Forecasting

Forecasting without a model

Forecasting without a model: Plan

- Converting time series into cross-sectional data

Forecasting without a model: Plan

- Converting time series into cross-sectional data
- Add lags to the y_t variable

Forecasting without a model: Plan

- Converting time series into cross-sectional data
- Add lags to the y_t variable
- Use aggregation and sliding or growing windows

Forecasting

Adding predictors

There are algorithms that, based on the training sample of the dependent variable y, learning matrix of predictors X, and new predictors X_F build a forecast $\hat{y}_F$

Forecasting

Adding predictors

There are algorithms that, based on the training sample of the dependent variable y, learning matrix of predictors X, and new predictors X_F build a forecast $\hat{y}_F$

Random Forest, gradient boosting…

Forecasting

Adding predictors

There are algorithms that, based on the training sample of the dependent variable y, learning matrix of predictors X, and new predictors X_F build a forecast $\hat{y}_F$

Random Forest, gradient boosting… and even linear regression!

Forecasting

Adding predictors

There are algorithms that, based on the training sample of the dependent variable y, learning matrix of predictors X, and new predictors X_F build a forecast $\hat{y}_F$

Random Forest, gradient boosting… and even linear regression!

You can average ARIMA/ETS forecasts and forecasts from other algorithms

How to create predictors?

From one column y you can create an entire matrix of predictors X!

- Use lags y_{t-k}

How to create predictors?

From one column y you can create an entire matrix of predictors X!

- Use lags y_{t-k}
- Use functions of lags as predictors

Using y lags

For example, let's take two lags, Ly_t and L^2y_t

Using y lags

For example, let's take two lags, Ly_t and L^2y_t

Training sample:

$$\begin{pmatrix} y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_T \end{pmatrix} \quad \begin{pmatrix} y_1 & y_2 \\ y_2 & y_3 \\ y_3 & y_4 \\ \vdots & \vdots \\ y_{T-2} & y_{T-1} \end{pmatrix}$$

Using y lags

For example, let's take two lags, Ly_t and L^2y_t

Training sample:

$$\begin{pmatrix} y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_T \end{pmatrix} \quad \begin{pmatrix} y_1 & y_2 \\ y_2 & y_3 \\ y_3 & y_4 \\ \vdots & \vdots \\ y_{T-2} & y_{T-1} \end{pmatrix}$$

Sample for prediction:

$$\begin{pmatrix} ? \end{pmatrix} \quad \begin{pmatrix} y_{T-1} & y_T \end{pmatrix}$$

How many lags to add?

- Each added lag reduces the training sample!

How many lags to add?

- Each added lag reduces the training sample!
- It is reasonable to add closest lags Ly_t, L^2y_t

How many lags to add?

- Each added lag reduces the training sample!
- It is reasonable to add closest lags Ly_t, $L^2 y_t$
- For seasonal data it is reasonable to add a seasonal lag $L^{12} y_t$

How many lags to add?

- Each added lag reduces the training sample!
- It is reasonable to add closest lags Ly_t, L^2y_t
- For seasonal data it is reasonable to add a seasonal lag $L^{12}y_t$
- Some algorithms are sensitive to extra predictors (e.g. regression)

How many lags to add?

- Each added lag reduces the training sample!
- It is reasonable to add closest lags Ly_t, L^2y_t
- For seasonal data it is reasonable to add a seasonal lag $L^{12}y_t$
- Some algorithms are sensitive to extra predictors (e.g. regression)
- Some algorithms are insensitive to extra predictors (e.g. a random forest)

Functions of lags

When predicting y_t we can use any function of previous y_{t-1}, y_{t-2}, $\ldots$lags

Functions of lags

When predicting y_t we can use any function of previous y_{t-1}, y_{t-2}, $\ldots$lags

For example:

- $\Delta y_{t-1} = y_{t-1} - y_{t-2}$;

Functions of lags

When predicting y_t we can use any function of previous y_{t-1}, y_{t-2}, …lags

For example:

- $\Delta y_{t-1} = y_{t-1} - y_{t-2}$;
- $\max\{y_{t-1}, y_{t-2}, y_{t-3}\}$;

Functions of lags

When predicting y_t we can use any function of previous y_{t-1}, y_{t-2}, ...lags

For example:

- $\Delta y_{t-1} = y_{t-1} - y_{t-2}$;
- $\max\{y_{t-1}, y_{t-2}, y_{t-3}\}$;
- $\min\{y_{t-1}, y_{t-2}, \dots, y_1\}$.

Typical Predictor

- Aggregate function:
  Min, Max, Mean, Median, Range, Sample Variance, Sample Standard Deviation, . . .

Typical Predictor

- Aggregate function:
  Min, Max, Mean, Median, Range, Sample Variance, Sample Standard Deviation, $\ldots$
- Sliding Window: The aggregate function can be applied to, say, the previous three values $y_{t-1}, y_{t-2}, y_{t-3}$.

Typical Predictor

- Aggregate function:
  Min, Max, Mean, Median, Range, Sample Variance, Sample Standard Deviation, $\ldots$
- Sliding Window: The aggregate function can be applied to, say, the previous three values y_{t-1}, y_{t-2}, y_{t-3}.
- Growing Window: The aggregate function can be applied to all previous values y_{t-1}, y_{t-2}, $\ldots$, y_1.

Using y lag functions

For example, let's take the maximum as a sliding window and the minimum as a growing window

Using y lag functions

For example, let's take the maximum as a sliding window and the minimum as a growing window

Training sample:

$$\begin{pmatrix} y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_T \end{pmatrix} \qquad \begin{pmatrix} \max\{y_1, y_2\} & \min\{y_1, y_2\} \\ \max\{y_2, y_3\} & \min\{y_1, y_2, y_3\} \\ \max\{y_3, y_4\} & \min\{y_1, \ldots, y_4\} \\ \vdots & \vdots \\ \max\{y_{T-2}, y_{T-1}\} & \min\{y_1, \ldots, y_{T-1}\} \end{pmatrix}$$

Using y lag functions

For example, let's take the maximum as a sliding window and the minimum as a growing window

Training sample:

$$
\begin{pmatrix} y_3 \\ y_4 \\ y_5 \\ \vdots \\ y_T \end{pmatrix}
\qquad
\begin{pmatrix}
\max\{y_1, y_2\} & \min\{y_1, y_2\} \\
\max\{y_2, y_3\} & \min\{y_1, y_2, y_3\} \\
\max\{y_3, y_4\} & \min\{y_1, \ldots, y_4\} \\
\vdots & \vdots \\
\max\{y_{T-2}, y_{T-1}\} & \min\{y_1, \ldots, y_{T-1}\}
\end{pmatrix}
$$

Sample for prediction:

$$
\begin{pmatrix} ? \end{pmatrix}
\qquad
\begin{pmatrix} \max\{y_{T-1}, y_T\} & \min\{y_1, \ldots, y_T\} \end{pmatrix}
$$

Forecasting without a model: Summary

- Remember about random forest, gradient boosting and even regular regression

Forecasting without a model: Summary

- Remember about random forest, gradient boosting and even regular regression
- Add dependent variable lags

Forecasting without a model: Summary

- Remember about random forest, gradient boosting and even regular regression
- Add dependent variable lags
- Add aggregation functions as a sliding and growing window

More predictors!

More predictors: Plan

- Trend predictors

More predictors: Plan

- Trend predictors
- Seasonal and holiday dummy

More predictors: Plan

- Trend predictors
- Seasonal and holiday dummy
- Cosines and sines as predictors

Let's use the time!

Let's take t and $\sqrt{t}$ as an example

Let's use the time!

Let's take t and $\sqrt{t}$ as an example

Training sample:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_T \end{pmatrix} \qquad \begin{pmatrix} 1 & \sqrt{1} \\ 2 & \sqrt{2} \\ 3 & \sqrt{3} \\ \vdots & \vdots \\ T & \sqrt{T} \end{pmatrix}$$

Let's use the time!

Let's take t and $\sqrt{t}$ as an example

Training sample:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_T \end{pmatrix} \quad \begin{pmatrix} 1 & \sqrt{1} \\ 2 & \sqrt{2} \\ 3 & \sqrt{3} \\ \vdots & \vdots \\ T & \sqrt{T} \end{pmatrix}$$

Sample for prediction:

$$\begin{pmatrix} ? \end{pmatrix} \quad \begin{pmatrix} T+1 & \sqrt{T+1} \end{pmatrix}$$

Monotonic transformations of time

- You can always try!

Monotonic transformations of time

- You can always try!
- For algorithms based on decision trees (random forest, gradient boosting) additional monotonic time transformations are useless

Monotonic transformations of time

- You can always try!
- For algorithms based on decision trees (random forest, gradient boosting) additional monotonic time transformations are useless
- Be aware of the possible transformation of the original variable (logarithm, Box-Cox transformation)

Seasonal and holiday dummy

If there are not many seasons, then it is reasonable to include a dummy for each season

Seasonal and holiday dummy

If there are not many seasons, then it is reasonable to include a dummy for each season

Training sample for quarterly data:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ \vdots \\ y_T \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

The dummy variable trap

In regression, be aware of the dummy variable trap!

The dummy variable trap

In regression, be aware of the dummy variable trap!

- Either use a dummy for every season and a model without a constant,

The dummy variable trap

In regression, be aware of the dummy variable trap!

- Either use a dummy for every season and a model without a constant,
- or use a dummy for all seasons except one and a model with a constant

The dummy variable trap

In regression, be aware of the dummy variable trap!

- Either use a dummy for every season and a model without a constant,
- or use a dummy for all seasons except one and a model with a constant

The dummy variable trap

In regression, be aware of the dummy variable trap!

- Either use a dummy for every season and a model without a constant,
- or use a dummy for all seasons except one and a model with a constant

Algorithms based on decision trees (random forest, gradient boosting) are resistant to the dummy variable trap

Why do we need sines and cosines?

Adding all dummy variables works poorly if you need a lot of them.

Why do we need sines and cosines?

Adding all dummy variables works poorly if you need a lot of them.

It is hardly worth adding 365 dummy variables for daily data.

Why do we need sines and cosines?

Adding all dummy variables works poorly if you need a lot of them.

It is hardly worth adding 365 dummy variables for daily data.

Sine and cosine will help to decrease the number of predictors!

Why do we need sines and cosines?

Adding all dummy variables works poorly if you need a lot of them.

It is hardly worth adding 365 dummy variables for daily data.

Sine and cosine will help to decrease the number of predictors!

Two facts:

- The period of $\sin t$ and $\cos t$ is 2π

Why do we need sines and cosines?

Adding all dummy variables works poorly if you need a lot of them.

It is hardly worth adding 365 dummy variables for daily data.

Sine and cosine will help to decrease the number of predictors!

Two facts:

- The period of $\sin t$ and $\cos t$ is 2π
- Multiplying the argument by a reduces the period by a factor of a

Fourier expansion

Theorem

Any continuous and differentiable function f with period 2π can be represented as

$$f(t) = c + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

Fourier expansion

Theorem

Any continuous and differentiable function f with period 2π can be represented as

$$f(t) = c + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

Practical recipe for daily data:

- Add predictors $\cos\left(\frac{2\pi}{365} \cdot t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot t\right)$

Fourier expansion

Theorem

Any continuous and differentiable function f with period 2π can be represented as

$$f(t) = c + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

Practical recipe for daily data:

- Add predictors $\cos\left(\frac{2\pi}{365} \cdot t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot t\right)$
- Add predictors $\cos\left(\frac{2\pi}{365} \cdot 2t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot 2t\right)$

Fourier expansion

Theorem

Any continuous and differentiable function f with period 2π can be represented as

$$f(t) = c + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

Practical recipe for daily data:

- Add predictors $\cos\left(\frac{2\pi}{365} \cdot t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot t\right)$
- Add predictors $\cos\left(\frac{2\pi}{365} \cdot 2t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot 2t\right)$
- Add predictors $\cos\left(\frac{2\pi}{365} \cdot 3t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot 3t\right)$

Fourier expansion

Theorem

Any continuous and differentiable function f with period 2π can be represented as

$$f(t) = c + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

Practical recipe for daily data:

- Add predictors $\cos\left(\frac{2\pi}{365} \cdot t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot t\right)$
- Add predictors $\cos\left(\frac{2\pi}{365} \cdot 2t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot 2t\right)$
- Add predictors $\cos\left(\frac{2\pi}{365} \cdot 3t\right)$ and $\sin\left(\frac{2\pi}{365} \cdot 3t\right)$
- ...

More predictors: Summary

- Use time as a predictor

More predictors: Summary

- Use time as a predictor
- Seasonality in predictors can be reflected using dummy variables or using cosine and sine functions

Predictors and $ARIMA$

Predictors and $ARIMA$: Plan

- Regression with $ARMA$ errors

Predictors and $ARIMA$: Plan

- Regression with $ARMA$ errors
- $ARMAX$ model

Predictors and $ARIMA$: Plan

- Regression with $ARMA$ errors
- $ARMAX$ model
- $ARDL$ model

Regression with $ARMA$ errors

Equation

$$y_t = \beta_1 + \beta_2 a_t + \beta_3 b_t + \varepsilon_t,$$

where a_t and b_t are predictors

Regression with $ARMA$ errors

Equation

$$y_t = \beta_1 + \beta_2 a_t + \beta_3 b_t + \varepsilon_t,$$

where a_t and b_t are predictors

- The series (y_t), (a_t), (b_t), (ε_t) stationary

Regression with $ARMA$ errors

Equation

$$y_t = \beta_1 + \beta_2 a_t + \beta_3 b_t + \varepsilon_t,$$

where a_t and b_t are predictors

- The series (y_t), (a_t), (b_t), (ε_t) stationary
- $\varepsilon_t \sim ARMA(p, q)$ with respect to white noise (u_t)

Regression with $ARMA$ errors

Equation

$$y_t = \beta_1 + \beta_2 a_t + \beta_3 b_t + \varepsilon_t,$$

where a_t and b_t are predictors

- The series (y_t), (a_t), (b_t), (ε_t) stationary
- $\varepsilon_t \sim ARMA(p,q)$ with respect to white noise (u_t)
- $\mathbb{E}(\varepsilon_t \mid a_{t-1}, b_{t-1}, a_{t-2}, b_{t-2}, \ldots) = 0$

Regression with $ARMA$ errors

Equation

$$y_t = \beta_1 + \beta_2 a_t + \beta_3 b_t + \varepsilon_t,$$

where a_t and b_t are predictors

- The series (y_t), (a_t), (b_t), (ε_t) stationary
- $\varepsilon_t \sim ARMA(p,q)$ with respect to white noise (u_t)
- $\mathbb{E}(\varepsilon_t \mid a_{t-1}, b_{t-1}, a_{t-2}, b_{t-2}, \ldots) = 0$
- Fourth moments of predictors are finite

$ARMAX$ model

Equation

$$y_t = c + \gamma_1 a_t + \gamma_2 b_t + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$

$$+ u_t + \alpha_1 u_{t-1} + \ldots + \alpha_q u_{t-q},$$

where a_t and b_t are predictors and (u_t) is white noise

$ARMAX$ model

Equation

$$y_t = c + \gamma_1 a_t + \gamma_2 b_t + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$
$$+ u_t + \alpha_1 u_{t-1} + \ldots + \alpha_q u_{t-q},$$

where a_t and b_t are predictors and (u_t) is white noise

- Series (y_t), (a_t), (b_t) stationary

$ARMAX$ model

Equation

$$y_t = c + \gamma_1 a_t + \gamma_2 b_t + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$
$$+ u_t + \alpha_1 u_{t-1} + \ldots + \alpha_q u_{t-q},$$

where a_t and b_t are predictors and (u_t) is white noise

- Series (y_t), (a_t), (b_t) stationary
- $\mathbb{E}(u_t \mid a_{t-1}, b_{t-1}, y_{t-1}, a_{t-2}, b_{t-2}, y_{t-2}, \ldots) = 0$

$ARMAX$ model

Equation

$$y_t = c + \gamma_1 a_t + \gamma_2 b_t + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$
$$+ u_t + \alpha_1 u_{t-1} + \ldots + \alpha_q u_{t-q},$$

where a_t and b_t are predictors and (u_t) is white noise

- Series (y_t), (a_t), (b_t) stationary
- $\mathbb{E}(u_t \mid a_{t-1}, b_{t-1}, y_{t-1}, a_{t-2}, b_{t-2}, y_{t-2}, \ldots) = 0$
- The fourth predictor moments are finite

$ARMAX$ model

Equation

$$y_t = c + \gamma_1 a_t + \gamma_2 b_t + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$

$$+ u_t + \alpha_1 u_{t-1} + \ldots + \alpha_q u_{t-q},$$

where a_t and b_t are predictors and (u_t) is white noise

- Series (y_t), (a_t), (b_t) stationary
- $\mathbb{E}(u_t \mid a_{t-1}, b_{t-1}, y_{t-1}, a_{t-2}, b_{t-2}, y_{t-2}, \ldots) = 0$
- The fourth predictor moments are finite

$ARMAX$ model

Equation

$$y_t = c + \gamma_1 a_t + \gamma_2 b_t + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$
$$+ u_t + \alpha_1 u_{t-1} + \ldots + \alpha_q u_{t-q},$$

where a_t and b_t are predictors and (u_t) is white noise

- Series (y_t), (a_t), (b_t) stationary
- $\mathbb{E}(u_t \mid a_{t-1}, b_{t-1}, y_{t-1}, a_{t-2}, b_{t-2}, y_{t-2}, \ldots) = 0$
- The fourth predictor moments are finite

$ARMAX$ model is not completely equivalent to regression with $ARMA$ errors, but gives approximately the same quality of predictions

Properties of the $ARMAX$ model and regression with $ARMA$ errors

- If the model assumptions aren't violated, then the maximum likelihood estimators are consistent

Properties of the $ARMAX$ model and regression with $ARMA$ errors

- If the model assumptions aren't violated, then the maximum likelihood estimators are consistent
- For non-stationary variables (y_t) and predictors (a_t) and (b_t), we can switch to the first differences

Properties of the $ARMAX$ model and regression with $ARMA$ errors

- If the model assumptions aren't violated, then the maximum likelihood estimators are consistent
- For non-stationary variables (y_t) and predictors (a_t) and (b_t), we can switch to the first differences
- Estimators remain consistent if trend, seasonality dummy and trigonometric predictors are added

Properties of the $ARMAX$ model and regression with $ARMA$ errors

- If the model assumptions aren't violated, then the maximum likelihood estimators are consistent
- For non-stationary variables (y_t) and predictors (a_t) and (b_t), we can switch to the first differences
- Estimators remain consistent if trend, seasonality dummy and trigonometric predictors are added
- Not any predictor makes it possible to obtain a consistent estimator of the coefficient

Properties of the $ARMAX$ model and regression with $ARMA$ errors

- If the model assumptions aren't violated, then the maximum likelihood estimators are consistent
- For non-stationary variables (y_t) and predictors (a_t) and (b_t), we can switch to the first differences
- Estimators remain consistent if trend, seasonality dummy and trigonometric predictors are added
- Not any predictor makes it possible to obtain a consistent estimator of the coefficient
- Sometimes you can get good predictions even if the assumptions are violated

$ARDL$ model

ARDL — AutoRegressive Distributed Lag model

Autoregressive model with distributed lags

The $ARDL(p, q)$ model equation

$$y_t = c + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$

$$+ x_t + \alpha_1 x_{t-1} + \ldots + \alpha_q x_{t-q} + u_t$$

$ARDL$ model

ARDL — AutoRegressive Distributed Lag model

Autoregressive model with distributed lags

The $ARDL(p, q)$ model equation

$$y_t = c + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$
$$+ x_t + \alpha_1 x_{t-1} + \ldots + \alpha_q x_{t-q} + u_t$$

- (u_t) errors are white noise

$ARDL$ model

ARDL — AutoRegressive Distributed Lag model

Autoregressive model with distributed lags

The $ARDL(p, q)$ model equation

$$y_t = c + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$

$$+ x_t + \alpha_1 x_{t-1} + \ldots + \alpha_q x_{t-q} + u_t$$

- (u_t) errors are white noise
- Process (x_t) or process (Δx_t) is stationary

$ARDL$ model

ARDL — AutoRegressive Distributed Lag model

Autoregressive model with distributed lags

The $ARDL(p, q)$ model equation

$$y_t = c + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$
$$+ x_t + \alpha_1 x_{t-1} + \ldots + \alpha_q x_{t-q} + u_t$$

- (u_t) errors are white noise
- Process (x_t) or process (Δx_t) is stationary
- Process (y_t) is non-stationary, but (Δy_t) is stationary

$ARDL$ model

ARDL — AutoRegressive Distributed Lag model

Autoregressive model with distributed lags

The $ARDL(p, q)$ model equation

$$y_t = c + \beta_1 y_{t-1} + \ldots + \beta_p y_{t-p} +$$

$$+ x_t + \alpha_1 x_{t-1} + \ldots + \alpha_q x_{t-q} + u_t$$

- (u_t) errors are white noise
- Process (x_t) or process (Δx_t) is stationary
- Process (y_t) is non-stationary, but (Δy_t) is stationary
- $\mathbb{E}(u_t \mid y_{t-1}, x_{t-1}, y_{t-2}, x_{t-2}, \ldots) = 0$

Properties of the $ARDL$ model

- Using predictor lags (x_t) instead of noise lags (u_t)

Properties of the $ARDL$ model

- Using predictor lags (x_t) instead of noise lags (u_t)
- Suitable for non-stationary (y_t)

Properties of the $ARDL$ model

- Using predictor lags (x_t) instead of noise lags (u_t)
- Suitable for non-stationary (y_t)
- Used to find a long-term relationships between time series

Properties of the $ARDL$ model

- Using predictor lags (x_t) instead of noise lags (u_t)
- Suitable for non-stationary (y_t)
- Used to find a long-term relationships between time series
- If the model assumptions aren't violated, then the OLS estimators are consistent, although they are biased

Properties of the $ARDL$ model

- Using predictor lags (x_t) instead of noise lags (u_t)
- Suitable for non-stationary (y_t)
- Used to find a long-term relationships between time series
- If the model assumptions aren't violated, then the OLS estimators are consistent, although they are biased
- You can add multiple predictors with different number of lags

Predictors and $ARIMA$: Summary

- For stationary data you can use regression with $ARMA$ errors or $ARMAX$ model

Predictors and $ARIMA$: Summary

- For stationary data you can use regression with $ARMA$ errors or $ARMAX$ model
- Regression with $ARMA$ errors can be constructed for the differences

Predictors and $ARIMA$: Summary

- For stationary data you can use regression with $ARMA$ errors or $ARMAX$ model
- Regression with $ARMA$ errors can be constructed for the differences
- For non-stationary series it is sometimes possible to use the $ARDL$ model

Model Quality

Model Quality: Plan

- Scale-based metrics

Model Quality: Plan

- Scale-based metrics
- Percentage-based metrics

Remember the goal!

If the goal of building a model is forecasts one step ahead, then it is reasonable to compare models in predictive strength one step ahead.

Remember the goal!

If the goal of building a model is forecasts one step ahead, then it is reasonable to compare models in predictive strength one step ahead.

If the goal is to detect the moment of model discord, then it is reasonable to look for a model that gives the minimum error when there is no discord, and the maximum error when there is discord.

Notations for brevity

For the forecast, it is important when it is built, and for how many steps ahead:

$$\hat{y}_{t+h|t}$$

Notations for brevity

For the forecast, it is important when it is built, and for how many steps ahead:

$$\hat{y}_{t+h|t}$$

Sometimes for short:

$$\hat{y}_{t+h}$$

Notations for brevity

For the forecast, it is important when it is built, and for how many steps ahead:

$$\hat{y}_{t+h|t}$$

Sometimes for short:

$$\hat{y}_{t+h}$$

Problem:

$$\hat{y}_{(t+1)+2} \neq \hat{y}_{(t+2)+1}$$

Anti-quality metrics

Forecast error: $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$

Anti-quality metrics

Forecast error: $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$

Mean Absolute Error:

$$MAE = \frac{|e_{T+1}| + |e_{T+2}| + \ldots + |e_{T+H}|}{H}$$

Anti-quality metrics

Forecast error: $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$

Mean Absolute Error:

$$MAE = \frac{|e_{T+1}| + |e_{T+2}| + \ldots + |e_{T+H}|}{H}$$

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{e_{T+1}^2 + e_{T+2}^2 + \ldots + e_{T+H}^2}{H}}$$

Scaling

Convert error $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$ to percentage $p_t = e_t / y_t \cdot 100$
or $p_t^s = e_t / (0.5 y_t + 0.5 \hat{y}_t) \cdot 100$

Scaling

Convert error $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$ to percentage $p_t = e_t / y_t \cdot 100$
or $p_t^s = e_t/(0.5 y_t + 0.5 \hat{y}_t) \cdot 100$

Mean Absolute Percentage Error:

$$MAPE = \frac{|p_{T+1}| + |p_{T+2}| + \ldots + |p_{T+H}|}{H}$$

Scaling

Convert error $e_{t+h} = y_{t+h} - \hat{y}_{t+h}$ to percentage $p_t = e_t/y_t \cdot 100$ or $p_t^s = e_t/(0.5y_t + 0.5\hat{y}_t) \cdot 100$

Mean Absolute Percentage Error:

$$MAPE = \frac{|p_{T+1}| + |p_{T+2}| + \ldots + |p_{T+H}|}{H}$$

Symmetric Mean Absolute Percentage Error:

$$sMAPE = \frac{\left|p^s_{T+1}\right| + \left|p^s_{T+2}\right| + \ldots + \left|p^s_{T+H}\right|}{H}$$

Automatically compare with naive model

Naive: $\hat{y}_t^{naive} = y_{t-1}$ or $\hat{y}_t^{naive} = y_{t-12}$

Automatically compare with naive model

Naive: $\hat{y}_t^{naive} = y_{t-1}$ or $\hat{y}_t^{naive} = y_{t-12}$ Let's scale our forecast error e_t to MAE^{naive}:

$$q_t = \frac{e_t}{MAE^{naive}}$$

Automatically compare with naive model

Naive: $\hat{y}_t^{naive} = y_{t-1}$ or $\hat{y}_t^{naive} = y_{t-12}$ Let's scale our forecast error e_t to MAE^{naive}:

$$q_t = \frac{e_t}{MAE^{naive}}$$

Mean Absolute Scaled Error:

$$MASE = \frac{|q_{T+1}| + |q_{T+2}| + \ldots + |q_{T+H}|}{H}$$

Automatically compare with naive model

Naive: $\hat{y}_t^{naive} = y_{t-1}$ or $\hat{y}_t^{naive} = y_{t-12}$ Let's scale our forecast error e_t to MAE^{naive}:

$$q_t = \frac{e_t}{MAE^{naive}}$$

Mean Absolute Scaled Error:

$$MASE = \frac{|q_{T+1}| + |q_{T+2}| + \ldots + |q_{T+H}|}{H}$$

Comparing q to 1 compares our model with the naive one

Model Quality: Summary

- MAE, RMSE

Model Quality: Summary

- MAE, RMSE
- MAPE

Model Quality: Summary

- MAE, RMSE
- MAPE
- MASE

Model Comparison

Model Comparison: Plan

- Cross-validation

Model Comparison: Plan

- Cross-validation
- Akaike criterion

Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets

Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets
2. Evaluate several models on the training set

Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets
2. Evaluate several models on the training set
3. Predict each observation of the test sample using each model

Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets
2. Evaluate several models on the training set
3. Predict each observation of the test sample using each model
4. Calculate prediction errors

Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets
2. Evaluate several models on the training set
3. Predict each observation of the test sample using each model
4. Calculate prediction errors
5. Compare models by MAE and choose the best one

Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets
2. Evaluate several models on the training set
3. Predict each observation of the test sample using each model
4. Calculate prediction errors
5. Compare models by MAE and choose the best one

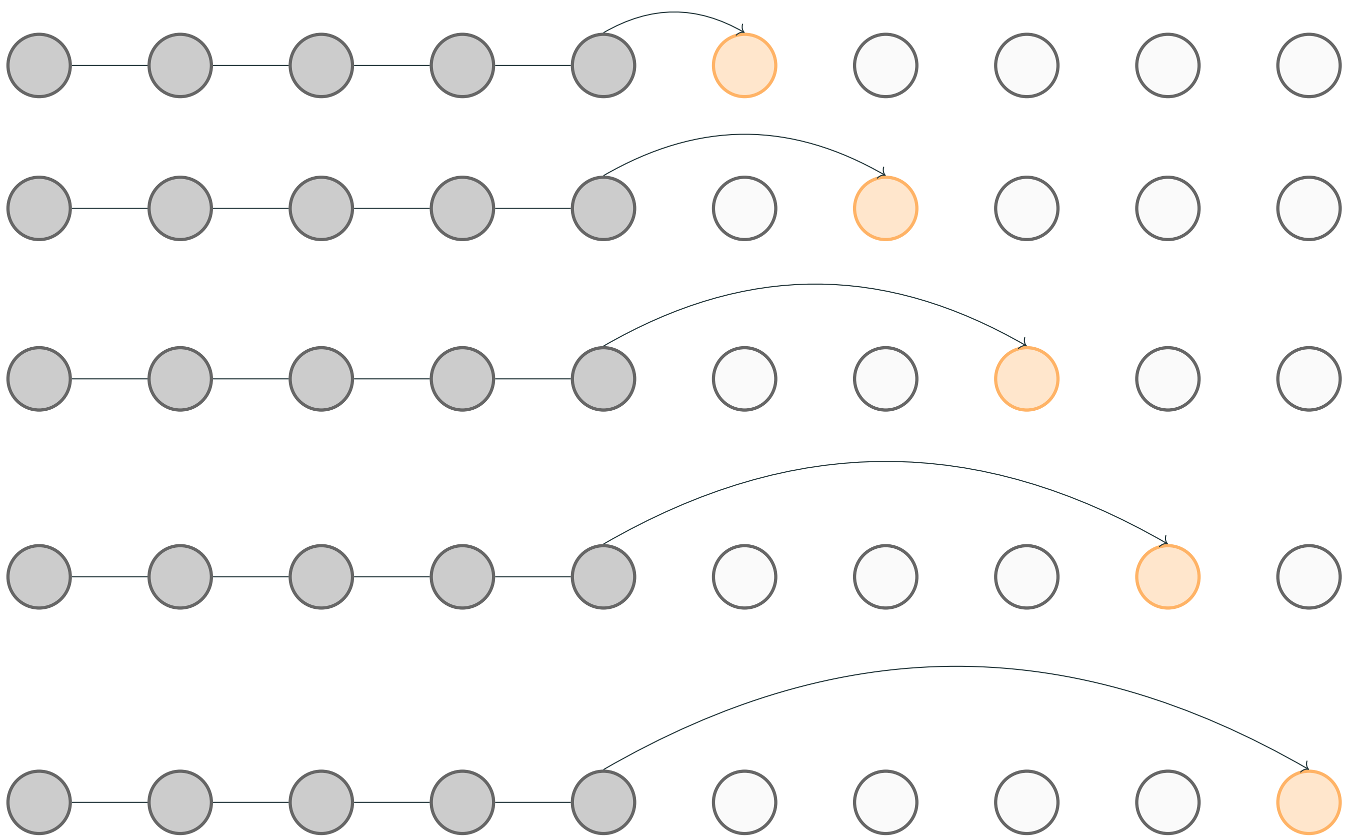
Train and test set

Strategy:

1. Split the entire sample into training (at the beginning) and test (at the end) sets
2. Evaluate several models on the training set
3. Predict each observation of the test sample using each model
4. Calculate prediction errors
5. Compare models by MAE and choose the best one

Disadvantage: forecasts have different horizons

Dividing into train and test



Sliding window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)

Sliding window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the train set

Sliding window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the train set
3. Predict one step ahead with each model

Sliding window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the train set
3. Predict one step ahead with each model
4. Calculate prediction errors

Sliding window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the train set
3. Predict one step ahead with each model
4. Calculate prediction errors
5. Shift the training sample one observation to the right

Sliding window Cross-validation

Strategy:

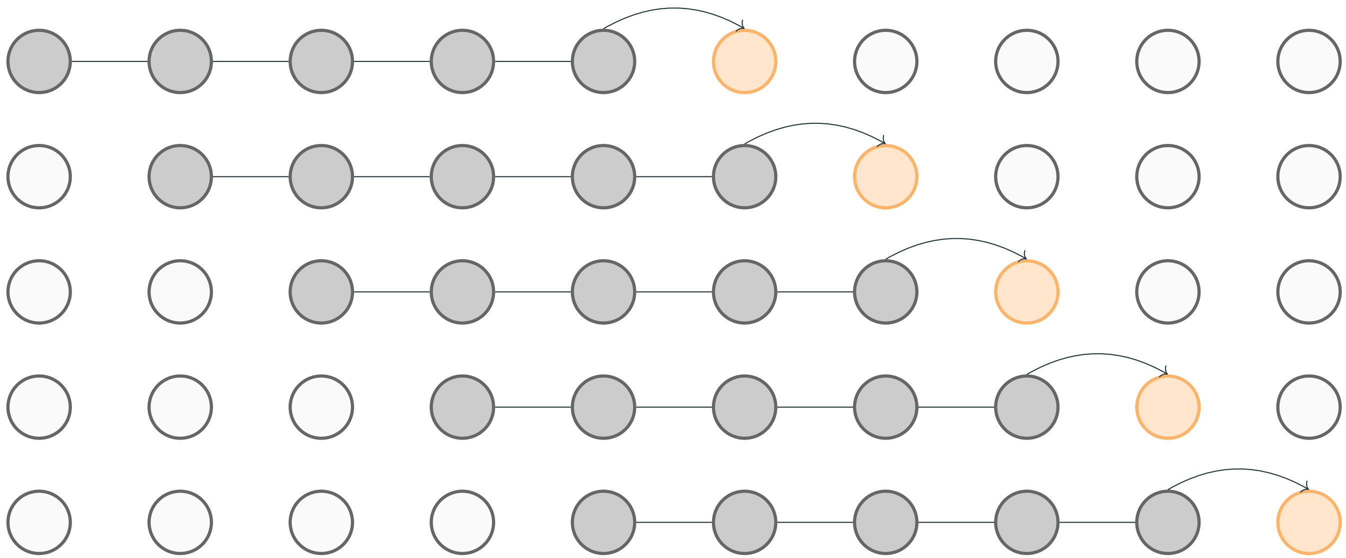
1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the train set
3. Predict one step ahead with each model
4. Calculate prediction errors
5. Shift the training sample one observation to the right
6. Repeat steps 2-5

Sliding window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the train set
3. Predict one step ahead with each model
4. Calculate prediction errors
5. Shift the training sample one observation to the right
6. Repeat steps 2-5
7. Compare models by MAE and choose the best one

Sliding window Cross-validation

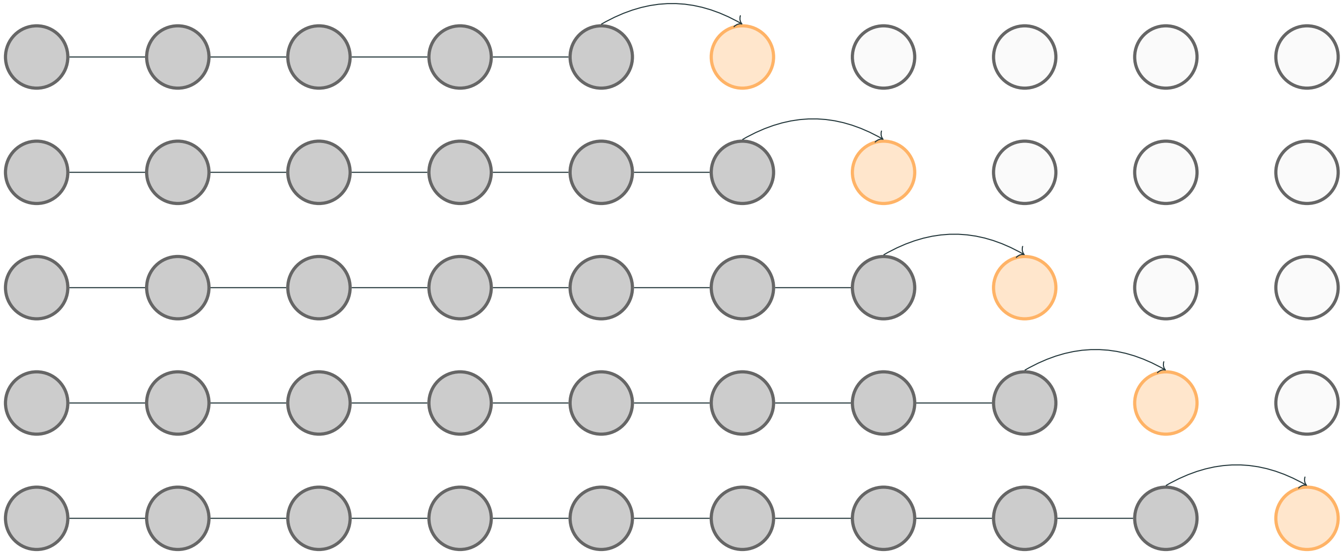


Growing window Cross-validation

Strategy:

1. Select the starting size for train sample (at the beginning)
2. Evaluate several models on the training set
3. Predict one step ahead with each model
4. Calculate prediction errors
5. Increase the training set by one observation.
6. Repeat steps 2-5
7. Compare models by MAE and choose the best one

Growing window Cross-validation



Cross-validation Discussion

Sliding window cross-validation: there are many observations and we suspect that dependencies between values can change.

Cross-validation Discussion

Sliding window cross-validation: there are many observations and we suspect that dependencies between values can change.

Growing window cross-validation: there are few observations or we are sure that the dependency persists.

Cross-validation Discussion

Sliding window cross-validation: there are many observations and we suspect that dependencies between values can change.

Growing window cross-validation: there are few observations or we are sure that the dependency persists.

Cross-validation can be time consuming!

Let's make cross-validation quicker!

Approximate cross-validation by one step forward based on $RMSE$ using... Akaike Information Criterion:

Let's make cross-validation quicker!

Approximate cross-validation by one step forward based on $RMSE$ using... Akaike Information Criterion:

$$AIC = -2 \ln L + 2k,$$

where $\ln L$ is the logarithm of the maximum likelihood on the training set, k is the total number of model parameters

Nuances of AIC

- AIC has theoretical grounds:

$$\frac{AIC_A - AIC_B}{2} \approx KL(\text{Truth}||\text{Model A}) -$$

$$-KL(\text{Truth}||\text{Model B})$$

Nuances of AIC

- AIC has theoretical grounds:

$$\frac{AIC_A - AIC_B}{2} \approx KL(\text{Truth}||\text{Model A}) - \\ -KL(\text{Truth}||\text{Model B})$$

- May be used for non-nested models

Nuances of AIC

- AIC has theoretical grounds:

$$\frac{AIC_A - AIC_B}{2} \approx KL(\text{Truth}||\text{Model A}) -$$
$$- KL(\text{Truth}||\text{Model B})$$

- May be used for non-nested models
- For Gaussian y_t models, the criterion approximates comparison over $RMSE$

Nuances of AIC

- AIC has theoretical grounds:

$$\frac{AIC_A - AIC_B}{2} \approx KL(\text{Truth}||\text{Model A}) - \\ -KL(\text{Truth}||\text{Model B})$$

- May be used for non-nested models
- For Gaussian y_t models, the criterion approximates comparison over $RMSE$
- The models being compared must be for the same observations

Model Comparison: Summary

- Cross-validation: sliding and growing window

Model Comparison: Summary

- Cross-validation: sliding and growing window
- AIC is a fast and approximate analogue of cross-validation

Forecast comparison

Forecast comparison: Plan

- Diebold-Mariano test

Forecast comparison: Plan

- Diebold-Mariano test
- Test assumptions

Forecast comparison: Plan

- Diebold-Mariano test
- Test assumptions
- Test implementation

Diebold-Mariano Test

- Used to compare two forecasts

Diebold-Mariano Test

- Used to compare two forecasts
- Compares the forecasts for the specified forecast horizon h

Diebold-Mariano Test

- Used to compare two forecasts
- Compares the forecasts for the specified forecast horizon h
- Not optimal for model comparison

Diebold-Mariano Test

- Used to compare two forecasts
- Compares the forecasts for the specified forecast horizon h
- Not optimal for model comparison
- Not suitable for pairwise comparison of multiple forecasts

DM test assumptions

Consider difference of two forecast losses:

$$d_t = e^2_{A,t} - e^2_{B,t}, \quad e_{\text{Model},t} = \hat{y}_{\text{Model},t} - y_t$$

DM test assumptions

Consider difference of two forecast losses:

$$d_t = e^2_{A,t} - e^2_{B,t}, \quad e_{\text{Model},t} = \hat{y}_{\text{Model},t} - y_t$$

Difference d_t assumed to be stationary:

DM test assumptions

Consider difference of two forecast losses:

$$d_t = e^2_{A,t} - e^2_{B,t}, \quad e_{\text{Model},t} = \hat{y}_{\text{Model},t} - y_t$$

Difference d_t assumed to be stationary:

$$\mathbb{E}(d_t) = \mu_d,$$

DM test assumptions

Consider difference of two forecast losses:

$$d_t = e^2_{A,t} - e^2_{B,t}, \quad e_{\text{Model},t} = \hat{y}_{\text{Model},t} - y_t$$

Difference d_t assumed to be stationary:

$$\mathbb{E}(d_t) = \mu_d,$$

$$\text{Cov}(d_t, d_{t-k}) = \gamma_k,$$

DM test assumptions

Consider difference of two forecast losses:

$$d_t = e^2_{A,t} - e^2_{B,t}, \quad e_{\text{Model},t} = \hat{y}_{\text{Model},t} - y_t$$

Difference d_t assumed to be stationary:

$$\mathbb{E}(d_t) = \mu_d,$$

$$\text{Cov}(d_t, d_{t-k}) = \gamma_k,$$

in particular,

$$\text{Var}(d_t) = \gamma_0$$

Test method

Under the correct $H_0 : \mu_d = 0$:

$$DM = \frac{\bar{d}}{se(\bar{d})} \to \mathcal{N}(0; 1),$$

where $se^2(\bar{d})$ is a consistent estimator for $\mathrm{Var}(\bar{d})$

Test method

Under the correct $H_0 : \mu_d = 0$:

$$DM = \frac{\bar{d}}{se(\bar{d})} \to \mathcal{N}(0; 1),$$

where $se^2(\bar{d})$ is a consistent estimator for $\mathrm{Var}(\bar{d})$

In practice, we evaluate the regression on a constant

$$\hat{d}_t = \hat{\beta}_1,$$

Test method

Under the correct $H_0 : \mu_d = 0$:

$$DM = \frac{\bar{d}}{se(\bar{d})} \to \mathcal{N}(0;1),$$

where $se^2(\bar{d})$ is a consistent estimator for $\mathrm{Var}(\bar{d})$

In practice, we evaluate the regression on a constant

$$\hat{d}_t = \hat{\beta}_1,$$

get $\hat{\beta}_1 = \bar{d}$ and use robust standard errors,

$$DM = \frac{\hat{\beta}_1}{se_{HAC}(\hat{\beta}_1)}.$$

How does robust estimators work?

Compare forecasts for P steps ahead,

$$\mathrm{Var}(\bar{d}) = \frac{\mathrm{Var}(d_1) + \mathrm{Var}(d_2) + \ldots + 2\,\mathrm{Cov}(d_1, d_2) + \ldots}{P^2}$$

How does robust estimators work?

Compare forecasts for P steps ahead,

$$\mathrm{Var}(\bar{d}) = \frac{\mathrm{Var}(d_1) + \mathrm{Var}(d_2) + \ldots + 2\,\mathrm{Cov}(d_1, d_2) + \ldots}{P^2}$$

From the stationarity of d_t, the variance of $\mathrm{Var}(\bar{d})$ is

$$\frac{P\gamma_0 + 2(P-1)\gamma_1 + 2(P-2)\gamma_2 + \ldots}{P^2}$$

How does robust estimators work?

Compare forecasts for P steps ahead,

$$\mathrm{Var}(\bar{d}) = \frac{\mathrm{Var}(d_1) + \mathrm{Var}(d_2) + \ldots + 2\,\mathrm{Cov}(d_1, d_2) + \ldots}{P^2}$$

From the stationarity of d_t, the variance of $\mathrm{Var}(\bar{d})$ is

$$\frac{P\gamma_0 + 2(P-1)\gamma_1 + 2(P-2)\gamma_2 + \ldots}{P^2}$$

The naive estimator for the variance of $\widehat{\mathrm{Var}}(\bar{d})$ is

$$\frac{P\hat{\gamma}_0 + 2(P-1)\hat{\gamma}_1 + 2(P-2)\hat{\gamma}_2 + \ldots}{P^2}$$

Why compare forecasts?

Nuance: comparison of forecasts and comparison of models are different tasks.

Why compare forecasts?

Nuance: comparison of forecasts and comparison of models are different tasks.

A model can win a lot in simplicity and lose a little in predictions.

Why compare forecasts?

Nuance: comparison of forecasts and comparison of models are different tasks.

A model can win a lot in simplicity and lose a little in predictions.

On a small sample loss of information about the quality of forecasts on the training sample is significant.

Why compare forecasts?

Nuance: comparison of forecasts and comparison of models are different tasks.

A model can win a lot in simplicity and lose a little in predictions.

On a small sample loss of information about the quality of forecasts on the training sample is significant.

Forecast comparison: Summary

- The Diebold-Mariano test is suitable for comparing two forecasts

Forecast comparison: Summary

- The Diebold-Mariano test is suitable for comparing two forecasts
- Comparing forecasts and comparing models are slightly different tasks