Machine Learning for Forecasting

Christoph Bergmeir

October 12th, 2023

University of Granada, Spain
https://www.cbergmeir.com

Introduction

What this talk is about

- This talk is mainly about "POML" for forecasting: Plain Old Machine Learning.
- Will cover deep learning briefly, but not in much detail
- Will not cover generative foundational models (GPT-like models)

The main intention of this training course is that you understand the basic problems and their general solutions in forecasting, for modelling and evaluation, so that you are enabled to use and judge results of complicated ML methods.

Why learn about POML if nowadays it is all about deep learning?

- Transformers, Graph Neural Networks, etc. may be all the fuzz.
- However: Many recent transformer papers have flawed experimental setups, misrepresenting their capabilities on the used datasets.
- They use series with thousands of data points per series.
- Companies like Amazon, Google, and Zalando are getting good results from them, but on datasets of hundreds of thousands of series.
- They have been largely irrelevant in the M5 competition, even though many deep-learning architectures were already available
- If you have 1k daily series, will deep-learning be the right method? Maybe, maybe not.
- Also see our results on forecastingdata.org that are quite mixed.

Why is this relevant if in 2 years it will all be done with some GPT variant?

- Foundational time series models are coming (see "timeGPT'' (Garza and Mergenthaler-Canseco, 2023))
- Will they work on any possible frequency? Probably yes, at least on the ~20 most common frequencies.
- Will they work with covariates? Maybe with embeddings/some form of dimensionality reduction? Maybe in a way like TabPFN (Hollmann et al., 2022)? Stacking?
- Will they work with your constraints in production on hardware, time frames, privacy/data sharing concerns? Will there be open-source pre-trained models?

–> Still a lot of unknowns.

My background

- Have been in forecasting since the start of my PhD at University of Granada, about 2009
- Have worked at Monash University (e.g., with Rob Hyndman), did recently a sabbatical in industry, at Meta Inc., and I'm now back to University of Granada.
- Topics I have worked on:
  - Forecasting for fault detection in wind turbines
  - Forecasting for pest outbreaks in green houses
  - Predictive maintenance
  - Renewable energy production forecasting (wind/solar)
  - Energy demand forecasting
  - Energy price forecasting
  - Retail sales/demand forecasting

What I forecast

- Sales forecasting in the supply chain, retail
- Forecasting in Energy
  - Demand
  - Prices
  - Renewable energy production
  - building efficiency
- Predictive maintenance (road, railway, ...mining)
- ...

What I don't forecast

- The weather
  - Lots of domain knowledge and specialised models exist
  - We leave it to the meteorologists
  - We often use weather forecasts as inputs to our models
- The stock market

Stock market forecasting (1)

Pros:

- Lots of freely available high-frequency data
- Clear motivation and use case (making money)

Stock market forecasting (2)

Cons:

- Shareprice not a function of its own past, but of its anticipated future
- Low signal to noise ratio
- Markets tend to be close to "efficient' '
- Forecasting in efficient markets is not possible (beyond the naive forecast)
- R. Engle: got the Nobel prize in Economics in 2003 "for methods of analyzing economic time series with time-varying volatility (ARCH)."
- All about predicting risk, not returns (i.e., predicting the tails of the distributions)

Stock market forecasting (3)

–> Predicting stock market returns from just a couple of lags won't work, no matter what method you are using.

–> All this is true in a similar way for exchange rates!

A bit of forecasting terminology

- model fitting, parameter estimation: model training
- in-sample: training set
- out-of-sample: test set
- forecast horizon: target variable
- forecast origin: from where you do the forecasting from, the last known observation
- rolling origin: the forecast origin changes (for every point in the test set)
- fixed origin: the forecast origin is fixed

A bit of forecasting terminology (2)

- forecast combination: ensembling
- lags, independent variables, regressors, covariates, predictors: features, inputs
- dummy variable, indicator variable: one-hot encoding
- seasonality, seasonal period: cyclic change in mean of the series. Its length is know a priori and will not change in the future
- trend: (smooth) change in the mean of the series

Naive and mean forecast

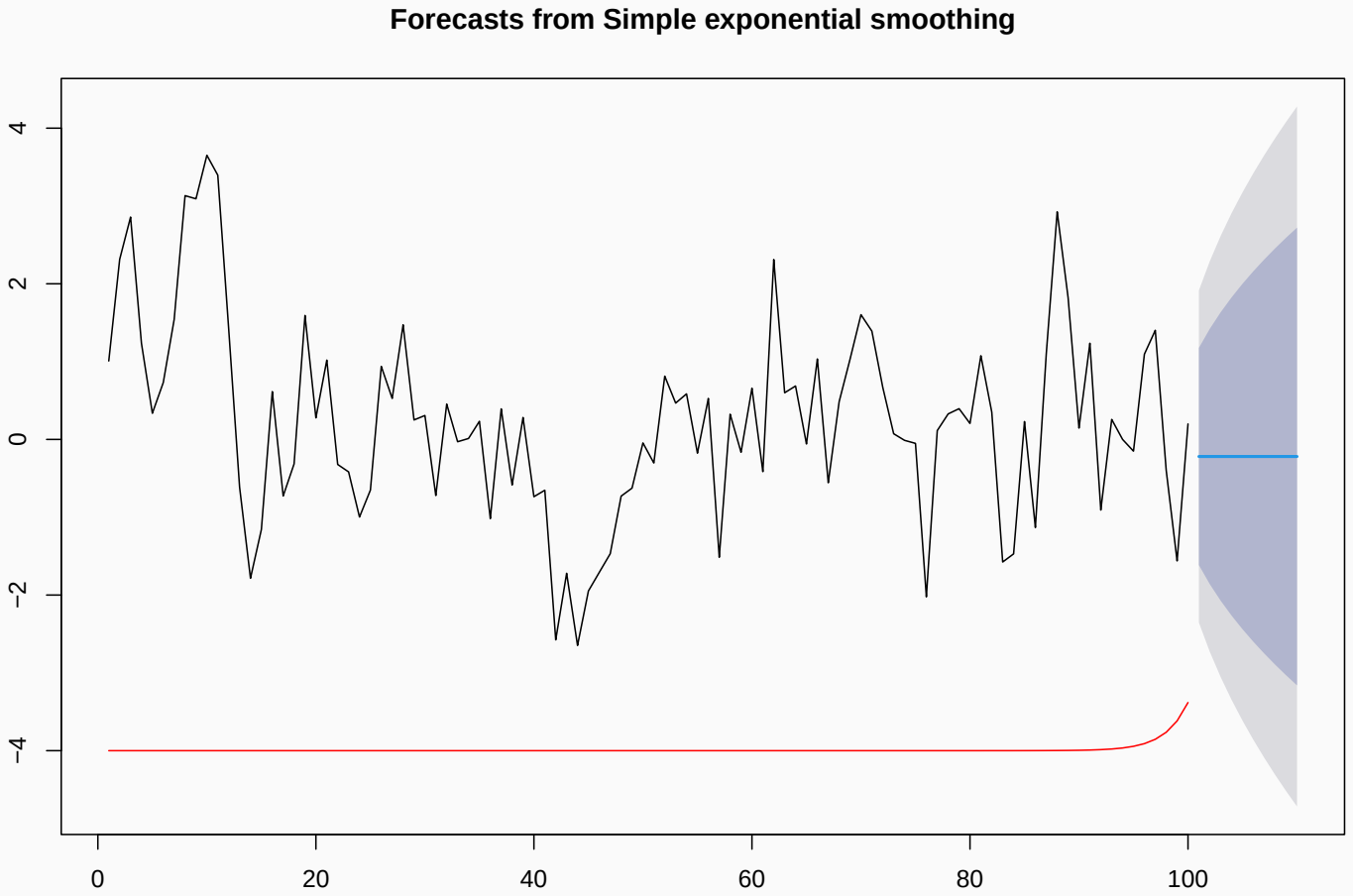
- Naive forecast: We use the last known observation as our forecast
  - Effectively weighting the last observation with 1, all the others with 0
- Mean forecast: We calculate the mean over all observations
  - All observations are weighted equally
- Naive and mean forecast are two extreme cases.

A central assumption in forecasting is often that the more recent past is more important than the more distant past.

Idea: What about weighting the observations with an exponential decay?

–> Exponential smoothing

Simple exponential smoothing (SES)



red line: exponentially decaying weights, $\alpha = 0.62$.

Exponential smoothing

- Exponential smoothing applied to components: level, seasonality, and trend
- Holt (1957), and his student Winters (1960); (see Goodwin, 2010, for an overview)
- Was used for a long time as a relatively ad-hoc method without a theoretical underpinning
- Hyndman et al. (2002) gave it a solid statistical foundation in state-space models
- ETS stands for both ExponenTial Smoothing and Error, Trend, and Seasonality

Box and Jenkins (1970): Linear modelling...

Autoregressive moving average model (ARMA):

$$\hat{x}_{t+1} = c + \phi_1 x_t + \cdots + \phi_p x_{t-p+1} + \theta_1 \epsilon_t + \cdots + \theta_q \epsilon_{t-q+1}$$

- Model is linear in the lags and linear in the errors
- When fitting the model, where do the errors come from?
- Need to estimate some initial conditions and step through the whole series (in ETS as well)
- No closed-form solution
- Need a non-linear fitting procedure. By default in R: BFGS
- fitting can be slow in long time series

ARIMA

Integrated:

- Do we want to model the values directly or the change to the last value?
- addresses non-stationarity
- Preprocessing step: Do differencing of the series before we do the ARMA
- Pro: Hopefully makes the series stationary
- Contra: We lose some information about the scale in the preprocessing

ARIMA (cont'd)

- Any (stationary) AR(1) model has an equivalent MA(∞) model, and any (invertible) MA(1) model has an equivalent AR(∞) model (Hyndman and Athanasopoulos, 2018)
- Thus, we can approximate the MA part of the ARMA model with a higher order AR part
- In practice, this "higher order" often is not very high to get satisfactory results (in Econometrics, 5 is often a "high order" already)

–> Can be seen as a re-parametrisation, to get fewer parameters and a smaller input window, at the cost of more complex model fitting

"Forecasting: principles and practice'' book

For more details, the standard resource for traditional time series forecasting is:

Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2

A brief history of Machine Learning in forecasting

M3 competition (1998)

- Results published in Makridakis and Hibon (2000)
- Had as a central conclusion that complex methods do not necessarily outperform simple methods
- ETS and ARIMA were the "complex" models, and simple models were models like a random walk with drift
- The only neural network that participated was quite bad
- Won by the "theta" method, which was later shown to be an average of a linear regression and simple exponential smoothing with drift
- **The** benchmark dataset in forecasting for almost 20 years
- A lot of focus (too much?) on this dataset

Controversy of ML vs Statistical methods

Long-standing controversy in the forecasting field, whether Machine Learning or Statistical methods work better for forecasting

- Forecasters "knew" that simple methods work best
  - ... because they had won the M3, the NN3, the NN5
- Machine Learners "knew" that Neural Networks work best
  - ... because, hey, it's a NN, it's a universal approximator
- Thousands of papers in Neural Network journals and others
- "A Novel method X for stock market forecasting..."
- Cherry-picked datasets, no proper benchmarking

–> Every time I hear somebody say "Informer is the state-of-the-art in forecasting" I get reminded that this is still very relevant today.

Controversy of ML vs Statistical methods (cont'd)

S Makridakis, E Spiliotis, V Assimakopoulos (2018), Statistical and Machine Learning forecasting methods: Concerns and ways forward. PloS one 13 (3), e0194889.

- Makridakis et al. (2018b) benchmarked ML methods against statistical benchmarks
- not surprisingly, the ML methods lost
- The paper is published in PLOS ONE because it got rejected at 2-3 Neural Network outlets beforehand.

Controversy of ML vs Statistical methods (cont'd)

- Reasons for rejection were that: there are many ML methods that have "proven to overcome the results provided," though the editors and reviewers didn't name any in particular
- Makridakis was upset over this and did what he seemingly does best when he is upset...organise another competition: the M4
- so that Machine Learners could submit forecasts and show that their methods work well

Controversy of ML vs Statistical methods (cont'd)

- The history explains why the forecasting field got so hung up on this controversy
- Still somewhat surprising as Forecasting emerged from Statistics just in the same way Machine Learning emerged from Statistics

–> Focus on OOS accuracy, not model properties

–> Some things were discovered in parallel in both fields:

- forecast combination (Bates and Granger, 1969) vs ensembling (in ML in the 1990s)
- simple methods vs regularisation

Controversy of ML vs Statistical methods (cont'd)

- Forecasters have been used to ML promising much and hardly delivering anything
- This has changed in recent years
- The arrogance and ignorance inherent to the field of Machine Learning ("disruptive mindset") doesn't help either.
- Forecasting is (usually) not like image classification or NLP, with very complex input data structures but zero noise.

–> Forecasting and Machine Learning should learn from each other!

M4 competition (2018)

- 100k series, across many different domains
- hourly, daily, weekly, monthly, quarterly, yearly series
- prediction intervals

Results

- published in Makridakis et al. (2018a)
- most forecasters expected that a combination approach of statistical methods would win
- however, such methods got 2nd and 3rd
- winner was RNN-ES, a hybrid between RNN and ETS, by Smyl (2020)
- a big surprise to many people that a ML model could win
- ... but not so much for me

Computational Intelligence in Forecasting (CIF) 2016 competition (Štěpnička and Burda, 2016)

- ... I got beaten in a competition by Slawek Smyl 2 years earlier
- 72 monthly series (lengths 22-108)
- I participated with plain ETS and BaggedETS
- the BaggedETS won several sub-categories and was also winning on median sMAPE
- On the competition metric mean sMAPE, BaggedETS got 9th, plain ETS got 3rd
- 1st and 2nd place were LSTMs, from Slawek Smyl.
- Globally trained across series

Global models

Name "global models" was introduced by Januschowski et al. (2020). It is arguably not a good name but it's the name we got for now.

- Traditionally, *one* time series is seen as a dataset
- One model is built per time series
- low sampling frequencies and non-stationarities like structural breaks make usually that we don't have enough data to fit complex (ML) models
- M3, M4 datasets are put together under this paradigm; very different series from different frequencies, different domains, etc.

Global models (cont'd)

Paradigm shift:

- a *set* of time series is a dataset (e.g., a set of series from retail, smart meters, etc.)
- build a model across the series
- names: global modelling, cross-learning, multi-task learning, pooled regression

–> Now, enough data, due to more series.

–> ML methods are competitive now

Global models (cont'd)

- Local model: typically fitting a model with few (<10) parameters to a single series
- if you have 10k series and fit 5 parameters, you end up with 50k parameters

–> fit a global model with 5k parameters instead

–> Overall complexity of set of local models grows when dataset grows; complexity of global model stays the same

Global models (cont'd)

- Global models can afford to be more complex
- Complexity can be added as:
  - longer memory (longer input windows, more lags)
  - non-linear/non-parametric models (NNs, GBT, ...)
  - data partitioning

Global models are not multivariate models

- Global models learn across series but predict every series in isolation
- they can work on datasets where series have different lengths and/or are not aligned, like the M3, M4 datasets
- they do not take into account interactions between series
- the concepts are orthogonal: methods can be local/univariate, global/univariate, local/multivariate, global/multivariate
- before, we talked about local/univariate models
- now, we talk about global/univariate models, later we will talk about local/multivariate models
- This course doesn't cover global/multivariate modelling

Global models are not multivariate models (cont'd)

History of global models

- Dating back to the early 2000s and earlier (Duncan et al., 2001)
- Pooled regression a standard statistics technique, see, e.g.: Gelman et al. (2007)
- Pooled regression for forecasting: Trapero et al. (2015)
- 2016: Smyl and Kuber (2016), CIF competition: Štěpnička and Burda (2016)
- 2017: DeepAR (Salinas et al., 2019b) and other works from Amazon, e.g., Wen et al. (2017); our work in Bandara et al. (2017)
- after 2018: ... many more works

Kaggle competitions

- A good overview give Bojer and Meldgaard (2020)
- The following are relevant forecasting competitions held on Kaggle:
  - Walmart Store Sales Forecasting (2014)
  - Walmart Sales in Stormy Weather (2015)
  - Rossmann Store Sales (2015)
  - Wikipedia Web Traffic Forecasting (2017)
  - Corporación Favorita Grocery Sales Forecasting (2018)
  - Recruit Restaurant Visitor Forecasting (2018)
- All competitions were won by global models, the latter four by either GBT or NN models or ensembles of those.

M5 competition

- held in 2020 on Kaggle
- dominated by LightGBM: Makridakis et al. (2020), DeepAR and NBEATS also successful
- A team of my students won a Kaggle Gold, 17th in the competition out of >5k participants.
- They used an ensemble of LightGBM and pooled (linear) regression.

Global models (cont'd)

- Global models have shown success to a surprising / unreasonable degree
- Idea in recent years was that the series have to be in some way "related/similar" so that we can learn something useful across them
- "Related" in terms of similarity of their DGP (not necessarily mere correlations)

Global models (cont'd)

Montero-Manso and Hyndman (2020):

- Global model can produce the same forecasts as local models, without any assumptions about similarity

–> The series don't have to be "related"

- Instead of fitting one complex pattern across series, a global model even works well to fit many simple patterns that are different in the series
- This result is quite remarkable, similar results, e.g., in multi-task learning, don't exist

Machine Learning methods for forecasting

Machine Learning methods for forecasting

Nowadays often a good option

- global modelling across series
- can apply them to a single time series, if long enough
- longer series due to finer granularities (secondly, minutely, half-hourly series available over years)
- additional metadata

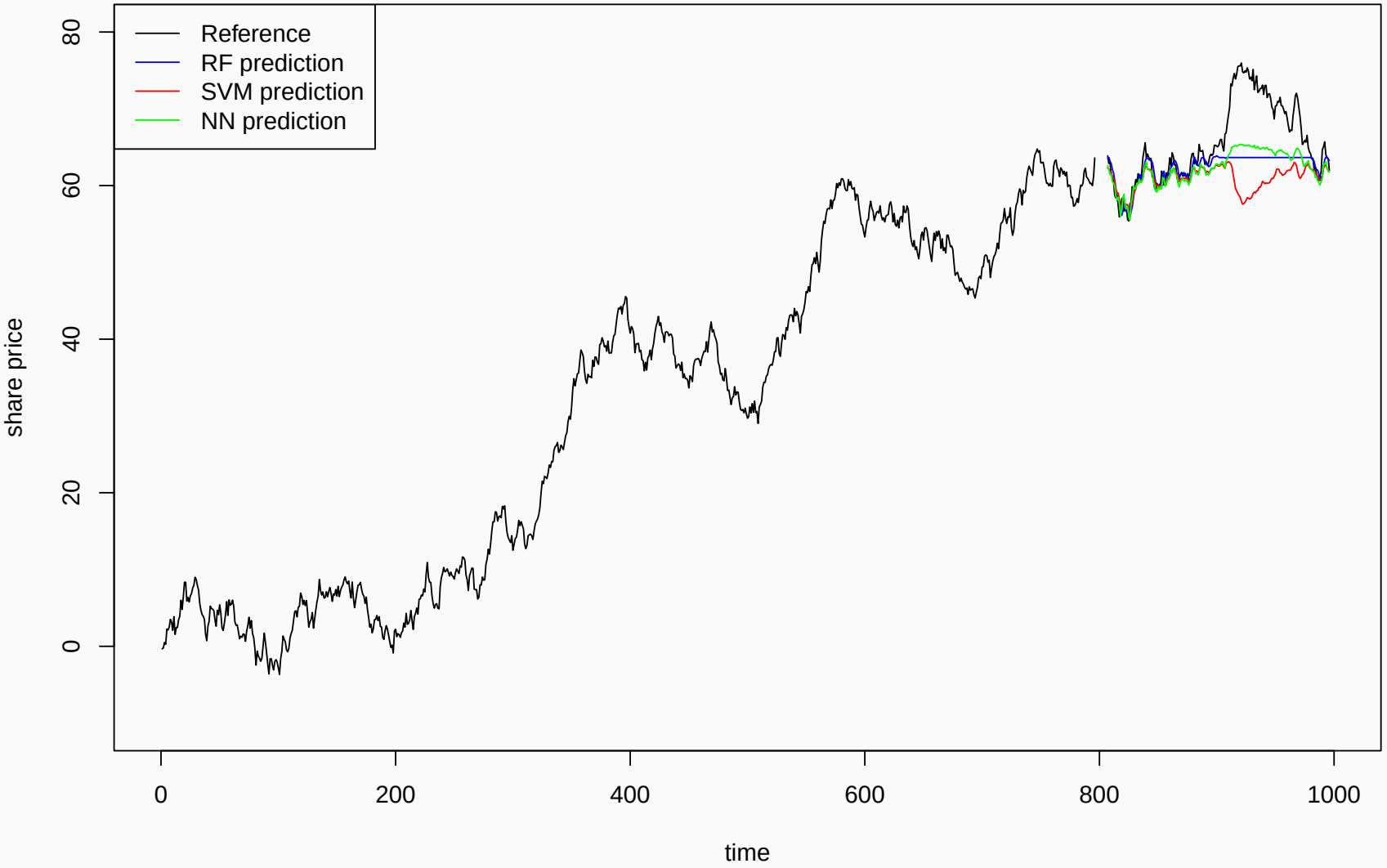
Non-linear autoregression

- basic setup is a (non-linear, non-parametric) autoregressive model
- then, you can use your favourite ML method out of the box
- we've seen that this can approximate an ARMA model, if we choose the input window larger than the ARMA model has it

Problem: Non-stationarity

- Data Distribution changes over time
- Many (most?) real-world problems have a time component and changing distributions
- Think about detecting cars on the street with a dataset from the 1970s
- In time series it is more explicit though and has more impact

Problem: Non-stationarity (2)



Problem: Non-stationarity (3)

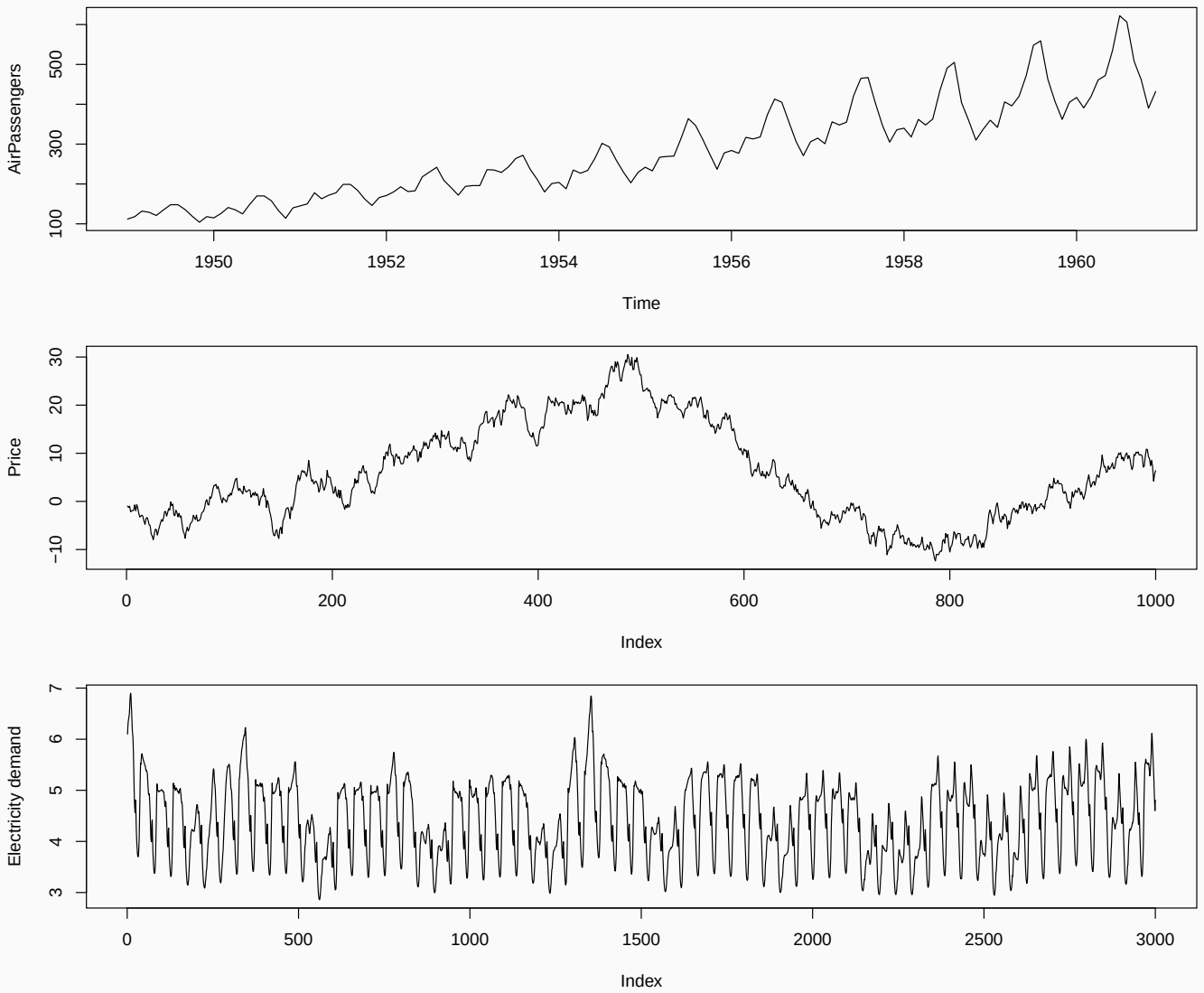
> *Dividing the world into linear and non-linear is like dividing the world into bananas and non-bananas.*

Just as there are many different forms of non-linearity, there are also many different forms of non-stationarity

Typical non-stationarities in time series:

- change in mean: seasonality, trend
- change in variance: heteroskedasticity
- stochastic trends: random walks (unit roots)

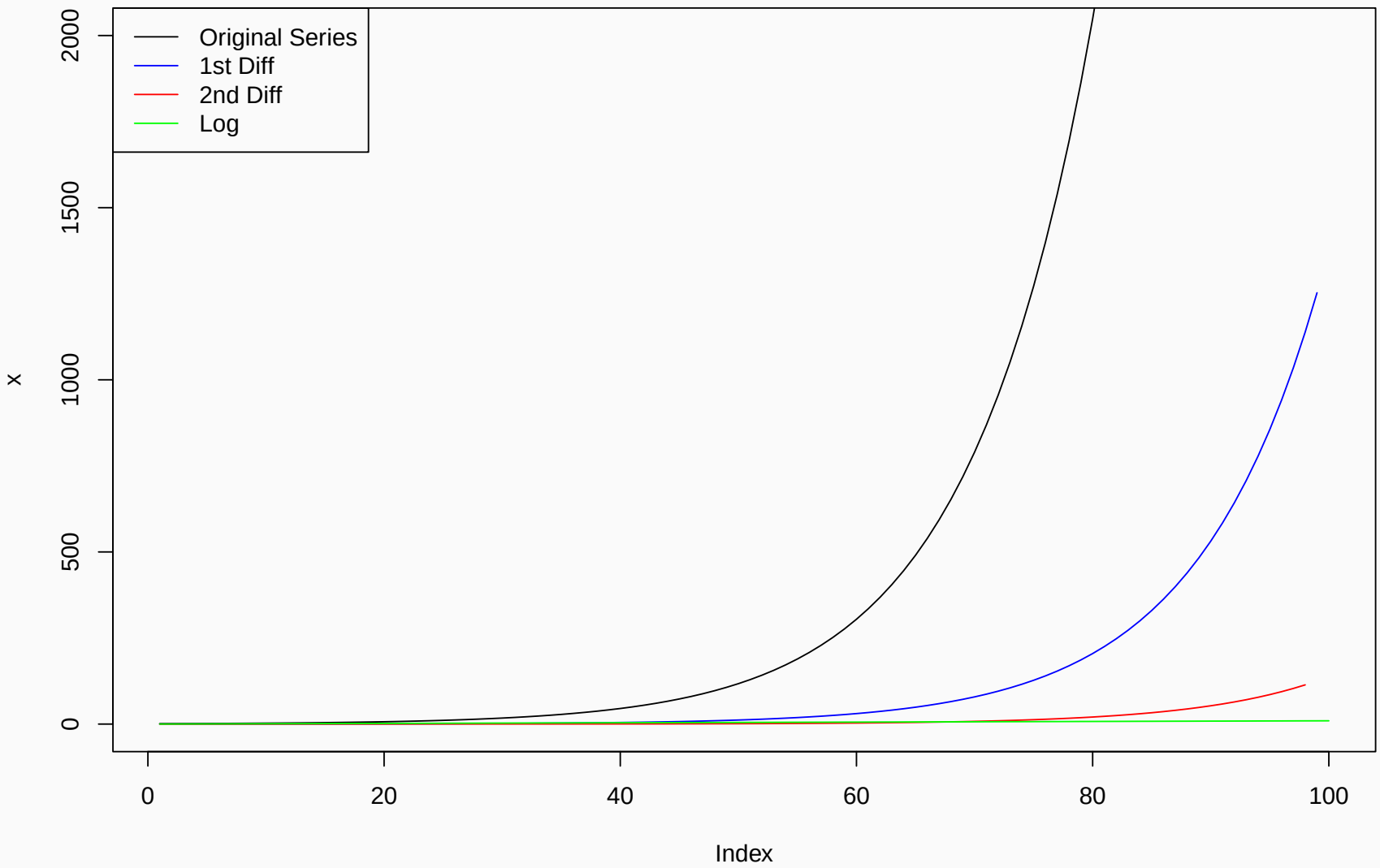
Problem: Non-stationarity (4)



How to achieve stationarity?

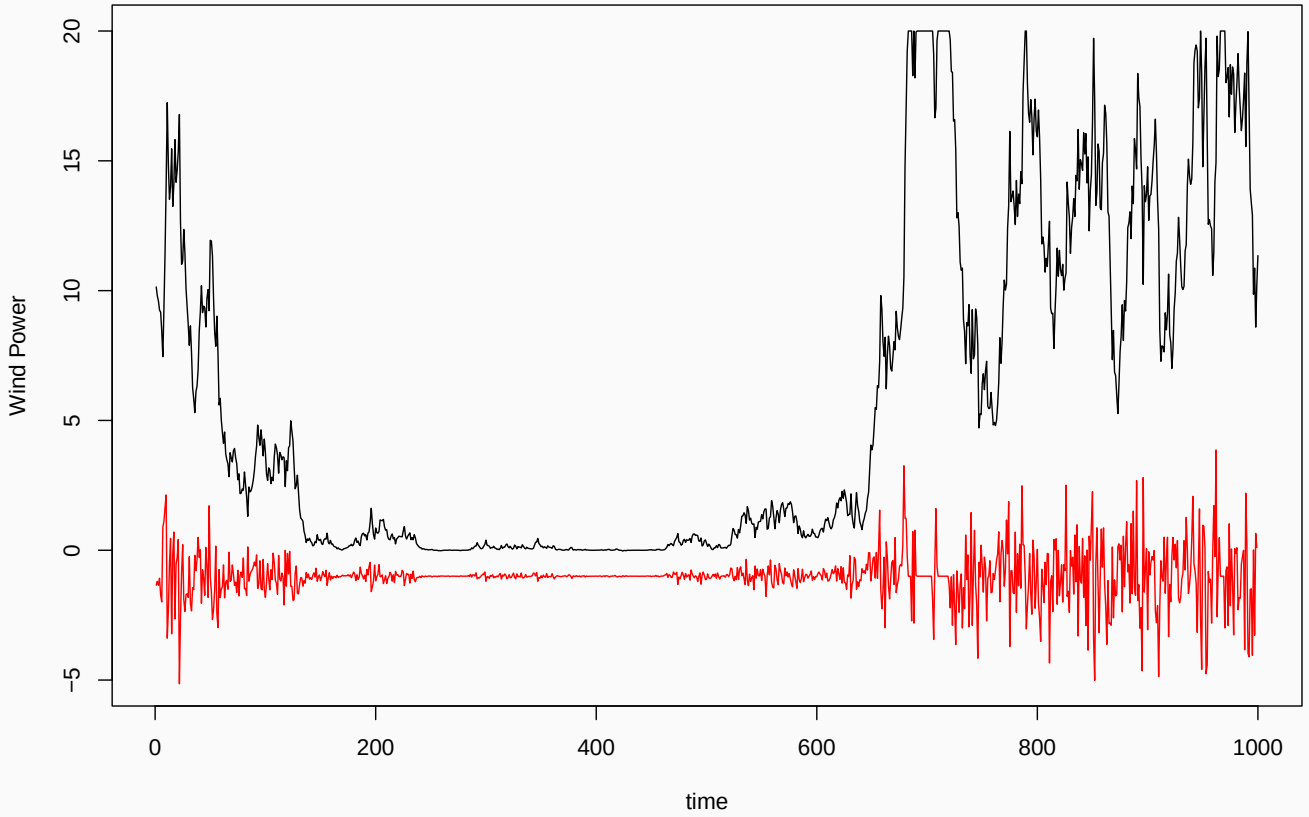
- As in Econometrics and Finance, many series are close to random walks, there, differencing is a common tool to achieve stationarity
- see ARIMA modelling earlier
- Differencing only solves some forms of non-stationarity, not others
- Loosing information about the scale. Can have an additional input as the (log of) the original scale.
- Differencing can help to make ML models more robust

Differencing does not always work



Differencing loses information

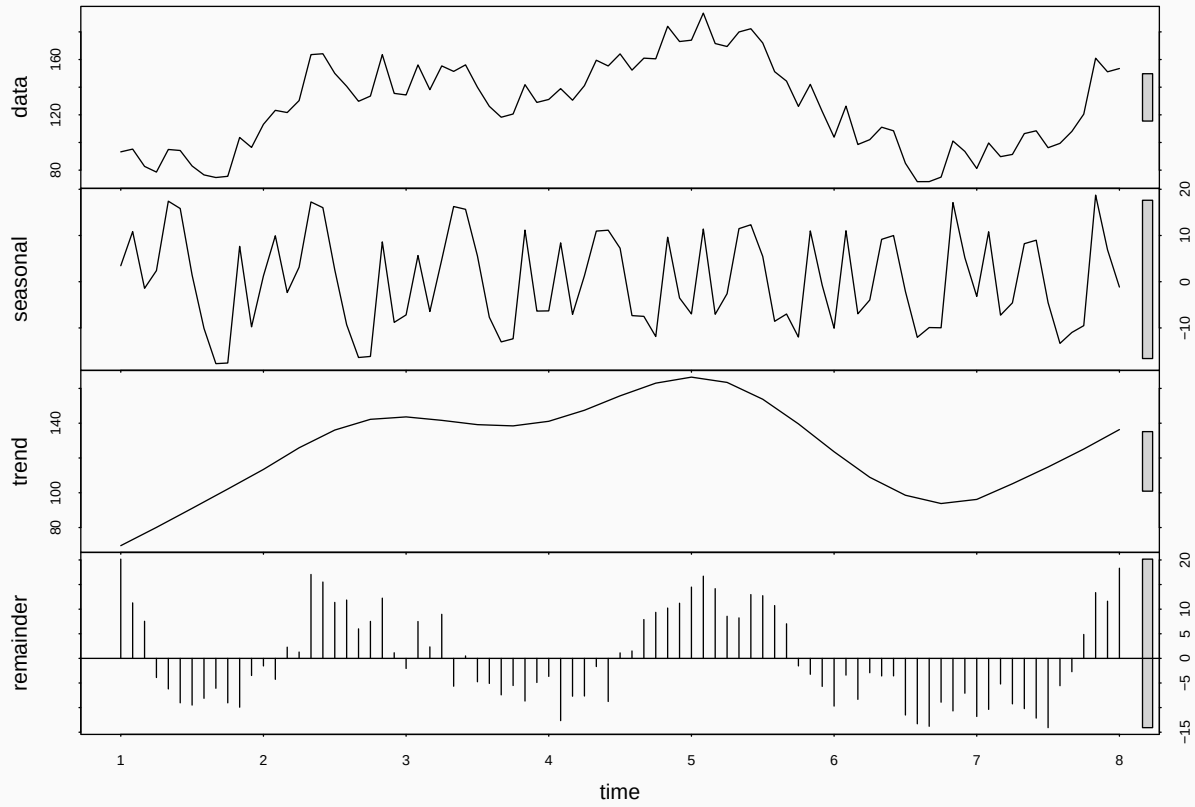
Example: Wind power forecasting



How to model trend?

Detrending

- Problem: it is not well specified what a trend is
- Essentially just a smoothed version of the series
- We still need to forecast the trend then



How to model trend? (cont'd)

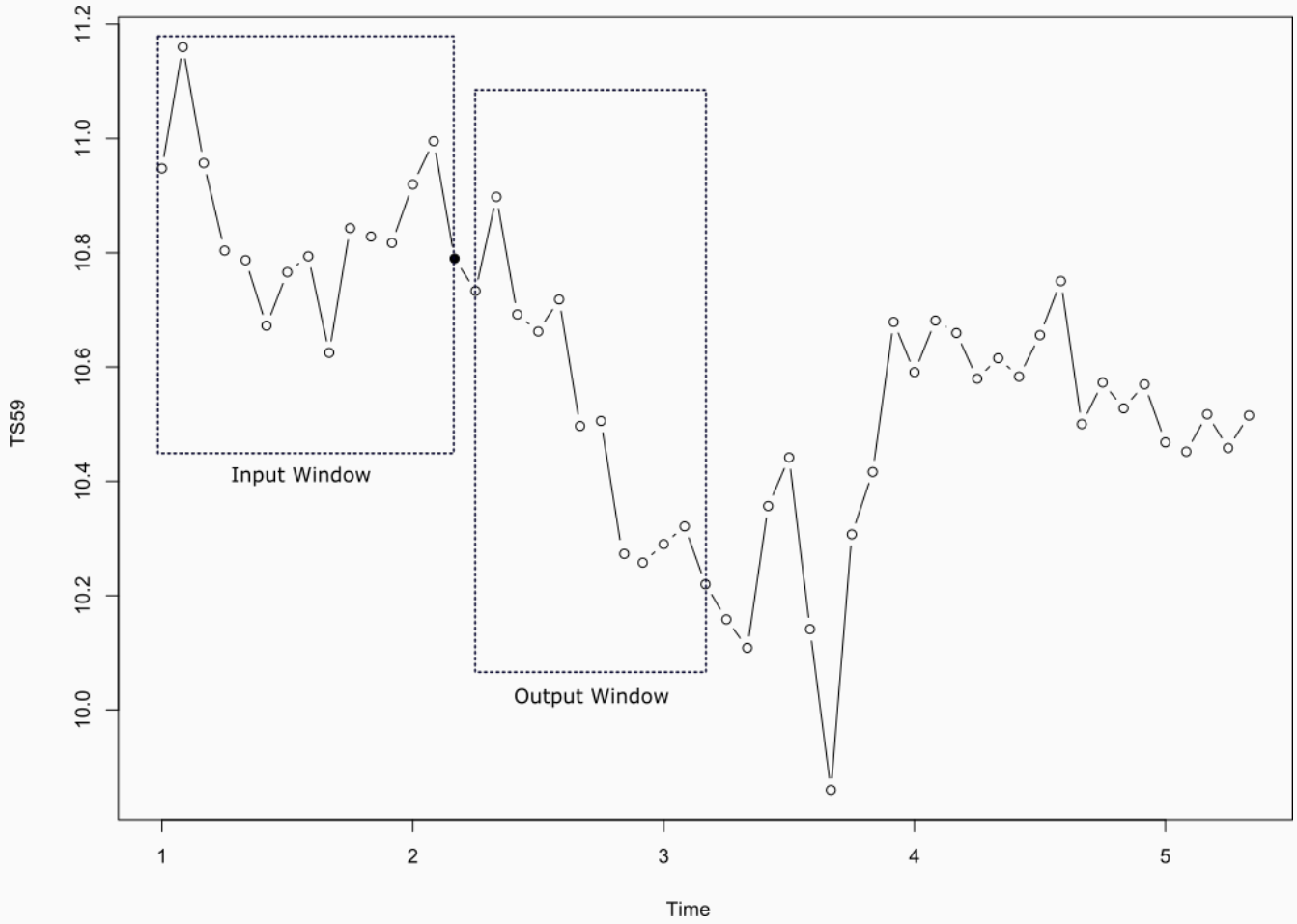
- Logarithm or Box-Cox transform
  - makes exponential trends linear
  - also stabilises the variance
  - Box-Cox Transformation

$$w_t = \begin{cases} log(y_t) & \text{if } \lambda = 0, \\ (y_t^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \end{cases}$$

  - Choice of optimal value for λ is difficult

Window-wise normalisation

Smyl and Kuber (2016);Bandara et al. (2017)

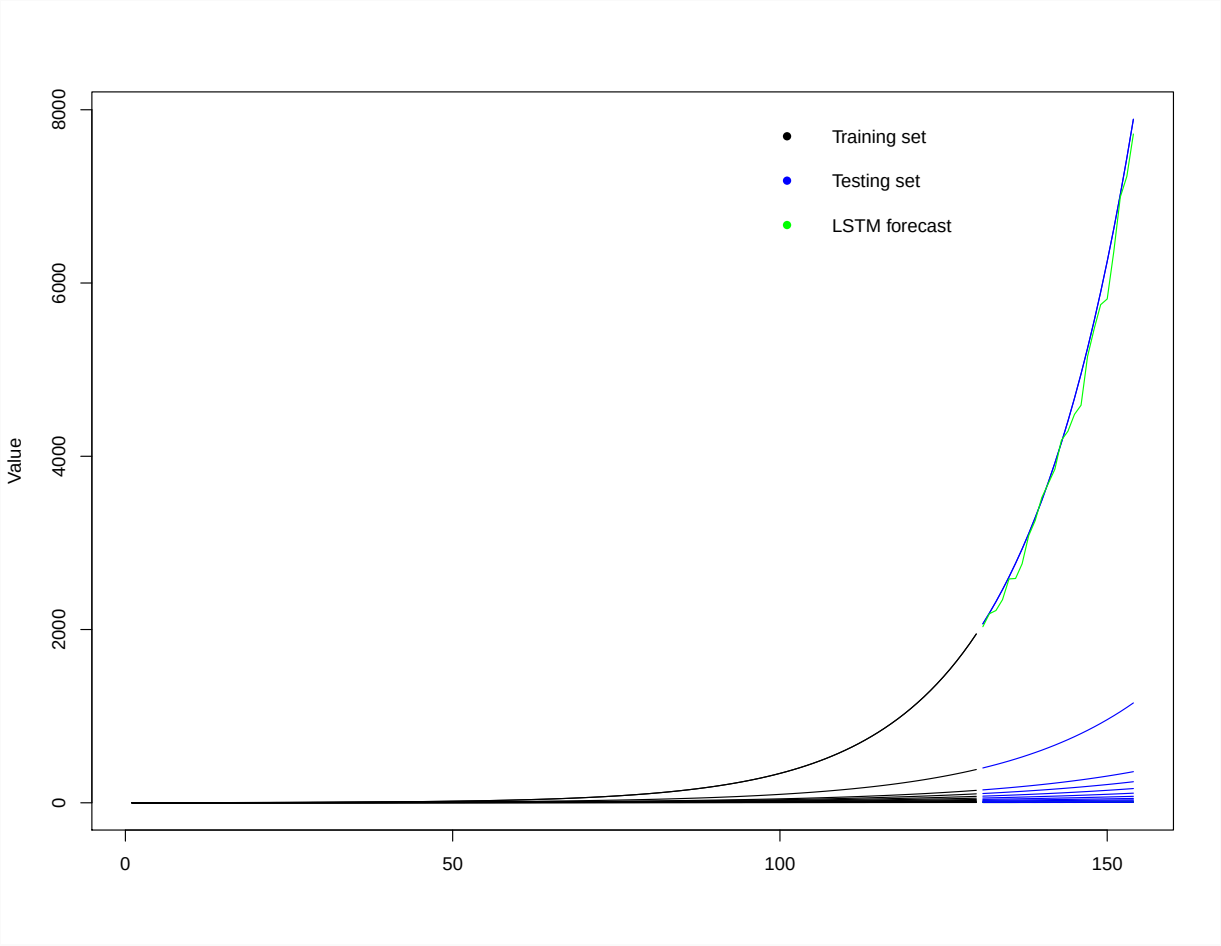


Window-wise normalisation (cont'd)

- To avoid saturation issues of sigmoid, tanh activation functions
- Similar to batch normalization
- Instead of saturating on the absolute values of the training data, it is saturating on the absolute value of the steepness of the trend in the training data
- It is usually a good idea for forecasts to be conservative.

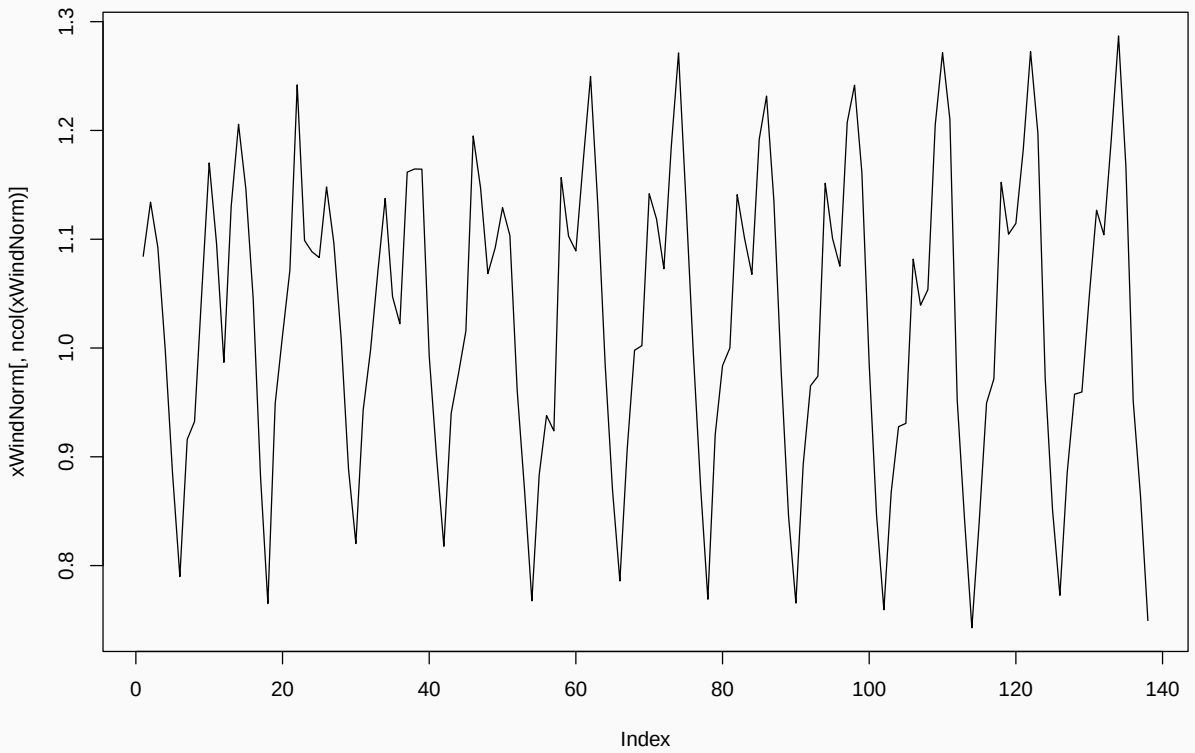
Window-wise normalisation (cont'd)

Such a system is able to predict even steep trends (Bandara et al., 2020b)



Window-wise normalisation (cont'd)

```
xEmb <- myEmb(AirPassengers, 7, 7, h=2)
xWindNorm <- t(apply(xEmb, 1, function(y) {
                    y/mean(y[1:(length(y-1))])}))
plot(xWindNorm[,ncol(xWindNorm)], type="l")
```



Expert knowledge about trends

- be careful with strong trends
- exponential trends will slow eventually
  - how much more can Facebook grow until every person on earth has 5 accounts each?
- even a linear trend is a very bold assumption oftentimes
- damped trends: often not justified from the data, just to be conservative about the forecasting
- forecasts should always be conservative

How to model seasonality?

- Some discussion in the literature whether a NN can model seasonality directly or not
- Early works suggest that NNs can model seasonality (Sharda and Patil, 1992; Tang et al., 1991).
- Later, works suggest that deseasonalization is necessary (Claveria and Torra, 2014; Zhang and Qi, 2005; Zhang and Kline, 2007; Nelson et al., 1999).
- Latest findings: Machine Learning models can model seasonality well *if they have enough data* (Bandara et al., 2020a)
- Simple experiment: Generate a sine wave, let an NN learn it. How many full periods to learn it? 2 full periods, 20 full periods?

–> Assumption in forecasting is usually that we know the seasonality beforehand and that it is valid to extrapolate it infinitly into the future.

Modelling seasonality

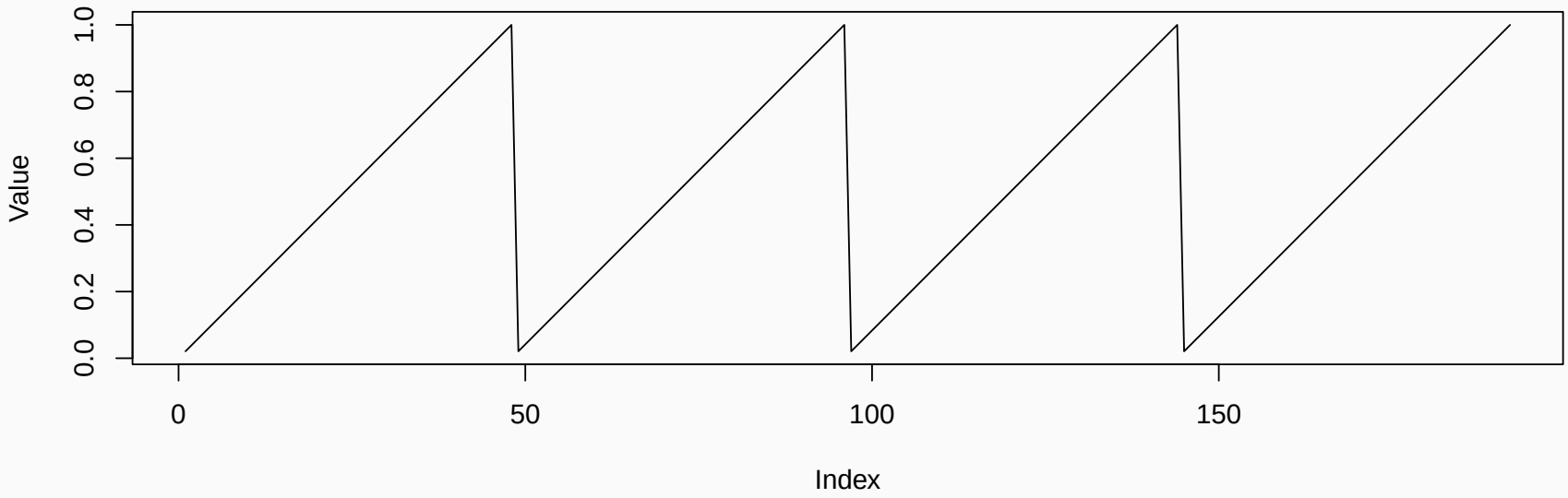
Seasonal indicator variables:

- Categorical variable: Monday, Tuesday, Wednesday, ...
- One-hot encoded version of this variable: "Seasonal dummy"

Problems if there are many seasons.

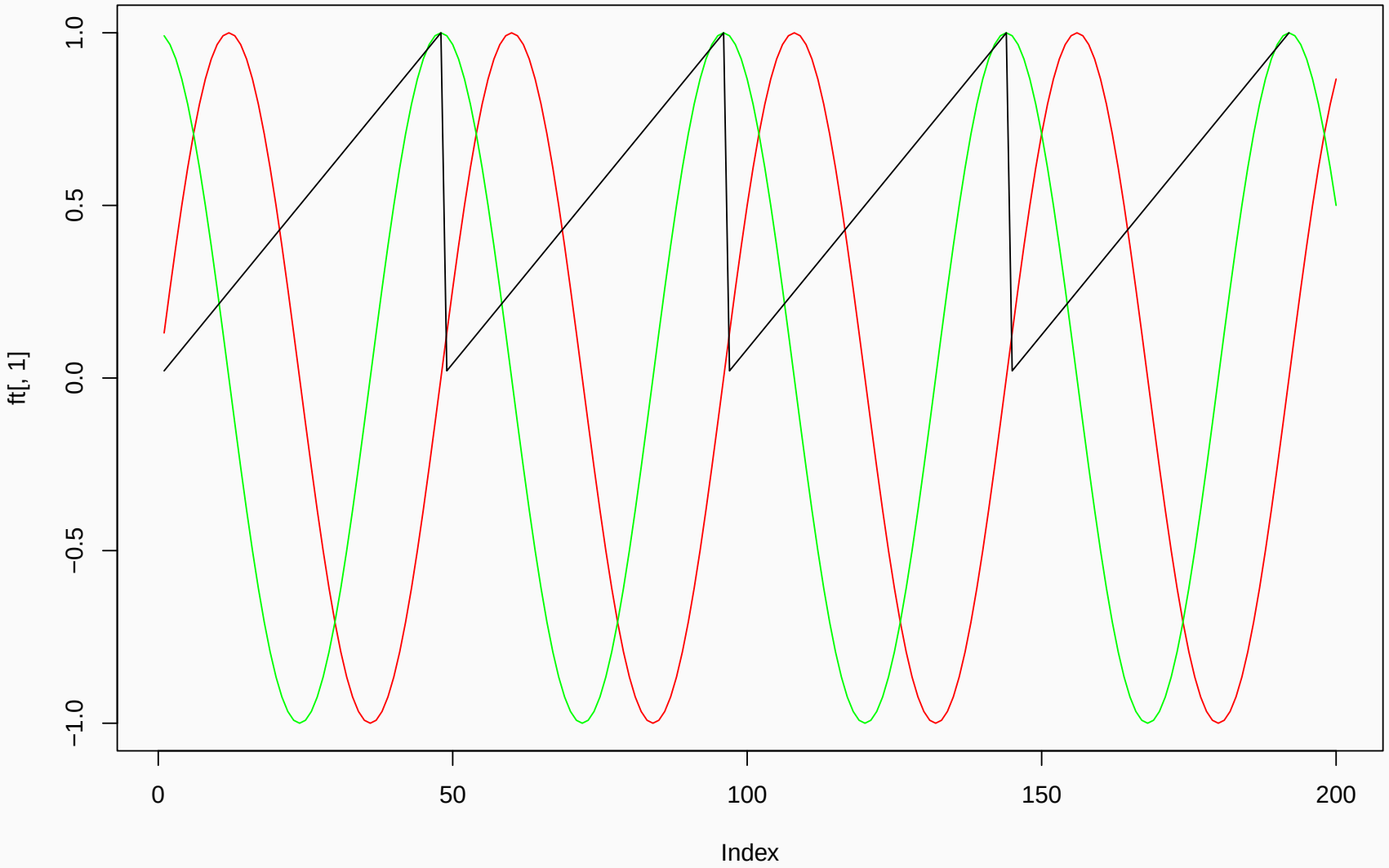
Modelling seasonality (2)

Continuous seasonal indicators



Problem: Abrupt changes. December much more distant from January than, e.g., January from February.

Modelling seasonality (3)



Fourier terms

see, e.g., Hyndman and Athanasopoulos (2018)

$$sin\left(\frac{2\pi kt}{s}\right), cos\left(\frac{2\pi kt}{s}\right)$$

t is the time point

s is the seasonal periodicity of the time series and

k is the number of sine cosine pairs used with the transformation

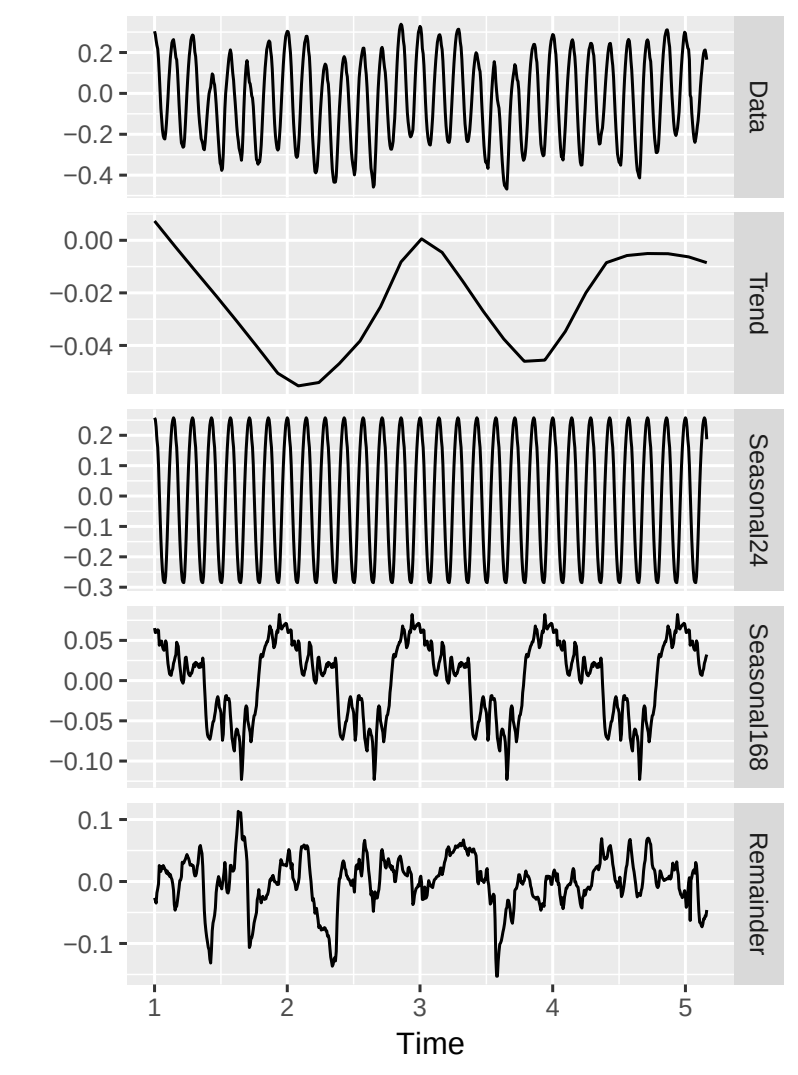
The number of Fourier terms controls the smoothness of the seasonal pattern.

Deseasonalisation

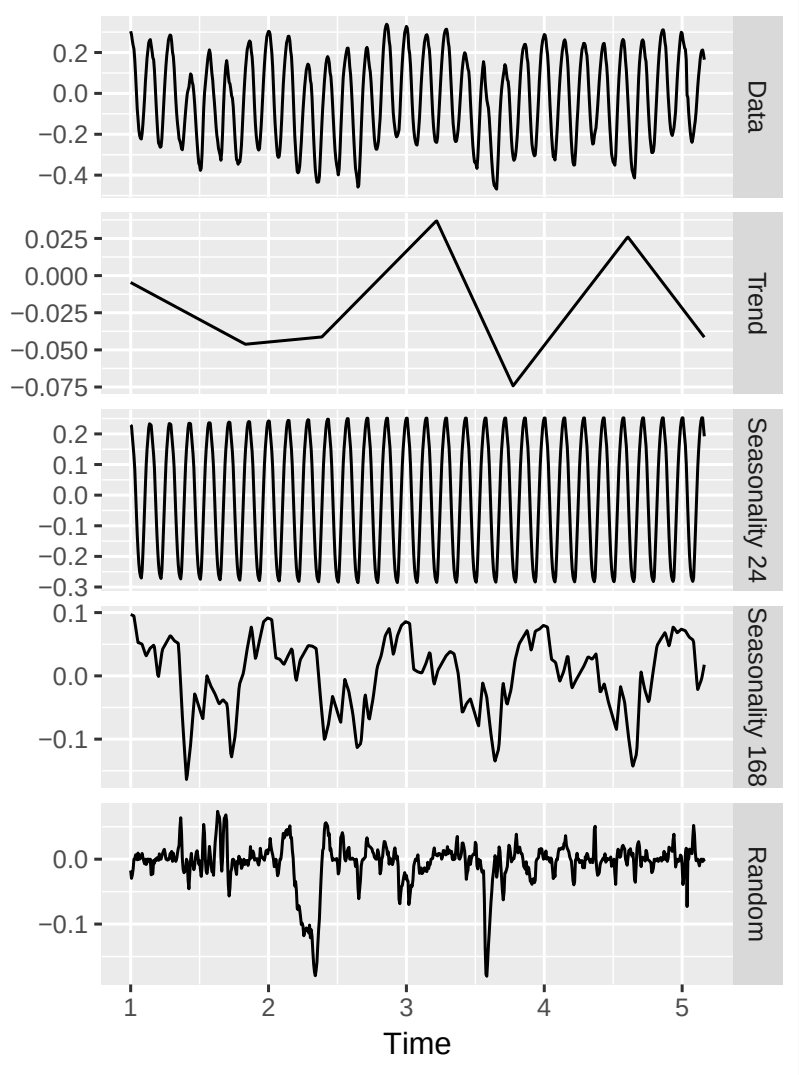
Decomposition methods such as

- STL (Cleveland et al., 1990)
- MSTL(Hyndman and Athanasopoulos, 2018)
- STR (Dokumentov and Hyndman, 2020)

Deseasonalisation (cont'd)



(a) MSTL Decomposition

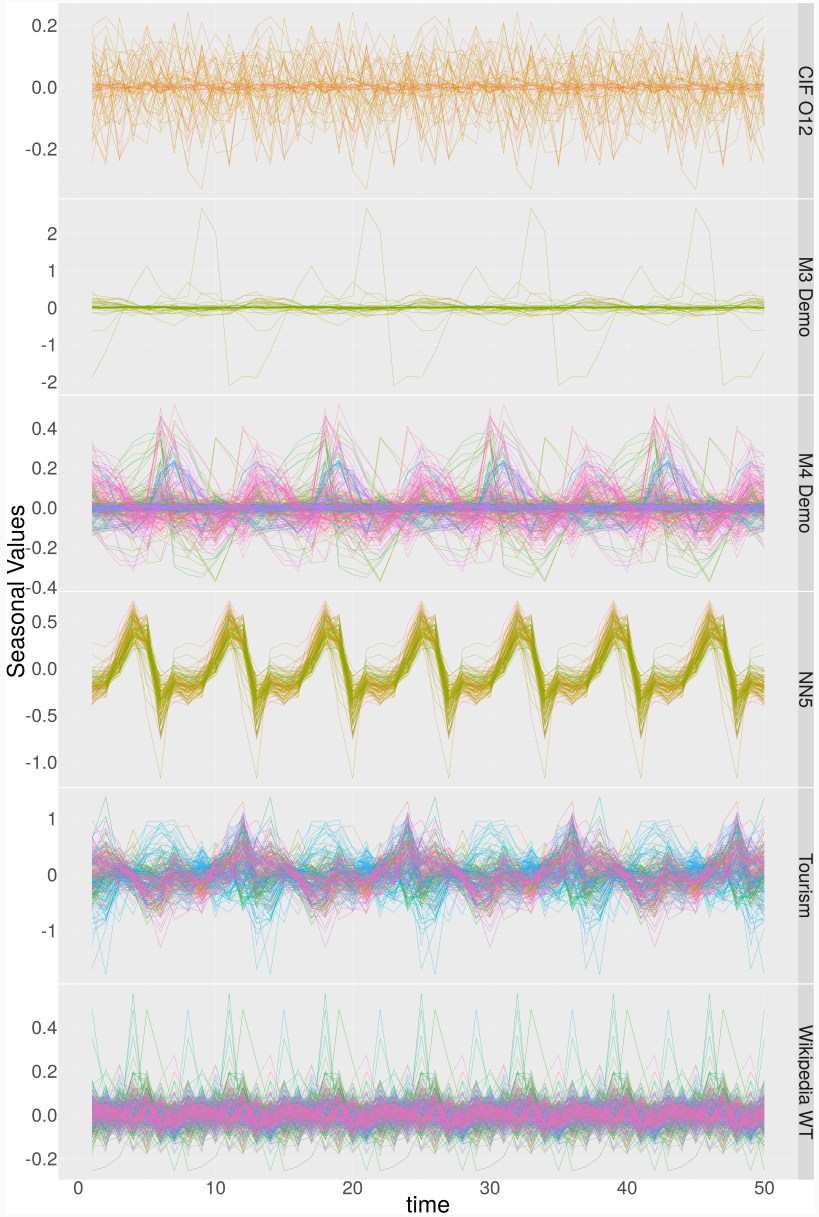


(b) STR Decomposition

Deseasonalisation (cont'd)

- we can deseasonalise and then only feed the trend and remainder component into the ML algorithm
- can be seen as a form of boosting (weak learner to model seasonality, ML model trained on residuals)
- effectively putting expert knowledge into the model
- works well if not enough data to model seasonality directly, or if seasonal components are very different between series

Deseasonalisation (cont'd)



Further considerations about seasonality

- M3, M4 datasets have different types of seasonalities mixed together, and series are not aligned
- in real-world datasets this will normally not be the case
- seasonal indicators and Fourier terms need aligned series

–> In real-world datasets, with enough data, seasonal indicators and Fourier terms will work better than deseasonalisation

Normalisation

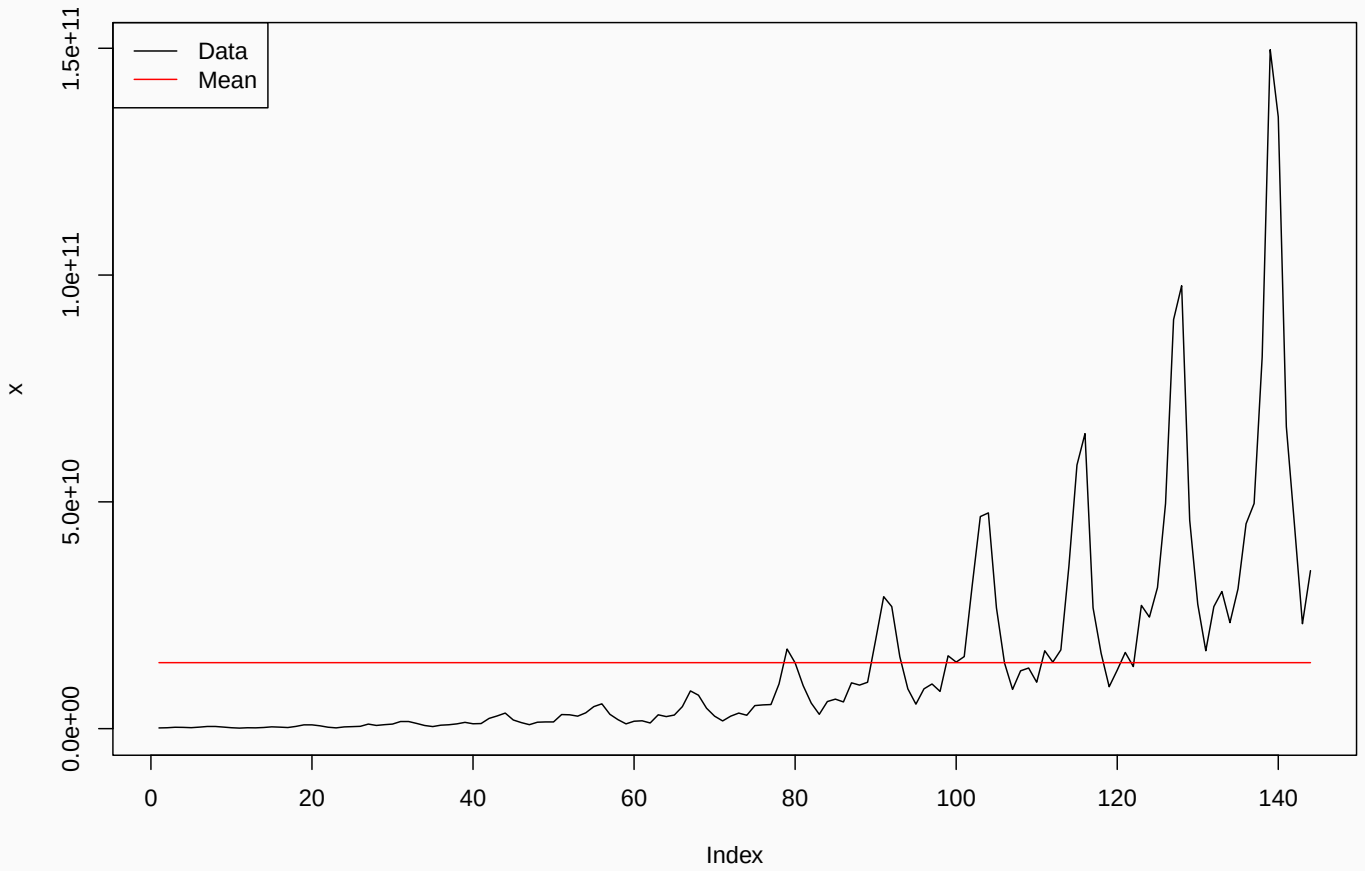
- in some forecasting problems, the forecasts are within a pre-specified domain (wind speed, wind power, electricity price)
- in others, they are not (share prices, web page hits, businesses that grow fast, like ride share applications, social networks, etc.)
- if the domain is limited, the scale has information (if you are at zero, you know the value will not be able to fall more)
- if the level is already high, trends tend to be less steep

–> include information about scale as additional input, if normalising

- error measures (sMAPE, MASE) are also often scale invariant, though this may not reflect the real world

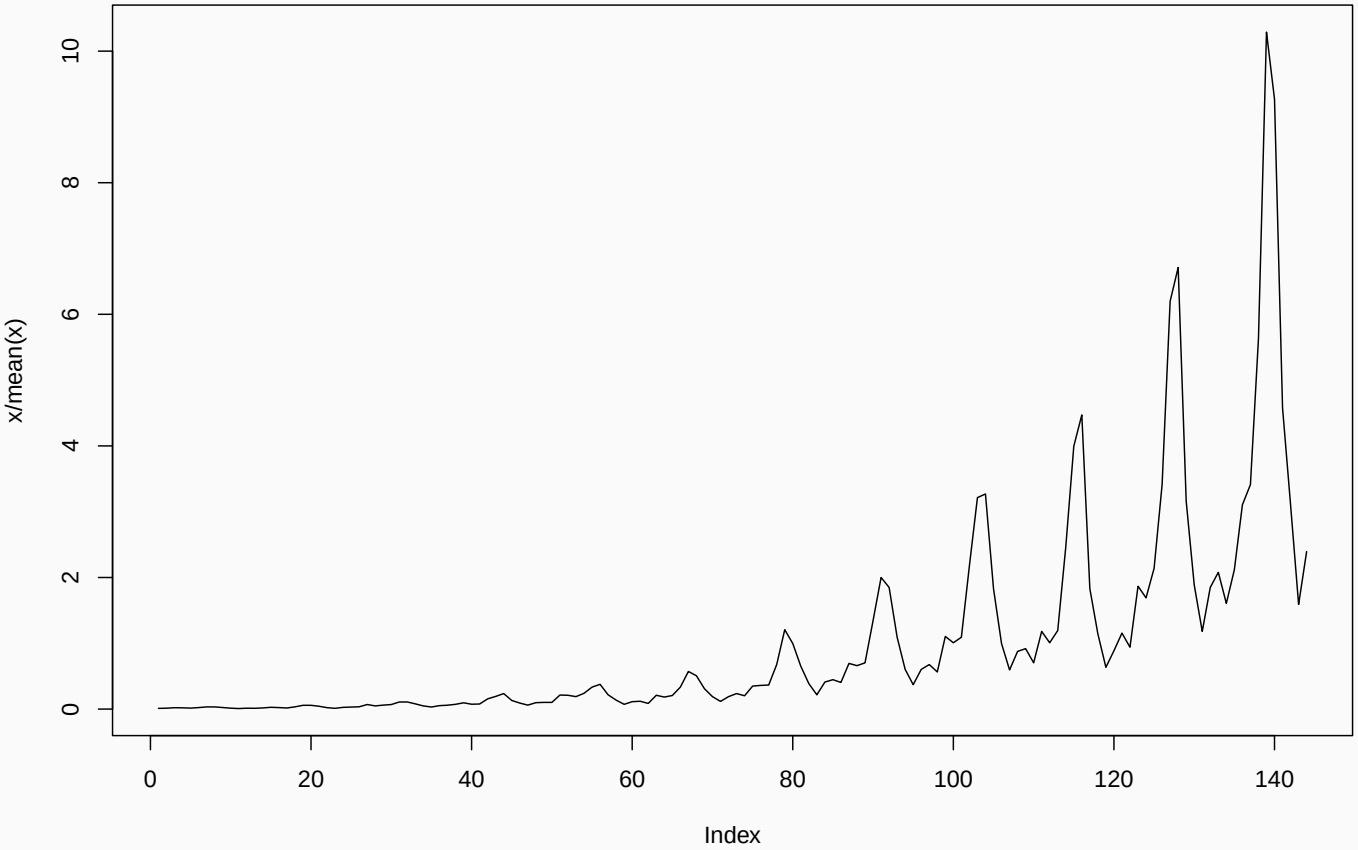
Normalisation (cont'd)

- Mean normalisation, if series don't have a strong trend
- Not a good idea if there are strong trends/level shifts:



Normalisation (cont'd)

Series divided by its mean:



Direct output versus iterative 1-step-ahead

- Traditional methods (ETS, ARIMA) optimise 1-step-ahead accuracy
- forecasts further out are obtained by iteration, by feeding back the forecasts as inputs into the model
- this can lead to error accumulation
- usually better to predict directly all horizons needed (Ben Taieb et al., 2012; Wen et al., 2017)

Direct output versus iterative 1-step-ahead (cont'd)

- methods such as NNs can have multiple outputs
  –> output windows
- other methods, such as GBT, can produce only one output
  –> either iterate out or train a separate model per horizon
- If horizon is too long, iteration may be the only feasible option
- M5 winning method was an ensemble of direct and iterative LightGBM models

Gradient boosted trees (GBT) for forecasting

- Successful in several forecasting competitions (GEFCom 2014: Landry et al. (2016), Kaggle competitions)
- M5 competition dominated by LightGBM
- Catboost (Prokhorenkova et al., 2018) performs *ordered* gradient boosting, especially suitable for time series
- Seasonality and trend handling as discussed earlier
- build one model per horizon
- especially suitable with (diverse) external variables
- engineer features such as rolling means, rolling sds
- use differences as additional inputs (of different lags, e.g., lag1 difference, lag 12 difference)

Further feature engineering

- Holiday effects
  - one-hot encoded
  - as distance maps (days before/after holiday)
- Monthly series: number of trading days in the month
- Promotions
- Out-of-stock events
- Weather
- ...

Ensembling and forecast combination

- Ensembling works in forecasting just as well as in any other area
- Heavily used, e.g., in Kaggle competitions
- Ensembles of GBTs, NNs, pooled regression
- Ensembles of local and global models

Forecast combination

- Seminal paper by Bates and Granger (1969)
- Show that combining forecasts often leads to better accuracy
- Widely accepted and adopted in the forecasting field since then
- Simple average is hard to beat, though many more sophisticated methods exist

Deep learning for forecasting

Overview: Deep learning for forecasting

- Not covered extensively here, only a brief overview
- A recent review is given by Benidis et al. (2020)
- Great recent tutorials by Amazon: Wang et al. (2020)
- Usually, recent NLP research (LSTM, attention, transformers, ...) is adapted to the time series use case
- Main differences to NLP:
  - most recent observations are the most important ones
  - long-term dependencies are relatively simple and stable (only seasonalities)

Beyond autoregression: Recurrent neural networks

- Overview by Hewamalage et al. (2020)
- Have an internal state which allows them to memorize
- They have problems to learn long memory (Pascanu et al., 2013)
- LSTM mitigates that to a certain extent
- They still work better in practice if used with input and output windows (Hewamalage et al., 2020)
  - making them more autoregressive
  - making the state less important
- Recent results suggest that in general RNNs that can be trained and are stable can be well approximated by feed-forward networks (Miller and Hardt, 2018; Miller, 2018)

Convolutional neural networks

- Some results suggest that CNNs work just as well as RNNs for forecasting, but are a lot faster to train (Borovykh et al., 2018)
- WaveNet (Oord et al., 2016; Sen et al., 2019): causal convolutions, dilations
- CNNs don't have a state, so windowing needs to cover everything

–> Long input windows, especially with dilations

–> RNNs are preferrable if input windows need to be small, e.g., across series of different length in a global model.

Specialised architectures

- DeepAR (Flunkert et al., 2017): Generative RNN model
- Deep state space models (Rangapuram et al., 2018): Parametrizes a linear state-space model with an RNN
- Deep Factors for forecasting (Wang et al., 2019): Combines a local probabilistic model and a global time series that is a linear combination of factors
- NBEATS (Oreshkin et al., 2019): Decomposes series into basis functions, residual stacking
  - State-of-the-art accuracy on M4
  - 2nd place in M5 (as part of an ensemble)

Transformers for forecasting

- Early works: Transformers for forecasting (Li et al., 2019), Temporal Fusion Transformers (TFT) (Lim et al., 2019)
- TFT addresses specifically common time series problems such as how to incorporate static and dynamic past and known future covariates into transformers, obtain prediction intervals etc.
- Reported to work well in practice in many situations
- TFT was already in existence at the time of the M5, but not used by any winning team. Maybe data of M5 was too intermittent?

Transformers for forecasting (2)

- Since 2019: A myriad of variants: Informer, Autoformer, ETSformer, Robformer, FEDformer, ...?
- Usually have plausible ideas of why they should work
- They solve "long sequence time-series forecasting,'' which to me seems somewhat an invented problem
- Many of them:
  - Evaluate on the same small amount of datasets, which only represent a small subset of forecasting use cases
  - Use an exchange rate dataset, where they fail to benchmark against naive, and in fact, we have shown that they lose (Hewamalage et al., 2023)
  - Fail to understand that their exchange rate dataset only contains trading days, which means the seasonality shifts around quite arbitrarily, due to the omission of bank holidays
  - Lose against linear models, if those are trained directly for all horizons, as the transformers are (Zeng et al., 2023)

Transformers for forecasting (3)

Zalando has reported a transformer architecture they have in production (Kunz et al., 2023). Works well, but trains on over 350k series.

–> If you have large amounts of data, Transformers should work.

–> If you have small amounts of data, there are strong competitors that will require less computation.

GluonTS and ForecastingData.org

GluonTS:

- `https://ts.gluon.ai/`
- Python library from Amazon that implements many deep-learning architectures

ForecastingData.org:

- `https://forecastingdata.org/`
- Data repository with many forecasting datasets
- Code and results of traditional and machine learning methods
- Implementations in GluonTS, forecast package in R

Multivariate forecasting

- Not the same as global models: series need to be aligned, and often have the same length (see also forecastingdata.org)
- Series can influence each other: Cannibalisation and substitution effects, etc.
- The holy grail in retail demand forecasting?
- Difficult and very active research area. Main problems:
  - How to scale methods to thousands of time series?
  - How to have series (products) come and go?
  - How to model changing and complex relationships?
- Usually sparseness constraints: Each series can interact only with few other series

Multivariate forecasting (cont'd)

- Yu et al. (2016): Matrix factorization with temporal regularization
- Lai et al. (2018), LSTNet: CNN feeding into RNN with skip connections
- Sen et al. (2019): Matrix factorization, causal convolutions and attention
- Salinas et al. (2019a): Gaussian copulas
- Wu et al. (2020), Sriramulu et al. (2023): Graph neural networks

Forecast Evaluation

Forecast Evaluation for Data Scientists

We'll have to skip this part due to time restrictions. Check out our recent papers:

- H Hewamalage, K Ackermann, and C Bergmeir. "Forecast evaluation for data scientists: common pitfalls and best practices." Data Mining and Knowledge Discovery 37.2 (2023): 788-832.
- Bergmeir, C (2023). Common Pitfalls and Better Practices in Forecast Evaluation for Data Scientists. Foresight: The International Journal of Applied Forecasting (70).
- Talk at ISF 2023 (see my web page)

Probabilistic forecasting

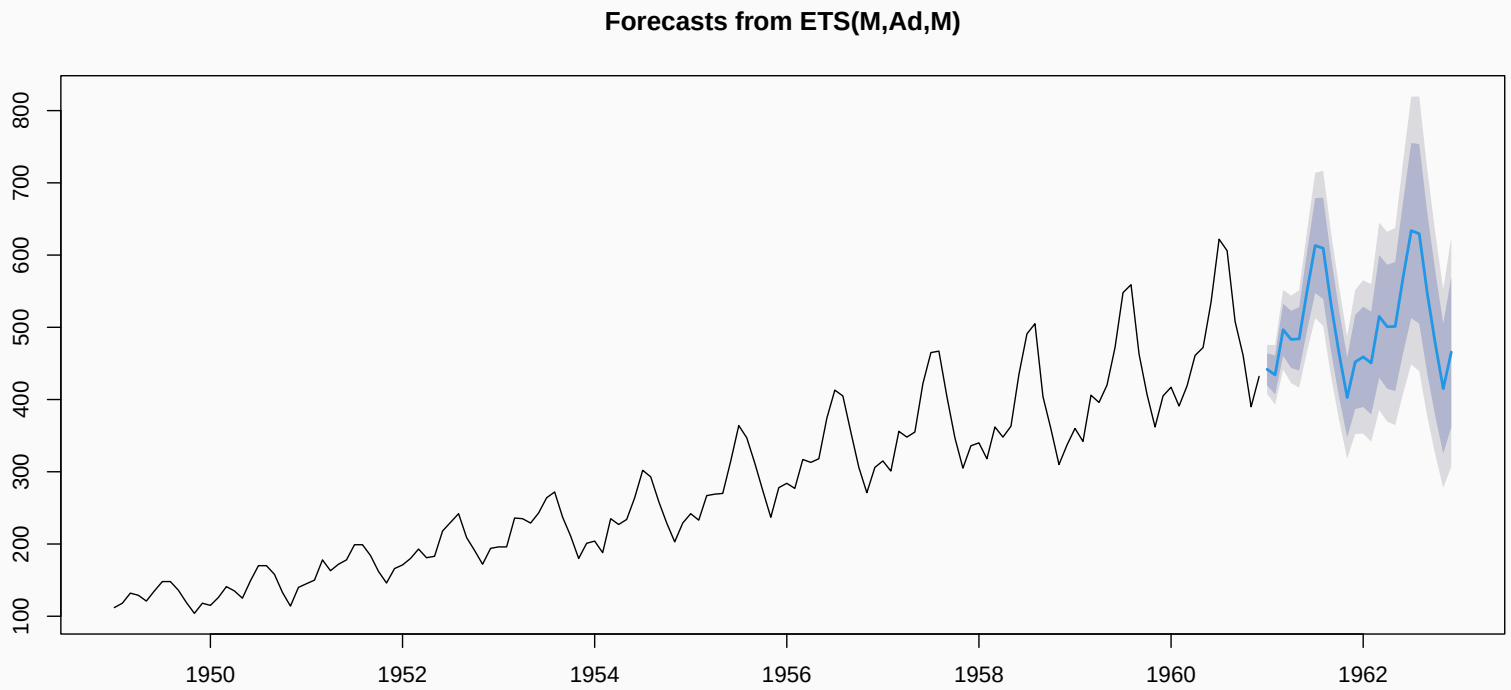
Probabilistic forecasting

Main ways for probabilistic forecasting:

- use analytical prediction intervals
- bootstrapping
- using a Bayesian model (MCMC sampling)
- forecast the parameters of a distribution
- use quantile regression (pinball loss)
- determine uncertainty empirically through backtesting (conformal prediction)
- other very recent approach: Levelset approach (Hasson et al., 2021)

Use analytical prediction intervals

- Possible for some (well-understood) models
- Usually assume normally-distributed errors
- ETS, ARIMA do this (see Hyndman and Athanasopoulos (2018))
- intervals tend to be too narrow (Bermudez et al., 2010)



Simulation and bootstrapping

- slow
- intervals can also be too narrow if, e.g., they only consider parameter uncertainty
- either need to bootstrap residuals (from an additional validation set) or assume a distribution
- then simulate forecasting paths by feeding the generated/bootstrapped values back into the model
- see Hyndman and Athanasopoulos (2018), Section "Neural Networks''

MCMC sampling

- needs to be a Bayesian model
- Examples: LGT (Smyl et al., 2019), Orbit (Ng et al., 2020), Bayesian ETS (Bermudez et al., 2010)
- slow

Determine uncertainty empirically through backtesting

- Also called conformal prediction, where it has some theory behind it
- often used by companies in practice
- leads to more realistic prediction intervals
- potentially needs a lot of past data and rolling origin forecasts (large validation sets) –> combine with cross-validation and bootstrapping schemes

Forecast the parameters of a distribution

- e.g., assuming a normal distribution: μ, σ
- DeepAR (Salinas et al., 2019b): Normal distribution and negative binomial distribution
- NGBoost (Duan et al., 2020)
- Have to assume a certain distribution
- Good if we have limited amounts of data, or knowledge of the distribution

Quantile regression (pinball loss)

- implemented in, e.g., Wen et al. (2017)
- no distribution assumptions need to be made; therewith better if a lot of data are available
- fast to compute and easy to implement
- only certain quantiles can be obtained, not the full distribution
- in practice, often 5 or 7 quantiles are enough anyway
- with, e.g., a Neural Network, we can fit different quantiles at the same time, by having multiple outputs
- can interpolate between quantiles to get full distribution (Gasthaus et al., 2019)

Pinball loss function

$$
\begin{aligned}
L_{q_t^{[u]}}(y, \hat{y}) &= (y - \hat{y}) q_t^{[u]} && \text{if } y \geq \hat{y} \\
&= (\hat{y} - y)(1 - q_t^{[u]}) && \text{if } \hat{y} > y
\end{aligned}
$$

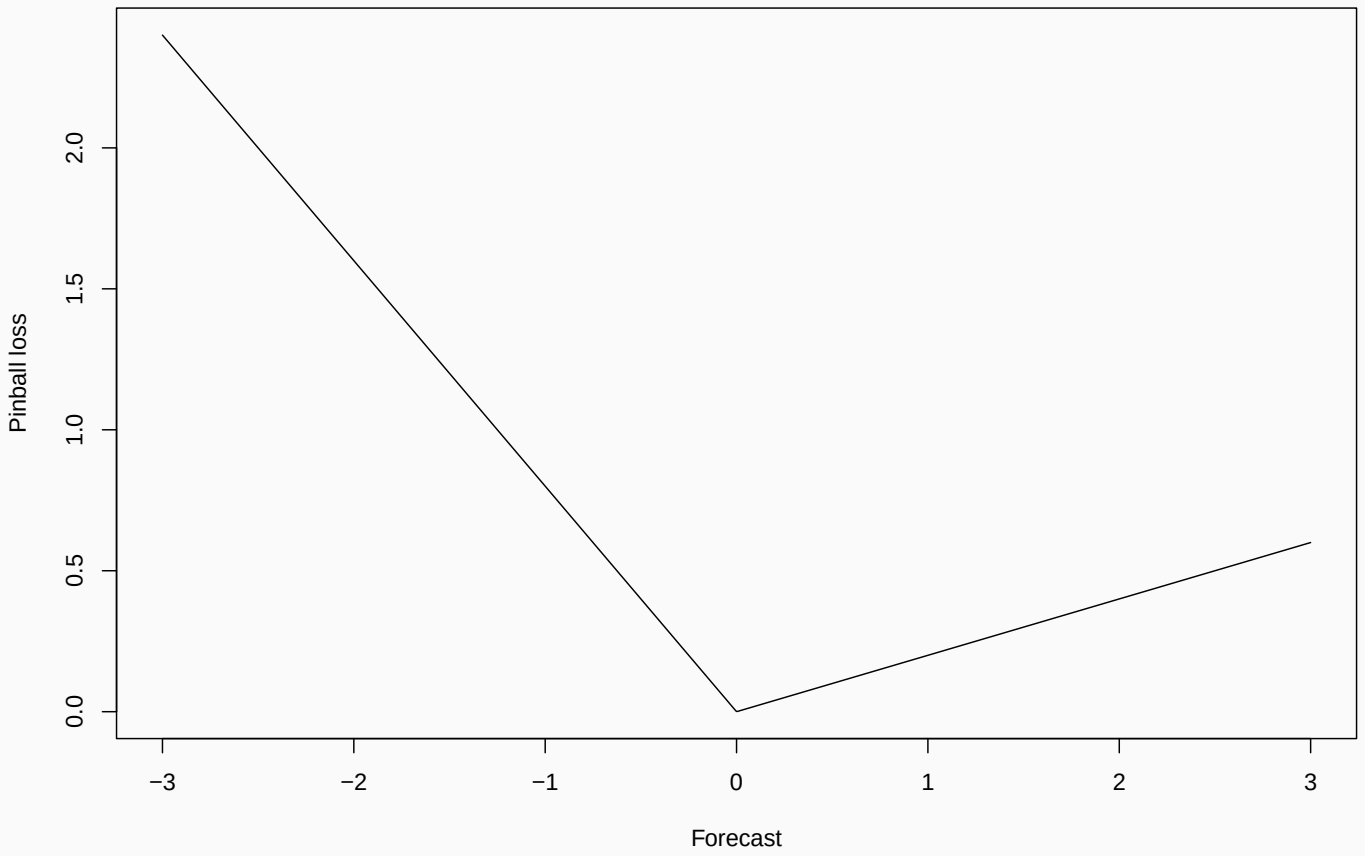
y is the true value

$\hat{y}$ is the forecast

$q_t^{[u]}$ is the target quantile, for example: $u = 0.9,\ q_t^{[0.9]}$

It can be proved that minimizing the pinball loss results in the most accurate quantile

Pinball loss (2)



Evaluation of probabilistic forecasts

Simple hit/miss rule

- can be used to evaluate forecasting interval
- directly interpretable (e.g., “80% of forecasts are within the interval”)
- doesn’t consider the magnitude of the error
- Problem: trivial solutions are possible by grossly over- and underpredicting certain amounts of times

Mean Scaled Interval Score (MSIS)

$$MSIS = \frac{1}{h} \times \frac{\sum_{t=n+1}^{n+h} \left(q_t^{[u]} - q_t^{[l]} + \frac{2}{\alpha}(q_t^{[l]} - y_t)\mathbb{1}_{y_t < q_t^{[l]}} + \frac{2}{\alpha}(y_t - q_t^{[u]})\mathbb{1}_{y_t > q_t^{[u]}} \right)}{\frac{1}{n-m}\sum_{t=m+1}^{n} |y_t - y_{t-m}|}$$

Mean Scaled Interval Score (MSIS) (2)

- used in the M4 competition (Makridakis et al., 2018a)
- evaluates a prediction interval
- sum over the size of the intervals and the magnitude of error for points that lie outside of the interval
- If model is optimised for pinball loss, it will not necessarily perform well under MSIS (Smyl, 2020)

Scaled Pinball Loss (SPL)

$$SPL[u] = \frac{1}{h} \frac{\sum_{t=n+1}^{n+h} (u(y_t - q_t^{[u]}) \mathbb{1}_{q_t^{[u]} \leq y_t} + (1-u)(q_t^{[u]} - y_t) \mathbb{1}_{q_t^{[u]} > y_t})}{\frac{1}{n-m} \sum_{t=m+1}^{n} |y_t - y_{t-m}|}$$

- 90th quantile forecast: $u = 0.9$, $q_t^{[0.9]}$
- 10th quantile forecast: $u = 0.1$, $q_t^{[0.1]}$
- $\mathbb{1}$ is the indicator function
- n is the time index of the last observation in the training set
- h is the forecast horizon

Weighted Scaled Pinball Loss (WSPL)

- used in the M5 uncertainty track
- Combines, e.g., each series' $SPL[0.1]$ and $SPL[0.9]$ together by taking the average
- Then (optionally) weighs SPL by a series-specific weight.

$$WSPL = \sum_{i=1}^{n} w_i \times \frac{1}{k} \sum_{j=1}^{k} SPL[u_k]$$

- k represents the number of quantiles
- We can set w_i to $\frac{1}{n}$ for all n time series to weight them equally
- Lower WSPL scores indicate more precise forecasts

Evaluation of probabilistic forecasts (cont'd)

- Overviews by Gneiting and Raftery (2007);Jordan et al. (2017)
- Continuous Ranked Probability Score (CRPS)
  - evaluates a full distribution
  - integrates over all quantiles
  - generalises the MAE to the probabilistic case
- logarithmic score, energy score, variogram score

Special forecasting problems

Topics

- External variables
- Intermittent data
- Hierarchical forecasting
- Forecasting in retail
- Interpretability
- Causal Inference

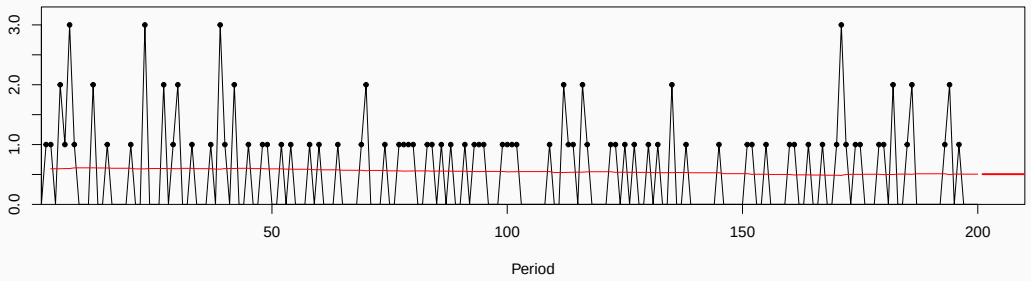
–> we'll briefly look at intermittent data and hierarchical forecasting

Intermittent data

Croston's method

Croston (1972):

- Build two time series
- Series 1: Omit all zeros, only model demand
- Series 2: Count wait time between events (amount of zeros)
- Predict both series with SES
- Forecast: Prediction series 1 / Prediction series 2



Other models for intermittent data

- Zero-inflated models
  - One model that predicts the probability for a zero
  - A second model that predicts a value under the assumption that it is non-zero
- Specialised loss functions
  - negative binomial loss function (used in DeepAR)
  - Poisson loss function
  - Tweedie loss function (Zhou et al., 2020)
- Other recent approaches: Türkmen et al. (2021)
- It seems that different models work better, depending on the degree of intermittency. Rule of thumb:
  - With $>$90% zeros, better to use a zero-inflated model
  - Otherwise, an adapted loss function

Hierarchical forecasting

Hierarchical forecasting

- Very common problem in forecasting
- Forecasts need to add up
  - Spatially: across stores, distribution centres, countries, ...
  - Product categories, store departments, ...
  - Temporally: Weekly, monthly, yearly forecasts
- Recent overview by Hyndman (2020)

Hierarchical forecasting: Classic approaches

- top down
  - predict the series at the top
  - disaggregate
  - Problem: How to disaggregate?
  - According to ratios from the past, train a model to disaggregate, etc.
- bottom up
  - predict series at the bottom
  - aggregate
  - Problem: Bottom-level series often have a lot of noise, no clear patterns of seasonality and trend
- middle out
  - predict series at a middle level
  - aggregate and disaggregate

Hierarchical forecasting: Illustration

Hierarchical forecasting: Optimal reconciliation

- Introduced by Hyndman et al. (2011)
- Forecast all series separately
- Make them consistent with a reconciliation step, least squares optimisation
- Leads to more accurate forecasts than the classical approaches
- A lot of research and theoretical insights in the forecasting literature since then (e.g., trace minimisation by Wickramasuriya et al. (2019))
- Geometric view: Forecasts must lie on a coherent subspace where linear constraints hold (Panagiotelis et al., 2020)

Probabilistic hierarchical forecasting

- What does "forecasts add up correctly" mean for a probabilistic forecast?
- Panagiotelis et al. (2020): Multivariate density must lie on a coherent subspace
- Some work in the machine learning community (Ben Taieb et al., 2017, 2020)

Forecasting and reconciliation in a single step

- Recent ML methods that directly produce reconciled forecasts in one step, as opposed to the current approach of forecasting and then reconciling
- Rangapuram et al. (2021): End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series (Hier-E2E)
- Directly trains towards reconciled forecasts, with an according loss function
- forecasts are not guaranteed to be consistent (as inconsistence is only penalised)
- can make forecasts consistent by a final bottom-up step
- In GluonTS (also contains a wrapper for hts in Python)
- `https://github.com/rshyamsundar/gluonts-hierarchical-ICML-2021`
- Similar approach: Han et al. (2021)

Conclusions

- Forecasting as a field has come a long way
- Great time to do forecasting as a Machine Learner or Data Scientist
- Machine Learning methods become more and more competitive
- Be aware of some of the common pitfalls of forecasting, such as evaluation, benchmarks, data-leakage, non-stationarity
- Happy forecasting!

Thank You

https://www.cbergmeir.com

christoph.bergmeir@monash.edu

References i

K. Bandara, C. Bergmeir, and S. Smyl. Forecasting across time series databases using long short-term memory networks on groups of similar series. *arXiv preprint arXiv:1710.03222*, 8:805–815, 2017.

K. Bandara, C. Bergmeir, and H. Hewamalage. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Transactions on Neural Networks and Learning Systems*, (forthcoming), 2020a. URL `https://arxiv.org/pdf/1909.04293`.

References ii

K. Bandara, C. Bergmeir, and S. Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140:112896, 2020b.

J. M. Bates and C. W. Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969.

References iii

S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Syst. Appl.*, 39(8):7067–7083, June 2012.

S. Ben Taieb, J. W. Taylor, and R. J. Hyndman. Coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pages 3348–3357, 2017.

References iv

S. Ben Taieb, J. W. Taylor, and R. J. Hyndman. Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, pages 1–17, 2020.

K. Benidis, S. S. Rangapuram, V. Flunkert, B. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, et al. Neural forecasting: Introduction and literature overview. *arXiv preprint arXiv:2004.10240*, 2020.

References v

J. Bermudez, J. Segura, and E. Vercher. Bayesian forecasting with the holt-winters model. *Journal of the Operational Research Society*, 61(1):164–171, 2010.

C. S. Bojer and J. P. Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 2020.

A. Borovykh, S. Bohte, and C. W. Oosterlee. Dilated convolutional neural networks for time series forecasting. *Journal of Computational Finance*, 2018. doi: 10.21314/jcf.2019.358. URL `https://doi.org/10.21314/jcf.2019.358`.

References vi

G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

O. Claveria and S. Torra. Forecasting tourism demand to catalonia: Neural networks vs. time series models. *Econ. Model.*, 36:220–228, Jan. 2014.

R. B. Cleveland, W. S. Cleveland, J. McRae, and I. Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6:3–73, 1990.

J. D. Croston. Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23 (3):289–303, 1972.

References vii

A. Dokumentov and R. J. Hyndman. Str: A seasonal-trend decomposition procedure based on regression. *arXiv preprint arXiv:2009.05894*, 2020.

T. Duan, A. Anand, D. Y. Ding, K. K. Thai, S. Basu, A. Ng, and A. Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*, pages 2690–2700. PMLR, 2020.

G. T. Duncan, W. L. Gorr, and J. Szczypula. Forecasting analogous time series. In *Principles of forecasting*, pages 195–213. Springer, 2001.

References viii

V. Flunkert, D. Salinas, and J. Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *CoRR*, abs/1704.04110, 2017. URL `http://arxiv.org/abs/1704.04110`.

A. Garza and M. Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski. Probabilistic forecasting with spline quantile function rnns. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1901–1910, 2019.

References ix

A. Gelman, P. Gelman, and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Analytical Methods for Social Research. Cambridge University Press, 2007. ISBN 9780521686891.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

P. Goodwin. The Holt-Winters approach to exponential smoothing: 50 years old and going strong. *Foresight: The International Journal of Applied Forecasting*, 19:30–33, 2010.

References x

X. Han, S. Dasgupta, and J. Ghosh. Simultaneously reconciled quantile forecasting of hierarchically related time series. In *International Conference on Artificial Intelligence and Statistics*, pages 190–198. PMLR, 2021.

H. Hasson, B. Wang, T. Januschowski, and J. Gasthaus. Probabilistic forecasting: A level-set approach. *Advances in Neural Information Processing Systems*, 34, 2021.

References xi

H. Hewamalage, C. Bergmeir, and K. Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, (forthcoming), 2020. URL `https://arxiv.org/pdf/1909.00590`.

H. Hewamalage, K. Ackermann, and C. Bergmeir. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2): 788–832, 2023.

References xii

N. Hollmann, S. Müller, K. Eggensperger, and F. Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

R. Hyndman, A. Koehler, R. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002.

References xiii

R. J. Hyndman. Ten years of forecast reconciliation, 2020. URL `https://robjhyndman.com/seminars/reconciliation_review_talk/`. Accessed 10 November 2020.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice, 2nd edition*. OTexts, otexts.com/fpp2, 2018.

R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9): 2579–2589, 2011.

References xiv

T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, and L. Callot. Criteria for classifying forecasting methods. *International Journal of Forecasting*, 36(1):167–177, 2020.

A. Jordan, F. Krüger, and S. Lerch. Evaluating probabilistic forecasts with scoringrules. *arXiv preprint arXiv:1709.04743*, 2017.

References xv

M. Kunz, S. Birr, M. Raslan, L. Ma, Z. Li, A. Gouttes, M. Koren, T. Naghibi, J. Stephan, M. Bulycheva, et al. Deep learning based forecasting: a case study from the online fashion industry. *arXiv preprint arXiv:2305.14406*, 2023.

G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long- and Short-Term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, pages 95–104, New York, NY, USA, 2018. ACM.

References xvi

M. Landry, T. P. Erlinger, D. Patschke, and C. Varrichio. Probabilistic gradient boosting machines for gefcom2014 wind forecasting. *International Journal of Forecasting*, 32 (3):1061–1066, 2016.

S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*, pages 5243–5253, 2019.

References xvii

B. Lim, S. O. Arik, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*, 2019.

S. Makridakis and M. Hibon. The M3-competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018a.

References xviii

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018b.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m5 accuracy competition: Results, findings and conclusions. 10 2020.

J. Miller. When recurrent models don't need to be recurrent, 2018. URL `http://www.offconvex.org/2018/07/27/approximating-recurrent/`. Accessed 10 November 2020.

References xix

J. Miller and M. Hardt. Stable recurrent models. *arXiv preprint arXiv:1805.10369*, 2018.

P. Montero-Manso and R. J. Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality. *arXiv preprint arXiv:2008.00444*, 2020.

M. Nelson, T. Hill, W. Remus, and M. O'Connor. Time series forecasting using neural networks: should the data be deseasonalized first? *J. Forecast.*, 18(5):359–367, 1999.

E. Ng, Z. Wang, H. Chen, S. Yang, and S. Smyl. Orbit: Probabilistic forecast with exponential smoothing. *arXiv preprint arXiv:2004.08492*, 2020.

References xx

A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

References xxi

A. Panagiotelis, P. Gamakumara, G. Athanasopoulos, R. Hyndman, et al. Probabilistic forecast reconciliation: Properties, evaluation and score optimisation. Technical report, Monash University, Department of Econometrics and Business Statistics, 2020.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

References xxii

L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pages 6638–6648, 2018.

S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. In *Advances in neural information processing systems*, pages 7785–7794, 2018.

References xxiii

S. S. Rangapuram, L. D. Werner, K. Benidis, P. Mercado, J. Gasthaus, and T. Januschowski. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pages 8832–8843. PMLR, 2021.

D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. In *Advances in Neural Information Processing Systems*, pages 6827–6837, 2019a.

References xxiv

D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2019b. ISSN 0169-2070.

R. Sen, H.-F. Yu, and I. S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems*, pages 4837–4846, 2019.

R. Sharda and R. B. Patil. Connectionist approach to time series prediction: an empirical test. *J. Intell. Manuf.*, 3(5): 317–323, Oct. 1992.

References xxv

S. Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

S. Smyl and K. Kuber. Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks. In *36th International Symposium on Forecasting*, 2016.

References xxvi

S. Smyl, C. Bergmeir, E. Wibowo, and T. W. Ng. *Rlgt: Bayesian Exponential Smoothing Models with Trend Modifications*, 2019. URL `https://CRAN.R-project.org/package=Rlgt`. R package version 0.1-3.

A. Sriramulu, N. Fourrier, and C. Bergmeir. Adaptive dependency learning graph neural networks. *Information Sciences*, 625:700–714, 2023.

References xxvii

M. Štěpnička and M. Burda. Computational intelligence in forecasting (CIF) 2016 time series forecasting competition. In *IEEE WCCI 2016, JCNN-13 Advances in Computational Intelligence for Applied Time Series Forecasting (ACIATSF)*, 2016.

Z. Tang, C. de Almeida, and P. A. Fishwick. Time series forecasting using neural networks vs. box- jenkins methodology. *Simulation*, 57(5):303–310, Nov. 1991.

References xxviii

J. R. Trapero, N. Kourentzes, and R. Fildes. On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society*, 66(2):299–307, Feb 2015. ISSN 1476-9360. doi: 10.1057/jors.2013.174.

A. C. Türkmen, T. Januschowski, Y. Wang, and A. T. Cemgil. Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. *Plos one*, 16(11):e0259764, 2021.

Y. Wang, A. Smola, D. C. Maddix, J. Gasthaus, D. Foster, and T. Januschowski. Deep factors for forecasting. *arXiv preprint arXiv:1905.12417*, 2019.

Y. Wang, C. Faloutsos, V. Flunkert, J. Gasthaus, and T. Januschowski. Forecasting big time series: theory and practice. In *The Web Conference*, 2020. URL `https://www.amazon.science/videos-and-tutorials/forecasting-big-time-series-theory-and-practice`.

References xxx

R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka. A Multi-Horizon quantile recurrent forecaster. In *Neural Information Processing Systems*, Nov. 2017.

S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. *arXiv preprint arXiv:2005.11650*, 2020.

References xxxi

H.-F. Yu, N. Rao, and I. S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.

A. Zeng, M. Chen, L. Zhang, and Q. Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

G. P. Zhang and D. M. Kline. Quarterly Time-Series forecasting with neural networks. *IEEE Trans. Neural Netw.*, 18(6):1800–1814, Nov. 2007.

References xxxii

G. P. Zhang and M. Qi. Neural network forecasting for seasonal and trend time series. *Eur. J. Oper. Res.*, 160(2): 501–514, 2005.

H. Zhou, W. Qian, and Y. Yang. Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Communications in Statistics-Simulation and Computation*, pages 1–23, 2020.