

# Untitled

Karel R  pp, Henry Enno Turu

29 11 2020

## 1 Methodology

### 1.1 Linear regressions

The first part of our methodology consists of replicating the methodology conducted by Cremers et al. (2017). In other words, we test whether the treasury implied volatility can be used to predict the future macroeconomic real activity.

To quantify the predictability of macroeconomic real activity using the treasury implied volatility, we run an ordinary least squares (OLS) regression. In the first regression, we take the 5-year YIV and use it to predict the forward-looking GDP growth. To specify,  $GDP_{i,t+j}$  refers to logarithmic values of year-on-year quarterly growth rate of the real GDP.  $H$  is equal to the periods predicted - e.g. if  $H=4$ , it means that we are taking the rolling overlapping average of GDP growth over the 4 quarters.

$$\sum_{j=1}^{j=H} \log(1 + GDP_{i,t+j})/H = \alpha_H + \beta_H \sigma_{IV,t} + \varepsilon_{t+H} \quad (1)$$

In the linear models, we compare the predictive ability of YIV within different time periods ( $H=1,2,3,...,12$ ). Furthermore, to validate YIVs predictive ability, we construct different models (including autoregressive models) by adding in various financial and economical control variables to see whether the significance of our main variable persists. The control variables include term spreads, credit spreads, stock market implied volatility (VIX), number of new residential construction starts (HOUSNG). In order to enable the comparison of the variables, we standardize all of the independent variables so the mean is equal to 1 and standard deviation to 0. Furthermore, all of our reported coefficients as well as standard errors are adjusted for heteroskedasticity and also autocorrelation (HAC) - to do so, we use the Newey-West methodology with automatic bandwidth selection process. Nevertheless, it is important to mention that the errors are not calculated manually, instead the R package ‘sandwich’<sup>1</sup> will be used.

$$\sum_{j=1}^{j=H} \log(1 + GDP_{i,t+j})/H = \alpha_H + \beta_H \sigma_{IV,t} + Controls + \varepsilon_{t+H} \quad (2)$$

---

<sup>1</sup>The documentation is available from <https://cran.r-project.org/web/packages/sandwich/sandwich.pdf>

Next, to determine the direction of the causality, we run a Vector Autoregressive model (VAR) Granger Causality test on every variable to make sure that e.g. YIV indeed granger-causes movements in GDP.

We also check whether there exist any asymmetries regarding economic cycles - i.e. if YIV exerts a bigger/smaller effect on GDP when the current state is a recession or an expansion. To do so, we introduce a model with a dummy variable that is equal to 1(0) during a recessionary(expansionary) period. The business cycle dating (i.e. recessionary and expansionary periods) for the US economy is taken from the National Bureau of Economic Research (NBER)<sup>2</sup>.

$$\sum_{j=1}^{j=H} \log(1 + GDP_{i,t+j})/H = \alpha_H + \beta_H \sigma_{IV,t} + Dummy + \varepsilon_{t+H} \quad (3)$$

## 1.2 Out-of-sample validation

After having done the regression with YIV, subsample dummy and controls as independent variables, we proceed with conducting an out-of-sample test to validate the robustness of the model. To evaluate the robustness of the out-of-sample forecast we use the root mean square forecasting error (RMSFE).

$$SFE = \sum_{j=1}^{j=H} (\log(1 + GDP_{i,t+j}) - \log(1 + \widetilde{GDP}_{i,t+j}))^2 \quad (4)$$

$$RMSFE = \sqrt{mean(SFE)}$$

To calculate the full model RMSFE we first have to obtain square forecasting errors (SFE). For that we construct a predictive model with a 5-year rolling estimation window. Thus, we use the previous 20 quarters to predict the next H quarters (note: here H denotes lead values not rolling averages). The reason behind opting for rolling estimation window is that in this way the predicted datapoint is continuously moving; thus the forecast error is not so much dependent on the selected forecast interval (i.e. there would be big differences if one used 85 datapoints to predict 20 datapoints, or 70 datapoints to predict 35 datapoints). From that regression we obtain the predicted values and through incorporating the actual values we can then compute the SFE-s. Having the SFE-s, we take the mean from the values and then take the square root.

## 1.3 Out of sample validation for different economic cycles

In the second part of our research we build upon the research conducted by Cremers et. al. We test whether the model holds when accounting also for the possible business cycle performance asymmetries as suggested by Siliverstovs & Wochner (2020).

More specifically, we compare the root-mean-square-forecasting-error (RMSFE) of the predictive model during recession and expansion with full sample forecast to find out whether

---

<sup>2</sup><https://www.nber.org/research/data/us-business-cycle-expansions-and-contractions>

full sample forecast's RMSFE forecasts are robust during the expansionary and recessionary subsamples (in other words, we check whether a distributional shift occurs during the recessionary and/or expansionary periods).

To calculate the recessionary subsample RMSFE, we first obtain SFE-s through doing the full regression as described before, however, in calculating RMSFE we take only the SFE-s related to recessions (according to NBER classification). Based on those recessionary SFE-s we calculated the RMSFE. Similarly, in expansionary-only RMSFE calculation we excluded those recessionary SFE-s.

$$rRMSFE = \frac{RMSFE_{subset}}{RMSFE_{full}} \quad (5)$$

To offer a better comparison of the models forecasting performance, we assess the sub-models with expansionary & recessionary data points compared to benchmark model by calculating RMSFE. In the case of the formula having a value smaller than 1 indicates that the subset model does have a superior performance over the fullsample model.

### 1.3.1 Cumulated sum of squared error differential

As the recessions in comparison with expansionary periods do have less observations, one might start to doubt the conclusion drawn from such few observations. To tackle this, we use cumulated sum of squared forecast error difference (CSS-FED) proposed by Welch & Goyal (2008). Even though the main use-case for the model is to compare the forecast performance of different models, it has another benefit more applicable to our needs - it helps to dissect the forecast error and hence see whether the (dis)improvement of relative performance is due to actual continuous (dis)improvement or it is dependent on few influential observations. Thus, the CSS-FED is applied in order to test how the few observations of recessions impact the whole forecasting error. As mentioned in the literature review, there is reason to believe their influence on the whole sample forecast error is crucial. In other words, recessionary periods RMSFE is more influential and outweighs the importance of expansionary periods RMSFE. The CS-SFED can be calculated according to the formula below:

$$CSSFED = \sum_{t=1}^T [(e_{hist,t})^2 - (e_{lm,t})^2] \quad (6)$$

where  $e_{hist,t}$  refers to forecast errors of historical mean model, and  $e_{lm,t}$  of the linear model at time  $t$ .

The resulting figure can be plotted over time to visually identify how each observation contributes to the cumulative forecasting error - i.e. if each datapoint increases the forecast error difference, the graph shows an upwards trend. Contrarily, if little difference exists between each observation's error differential, the graph will stay fairly smooth. In the case when one observation's sum of forecasting error difference(SFED) is significantly bigger from the previous ones, big jumps can be identified on the graph - these jumps mark the observations with largest influence on the overall forecasting performance.

## 1.4 Machine learning methods

As Machine Learning (ML) methods successfully tackle nonlinearities that cannot be accounted for in simple linear models, the forecasting gains tend to be the highest during times of high economic uncertainty Coulombe et al. (2020). As our research includes many different economic crises, we expect to find forecasting gains by utilizing the Macroeconomic Random Forest method. In our research, we focus specifically on a (modified) random forest based method due to its advantages over other ML methods, which will be discussed in the following paragraphs.

### 1.4.1 Decision Trees

In order to dig deeper into the methodology of random forest, it is crucial to understand one of its core elements - decision trees. Decision tree is a rather straightforward non-parametric algorithm that can be used for both regressions and classifications. The name ‘decision tree’ derives from the fact that the algorithm is built in a tree-like structure - it recursively splits the whole sample into subsets by following predefined criteria. In our case when the dependent variable is continuous, reducing variance is used as the selection criteria. Thus, the algorithm takes the whole dataset and picks the variables that possess the biggest influence on the dependent variable, splitting the dataset further until stopping criteria is met; so it arrives at the leaf node where the decision is made. The stopping criteria are hyper-parameters defined by the user such as maximum depth, minimum leaf size (no of observations in the leaf node), minimum number of samples, etc (Zhang & Trubey, 2019).

As decision trees are non-parametric, it means that no strong assumptions about the underlying data and its form is made, which in return enables capturing different forms such as non-linearities in the data. However, this makes them subject to overfitting - it might be the case that the algorithm might start capturing random movements (noise) instead of actual meaningful patterns. There are three main ways to tackle this

- 1) Tuning the hyperparameters
- 2) Pruning - growing the full tree and then eliminating decision nodes so that the general accuracy preserves
- 3) Random forest

### 1.4.2 Random Forest

Decision Trees are not a robust method as the results are extremely dependent on the dataset, even a small change in the initial training data can yield different outcomes. This is the very reason why Random Forest was proposed by Breiman (2001). Random forest itself is an ensemble based supervised learning method. As the name suggests there are two main elements behind it.

```
knitr::include_graphics("images/randomforest.png")
```

First is the randomness part - random forest uses bootstrap aggregation (bagging) to construct random samples of the initial dataset. Furthermore, the randomness is also included

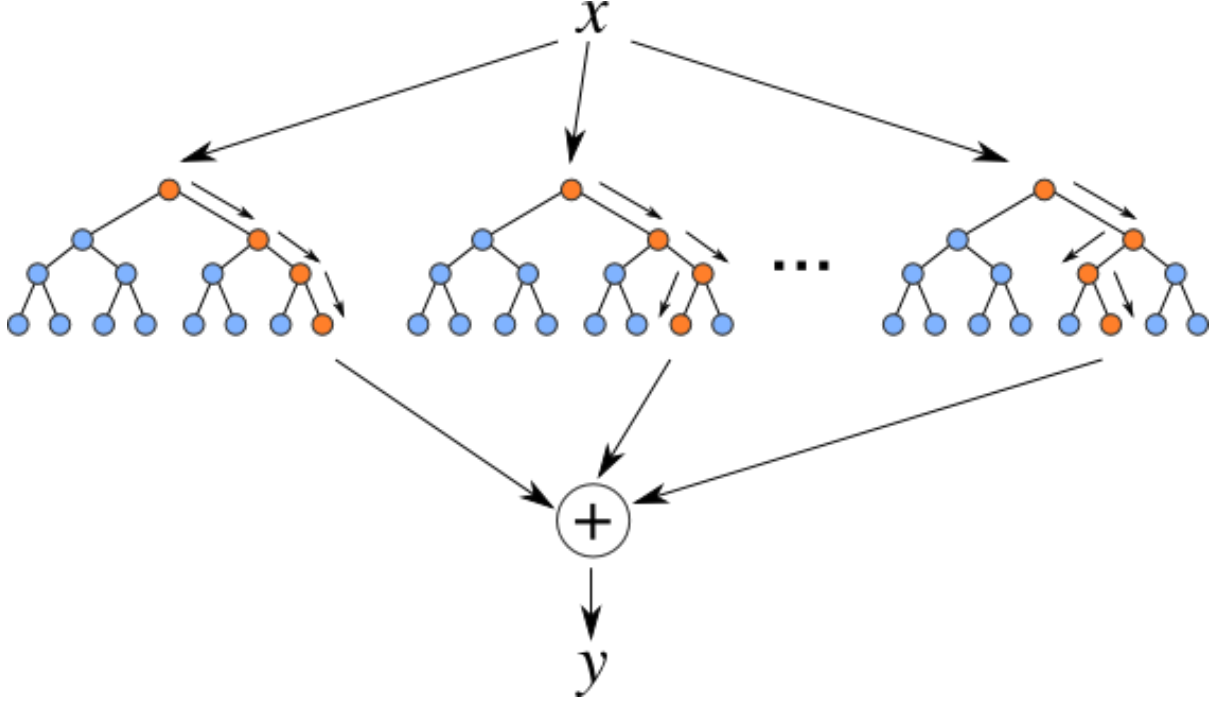


Figure 1.1: Random forest example

in variable selection - it randomly selects variables used for splitting at each node. Secondly, forest, which refers to the aggregation. The conclusion of the final model is reached by aggregating and averaging the output of individual decision trees. These two elements help to tackle overfitting as RF randomly creates a high number of combinations on the basis of which to create splits, which reduces the correlations between samples (Zhang & Trubey, 2019). Hence, the final outcome is much more robust & less likely to be subject to overfitting.

To take this process together in algorithmical terms:

- 1) Through bootstrap aggregation a sample set out of the predefined training data is created.
- 2) Then the model randomly selects  $x$  number variables amongst the total set  $X$ .
- 3) Subsequently, the best variable and splitting criterion is selected, on the basis of which the current node is splitted into two sub-nodes.
- 4) More specifically, the choice is made on the basis of mean squared error (MSE) - MSE is minimized at each split.
- 5) This process is repeated until each terminal node reaches minimum size (by default 5).
- 5) The output is achieved by averaging the estimation of each tree in the model.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \gamma)^2 \quad (7)$$

Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32. <https://doi.org/10.1201/9780429469275-8>

Coulombe, P. G., Leroux, M., Stevanovic, D., & Surprenant, S. (2020). How is machine learning useful for macroeconomic forecasting? *arXiv*. <http://arxiv.org/abs/2008.12477>

Cremers, M., Fleckenstein, M., & Gandhi, P. (2017). Treasury Yield Implied Volatility and Real Activity. *Journal of Financial Economics (JFE)*, *Forthcoming*. <https://doi.org/10.2139/ssrn.3006473>

Siliverstovs, B., & Wochner, D. (2020). State-Dependent Evaluation of Predictive Ability <sup>†</sup>. *Journal of Forecasting*. <https://doi.org/10.1002/for.2715>

Welch, I., & Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, *21*(4), 1455–1508. <https://doi.org/10.1093/rfs/hhm014>

Zhang, Y., & Trubey, P. (2019). Machine Learning and Sampling Scheme: An Empirical Study of Money Laundering Detection. *Computational Economics*, *54*(3), 1043–1063. <https://doi.org/10.1007/s10614-018-9864-z>