User Guide: Inflation Forecasting Scripts

Alessandro Dodon

February 11, 2025

Contents

1 Introduction

This guide provides a structured overview of the inflation forecasting scripts, covering data preprocessing, model implementation, and expected outputs to ensure ease of use and reproducibility. Please note that the scripts must be run sequentially, beginning with data preprocessing, as each forecasting model depends on the prepared dataset. Running a script will overwrite previous variables, allowing models to be tested multiple times in any order. For theoretical explanations, refer to the slides in `Presentation.pdf`.

2 Requirements

The dataset `current.csv` (containing macroeconomic indicators) must be placed in the working directory before execution. Ensure R is installed along with the required packages. Copy and paste the following command in your R console to install necessary packages:

```
install.packages(c("stats", "zoo", "readr", "pracma", "reshape2",
"utils", "viridis", "imputeTS", "forecast", "dplyr", "glmnet",
"randomForest", "vars", "pls", "pheatmap"))
devtools::install_github("cykbennie/fbi")
```

3 Data Preprocessing

`PreProcessing.R` must be run before any forecasting model.

3.1 Operations Performed

- Loads FRED-MD data and applies transformations.

- Handles missing values by cutting the time series short and using a Simple Moving Average (SMA) algorithm.

- Removes the COVID-19 period.

- Removes variables with excessive missing values: `ACOGNO`, `ANDENOx`, `TWEXAFEGSMTHx`, `UMCSENTx`, `VIXCLSx`.

- Computes covariance matrix and eigenvalues for PCA.

- Splits dataset into training and test sets.

- Saves cleaned datasets and key visualizations as PDFs in the same directory as the script:

  - `inflation_plots.pdf`
  - `time_series_original.pdf`
  - `time_series_transformed.pdf`
  - `autocorrelation_original.pdf`
  - `autocorrelation_transformed.pdf`
  - `covariance_matrix_heatmap.pdf`
  - `scree_plot_eigenvalues_covariance_matrix.pdf`
  - `eigenvalue_ratios_covariance_matrix.pdf`

- Prints key statistics, including eigenvalues and summary of missing data handling.

3.2 Modifiable Parameters

- The dataset is split into training and testing sets using the date `split_date <- as.Date("2000-01-01")` . This can be modified by the user.

- The target variable is set to `CPIULFSL` (inflation). Users can replace this code with another variable from the FredMD dataset, such as GDP or industrial production, to forecast different economic indicators. The `fredmd_description` dataset in the preprocessing script can be consulted for available variables. Using `Replace` in R is recommended for this task.

4 Forecasting Models

Each model is manually implemented using custom functions. Predictions are compared against actual test values, and Mean Squared Error (MSE) is printed in the console. The outputs include forecast plots saved as PDFs in the same directory as the script.

4.1 Autoregressive (AR(1)), OLS with All Variables, and Historical Average

`AR(1).R`

- AR(1): Uses past inflation values for forecasting.

- OLS: Utilizes all available predictors for a multivariate approach.

- Both methods manually lag variables and apply OLS using `lm()` instead of automatic packages.

- Historical Average: Creates a simple example to illustrate underfitting.

- Outputs:

  - `actual_vs_predicted_ols.pdf`
  - `actual_vs_predicted_average.pdf`
  - `actual_vs_predicted_ols_all.pdf`
  - MSE printed in the console.

4.2 Lasso and Ridge Regression

`Lasso.R`, `Ridge.R`

- Lasso: Performs feature selection via L1 regularization.

- Ridge: Implements L2 regularization.

- The user can modify the `lambda_values` parameter to adjust the penalty in Lasso and Ridge regression.

- Outputs: `actual_vs_predicted_lasso.pdf`, `actual_vs_predicted_ridge.pdf`, MSE printed in the console.

4.3 Principal Component Regression (PCR)

`PCR.R`

- Reduces dimensionality using PCA before regression.

- The number of principal components can be modified using the `num_pc_list` parameter.

- Outputs: `actual_vs_predicted_pcr.pdf`, MSE printed in the console.

4.4 Vector Autoregression (VAR)

`VAR.R`

- Implements VAR models with and without PCA-based dimensionality reduction.

- Includes additional preprocessing since the VAR package assumes each time series is temporally aligned.

- The number of principal components used in PCA can be modified via the `pc_values` parameter.

- Outputs: `actual_vs_predicted_var_pca.pdf`, `actual_vs_predicted_var_all.pdf`, MSE printed in the console.

4.5 Random Forest (RF)

`RF.R`

- Uses a recursive Random Forest model for time series forecasting.

- The function `perform_rf` applies the Random Forest model with manually set hyperparameters. Users can modify the function or create their own version to test different configurations.

- Outputs: `actual_vs_predicted_rf.pdf`, MSE printed in the console.