WP/20/45

IMF Working Paper

Deus ex Machina? A Framework for Macro Forecasting with Machine Learning

by Marijn A. Bolhuis and Brett Rayner

INTERNATIONAL MONETARY FUND

 WP/20/45

IMF Working Paper

European Department

Deus ex Machina? A Framework for Macro Forecasting with Machine Learning[1]

Prepared by Marijn A. Bolhuis and Brett Rayner

Authorized for distribution by Donal McGettigan

February 2020

Abstract

We develop a framework to nowcast (and forecast) economic variables with machine learning techniques. We explain how machine learning methods can address common shortcomings of traditional OLS-based models and use several machine learning models to predict real output growth with lower forecast errors than traditional models. By combining multiple machine learning models into ensembles, we lower forecast errors even further. We also identify measures of variable importance to help improve the transparency of machine learning-based forecasts. Applying the framework to Turkey reduces forecast errors by at least 30 percent relative to traditional models. The framework also better predicts economic volatility, suggesting that machine learning techniques could be an important part of the macro forecasting toolkit of many countries.

JEL Classification Numbers: C53, C45.

Keywords: Forecasts, Nowcasting, Machine learning, GDP growth, Cross-validation, Random Forest, Ensemble, Turkey.

Authors' E-Mail Addresses: marijn.bolhuis@mail.utoronto.ca; brayner@imf.org

[1] The authors would like to thank Donal McGettigan, Alex Culiuc, Vincenzo Guzzo, Romain Lafarguette, and participants at the IMF EUR seminar for helpful comments and suggestions, and Morgan Maneely for outstanding research assistance. All remaining errors are our own. R codes are available at marijnbolhuis.org.

CONTENTS

Deus ex machina *(noun)*

de·us ex ma·chi·na | ˈdā-əs-ˌeks-ˈmä-ki-nə

"An unexpected power or event that saves a situation that seems without hope, especially in a play or novel."

–Oxford Dictionary

I. INTRODUCING MACHINE LEARNING

Traditional forecasting methods often provide poor macro forecasts. Techniques based on ordinary least squares (OLS) struggle to overcome several issues, including collinearity, dimensionality, predictor relevance, and nonlinearity. Some state-of-the-art forecasting models, including dynamic factor models, can help address collinearity and dimensionality problems, but do not address predictor relevance and nonlinearity problems. As a result, even state-of-the art forecasting models often result in large forecast errors. Furthermore, dynamic factor models perform particularly poorly when the variable to be predicted is volatile, such as output growth in many emerging market and developing economies.

Machine learning (ML) methods present an alternative to traditional forecasting techniques. ML models can outperform traditional forecasting methods because they emphasize out-of-sample (rather than in-sample) performance and better handle nonlinear interactions among a large number of predictors. ML methods are specifically designed to learn complex relationships from past data while resisting the tendency of traditional methods to over-extrapolate historical relationships into the future. Indeed, a literature is beginning to emerge which suggests that ML methods often outperform traditional linear regression-based methods in terms of accuracy and robustness.[2]

We develop a framework to use ML methods for macro forecasting. We use the framework to nowcast (and forecast) economic growth in Turkey and are able to reduce forecast errors by at least 30 percent relative to traditional models. Importantly for Turkey and other countries with volatile economies, the framework also better predicts large swings in the growth rate, suggesting that machine learning techniques could be an important part the macro forecasting toolkit of many countries. We also attempt to improve transparency and interpretability of ML forecasts by uncovering the contribution of each predictor to individual forecasts.

[2] For example, Smeekes & Wijler (2016) and Carrasco & Rossi (2016) find that penalized ML methods tend to outperform traditional factor models in terms of forecast accuracy. The former also show that ML methods are more robust to model misspecification. Tu & Lee (2018) show that traditional factor models tend to be inferior to supervised factor models that perform variable selection. Kim & Swanson (2014) assess the predictive accuracy of both traditional, ML and 'hybrid' forecasting methods and find that the latter two dominate in most settings. Tiffin (2016), Jung et al. (2018), and Richardson et al. (2018) use different ML methods to forecast GDP growth for several countries. Smalter Hall (2018) employs ML methods to forecast unemployment in the United States and Medeiros et al. (2018) forecast inflation in Brazil.

II. THE BASICS OF FORECASTING—A BIAS-VARIANCE TRADEOFF

All forecasting methods aim to minimize expected forecast errors. Forecasting consists of selecting a function that maps indicator data to a forecast while minimizing a particular loss function. Suppose a researcher wants to forecast a variable y_t (e.g., real GDP growth) using K predictor variables summarized in the $K \times 1$ vector X_t, with the h-step ahead forecast of y_t denoted as y_{t+h}:

$$y_{t+h} = f(X_t) + \epsilon_{t+h}$$

where ϵ_{t+h} is an error term. The goal is to forecast y_{t+h} by choosing the function $f(\cdot)$ that minimizes the average loss:

$$\min_{f(X_t)} L\left(y_{t+h} - f(X_t)\right)$$

where $L(\cdot)$ is a loss function that assigns relative weights to different forecast errors. If the loss function is quadratic, for example, the expected loss to minimize by picking the function $\hat{f}(X_t) = \widehat{y_{t+h}}$ can be decomposed as (James et al., 2013):

$$\underbrace{E((f(X_t) - \widehat{y_{t+h}})^2)}_{\text{exp. squared forecast error}} = \underbrace{[E(\widehat{y_{t+h}}) - f(X_t)]^2}_{\text{squared bias}} + \underbrace{Var[\widehat{y_{t+h}}]}_{\text{variance}} + \underbrace{\sigma^2}_{\text{irreducible error}} \quad (1)$$

where the first term on the right-hand side is the squared bias of the forecast, the second term is the variance of the forecast and the third term is the idiosyncratic contribution of the error term to total loss.

The optimal forecast uses the past to predict the future without over-extrapolating. Minimizing the loss function (1) amounts to picking the function $\hat{f}(X_t)$ that minimizes the expected sum of the squared bias and the variance of the forecast. Unfortunately, it is typically impossible to reduce both terms simultaneously (Annex I). This bias-variance tradeoff is a central concept in both the forecasting and the machine learning literatures (James et al., 2013). In general, more complex forecasting models exhibit lower bias, because they better capture nuances in the mapping from X_t to y_{t+h}.[3] However, as complex models provide sharper predictions, they are also more likely to capture perturbations (or 'noise') in the historical data that are uninformative for future predictions. This tendency, known as 'overfitting', increases the variance of forecasts, potentially resulting in higher forecast errors.

[3] There is no universal definition of complexity in the ML literature, as the degree of complexity often depends on the nature of the underlying learning model. Common sources of complexity are the number of included variables (e.g., penalized linear models), the number of parameters a model 'learns' (e.g., random forest), the number of relationships specified (e.g., neural networks), and the number of observations used per individual prediction (e.g., nearest neighbors).

A. Shortcomings of OLS-Based Forecasting Methods

Forecasting methods based on OLS struggle to optimize the bias-variance tradeoff. Suppose the predictors are mean zero and the error term is i.i.d. $N(0, \sigma^2)$ and independent of X_t (Stock & Watson, 2006). With OLS, the expected loss under the quadratic loss function becomes:

$$E\left(\left(f(X_t) - \widehat{y_{t+h}^{OLS}}\right)^2\right) = [E(\widehat{y_{t+h}}) - f(X_t)]^2 + (X_t' X_t \frac{1}{T} + 1)\sigma^2 \quad (2)$$

and several issues arise, including:

- **Collinearity.** The variance of the OLS forecast is increasing in the degree of correlation between predictors. To see this, note that the expected value of the inner product $X_t'X_t$ (for a given observation) equals the covariance of X_t. The more correlated the predictors are, the higher this covariance.

- **Dimensionality.** The variance of the OLS forecast is increasing in the number of predictors, K. To see this, suppose the predictors X_t are orthogonal such that $\frac{1}{T}\sum_{t=1}^{T} X_t X_t' = I_K$ (a $K \times K$ identity matrix). In this case it can be shown that (Stock & Watson, 2006):

  $$\widehat{y_{t+h}^{OLS}} \sim N\left(E\left(\widehat{y_{t+h}^{OLS}}\right), \frac{c\ K\ \sigma^2}{T}\right)$$

  where c is a constant. For a given number of historical observations, T, the variance of the forecast is proportional to the number of predictors.

- **Predictor relevance.** Related to dimensionality, irrelevant predictors unambiguously increase the forecast error because they do not reduce bias, but increase the forecast variance by increasing $X_t'X_t$.

- **Nonlinearity.** If the data-generating process (DGP) is non-linear, the OLS forecast is biased. To see this, note that the first term on the right-hand side of (2) is minimized at zero if $f(X_t) = E(\widehat{y_{t+h}})$, which is the case if the underlying model is linear, i.e., $f(X_t) = \beta' X_t$.

State-of-the-art forecasting techniques such as dynamic factor models can address some of these issues. Specifically, factor models (Annex II) aim to address collinearity and dimensionality by summarizing the variation in the predictor data using a small set of orthogonal factors.[4] In particular, if the selected indicators capture the underlying forces that affect the forecasted variable, and there is a high degree of co-movement among indicators, this variation can be explained by a small set of latent variables (Sargent & Sims, 1977).

[4] For a detailed review, see Stock & Watson (2006, 2011, 2012, 2017) and Bai & Ng (2008).

Factor models do not, however, address predictor relevance or nonlinearity. In attempting to summarize the information content of a large number of predictors into a small number of factors, there may be settings where the predictors follow a factor structure, but the factors do not predict the forecast variable (Tu & Lee, 2018). While factor models can help reduce dimensionality, they do not provide a means to identify the most relevant predictors. Furthermore, factor models rely on the assumption that the DGP follows a linear factor structure, which may not necessarily be the case.

B. The Advantages of Machine Learning Methods

Unlike traditional forecasting techniques, ML methods are specifically designed to optimize the bias-variance tradeoff. In particular, ML models can address the above issues with which traditional forecasts have struggled because they select predictors to optimize out-of-sample (rather than in-sample) performance and are better able to handle nonlinear interactions among a large number of predictors (Annex III). In this study we focus on three specific ML methods: Random Forest; Gradient Boosted Trees; and Support Vector Machines.

Random Forest (RF) is an algorithm that uses forecast combinations of multiple decision trees to construct an aggregate forecast. The key elements of RF include:

- *Decision trees.* A decision tree is an algorithm that repeatedly separates categorical data into two groups, with each split chosen by the algorithm to yield the largest reduction in the forecast error of the variable of interest. Regression trees are a type of decision tree used for predicting a continuous variable and are particularly well suited for nonlinear relationships. A regression tree minimizes the forecast error by repeatedly splitting the continuous data into two groups, with a prediction for each group that is based on the mean of that group's data (Hastie et al., 2009).[5] Decision trees can be as complex (i.e., long) as needed to fit to the in-sample data well. However, they often 'overfit' the in-sample data at the expense of out-of-sample performance Also, decision trees use local, rather than global, optimization which can create path dependence and model instability. Modifications to the basic decision tree, such as random sampling, are often made to prevent overfitting and improve model performance.

- *Random sampling.* RFs modify the decision tree approach in two ways to maximize the information content of the data by using subsamples of observations and predictors. First, they use bootstrap aggregation ('bagging') by building each individual tree on only a random sample of the observations in the training data. Second, at each split in the tree, the RF algorithm uses only a random subsample of the predictors. Bagging therefore generates a large number of uncorrelated trees. Individually, the trees tend to have low bias but poor out-of-sample accuracy due to high variance (i.e., they overfit

[5] Formally, regression trees pick regions R_m and region predictions c_m (for M different regions) and: $min_{\{R_m,c_m\}_{m=1}^{M}} \sum[y_t - \sum_{m=1}^{M} c_m I(X_t \in R_m)]^2$

on the training data). However, for a large enough number of uncorrelated trees, these errors tend to average out to zero. RF is one of the most popular ML algorithms available because it is computationally easy to use and requires almost no tuning of model parameters. This makes it an ideal algorithm for forecasting on time-series data with relatively few observations.

Gradient Boosted Trees (GBT) is an algorithm that constructs sequential decision trees to learn from previous trees' errors. Just like the RF, GBT combines individually-weak trees into a robust forecast. The algorithm starts out by training an initial decision tree on the historical data. It then uses the prediction errors from the first tree to train a second tree. In turn, the errors from the second tree are used to train the third tree, etc. After the final iteration, the algorithm uses the sum of the individual predictions for the final forecast.[6] Whereas RF combines relatively deep trees with low bias and high variance, GBT combines relatively shallow trees with high bias and low variance. As each subsequent tree targets the bias from the previous tree, the bias errors of subsequent trees tend to sum towards zero, resulting in an overall prediction with both low bias and low variance.

Support Vector Machine (SVM) is an algorithm that constructs hyperplanes to partition predictor combinations and make a point forecast for each of the sections. Unlike tree-based algorithms, SVM is similar to kernel regression with a penalty imposed on the use of coefficients (i.e., penalized kernel regression). Formally, SVM regressions find the function $f(X_t) = X_t'\beta + b$ and observation-specific slack constants ζ_i and ζ_i^* that minimize $\beta'\beta + C\sum(\zeta_i + \zeta_i^*)$, subject to $y_i - f(X_i) \leq \epsilon + \zeta_i$ and $f(X_i) - y_i \leq \epsilon + \zeta_i^*$. The complexity parameters ϵ and C govern the acceptable margin and the penalty imposed on observations that lie outside this margin. The cost parameter, C, mainly determines the degree of model complexity. If C = 0, the algorithm disregards individual deviations and constructs the simplest hyperplane for which every observation is still within the acceptable margin ϵ. For sufficiently large C, the algorithm will construct the most complex hyperplane that predicts the outcome for the training data with zero error, i.e. the algorithm will fit the training data perfectly. Through cross-validation, SVM finds the optimal value of C that balances this bias-variance tradeoff and maximizes out-of-sample accuracy on the historical data.

III. A Framework for Macro Forecasting with Machine Learning

A. Limiting Preselection

We apply the framework to Turkey, a country for which traditional forecasting techniques have been unsatisfactory. We collect a database of country-specific and global indicators, with 234 separate series in total (Tables A5.1 and A5.2). The data consist of an array of mixed-frequency (monthly and quarterly) leading and coincident indicators from

[6] Let $F_1(X_t)$ denote the in-sample prediction from the first decision tree. The second tree thus constructs the tree that solves $min_{F_2(X_t)} \sum[y_t - F_1(X_t) - F_2(X_t)]^2$, the third tree solves $min_{F_3(X_t)} \sum[y_t - F_1(X_t) - F_2(X_t) - F_3(X_t)]^2$ etc. With three trees, the final forecast equals $F_1(X_t) + F_2(X_t) + F_3(X_t)$.

Haver Analytics. We then apply some basic transformations to each raw indicator. In addition to deflating nominal indicators where appropriate and including 12 lags, we include two transformations of each indicator series. For stationary variables (e.g., capacity utilization, consumer confidence), we use the level and quarter-on-quarter difference. For non-stationary variables (e.g., production, money) we take first- and second-order log differences. Moreover, we construct several indicators such as the sovereign term spread, sovereign yield spread, the US sovereign term spread, and the US high yield spread.[7]

We use hard thresholding to help address the dimensionality problem of a large set of predictors. More data is not always better and can increase forecast errors even when using dimensionality reduction techniques (Boivin & Ng, 2006). Hard thresholding (Bai & Ng, 2007) consists of regressing the forecast variable on its lags and each individual indicator and selecting all indicators with an absolute t-statistic above a certain threshold. In this case, the threshold is obtained by comparing out-of-sample performance of forecasts across a range of thresholds and choosing the threshold that delivers the lowest forecast errors.

B. Identifying Complementary Algorithms

The chosen ML models (RF, GBT, and SVM) are relatively simple and accessible. All three models require little parameter tuning and are thus less likely to overfit than other types of ML models.[8] In addition, all three models are computationally relatively inexpensive.

The three models are also complementary. We combine the individual ML models into several ensembles. Ensembles can lower forecast errors relative to any of the individual models by producing a single, weighted forecast of the individual models. Ensembles tend to outperform individual forecasts, especially when the models are relatively independent yet similar in forecast accuracy (Timmermann, 2006). In this case, we combine the forecasts of the three models using equal weights (Ensemble 1), inverse root mean squared error (RMSE) weights (Ensemble 2) and inverse-RMSE rank weights (Ensemble 3).

Our ensembles combine the best aspects of the individual models. More complex ML methods such as SVM tend to overfit when training data is relatively limited (e.g., short time series), resulting in predictions that are sensitive to small perturbations in the leading indicators. As such, SVM acts as a counterweight against GBT and RF, which tend to

[7] A common drawback of assembling a large dataset is that many series may have missing observations for a significant period of time. ML techniques offer a way to impute missing values in order to take advantage of all available indicators and observations. Specifically, the algorithm we use initially imputes the missing values with each indicator's median, then runs a Random Forest. It then replaces the missing value of an indicator with the weighted average of the non-missing observations, where the weights are the proximities (i.e., the fraction of final nodes shared by two observations) of the random forest.

[8] Avoiding overfitting to a complex model is our main reason for not deploying neural networks, which tend to require large datasets for good performance. Indeed, in Jung et al. (2018) elastic net tends to outperform recurrent neural networks when forecasting GDP growth. We avoid linear penalized regression methods such as ridge, LASSO and elastic net because they are sensitive to large, unexpected changes in predictor values that are not in the training dataset. As a result, forecasts using these methods tend to be unstable for datasets like ours.

outperform during stable periods of growth, whereas the SVM is more likely to pick up the effect of extreme shocks.

C. Evaluating Performance and Interpreting Results

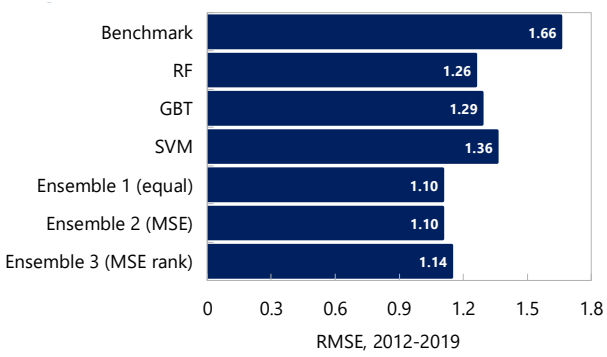
To evaluate model performance, we use rolling out-of-sample forecasts. This method provides an intuitive test of how the models would have performed in the past. Specifically, for each individual nowcast, we split the historical data available at the time of the nowcast into a training set and a test set and use cross-validation techniques to tune the parameters of the model (Annex III). Once calibrated, we then run the model using all historical data available at that time to obtain each individual nowcast, and ultimately assess the performance of the model.

We also assess the importance of each predictor by constructing variable importance measures for each of the ML models. To improve transparency and interpretability of our ML forecasts, we identify the contribution of each predictor to individual forecasts. Shapley Values provide an intuitive summary of each variable's contribution to the forecast's deviation from its historical mean (Annex IV).

IV. Results—More Accurate Forecasts

Individual ML methods can improve forecast performance. Figure 1 plots the RMSE of the benchmark factor model nowcast, against the RSME of the three machine learning models (RF, GBT and SVM) for the 2012– 2019 period.[9] The benchmark has a RMSE of 1.66, which corresponds to a mean absolute deviation of about 1.2 percentage points per nowcast. Using RF, GBT, or SVM reduces the RMSE by 24, 22, and 18 percent, respectively. We find similar improvements for the forecast models (Figure A5.1), where the RF and GBT outperform the benchmark by 18 and 22 percent.

Figure 1. Nowcast RMSE

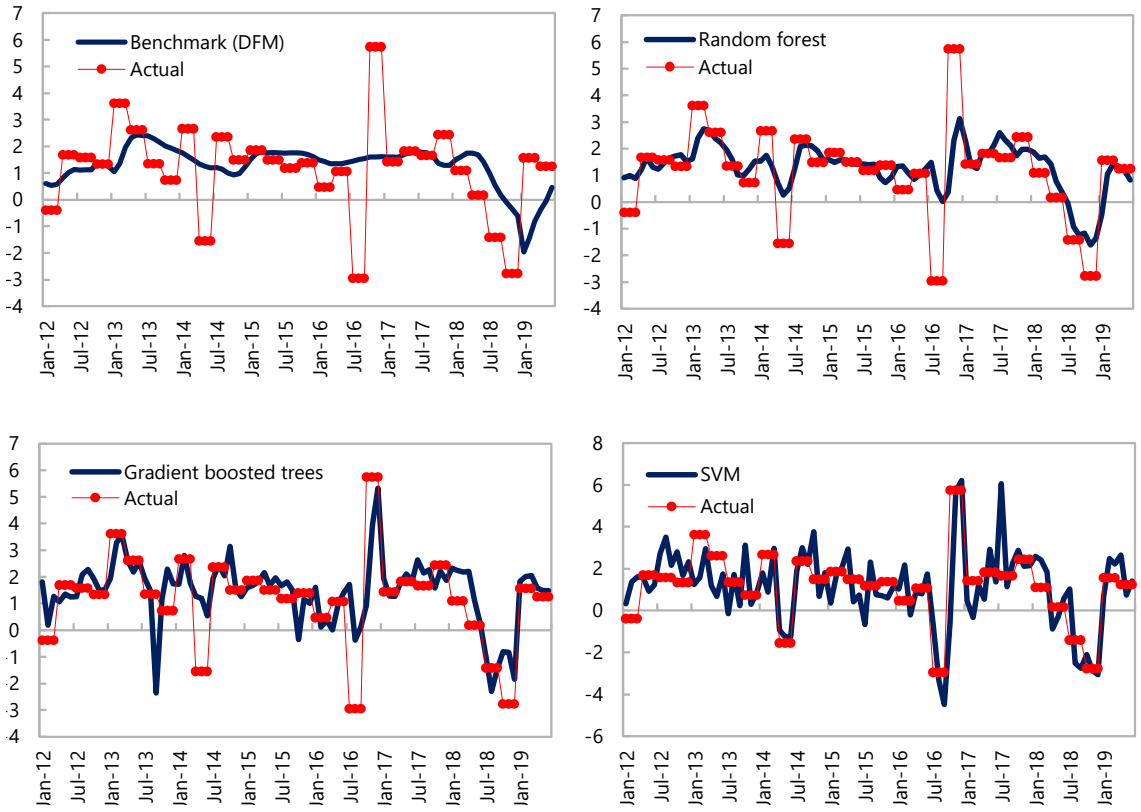


ML methods not only increase average accuracy, but also better predict economic volatility. Figure 2 plots the rolling out-of-sample nowcasts against actual quarterly real GDP growth. While the forecasts of the benchmark factor are relatively stable, the three ML methods all better predict the large growth swings seen in 2014, 2016 and 2018–19. Figure 3 plots the RMSE for the different nowcast models for 'volatile' quarters only, where we define a volatile quarter as one with a more than 3 percentage points higher or lower growth rate than

[9] We compare the performance of the ML models and ensembles against a more traditional forecasting model. As a benchmark, we use a static dynamic factor model (DFM). We employ three factors, as is standard in the DFM literature (Barhoumi et al., 2013).

the previous quarter. In this setting, SVM outperforms any of the models, improving upon the factor model by 39 percent. Moreover, the ML methods tend to move closer to the actual quarterly growth rate as we get closer to the end of the quarter. These patterns are similar in case of the forecast (Figure A5.2).

Figure 2. Individual Model Nowcasts vs. Actual Real GDP Growth
(percent, quarter on quarter seasonally adjusted)



The accuracy of ML methods increases with the availability of training data. Figure 4 plots the smoothed root-squared (or absolute) error of the benchmark model and the ML methods. Relative to the benchmark, nowcast errors decrease substantially over time as more data become available to train and test the models. From 2012 to 2019, RF and GBT gain roughly 60 percent in accuracy relative to the benchmark, while the SVM lowers errors by almost 80 percent. Again, we observe similar patterns for the forecast (Figure A5.1).

Figure 3. Nowcast RMSE, Volatile Quarters

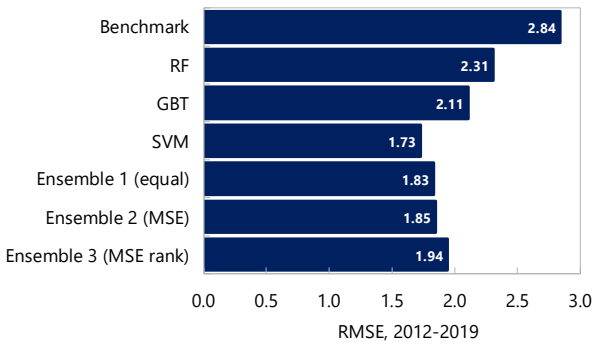
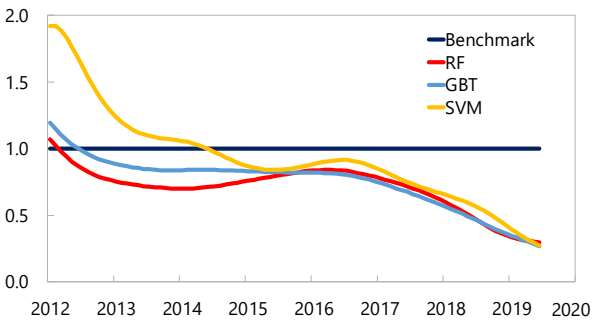


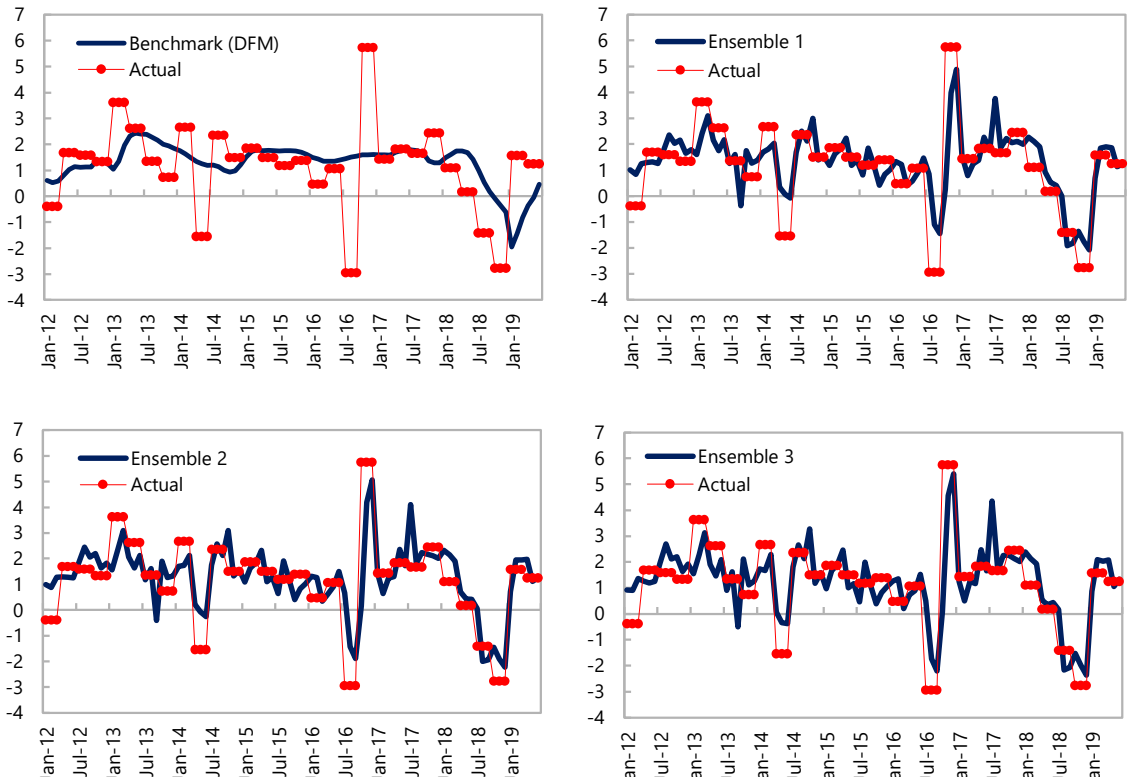
Figure 4. Nowcast Smoothed RSE



Different ML methods have different strengths, making them ideal as combinations in ensembles. RF seems to have good predictive performance overall, but does not fully capture the large swings in growth (Figure 2). Predictions from GBT are a bit more volatile, but also better capture the large swings in growth. SVM appears best at capturing the large swings, but at the expense of even more volatility.

Ensembles exploit the different strengths of the individual ML models to further improve predictions. Figure 5 plots the RMSE of the three ensemble nowcasts and the benchmark factor model for the 2012–2019 period. The ensembles differ little in terms overall performance. All outperform the benchmark by about 33 percent, which is an improvement of at least 9 percentage points compared to the individual models. The outperformance of the ensembles is also more stable over time.

Figure 5. Ensemble Model Nowcasts vs. Actual Real GDP Growth
(percent, quarter on quarter seasonally adjusted)



Measures of variable importance can improve the economic interpretability of the predictions. Using nowcasts for Turkey as an illustrative example:

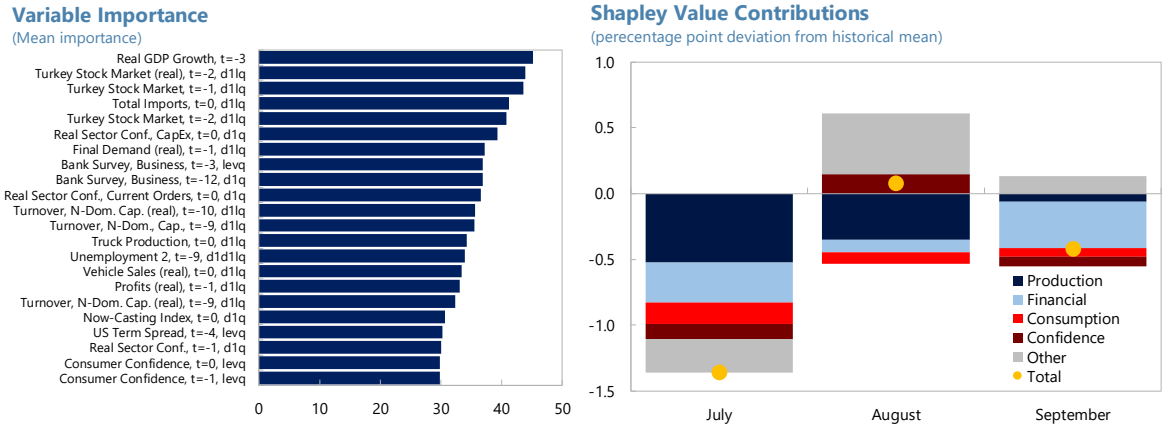
- First, we construct variable importance measures for Ensemble 1 (equal weights between RF, GBT and SVM), which are model-specific estimates of the relative importance of predictors in generating the forecasts.[10] These are scaled from 0 to 100 in

[10] We use R package *caret*.

ascending order of relative importance. Figure 6 plots the 25 most importance predictors for the Turkey nowcast model in July 2019. In addition to the previous quarter's GDP growth, the nowcast mainly relies on changes in the stock market, imports, business confidence, unemployment and the manufacturing PMI.

- Second, we use Shapley Values to decompose recent Turkey nowcasts into contributions of different predictor categories. Figure 6 also plots the Shapley Values by categories for three Turkey nowcasts. Relative to the historic mean, lower production indicators and higher inflation contributed to lower forecasts in all months. Over time, the nowcast mainly deteriorated due to worsening financial conditions and consumption indicators.

Figure 6. Variable Importance and Shapley Values



V. CONCLUSIONS

Machine learning techniques can improve forecasting performance relative to traditional models. Techniques based on OLS struggle to overcome several issues, including collinearity, dimensionality, predictor relevance, and nonlinearity. As a result, even state-of-the art forecasting models often result in large forecast errors, especially when the variable to be predicted is volatile, such as output growth in many emerging market and developing economies. ML models can outperform traditional forecasting methods because they emphasize out-of-sample (rather than in-sample) performance and better handle nonlinear interactions among a large number of predictors. ML methods are specifically designed to learn complex relationships from past data while resisting the tendency of traditional methods to over-extrapolate historical relationships into the future.

VI. REFERENCES

Bai, J., & Ng, S., 2008. "Forecasting Economic Time Series Using Target Predictors," *Journal of Econometrics*, *146*(2), 304–317.

Bai, J., & Ng, S., 2008. "Large Dimensional Factor Analysis," *Foundations and Trends in Econometrics*, *3*(2), 89–163.

Barhoumi, K., Darné, O., & Ferrara, L., 2014. "Dynamic Factor Models: A Review of the Literature," *Journal of Business Cycle Research*, *2013*(2), 73.

Boivin, J., & Ng, S., 2006. "Are More Data Always Better for Factor Analysis?" *Journal of Econometrics*, *132*(1), 169–194.

Carrasco, M., & Rossi, B., 2016. "In-sample Inference and Forecasting in Misspecified Factor Models," *Journal of Business & Economic Statistics*, *34*(3), 313–338.

Hastie, T., Tibshirani, R., & Friedman, J., 2009. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer Science & Business Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R., 2013. "An Introduction to Statistical Learning," (Vol. 112, p. 18). New York: Springer.

Jung, J. K., Patnam, M., & Ter-Martirosyan, A., 2018. "An Algorithmic Crystal Ball: Forecasts Based on Machine Learning," IMF Working Paper 18/230.

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

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., & Zilberman, E., 2019. "Forecasting Inflation in a Data-rich Environment: the Benefits of Machine Learning Methods," *Journal of Business & Economic Statistics*, 1–45.

Richardson, A., & Mulder, T., 2018. "Nowcasting New Zealand GDP Using Machine Learning Algorithms," Mimeo.

Sargent, T. J., & Sims, C. A., 1977. "Business Cycle Modeling Without Pretending to Have Too Much A Priori Economic Theory," New Methods in Business Cycle Research, 1, 145–168.

Shapley, L. S., 1953. "A Value For N-person Games," *Contributions to the Theory of Games*, *2*(28), 307–317.

Smalter Hall, A., 2018. "Machine Learning Approaches to Macroeconomic Forecasting," *Economic Review-Federal Reserve Bank of Kansas City*, *103*(4), 63.

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

Stock, J. H., & Watson, M. W., 2006. "Forecasting With Many Predictors," *Handbook of Economic Forecasting*, *1*, 515–554.

Stock, J. H., & Watson, M., 2011. "Dynamic Factor Models," *Oxford Handbooks Online*.

Stock, J. H., & Watson, M., 2012. "Generalized Shrinkage Methods for Forecasting Using Many Predictors," *Journal of Business & Economic Statistics*, *30*(4), 481–493.

Stock, J. H., & Watson, M., 2017. "Twenty Years of Time Series Econometrics in Ten Pictures," *Journal of Economic Perspectives*, *31*(2), 59–86.

Tiffin, A., 2016. "Seeing in the Dark: A Machine-Learning Approach to Nowcasting in Lebanon," IMF Working Paper 16/56.

Timmermann, A., 2006. "Forecast Combinations," *Handbook of Economic Forecasting*, *1*, 135–196.

Tu, Y., & Lee, T. H., 2019. "Forecasting Using Supervised Factor Models," *Journal of Management Science and Engineering*.

ANNEX I. THE BIAS-VARIANCE TRADEOFF

We demonstrate the bias-variance tradeoff with two simple examples. Suppose a researcher has T periods of historical data on y_{t+h} and a set of predictors, X_t. The least complex model would simply forecast the historical mean of y_{t+h}. Doing so leads to substantial bias as it is unlikely that y_{t+h} is constant over time. However, the variance of this simple forecast is minimized. At the other extreme, a forecaster could pick one historical observation that it believes to be most representative ('closest') to the current environment in terms of X_t, and use this observation's historical outcome as the forecast. Such a complex forecast will have low bias but high variance.

The K-Nearest Neighbors algorithm is one way to minimize the bias-variance tradeoff. The two extreme types of forecasts described above are examples of the K-Nearest Neighbors (KNN) algorithm. This ML method uses observations in the historical data closest to X_t to form the forecast $\widehat{y_{t+h}}$, which is formally defined as (Hastie et al., 2009):

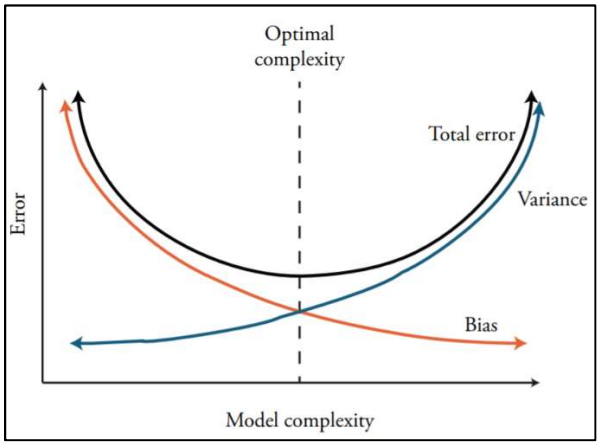
$$\widehat{y_{t+h}} = \frac{1}{K} \sum_{n \in N_K(X_t\}} y_n$$

where $N_K(X_t)$ is the neighborhood of the forecast defined by the K closest points n in the historical sample. This neighborhood is usually constructed using the Euclidian distance. KNN has a convenient closed form expression for expected loss:

$$E((f(X_t) - \widehat{y_{t+h}})^2) = \left[f(X_t) - \tfrac{1}{K}\textstyle\sum_{n \in N_K(X_t)} y_n\right]^2 + \sigma^2\left(\tfrac{1}{K} + 1\right)$$

which nicely summarizes the bias-variance tradeoff. The squared bias (first term on RHS) is monotonically increasing in K as observations 'farther' from X_t tend to be less informative for the forecast. The variance (second term on RHS) is monotonically decreasing in K. As a result, the K that minimizes forecast errors tends to be somewhere in between the two extreme cases. Figure 1.1 expresses this bias-variance tradeoff visually.

Figure 1.1. Model Complexity and the Bias-Variance Tradeoff



Source: Smalter Hall (2018)

ANNEX II. STATIC DYNAMIC FACTOR MODELS

Traditionally, the factor model literature assumes predictors take the form (Stock & Watson, 2006; Smeekes & Wijler, 2016):

$$x_{kt} = \lambda_i(L)' f_t + e_{it}$$

where x_{kt} is the predictor k time series observed at time t with zero mean and unit variance. f_t is a $Q \times 1$ vector containing latent factors and e_{it} is a idiosyncratic disturbance term. $\lambda_i(L)$ is a lag polynomial of order K_Q , often referred to as the "dynamic factor loadings." Both the factors and disturbances are assumed to be uncorrelated at all leads and lags. We also assume the forecast variable admits a factor structure:

$$y_{t+h} = \lambda_Y(L)' f_t + e_{yt}$$

the single forecasting equation for Y_{t+h} from (X) takes the form:

$$y_{t+h} = \beta(L) f_t + \gamma(L)\acute{} Y_t + \epsilon_{t+h}$$

where $\beta(L)$ is a lag polynomial, and ϵ_{t+h} is a conditional mean zero disturbance term. (Y) can be estimated using MLE, although this is computationally demanding and only consistent under somewhat restrictive assumptions. As a result, it is standard in the macro forecasting literature to rewrite the dynamic factor model summarized in (X) and (Y) in its *static* form, which can be estimated using principal components analysis (PCA).

If the lag polynomials $\beta(L)$ and $\lambda_i(L)$ have finite order K_P, we can rewrite (X) and (Y) as (Stock & Watson, 2006):

$$X_t = \Lambda F_t + u_t$$

$$y_{t+h} = \beta_F' F_t + \gamma(L) y_t + \mathrm{v}_{t+h}$$

Where Λ and F_t represent unobserved factor loadings and factors. u_t is an error term that is i.i.d. $N(0, \sigma_v^2)$ and independent of F_t. We can now recast estimating the general model as (Smeekes & Wijler, 2016):

$$y_{t+h} = f(\Lambda F_t + u_t) + \epsilon_{t+h}$$

For a given estimated $\widehat{\Lambda}$ and $\widehat{F_t}$, a static factor model assumes $f(\cdot)$ is linear and thus runs OLS such that:

$$\widehat{y_{t+h}^{DFM}} = (\widehat{F_t}\widehat{\Lambda})'\hat{\beta}$$

In this case, expected mean loss can be decomposed as:

$$E\left(\left(f(X_t) - \widehat{y_{t+h}^{DFM}}\right)^2\right) = [E(\widehat{y_{t+h}}) - f(X_t)]^2 + (\widehat{F_t}\widehat{\Lambda})'(\widehat{F_t}\widehat{\Lambda})\frac{V(u_t + \epsilon_{t+h})}{T} + V(u_t + \epsilon_{t+h})$$

ANNEX III. MACHINE LEARNING AND CROSS VALIDATION

Any ML algorithm can be cast as a series of general steps. ML methods are designed to find the optimal degree of complexity of a model that maximizes out-of-sample forecast accuracy. Suppose a researcher can pick $f(\cdot)$ from a class of models (e.g., linear, nearest neighbors). Given the model class, we can represent this as the researcher selecting parameters β and α:

$$\min_{\beta,\alpha} L\left(y_{t+h} - f(X_t, \beta)\right) \qquad \text{s.t. } \beta \in \Theta(\alpha)$$

where β determine the specific function within the model class, and α are 'tuning parameters' or 'regularizers' that determine the potential model complexity by constraining β to be in $\Theta(\alpha)$. The table below summarizes α and β of popular ML algorithms. Any ML algorithm consists of the following steps:

(a) For every degree of model complexity α, find the model configuration β that maximizes forecast accuracy on the training data.

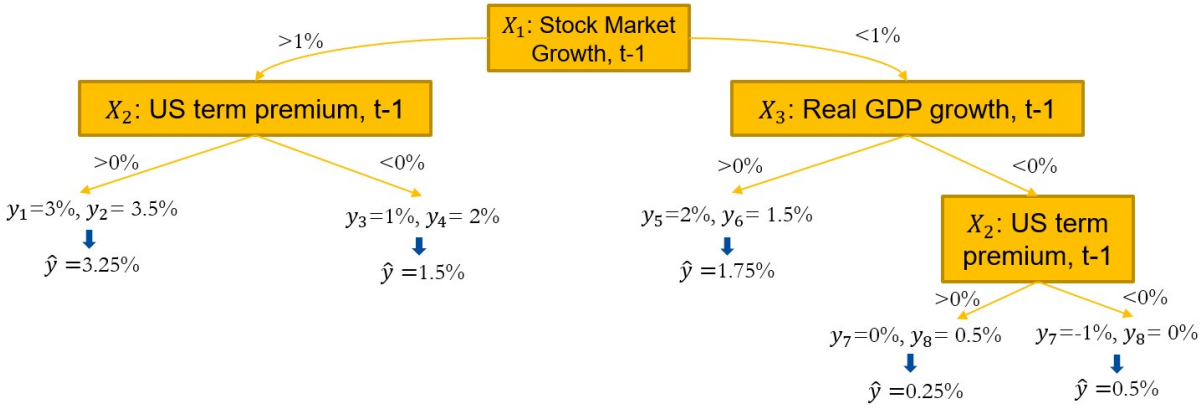
(b) Forecast on the test data using this model configuration β.

(c) Across all possible α, pick the degree of model complexity α that maximizes forecast accuracy on the test data.

This process of finding the optimal model parameters is called cross validation (CV). With CV, the entire data set is split into multiple subgroups ('folds'), which are all used as separate test sets. In this paper, we use 10 folds to tune the model complexity parameters.

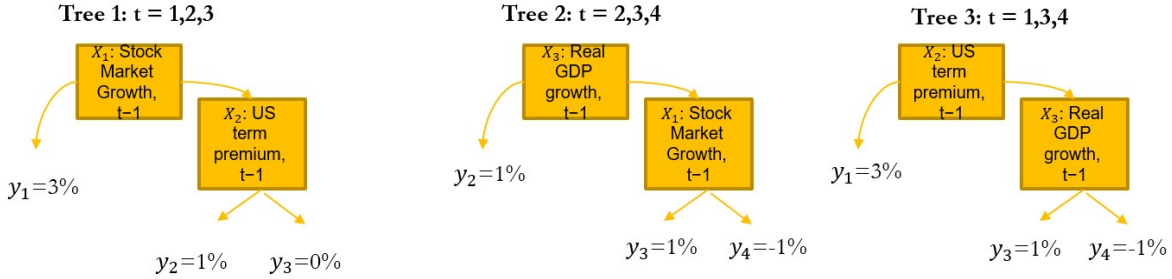
Class	**Model**	**β**	**α (most common)**
Linear	OLS	Linear coefficients	# of variables (e.g., forward stepwise regression)
	Ridge		L_2 norm penalty
	LASSO		L_1 norm penalty
	Elastic Net		Overall penalty, LASSO/ridge weight
Tree-based	Decision tree	Splits	Depth, # of leaves, observations per leaf
	Random forest	Splits, aggregation rule	(…), # of variables, observations per bootstrap
	Boosted Trees	Splits	(…), # of iterations
Prototype methods	KNN	Sets of nearest neighbors	K, weighting of neighbors
Support Vector Machines	Linear	Linear coefficients	Cost
	Polynomial	Coefficients	Cost, scale, degree
	Exponential	Coefficients	Cost, decay factor

Figure A3.1. Decision Tree Example



Notes: Figure plots a hypothetical decision tree nowcasting real GDP growth at time *t* using lags of real GDP growth, stock market growth, and the US term premium. Each leaf contains two training observations, and the trained decision tree predicts the average observed GDP growth of these two observations.

Figure A3.2. Random Forest Example



Notes: Figure plots a hypothetical decision Random Forest nowcasting real GDP growth at time *t* using lags of real GDP growth, stock market growth, and the US term premium. Each tree uses different observations and considers different variables at each split. In this example, each leaf contains only one training observation. The trained RF predicts the average of the GDP growth rates of the leaves that the new observation belongs to.

ANNEX IV. INTERPRETING FORECASTS: SHAPLEY VALUES

Shapley Values can help with the interpretation of the results of ML forecasts. Shapley Values are a concept from coalitional game theory that measures the contribution of each player in a game when the game's payoff depends on interactions ('coalitions') between the players (Shapley, 1953). They are constructed as the mean of each player's marginal contributions for every possible combination of other player's actions. In the context of ML methods, Shapley Values measure each variable's contribution to an individual prediction's deviation from the historical mean. For an OLS-based model, these contributions are the same as the predictor's coefficient multiplied by its specific value. Shapley Values are thus particularly useful for decomposing predictions from methods with interactions among predictors (e.g., a Random Forest).

The forecast decomposition using Shapley Values can be demonstrated with an example. Suppose we have trained an ML model to predict real GDP growth. The model predicts 5 percent for a certain period in which: (i) nominal credit growth is above 10 percent; and (ii) the country's major trading partner is expanding. We want to decompose this prediction into contributions of the two predictors (credit growth and trading partner growth). The matrix below summarizes the model's predictions contingent on the values of the two predictors. In this case, the two variables act as complements. The average marginal contribution of credit growth being above 10 percent is 4.5 percent, and the average marginal contribution of the trading partner expanding is 1.5 percent. If we assume the historical mean of the model forecast is 1 percent, the Shapley Values for credit growth and trading partner expansion would be 3 percent and 1 percent, respecitvely.[1]

	Credit growth>10%	Credit growth random	Marginal Contribution
Trading partner expanding	5%	1%	$1.5\% = \frac{5-4+1-(-1)}{2}$
Trading partner's state random	4%	-1%	
Marginal contribution	$4.5\% = \frac{5-1+4-(-1)}{2}$		

[1] $3.0 = (5-1)\cdot\frac{4.5}{4.5+1.5}$; $1.0 = (5-1)\cdot\frac{1.5}{4.5+1.5}$

ANNEX V. ADDITIONAL FIGURES AND TABLES

Figure A5.1. Forecast RSME

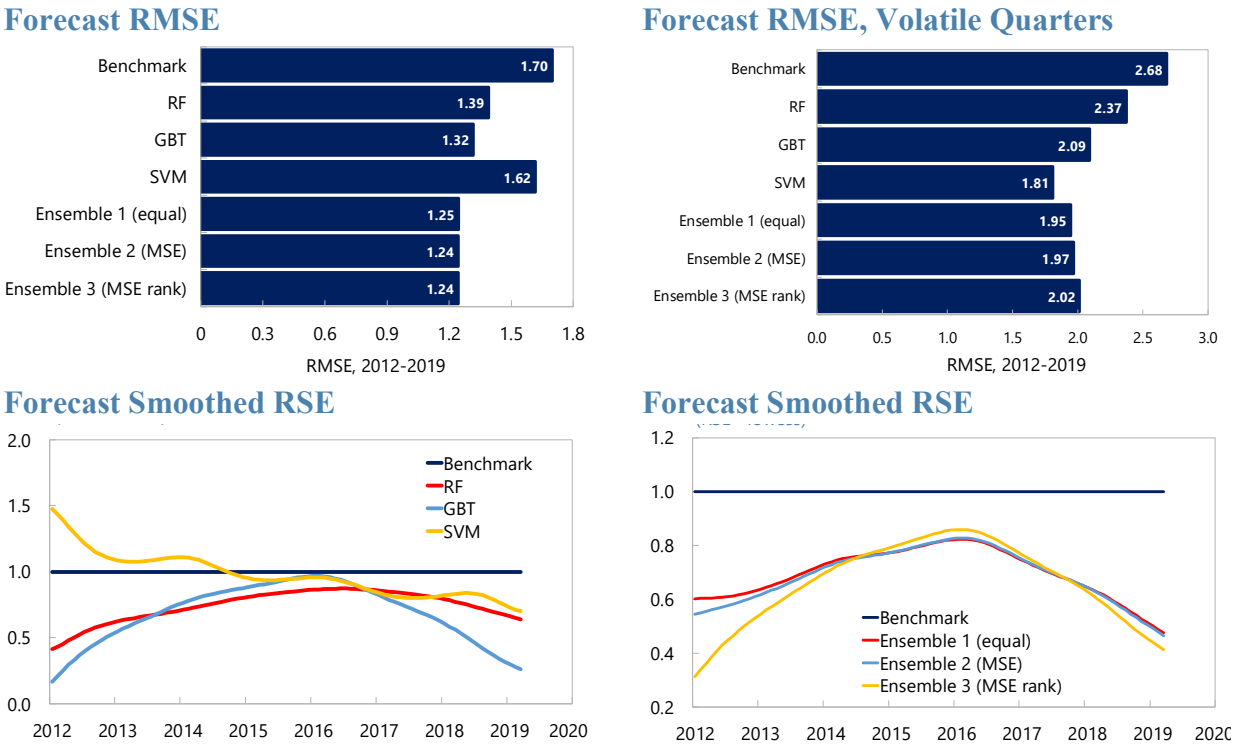


Figure A5.2. Rolling Out-of-Sample Forecasts vs. Actual Real GDP Growth
(percent, quarter on quarter seasonally adjusted)

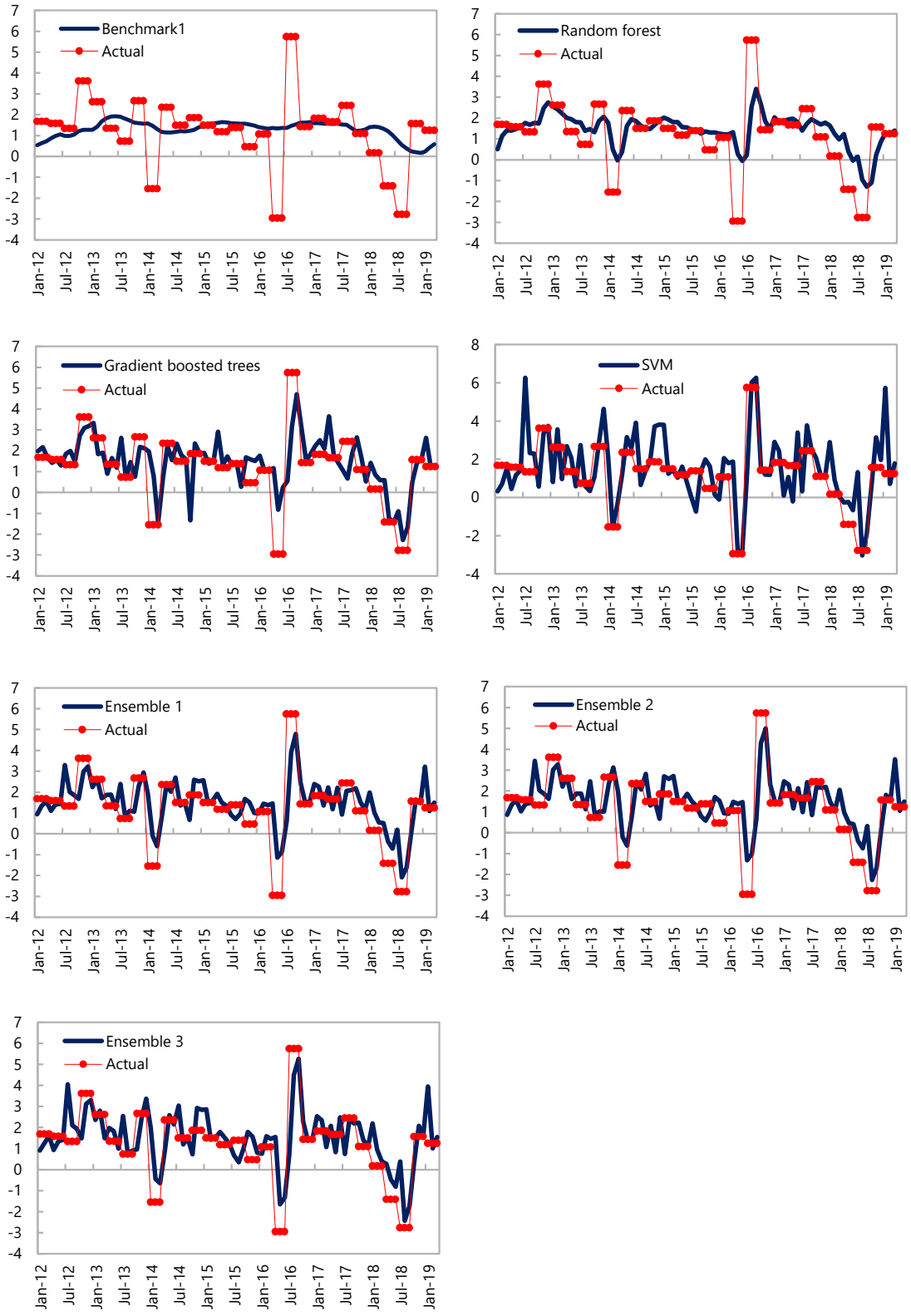


Table A5.1. Stationary Variables—Level and First Difference

Consumer Confidence	Capacity Utilization: Consumer Goods
Capacity Utilization: Intermediate Goods	Capacity Utilization: Investment Goods
Capacity Util: Manufacture of Other Non-metallic Mineral Prods	CCI: Buying Time of Durables [Present]
CCI: Assmt on Consumer Price Chg Rate [Last 12 Mo]	Real Sector Confidence Index
Real Sec Conf Index: Tot Amt of Orders [Curr Sit]	Real Sec Conf Index: Stocks of Fin Goods [Curr Sit]
Real Sec Conf Index: Vol of Output over Next 3 Months	Real Sector Conf Index: Employment over Next 3 Months
Real Sec Conf Index: Tot Amt of Orders Past 3 Months	Real Sector Conf Index: Exp Orders over Next 3 Months
Real Sector Conf Index: Fixed Investment Expend	Real Sector Conf Index: Gen Business Situation
CCI: Expect for Wage Chg Rate [12M vs Past 12M]	CCI: Number of Ppl Unemployed Exp. [Next 12M]
Reference Ask Rate: 1-Month	Reference Ask Rate: 3-Month
Reference Ask Rate: 9-Month	1 Week Repo Rate [Policy Rate]
Late Liquidity Borrowing Rate	Late Liquidity Lending Rate
Overnight Borrowing Rate	Overnight Lending Rate
Reference Bid Rate: 3-Month	Labor Force Survey: Labor Force Participation Rate
Labor Force Survey: Employment Rate	Labor Force Survey: Unemployment Rate
Unemployment Rate	Official/policy interest rates
10-year interest rates	Gross ED, percent of GDP
Current Account, percent of GDP	BoP, percent of GDP
Now-Casting Index (NCI)	World Uncertainty Index
Manufacturing PMI	Sovereign CDS Spread
Equity Fund Flows, monthly percent	Bond Fund Flows, monthly percent
Bank Loans Tendency Survey, Enterprises, Expected	Bank Loans Tendency Survey, Housing, Expected
Bank Loans Tendency Survey, Funding Conditions, Expected	JPM Global Composite PMI
JPM Global Manufacturing PMI	US Corporate High Yield
US Federal Funds Effective Rate	US 10-year Treasury Yield
World Uncertainty Index	Sentix Economic Expectations
Sentix Current Economic Situation	CBOE VIX
CBOE 10-year Treasury VIX	Real short rate
Real long rate	Term spread (long rate—short rate)
Long dollar spread (long rate – US 10-year yield)	Real M2 growth
Real US Federal Funds Rate	Real US 10 year
Real US term spread (10-year yield – FFR)	US credit spread (high yield – 10-year yield)

Table A5.2. Non-Stationary Variables—First and Second Log Difference
(Y: in real terms)

Gross Domestic Product (Y)	Industrial Production
Industrial Production: Mining & Quarrying	Industrial Production: Manufacturing
IP: Intermediate Goods	IP: Durable Consumer Goods
IP: Nondurable Consumer Goods	IP: Energy
IP: Capital Goods	Automobile Production
Truck Production	Const Permits: Buildings
Const Permits: One Dwelling Residential Buildings	Const Permits: 2+ Dwelling Residential Buildings
Const Permits: Residences for Communities	Const Permits: Hotels & Similar Buildings
Const Permits: Office Buildings	Const Permits: Wholesale & Retail Trade Buildings
Const Permits: Traffic & Communication Buildings	Const Permits: Industrial Buildings & Warehouses
Const Permits: Public Entertainment	Const Permits: Other Nonresidential Buildings
Const Permits: Buildings	Const Permits: One Dwelling Residential Buildings
Const Permits: 2+ Dwelling Residential Buildings	Registered Motor Vehicles
Registered Motor Vehicles: Cars	Registered Motor Vehicles: Trucks
Spot Exchange Middle Rate, NY Close: U.S.	CPI Based Real Effective Exchange Rate
JPMorgan Real Broad Effective Exchange Rate Index, PPI Based	PPI Based Real Effective Exchange Rate
Exchange Rate, Selling (TL/Euro)	Exchange Rate, Selling (TL/100 Yen)
Exchange Rate, Selling (TL/Pound)	Exchange Rate, Selling (TL/US$)
Foreign Trade: Total Merchandise Imports, c.i.f.	BOP: Current Account
BOP: Current Acct: Goods, Services & Primary Income	BOP: Current Account: Goods And Services
CB Balance Sheet: Assets	CB Balance Sheet: Liabilities
CBBS: Liabilities: CB Money: Reserve Money: Deps of Banking Sector	CB Bal Sheet: Liabilities: CB Money: Reserve Money: Currency Issued
CBBS: Liabilities: CB Money: Reserve Money: Deposits of Nonbank Sector	CB Balance Sheet: Foreign Liabilities
CB Balance Sheet: Liabilities: Central Bank Money	CB Bal Sheet: Domestic Assets : Treasury: Other
Consumer Loans and Credit Cards	Banking Sector Credit Vol: Deposit Money Banks' Loans
ISE National 100 Stock Price Index (Y)	Labor Force Survey: Total Labor Force
Labor Force Survey: Employment	Labor Force Survey: Unemployment
Total Labor Force	Employment
Unemployment	Labor Force Survey: Population, 15 Years & Over
Labor Force Survey: Nonagricultural Unemployment Rate	Labor Force Survey: Not in Labor Force
CPI: All Items	CPI: Food and Non-alcoholic Beverages
CPI: Alcoholic Beverages and Tobacco	CPI: Clothing and Footwear
CPI: Housing, Water, Electricity, Gas and Other Fuels	CPI: Furniture/Furnishings/Carpets/Other Floor Coverings
CPI: Health	CPI: Transport
CPI: Communication	CPI: Recreation and Culture
CPI: Education	CPI: Restaurants and Hotels
CPI: Miscellaneous Goods and Services	M2
Gross ED, nominal	PPI
EPI	IPI
ToT	Housing Prices (Y)
Manufacturing Shipments	Retail Sales, value
Exports, value	Gross Operating Surplus or Corporate Profits (Y)
Nominal Final Domestic Demand (Y)	Earnings, industry (Y)

Earnings, trade and services (Y)	Earnings, construction (Y)
Exports, capital goods, volume	Exports, intermediate goods, volume
Exports, consumption goods, volume	Imports, capital goods, volume
Imports, intermediate goods, volume	Imports, consumption goods, volume
House sales	Total Tourism Income (Y)
Total Number of Visitors	Retail sales, food, volume
Retail sales, non-food, volume	Retail sales, automotive fuel, volume
Exports: Motor Vehicles and Trailers (Y)	Domestic Taxes on Goods and Services (Y)
Stamp Duties (Y)	Income Taxes (Y)
Government: Compensation of Employees (Y)	Government: Social Security Contributions (Y)
Government: Goods and Services Purchases (Y)	Government: Capital Expenditures (Y)
Industrial Domestic Turnover: Intermediate (Y)	Industrial Domestic Turnover: Durable (Y)
Industrial Domestic Turnover: Nondurable (Y)	Industrial Domestic Turnover: Capital Goods (Y)
Industrial Domestic Turnover: Energy (Y)	Industrial Non-Domestic Turnover: Intermediate (Y)
Industrial Non-Domestic Turnover: Durable (Y)	Industrial Non-Domestic Turnover: Nondurable (Y)
Industrial Non-Domestic Turnover: Capital Goods (Y)	Industrial Non-Domestic Turnover: Energy (Y)
Transportation and Storage Turnover (Y)	Motor Vehicle Sales (Y)
Domestic Cement Sales (Y)	Gross Demand of Electricity (Y)
VAT on Imports (Y)	Total Tax Revenue (Y)
Banking Sector TRY Assets: Non-Performing Loans	Banking Sector FX Assets: Non-Performing Loans
Foreign Liabilities to Residents: FX Deposits of Banking Sector	LT Pvt Loans from Abroad
Banking Sector Credit Vol: Dom Loans: TRY Loans	Banking Sector Credit Vol: Inv & Dev Bks: TRY Dom Loans
Banking Sector Cred Vol: Participation Banks: TRY Dom Loans	Banking Sector Credit Vol: Past Due Loans in TRY
Banking Sec Cred Vol: TRY Past Due Loans for Inv & Dev Bks	Banking Sector Credit Vol: Part Banks Past Due TRY Loans
Banking Sector Credit Vol: Dom Loans: FX Loans	Banking Sector Credit Vol: Inv & Dev Bks: FX Dom Loans
Banking Sector Cred Vol: Participation Banks: FX Dom Loans	Banking Sector Credit Vol: Past Due Loans in FX
Banking Sec Cred Vol: FX Past Due Loans for Inv & Dev Bks	Banking Sector Credit Vol: Part Banks Past Due FX Loans
Banking Sector: Loans [FX]	Banking Sector: FX Indexed Loans [LC]
Banking Sector: Non-Performing Loans [FX]	Banking Sector: Loans [LC]
Banking Sector: FX Indexed Loans [LC]	Banking Sector: Non-Performing Loans [LC]
Consumer Loans and Credit Cards	Banking Sector Credit Vol: Deposit Money Banks' Loans
Consumer Housing Loans	Consumer Automobile Loans
Individual and Corporate Credit Cards	Loans Indexed to FX: Housing Loans
Loans Indexed to FX: Automobile Loans	Individual Credit Cards: YTL
Individual Credit Cards: FX	Deposit Banks TRY Loans: Corporate Credit Cards
Total Bank Loans (Y)	Non-Performing Loans, TRY
Non-Performing Loans, FX	Residents FX deposits in banking sector, USD
Baltic Exchange Dry Index	Composite CPI for Advanced Economies
HWWI Commodity Price Index	Dallas Fed House Price Index World
World Industrial Production ex Construction	CPB World Trade Volume
Dow Jones Global Index World	WTI Weekly Average Price
BIS Narrow NEER Dollar	AUD/JPY Nominal ER
Deposit Banks FX Loans: Corporate Credit Cards	