An interpretable machine learning workflow with an application to economic forecasting

Marcus Buckmann[1], Andreas Joseph*[1], and Helena Robertson[2]

[1]Bank of England
[2]Financial Conduct Authority, UK

June 9, 2021

Abstract

We propose a generic workflow for the use of machine learning models to inform decision making and to communicate modelling results with stakeholders. It involves three steps: A comparative model evaluation, a decomposition of predicted values into feature contributions, and statistical inference on feature attributions. We use this workflow to forecast US unemployment one year ahead in a monthly dataset and find that universal function approximators, including random forests and neural networks, outperform conventional models. This better performance is associated with their greater flexibility in accounting for time-varying and nonlinear relationships in the data generating process. We use Shapley values to explain the predictions of the machine learning models and to identify the economically meaningful nonlinearities learned by the models, which allow us to make nuanced interpretations of model workings. Shapley regressions for statistical inference on machine learning models enable us to assess and communicate variable importance akin to conventional econometric approaches.

1 Introduction

A machine learning model may make an investment decision, assist a human driver in heavy traffic, suggest a movie to watch or inform the decisions of economic policy makers. While it is desirable to understand how the models works in the first three instances, it is essential in the last. Policy makers, such as in central banks, are accountable for their decisions and need to be able to clearly communicate their rationale to stakeholders.

Such transparency is necessary when policy makers make decisions informed by the output of models (George, 1999; Burgess et al., 2013; Independent Evaluation Office,

*Corresponding author. *Disclaimer:* The content and views presented in this article do not represent the views of the Bank of England (BoE) or the Financial Conduct Authoricty in the UK (FCA). The research was supported by the BoE and FCA and conducted at the BoE. *Acknowledgement:* Many thanks to David Bholat for support of the project and helpful comments on the manuscript.

2015). Economic policy makers see the potential merits in using machine learning in their decision making process (Haldane, 2018) but there exists a key challenge. Whilst these models often offer improved predictive accuracy, many nonlinear models such as neural networks and random forests are opaque and are, as such, hard to communicate clearly.[1]

We address these issues of transparency and communication by presenting a multi-step workflow for the use of machine learning models, going from training a model to interpreting the results and communicating them in a standardised way. Throughout the paper, we apply the procedure to a macroeconomic case study, where we forecast changes in unemployment on a one-year horizon—an important input for fiscal and monetary policy decisions (Burgess et al., 2013). The presented workflow consists of three steps which can be applied to other contexts in a straightforward manner. First, a horse race is conducted between conventional statistical methods and machine learning models to provide prima facie evidence of whether a machine learning approach is likely to deliver benefits in terms of predictive accuracy. Second, the machine learning predictions are decomposed into the contributions of the individual model variables. This allows us to uncover the relative importance of features and understand the functional forms learned by the machine learning models. By a comparison across models, one can gauge how robust feature decompositions are to the choice of the algorithm. Third, statistical inference is conducted to understand which features make a statistically significant contribution to the accuracy of a model, providing a level of confidence for our interpretations. This inference uses a parametric regression analysis, allowing for a standardised communication of statistical model results.

The present paper connects different fields, ranging from machine learning and model interpretability to statistical inference and economic forecasting. There is a growing literature that suggests that machine learning methods can outperform more conventional models in economic prediction problems including forecasting. For example, machine learning methods have been shown to be better at predicting bond risk premia (Bianchi et al., 2019), forecasting macroeconomic variables such as unemployment and inflation (Sermpinis et al., 2014; Chen et al., 2019), recessions (Döpke et al., 2017), and financial crises (Bluwstein et al., 2020).[2] However, other papers do not observe consistently improved performance by using machine learning, instead finding that it is state or horizon dependent (Kock and Teräsvirta, 2014). This mixed evidence validates our horse race as an important first step for the workflow. We find that machine learning models outperform econometric benchmarks in predicting 1-year changes in US unemployment.

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 (Ng and Wright, 2013; Elliott and Timmermann, 2008). In line with the literature, we suggest that it is the inherent nonlinearity of nonparametric models that allows them to learn and exploit complex relationships for

[1] Additional barriers to the wider use of machine learning models for decision making processes are the intertwined ethical, safety, privacy, and legal concerns about the application of opaque models (Crawford, 2013; European Union, 2016; Fuster et al., 2017), but these are not covered here.

[2] In these problems, several variables are used to forecast the outcome variable. In the univariate case, when only the lagged outcome is used for prediction, evidence suggests that statistical methods or hybrid models combining statistical and machine learning approaches outperform pure machine learning methods, on average (Makridakis et al., 2018a,b; Parmezan et al., 2019).

prediction (Wang and Manning, 2013). Coulombe et al. (2019) show that this advantage of machine learning models to exploit nonlinearities in macroforecasting is enhanced at longer horizons. However, the nonlinear relationships learned are not directly observable, which has led to the black box critique of these models as a major challenge to their applicability to inform decisions. Therefore, model interpretability methods are necessary for an explanation of how machine learning models make predictions.

Approaches to interpretable machine learning come from different directions: epistemic discussions about what it means for a model to be interpretable (Rudin, 2019), technical approaches in machine learning research (Doshi-Velez and Kim, 2017), and methodology in econometrics and statistics (Chernozhukov et al., 2018). This paper primarily focuses on the latter two.

Miller (2019) analyses the psychology of explanations and suggests that humans expect explanations that are based on a limited number of causes rather than an exhaustive account of all factors—acknowledging that the simplification of the problem risks introducing bias. Relatedly, Lipton (2016) argues that a high-dimensional linear model is not necessarily more interpretable than a compact neural network that learns from only few features. Also, if the linear model is trained on abstract features, for instance, obtained by principal component analysis or an autoencoder, its parameters may not provide an obvious economic interpretation.[3]

In the machine learning literature, approaches to interpretability usually focus on measuring how important input variables are for prediction (Lundberg et al., 2020). *Variable attributions* can be either global, by assessing the variable importance across the whole data set or local, by measuring the importance of the variables at the level of individual observations. Global methods usually track the importance of variables by assessing a variable's impact on the accuracy of the model (Kazemitabar et al., 2017) while local methods decompose individual predictions into variable contributions (Štrumbelj and Kononenko, 2010; Ribeiro et al., 2016; Shrikumar et al., 2017; Lundberg and Lee, 2017). Local attribution can always be summarised in a global variable attribution measure by averaging local attributions across all observations. Popular global methods are permutation importance or Gini importance for tree-based models (Breiman, 2001). Popular local decomposition methods are LIME[4] (Ribeiro et al., 2016), DeepLIFT (Shrikumar et al., 2017) and Shapley values (Štrumbelj and Kononenko, 2010). Lundberg and Lee (2017) demonstrate that Shapley values offer a unified framework of LIME and DeepLIFT with appealing properties. Most importantly, Shapley values guarantee *consistency*, where a consistent measure of variable importance preserves the relative importance between variables although, this statistical property comes at the cost of computational complexity. We justify our choice of two feature attribution methods for a comparison of machine learning model interpretability: permutation importance of features (Breiman, 2001; Fisher et al., 2019) and Shapley values (Štrumbelj and Kononenko, 2010; Lundberg and Lee, 2017). The former is economical to compute whilst the latter has the advantage that it allows to depict the nonlinear relationships learned by the machine learning models. To the best of our knowledge, this is the first study revealing the functional form learned by machine learning models for macroeconomic forecasting.

[3]Indeed, in the forecasting literature, it is common to use many variables as predictors (Giannone et al., 2017) or latent factors that summarise individual variables (Stock and Watson, 2002).

[4]Local Interpretable Model-agnostic Explanations.

Plumb et al. (2018) use a novel method to demonstrate how comparing global with local interpretations helps identify limitations to each approach, which motivates why our comparison of global and local attributions with our identification of the underlying functional form is a key step in workflow that allows us to make nuanced interpretations of the data generating process.

However, these global and local attribution methods are only descriptive—they explain the drivers of model predictions, but they do not assess the predictors' statistical significance, i.e. how certain one can be that a variable is actually important to describe a specific outcome. We extend our interpretation of machine learning models for forecasting by statistically testing the predictors in a Shapley regression framework (Joseph, 2019). Shapley values and inference based on them is arguably the most general and rigorous approach to address the issues of machine learning interpretability and model communication. In this way, we close the gap between two traditional modelling approaches, the maximisation of predictive performance using 'black box' machine learning methods and the application of statistical techniques to make inferences about the data generating process (Breiman et al., 2001).

The remainder of this paper is structured as follows. The data and the forecasting methodology used throughout this paper is introduced in Section 2. Forecasting results are discussed in Section 3. Contrastive model interpretability methods and results for our forecasting exercise are discussed in Section 4. We conclude in Section 5. The technical appendix discusses the computation of Shapley values in a modelling context.

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 outcome, respectively, where m is the number of observations in the time series. 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.

2.1 Data

We use the *FRED-MD* macroeconomic database (McCracken and Ng, 2016). The data contains monthly series of 127 macroeconomic indicators of the US between 1959 and 2019.[5] Our outcome variable is unemployment and we choose nine variables as predictors, each capturing a different macroeconomic channel. We additionally add a variable for the slope of the yield curve (difference in interest rates for the 10-year treasury rate and the 3-month treasury bill). We use the stationarity transformations suggested by the authors of the dataset that include first differences, log differences and second order log differences. Given that we predict the yearly change of unemployment, we set lag length l to 12 for the outcome and lagged outcome (predictor) variables. For the remaining predictors, we set $l = 3$ in our baseline set-up. This generally leads to the best performance (see Table III for other choices of l). Table I shows the variables, with the respective transformations

[5] We do not include data from 2020 in our sample because these are likely impacted by the Covid-19 pandemic which would only be partially covered.

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

Table I: Series used in the forecasting experiment. The middle column shows the transformations suggested in by the authors of the FRED-MD database, the right column shows how the series are named in that database.

and the series names in the original database. The augmented Dickey-Fuller test confirms that all transformed series are stationary ($p < 0.01$).

2.2 Models

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

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 lagged values of the response variable as predictors: $\hat{y}_i = \alpha + \sum_{l=1}^{h} \theta_i y_{i-l}$. We test AR models of different lag lengths $1 \leq h \leq 12$, and chose h using the Akaike information criterion.

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 is any given predictive model. For example, if f is a linear model, $f(y_i, x_i) = \alpha + \theta_0 y_{i-12} + \sum_{k=1}^{n} \theta_k x_{i-12,k}$. To simplify notation in what follows, we include the lagged outcome in the feature matrix x. We test five full information models: ordinary least squares regression, regularised regression with ridge and lasso penality, and three machine learning models (random forests (Breiman, 2001), support vector regression (Drucker et al., 1997), and artificial neural networks (Goodfellow et al., 2016)).

2.3 Experimental procedure

We evaluate how all models predict changes in unemployment one year ahead. After transforming the variables (see Table I), 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. As the models predict 12-month changes, 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 $1 \dots i$, the earliest observation in the test set that provides a pseudo real-time 12-month forecast is $i + 12$. For observations $i + 1, \dots, i + 11$, the time difference from the last observation in the training set i is less than one year.
All machine learning models that we test have hyperparameters. We optimise their values using 5-fold cross validation in the training dataset.[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 sampling. We train 100 models on different bootstrapped samples of the training set and average their predictions (bagging). We do not use bagging for the random forest as each individual tree is already calibrated on a different bootstrap sample.

3 Forecasting performance

We compare the performance of the different prediction models across all observations and at different time periods. This is the first step in the proposed workflow; the results will indicate if the use of machine learning models is worthwhile for the application at hand.

3.1 Baseline setting

The results of the horse race are shown in Table II. We consider three measures to assess the 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 minimise 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.
Table II also breaks down the performance in three periods: the 1990s and the periods before and after the global financial crisis (GFC, Sep. 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]

[6] For the hyperparameter search, we also consider partitionings of the training set that take the temporal dependency of our data into account (Bergmeir and Benítez, 2012). We use block cross-validation (Snijders, 1988) and hv-block cross-validation (Racine, 2000). However, both methods do not improve the forecasting accuracy.

[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.

	Correlation	MAE	RMSE (normalised by first row)			
Time period	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 lag model	0.242	1.163***	1.226***	1.027	1.057	1.352***

Table II: 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 one. 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.

Apart from the support vector regression (SVR), all machine learning models outperform the linear models on the whole sample. The inferior performance of the SVR is not surprising as it does not minimise 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 1990 and the periods before the global financial crisis, there are only small difference in performance between the models, with the neural network being the most accurate model. Only after the onset of the crisis, the random forest outperforms the other models by a large and statistically significant margin of up to 35%.

Figure I 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 II. 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 forecast volatility can be useful in registering negative turning points. A similar observation can be made after the burst of the dot-com bubble in 2000.

3.2 Robustness checks

We altered several parameters in our baseline set-up to investigate their effect on the predictive performance. The results are shown in Table III. The RMSE of alternative specifications is again divided by the RMSE of the random forest in the baseline set-up

[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.

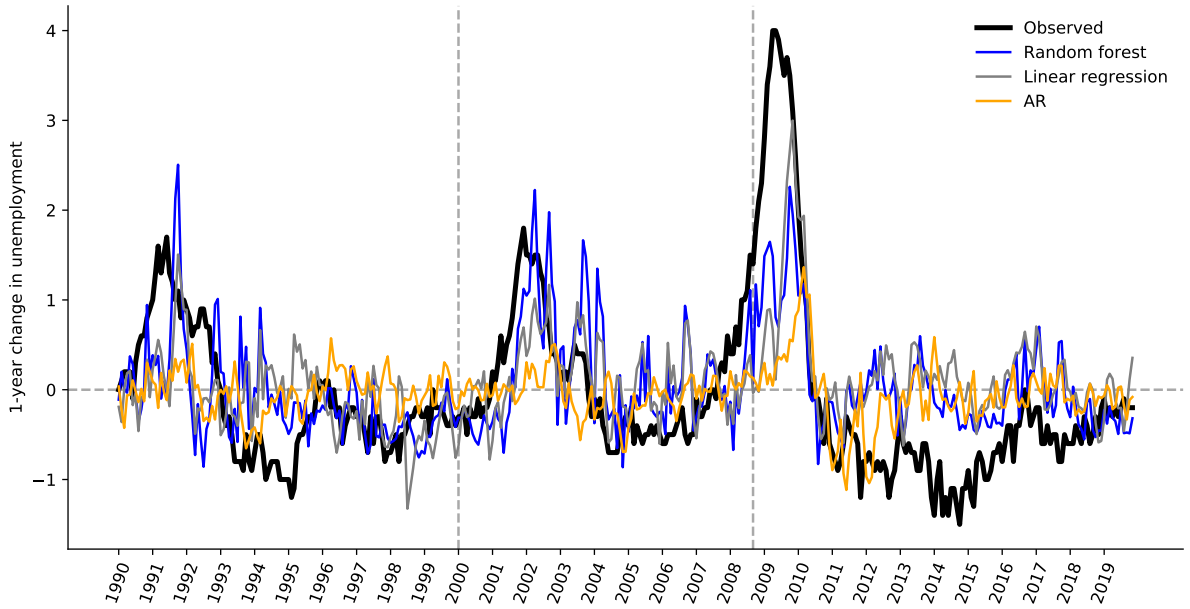

Figure I: Observed and predicted 1-year change in unemployment for the whole forecasting period.

for a clearer comparison.

Window size. In the baseline set-up, the training set grows over time (expanding window). This can potentially improve the performance 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 prevents quick adaption to structural changes. To differentiate between these two cases, 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 "memorise" different sets of information through the incorporation of multiple specification in the same model. For instance, different paths down a tree model, or different trees in a forest, are all different submodels, e.g. characterisations of different time periods. By contrast, the 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 set-up, we use a horizon of three months, when calculating first differences, log diffferences and second order log differences of the predictors (see Table I). Testing lag lengths of 1, 6, 9, and 12 months, we find that three 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 commonly used for short-term economic projections.

Bootstrapped models. In the baseline set-up, we bootstrapped all full information models except the random forest, which builds 500 bootstrapped decision trees by design.

	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	-	-
Bootstrapped models						
no	1.000	1.179	1.089	1.117	1.160	1.226
100 models	-	1.049	1.082	1.105	-	-

Table III: Performance for different parameter specifications. The shown metric is RMSE divided by the RMSE of the random forest in the baseline set-up.

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. Bootstrap aggregation (bagging) reduces the variance and thus the degree of overfitting. Not surprisingly, the improvement due to bootstrapping was limited for the linear model, as it is more stable than the machine learning methods, i.e different random samples are likely to lead to almost identical models.

Overall, machine learning models are mostly more accurate than conventally used benchmarks. This justifies additional investments in model interpretability discussed in the next section.

4 Model interpretability

In this section, under the second and third steps of the workflow, we compare methods to obtain feature attributions, contrasting their different interpretations of model outputs, at both a local and global level, before undertaking statistical inference on model decompositions.

4.1 Methodology

First, we introduce and compare two different methods for interpreting machine learning models: *permutation importance* (Breiman, 2001; Fisher et al., 2019) and *Shapley values*.

Both approaches are *model-agnostic*, meaning that they can be applied to any model, unlike other approaches, such as Gini impurity (Kazemitabar et al., 2017; Friedman et al., 2009) that 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 across all predictions whilst Shapley values are constructed locally, i.e. for each prediction individually. We note that both importance measures generally require column-wise independence of the features, that is, contemporaneous independence in our forecasting experiments, an assumption that will not hold under all contexts. Only for tree models there exists an efficient algorithm that considers the dependencies of features (Lundberg et al., 2018).

Permutation importance

The permutation importance of a variable measures the change of model performance when the values of that variables are randomly scrambled. 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. $e_k^{perm} >> e$. The value of the permutation error e_k^{perm} depends on the realisation 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.[9]

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.
   (b) Re-evaluate the test score for x_k^{perm}, resulting in e_k^{perm}.
   (c) The permutation importance of x_k is given by $I(x_k) = e_k^{perm}/e$. Alternatively, the difference $e_k^{perm} - e$ can be considered.
   (d) Repeat and average over Q iterations and average $I_k = 1/Q \sum_q I(x_k)$.
2. If I_k is based on the ratio of errors e_k^{perm}/e, consider the normalised quantity $\bar{I}_k = (I_k - 1)/\sum_k (I_k - 1) \in (0, 1)$.[10]

Permutation importance is an intuitive measure that is relatively cheap to compute—it only requires generating predictions on the permuted data but no model retraining. However, this ease of use comes at some cost. 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. Only permuting both or excluding one would do so. This motivates our comparison with Shapley values because they identify the individual

[9] At a test set of size m, where each observation havs a unique value, there are $m!$ permutations to consider for an exhaustive evaluation. This is intractable to compute for larger m.

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

marginal effect of a feature, accounting for its interaction with all other features. Additionally, the computation of permutation importance requires access to true outcome target values and in many situations, e.g. when working with models trained on sensitive or confidential data, these may not be available.

Shapley values and regressions

Shapley values originate from game theory as a general solution to the problem of attributing a pay-off obtained in a cooperative game to the individual players based on their contribution to the game (Shapley, 1953). Štrumbelj and Kononenko (2010) introduced the analogy between players in a cooperative game and variables in a general supervised model. In the latter, variables jointly generate a prediction, the pay-off.

The Shapley values of a model offer a local decomposition of each model prediction, i.e. they add up to the predicted values of a model. For an input value x_i, we have

$$f(x_i) = \sum_{k=0}^{n} \phi_k^S(x_i)\,, \tag{1}$$

where $\phi_k^S(x_i)$ is the Shapley value associated with predictor k and ϕ_0^S an intercept, usually the model mean prediction. Shapley values come with a host of appealing analytical properties which are inherited from their game theoretic origins. Moreover, the decomposition in Equation 1 does not need to refer to single variables but can also include interactions or even higher-order terms of interest. This flexibility comes at the cost of computing Shapley values, which is discussed in a technical appendix.

We next formulate an inference framework—*Shapley regression*—to analyse the statistical significance of Shapley value components (Joseph, 2019),

$$y_i = \phi_0^S + \sum_{k=1}^{n} \phi_k^S(f, x_i)\beta_k^S + \epsilon_i \equiv \Phi_i^S \beta^S + \epsilon, \tag{2}$$

where $\Phi^S[f(x_i)]$ is the Shapley decomposition of model f and $\hat{\epsilon}_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The rightmost term uses inner product notation with $\beta_0^S \equiv 1$. The surrogate coefficients β_k^S, $k > 0$ are tested against the null hypothesis

$$\mathcal{H}_0^k(\Omega) \;:\; \{\beta_k^S \leq 0 \,\big|\, \Omega\}\,, \tag{3}$$

with $\Omega \in \mathbb{R}^n$ as (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 can be seen when assuming that the vector $\beta^S \approx 1$,[11] in which case the Shapley components of a model vary one-for-one with the target up to the random component ϵ. However, when the null hypothesis cannot be rejected, there is no significant co-movement between variable Shapley components and the target, in line with non-significance of a variable in a conventional regression analysis.

A key difference to the linear case is the regional dependence on Ω of inference results. We can only make *local* statements about the contributions of a variable, i.e.

[11] This is also the asymptotic value of β^S assuming that each predictor contains some information for predicting y. See Joseph (2019) for details.

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, such that variables are only related to the target functions for subsets of the input space.

The Shapley value decomposition (Equation 1) absorbs the sign 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 $\mathcal{H}_0$ can not be rejected.
Finally, the coefficients β^S are only informative about alignment, not the magnitude of importance of a variable, which is captured by the Shapley decomposition (1). Both together can be summarised by *Shapley share coefficients*,

$$\Gamma_k^S(f,\Omega) \quad \equiv \quad \left[sign\big(\beta_k^{lin}\big) \left\langle \frac{|\phi_k^S(f)|}{\sum_{l=1}^n |\phi_l^S(f)|} \right\rangle_\Omega \right]^{(*)} \in [-1,1], \tag{4}$$

$$\overset{f(x)=x\beta^{lin}}{=} \quad \beta_k^{lin(*)} \quad \left\langle \frac{|(x_k - \langle x_k \rangle)|}{\sum_{l=1}^n |\beta_k (x_l - \langle x_l \rangle)|} \right\rangle_\Omega, \tag{5}$$

where $\langle\cdot\rangle_\Omega$ stands for the average across Ω. 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 consist 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 the coefficient size. It is defined as the fraction of absolute variable attribution allotted to x_k across Ω and measures how much of the model output is explained by x_k. The sum of the absolute value of Shapley share coefficients is one by construction.[12] The last component $(*)$ is used to indicate the significance level of Shapley attributions from x_k against the null hypothesis (Equation 3) and, thus, the confidence one can have in information derived from that variable.
Equation 5 provides the explicit form for the linear model, where an analytical form exists. The only difference to the conventional case is the normalising factor.

4.2 Interpretation of results

We explain the predictions of the machine learning models and the linear regression as calibrated in our basline set-up. 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 train the model on a bootstrapped sample of the whole time series and make predictions for those observations not in a bootstrapped training sample. This *out-of-bag* 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.

[12] The normalisation 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.

Feature importance

We first analyse our two methods of model interpretation at a global level. Figure II 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 to highlight the disagreement between models.

We use two different methods to compute Shapley values for the random forest model (Lundberg and Lee, 2017). The first method (blue squares) computes Shapley values under feature dependence and the second method (blue crosses) assumes variable independence (see the technical appendix). The difference in Shapley shares obtained by the two methods is negligible, indicating that the independence assumption which considerably facilitates the computation of Shapley values, should not bias our measures of feature importance.

The two importance measures, permutation importance and Shapley values, 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 less important according to Shapley calculations. The permutation importance shown is based on the forecasting error. It is a measure of a feature's influence on the accuracy of the model and thus affected by how the relationship between outcome and features changes over time. In contrast, Shapley values reflect a variable's influence on

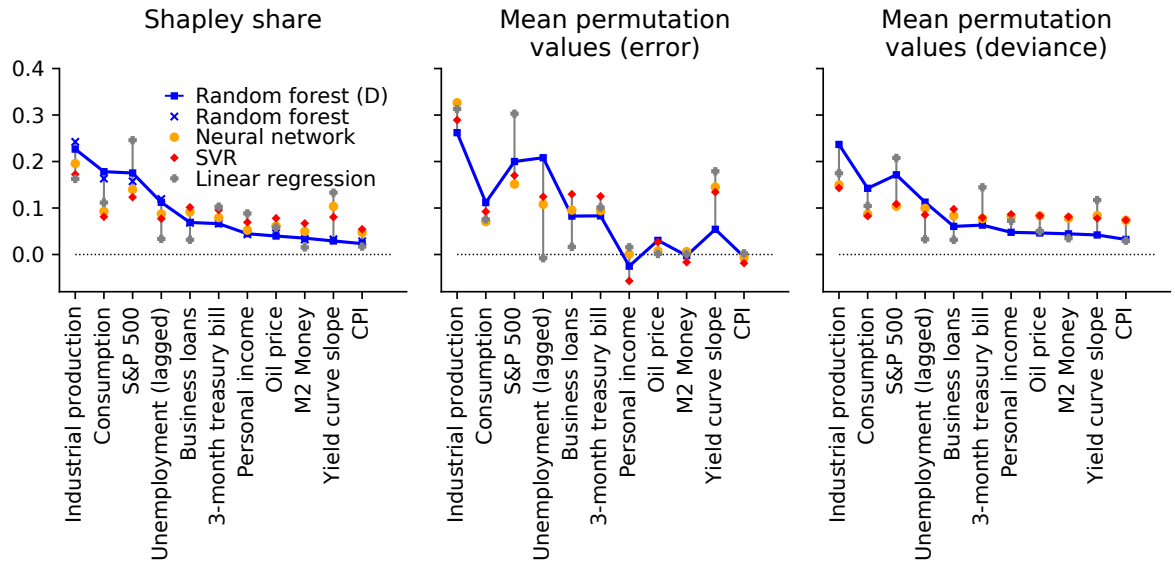

Figure II: Variable importance according to different measures. The left panel shows the importance according to the Shapley shares $|\Gamma^S|$ and the middle panel shows the variable importance according to permutation importance. The right panel shows an altered metric of permutation importance that measures the effect of permutation on the predicted value rather than prediction error. Only the Shapley values of random forest (D) are based on a decomposition that takes the features dependence into account. The other Shapley values shown assume feature independence.

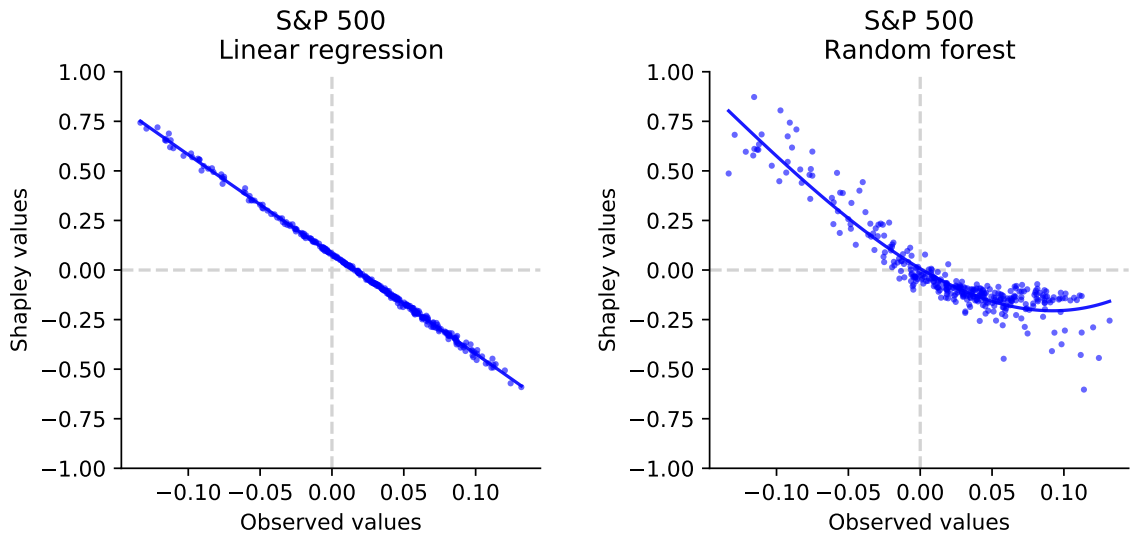

Figure III: Functional form learned by the random forest (left panel) and linear regression. The lines shows a polynomial fitted to the data. The Shapley values are computed on the out-of-bag predictions and are therefore subject to look-ahead bias. Extreme input values (below 2.5% and above 97.5% quantile) are excluded.

the predicted value, independent of that value's accuracy. The right panel of Figure II 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.[13] We see that this importance measure is more closely aligned with Shapley values. Further, when we evaluate permutation importance using predictions based on the out-of-bag analysis, we find a strong alignment with Shapley values (not shown) as the relationship between variables is not affected by the changes between the training and test set.

The different prediction models have a similar importance ranking of the features. There are, however some notable differences such as the difference of the forest and the linear regression in the unemployment and yield curve slope predictors.

The preceding global analysis only shows which variables are important, it does not reveal the functional form learned by the model. Figure III demonstrates how local Shapley decompositions uncover nonlinearities machine learning models have learned from the data. It plots *local* Shapley values (based on the out-of-bag analysis) attributed to the S&P 500 (vertical axis) against its input values (horizontal axis) for the linear regression (left panel) and the random forest (right panel). The approximate functional forms learned by both models are traced out by best-fit first and third-degree polynomials, respectively. We exclude extreme input values (below 2.5% and above 97.5% quantile) to avoid this fit being driven by outliers on the edges. The linear regression learns a steep

[13]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 Lemaire et al. (2008) and Robnik-Šikonja and Kononenko (2008) for similar approaches to measure variable importance).

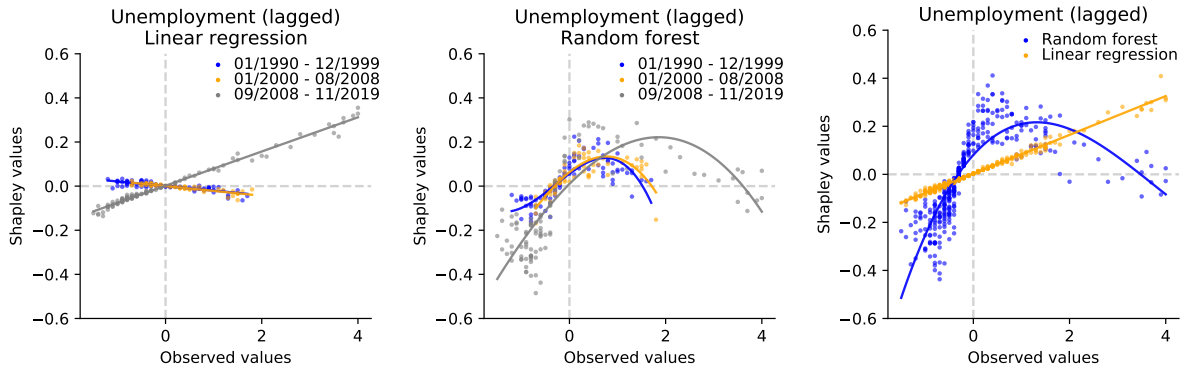

Figure IV: Functional form of lagged unemployment change learned by the random forest (left panel) and linear regression (middle panel) for three models trained up to different points in time. The right panel shows the functional form across all observations when the model was trained at the latest point in time. The lines show polynomials fitted to the data. The Shapley values are computed on the out-of-bag predictions and are therefore subject to look-ahead bias.

negative slope, i.e. higher stock market values are associated with lower unemployment one year ahead. This makes economic sense. However, we can make more nuanced statements when looking at the random forest. The model learns a saturation effect for high market valuations. Changes beyond a certain point do not suggest further drops in unemployment as the line flattens.[14]

Our forecasting models gradually change over time as the training set grows. To investigate this model drift, we consider the out-of-bag predictions of the models trained up to three different points in time. Figure IV shows the functional form for the lagged unemployment change variable. The linear regression models (left panel) trained up to the periods 2000 and 2008 find no predictive power for lagged unemployment. It is only after the onset of the global financial crisis that the linear regression learns a positive relationship—an increase of unemployment increase the predicted increase unemployment one year ahead. However, this is simply reflective of the trend—the 1-year unemployment change was high for a prolonged period following the financial crisis: it was persistently larger than 1 percentage point for 21 consecutive months (July 2008–March 2010). In contrast, the functional form of the random forest (middle panel) is rather stable. Across the three time periods it learns a non-monotonic but intuitive relationship where a high increase in the unemployment makes future increases in unemployment less likely compared to a medium increase. The right panel of Figure IV directly compares the functional form learned by latest regression and random forest across all data points between 1990–2019. The other machine learning models learn the same quadratic relationship. This also explains why the Shapley share of lagged unemployment is much lower for the linear regression compared to the other models (Figure II). Its contribution up the crisis was close to zero.

To better understand the non-monotonic function of lagged unemployment change

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

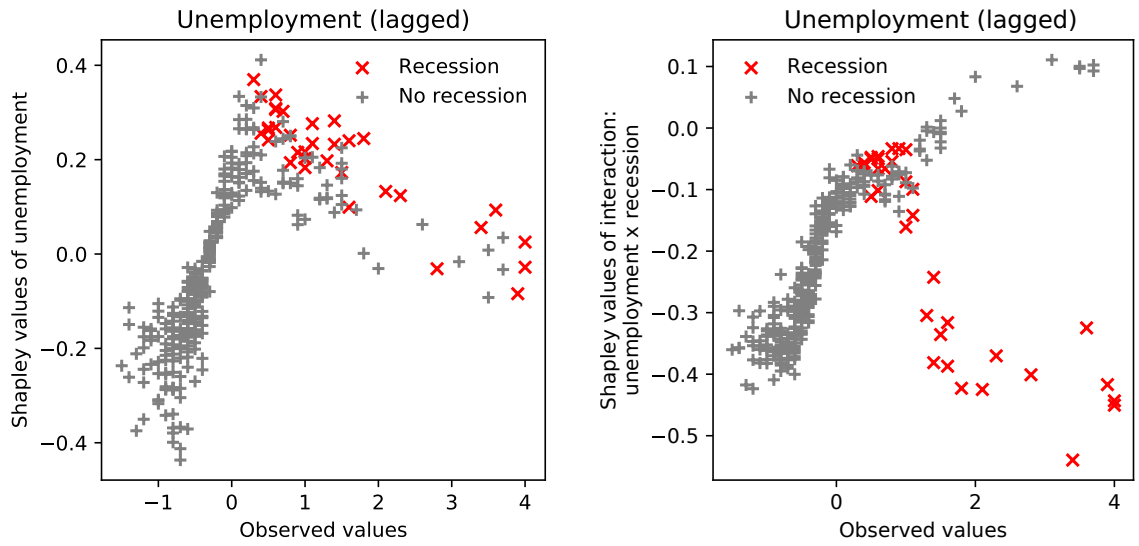

Figure V: Interaction between unemployment changes and recessions as learned by a random forest. The left panel shows the functional form of lagged unemployment changes when the model is trained on the baseline features without a recession indicator (as in Figure IV). The right panel shows the Shapley values of the interaction when the model was trained with a recession indicator. The Shapley values are computed on the out-of-bag predictions.

learned by the random forest, we look into the role of recessions in our model.[15] Figure V (left) again shows the functional form of lagged unemployment as learned by the random forest in the out-of-bag set-up. But now recession observations in the input space are marked in red and extreme values are not excluded. Even though we did not include recessions as an indicator the model could learn from, these account for a large share of the downwards sloping part at high values of positive unemployment change. We further elaborate on this observation by including a recession dummy in our models and compute the Shapley-Taylor index (Agarwal et al., 2019) to decompose the predictions into the main effects of the predictors and interactions.[16]

The interaction (Figure V, right panel) shows two distinct functional forms that cross and confirms that an increase of unemployment during a recession predicts a decrease of future unemployment. When computing the Shapley shares $|\Gamma^S|$ of the main effects of variables (as in Figure II) and their two and three way interactions, the interaction shown here obtains the second highest score. It is higher than the main effect of all variables except the recession indicator. While including the recession indicator improves the interpretation of the results, the predictive accuracy of random forest does not increase. Instead the model learned the role of recession periods implicitly from the other variables implicitly incorporating two different regimes, normal times and recessions.

[15] We use the definition of recessions provided by the Federal Reserve Bank of St. Louis (Federal Reserve Bank of St. Louis, 2020).

[16] Generally, each model prediction can be decomposed in variable main effects (first order terms) and interactions of variables of order two or higher.

	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 (lagged)	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

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

Statistical inference with Shapley regressions

Shapley value-based inference (Equation 2) allows us to communicate machine learning models analogously to a linear regression analysis. We summarise the Shapley regression on the forecasting predictions (1990–2019) of the random forest and linear regression in Table IV.

As mentioned above, 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 can intuitively be understood as 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.

Variables with higher Shapley shares $|\Gamma^S|$ (same as in Figure II) tend to have lower p-values. This is intuitive, demonstrating that the model learns to rely more on features that are important for predicting the target variable. However this does not hold by construction. This is especially so in a forecasting setting where the relationships between variables changes over time, any statistical significance may disappear in the test set, even for features with high Shapley shares.
More variables are statistically significant for the random forest than for the linear regression model. This is expected given the greater flexibility of machine learning models. It also provides further evidence of how non-parametric models, like random forests or other machine learning models, exploit nonlinear relationships that linear regression models cannot account for (as in Figure III in below).

5 Conclusion

This paper presents a workflow for using machine learning to inform decision making in situations where transparency and ease of communicating results are key. The three steps of the workflow are: a horse race between model types, a decomposition of predictions into feature contributions, and statistical inference on model results.

In the first step of our case study, we found a significantly better performance of machine learning models for forecasting yearly changes in US unemployment compared to linear benchmarks. For the second step, we observe pronounced nonlinearities learned by the machine learning models and which also have clear economic interpretations. In the third step of the workflow, we use the Shapley regression framework to show that a larger number of variables are statistically significant predictors for machine learning models than for the linear benchmark. This is line with the former exploiting meaningful nonlinear relationships in the data.

Machine learning methods are increasingly used in economic and social science research. However, most studies using machine learning focus on maximising predictive accuracy and accept the black box nature of the models. Research that does attempt statistical inference on machine learning models often uses controlled and usually less volatile data, for instance from randomised controls trials. Our study shows that the use of machine learning models and statistical inference can be combined to answer real-world problems.

Many decision makers may not be familiar with machine learning methods but we believe that their increased predictive accuracy and ability to detect richer, more nuanced signals in the data justify their use to inform policy decisions. With our workflow, model results can be communicated analogously to familiar and well-understood regression results. Further, we show that our workflow can also be used to identify structural breaks in the data generating process. Future work could directly compare our approach with the respective state-of-the-art techniques in the econometric literature.

A general caveat to using the Shapley regression framework to interpret model results is that potentially complex and nonlinear functional forms cannot be *fully* communicated by a single statistic, such as Shapley share coefficients. However, we believe that the combination of evidence for learned functional forms and statistical inference on feature attributions well justifies the use of our machine learning workflow to inform policy decisions.

Machine learning approaches often provide more accurate predictions than standard linear models. In this case, our workflow helps decision makers to profit both from more accurate predictions and a better understanding of the data generating process—instead of trading off interpretability for accuracy when treating the machine learning model as a black box.

Technical Appendix: Model Shapley values

The Shapley attribution $\phi_k^S(x_i; f)$ for variable k in observation x_i and model f in (1) is given by

$$\phi_k^S(x_i; f) = \sum_{x' \subseteq \mathcal{C}(x) \setminus \{k\}} \frac{|x'|!(n - |x'| - 1)!}{n!} \left[f(x_i | x' \cup \{k\}) - f(x_i | x') \right], \tag{6}$$

$$= \sum_{x' \subseteq \mathcal{C}(x) \setminus \{k\}} \omega_{x'} \left[\mathbb{E}_b[f(x_i) | x' \cup \{k\}] - \mathbb{E}_b[f(x_i) | x'] \right], \tag{7}$$

$$\text{with} \qquad \mathbb{E}_b[f(x_i) | x'] \equiv \int f(x_i) \, \mathrm{d}b(\bar{x'}) = \frac{1}{|b|} \sum_b f(x_i | \bar{x'}) \,. \tag{8}$$

Here, $\mathcal{C}(x) \setminus \{k\}$ is the set of all possible variable combinations (coalitions) when excluding variable k and $|x'|$ 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 coalitions, b is a background dataset and $\bar{x'}$ stands for the set of variables not included in x'.

Equation 6 is the weighted sum of marginal contributions of variable k to all possible variable coalitions.[17] It usually is not possible to just exclude a variable from a model to form the coalition set x'. Instead, the contributions from features not included in x' are integrated out over a suitable background dataset b accroding to Equation 8. Here, $\{x_i | \bar{x'}\}$ is the set of points with variables not in x' being replaced by their corresponding values in b along these dimensions. A reasonable choice for the background data is the training dataset (or a subset or summary thereof) incorporating all information the model has learned from. The background data should provide an informative reference point by determining the intercept ϕ_0^S.

Shapley variable attributions inherit many appealing analytical properties from their game theoretic origins. Particularly, they are the only variable attribution scheme which is local, exact, linear and consistent (see Young (1985); Štrumbelj and Kononenko (2010); Lundberg and Lee (2017) for details). However, the above computation of Shapley values based on conditional expectations also poses some challenges which we briefly discuss here:

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

1. *Computational complexity:* The time to evaluate the above expressions grows exponentially in the number of features, which makes it intractable for already moderate feature sets and dataset sizes. Two possible solutions are either to sample coalitions x' from $\mathcal{C}(x)$, as implemented in the SHAP package by Lundberg and Lee (2017), or to group those features that are not of interest in a single group "*others*" (see Joseph (2019)). The latter has the advantage that computation is still exact.

2. *Feature dependence:* The evaluation of conditional expectations (Equation 8) makes the implicit assumption of feature independence which may be violated in real-world applications. There are again two ways to address this. First, one can estimate Shapley values of tree-based models for which there exists an efficient algorithm that accounts for feature dependence Lundberg et al. (2018). By comparing Shapley values when respecting or ignoring feature dependence, one can gauge the importance of the dependencies. However, caution is warranted when transferring the findings to other model families, e.g. artificial neural networks. Second, one can net out the effect of higher-order feature interactions using the Shapley-Taylor index (Agarwal et al., 2019) to understand dependencies between features.

3. *Expectation consistency:* As shown by Sundararajan and Najmi (2019), attribution consistency which, casually put, avoids contradictions in feature attribution, can be violated when using conditional expectation for the computation of Shapley values, and a single reference value is advocated for. However, this discards much of the potentially rich information a model has learned, such as nonlinearities. A solution to this is provided in Joseph (2019) in the form of an additional condition when comparing different models against a common background. The models' expected values over the background data b need to coincide leading to the same reference $\phi_0^S(b)$. This is fulfilled in many practical situations where models optimise the same objective functions, like the mean squared error.

None of the above challenges is fatal for the application of Shapley values for model interpretability. However, one has to be aware of the possible pitfalls and consequences of approximations and their consequences for model interpretations and any decisions based on them.

References

Agarwal, Ashish, Kedar Dhamdhere, and Mukund Sundararajan (2019) "A new interaction index inspired by the taylor series", *arXiv e-prints*, Vol. 1902.05622.

Bergmeir, Christoph and José M Benítez (2012) "On the use of cross-validation for time series predictor evaluation", *Information Sciences*, Vol. 191, pp. 192–213.

Bianchi, Daniele, Matthias Büchner, and Andrea Tamoni (2019) "Bond risk premia with machine learning", *USC-INET Research Paper*, No. 19-11.

Bluwstein, Kristina, Marcus Buckmann, Andreas Joseph, Miao Kang, Sujit Kapadia, and Özgür Simsek (2020) "Credit growth, the yield curve and financial crisis prediction: evidence from a machine learning approach", *Bank of England Staff Working Paper*, No. 848.

Breiman, Leo (2001) "Random forests", *Machine learning*, Vol. 45, No. 1, pp. 5–32.

Breiman, Leo et al. (2001) "Statistical modeling: The two cultures (with comments and a rejoinder by the author)", *Statistical science*, Vol. 16, No. 3, pp. 199–231.

Burgess, Stephen, Emilio Fernandez-Corugedo, Charlotta Groth, Richard Harrison, Francesca Monti, Konstantinos Theodoridis, and Matt Waldron (2013) "The Bank of England's forecasting platform: COMPASS, MAPS, EASE and the suite of models", Staff Working Paper No. 471, Bank of England.

Chen, Jeffrey C, Abe Dunn, Kyle K Hood, Alexander Driessen, and Andrea Batch (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*: University of Chicago Press.

Chernozhukov, Victor, Mert Demirer, Esther Duflo, and Ivan Fernandez-Val (2018) "Generic machine learning inference on heterogenous treatment effects in randomized experiments", Technical report, National Bureau of Economic Research.

Coulombe, Philippe Goulet, Maxime Leroux, Dalibor Stevanovic, Stéphane Surprenant et al. (2019) "How is machine learning useful for macroeconomic forecasting?", *Working Paper*.

Crawford, Kate (2013) "The hidden biases of big data", Harvard Business Review, Microsoft Research.

Döpke, Jörg, Ulrich Fritsche, and Christian Pierdzioch (2017) "Predicting recessions with boosted regression trees", *International Journal of Forecasting*, Vol. 33, No. 4, pp. 745–759.

Doshi-Velez, Finale and Been Kim (2017) "Towards A Rigorous Science of Interpretable Machine Learning", *ArXiv e-prints*, Vol. 1702.08608.

Drucker, Harris, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik (1997) "Support vector regression machines", in *Advances in Neural Information Processing Systems*, pp. 155–161.

Elliott, Graham and Allan Timmermann (2008) "Economic forecasting", *Journal of Economic Literature*, Vol. 46, No. 1, pp. 3–56.

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*, Vol. L119, pp. 1–88.

Federal Reserve Bank of St. Louis (2020) "NBER based recession indicators for the United States from the period following the peak through the trough [USREC]", , November.

Fisher, Aaron, Cynthia Rudin, and Francesca 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*, Vol. 20, No. 177, pp. 1–81.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2009) *The Elements of Statistical Learning*: Springer Series in Statistics Springer, Berlin.

Fuster, Andreas, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther (2017) "Predictably unequal? the effects of machine learning on credit markets", *CEPR Discussion Papers*, No. 12448.

George, Eddie (1999) "Economic models at the bank of england", Technical report, Bank of England.

Giannone, Domenico, Michele Lenza, and Giorgio E Primiceri (2017) "Economic predictions with big data: The illusion of sparsity", *CEPR Discussion Paper*, No. 12256.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016) *Deep Learning*: MIT Press.

Haldane, Andrew G (2018) "Will big data keep its promise", *Speech at the Bank of England Data Analytics for Finance and Macro Research Centre, King's Business School.*

Independent Evaluation Office (2015) "Evaluating forecast performance report", Technical report, Bank of England. http://www.bankofengland.co.uk/about/Documents/ieo/evaluation1115.pdf, last accessed: 31 July 2017.

Joseph, Andreas (2019) "Parametric inference with universal function approximators", *arXiv preprint arXiv:1903.04209.*

Kazemitabar, Jalil, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar (2017) "Variable importance using decision trees", in *Advances in Neural Information Processing Systems 30*, pp. 426–435.

Kock, Anders Bredahl and Timo Teräsvirta (2014) "Forecasting performances of three automated modelling techniques during the economic crisis 2007–2009", *International Journal of Forecasting*, Vol. 30, No. 3, pp. 616–631.

Lemaire, Vincent, Raphael Féraud, and Nicolas Voisine (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.

Lipton, Zachary Chase (2016) "The Mythos of Model Interpretability", *ArXiv e-prints*, Vol. 1606.03490.

Lundberg, Scott and Su-In Lee (2017) "A Unified Approach to Interpreting Model Predictions", in *Advances in Neural Information Processing Systems 30*, pp. 4765–4774.

Lundberg, Scott, Gabriel Erion, and Su-In Lee (2018) "Consistent individualized feature attribution for tree ensembles", *ArXiv e-prints*, Vol. 1802.03888.

Lundberg, Scott M, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee (2020) "From local explanations to global understanding with explainable ai for trees", *Nature Machine Intelligence*, Vol. 2, No. 1, pp. 56–67.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos (2018a) "The m4 competition: Results, findings, conclusion and way forward", *International Journal of Forecasting*, Vol. 34, No. 4, pp. 802–808.

——— (2018b) "Statistical and machine learning forecasting methods: Concerns and ways forward", *PloS one*, Vol. 13, No. 3.

McCracken, Michael W and Serena Ng (2016) "FRED-MD: A monthly database for macroeconomic research", *Journal of Business & Economic Statistics*, Vol. 34, No. 4, pp. 574–589.

Miller, Tim (2019) "Explanation in artificial intelligence: Insights from the social sciences", *Artificial Intelligence*, Vol. 267, pp. 1–38.

Ng, Serena and Jonathan H Wright (2013) "Facts and challenges from the great recession for forecasting and macroeconomic modeling", *Journal of Economic Literature*, Vol. 51, No. 4, pp. 1120–54.

Parmezan, Antonio Rafael Sabino, Vinicius MA Souza, and Gustavo EAPA Batista (2019) "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model", *Information Sciences*, Vol. 484, pp. 302–337.

Plumb, Gregory, Denali Molitor, and Ameet S Talwalkar (2018) "Model agnostic supervised local explanations", in *Advances in Neural Information Processing Systems*, pp. 2515–2524.

Racine, Jeff (2000) "Consistent cross-validatory model-selection for dependent data: hv-block cross-validation", *Journal of Econometrics*, Vol. 99, No. 1, pp. 39–61.

Ribeiro, Marco, Sameer Singh, and Carlos Guestrin (2016) "Why should I trust you?": Explaining the predictions of any classifier", , Proceedings of the 22nd ACM SIGKDD, pp. 1135–11134.

Robnik-Šikonja, Marko and Igor Kononenko (2008) "Explaining classifications for individual instances", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 20, No. 5, pp. 589–600.

Rudin, Cynthia (2019) "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead", *Nature Machine Intelligence*, Vol. 1, No. 5, pp. 206–215.

Sermpinis, Georgios, Charalampos Stasinakis, Konstantinos Theofilatos, and Andreas Karathanasopoulos (2014) "Inflation and unemployment forecasting with genetic support vector regression", *Journal of Forecasting*, Vol. 33, No. 6, pp. 471–487.

Shapley, Lloyd (1953) "A value for n-person games", *Contributions to the Theory of Games*, Vol. 2, pp. 307–317.

Shrikumar, Avanti, Peyton Greenside, and Kundaje Anshul (2017) "Learning important features through propagating activation differences", *ArXiv e-prints*, Vol. 1704.02685.

Snijders, Tom A. B. (1988) "On cross-validation for predictor evaluation in time series", in Theo K. Dijkstra ed. *On model uncertainty and its statistical implications*: Springer, pp. 56–69.

Stock, James H and Mark W Watson (2002) "Forecasting using principal components from a large number of predictors", *Journal of the American Statistical Association*, Vol. 97, No. 460, pp. 1167–1179.

Štrumbelj, Erik and Igor Kononenko (2010) "An efficient explanation of individual classifications using game theory", *Journal of Machine Learning Research*, Vol. 11, pp. 1–18.

Sundararajan, Mukund and Amir Najmi (2019) "The many shapley values for model explanation", *ArXiv e-prints*, Vol. 1908.08474.

Wang, Mengqiu and Christopher D Manning (2013) "Effect of non-linear deep architecture in sequence labeling", in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 1285–1291.

Young, Peyton (1985) "Monotonic solutions of cooperative games", *International Journal of Game Theory*, Vol. 14, pp. 65–72.