computing PBSV_p for a specific loss function, consider the RMSE criterion:

$$\text{RMSE} = \left\{ \frac{1}{|W|} \sum_{i \in W} \left[y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} - \hat{f}\left(\boldsymbol{x}_{T_{\text{in}}+(i-1)}; W_i, h\right) \right]^2 \right\}^{0.5}. \tag{30}$$

To obtain the global PBSV_p for the RMSE using the algorithm, we use the following version of Equation (27):

$$\begin{aligned}
&\theta^{\text{out}}_{p,m}(W, h, \text{RMSE}) = \\
&\left\{ \frac{1}{|W|} \sum_{i \in W} \left[y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} - \frac{1}{|W_i|} \sum_{s \in W_i} \hat{f}\left(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m) \cup \{p\}, \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m); W_i, h\right) \right]^2 \right\}^{0.5} - \\
&\left\{ \frac{1}{|W|} \sum_{i \in W} \left[y_{T_{\text{in}}+i:T_{\text{in}}+h+(i-1)} - \frac{1}{|W_i|} \sum_{s \in W_i} \hat{f}\left(\boldsymbol{x}_{j,T_{\text{in}}+(i-1)} : j \in \text{Pre}_p(\mathcal{O}_m), \boldsymbol{x}_{k,s} : k \in \text{Post}_p(\mathcal{O}_m) \cup \{p\}; W_i, h\right) \right]^2 \right\}^{0.5}.
\end{aligned} \tag{31}$$

for $p \in S$. Equation (28) is then given by

$$\hat{\phi}^{\text{out}}_p(W, h, \text{RMSE}) = \frac{1}{2M} \sum_{m=1}^{2M} \theta^{\text{out}}_{p,m}(W, h, \text{RMSE}) \tag{32}$$

for $p \in S$.[12] According to the efficiency property,

$$\sum_{p \in S} \hat{\phi}^{\text{out}}_p(W, h, \text{RMSE}) = \text{RMSE} - \hat{\phi}^{\text{out}}_{\emptyset}(W, h, \text{RMSE}). \tag{33}$$

[11] PBSV_p has a different focus from the "Shapley regressions" proposed by Joseph (2021). Shapley regressions relate the realized target values to Shapley values for the out-of-sample observations in a linear regression framework.

[12] We use $M = 500$ for the algorithms when computing iShapley-VI_p, oShapley-VI_p, and PBSV_p for the empirical application in Section 3.

2.5. Algorithm

We created the `Python` package `anatomy` to implement the algorithms for computing oShapley-VI$_p$ and PBSV$_p$. The algorithms divide the estimation procedure into two steps: (1) evaluate the fitted models using coalitions of predictors from the sampled permuted orders and store the forecasts; (2) compute the Shapley-based metrics from the stored forecasts. After the models are evaluated in the computationally expensive first step, arbitrary combinations of models and transformations of the forecasts can be evaluated inexpensively in the second step to compute the desired metric. Algorithm 1 provides the structure for the first step. Using the results from the first step, any metric can be computed inexpensively in the second step without the need to rerun the first step. Section A.1 of the Online Appendix provides examples of how to compute oShapley-VI$_p$ in Equation (20), the local PBSV$_p$ for the squared error loss in Equation (23), and the global PBSV$_p$ for the RMSE in Equation (32) from the output of Algorithm 1.

3. Forecasting Inflation

In this section, we use the time-series metrics developed in Section 2 to analyze out-of-sample forecasts of US inflation. Inflation forecasting is an important topic for, among others, policymakers, business managers, and investors. Recent evidence shows that traditional inflation benchmark forecasts can be outperformed by the use of big data in conjunction with machine learning methods and that the outperformance is largely attributable to nonlinearities, especially at long horizons (e.g., Medeiros et al. 2021; Goulet Coulombe 2022; Goulet Coulombe et al. 2022; Hauzenberger, Huber, and Klieber 2023). We forecast inflation using a large dataset and a variety of models.

Algorithm 1: Forecast Evaluation of Permuted Orders of Predictors

Result: $\hat{\boldsymbol{Y}}$: $T \times K \times P \times 2M \times 2$ array of forecasts for T out-of-sample periods and K models evaluated over coalitions of P predictors deactivated and activated in M forward and reversed permuted orders; $\bar{\boldsymbol{Y}}$: $T \times K$ matrix of naïve forecasts (i.e., model evaluations with empty predictor coalitions)

Input: $\hat{\boldsymbol{F}}$: $T \times K$ matrix of forecast functions; $\boldsymbol{X}$: T training data matrices of sizes $\mathcal{T}_t \times P$ for $t = 1, \ldots, T$; $\boldsymbol{\mathcal{X}}$: $P \times T$ out-of-sample data matrix; M: number of ordered permutations to draw from $\pi(P)$

Generate permutation matrix $\boldsymbol{\mathcal{O}}$ of size $M \times P$ containing M permutations of $\{1, \ldots, P\}$

for $t = 1$ **to** T **do** `// loop over out-of-sample periods`

  for $k = 1$ **to** K **do** `// loop over models`

    Store forecast with all predictors deactivated (naïve forecast): $\bar{\boldsymbol{Y}}_{t,k} = \frac{1}{\mathcal{T}_t} \sum_{s=1}^{\mathcal{T}_t} \hat{\boldsymbol{F}}_{t,k}\left(\boldsymbol{X}_{s,\cdot}^{(t)}\right)$

    for $m = 1$ **to** M **do** `// loop over permutations`

      Copy order to preserve it across runs: $\boldsymbol{o} = \{o_1, \ldots, o_P\} = \boldsymbol{\mathcal{O}}_{m,\cdot}$

      for $i \in \{0, 1\}$ **do** `// original and reverse order`

        Copy training data to preserve it across runs: $\boldsymbol{X}^{(m)} = \boldsymbol{X}^{(t)}$

        Initialize previous activation as naïve forecast: $\hat{y}_{\text{pre}} = \bar{\boldsymbol{Y}}_{t,k}$

        for $p \in \{o_1, \ldots, o_P\}$ **do** `// loop over predictors`

          Store forecast with previously activated predictors: $\hat{\boldsymbol{Y}}^{(\boldsymbol{o})}_{t,k,p,iM+m,1} = \hat{y}_{\text{pre}}$

          Activate predictor p in $\boldsymbol{X}^{(m)}$ by setting all elements of column p to $\boldsymbol{\mathcal{X}}_{p,t}$: $\boldsymbol{X}^{(m)}_{\cdot,p} = \boldsymbol{\mathcal{X}}_{p,t}$

          Store forecast with p and previously activated predictors: $\hat{\boldsymbol{Y}}^{(\boldsymbol{o})}_{t,k,p,iM+m,2} = \frac{1}{\mathcal{T}_t} \sum_{s=1}^{\mathcal{T}_t} \hat{\boldsymbol{F}}_{t,k}\left(\boldsymbol{X}_{s,\cdot}^{(m)}\right)$

          Update previous activation for next iteration: $\hat{y}_{\text{pre}} = \hat{\boldsymbol{Y}}^{(\boldsymbol{o})}_{t,k,p,iM+m,2}$

        end

        Reverse $\boldsymbol{o}$ for antithetic sampling

      end

    end

  end

end

3.1. Forecasting Models

Consider the following general prediction model for inflation:

$$\pi_{t+1:t+h} = f\Big(\underbrace{\boldsymbol{\pi}^{\text{AR}}_{t-L:t}, \boldsymbol{w}_t, \boldsymbol{w}_t^{\text{MA}(q)}}_{\boldsymbol{x}_t}\Big) + \varepsilon_{t+1:t+h}, \tag{34}$$

where $\pi_{t+1:t+h} = (1/h)\sum_{k=1}^{h} \pi_{t+k}$, $\pi_t = \log(\text{CPI}_t) - \log(\text{CPI}_{t-1})$, CPI_t is the month-t US consumer price index (CPI), $\boldsymbol{\pi}^{\text{AR}}_{t-L:t} = [\ \pi_t \ \cdots \ \pi_{t-L}\ ]'$ captures the AR component in

inflation, $\boldsymbol{w}_t$ is a vector of predictors, and $\boldsymbol{w}_t^{\mathrm{MA}(q)} = (1/q)\sum_{k=1}^{q} \boldsymbol{w}_{t-(k-1)}$ is a vector of moving averages (MAs) of order q for the predictors in $\boldsymbol{w}_t$. We collect the entire set of predictors in the P-dimensional vector $\boldsymbol{x}_t = [\ \boldsymbol{\pi}_{t-L:t}^{\mathrm{AR}\,\prime} \quad \boldsymbol{w}_t' \quad \boldsymbol{w}_t^{\mathrm{MA}(q)\,\prime}\ ]'$. The inclusion of MAs of the predictors is motivated by Goulet Coulombe et al. (2021), who find that MAs of predictors provide substantive out-of-sample gains for forecasting macroeconomic variables. We set $q = 3$, which allows predictors up to a quarter in the past to affect the prediction. In terms of the AR component, we set $L = 11$, corresponding to twelve lags of inflation in Equation (34). Based on Equation (34), the forecast of $\pi_{t+1:t+h}$ is given by

$$\hat{\pi}_{t+1:t+h} = \hat{f}(\boldsymbol{x}_t), \tag{35}$$

where $\hat{f}$ is the fitted prediction function based on data through t.

A natural starting point for generating an inflation forecast based on $\boldsymbol{x}_t$ is a linear predictive regression:

$$\pi_{t+1:t+h} = \underbrace{\alpha + \boldsymbol{x}_t'\boldsymbol{\beta}}_{f(\boldsymbol{x}_t)} + \varepsilon_{t+1:t+h}, \tag{36}$$

where α is the intercept, and $\boldsymbol{\beta} = [\ \beta_1 \quad \dots \quad \beta_P\ ]'$ is a P-dimensional vector of slope coefficients. It is straightforward to estimate Equation (36) via ordinary least squares (OLS), leading to the forecast:

$$\hat{\pi}_{t+1:t+h}^{\mathrm{OLS}} = \hat{\alpha}^{\mathrm{OLS}} + \boldsymbol{x}_t'\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}, \tag{37}$$

where $\hat{\alpha}^{\mathrm{OLS}}$ and $\hat{\boldsymbol{\beta}}^{\mathrm{OLS}}$ are the OLS estimates of α and $\boldsymbol{\beta}$, respectively, in Equation (36) based on data through t. Although straightforward to compute, the forecast in Equation (37) tends to perform poorly in practice. By construction, OLS maximizes the fit of the model over the training sample, which can result in in-sample overfitting and thus poor out-of-sample performance. Because inflation contains a sizable unpredictable component, the signal-to-

noise ratio is limited, so the forecast in Equation (37) is likely to perform poorly, especially when P is large and the predictors are correlated.

3.1.1. Principal Component Regression

Based on Stock and Watson (2002a,b), an ample literature employs PCR as a dimension-reduction technique for large datasets to forecast macroeconomic variables, including inflation (e.g., Stock and Watson 1999b; Bernanke and Boivin 2003; Banerjee and Marcellino 2006). Let $\boldsymbol{z}_t = [\ z_{1,t} \quad \cdots \quad z_{C,t}\ ]'$ denote the vector containing the first C principal components corresponding to $\boldsymbol{x}_t$, where $C \ll P$. The PCR specification can be expressed as

$$\pi_{t+1:t+h} = \alpha_z + \boldsymbol{z}_t' \boldsymbol{\beta}_z + \varepsilon_{t+1:t+h}, \tag{38}$$

where $\boldsymbol{\beta}_z = [\ \beta_{z,1} \quad \cdots \quad \beta_{z,C}\ ]'$ is a C-dimensional vector of slope coefficients. The forecast corresponding to Equation (38) is given by

$$\hat{\pi}^{\text{PCR}}_{t+1:t+h} = \hat{\alpha}^{\text{OLS}}_z + \hat{\boldsymbol{z}}_t' \hat{\boldsymbol{\beta}}^{\text{OLS}}_z, \tag{39}$$

where $\hat{\alpha}^{\text{OLS}}_z$ and $\hat{\boldsymbol{\beta}}^{\text{OLS}}_z$ are the OLS estimates of α_z and $\boldsymbol{\beta}_z$, respectively, in Equation (38), and $\hat{\boldsymbol{z}}_t$ is the C-dimensional vector of the first C principal components computed from $\boldsymbol{x}_t$, all of which are based on data through t. Because the principal components are linear combinations of the underlying predictors in $\boldsymbol{x}_t$, the PCR forecast itself is linear in the predictors. Intuitively, we extract a limited set of principal components from $\boldsymbol{x}_t$ to estimate the key latent variables that underlie the comovements among the entire set of predictors; the principal components then serve as predictors in a low-dimensional predictive regression with uncorrelated explanatory variables.[13] We select L in $\boldsymbol{\pi}^{\text{AR}}_{t-L:t}$ and C by choosing the combination that maximizes the adjusted R^2 for the training sample (allowing for maximum values of eleven and ten for L and C, respectively).

[13]The principal components are uncorrelated by construction. Following convention, we standardize the predictors (using data through t) before computing the principal components.

3.1.2. Elastic Net

Next, we use the ENet (Zou and Hastie 2005) to estimate the linear predictive regression in Equation (36). The ENet is a refinement of the least absolute shrinkage and selection operator (LASSO, Tibshirani 1996), a seminal machine-learning device for implementing shrinkage. The LASSO and ENet employ penalized regression to shrink the estimated slope coefficients toward zero to guard against overfitting, and there is evidence that penalized regression helps to improve inflation forecasts (e.g., Li and Chen 2014; Medeiros and Mendes 2016; Smeekes and Wijler 2018). The LASSO relies on the ℓ_1 norm in its penalty term, so it can shrink slope coefficients to exactly zero, thereby performing variable selection. A potential drawback to the LASSO is that it tends to arbitrarily select a single predictor from a group of highly correlated predictors. The ENet mitigates this tendency by including both ℓ_1 and ℓ_2 components in its penalty term; the latter is from ridge regression (Hoerl and Kennard 1970).

The objective function for ENet estimation of Equation (36) can be expressed as

$$\underset{\alpha,\boldsymbol{\beta}}{\arg\min}\ \frac{1}{2[t-(h-1)-1]}\left\{\sum_{s=1}^{t-(h-1)-1}\left[\pi_{s+1:s+h}-(\alpha+\boldsymbol{x}_s'\boldsymbol{\beta})\right]^2\right\}+\lambda P_\delta(\boldsymbol{\beta}), \tag{40}$$

where

$$P_\delta(\boldsymbol{\beta})=0.5(1-\delta)\|\boldsymbol{\beta}\|_2^2+\delta\|\boldsymbol{\beta}\|_1; \tag{41}$$

$\lambda \geq 0$ is a hyperparameter that governs the degree of shrinkage; $\|\cdot\|_1$ and $\|\cdot\|_2$ are the ℓ_1 and ℓ_2 norms, respectively; and $0 \leq \delta \leq 1$ is a hyperparameter for blending the ℓ_1 and ℓ_2 components in the penalty term.[14] We follow the recommendation of Hastie, Qian, and Tay (2023) and set $\delta = 0.5$, which they point out results in a stronger tendency to select highly correlated predictors as a group. To tune λ, we use a walk-forward cross-validation

[14]The ENet objective function in Equation (40) reduces to that for OLS when $\lambda = 0$. If $\delta = 1$ ($\delta = 0$), then Equation (40) corresponds to the LASSO (ridge) objective function.

procedure designed for a time-series context. The ENet forecast based on Equation (36) is given by

$$\hat{\pi}_{t+1:t+h}^{\text{ENet}} = \hat{\alpha}^{\text{ENet}} + \boldsymbol{x}_t' \hat{\boldsymbol{\beta}}^{\text{ENet}}, \tag{42}$$

where $\hat{\alpha}^{\text{ENet}}$ and $\hat{\boldsymbol{\beta}}^{\text{ENet}}$ are the ENet estimates of α and $\boldsymbol{\beta}$, respectively, in Equation (36) based on data through t.

3.1.3. Random Forest

Our third strategy employs a random forest (Breiman 2001), a nonlinear machine-learning technique with a strong track record in macroeconomic forecasting (e.g., Medeiros et al. 2021; Borup and Schütte 2022; Goulet Coulombe et al. 2022). Random forests build on regression trees, machine-learning devices for incorporating nonlinearities in a flexible manner via multi-way interactions and higher-order effects of the predictors. A regression tree is constructed by sequentially splitting the predictor space into regions, with the final set of regions referred to as "terminal nodes" or "leaves." The prediction is the average value of the target in a given leaf. We can express the forecast corresponding to a regression tree with U leaves as

$$\hat{\pi}_{t+1:t+h}^{\text{RT}} = \sum_{u=1}^{U} \bar{\pi}_u \mathbf{1}_u(\boldsymbol{x}_t\,;\boldsymbol{\eta}_u), \tag{43}$$

where the indicator function $\mathbf{1}_u(\boldsymbol{x}_t\,;\boldsymbol{\eta}_u) = 1$ if $\boldsymbol{x}_t \in R_u(\boldsymbol{\eta}_u)$ for the uth region denoted by R_u (which is determined by the parameter vector $\boldsymbol{\eta}_u$) and 0 otherwise, and $\bar{\pi}_u$ is the average value of the target observations in R_u for the training sample based on data through t.

A large (or "deep") regression tree is typically able to capture complex nonlinear relations in the data. However, in light of the bias-variance trade-off, it is susceptible to overfitting due to the high variance of the tree. A random forest reduces the variance by averaging forecasts across many deep regression trees, where each tree is constructed based on a bootstrap sample of the original data using a randomly selected subset of the predictors for each split.

By using a randomly selected subset of the predictors, we "decorrelate" the trees to further reduce the variance. Indexing the bootstrap samples by b, the random forest forecast is given by

$$\hat{\pi}^{\text{RF}}_{t+1:t+h} = \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{u=1}^{U}\bar{\pi}_u^{(b)}\mathbf{1}_u^{(b)}(\boldsymbol{x}_t\,;\boldsymbol{\eta}_u)\right], \tag{44}$$

where B is the number of bootstrap samples, and $\bar{\pi}_u^{(b)}$ and $\mathbf{1}_u^{(b)}(\boldsymbol{x}_t\,;\boldsymbol{\eta}_u)$ are the counterparts to $\bar{\pi}_u$ and $\mathbf{1}_u(\boldsymbol{x}_t\,;\boldsymbol{\eta}_u)$, respectively, in Equation (43) for the bth bootstrap sample. We set $B = 500$ and let each tree grow fully deep. The proportion of predictors randomly selected for each split is tuned via a walk-forward cross-validation procedure.

3.1.4. XGBoost

Another strategy for forecasting with a regression tree is a boosted tree, which is based on gradient boosting (Breiman 1997; Friedman 2001), a sequential ensemble method for improving out-of-sample prediction. The basic idea is to fit a prediction function additively:

$$\hat{f}(\boldsymbol{x}_t\,;\hat{\boldsymbol{\eta}}) = \sum_{j=1}^{J}\hat{f}_j(\boldsymbol{x}_t\,;\hat{\boldsymbol{\eta}}_j). \tag{45}$$

Each function $\hat{f}_j(\boldsymbol{x}_t\,;\hat{\boldsymbol{\eta}}_j)$ on the right-hand-side of Equation (45) is a "weak" learner (i.e., a relatively simple model); for a boosted tree, $\hat{f}_j(\boldsymbol{x}_t\,;\hat{\boldsymbol{\eta}}_j)$ corresponds to a fitted tree with a forecast of the form in Equation (43). Relatively simple models help to guard against overfitting; however, they are more likely to be exhibit substantive bias and thus poor fit. Boosting improves the fit by adding another tree that is trained using the residuals from the previous function in the sequence. In sum, boosting entails constructing a sequence of relatively "shallow" trees, which are then combined into an ensemble. While a random forest starts with a deep tree with low bias and uses bagging across a large number of trees to reduce the variance, a boosted tree starts with a shallow tree with low variance and refines the tree to reduce the bias.

Friedman (2002) proposes stochastic gradient boosting to make boosting more robust. Instead of basing each $\hat{f}_j(\boldsymbol{x}_t\,;\hat{\boldsymbol{\eta}}_j)$ in Equation (45) on all of the training data, each element is based on a randomly drawn (without replacement) subsample of the data. We fit boosted trees via stochastic gradient boosting using the popular XGBoost algorithm (Chen and Guestrin 2016), where we tune the hyperparameters for the algorithm using a walk-forward cross-validation procedure.

3.1.5. Neural Network

Our final forecasting model is a feedforward neural network. Neural networks are flexible and powerful machine learning devices that permit general forms of nonlinearities. A neural network contains multiple layers. The first is the input layer, which is comprised of the set of predictors, followed by $L \geq 1$ hidden layers. Each hidden layer l contains P_l neurons, where each neuron takes signals from the neurons in the previous layer to generate a subsequent signal via a nonlinear activation function:

$$h_m^{(l)} = g\left(\omega_{m,0}^{(l)} + \sum_{j=1}^{P_{l-1}} \omega_{m,j}^{(l)} h_j^{(l-1)}\right) \tag{46}$$

for $m = 1, \ldots, P_l$ and $l = 1, \ldots, L$, where $h_m^{(l)}$ is the signal corresponding to the mth neuron in the lth hidden layer[15]; $\omega_{m,0}^{(l)}, \omega_{m,1}^{(l)}, \ldots, \omega_{m,P_{l-1}}^{(l)}$ are weights; and $g(\cdot)$ is the activation function. The output layer is the final layer. It takes the signals from the last hidden layer and converts them into a prediction:

$$\hat{\pi}_{t+1:t+h}^{\text{NN}} = \omega_0^{(L+1)} + \sum_{j=1}^{P_L} \omega_j^{(L+1)} h_j^{(L)}. \tag{47}$$

The activation function determines the strength of the signal passed through the network. For the activation function, we use the popular rectified linear unit (ReLU) function: $g(x) =$

[15]For the first hidden layer, $h_j^{(0)} = x_{j,t}$ for $j = 1, \ldots, P$.

$\max\{x, 0\}$. The interactions in the network and activation function permit complex nonlinearities as the inputs feed through to the hidden layers and finally to the output layer.

Theoretically, a single hidden layer is sufficient for approximating any smooth function (Cybenko 1989; Funahashi 1989; Hornik, Stinchcombe, and White 1989; Hornik 1991; Barron 1994); however, there are potential advantages to including multiple hidden layers in neural networks (Goodfellow, Bengio, and Courville 2016; Rolnick and Tegmark 2018). Determining the neural network architecture (i.e., the number of hidden layers and the number of neurons in each layer) for a given application is largely an empirical matter, and we cannot know that the optimal architecture has been selected (Goodfellow, Bengio, and Courville 2016). Accordingly, we choose an equal-weighted ensemble of two different architectures: a "shallow" neural network with one hidden layer and a "deep" neural network with three hidden layers. We follow a conventional geometric pyramid rule (Masters 1993) in setting the number of neurons in the hidden layers, so the shallow neural network has $\lceil\sqrt{P}\rceil$ neurons in its hidden layer, while the deep neural network has $\lceil P^{3/4}\rceil$, $\lceil P^{2/4}\rceil$, and $\lceil P^{1/4}\rceil$ in its first, second, and third hidden layers, respectively.

We fit the neural networks (i.e., estimate the weights) by minimizing the training sample MSE using the Adam stochastic gradient descent algorithm (Kingma and Ba 2015). To reduce the influence of the random number generator in the initialization of the weights when fitting the neural networks, we fit each model 199 times with a different seed each time and use the median of the predictions.[16]

[16] Although the Adam algorithm is a powerful optimizer, it is our experience that neural networks at times get stuck near local minima. Using the median of 199 fitted neural networks substantially reduces the influence of local minima in computing the prediction. We fit the neural networks using the `scikit-learn` package in `Python`. We augment the objective function with an ℓ_2 penalty term and set the hyperparameter for the ℓ_2 penalty term to 0.0001 in the `MPLregressor` function. The batch size and number of epochs are set to 32 and 1,000, respectively.

3.1.6. Ensembles

We also consider ensembles of models, which are popular in the machine-learning literature. An ensemble forecast can be straightforwardly computed as a simple average of the forecasts generated by the models in the ensemble.[17] We construct three ensembles:

Ensemble-linear Average of the PCR and ENet forecasts.

Ensemble-nonlinear Average of the random forest, XGBoost, and neural network forecasts.

Ensemble-all Average of the PCR, ENet, random forest, XGBoost, and neural network forecasts.

3.2. Data

We measure inflation based on the US CPI. CPI data are from the FRED database at the Federal Reserve Bank of St. Louis (ticker `CPIAUCSL`). The predictors are from two data sources. We use a set of 118 predictors from the FRED-MD database (McCracken and Ng 2016), which is employed by a number of recent macroeconomic forecasting studies (e.g., Kotchoni, Leroux, and Stevanovic 2019; Medeiros et al. 2021; Borup and Schütte 2022; Goulet Coulombe et al. 2022; Hauzenberger, Huber, and Klieber 2023). We also include three predictors from the University of Michigan Survey of Consumers.[18] The sample period covers 1960:01 to 2022:12. We specify 1960:01 to 1989:12 as the initial in-sample period and compute out-of-sample forecasts for 1990:01 to 2022:12. As in Medeiros et al. (2021), among others, we generate out-of-sample inflation forecasts using a rolling estimation window.

[17] The algorithm for computing PBSV_p straightforwardly accommodates ensemble forecasts (as shown in Section A.1 of the Online Appendix), including those that use data-driven methods to select the combining weights (e.g., Gospodinov and Maasoumi 2021).

[18] Section A.2 of the Online Appendix provides a complete list of the inflation predictors.

3.3. Results

An AR model of order k serves as the benchmark, where we determine k recursively using the Bayesian information criterion (BIC, Schwarz 1978), considering a maximum value of twelve. Like the models in Section 3.1, we estimate the AR benchmark model via a rolling window. The AR model is a standard benchmark in the macroeconomic forecasting literature, including for inflation (e.g., Kotchoni, Leroux, and Stevanovic 2019; Medeiros et al. 2021). It is designed to account for the evident persistence in inflation.

Table 1. Out-of-Sample Forecasting Results

The table reports the root mean squared error (RMSE) for an autoregressive benchmark forecast and RSME ratios for the competing forecasts in Section 3.1 vis-à-vis the autoregressive benchmark forecast for inflation for the 1990:01 to 2022:12 out-of-sample period and the forecast horizon (h) in the column heading. The Diebold and Mariano (1995) and West (1996) statistic is used to test the null hypothesis that the benchmark forecast MSE is less than or equal to the competing forecast MSE against the (one-sided, upper tail) alternative hypothesis that the benchmark forecast MSE is greater than the competing forecast MSE; *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

(1) Forecast	(2) $h = 1$	(3) $h = 3$	(4) $h = 6$	(5) $h = 12$
Autoregressive benchmark RMSE	0.26%	0.23%	0.20%	0.16%
Principal component regression	1.08	1.01	0.96	0.92**
Elastic net	0.93**	0.95*	0.96	0.94
Random forest	0.96	0.97	0.92*	0.82***
XGBoost	1.00	0.98	0.91**	0.85***
Neural network	0.94**	0.93**	0.94	0.83***
Ensemble-linear	0.96	0.96	0.93*	0.90**
Ensemble-nonlinear	0.93**	0.93**	0.90**	0.81***
Ensemble-all	0.93**	0.93**	0.90**	0.84***

We evaluate the forecasts using the RMSE criterion. Table 1 reports results for the accuracy of the inflation forecasts for horizons of one, three, six, and twelve months. The table provides the RMSE for the AR benchmark forecast as well as the RMSE ratio for each of the competing models in Section 3.1 vis-à-vis the AR benchmark. We use the Diebold and Mariano (1995) and West (1996) statistic to test the null hypothesis that the MSE (in

population) for the AR benchmark forecast is less than or equal to that for the competing forecast against the (one-sided, upper-tail) alternative that the AR forecast MSE is greater than the competing forecast MSE.[19]

The RMSE for the AR benchmark forecast decreases monotonically with the horizon from 0.26% ($h = 1$) to 0.16% ($h = 12$) in Table 1. At the one-month horizon in the second column, six of the eight competing forecasts deliver a lower RMSE than the AR benchmark (the exceptions are PCR and XGBoost), and the improvement in forecasting accuracy is statistically significant for the ENet, neural network, ensemble-nonlinear, and ensemble-all forecasts. The ENet, ensemble-nonlinear, and ensemble-all forecasts provide the largest improvements in accuracy, each with an RMSE ratio of 0.93. Seven of the eight competing forecasts outperform the AR benchmark at the three-month horizon in the third column. The improvements are again significant for the ENet, neural network, ensemble-nonlinear, and ensemble-all forecasts. The biggest gain is accuracy is for the neural network, ensemble-nonlinear, and ensemble-all forecasts (RMSE ratio of 0.93 for each). The results are fairly similar for the six-month horizon in the fourth column, although now all of the competing forecasts outperform the AR benchmark, and the improvement is significant in five cases (random forest, XGBoost, ensemble-linear, ensemble-nonlinear, and ensemble-all).

The best overall results are for the twelve-month horizon in the last column of Table 1. All eight of the competing forecasts outperform the AR benchmark, and seven of the improvements are significant (the exception is the ENet). The nonlinear forecasts perform very well for $h = 12$, which RMSE reductions of 18%, 15%, and 17% vis-à-vis the AR benchmark for the random forest, XGBoost, and neural network forecasts, respectively. This pattern is consistent with the recent literature that finds that nonlinear machine learning models are particularly useful for forecasting inflation at longer horizons. The ensemble forecasts also perform well in the last column, as each delivers a significant improvement in forecasting ac-

[19]We use a robust standard error (Newey and West 1987) to compute the Diebold and Mariano (1995) and West (1996) statistic, which accounts for the autocorrelation induced by overlapping observations when $h > 1$.

curacy. Reiterating the strong performance of the nonlinear models, the ensemble-nonlinear forecast performs the best at the 12-month horizon, reducing the RMSE by 19% relative to the AR benchmark.

The results in Table 1 show that large datasets and machine learning are an effective combination for improving inflation forecasts, especially at longer horizons. Next, we use the time-series metrics developed in Section 2 to examine the relevance of the predictors in the fitted prediction models, thereby aiding in the interpretation of the models. We are especially interested in the global PBSV_p, as it measures the contribution of a predictor to forecasting accuracy, so we can identify predictors that are most responsible for improvements (as well as deteriorations) in out-of-sample performance.

Figure 1 depicts the iShapley-VI_p in Equation (14), oShapley-VI_p in Equation (20), and PBSV_p based on the RMSE in Equation (32) for the ENet forecast, an example of a linear model. The different panels display results for the different horizons. The predictors on the horizontal axis in each panel are ordered according to iShapley-VI_p. The red bars and black lines correspond to iShapley-VI_p and oShapley-VI_p, respectively, while the green bars correspond to $\hat{\phi}_p^{\text{out}}(W, h, \text{RMSE})$ in Equation (32).[20] To conserve space, the horizontal axis shows the 25 most important and the five least important predictors in descending order based on iShapley-VI_p. The numbers associated with the green bars are rankings for the contributions of the predictors to out-of-sample forecasting accuracy, where predictors with a positive (negative) ranking contribute negatively (positively) to RMSE over the out-of-sample period; for example, a ranking of 1 (-1) signifies the predictor that contributes the most in a positive (negative) sense to out-of-sample forecasting accuracy.[21]

Comparing the red bars with the black lines in Figure 1, there is generally a close correspondence between in-sample and out-of-sample variable importance according to iShapley-VI_p and oShapley-VI_p, respectively. This is perhaps not surprising, as the in-sample and out-

[20] In Figure 1, we sum the Shapley values for each predictor and its corresponding MA(q) term. We also sum the Shapley values for the twelve lags of inflation.

[21] By a positive (negative) contribution to out-of-sample forecasting accuracy, we mean a decrease (increase) in loss.

Figure 1. Variable importance and PBSV: ENet. The figure shows iShapley-VI (left axis), oShapley-VI (left axis), and PBSV (right axis) for the ENet inflation forecast for the 1990:01 to 2022:12 out-of-sample period. The predictors on the horizontal axis are the top 25 and the bottom five predictors ordered according to their importance based on iShapley-VI. The numbers associated with the green bars are rankings of predictors according to PBSV; a positive (negative) ranking indicates predictors that improve (decrease) out-of-sample forecasting accuracy.

of-sample predicted target values are based on the same fitted models when determining the importance of individual predictors. Comparing the red to the green bars, we also see considerable accord across the in-sample iShapley-VI$_p$ and the out-of-sample PBSV$_p$. This is especially evident for the AR component (`ar`), which is the most relevant predictor

according to both iShapley-VI$_p$ and PBSV$_p$ at horizons of three, six, and twelve months; at the one-month horizon, the AR component is the second most relevant predictor based on iShapley-VI$_p$ and PBSV$_p$. Another leading example is the price of oil (`oilpricex`) at the one-month horizon, which is the most important predictor based on both iShapley-VI$_p$ and PBSV$_p$; it is also the fourth (third) most relevant predictor according to iShapley-VI$_p$ (PBSV$_p$) at the three-month horizon. Other predictors that appear relatively important based on both iShapley-VI$_p$ and PBSV$_p$ at various horizons include the durables component of the CPI (`cusr0000sad`), the index of consumer expectations (`soc_ice`), the durable goods component of the personal consumption expenditures price index, (`ddurrg3m086sbea`), the the medical services component of the CPI (`cpimedsl`), and an interest-rate spread (`baaffm`).

However, there are also major points of discord between the in-sample iShapley-VI$_p$ and the out-of-sample PBSV$_p$ in Figure 1. For example, at the one-month horizon, although real personal income (`rpi`) and the materials component of industrial production (`ipmat`) are among the top 25 predictors based based on the in-sample iShapley-VI$_p$, they are the fourth and first predictors, respectively, most responsible for increasing the out-of-sample loss based on PBSV$_p$. Other similar discrepancies at the one-month horizon are the nonrevolving consumer credit to personal income ratio (`conspi`) and the fuels component of industrial production (`ipfuels`). At the three-month horizon, the total reserves of depository institutions (`totresns`), real personal income, new housing permits in the West (`permitw`), and the business equipment component of industrial production (`ipbuseq`) are among the top 25 predictors according to iShapley-VI$_p$, but they are the four predictors most responsible for increasing the out-of-sample loss according to PBSV$_p$. Similar disparities are also evident between iShapley-VI$_p$ and PBSV$_p$ for the business equipment component of industrial production and new housing permits in the West at the six-month horizon, along with housing starts in the Midwest and West (`houstmw` and `houstw`, respectively) and real estate loans (`realln`). Finally, at the twelve-month horizon, notable discrepancies exist for housing starts in the Midwest, new housing permits in the West, real manufacturing and trade industries

sales (`dpcera3m086sbea`), the fuels component of industrial production, and the intermediate materials component of the producer price index (`wpsid61`). Figure A.1 in the Online Appendix provides iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ for the linear PCR forecast. Overall, the results are similar to those in Figure 1, although the discrepancies between oShapley-VI$_p$, and PBSV$_p$ are typically more muted than those for the ENet forecast.

Figure 2. Variable importance and PBSV: neural network. See the notes to Figure 1 with "neural network" replacing "ENet."

Next, we examine iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ for the neural network, a nonlinear model that performs well overall in Table 1. The results are shown in Figure 2.

Overall, the results are reminiscent of those for the ENet in Figure 1. First, iShapley-VI$_p$ and oShapley-VI$_p$ align relatively closely in Figure 2, so there is substantive agreement between the in-sample and out-of-sample measures of variable importance. Second, there is considerable correspondence between the in-sample iShapley-VI$_p$ and the out-of-sample PBSV$_p$ for many predictors. Examples at various horizons include the AR component, the price of oil, the durables component of the CPI, average weekly hours in manufacturing (`awhman`), and the medical services component of the CPI. Third, a number of predictors that appear important according to iShapley-VI$_p$ are nevertheless responsible for decreases in out-of-sample forecasting accuracy according to PBSV$_p$. Such divergences at sundry horizons in Figure 2 include housing starts in the South and Northeast (`housts` and `houstne`, respectively), a pair of interest rate spreads (`tb3smffm` and `tb6smffm`), the Japanese yen-US dollar exchange rate (`exjpusx`), the number of unemployed for 5–14 weeks (`uemp5to14`), and the index of consumer confidence (`soc_icc`). Figures A.2 and A.3 in the Online Appendix report iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ for the nonlinear random forest and XGBoost forecasts, respectively. The pattern of results is similar to that in Figure 2, especially for XGBoost. For the random forest, the agreement between oShapley-VI$_p$ and PBSV$_p$ is stronger than that for the neural network.

Figure 3 depicts iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ for the ensemble-all forecast. Like Figures 1 and 2, iShapley-VI$_p$ and oShapley-VI$_p$ match up quite closely. Relative to Figures 1 and 2, there is generally greater alignment between oShapley-VI$_p$ and PBSV$_p$. There are still a few noteworthy disparities, including housing starts in the West at the one- and six-month horizons, the Japanese yen-US dollar exchange rate at the three-month horizon, and the index of consumer confidence and housing starts in the Midwest at the six-month horizon. Figures A.4 and A.5 in the Online Appendix present iShapley-VI$_p$, oShapley-VI$_p$, and PBSV$_p$ for the ensemble-linear and ensemble-nonlinear forecasts, respectively. The results are similar to those in Figure 3.

Figure 3. Variable importance and PBSV: ensemble-all. See the notes to Figure 1 with "ensemble-all" replacing "ENet."

In sum, PBSV_p quantifies the contributions of predictors to the accuracy of CPI inflation forecasts for the 1990:01 to 2022:12 out-of-sample period. Specifically, it allows us to pinpoint the predictors that play leading roles in accounting for the out-of-sample gains in forecasting accuracy provided by the different models. It also allows us to identify which predictors detract from out-of-sample forecasting accuracy. In a number of cases, we find that predictors that appear important according to the iShapley-VI_p and oShapley-VI_p variable importance measures—which do not take account of the realized target value—actually

lead to increases in the out-of-sample loss. By taking into account both the forecasts and the realized target values embodied in the loss function, PBSV_p allows us to anatomize the out-of-sample forecasting performance of different models in terms of the underlying predictors. As illustrated by our application, it provides a warning to researchers: predictors that appear relevant according to variable importance measures do not necessarily contribute to improvements in out-of-sample forecasting accuracy.

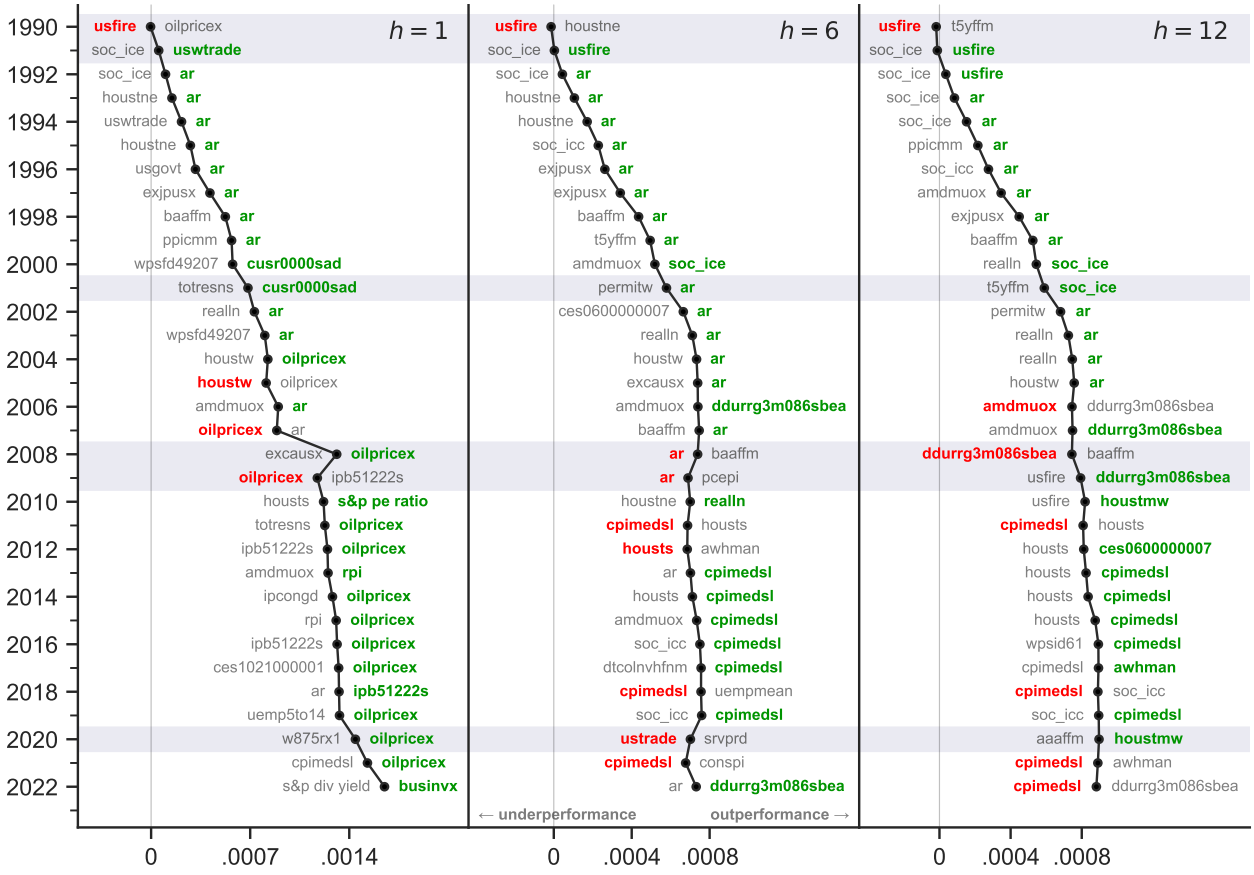


Figure 4. Cumulative difference in squared errors: ensemble-all. The figure shows the cumulative difference in squared errors for a naïve forecast that ignores the information in the predictors vis-à-vis the ensemble-all forecast for the 1990:01 to 2022:12 out-of-sample period. Shifts to the right (left) imply an improvement (deterioration) in forecasting accuracy relative to the naïve forecast. The figure also shows the top (bottom) contributor to the improvement (deterioration) in forecasting performance for non-overlapping twelve-month subsamples; a green (red) color for the predictor indicates that the subsample is associated with an improvement (deterioration) in performance. Horizontal gray bars indicate twelve-month subsamples that contain an NBER-dated recession.

Finally, we illustrate how PBSV_p can shed light on the most relevant predictors with respect to forecasting accuracy for subsamples of the entire sequence of time-series forecasts. This provides a sense of the contributions of predictors to forecasting accuracy over time. Figure 4 plots the cumulative difference in squared errors (CDSE, Goyal and Welch 2008) between a naïve forecast that ignores the information in the predictors and the ensemble-all forecast. To conserve space, we report results for horizons of one, six, and twelve months. The CDSE provides a convenient and informative graphical device for ascertaining whether a competing forecast is more accurate than the naïve forecast for any subsample of the out-of-sample period. In terms of Figure 4, we compare the CDSE at the beginning and end of the interval corresponding to a subsample. If the curve lies more to the right (left) at the end of the interval relative to the beginning, then the ensemble-all (naïve) forecast is more accurate in terms of MSE for the subsample. In addition, we use our procedure in Section 2.4 to compute PBSV_p for the ensemble-all forecast for non-overlapping twelve-month rolling subsamples. The abbreviation to the right (left) of the curve in Figure 4 indicates the predictor that contributes the most to positive (negative) performance during a subsample. A predictor in green (red) to the right (left) of the curve indicates that the ensemble-all forecast delivers a lower (higher) MSE than the naïve forecast for the subsample. The horizontal gray bars indicate twelve-month subsamples that contain an NBER-dated recession.

The CDSE plots in Figure 4 are consistently positively sloped (when viewed from top to bottom), so the ensemble-all forecast outperforms the naïve forecast on a consistent basis over time. For numerous twelve-month periods before the Great Recession in 2008, the AR component is the predictor most responsible for the outperformance of the ensemble-all forecast, consistent with the top and bottom two panels of Figure 3. This highlights the relevance of accounting for inflation persistence when forecasting inflation. Consistent with the top panel of Figure 3, for the one-month horizon in the left panel of Figure 4, there are eleven twelve-month periods for which the price of oil is the predictor most responsible for the outperformance of the ensemble-all forecast, including during the Great Recession and

the recent recession corresponding to the advent of COVID-19 as well as the inflation surge starting in mid 2021. This is consistent with the important influence of energy prices on short-run CPI fluctuations.

The medical services component of the CPI is the leading predictor in terms of the outperformance of the ensemble-all forecast for six and five of the twelve-month subsamples at the six- and twelve-month horizons in the middle and right panels, respectively, of Figure 4. This is consistent with the bottom two panels of Figure 3. Economically, it aligns with Bils and Klenow (2004) and Bryan and Meyer (2010), who rank medical care among the stickiest components of the CPI (in terms of its low frequency of price adjustment), and it is an important component in the Federal Reserve Bank of Atlanta's Sticky-Price CPI. Consistent with the discrepancies between iShapley-VI$_p$ and PBSV$_p$ for the index of consumer confidence in the third row of Figure 3, there are multiple cases in the second column of Figure 4 for which the index of consumer confidence is the predictor that contributes the most to negative performance (i.e., lies to the left of the curve) during a twelve-month period.[22]

4. Conclusion

As large datasets and machine learning become more popular in macroeconomics and finance, researchers are increasingly concerned with interpreting forecasting models fitted with time-series data. While the literature provides a variety of informative tools for interpreting fitted prediction models, existing tools are typically more appropriate for models estimated with cross-sectional data. We develop metrics based on Shapley values for interpreting time-series forecasting models. The metrics recognize that forecasting models are refitted on a regular basis as additional data become available over time. Our iShapley-VI$_p$ and oShapley-VI$_p$ metrics measure the importance of a predictor for explaining the in- and out-of-sample predicted target values, respectively, corresponding to a sequence of fitted prediction models. Our primary methodological contribution is PBSV$_p$, which measures the contribution of a

[22]Figures A6 to A12 in the Online Appendix show CDSE plots for the other forecasts, which are similar to Figure 4.

predictor to the out-of-sample loss. By computing PBSV_p for the set of predictors used to generate a sequence of time-series forecasts, we anatomize the model's out-of-sample forecasting accuracy. Our metrics are flexible: they are model agnostic, so they can be applied to any prediction model (or ensembles of models), and PBSV_p can be applied to any loss function.

We use our metrics to interpret fitted machine learning models used to forecast US inflation based on a large dataset. In line with the recent literature, we find that large datasets in conjunction with machine learning generate significant out-of-sample gains for forecasting inflation. When it comes to model interpretation, iShapley-VI_p and oShapley-VI_p generally paint the same picture in terms of the importance of individual predictors in accounting for the in- and out-of-sample predicted target values, respectively, produced by the sequence of fitted models. The in-sample iShapley-VI_p and the out-of-sample PBSV_p measures also identify similar predictors as being relevant (e.g., the price of oil at short horizons). However, iShapley-VI_p and PBSV_p also detect a number of substantial differences in the rankings of predictors, providing evidence that predictors that are important for determining a model's predicted values are not necessarily those that are primarily responsible for the model's out-of-sample forecasting accuracy. In sum, PBSV_p allows researchers to quantify the roles of predictors in time-series forecasting models along perhaps the most relevant dimension—namely, their contributions to out-of-sample forecasting accuracy.

References

Apley, D. W. and J. Zhu (2020). Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 82:4, 1059–1086.

Avramov, D., S. Cheng, and L. Metzker (forthcoming). Machine Learning Versus Economic Restrictions: Evidence from Stock Return Predictability. *Management Science*.

Banerjee, A. and M. Marcellino (2006). Are There Any Reliable Leading Indicators for US Inflation and GDP Growth? *International Journal of Forecasting* 22:1, 137–151.

Barron, A. R. (1994). Approximation and Estimation Bounds for Artificial Neural Networks. *Machine Learning* 14:1, 115–133.

Bernanke, B. S. and J. Boivin (2003). Monetary Policy in a Data-Rich Environment. *Journal of Monetary Economics* 50:3, 525–546.

Bils, M. and P. J. Klenow (2004). Some Evidence on the Importance of Sticky Prices. *Journal of Political Economy* 112:5, 947–985.

Borup, D., D. E. Rapach, and E. C. M. Schütte (forthcoming). Mixed-Frequency Machine Learning: Nowcasting and Backcasting Weekly Initial Claims with Daily Internet Search Volume Data. *International Journal of Forecasting*.

Borup, D. and E. C. M. Schütte (2022). In Search of a Job: Forecasting Employment Growth Using Google Trends. *Journal of Business & Economic Statistics* 40:1, 186–200.

Breiman, L. (1997). Arcing the Edge. Technical Report 486, Statistics Department, University of California, Berkeley.

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

Bryan, M. F. and B. Meyer (2010). Are Some Prices in the CPI More Forward Looking than Others? We Think So. Federal Reserve Bank of Cleveland Economic Commentary Number 2010-02.

Bryzgalova, S., M. Pelger, and J. Zhu (2021). Forest Through the Trees: Building Cross-Sections of Stock Returns. Working Paper (available at https://ssrn.com/abstract=3493458).

Çakmaklı, C. and D. van Dijk (2016). Getting the Most Out of Macroeconomic Information for Predicting Excess Stock Returns. *International Journal of Forecasting* 32:3, 650–668.

Casalicchio, G., C. Molnar, and B. Bischl (2018). Visualizing the Feature Importance for Black Box Models. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 655–670.

Castro, J., D. Gómez, and J. Tejada (2009). Polynomial Calculation of the Shapley Value Based on Sampling. *Computer and Operations Research* 36:5, 1726–1730.

Chen, L., M. Pelger, and J. Zhu (forthcoming). Deep Learning in Asset Pricing. *Management Science*.

Chen, T. and C. Guestrin (2016). XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Chinco, A., A. D. Clark-Joseph, and M. Ye (2019). Sparse Signals in the Cross-Section of Returns. *Journal of Finance* 74:1, 449–492.

Cong, L. W., K. Tang, J. Wang, and Y. Zhang (2022). AlphaPortfolio: Direct Construction Through Deep Reinforcement Learning and Interpretable AI. Working Paper (available at https://ssrn.com/abstract=3554486).

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems* 2:4, 303–314.

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

Dong, X., Y. Li, D. E. Rapach, and G. Zhou (2022). Anomalies and the Expected Market Return. *Journal of Finance* 77:1, 639–681.

Döpke, J., U. Fritsche, and C. Pierdzioch (2017). Predicting Recessions with Boosted Regression Trees. *International Journal of Forecasting* 33:4, 745–759.

Exterkate, P., P. J. F. Groenen, C. Heij, and D. van Dijk (2016). Nonlinear Forecasting with Many Predictors Using Kernel Ridge Regression. *International Journal of Forecasting* 32:3, 736–753.

Faust, J. and J. H. Wright (2013). Forecasting Inflation. In: G. Elliott and A. Timmermann, eds. *Handbook of Economic Forecasting*. Vol. 2A. Amsterdam: Elsevier, pp. 2–56.

Fisher, A., C. Rudin, and F. Dominici (2019). All Models Are Wrong, But Many Are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research* 20:177, 1–81.

Freyberger, J., A. Neuhierl, and M. Weber (2020). Dissecting Characteristics Nonparametrically. *Review of Financial Studies* 33:5, 2326–2377.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29:5, 1189–1232.

Friedman, J. H. (2002). Stochastic Gradient Boosting. *Computational Statistics & Data Analysis* 38:4, 367–378.

Funahashi, K.-I. (1989). On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks* 2:3, 183–192.

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin (2015). Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics* 24:1, 44–65.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. Boston: MIT Press.

Gospodinov, N. and E. Maasoumi (2021). Generalized Aggregation of Misspecified Models: With an Application to Asset Pricing. *Journal of Econometrics* 222:1B, 451–467.

Goulet Coulombe, P. (2022). A Neural Phillips Curve and a Deep Output Gap. Working Paper arXiv:2202.04146v1.

Goulet Coulombe, P., M. Leroux, D. Stevanovic, and S. Surprenant (2021). Macroeconomic Data Transformations Matter. *International Journal of Forecasting* 37:4, 1338–1354.

Goulet Coulombe, P., M. Leroux, D. Stevanovic, and S. Surprenant (2022). How is Machine Learning Useful for Macroeconomic Forecasting? *Journal of Applied Econometrics* 37:5, 920–964.

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

Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy (2018). A Simple and Effective Model-Based Variable Importance Measure. Working Paper arXiv:1805.04755v1.

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

Hastie, T., J. Qian, and K. Tay (2023). An Introduction to `glmnet`. Manuscript.

Hauzenberger, N., F. Huber, and K. Klieber (2023). Real-Time Inflation Forecasting Using Non-Linear Dimension Reduction Techniques. *International Journal of Forecasting* 39:2, 901–921.

Hoerl, A. E. and R. W. Kennard (1970). Ridge Regression: Applications to Nonorthogonal Problems. *Technometrics* 12:1, 69–82.

Hornik, K. (1991). Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* 4:2, 251–257.

Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* 2:5, 359–366.

Janzing, D., L. Minorics, and P. Blöbaum (2020). Feature Relevance Quantification in Explainable AI: A Causal Problem. In: *23rd International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916.

Joseph, A. (2021). Shapley Regressions: A Framework for Statistical Inference on Machine Learning Models. Working Paper arXiv:1903.04209v1.

Kim, H. H. and N. R. Swanson (2018). Mining Big Data Using Parsimonious Factor, Machine Learning, Variable Selection and Shrinkage Methods. *International Journal of Forecasting* 34:2, 339–354.

Kingma, D. P. and J. Ba (2015). Adam: A Method for Stochastic Optimization. In: *Proceedings of the Third Annual International Conference on Learning Representations*.

Kotchoni, R., M. Leroux, and D. Stevanovic (2019). Macroeconomic Forecast Accuracy in a Data-Rich Environment. *Journal of Applied Econometrics* 34:7, 1050–1072.

Kozak, S., S. Nagel, and S. Santosh (2020). Shrinking the Cross Section. *Journal of Financial Economics* 135:2, 271–292.

Kuan, C.-M. and H. White (1994). Artificial Neural Networks: An Econometric Perspective. *Econometric Reviews* 13:1, 1–91.

Lee, T.-H., H. White, and C. W. J. Granger (1993). Testing for Neglected Nonlinearity in Time Series Models: A Comparison of Neural Network Methods and Alternative Tests. *Journal of Econometrics* 56:3, 269–290.

Li, J. and W. Chen (2014). Forecasting Macroeconomic Time Series: LASSO-Based Approaches and Their Forecast Combinations with Dynamic Factor Models. *International Journal of Forecasting* 30:4, 996–1015.

Ludvigson, S. C. and S. Ng (2007). The Empirical Risk-Return Relation: A Factor Analysis Approach. *Journal of Financial Economics* 83:1, 171–222.

Ludvigson, S. C. and S. Ng (2009). Macro Factors in Bond Risk Premia. *Review of Financial Studies* 22:12, 5027–5067.

Lundberg, S. M. and S.-I. Lee (2017). A Unified Approach to Interpreting Model Predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777.

Marcellino, M. (2008). A Linear Benchmark for Forecasting GDP Growth and Inflation? *Journal of Forecasting* 27:4, 305–340.

Masters, T. (1993). *Practical Neural Network Recipes in C++*. Boston: Academic Press.

McCracken, M. W. and S. Ng (2016). FRED-MD: A Monthly Database for Macroeconomic Research. *Journal of Business & Economic Statistics* 34:4, 574–589.

Medeiros, M. C. and E. F. Mendes (2016). ℓ_1-Regularization of High-Dimensional Time-Series Models with Non-Gaussian and Heteroskedastic Errors. *Journal of Econometrics* 191:1, 255–271.

Medeiros, M. C., T. Teräsvirta, and G. Rech (2006). Building Neural Network Models for Time Series: A Statistical Approach. *Journal of Forecasting* 25:1, 49–75.

Medeiros, M. C., G. F. R. Vasconcelos, Á. Veiga, and E. Zilberman (2021). Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business & Economic Statistics* 39:1, 98–119.

Mitchell, R., J. Cooper, E. Frank, and G. Holmes (2022). Sampling Permutations for Shapley Value Estimation. *Journal of Machine Learning Research* 23:43, 1–46.

Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Independently published.

Nakamura, E. (2005). Inflation Forecasting Using a Neural Network. *Economics Letters* 86:3, 373–378.

Neely, C. J., D. E. Rapach, J. Tu, and G. Zhou (2014). Forecasting the Equity Risk Premium: The Role of Technical Indicators. *Management Science* 60:7, 1772–1791.

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

Pearl, J. (2009). *Causality*. Second Edition. Cambridge: Cambridge University Press.

Rapach, D. E., J. K. Strauss, J. Tu, and G. Zhou (2019). Industry Return Predictability: A Machine Learning Approach. *Journal of Financial Data Science* 1:3, 9–28.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.

Rolnick, D. and M. Tegmark (2018). The Power of Deeper Networks for Expressing Natural Functions. In: *Sixth Annual International Conference on Learning Representations*.

Schwarz, G. (1978). Estimating the Dimension of a Model. *Annals of Statistics* 6:2, 461–464.

Shapley, L. S. (1953). A Value for n-Person Games. *Contributions to the Theory of Games* 2:28, 307–317.

Smeekes, S. and E. Wijler (2018). Macroeconomic Forecasting Using Penalized Regression Methods. *International Journal of Forecasting* 34:3, 408–430.

Stock, J. H. and M. W. Watson (1999a). A Comparison of Linear and Nonlinear Univariate Models for Forecasting Macroeconomic Time Series. In: R. F. Engle and H. White, eds. *Cointegration, Causality and Forecasting: A Festschrift for Clive W. J. Granger*. Oxford: Oxford University Press, pp. 1–44.

Stock, J. H. and M. W. Watson (1999b). Forecasting Inflation. *Journal of Monetary Economics* 44:2, 293–335.

Stock, J. H. and M. W. Watson (2002a). Forecasting Using Principal Components From a Large Number of Predictors. *Journal of the American Statistical Association* 97:460, 1167–1179.

Stock, J. H. and M. W. Watson (2002b). Macroeconomic Forecasting Using Diffusion Indexes. *Journal of Business & Economic Statistics* 20:2, 147–162.

Štrumbelj, E. and I. Kononenko (2010). An Efficient Explanation of Individual Classifications Using Game Theory. *Journal of Machine Learning Research* 11:1, 1–18.

Štrumbelj, E. and I. Kononenko (2014). Explaining Prediction Models and Individual Predictions with Feature Contributions. *Knowledge and Information Systems* 41:1, 647–665.

Swanson, N. R. and H. White (1997). A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks. *Review of Economics and Statistics* 79:4, 540–550.

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

Trapletti, A., F. Leisch, and K. Hornik (2000). Stationary and Integrated Autoregressive Neural Network Processes. *Neural Computation* 12:10, 2427–2450.

Vrontos, S. D., J. Galakis, and I. D. Vrontos (2021). Modeling and Predicting U.S. Recessions Using Machine Learning Techniques. *International Journal of Forecasting* 37:2, 647–671.

West, K. D. (1996). Asymptotic Inference About Predictive Ability. *Econometrica* 64:5, 1067–1084.

Yousuf, K. and S. Ng (2021). Boosting High Dimensional Predictive Regressions with Time Varying Parameters. *Journal of Econometrics* 224:1, 60–87.

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