Opening the Black Box: Machine Learning Interpretability and Inference Tools with an Application to Economic Forecasting

Marcus Buckmann, Andreas Joseph, and Helena Robertson

Abstract We present a comprehensive comparative case study for the use of machine learning models for macroeconomics forecasting. We find that machine learning models mostly outperform conventional econometric approaches in forecasting changes in US unemployment on a 1-year horizon. To address the black box critique of machine learning models, we apply and compare two variables attribution methods: permutation importance and Shapley values. While the aggregate information derived from both approaches is broadly in line, Shapley values offer several advantages, such as the discovery of unknown functional forms in the data generating process and the ability to perform statistical inference. The latter is achieved by the Shapley regression framework, which allows for the evaluation and communication of machine learning models akin to that of linear models.

1 Introduction

Machine learning provides a toolbox of powerful methods that excel in static prediction problems such as face recognition [37], language translation [12], and playing board games [41]. The recent literature suggests that machine learning methods can also outperform conventional models in forecasting problems; see, e.g., [4] for bond risk premia, [15] for recessions, and [5] for financial crises. Predicting macroeconomic dynamics is challenging. Relationships between variables may not hold over time, and shocks such as recessions or financial crises might lead to a breakdown of previously observed relationships. Nevertheless, several studies have shown that machine learning methods outperform econometric baselines in predicting unemployment, inflation, and output [38, 9].

M. Buckmann · A. Joseph (✉)
Bank of England, London, UK
e-mail: marcus.buckmann@bankofengland.co.uk; andreas.joseph@bankofengland.co.uk

H. Robertson
Financial Conduct Authority, London, UK
e-mail: helena.robertson2@fca.org.uk

S. Consoli et al. (eds.), *Data Science for Economics and Finance*,
https://doi.org/10.1007/978-3-030-66891-4_3

While they learn meaningful relationships between variables from the data, these are not directly observable, leading to the criticism that machine learning models such as random forests and neural networks are opaque black boxes. However, as we demonstrate, there exist approaches that can make machine learning predictions transparent and even allow for statistical inference.

We have organized this chapter as a guiding example for how to combine improved performance and statistical inference for machine learning models in the context of macroeconomic forecasting.

We start by comparing the forecasting performance and inference on various machine learning models to more commonly used econometric models. We find that machine learning models outperform econometric benchmarks in predicting 1-year changes in US unemployment. Next, we address the black box critique by using Shapley values [44, 28] to depict the nonlinear relationships learned by the machine learning models and then test their statistical significance [24]. Our method closes the gap between two distinct data modelling objectives, using black box machine learning methods to maximize predictive performance and statistical techniques to infer the data-generating process [8].

While several studies have shown that multivariate machine learning models can be useful for macroeconomic forecasting [38, 9, 31], only a little research has tried to explain the machine learning predictions. Coulombe et al. [13] shows generally that the success of machine learning models in macro-forecasting can be attributed to their ability to exploit nonlinearities in the data, particularly at longer time horizons. However, we are not aware of any macroeconomic forecasting study that attempted to identify the functional form learned by the machine learning models.[1] However, addressing the explainability of models is important when model outputs inform decisions, given the intertwined ethical, safety, privacy, and legal concerns about the application of opaque models [14, 17, 20]. There exists a debate about the level of model explainability that is necessary. Lipton [27] argues that a complex machine learning model does not need to be less interpretable than a simpler linear model if the latter operates on a more complex space, while Miller [32] suggests that humans prefer simple explanations, i.e., those providing fewer causes and explaining more general events—even though these may be biased.

Therefore, with our focus on explainability, we consider a small but diverse set of variables to learn a forecasting model, while the forecasting literature often relies on many variables [21] or latent factors that summarize individual variables [43]. In the machine learning literature, approaches to interpreting machine learning models usually focus on measuring how important input variables are for prediction. These *variable attributions* can be either global, assessing variable importance across the whole data set [23, 25] or local, by measuring the importance of the variables at the level of individual observations. Popular global methods are permutation importance or Gini importance for tree-based models [7]. Popular local methods are

[1] See Bracke et al. [6], Bluwstein et al. [5] for examples that explain machine learning predictions in economic prediction problems.

LIME[2] [34], DeepLIFT[3] [40] and Shapley values [44]. Local methods decompose *individual* predictions into variable contributions [36, 45, 44, 34, 40, 28, 35]. The main advantage of local methods is that they uncover the functional form of the association between a feature and the outcome as learned by the model. Global methods cannot reveal the direction of association between a variable and the outcome of interest. Instead, they only identify variables that are relevant on average across all predictions, which can also be achieved via local methods and averaging attributions across all observations.

For model explainability in the context of macroeconomic forecasting, we suggest that local methods that uncover the functional form of the data generating process are most appropriate. Lundberg and Lee [28] demonstrate that local method Shapley values offer a unified framework of LIME and DeepLIFT with appealing properties. We chose to use Shapely values in this chapter because of their important property of *consistency*. Here, consistency is when on increasing the impact of a feature in a model, the feature's estimated attribution for a prediction does not decrease, independent of all other features. Originally, Shapley values were introduced in game theory [39] as a way to determine the contribution of individual players in a cooperative game. Shapely values estimate the increase in the collective pay-off when a player joins all possible coalitions with other players. Štrumbelj and Kononenko [44] used this approach to estimate the contribution of variables to a model prediction, where the variables and the predicted value are analogous to the players and payoff in a game.

The global and local attribution methods mentioned here are descriptive—they explain the drivers of a model's prediction but they do not assess a model's goodness-of-fit or the predictors' statistical significance. These concepts relate to statistical inference and require two steps: (1) measuring or estimating some quantity, such as a regression coefficient, and (2) inferring how certain one is in this estimate, e.g., how likely is it that the true coefficient in the population is different from zero.

The econometric approach of statistical inference for machine learning is mostly focused on measuring low-dimensional parameters of interest [10, 11], such as treatment effects in randomized experiments [2, 47]. However, in many situations we are interested in estimating the effects for *all* variables included in a model. To the best of our knowledge, there exists only one general framework that performs statistical inference jointly on all variables used in a machine learning prediction model to test for their statistical significance [24]. The framework is called *Shapley regressions*, where an auxiliary regression of the outcome variable on the Shapley values of individual data points is used to identify those variables that significantly improve the predictions of a nonlinear machine learning model. We will discuss this framework in detail in Sect. 4. Before that, we will describe the data and the

[2] Local Interpretable Model-agnostic Explanations.

[3] Deep Learning Important FeaTures for NN.

forecasting methodology (Sect. 2) and present the forecasting results (Sect. 3). We conclude in Sect. 5.

2 Data and Experimental Setup

We first introduce the necessary notation. Let y and $\hat{y} \in \mathbb{R}^m$ be the observed and predicted continuous outcome, respectively, where m is the number of observations in the time series.[4] The feature matrix is denoted by $x \in \mathbb{R}^{m \times n}$, where n is the number of features in the dataset. The feature vector of observation i is denoted by x_i. Generally, we use i to index the point in time of the observation and k to index features. While our empirical analysis is limited to numerical features, the forecasting methods as well as the techniques to interpret their predictions also work when the data contains categorical features. These just need to be transformed into binary variables, each indicating membership of a category.

2.1 Data

We use the *FRED-MD* macroeconomic database [30]. The data contains monthly series of 127 macroeconomic indicators of the USA between 1959 and 2019. Our outcome variable is unemployment and we choose nine variables as predictors, each capturing a different macroeconomic channel. We add the slope of the yield curve as a variable by computing the difference of the interest rates of the 10-year treasury note and the 3-month treasury bill. The authors of the database suggest specific transformations to make each series stationary. We use these transformations, which are (for a variable a:) (1) changes ($a_i - a_{i-l}$), (2) log changes ($\log_e a_i - \log_e a_{i-l}$), and (3) second-order log changes (($\log_e a_i - \log_e a_{i-l}) - (\log_e a_{i-l} - \log_e a_{i-2l}$)). As we want to predict the year-on-year change in unemployment, we set l to 12 for the outcome and the lagged outcome when used as a predictor. For the remaining predictors, we set $l = 3$ in our baseline setup. This generally leads to the best performance (see Table 3 for other choices of l). Table 1 shows the variables, with the respective transformations and the series names in the original database. The augmented Dickey-Fuller test confirms that all transformed series are stationary ($p < 0.01$).

[4] That is, we are in the setting of a regression problem in machine learning speak, while classification problems operate on categorical targets. All approaches presented here can be applied to both situations.

Table 1 Series used in the forecasting experiment. The middle column shows the transformations suggested by the authors of the FRED-MD database and the right column shows the names in that database

Variable	Transformation	Name in the FRED-MD database
Unemployment	Changes	UNRATE
3-month treasury bill	Changes	TB3MS
Slope of the yield curve	Changes	–
Real personal income	Log changes	RPI
Industrial production	Log changes	INDPRO
Consumption	Log changes	DPCERA3M086SBEA
S&P 500	Log changes	S&P 500
Business loans	Second-order log changes	BUSLOANS
CPI	Second-order log changes	CPIAUCSL
Oil price	Second-order log changes	OILPRICEx
M2 Money	Second-order log changes	M2SL

2.2 Models

We test three families of models that can be formalized in the following way assuming that all variables have been transformed according to Table 1.

- The **simple linear lag model** only uses the 1-year lag of the outcome variable as a predictor: $\hat{y}_i = \alpha + \theta_0 y_{i-12}$.
- **The autoregressive model (AR)** uses several lags of the response as predictors: $\hat{y}_i = \alpha + \sum_{l=1}^{h} \theta_i y_{i-l}$. We test AR models with a horizon $1 \leq h \leq 12$, chosen by the Akaike Information Criterion [1].
- The **full information models** use the 1-year lag of the outcome and 1-year lags of the other features as independent variables: $\hat{y}_t = f(y_{i-12}; x_{i-12})$, where f can be any prediction model. For example, if f is a linear regression, $f(y_i, x_i) = \alpha + \theta_0 y_{i-12} + \sum_{k=1}^{n} \theta_k x_{i-12,k}$. To simplify this notation we imply that the lagged outcome is included in the feature matrix x in the following. We test five full information models: Ordinary least squares regression and Lasso regularized regression [46], and three machine learning regressors—random forest [7], support vector regression [16], and artificial neural networks [22].[5]

[5] In machine learning, classification is arguably the most relevant and most researched prediction problem, and while models such as random forests and support vector machines are best known as classification, their variants being used in regression problems are also known to perform well.

2.3 Experimental Procedure

We evaluate how all models predict changes in unemployment 1 year ahead. After transforming the variables (see Table 1) and removing missing values, the first observation in the training set is February 1962. All methods are evaluated on the 359 data points of the forecasts between January 1990 and November 2019 using an expanding window approach. We recalibrate the full information and simple linear lag models every 12 months such that each model makes 12 predictions before it is updated. The autoregressive model is updated every month. Due to the lead-lag structure of the full information and simple linear lag models, we have to create an initial gap between training and test set when making predictions to avoid a look-ahead bias. For a model trained on observations $1 \ldots i$, the earliest observation in the test set that provides a true 12-month forecast is $i + 12$. For observations $i + 1, \ldots, i + 11$, the time difference to the last observed outcome in the training set is smaller than a year.

All machine learning models that we tested have hyperparameters. We optimize their values in the training sets using fivefold cross-validation.[6] As this is computationally expensive, we conduct the hyperparameter search every 36 months with the exception of the computationally less costly Lasso regression, whose hyperparameters are updated every 12 months.

To increase the stability of the full information models, we use bootstrap aggregation, also referred to as bagging. We train 100 models on different bootstrapped samples (of the same size as the training set) and average their predictions. We do not use bagging for the random forest as, by design, each individual tree is already calibrated on a different bootstrapped sample of the training set.

3 Forecasting Performance

3.1 Baseline Setting

Table 2 shows three measures of forecasting performance: the correlation of the observed and predicted response, the mean absolute error (MAE), and the root mean squared error (RMSE). The latter is the main metric considered, as most models minimize RMSE during training. The models are ordered by decreasing RMSE on the whole test period between 1990 and 2019. The random forest performs best and we divide the MAE and RMSE of all models by that of the random forest for ease of comparison.

[6] For the hyperparameter search, we also consider partitionings of the training set that take the temporal dependency of our data into account [3]. We use block cross-validation [42] and hv-block cross-validation [33]. However, both methods do not improve the forecasting accuracy.

Table 2 Forecasting performance for the different prediction models. The models are ordered by decreasing RMSE on the whole sample with the errors of the random forest set to unity. The forest's MAE and RMSE (full period) are 0.574 and 0.763, respectively. The asterisks indicate the statistical significance of the Diebold-Mariano test, comparing the performance of the random forest with the other models, with significance levels $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

	Corr.	MAE	RMSE (normalized by first row)			
		01/1990–11/2019	01/1990–11/2019	01/1990–12/1999	01/2000–08/2008	09/2008–11/2019
Random forest	0.609	1.000	1.000	1.000	1.000	1.000
Neural network	0.555	1.009	1.049	0.969	0.941	1.114**
Linear regression	0.521	1.094***	1.082**	1.011	0.959	1.149***
Lasso regression	0.519	1.094***	1.083***	1.007	0.949	1.156***
Ridge regression	0.514	1.099***	1.087***	1.019	0.952	1.157***
SVR	0.475	1.052	1.105**	1.000	1.033	1.169**
AR	0.383	1.082(*)	1.160(***)	1.003	1.010	1.265(***)
Linear regression (lagged response)	0.242	1.163***	1.226***	1.027	1.057	1.352***

Table 2 also breaks down the performance in three periods: the 1990s and the period before and after the onset of the global financial crisis in September 2008. We statistically compare the RMSE and MAE of the best model, the random forest, against all other models using a Diebold-Mariano test. The asterisks indicate the *p*-value of the tests.[7]

Apart from support vector regression (SVR), all machine learning models outperform the linear models on the whole sample. The inferior performance of SVR is not surprising as it does not minimize a squared error metric such as RMSE but a metric similar to MAE which is lower for SVR than for the linear models. In the 1990s and the periods before the global financial crisis, there are only small differences in performance between the models, with the neural network being the most accurate model. Only after the onset of the crisis does the random forest outperform the other models by a large and statistically significant margin.

Figure 1 shows the observed response variable and the predictions of the random forest, the linear regression, and the AR. The vertical dashed lines indicate the different time periods distinguished in Table 2. The predictions of the random forest are more volatile than that of the regression and the AR.[8] All models underestimate unemployment during the global financial crisis and overestimate it during the recovery. However, the random forest is least biased in those periods and forecasts high unemployment earliest during the crisis. This shows that its relatively high

[7] The horizon of the Diebold-Mariano test is set to 1 for all tests. Note, however, that the horizon of the AR model is 12 so that the *p*-values for this comparison are biased and thus reported in parentheses. Setting the horizon of the Diebold-Mariano test to 12, we do not observe significant differences between the RMSE of the random forest and AR.

[8] The mean absolute deviance from the models' mean prediction are 0.439, 0.356, and 0.207 for the random forest, regression, and AR, respectively.

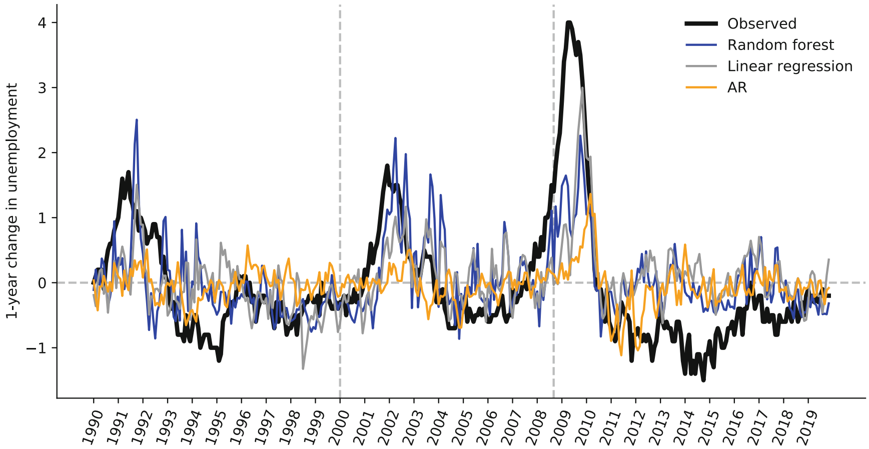

Fig. 1 Observed and predicted 1-year change in unemployment for the whole forecasting period comparing different models

forecast volatility can be useful in registering negative turning points. A similar observation can be made after the burst of the dotcom bubble in 2000. This points to an advantage of machine learning models associated with their greater flexibility incorporating new information as it arrives. This can be intuitively understood as adjusting model predictions locally, e.g., in regions (periods) of high unemployment, while a linear model needs to realign the full (global) model hyperplane.

3.2 Robustness Checks

We altered several parameters in our baseline setup to investigate their effects on the forecasting performance. The results are shown in Table 3. The RMSE of alternative specifications is again divided by the RMSE of the random forest in the baseline setup for a clearer comparison.

- **Window size.** In the baseline setup, the training set grows over time (expanding window). This can potentially improve the performance over time as more observations may facilitate a better approximation of the true data generating process. On the other hand, it may also make the model sluggish and prevent quick adaptation to structural changes. We test sliding windows of 60, 120, and 240 months. Only the simplest model, linear regression with only a lagged response, profits from a short horizon; the remaining models perform best with the biggest possible training set. This is not surprising for machine learning models, as they can "memorize" different sets of information through the incorporation of multiple specification in the same model. For instance, different

Table 3 Performance for different parameter specifications. The shown metric is RMSE divided by the RMSE of the random forest in the baseline setup

	Random forest	Neural network	Linear regression	SVR	AR	Linear regression (lagged response)
Training set size (in months)						
Max (baseline)	1.000	1.049	1.082	1.105	1.160	1.226
60	1.487	1.497	1.708	1.589	2.935	1.751
120	1.183	1.163	1.184	1.248	1.568	1.257
240	1.070	1.051	1.087	1.106	1.304	1.198
Change horizon (in months)						
3 (baseline)	1.000	1.049	1.082	1.105	1.160	1.226
1	1.077	1.083	1.128	1.148	–	–
6	1.043	1.111	1.142	1.162	–	–
9	1.216	1.321	1.251	1.344	–	–
12	1.345	1.278	1.336	1.365	–	–
Bootstrap aggregation						
No	1.000	1.179	1.089	1.117	1.160	1.226
100 models	–	1.049	1.082	1.105	–	–

paths down a tree model, or different trees in a forest, are all different submodels, e.g., characterizing different time periods in our setting. By contrast, a simple linear model cannot adjust in this way and needs to fit the best hyperplane to the current situation, explaining its improved performance for some fixed window sizes.

- **Change horizon.** In the baseline setup, we use a horizon of 3 months, when calculating changes, log changes, and second-order log changes of the predictors (see Table 1). Testing the horizons of 1, 6, 9, and 12 months, we find that 3 months generally leads to the best performance of all full information models. This is useful from a practical point of view, as quarterly changes are one of the main horizons considered for short-term economic projections.
- **Bootstrap aggregation (bagging).** The linear regression, neural network, and SVR all benefit from averaging the prediction of 100 bootstrapped models. The intuition is that our relatively small dataset likely leads to models with high variance, i.e., overfitting. The bootstrap aggregation of models reduces the models' variance and the degree of overfitting. Note that we do not expect much improvement for bagged linear models, as different draws from the training set are likely to lead to similar slope parameters resulting in almost identical models. This is confirmed by the almost identical performance of the single and bagged model.

4 Model Interpretability

4.1 Methodology

We saw in the last section that machine learning models outperform conventional linear approaches in a comprehensive economic forecasting exercise. Improved model accuracy is often the principal reason for applying machine learning models to a problem. However, especially in situations where model results are used to inform decisions, it is crucial to both understand and clearly communicate modelling results. This brings us to a second step when using machine learning models—explaining them.

Here, we introduce and compare two different methods for interpreting machine learning forecasting models *permutation importance* [7, 18] and *Shapley values and regressions* [44, 28, 24]. Both approaches are *model-agnostic*, meaning that they can be applied to *any* model, unlike other approaches, such as Gini impurity [25, 19], which are only compatible with specific machine learning methods. Both methods allow us to understand the relative importance of model features. For permutation importance, variable attribution is at the global level while Shapley values are constructed locally, i.e., for each single prediction. We note that both importance measures require column-wise independence of the features, i.e., contemporaneous independence in our forecasting experiments, an assumption that will not hold under all contexts.[9]

4.1.1 Permutation Importance

The permutation importance of a variable measures the change of model performance when the values of that variable are randomly scrambled. Scrambling or permuting a variable's values can either be done within a particular sample or by swapping values between samples. If a model has learnt a strong dependency between the model outcome and a given variable, scrambling the value of the variable leads to very different model predictions and thus affects performance. A variable k is said to be important in a model, if the test error e after scrambling feature k is substantially higher than the test error when using the original value for k, i.e., $e_k^{perm} >> e$. Clearly, the value of the permutation error e_k^{perm} depends on the realization of the permutation, and variation in its value can be large, particularly in small datasets. Therefore, it is recommended to average e_k^{perm} over several random draws for more accurate estimates and to assess sampling variability.[10]

[9] Lundberg et al. [29] proposed TREESHAP, which correctly estimates the Shapley values when features are dependent for tree models only.

[10] Considering a test set of size m with each observation having a unique value, there are $m!$ permutations to consider for an exhaustive evaluation, which is intractable to compute for larger m.

The following procedure estimates the permutation importance.

1. For each feature x_k:
   (a) Generate a permutation sample x_k^{perm} with the values of x_k permuted across observations (or swapped between samples).
   (b) Reevaluate the test score for x_k^{perm}, resulting in e_k^{perm}.
   (c) The permutation importance of x_k is given by $I(x_k) = e_k^{perm}/e$.[11]
   (d) Repeat and average over Q iterations and average $I_k = 1/Q \sum_q I_q(x_k)$.
2. If I_q is given by the ratio of errors, consider the normalized quantity $\bar{I}_k = (I_k - 1) \sum_k (I_k - 1) \in (0, 1)$.[12]
3. Sort features by I_k (or, $\bar{I}_k$).

Permutation importance is an intuitive measure that is relatively cheap to compute, requiring only new predictions generated on the permuted data and not model retraining. However, this ease of use comes at some cost. First, and foremost, permutation importance is *inconsistent*. For example, if two features contain similar information, permuting either of them will not reflect the actual importance of this feature relative to all other features in the model. Only permuting both or excluding one would do so. This situation is accounted for by Shapley values because they identify the individual marginal effect of a feature, accounting for its interaction with all other features. Additionally, the computation of permutation importance necessitates access to true outcome values and in many situations, e.g., when working with models trained on sensitive or confidential data, these may not be available. As a global measure, permutation importance only explains *which* variables are important but not *how* they contribute to the model, i.e., we cannot uncover the functional form or even the direction of the association between features and outcome that was learned by the model.

4.1.2 Shapley Values and Regressions

Shapley values originate from game theory [39] as a general solution to the problem of attributing a payoff obtained in a cooperative game to the individual players based on their contribution to the game. Štrumbelj and Kononenko [44] introduced the analogy between players in a cooperative game and variables in a general supervised model, where variables jointly generate a prediction, the payoff. The calculation is analogous in both cases (see also [24]),

$$\Phi^S\Big[f(x_i)\Big] \;\equiv\; \phi_0^S + \sum_{k=1}^{n} \phi_k^S(x_i) \;=\; f(x_i)\,, \tag{1}$$

[11] Alternatively, the difference $e_j^{perm} - e$ can be considered.

[12] Note, $I_k \geq 1$ in general. If not, there may be problems with model optimization.

$$\phi_k^S(x_i; f) = \sum_{x' \subseteq \mathcal{C}(x) \setminus \{k\}} \frac{|x'|!(n - |x'| - 1)!}{n!} \left[f(x_i | x' \cup \{k\}) - f(x_i | x') \right], \quad (2)$$

$$= \sum_{x' \subseteq \mathcal{C}(x) \setminus \{k\}} \omega_{x'} \left[\mathbb{E}_b[f(x_i) | x' \cup \{k\}] - \mathbb{E}_b[f(x_i) | x'] \right], \quad (3)$$

$$\text{with} \quad \mathbb{E}_b[f(x_i)|x'] \equiv \int f(x_i)\, \mathrm{d}b(\bar{x}') = \frac{1}{|b|} \sum_b f(x_i|\bar{x}').$$

Equation 1 states that the Shapley decomposition $\Phi^S[f(x_i)]$ of model f is local at x_i and exact, i.e., it precisely adds up to the actually predicted value $f(x_i)$. In Eq. 2, $\mathcal{C}(x) \setminus \{k\}$ is the set of all possible variable combinations (coalitions) of $n - 1$ variables when excluding the k^{th} variable. $|x'|$ denotes the number of variables included in that coalition, $\omega_{x'} \equiv |x'|!(n - |x'| - 1)!/n!$ is a combinatorial weighting factor summing to one over all possible coalition, b is a background dataset, and $\bar{x}'$ stands for the set of variables not included in x'.

Equation 2 is the weighted sum of marginal contributions of variable k accounting for the number of possible variable coalitions.[13] In a general model, it is usually not possible to put an arbitrary feature to missing, i.e., exclude it. Instead, the contributions from features not included in x' are integrated out over a suitable background dataset, where $\{x_i|\bar{x}'\}$ is the set of points with variables not in x' being replaced by values in b. The background provides an informative reference point by determining the intercept ϕ_0^S. A reasonable choice is the training dataset incorporating all information the model has learned from.

An obvious disadvantage of Shapley values compared to permutation importance is the considerably higher complexity of their calculation. Given the factorial in Eq. 2, an exhaustive calculation is generally not feasible with larger feature sets. This can be addressed by either sampling from the space of coalitions or by setting all “not important” variables to “others,” i.e., treating them as single variables. This substantially reduces the number of elements in $\mathcal{C}(x)$.

Nevertheless, these computational costs come with significant advantages. Shapley values are the only feature attribution method which is model independent, local, accurate, linear, and consistent [28]. This means that it delivers a granular high-fidelity approach for assessing the contribution and importance of variables. By comparing the local attributions of a variable across all observations we can visualize the functional form learned by the model. For instance, we might see that observations with a high (low) value on the variable have a disproportionally high (low) Shapley value on that variable, indicating a positive nonlinear functional form.

[13] For example, assuming we have three players (variables) $\{A, B, C\}$, the Shapley value of player C would be $\phi_C^S(f) = 1/3[f(\{A, B, C\}) - f(\{A, B\})] + 1/6[f(\{A, C\}) - f(\{A\})] + 1/6[f(\{B, C\}) - f(\{B\})] + 1/3[f(\{C\}) - f(\{\emptyset\})]$.

Based on these properties, which are directly inherited from the game theoretic origins of Shapley values, we can formulate an inference framework using Eq. 1. Namely, the *Shapley regression* [24],

$$y_i = \sum_{k=0}^{n} \phi_k^S(f, x_i)\beta_k^S + \hat{\epsilon}_i \equiv \Phi_i^S \beta^S + \hat{\epsilon}_i, \tag{4}$$

where $k = 0$ corresponds to the intercept and $\hat{\epsilon}_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The surrogate coefficients β_k^S are tested against the null hypothesis

$$\mathcal{H}_0^k(\Omega) : \{\beta_k^S \leq 0 \,|\, \Omega\}, \tag{5}$$

with $\Omega \in \mathbb{R}^n$ (a region of) the model input space. The intuition behind this approach is to test the alignment of Shapley components with the target variable. This is analogous to a linear model where we use "raw" feature values rather than their associated Shapley attributions. A key difference to the linear case is the regional dependence on Ω. We only make *local* statements about the significance of variable contributions, i.e., on those regions where it is tested against $\mathcal{H}_0$. This is appropriate in the context of potential nonlinearity, where the model plane in the original input-target space may be curved, unlike that of a linear model. Note that the Shapley value decomposition (Eqs. 1–3) absorbs the signs of variable attributions, such that only positive coefficient values indicate significance. When negative values occur, it indicates that a model has poorly learned from a variable and $\mathcal{H}_0$ cannot be rejected.

The coefficients β^S are only informative about variable alignment (the strength of association between the output variable and feature of interest), not the magnitude of importance of a variable. Both together can be summarized by *Shapley share coefficients*,

$$\Gamma_k^S(f, \Omega) \quad \equiv \quad \left[sign\big(\beta_k^{lin}\big) \left\langle \frac{|\phi_k^S(f)|}{\sum_{l=1}^n |\phi_l^S(f)|} \right\rangle_\Omega \right]^{(*)} \in [-1, 1], \tag{6}$$

$$\overset{f(x)=x\beta}{=} \quad \beta_k^{(*)} \quad \left\langle \frac{|(x_k - \langle x_k \rangle)|}{\sum_{l=1}^n |\beta_k (x_l - \langle x_l \rangle)|} \right\rangle_\Omega , \tag{7}$$

where $\langle \cdot \rangle_\Omega$ stands for the average over x_k in $\Omega_k \in \mathbb{R}$. The Shapley share coefficient $\Gamma_k^S(f, \Omega)$ is a summary statistic for the contribution of x_k to the model over a region $\Omega \subset \mathbb{R}^n$ for modelling y.

It consists of three parts. The first is the sign, which is the sign of the corresponding linear model. The motivation for this is to indicate the direction of alignment of a variable with the target y. The second part is coefficient size. It is defined as the fraction of absolute variable attribution allotted to x_k across Ω. The sum of the absolute value of Shapley share coefficients is one by construction.[14] It measures how much of the model output is explained by x_k. The last component is the significance level, indicated by the star notation ($*$), and refers to the standard notation used in regression analysis to indicate the certainty with which we can reject the null hypothesis (Eq. 5). This indicates the confidence one can have in information derived from variable x_k measured by the strength of alignment of the corresponding Shapley components and the target, which is the same as its interpretation in a conventional regression analysis.

Equation 7 provides the explicit form for the linear model, where an analytical form exists. The only difference to the conventional regression case is the normalizing factor.

4.2 Results

We explain the predictions of the machine learning models and the linear regression as calibrated in the baseline setup of our forecasting. Our focus is largely on explaining forecast predictions in a pseudo-real-world setting where the model is trained on earlier observations that predate the predictions. However, in some cases it can be instructive to explain the predictions of a model that was trained on observations across the whole time period. For that, we use fivefold block cross-validation [3, 42].[15] This cross-validation analysis is subject to look-ahead bias, as we use future data to predict the past, but it allows us to evaluate a model for the whole time series.

4.2.1 Feature Importance

Figure 2 shows the global variable importance based on the analysis of the forecasting predictions. It compares Shapley shares $|\Gamma^S|$ (left panel) with permutation importance $\bar{I}$ (middle panel). The variables are sorted by the Shapley shares of the best-performing model, the random forest. Vertical lines connect the lowest and highest share across models for each feature as a measure for disagreement between models.

The two importance measures only roughly agree in their ranking of feature importance. For instance, using a random forest model, past unemployment seems to be a key indicator according to permutation importance but relatively less crucial

[14] The normalization is not needed in binary classification problems where the model output is a probability. Here, the a Shapley contribution relative to a base rate can be interpreted as the expected change in probability due to that variable.

[15] The time series is partitioned in five blocks of consecutive points in time and each block is once used as the test set.

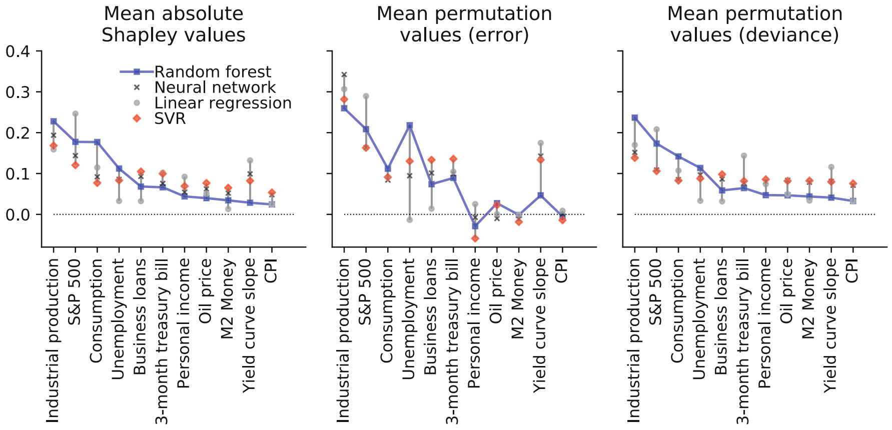

Fig. 2 Variable importance according to different measures. The left panel shows the importance according to the Shapley shares and the middle panel shows the variable importance according to permutation importance. The right panel shows an altered metric of permutation importance that measures the effect of permutation on the predicted value

according to Shapley calculations. Permutation importance is based on model forecasting error and so is a measure of a feature's predictive power (how much does its inclusion in a model improve predictive accuracy) and it is influenced by how the relationship between outcome and features may change over time. In contrast, Shapley values indicate which variables influence a predicted value, independent of predictive accuracy. The right panel of Fig. 2 shows an altered measure of permutation importance. Instead of measuring the change in the error due to permutations, we measure the change in the predicted value.[16] We see that this importance measure is more closely aligned with Shapley values. Furthermore, when we evaluate permutation importance using predictions based on block cross-validation, we find a strong alignment with Shapley values as the relationship between variables is not affected by the change between the training and test set (not shown).

Figure 3 plots Shapley values attributed to the S&P500 (vertical axis) against its input values (horizontal axis) for the random forest (left panel) and the linear regression (right panel) based on the block cross-validation analysis.[17] Each point reflects one of the observations between 1990 and 2019 and their respective value

[16] This metric computes the mean absolute difference between the observed predicted values and the predicted values after permuting feature k : $\frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - \hat{y}_{i(k)}^{perm}|$. The higher this difference, the higher the importance of the feature k (see [26, 36] for similar approaches to measure variable importance).

[17] Showing the Shapley values based on the forecasting predictions makes it difficult to disentangle whether nonlinear patterns are due to a nonlinear functional form or to (slow) changes of the functional form over time.

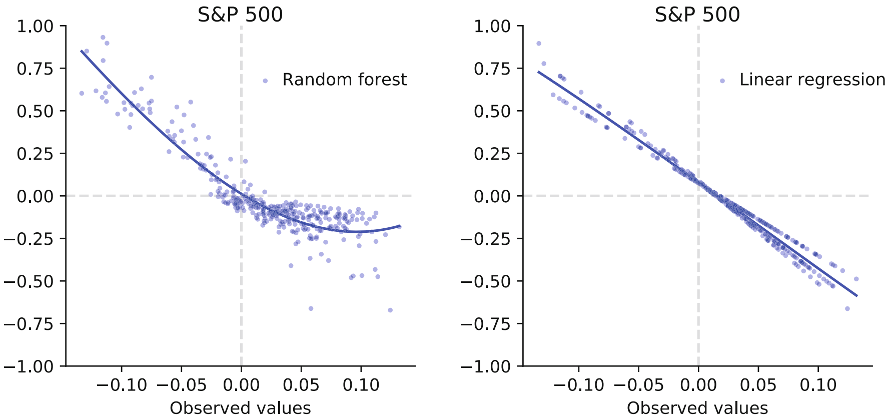

Fig. 3 Functional form learned by the random forest (left panel) and linear regression. The gray line shows a 3-degree polynomial fitted to the data. The Shapley values shown here are computed based on fivefold block cross-validation and are therefore subject to look-ahead bias

on the S&P500 variable. The approximate functional forms learned by both models are traced out by best-fit degree-3 polynomials. The linear regression learns a steep negative slope, i.e., higher stock market values are associated with lower unemployment 1 year down the road. This makes economic sense. However, we can make more nuanced observations for the random forest. There is satiation for high market valuations, i.e., changes beyond a certain point do not provide greater information for changes in unemployment.[18] A linear model is not able to reflect those nuances, while machine learning models provide a more detailed signal from the stock market and other variables.

4.2.2 Shapley Regressions

Shapley value-based inference allows to communicate machine learning models analogously to a linear regression analysis. The difference between the coefficients of a linear model and Shapley share coefficients is primarily the normalization of the latter. The reason for this is that nonlinear models do not have a “natural scale,” for instance, to measure variation. We summarize the Shapley regression on the forecasting predictions (1990–2019) of the random forest and linear regression in Table 4.

The coefficients β^S measure the alignment of a variable with the target. Values close to one indicate perfect alignment and convergence of the learning process. Values larger than one indicate that a model underestimates the effect of a variable on the outcome. And the opposite is the case for values smaller than one. This

[18] Similar nonlinearities are learned by the SVR and the neural network.

Table 4 Shapley regression of random forest (left) and linear regression (right) for forecasting predictions between 1990–2019. Significance levels: $^{*}p$ <0.1; $^{**}p$ <0.05; $^{***}p$ <0.01

	Random forest			Linear regression		
	β^S	p-value	Γ^S	β^S	p-value	Γ^S
Industrial production	0.626	0.000	−0.228***	0.782	0.000	−0.163***
S&P 500	0.671	0.000	−0.177***	0.622	0.000	−0.251***
Consumption	1.314	0.000	−0.177***	2.004	0.000	−0.115***
Unemployment	1.394	0.000	+0.112***	2.600	0.010	+0.033***
Business loans	2.195	0.000	−0.068***	2.371	0.024	−0.031**
3-month treasury bill	1.451	0.008	−0.066***	−1.579	1.000	−0.102
Personal income	−0.320	0.749	+0.044	−0.244	0.730	+0.089
Oil price	1.589	0.018	−0.040**	−0.246	0.624	−0.052
M2 Money	0.168	0.363	−0.034	−4.961	0.951	−0.011
Yield curve slope	1.952	0.055	+0.029*	0.255	0.171	+0.132
CPI	0.245	0.419	−0.024	−0.790	0.673	−0.022

can intuitively be understood from the model hyperplane of the Shapley regression either tilting more towards a Shapley component from a variable (underestimation, $\beta_k^S > 1$) or away from it (overestimation, $\beta_k^S < 1$). Significance decreases as the β_k^S approaches zero.[19]

Variables with lower p-values usually have higher Shapley shares $|\Gamma^S|$, which are equivalent to those shown in Fig. 2. This is intuitive as the model learns to rely more on features which are important for predicting the target. However this does not hold by construction. Especially in the forecasting setting where the relationships of variables change over time, the statistical significance may disappear in the test set, even for features with high shares.

In the Shapley regression, more variables are statistically significant for the random forest than for the linear regression model. This is expected, because the forest, like other machine learning models, can exploit nonlinear relationships that the regression cannot account for (as in Fig. 3), i.e., it is a more flexible model. These are then reflected in localized Shapley values providing a stronger, i.e., more significant, signal in the regression stage.

5 Conclusion

This chapter provided a comparative study of how machine learning models can be used for macroeconomic forecasting relative to standard econometric approaches. We find significantly better performance of machine learning models for forecasting

[19] The underlying technical details for this interpretation are provided in [24].

changes in US unemployment at a 1-year horizon, particularly in the period after the global financial crisis of 2008.

Apart from model performance, we provide an extensive explanation of model predictions, where we present two approaches that allow for greater machine learning interpretability—permutation feature importance and Shapley values. Both methods demonstrate that a range of machine learning models learn comparable signals from the data. By decomposing individual predictions into Shapley value attributions, we extract learned functional forms that allow us to visually demonstrate how the superior performance of machine learning models is explained by their enhanced ability to adapt to individual variable-specific nonlinearities. Our example allows for a more nuanced economic interpretation of learned dependencies compared to the interpretation offered by a linear model. The Shapley regression framework, which enables conventional parametric inference on machine learning models, allows us to communicate the results of machine learning models analogously to traditional presentations of regression results.

Nevertheless, as with conventional linear models, the interpretation of our results is not fixed. We observe some variation under different models, different model specifications, and the interpretability method chosen. This is in part due to small sample limitations; this modelling issue is common, but likely more aggravated when using machine learning models due to their nonparametric structure.

However, we believe that the methodology and results presented justify the use of machine learning models and such explainability methods to inform decisions in a policy-making context. The inherent advantages of their nonlinearity over conventional models are most evident in a situation where the underlying data-generating process is unknown and expected to change over time, such as in a forecasting environment as presented in the case study here. Overall, the use of machine learning in conjunction with Shapley value-based inference as presented in this chapter may offer a better trade-off between maximizing predictive performance and statistical inference thereby narrowing the gap between Breiman's two cultures.

References

1. Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716–723.
2. Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences, 113*(27), 7353–7360.
3. Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences, 191*, 192–213.
4. Bianchi, D., Büchner, M., & Tamoni, A. (2019). Bond risk premia with machine learning. In *USC-INET Research Paper*, No. 19–11.
5. Bluwstein, K., Buckmann, M., Joseph, A., Kang, M., Kapadia, S., & Simsek, Ö. (2020). Credit growth, the yield curve and financial crisis prediction: evidence from a machine learning approach. In *Bank of England Staff Working Paper, No. 848*.
6. Bracke, P., Datta, A., Jung, C., & Sen, S. (2019). Machine learning explainability in finance: an application to default risk analysis. In *Bank of England Staff Working Paper, No. 816*.
7. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

8. Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science, 16*(3), 199–231.
9. Chen, J. C., Dunn, A., Hood, K. K., Driessen, A., & Batch, A. (2019). Off to the races: A comparison of machine learning and alternative data for predicting economic indicators. In *Big Data for 21st Century Economic Statistics*. Chicago: National Bureau of Economic Research, University of Chicago Press. Available at: http://www.nber.org/chapters/c14268.pdf
10. Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., et al. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal, 21*(1), C1–C68.
11. Chernozhukov, V., Demirer, M., Duflo, E., & Fernandez-Val, I. (2018). Generic machine learning inference on heterogenous treatment effects in randomized experiments. In *NBER Working Paper Series, No. 24678*.
12. Conneau, A., & Lample, G. (2019). Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems, NIPS 2019* (Vol. 32, pp. 7059–7069). Available at: https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf
13. Coulombe, P. G., Leroux, M., Stevanovic, D., & Surprenant, S. (2019). How is machine learning useful for macroeconomic forecasting. In *CIRANO Working Papers 2019s-22*. Available at: https://ideas.repec.org/p/cir/cirwor/2019s-22.html
14. Crawford, K. (2013). The hidden biases of big data. *Harvard Business Review*, art number H00ADR-PDF-ENG. Available at: https://hbr.org/2013/04/the-hidden-biases-in-big-data
15. Döpke, J., Fritsche, U., & Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International Journal of Forecasting, 33*(4), 745–759.
16. Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems, NIPS 2016* (Vol. 9, pp. 155–161). Available at: https://papers.nips.cc/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf
17. European Union. (2016). Regulation (EU) 2016/679 of the European Parliament, Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union, L119*, 1–88.
18. Fisher, A., Rudin, C., & Dominici, F. (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.
19. Friedman, J., Hastie, T., & Tibshirani, R. (2009). *The Elements of Statistical Learning. Springer Series in Statistics*. Berlin: Springer.
20. Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., & Walther, A. (2017). Predictably unequal? the effects of machine learning on credit markets. In *CEPR Discussion Papers* (No. 12448).
21. Giannone, D., Lenza, M., & Primiceri, G. E. (2017). Economic predictions with big data: The illusion of sparsity. In *CEPR Discussion Paper* (No. 12256).
22. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge: MIT Press.
23. Henelius, A., Puolamäki, K., Boström, H., Asker, L., & Papapetrou, P. (2014). A peek into the black box: exploring classifiers by randomization. *Data Mining and Knowledge Discovery, 28*(5–6), 1503–1529.
24. Joseph, A. (2020). *Parametric inference with universal function approximators*, arXiv, CoRR abs/1903.04209
25. Kazemitabar, J., Amini, A., Bloniarz, A., & Talwalkar, A. S. (2017). Variable importance using decision trees. In *Advances in Neural Information Processing Systems, NIPS 2017* (Vol. 30, pp. 426–435). Available at: https://papers.nips.cc/paper/2017/file/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Paper.pdf
26. Lemaire, V., Féraud, R., & Voisine, N. (2008). Contact personalization using a score understanding method. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (pp. 649–654).
27. Lipton, Z. C. (2016). *The mythos of model interpretability*, ArXiv, CoRR abs/1606.03490

28. Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems, NIPS 2017* (Vol. 30, pp. 4765–4774). Available: https://papers.nips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
29. Lundberg, S., Erion, G., & Lee, S.-I. (2018). *Consistent individualized feature attribution for tree ensembles*. ArXiv, CoRR abs/1802.03888
30. McCracken, M. W., & Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics, 34*(4), 574–589.
31. Medeiros, M. C., Vasconcelos, G. F. R., Veiga, Á., & Zilberman, E. (2019). Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics, 39*(1), 98–119.
32. Miller, T. (2017). *Explanation in Artificial Intelligence: Insights from the Social Sciences*. ArXiv, CoRR abs/1706.07269
33. Racine, J. (2000). Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics, 99*(1), 39–61.
34. Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD* (pp. 1135–11134).
35. Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018* (pp. 1527–1535), art number 16982. Available at: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982
36. Robnik-Šikonja, M., & Kononenko, I. (2008). Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering, 20*(5), 589–600.
37. Schroff, F., Kalenichenko, D., & Philbin. J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 815–823).
38. Sermpinis, G., Stasinakis, C., Theofilatos, K., & Karathanasopoulos, A. (2014). Inflation and unemployment forecasting with genetic support vector regression. *Journal of Forecasting, 33*(6), 471–487.
39. Shapley, L. (1953). A value for n-person games. *Contributions to the Theory of Games, 2*, 307–317.
40. Shrikumar, A., Greenside, P., & Anshul, K. (2017). *Learning important features through propagating activation differences*. ArXiv, CoRR abs/1704.02685.
41. Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science, 362*(6419), 1140–1144.
42. Snijders, T. A. B. (1988). On cross-validation for predictor evaluation in time series. In T. K. Dijkstra (Ed.), *On model uncertainty and its statistical implications, LNE* (Vol. 307, pp. 56–69). Berlin: Springer.
43. Stock, J. H., & Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association, 97*(460), 1167–1179.
44. Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research, 11*, 1–18.
45. Štrumbelj, E., Kononenko, I., Robnik-Šikonja, M. (2009). Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering, 68*(10), 886–904.
46. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58*(1), 267–288.
47. Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association, 113*(523), 1228–1242.