A Machine Learning Approach to Inflation Forecasting

Alessandro Dodon

MS Quantitative Finance Student, USI Lugano
BS Economics, Politics and Social Sciences, Unibo Bologna

February 2025

Outline

This presentation covers (very briefly) the following topics:

- Introduction and objective of the study
- The FRED-MD dataset
- Pre-processing for time series
- A simple forecasting setup
- Setting a baseline with AR(1)
- Regularisation methods (Lasso, Ridge)
- Principal Component Regression
- VAR and its challenges with "Big Data"
- Random Forest

Introduction (1/3)

- Objective: Compare methods from econometrics and ML for forecasting, using "Big Data" and addressing the "curse of dimensionality"
- Baseline: The classic AR(1) model serves as the benchmark for comparison
- Focus: Explore whether advanced techniques (Lasso, Ridge, PCR, VAR, Random Forest) can surpass the benchmark
- Outcome: Surpassing AR(1) is challenging, but careful experimentation yields moderate improvements

Inspired By: De Mol, Giannone, Reichlin (2008), *"Forecasting Using a Large Number of Predictors: Is Bayesian Shrinkage a Valid Alternative to Principal Components?"*, *Journal of Econometrics*.

Introduction (2/3)

- This study loosely follows the referenced paper, making simpler assumptions and using simpler techniques (e.g., no Bayesian methods)
- The focus is solely on judging the forecasting accuracy of each method, as done in the paper
- A key downside of ML techniques is that they can act as a "black box" (briefly covered later)
- However new and exciting literature is emerging on causal ML (not covered here)

Introduction (3/3)

- ML is a math-heavy topic, I will focus on the intuition behind each method and their application (this is an introductory discussion)
- Results will be analyzed after each method, with some thoughts on limitations provided at the end
- All my code is in R, and you can find each script in my GitHub repository: InflationForecast (link)
- Some coding issues are not discussed here, but they may be mentioned briefly in the scripts (with `# NOTE:`)

Forecasting & Data Science Terminology

- Model training = Model fitting, parameter estimation
- Training set = In-sample
- Test set = Out-of-sample
- Forecast horizon = Number of periods ahead being predicted
- Forecast origin = From where you do the forecasting from
- Target variable = Dependent variable being forecasted
- Features, inputs = Independent variables, regressors, predictors

Note: In economic research, certain terms are sometimes used loosely, including "Big Data", "curse of dimensionality", "overfitting", etc.

The FRED-MD Dataset

- A gold standard for modern macroeconomics and macroeconometrics
- Contains convenient (US) macroeconomic data from 1960 to today, spanning different sectors
- Uses (also) monthly observations, providing more data
- Offers convenient functions for pre-processing through their R package "fbi"
- Used in De Mol et al.'s paper (older version); this study uses the updated dataset
- Dataset link: FRED-MD Database
- R package link: fbi R Package

Pre-Processing for Time Series (1/2)

- Unlike traditional methods, ML techniques are not designed specifically for time series data, requiring extensive pre-processing
- Stationarity is crucial and can be handled easily using the `fredmd` function from the "fbi" package
- To double-check, I plot each time series before and after and their autocorrelation
- Other essential steps include handling missing values and treating outliers

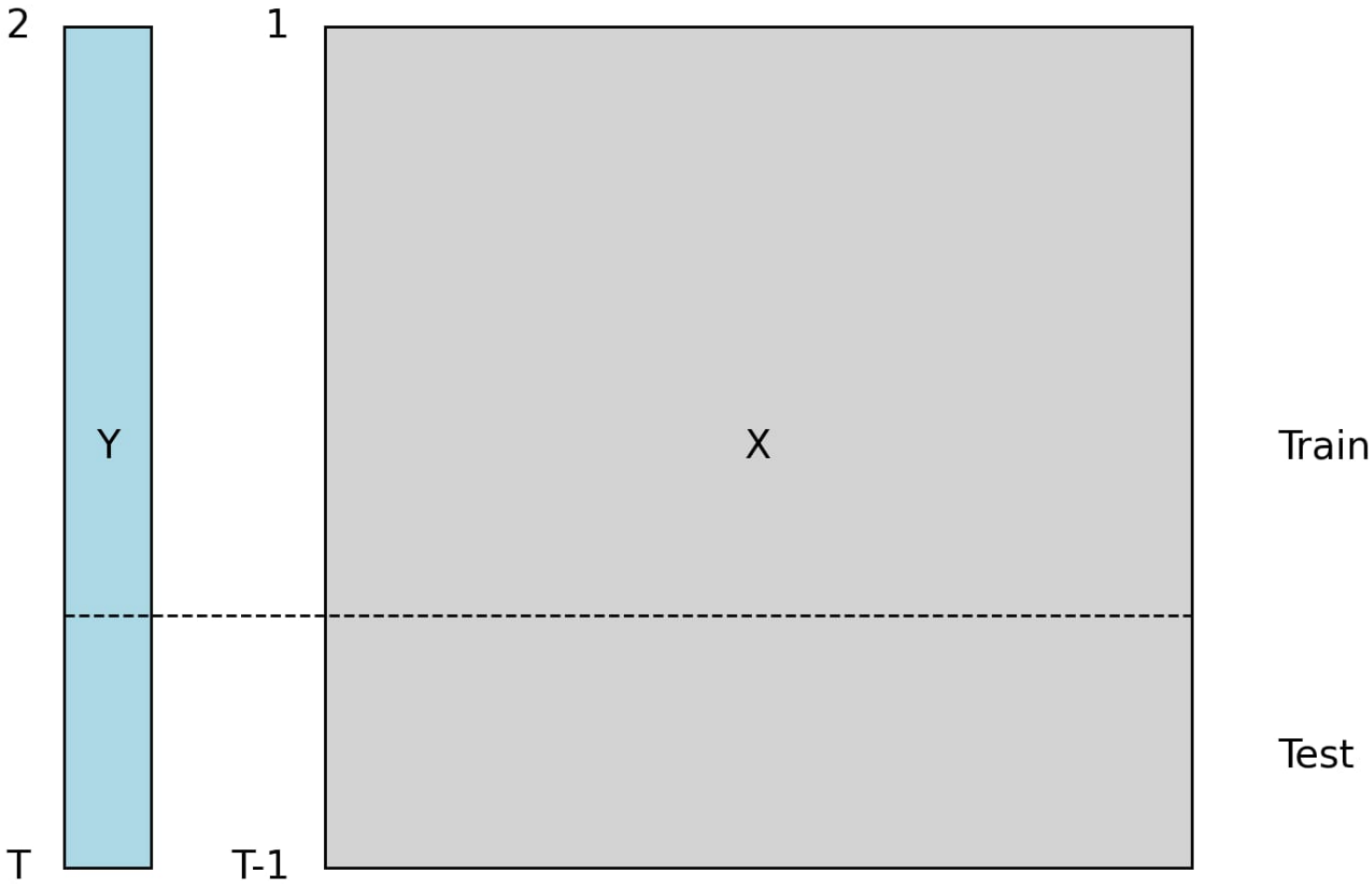
Pre-Processing for Time Series (2/2)

- I handle missing values and outliers manually in R
- The time series is shortened to cover the period from January 1960 to December 2019
- Time series with too many NA values are removed, leaving 121 variables
- I use a SMA algorithm for the remaining NA values
- For outliers, I exclude the COVID-19 period by cutting the data short
- This approach is very approximative and may influence the results
- E.g., financial crises periods may be problematic

A Simple Forecasting Setup (1/2)

- The data is manually lagged, with Y_t as the dependent variable and X_{t-1} as the predictor
- The dataset is split into $\approx 70\%$ training and 30% testing
- The (training) data is standardized each time
- The model is trained at each iteration
- Predictions are made recursively, using X_{test} as input to generate one-month-ahead predictions (no bias since X_{t-1} is in the same information set)
- After each prediction, the test data is added to the training set to update the (recursive) model

A Simple Forecasting Setup (2/2)

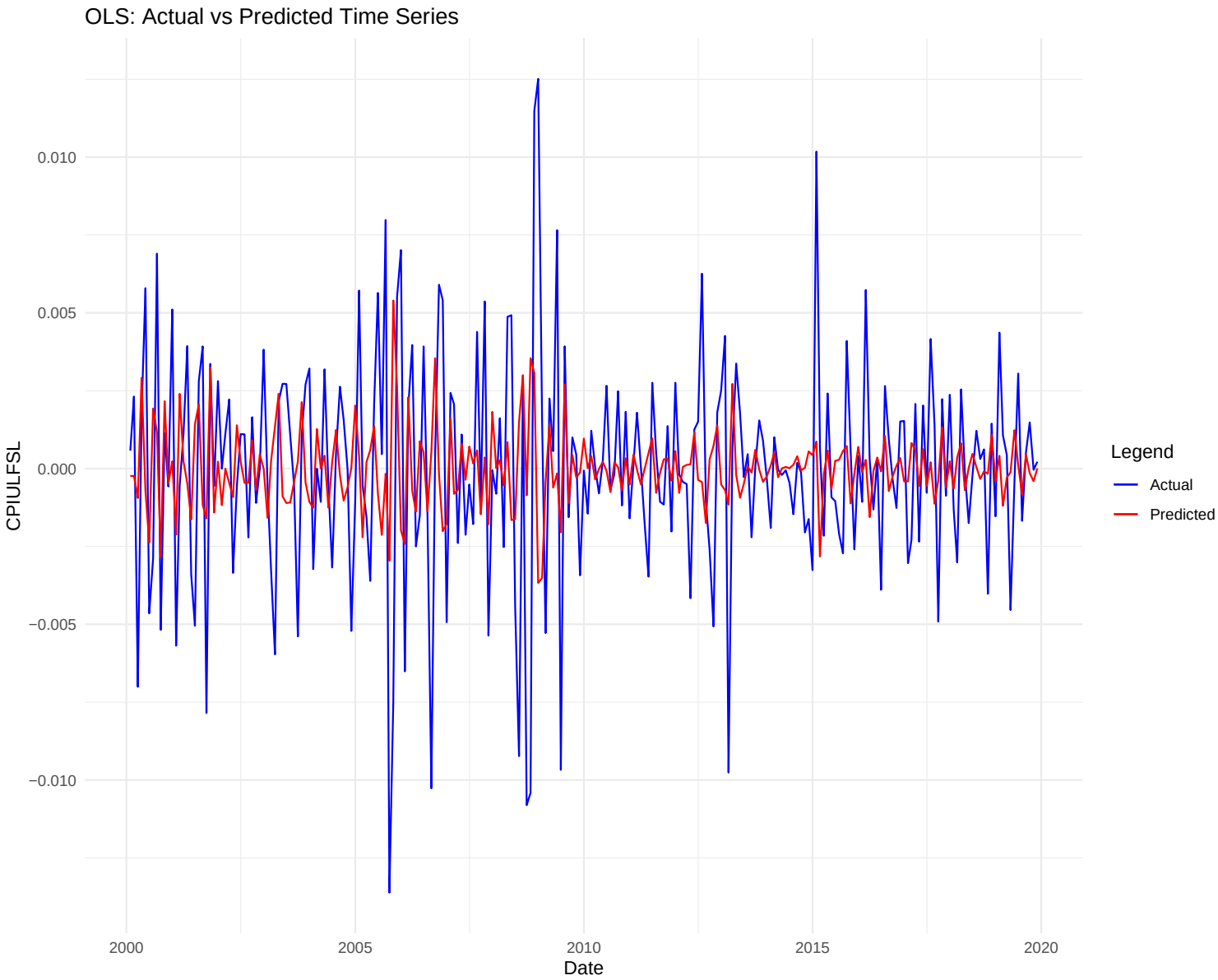


First month of Y and last month of X are excluded

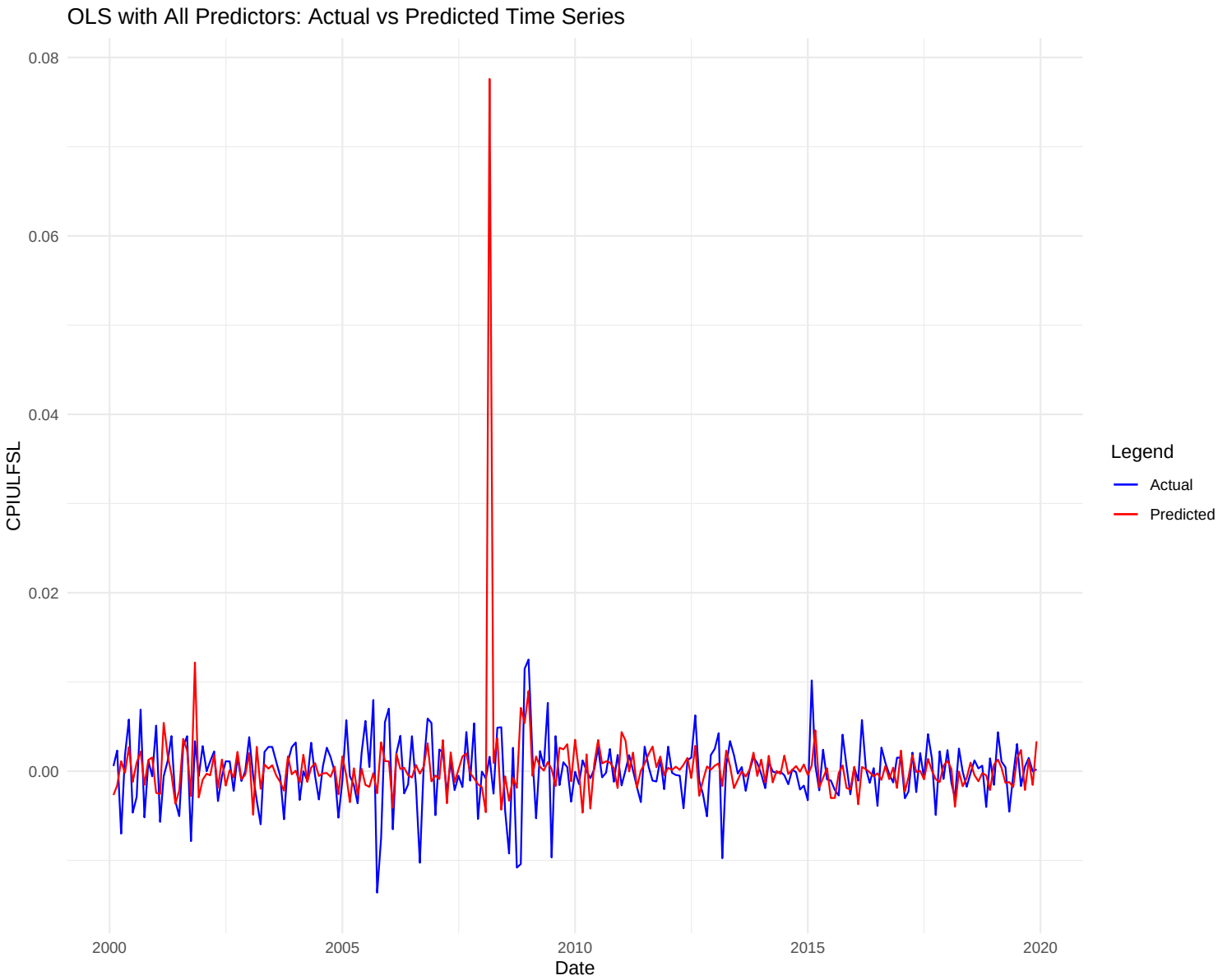
Setting a Baseline with AR(1)

- AR(1) is the simplest model and surprisingly hard to beat
- It requires little to no computational power
- MSE and a plot of actual vs. forecasted values are used to compare models
- This provides an effective baseline to analyze trade-offs (approximatory, as this is an exploratory study)
- E.g., while an advanced neural network may outperform AR(1), it is not guaranteed
- Neural networks also require significantly more time and computational power

AR(1) Results (MSE: 1.351304e-05)



"Overfitting" Example (OLS with All Regressors)



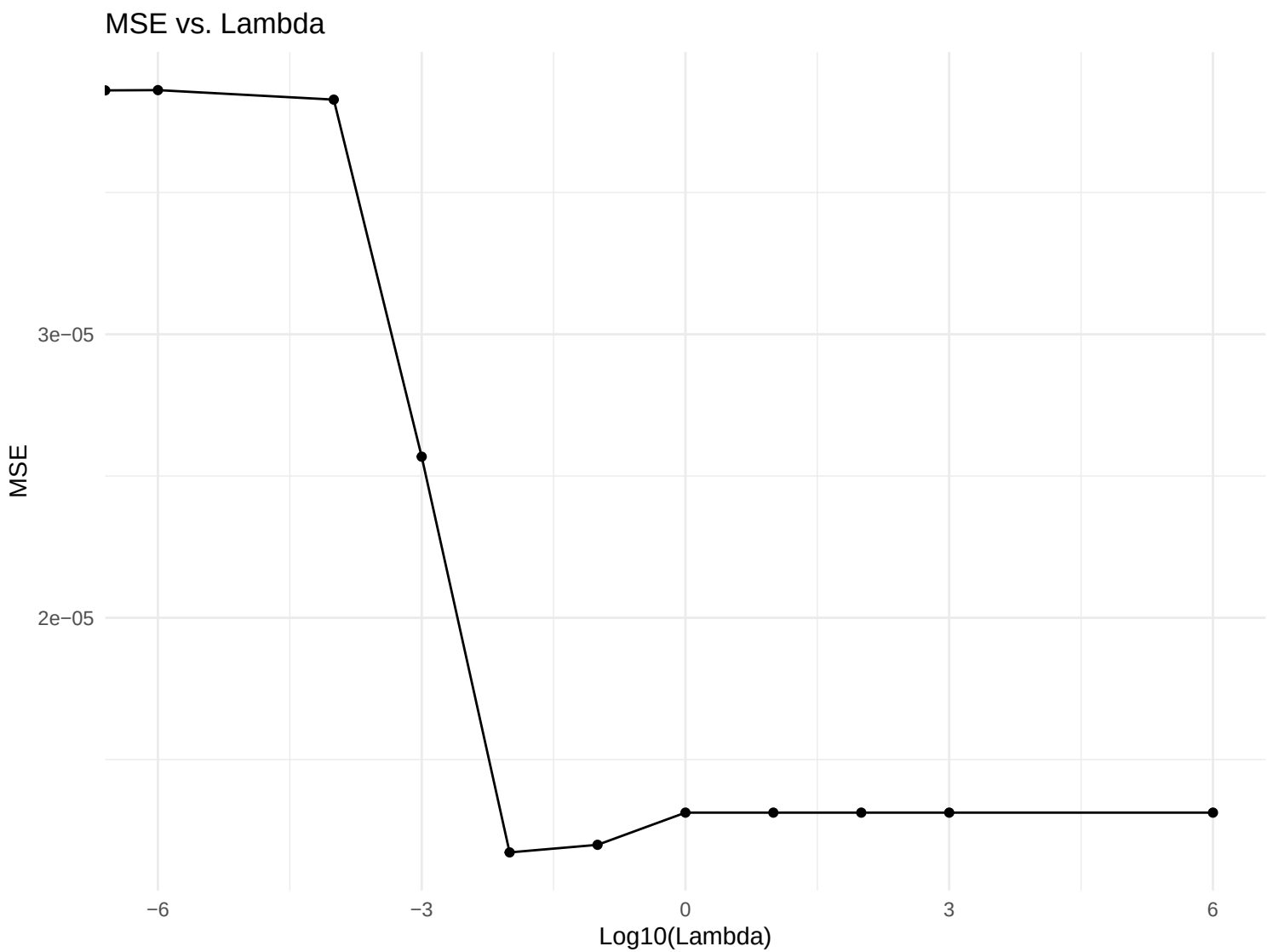
Regularisation Methods (Lasso)

- Lasso minimizes the following objective function:

$$\min_{\beta} \left(\sum_{t=1}^{n} (Y_t - X_t\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right)$$

- Can set some coefficients exactly to zero, effectively performing variable selection
- Requires an appropriate penalty parameter λ, typically selected through cross-validation
- In time series, cross-validation is particularly challenging
- I perform a "grid search" to find the optimal λ (similar to the approach used in the paper)

Lasso Results (Best MSE = 1.172451e-05)



Regularisation Methods (Ridge)

- Ridge minimizes the following objective function:

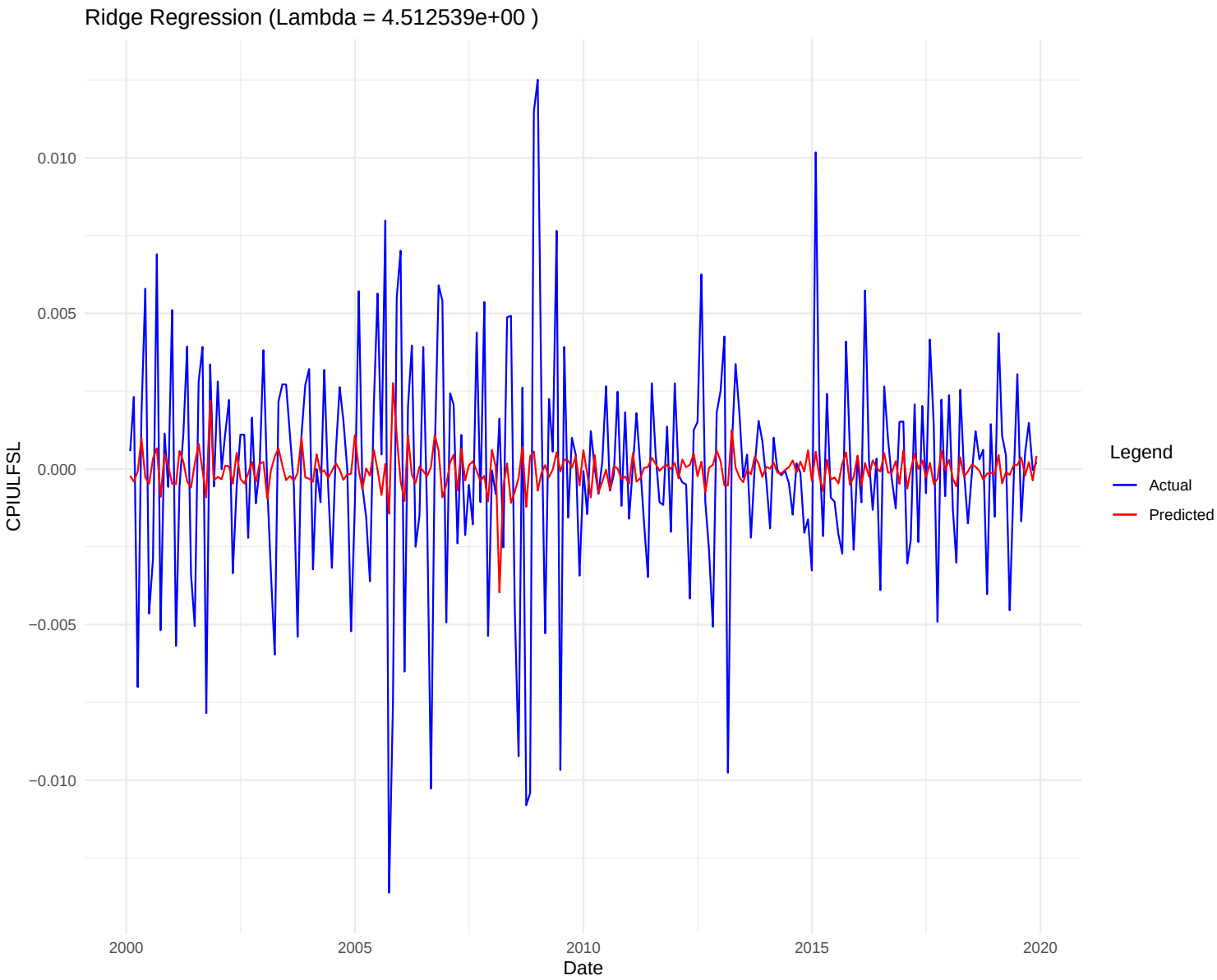
$$\min_{\beta} \left(\sum_{t=1}^{n} (Y_t - X_t\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

- Similar to Lasso but shrinks coefficients towards zero instead of setting them exactly to zero
- The optimal λ is (roughly) suggested by the paper as:

$$\lambda \approx \frac{P}{\sqrt{T}}$$

- As $\lambda \to \infty$, heavy penalization reduces coefficients, effectively fitting an intercept ("underfitting")
- As $\lambda \to 0$, it resembles OLS with all regressors ("overfitting")

Ridge Results (Optimal λ MSE = 1.257817e-05)



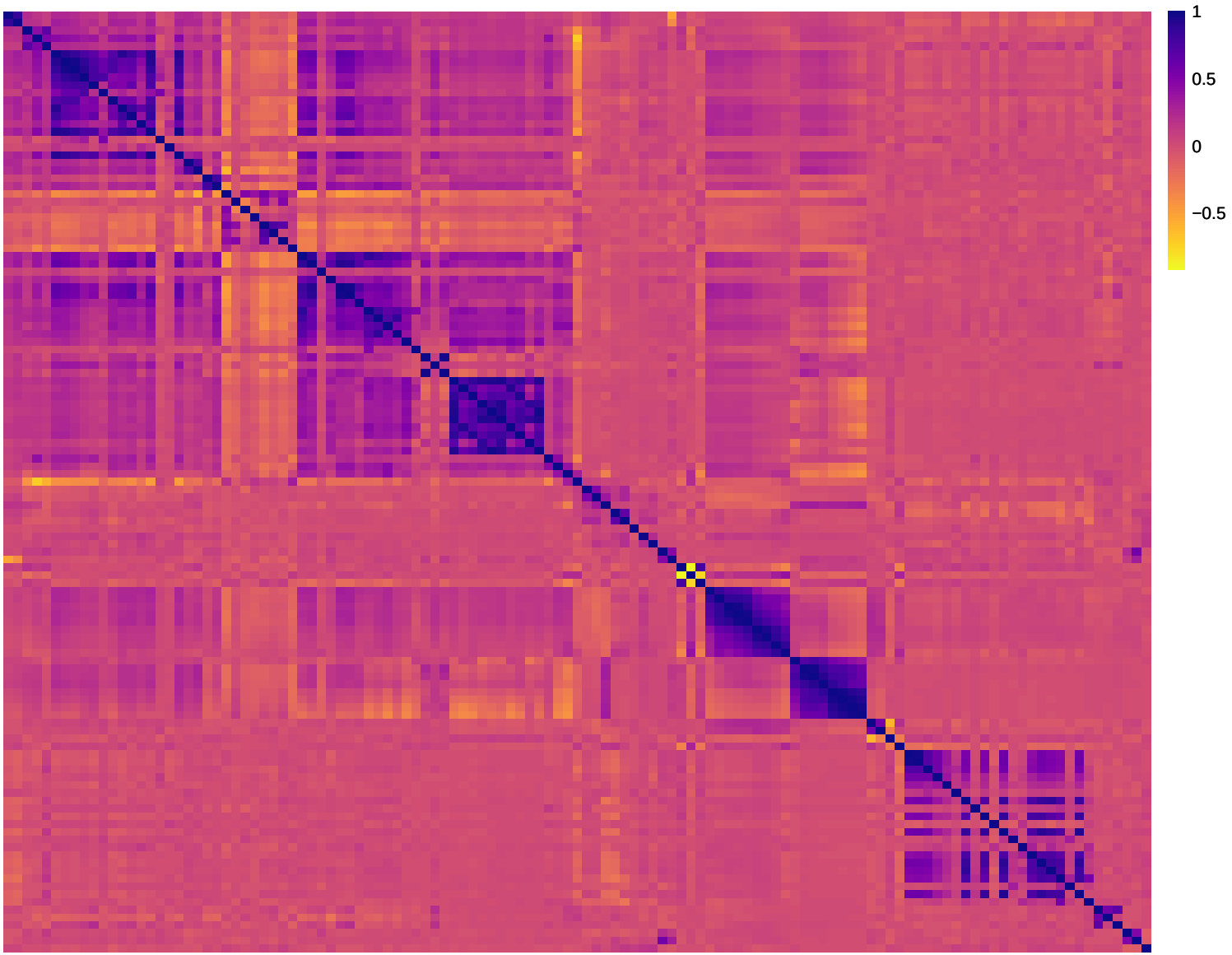
Ridge Results (Rounded Values)

Lambda Value	MSE
1e-06	3.861e-05
1e-04	3.847e-05
1e-03	3.538e-05
1e-02	2.819e-05
1e-01	1.596e-05
0e+00	3.860e-05
1e+00	1.186e-05
1e+01	1.280e-05
1e+02	1.308e-05
1e+03	1.312e-05
1e+06	1.313e-05
Optimal Guess 1	1.264e-05
Optimal Guess 2	1.258e-05

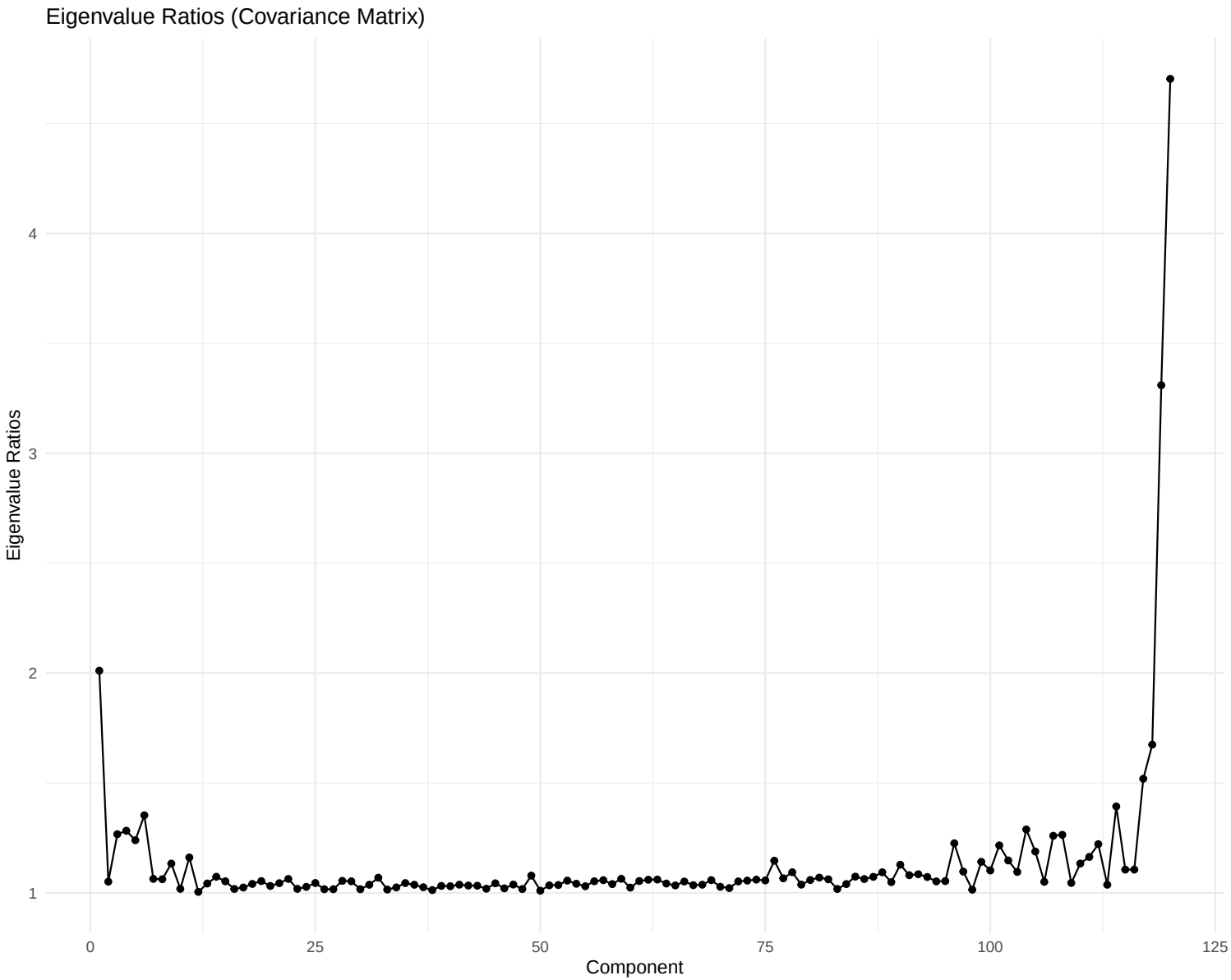
Principal Component Regression (PCR)

- PCA identifies directions (principal components) that maximize variance in the data
- A classic technique for dimensionality reduction, heavily simplifying calculations
- Principal components are (here) not interpretable, making PCR "black box"
- PCR is simply linear regression with the principal components (PCs) as regressors
- I use both an empirical approach (trying different PCs) and a formal one (eigenvalue analysis from the covariance matrix)

Covariance Matrix Heatmap of FRED-MD



Eigenvalue Ratios for Component Selection



PCR Results (Rounded Values)

Number of Components	MSE
1	1.323e-05
2	1.311e-05
3	1.340e-05
5	1.351e-05
10	1.307e-05
25	2.002e-05
50	1.219e-05
75	5.756e-05
121 (all regressors)	3.704e-05

The sweet spot for dimensionality reduction lies around 10 or 50 PC's

VAR and its Challenges with "Big Data"

- VAR is simply the multivariate extension of AR(1):

$$Y_t = A_1 Y_{t-1} + \cdots + A_p Y_{t-p} + \epsilon_t$$

- Variables must be aligned and not lagged (pre-processing is crucial)
- This application focuses on forecasting accuracy, though VAR is versatile
- VAR struggles with many variables due to overparameterization
- E.g., Bayesian VAR is popular as it introduces shrinkage to address this issue
- I experiment with a VAR using PCA on X to forecast Y, which is effective (but not interpretable)

VAR Results (Rounded Values)

Number of Components	MSE
2	1.380e-05
3	1.334e-05
5	1.332e-05
10	1.299e-05
25	1.253e-05
50	1.193e-05
75	1.245e-05
No PCA (all variables)	3.528e-05

Like PCR, more components help until overfitting occurs, as in OLS

Random Forest

- Random Forest (RF) averages predictions from multiple decision trees to reduce variance:

$$\hat{y} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_m$$

- Unlike previous models, RF is non-linear and can capture complex patterns (also scale-invariant)
- Regularisation involves experimenting with the number of trees (M) and maximum nodes per tree
- More trees reduce variance, improving stability but increasing computation time
- Larger trees (more nodes) capture finer details but risk overfitting

Random Forest Results (Rounded Values)

Number of Trees	Max Depth	MSE
10	5	1.304e-05
25	8	1.293e-05
50	10	1.285e-05
75	12	1.256e-05
100	15	1.267e-05
150	18	1.266e-05
200	20	1.245e-05
500	25	1.262e-05

Noticeable improvement with 75 trees, gains stall around 200 trees

Conclusions (1/2)

- ML techniques can yield moderate improvements in forecasting performance with proper regularisation (5–10% on average)
- Basic models, like AR(1), are still surprisingly effective
- The trade-off is clear: ML is effective but requires significantly more time, skills, and computational power
- A major downside is their "black box" nature (uninterpretable), limiting practical use (?)

Conclusions (2/2)

- Another limitation is that macroeconomic data is not truly "Big Data"
- Interesting observation: There is growing literature on this topic in economics, even from as early as 2008!
- Personal consideration: Quantitative finance and business analytics deal with truly massive datasets (where ML may excel)
- However, for forecasting, the Efficient Market Hypothesis (EMH) works against you in finance
- And most business data is private!

Potential Future Development

- Bayesian methods are very interesting and effective (as demonstrated in the paper)
- There are many more advanced ML techniques to explore (also specific to time series), but they raise questions
- E.g., can we justify their use? Do we have enough data to avoid overfitting?

References

- Hamilton, J. D. (1994), "Time Series Analysis", Princeton University Press
- James, G., Witten, D., Hastie, T., Tibshirani, R. (2023), "An Introduction to Statistical Learning with Applications in R" (2nd ed.), Springer
- Mullainathan, S., Spiess, J. (2017), "Machine Learning: An Applied Econometric Approach", Journal of Economic Perspectives
- De Mol, C., Giannone, D., Reichlin, L. (2008), "Forecasting Using a Large Number of Predictors: Is Bayesian Regression a Valid Alternative to Principal Components?", Journal of Econometrics
- Stock, J. H., Watson, M. W. (2004), "An Empirical Comparison of Methods for Forecasting Using Many Predictors", Unpublished Manuscript