Modern Time Series Analysis

ttps://github.com/AileenNielsen/TimeSeriesAnalysisWithPython

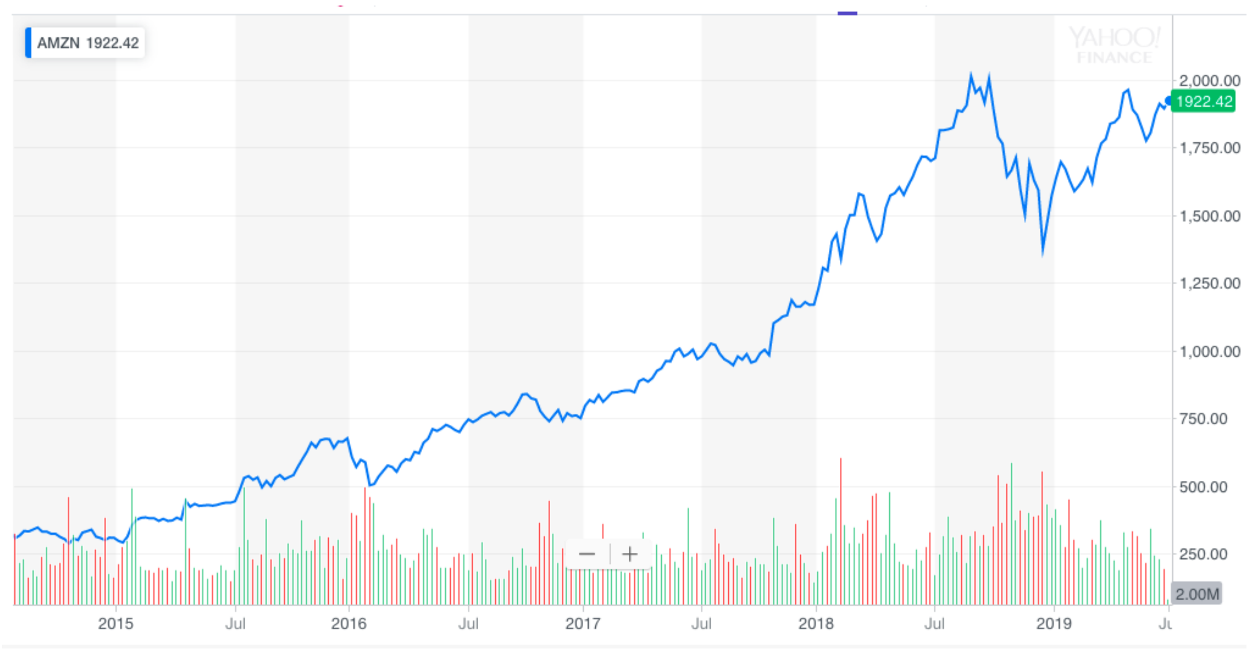
1. Dates & Times.ipynb
2. Time Zone Handling.ipynb
3. Reading in data and making sensible data frames.ipynb
4. Resampling.ipynb
5. Moving Window Functions.ipynb
6. Trend & Seasonality.ipynb
7. Forecasting.ipynb
8. Spectral Analysis.ipynb
9. Clustering & Classification.ipynb

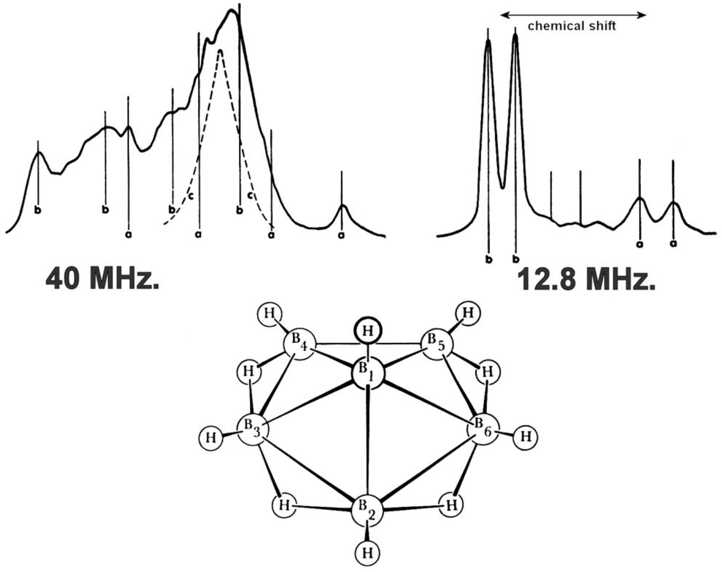
Outline

- Brief overview of time series
- State space models for time series
- Machine learning methods for time series
- Deep learning for time series

Time series generally

What are time series?





finance.yahoo.com
arstechnica.com/gadgets/2018/12/an-ecg-on-your-wrist-apples-new-heart-monitoring-software/
en.wikipedia.org/wiki/Nuclear_magnetic_resonance_spectroscopy

Tasks for time series analysis

- Visualization and exploratory data analysis
  - Understanding temporal behavior of data: **seasonality, stationarity**
  - Identifying underlying distributions and nature of temporal process producing data
- Estimation of past, present, and future values
  - **filtering vs forecasting**
- **Classification** of time series
- **Anomaly detection** of outlier points within time series

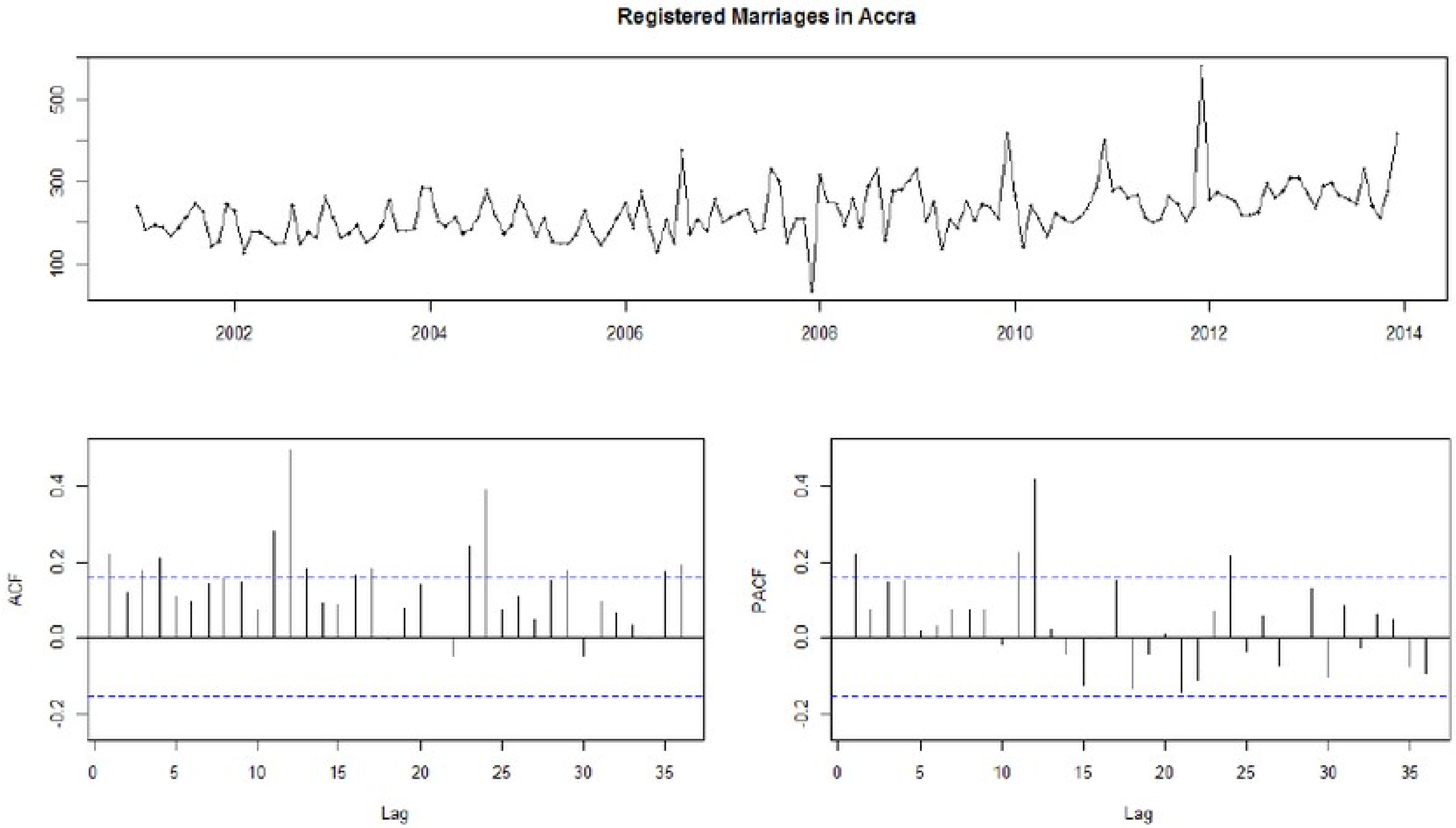
Time series data versus cross-sectional data

- More opportunities for missing data points
  - Quite challenging following the same samples (or people) through time
- Often, there is a high degree of correlation between data points
  - Values in the past predict values in the future
  - This is good for prediction but bad for models that assume independent inputs
- Time stamps or other measures of distance travelled along the temporal axis introduce all kinds of data messiness
  - Time zones, frequency irregularities, etc

Characteristics of time series data

- Data is collected sequentially; one axis is monotonically increasing
- Structure is characterized across data points:
  - Seasonality & cycles
  - Autocorrelation
  - Trends
- Stochastic behavior even as to behavioral regime
  - Change points / regime shifts v. drift / gradual change

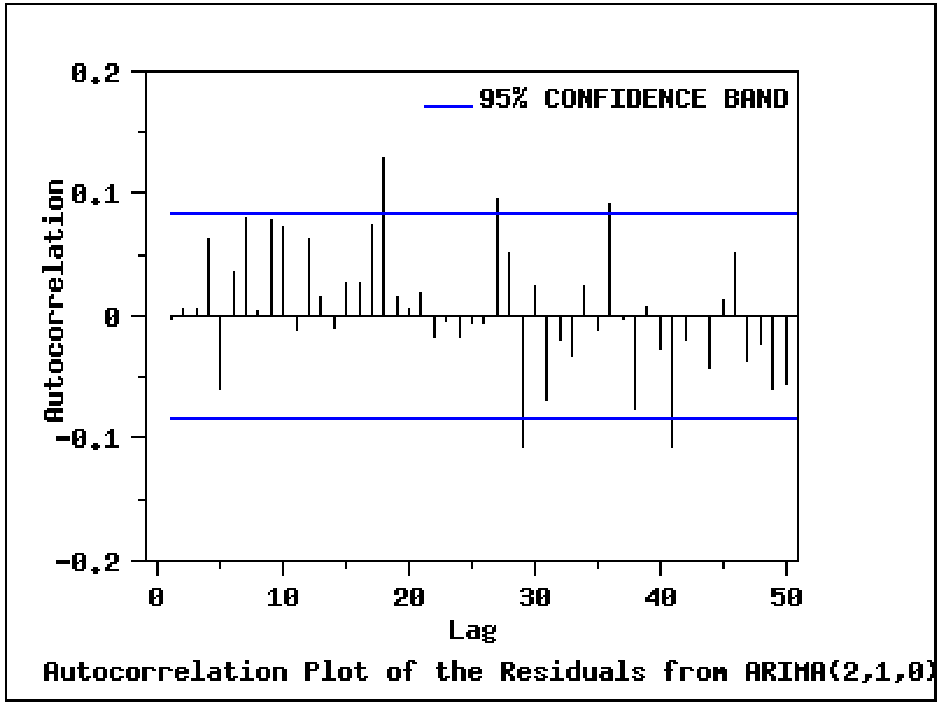
Autocorrelation



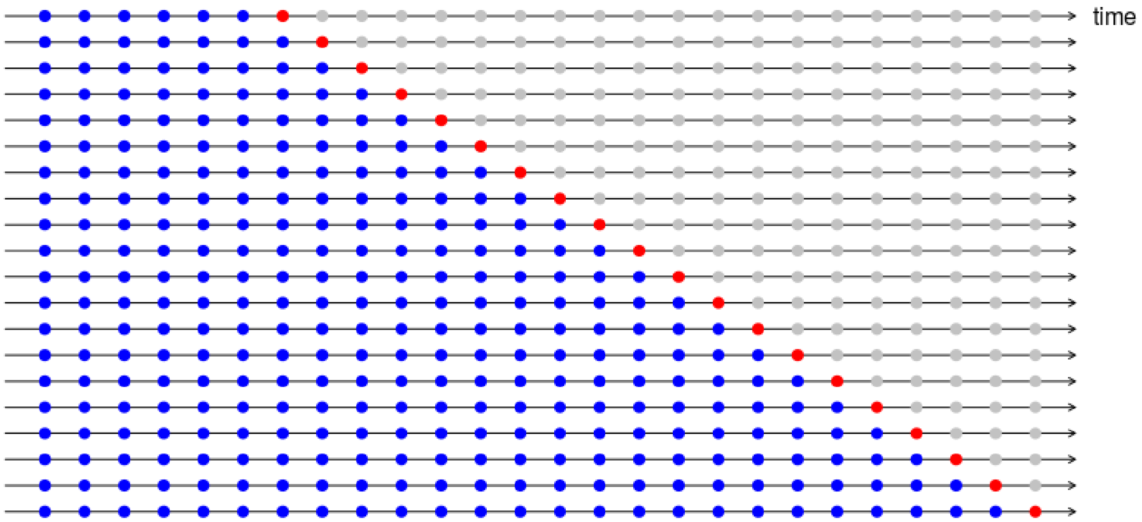
https://www.itl.nist.gov/div898/handbook/pmc/section6/pmc624.htm

Special concerns for time series data

- Correlated errors
- Cross-validation
- Lookahead (forecasting)
- Stationarity



https://www.itl.nist.gov/div898/handbook/pmc/section6/pmc624.htm

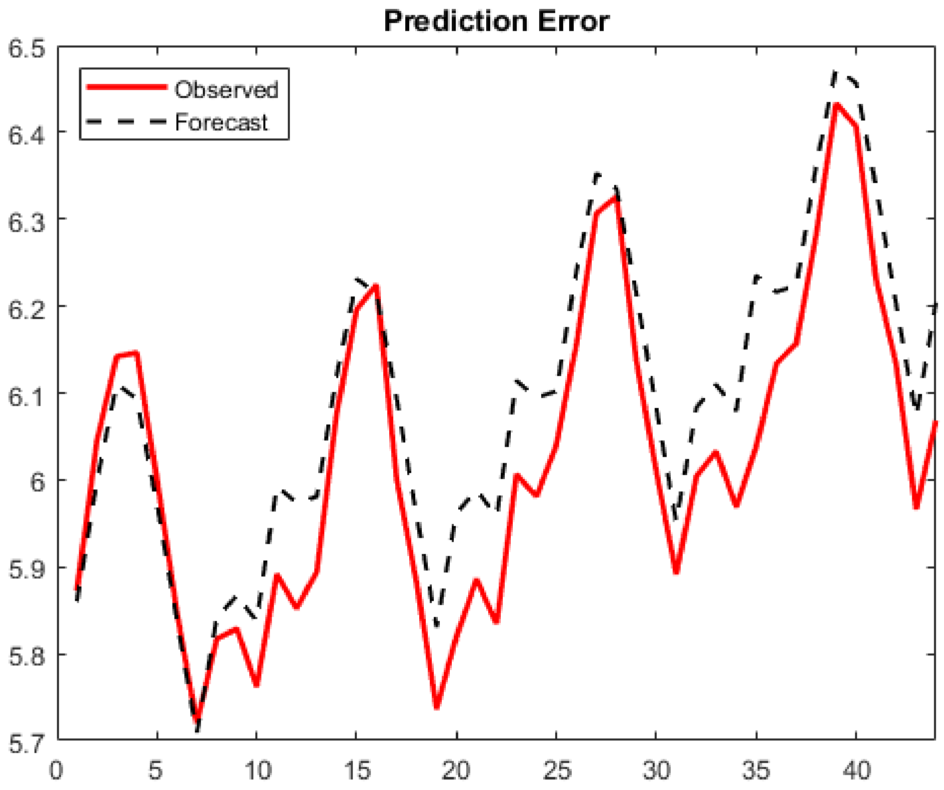


https://robjhyndman.com/hyndsight/tscv/

State space models

Box-Jenkins ARIMA modeling (background)

- Remarkably successful; remain quite close to cutting edge performance
- Excellent performance on small data sets



https://www.mathworks.com/help/econ/check-model-for-airline-passenger-data.htm

Box-Jenkins ARIMA modeling (background)

- L = lag operator (moves a variable in time)
- Alphas = autoregressive components
- Thetas = moving average components

$$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_ave

ARIMA(p, d, q)

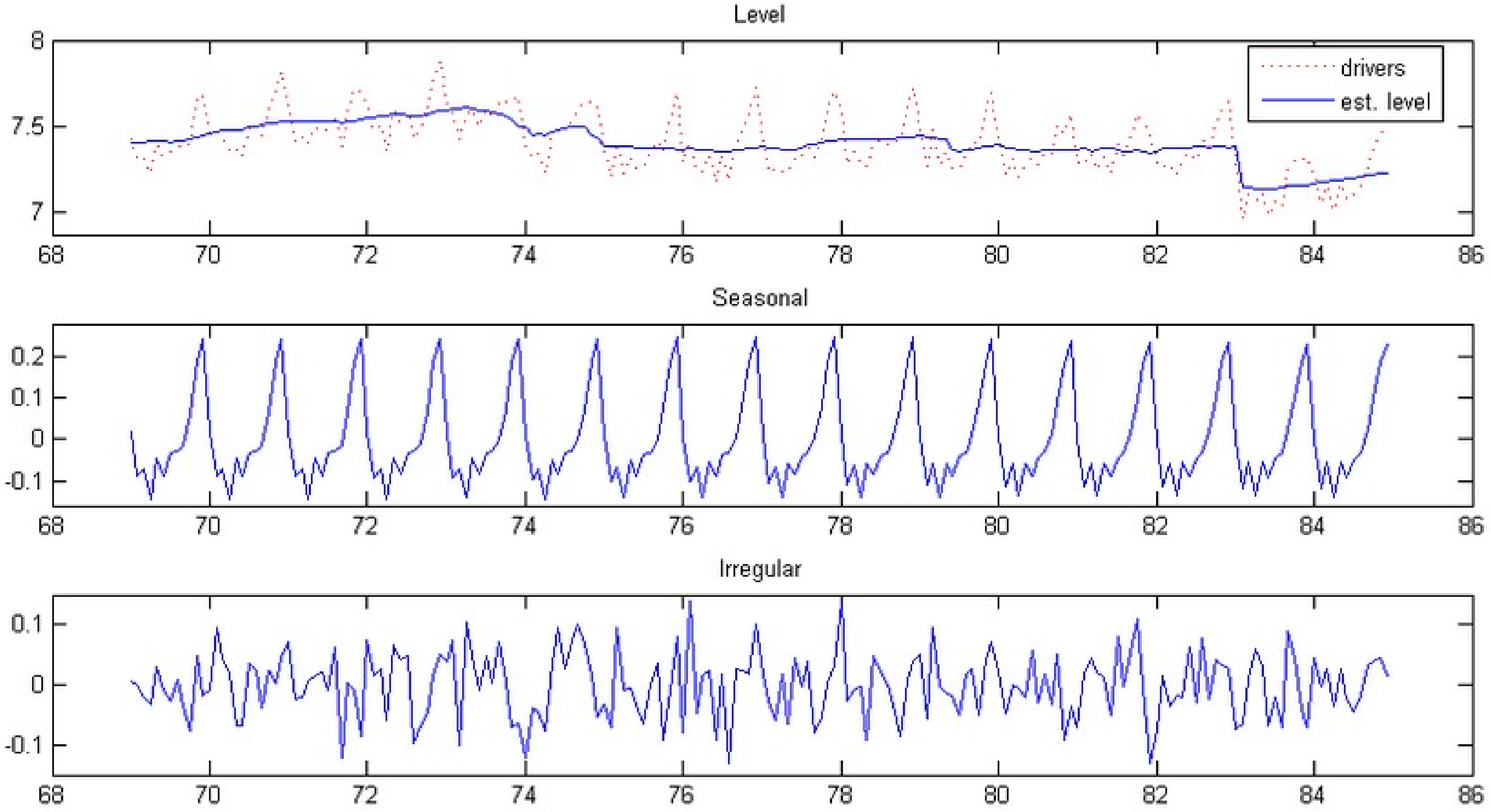
$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) (1 - L)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_ave

What’s missing from ARIMA

- Not especially intuitive
- No way to build in our underlying understanding about how it works
  - Random walk element
  - Cyclical element
  - External regressors
- Some systems cycle more slowly or stochastically than can be easily described with an ARIMA model

Structural time series



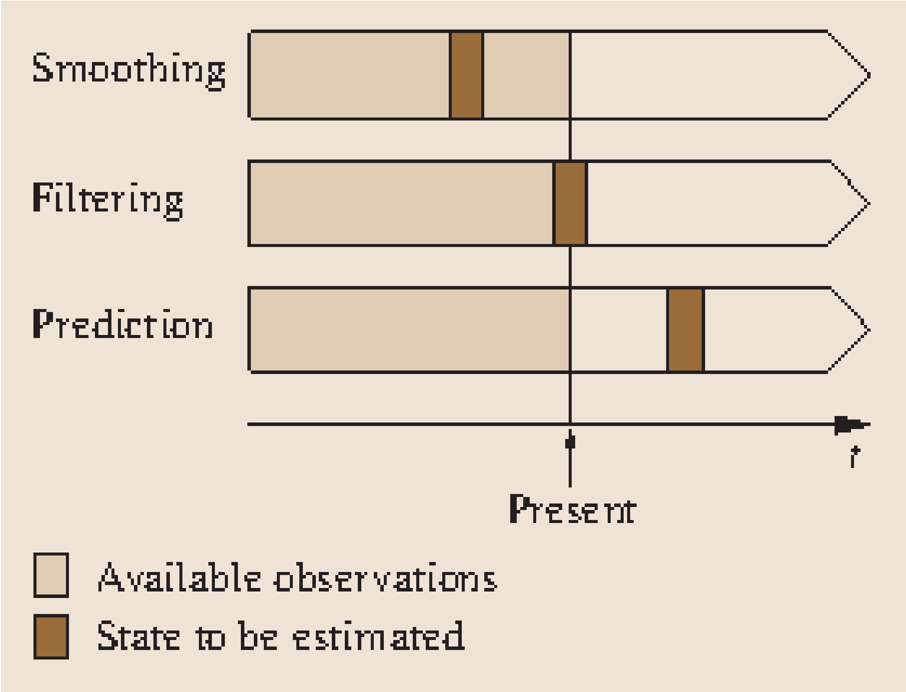
https://www.researchgate.net/figure/Driver-casualties-estima
ed-by-basic-structural-time-series-model_fig3_222100865

Structural time series

- Can also be expressed in ARIMA form
- Fit via maximum likelihood/Kalman filter
- Largely developed in econometrics
- Offer insights into underlying structure
- Also possible to inject Bayesian analysis via priors on parameters

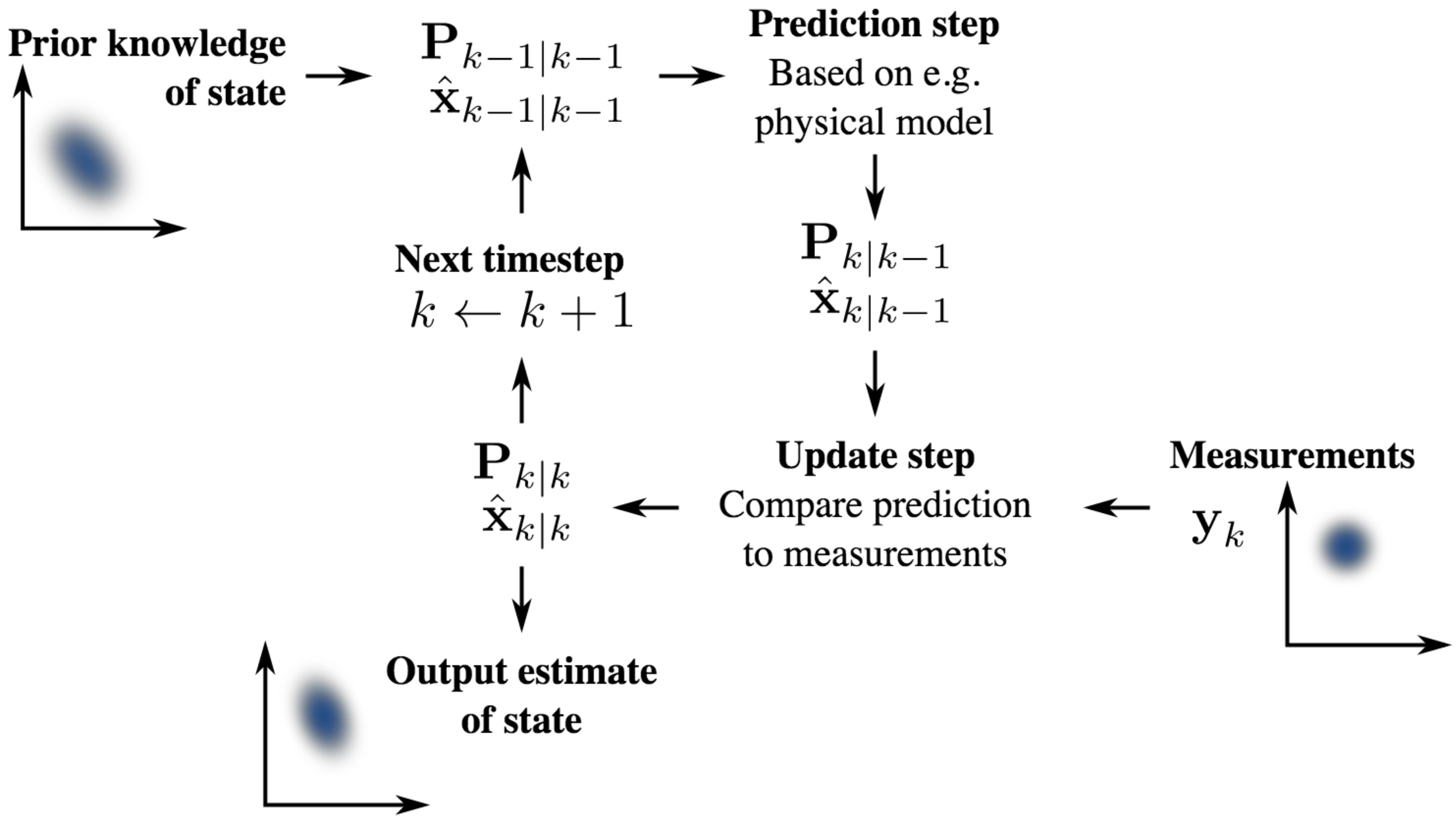
State space models offer three solutions

- **Filtering**: the distribution of the current state at time t given all previous measurements up to and including time t
- **Prediction**: the distribution of the future state at time t + k given all previous measurements up to and including time t
- **Smoothing**: the distribution of a given state at time k given all previous and future measurements from 0 to T (last time)



https://link.springer.com/chapter/10.1007/978-3-319-42928-1_22

Kalman filter



https://en.wikipedia.org/wiki/Kalman_filte

Kalman filter cont'd

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

https://en.wikipedia.org/wiki/Kalman_filte

Kalman filter cont'd

Predict [edit]

Predicted (*a priori*) state estimate	$\hat{\mathbf{x}}_{k\mid k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1\mid k-1} + \mathbf{B}_k \mathbf{u}_k$
Predicted (*a priori*) error covariance	$\mathbf{P}_{k\mid k-1} = \mathbf{F}_k \mathbf{P}_{k-1\mid k-1} \mathbf{F}_k^\mathsf{T} + \mathbf{Q}_k$

Update [edit]

Innovation or measurement pre-fit residual	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k\mid k-1}$
Innovation (or pre-fit residual) covariance	$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k\mid k-1} \mathbf{H}_k^\mathsf{T} + \mathbf{R}_k$
Optimal Kalman gain	$\mathbf{K}_k = \mathbf{P}_{k\mid k-1} \mathbf{H}_k^\mathsf{T} \mathbf{S}_k^{-1}$
Updated (*a posteriori*) state estimate	$\hat{\mathbf{x}}_{k\mid k} = \hat{\mathbf{x}}_{k\mid k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$
Updated (*a posteriori*) estimate covariance	$\mathbf{P}_{k\mid k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k\mid k-1}$
Measurement post-fit residual	$\tilde{\mathbf{y}}_{k\mid k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k\mid k}$

https://en.wikipedia.org/wiki/Kalman_filte

Akaike information criterion (AIC)

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

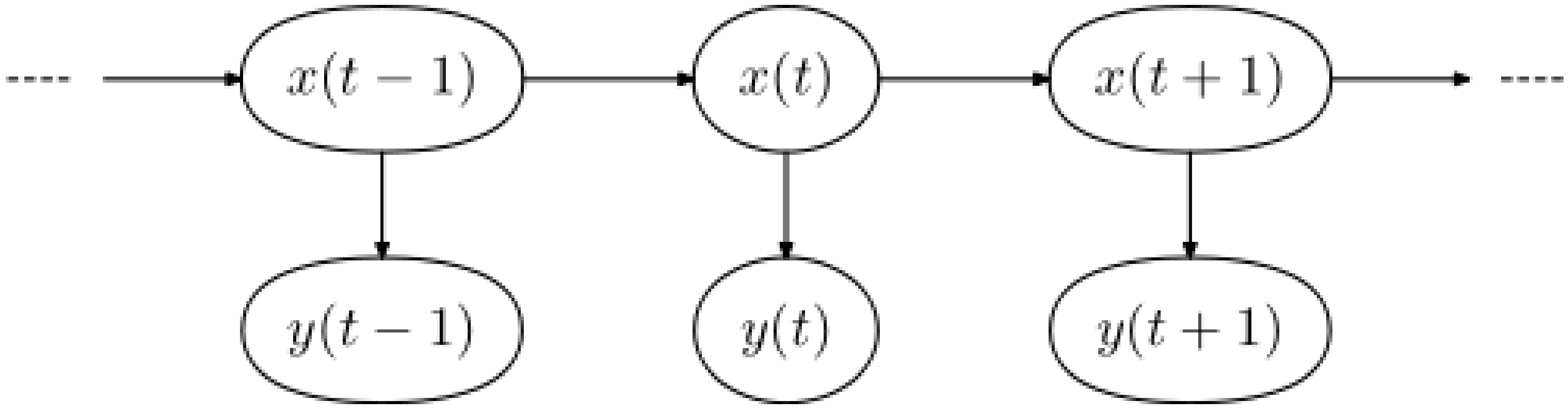
https://en.wikipedia.org/wiki/Akaike_information_criterion

Models we will fit

Local linear trend	*'local linear trend'*	*'lltrend'*	$y_t = \mu_t + \varepsilon_t$ $\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t$ $\beta_t = \beta_{t-1} + \zeta_t$
Smooth trend	*'smooth trend'*	*'strend'*	$y_t = \mu_t + \varepsilon_t$ $\mu_t = \mu_{t-1} + \beta_{t-1}$ $\beta_t = \beta_{t-1} + \zeta_t$

https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.structural.UnobservedComponents.html

Hidden Markov Models (HMMs)



https://en.wikipedia.org/wiki/Hidden_Markov_model

Hidden Markov Models (HMMs)

- State space model: observations are an indicator of underlying state
- Markov process: past doesn’t matter if present status is known
- Parameter estimation: Baum-Welch algorithm
- Smoothing/state labeling: Viterbi algorithm

Baum-Welch Algorithm for Determining Parameters

- Expectation-maximization parameter estimation:
  - Initialize parameters (with informative priors or randomly)
  - EM iterations:
    - Compute the expectation of the log likelihood given the data
    - Choose the parameters that maximize the log likelihood expectation
  - Exit when desired convergence is reached
- Guaranteed that the likelihood increases with each iteration
- BUT converges to a **local** maximum not a global maximum
- BUT can overfit the data

Baum-Welch Algorithm Details

- Problem

$$\theta = (A, B, \pi)$$

- Forward step

$$\alpha_i(t) = P(Y_1 = y_1, \ldots, Y_t = y_t, X_t = i \mid \theta)$$

$$\alpha_i(1) = \pi_i b_i(y_1),$$

$$\alpha_i(t+1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) a_{ji}.$$

- Backward step

$$\beta_i(t) = P(Y_{t+1} = y_{t+1}, \ldots, Y_T = y_T \mid X_t = i, \theta)$$

$$\beta_i(T) = 1,$$

$$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_j(y_{t+1}).$$

https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm

Baum-Welch Algorithm Details cont’d

$$\gamma_i(t) = P(X_t = i \mid Y, \theta) = \frac{P(X_t = i, Y \mid \theta)}{P(Y \mid \theta)} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)},$$

$$\xi_{ij}(t) = P(X_t = i, X_{t+1} = j \mid Y, \theta) = \frac{P(X_t = i, X_{t+1} = j, Y \mid \theta)}{P(Y \mid \theta)} = \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij} \beta_j(t+1) b_j(y_{t+1})},$$

https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm

Baum-Welch Algorithm Details cont'd

$$\pi_i^* = \gamma_i(1),$$

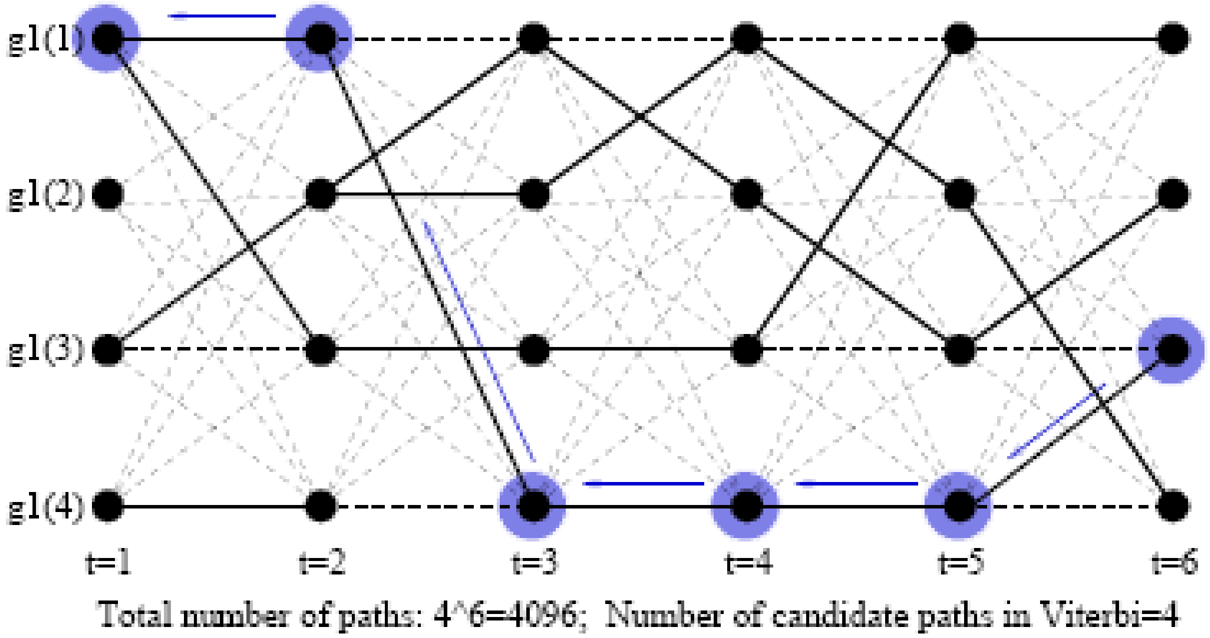
$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)},$$

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T} 1_{y_t=v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}, \quad 1_{y_t=v_k} = \begin{cases} 1 & \text{if } y_t = v_k, \\ 0 & \text{otherwise} \end{cases}$$

https://en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm

Viterbi Algorithm for Determining Sequence

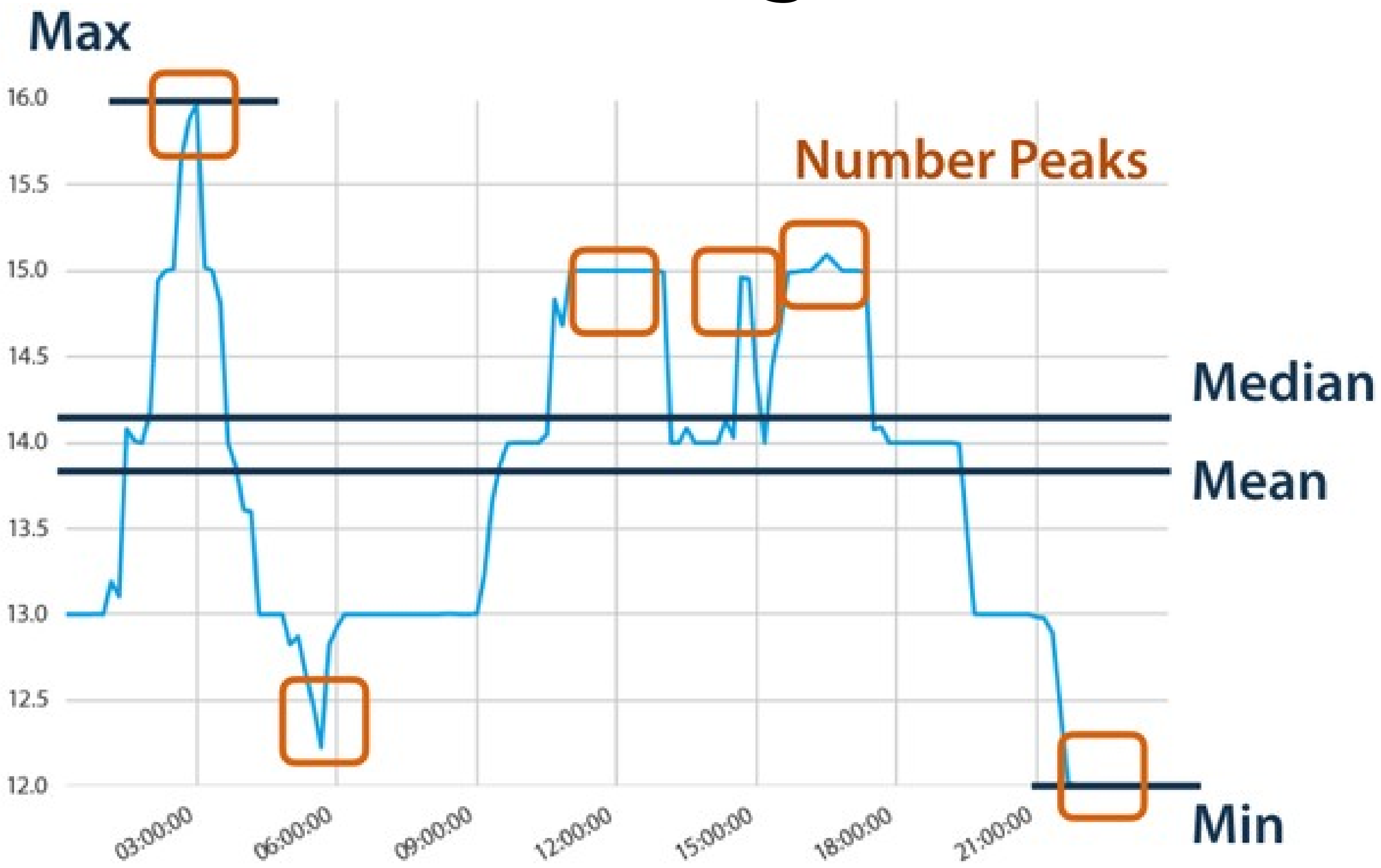
- Dynamic programming



https://newonlinecourses.science.psu.edu/stat508/lesson/13/13.9

Machine learning for time series

Time series feature generation



https://www.quora.com/How-can-I-extract-the-features-from-
me-series

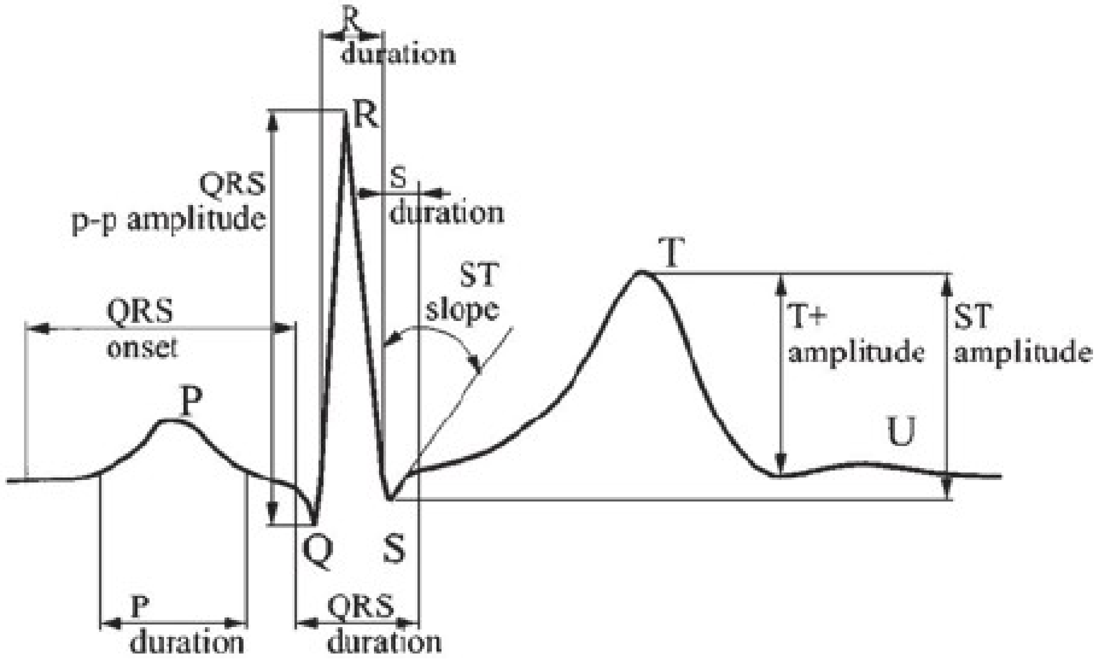
Time series features: catch22 canonical set

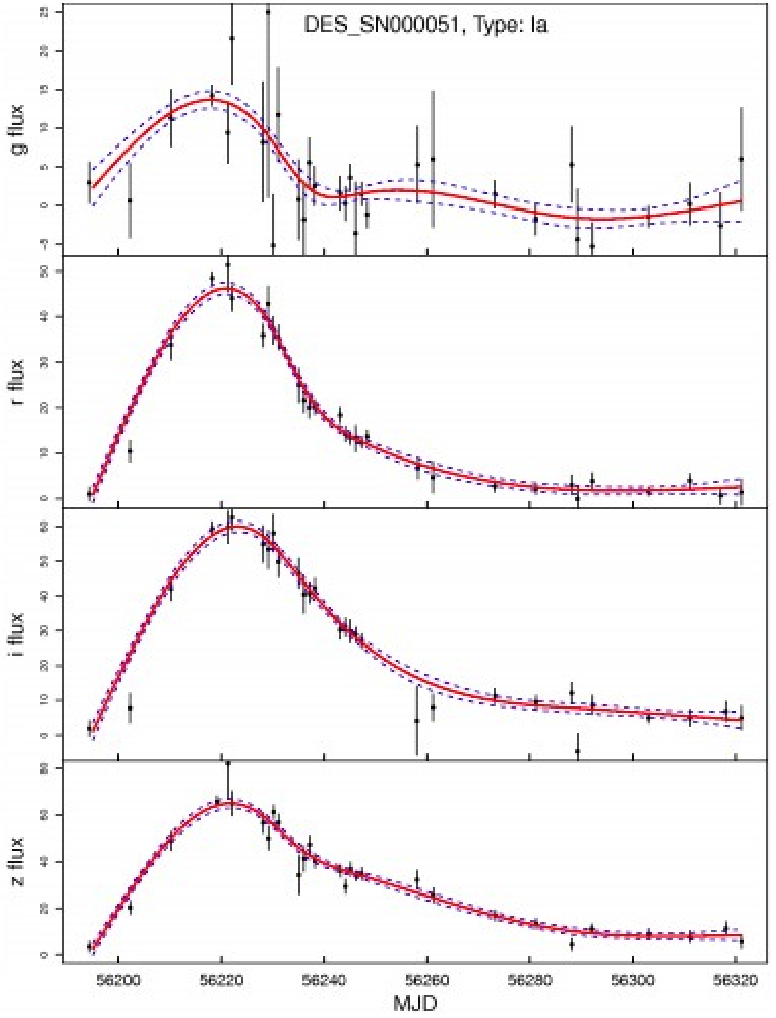
hctsa feature name	Description
Distribution	
DN_HistogramMode_5	Mode of z-scored distribution (5-bin histogram)
DN_HistogramMode_10	Mode of z-scored distribution (10-bin histogram)
Simple temporal statistics	
SB_BinaryStats_mean_longstretch1	Longest period of consecutive values above the mean
DN_OutlierInclude_p_001_mdrmd	Time intervals between successive extreme events above the mean
DN_OutlierInclude_n_001_mdrmd	Time intervals between successive extreme events below the mean
Linear autocorrelation	
CO_f1ecac	First $1/e$ crossing of autocorrelation function
CO_FirstMin_ac	First minimum of autocorrelation function
SP_Summaries_welch_rect_area_5_1	Total power in lowest fifth of frequencies in the Fourier power spectrum
SP_Summaries_welch_rect_centroid	Centroid of the Fourier power spectrum
FC_LocalSimple_mean3_stderr	Mean error from a rolling 3-sample mean forecasting
Nonlinear autocorrelation	
CO_trev_1_num	Time-reversibility statistic, $\langle (x_{t+1} - x_t)^3 \rangle_t$
CO_HistogramAMI_even_2_5	Automutual information, $m = 2, \tau = 5$
IN_AutoMutualInfoStats_40_gaussian_fmmi	First minimum of the automutual information function
Successive differences	
MD_hrv_classic_pnn40	Proportion of successive differences exceeding 0.04σ [20]
SB_BinaryStats_diff_longstretch0	Longest period of successive incremental decreases
SB_MotifThree_quantile_hh	Shannon entropy of two successive letters in equiprobable 3-letter symbolization
FC_LocalSimple_mean1_tauresrat	Change in correlation length after iterative differencing
CO_Embed2_Dist_tau_d_expfit_meandiff	Exponential fit to successive distances in 2-d embedding space
Fluctuation Analysis	
SC_FluctAnal_2_dfa_50_1_2_logi_prop_r1	Proportion of slower timescale fluctuations that scale with DFA (50% sampling)
SC_FluctAnal_2_rsrangefit_50_1_logi_prop_r1	Proportion of slower timescale fluctuations that scale with linearly rescaled range fits
Others	
SB_TransitionMatrix_3ac_sumdiagcov	Trace of covariance of transition matrix between symbols in 3-letter alphabet
PD_PeriodicityWang_th0_01	Periodicity measure of [31]

Table 1 The *catch22* feature set spans a diverse range of time-series characteristics representative of the diversity of interdisciplinary methods for time-series analysis. Features in *catch22* capture time-series properties of the distribution of values in the time series, linear and nonlinear temporal autocorrelation properties, scaling of fluctuations, and others.

https://arxiv.org/pdf/1901.10200.pdf

Time series features: discipline specific

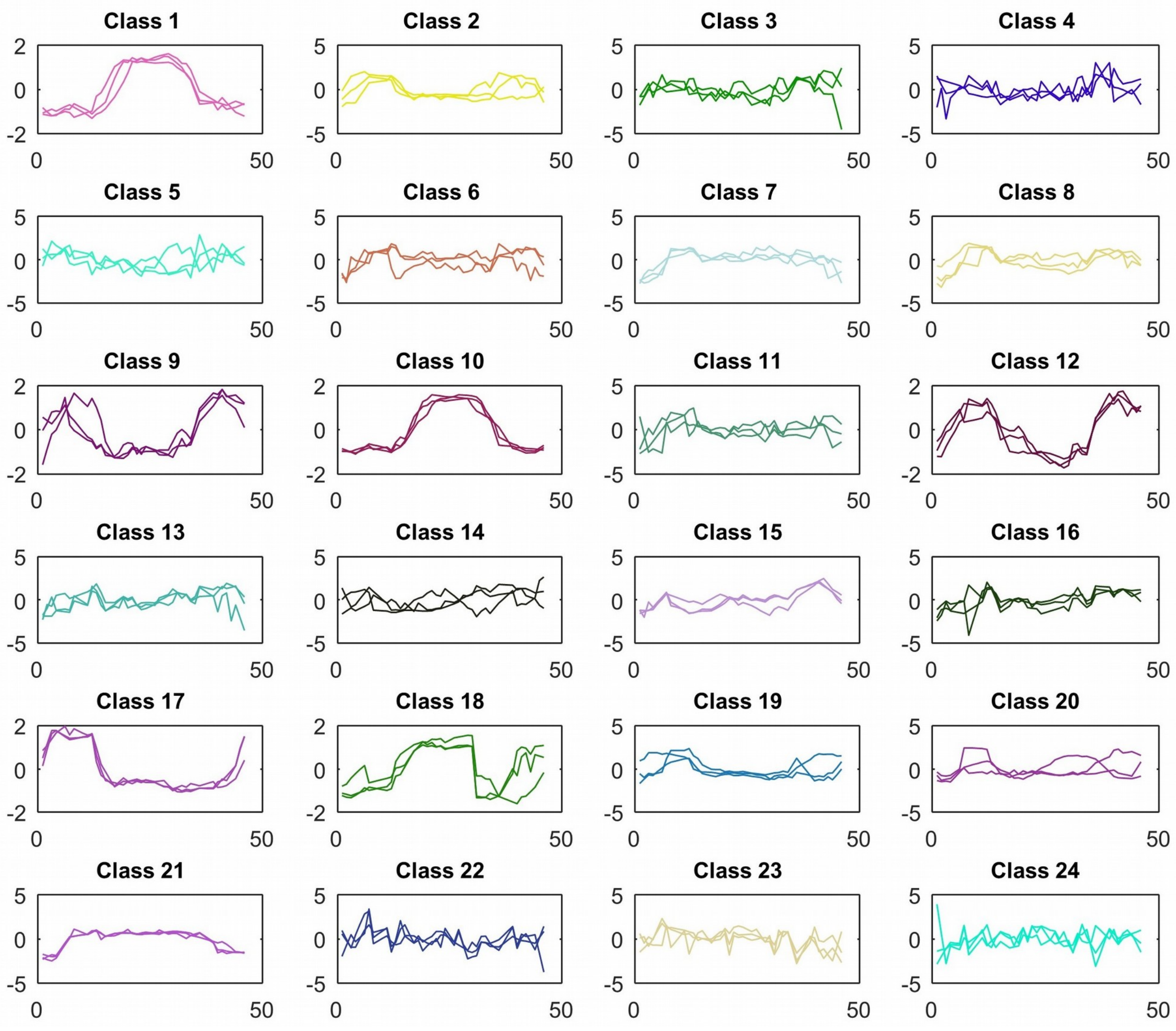




https://www.researchgate.net/figure/Graphical-representation-of-time-domain-ECG-characteristics-used-as-features-for-ECG_fig4_221914488

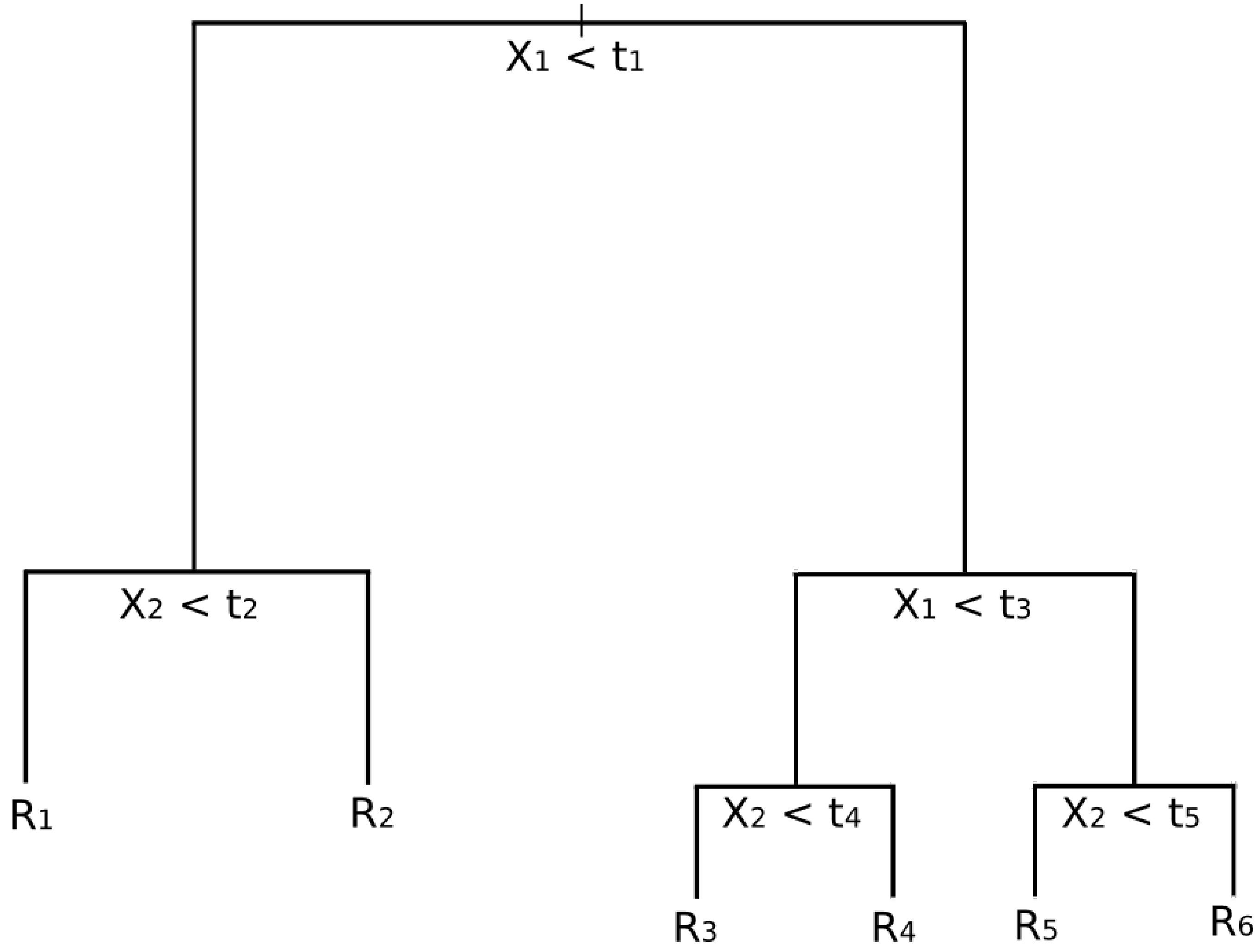
https://web.stanford.edu/group/mmds/slides2012/s-richar...df

Classification



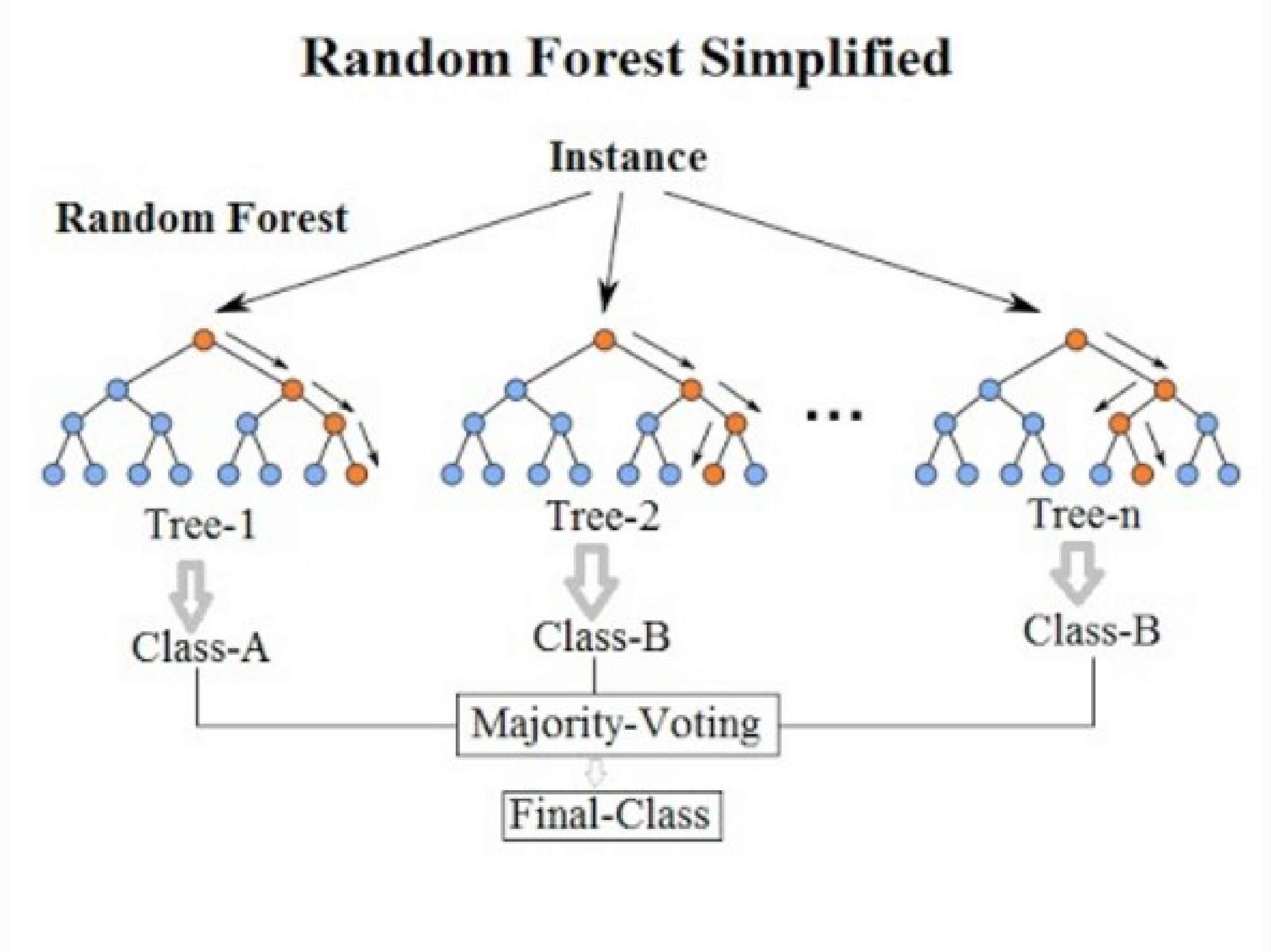
http://www.timeseriesclassification.com/description.php?Dataset=Crop

Trees



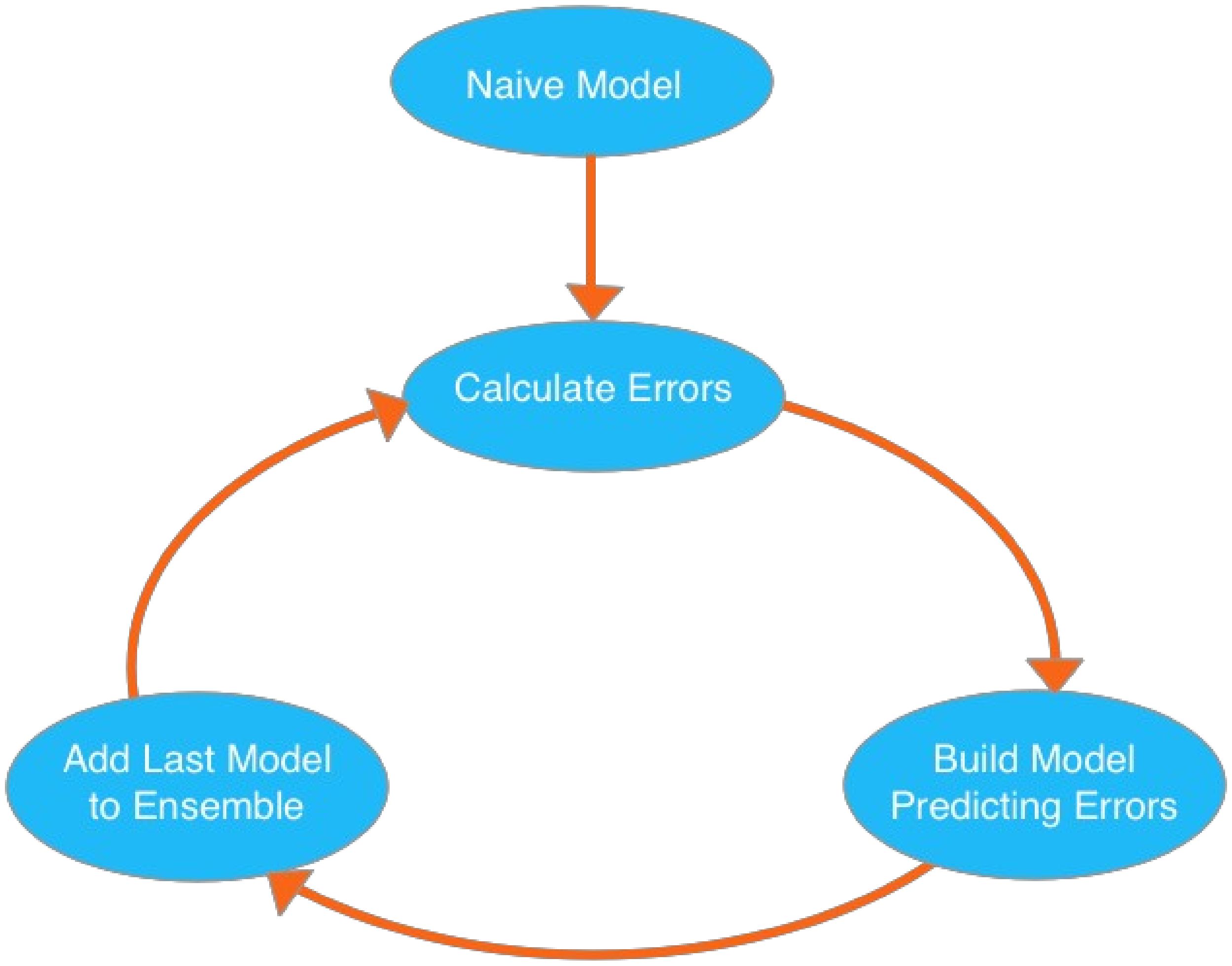
https://www.quantstart.com/articles/Beginners-Guide-to-Dec
ion-Trees-for-Supervised-Machine-Learning

Random Forest



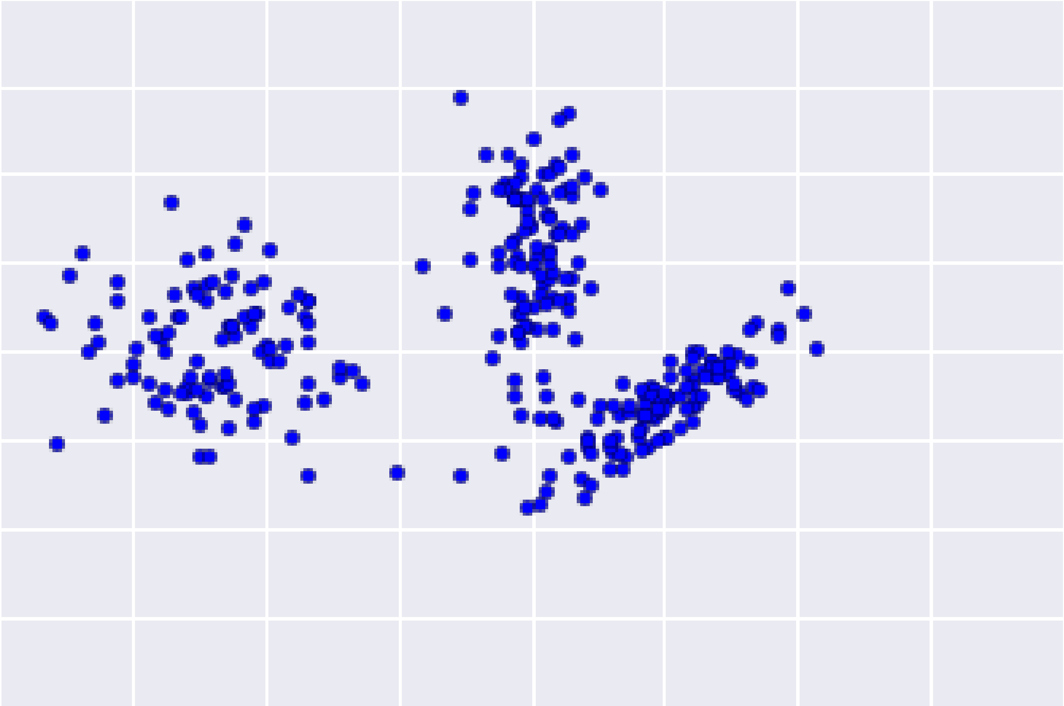
https://medium.com/@williamkoehrsen/random-forest-simple-exp
n-377895a60d2d

xgboost



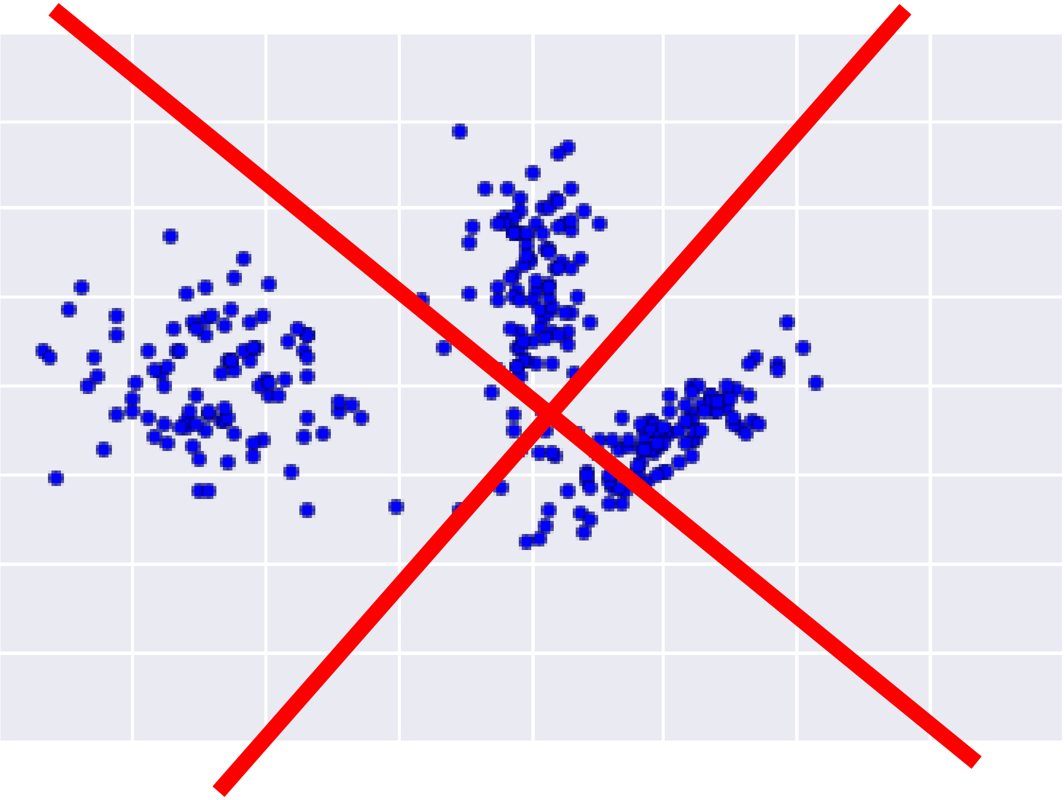
https://www.kaggle.com/dansbecker/xgboo

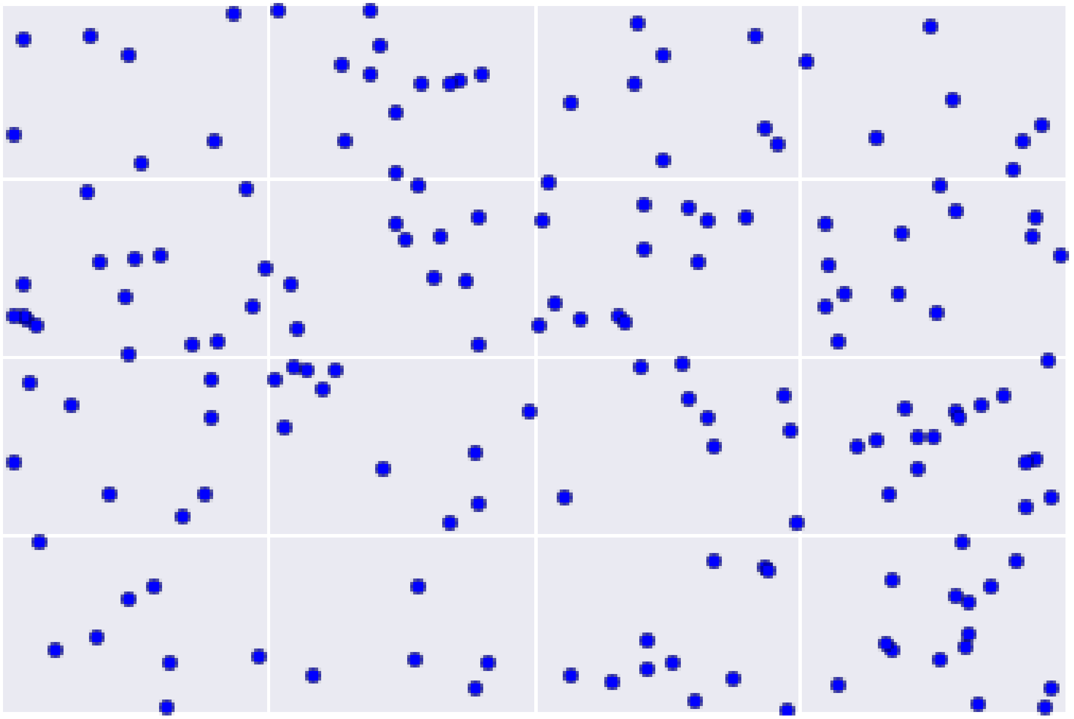
Clustering



https://developers.google.com/machine-learning/clusterin

Clustering: the reality

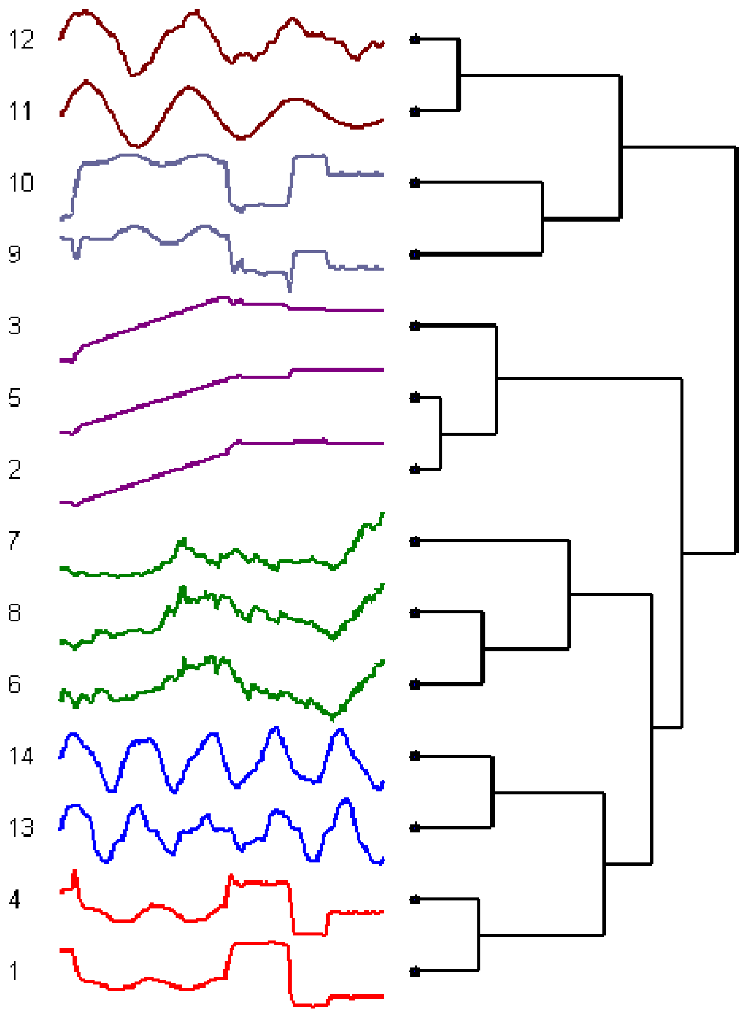




https://developers.google.com/machine-learning/clusteri

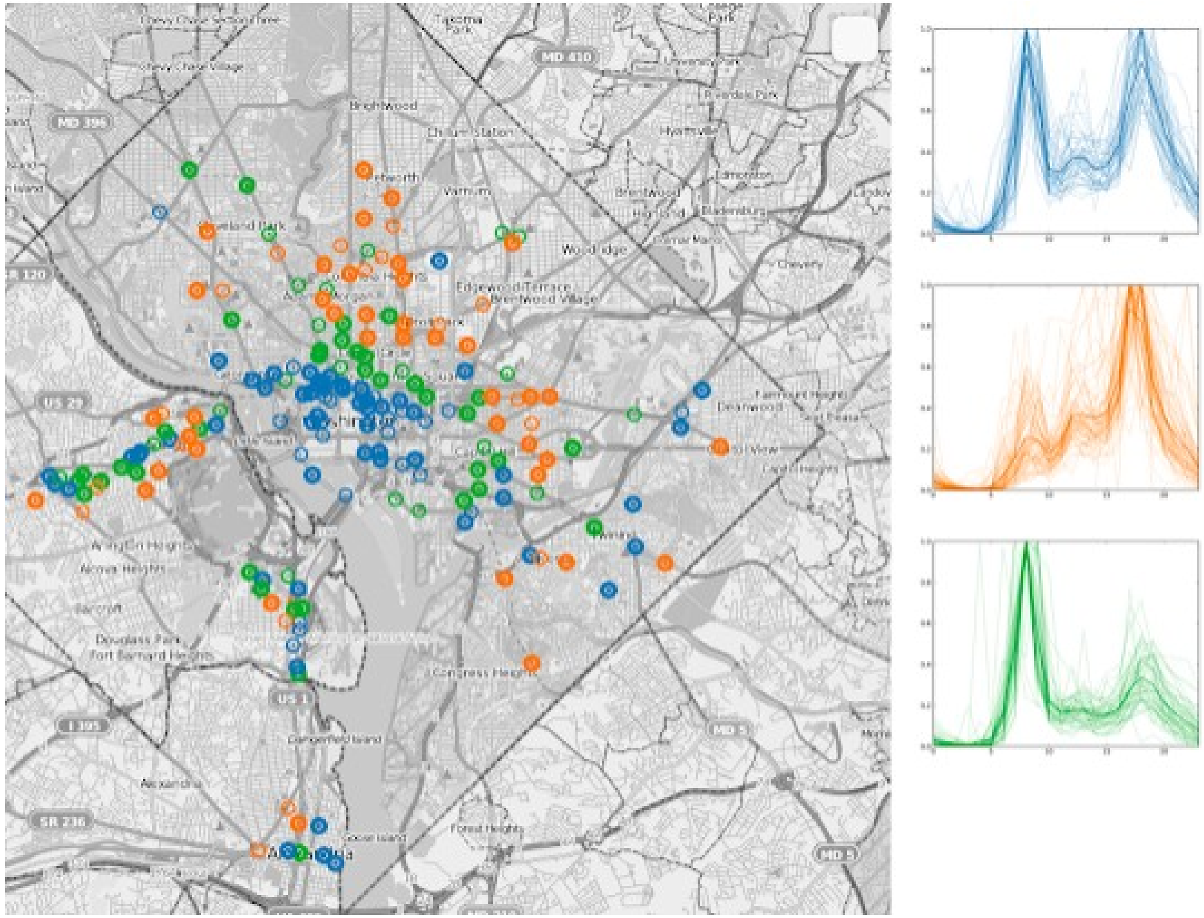
Time series clustering

- Surprisingly difficult
  - Conceptually
  - Computational costs
  - Pitfall: Euclidean distance
- Used across many disciplines
  - Medicine
  - Finance
  - Chemistry
  - Etc



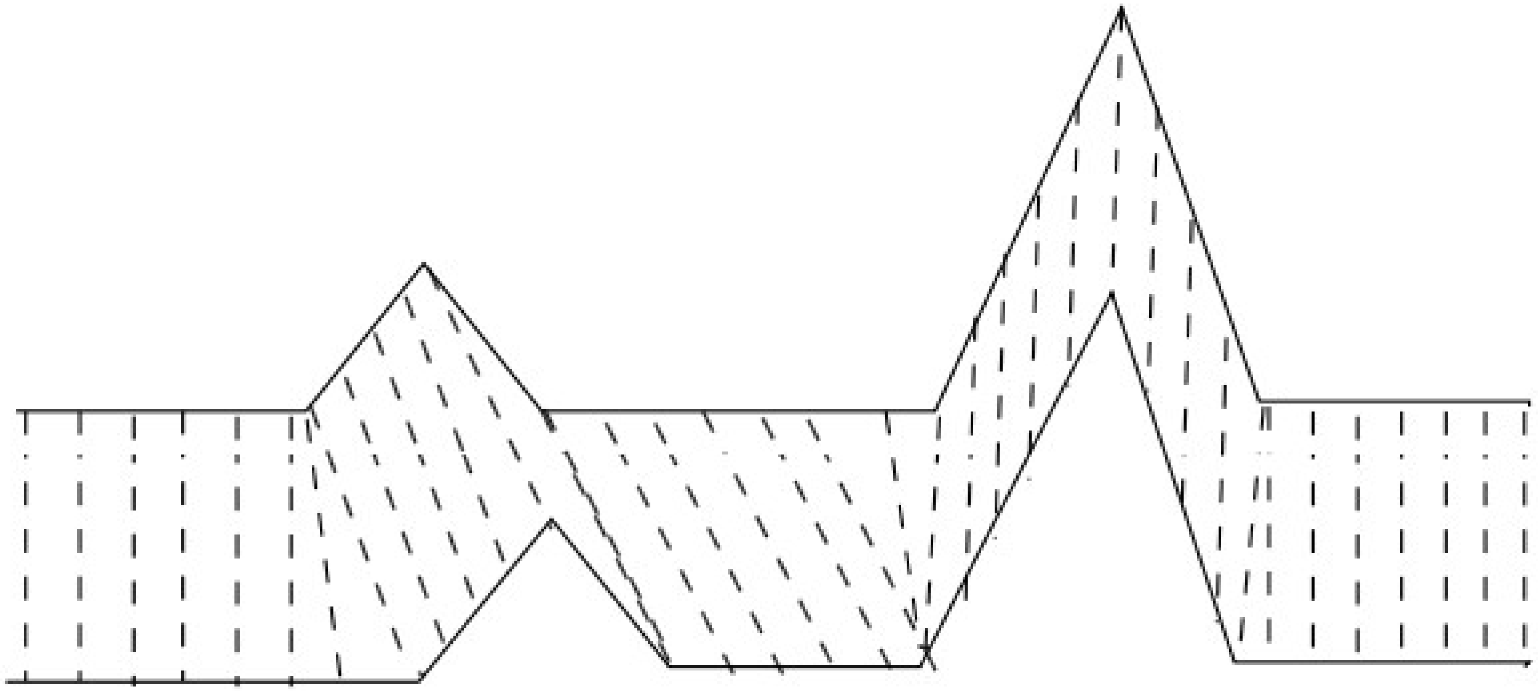
https://www.semanticscholar.org/paper/1-Characteristic-based-Cluster
Time-Series/0a5e8d1390f3ceb851f4a37a7ec8edb95e05f698/figure/8

A Practical Example



http://ofdataandscience.blogspot.com/2013/03/capital-bikeshare-time-series-clustering.html

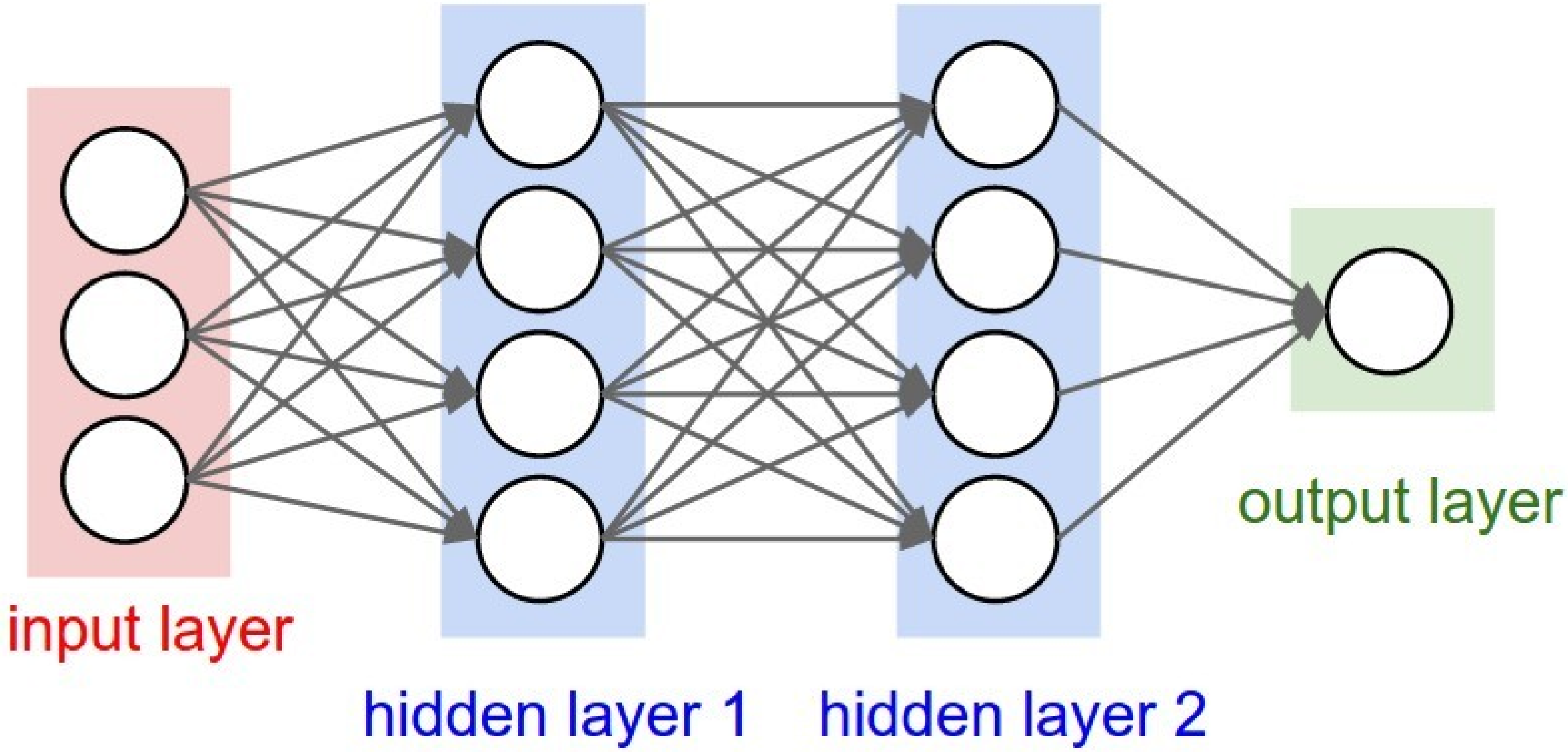
Dynamic time warping



https://en.wikipedia.org/wiki/Dynamic_time_warpi

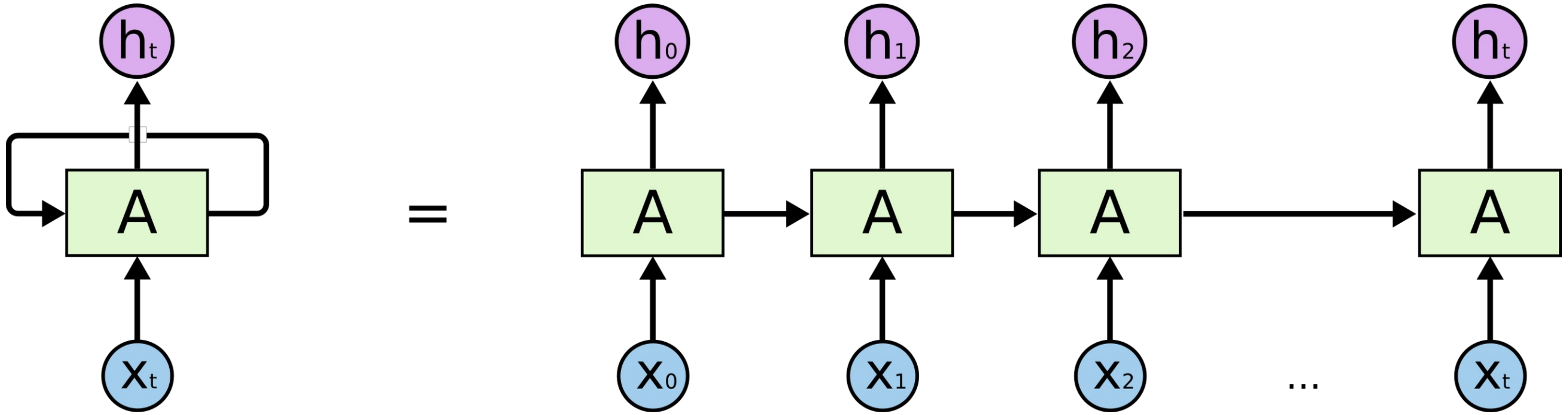
Deep learning for time series

Fully connected (FC)



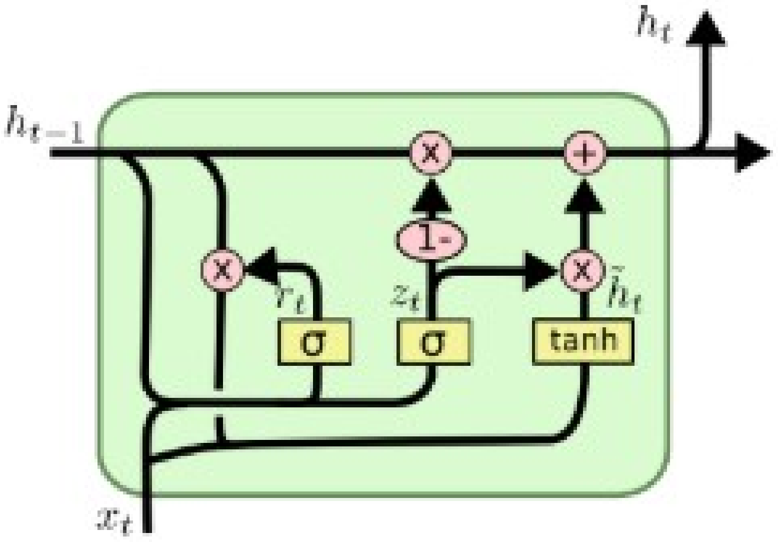
http://cs231n.github.io/convolutional-networ

Recurrent neural networks (RNNs)



https://colah.github.io/posts/2015-08-Understanding-LST

GRU



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

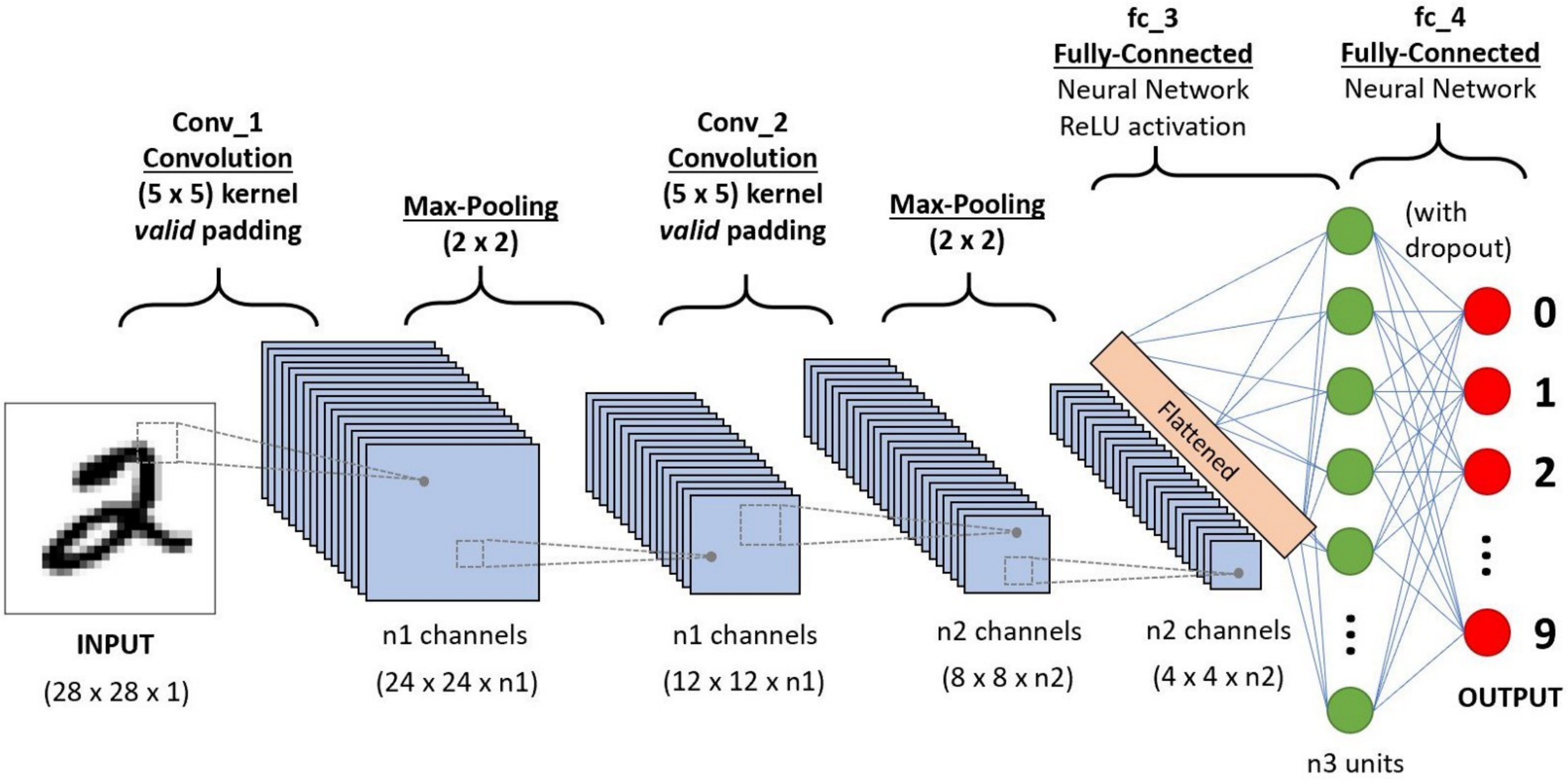
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

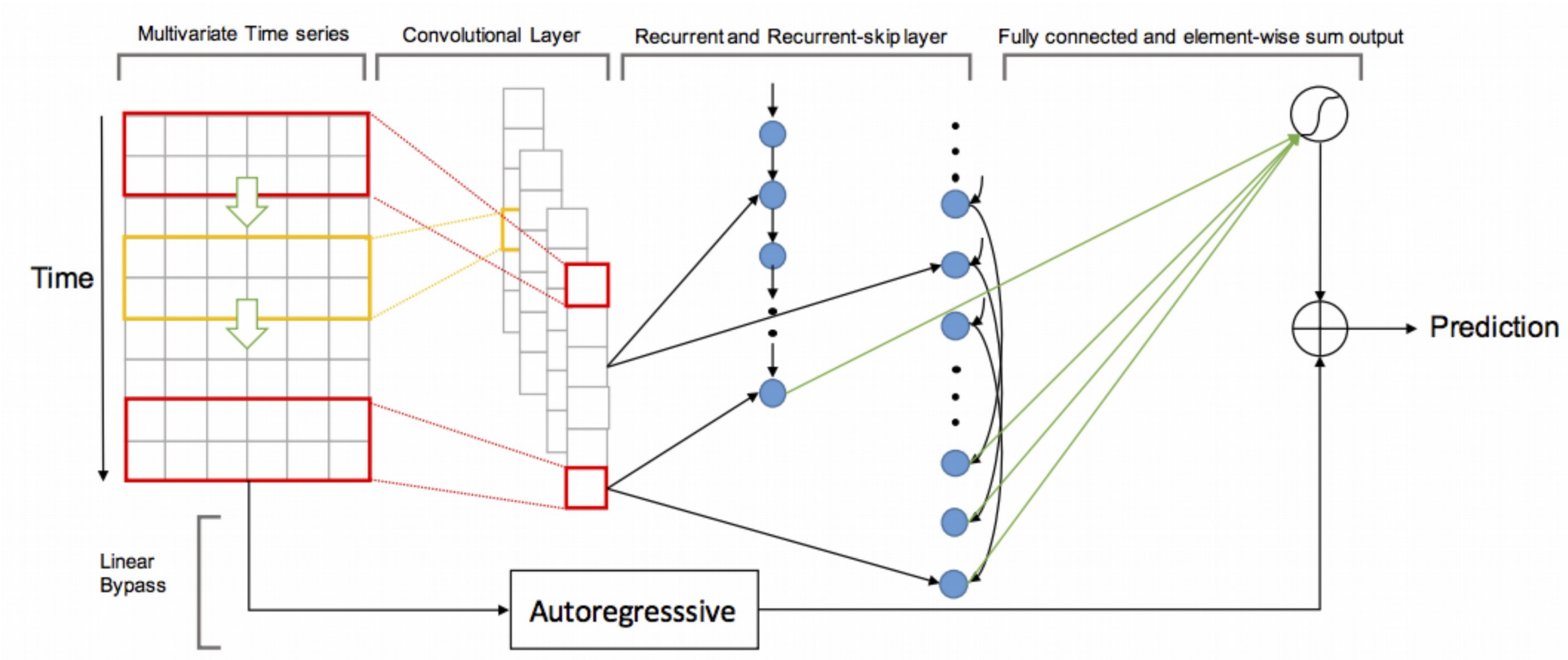
https://colah.github.io/posts/2015-08-Understanding-LST

Convolutional neural networks (CNNs)



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

LSTNet



https://modelzoo.co/model/lstnet
https://github.com/laiguokun/LSTNet

More options

- Anomaly detection
- New and old libraries (more R than Python)
- Automated forecasting at scale
- Combining machine learning and statistical approaches