AIME 2023

21st International Conference on Artificial Intelligence in Medicine

Portoroz, Slovenia, June 12-15

Part III: Medical Time Series Forecasting

12 June 2023, Portoroz, Slovenia

Ioanna Miliou

Assistant Professor, Data Science Group, Stockholm University

Stockholm University

Outline

- Intro to Time Series Forecasting
- Statistical Forecast Methods
- Machine Learning Methods
- Deep Learning Models
- Evaluation Metrics
- Nowcasting

Time Series

A sequence of observations obtained in successive order over some period

Financial Data

Heights of Ocean Tides

Weather measurements

Product Sales

…

Very common in the medical domain

Electroencephalographic signals

Heart rate

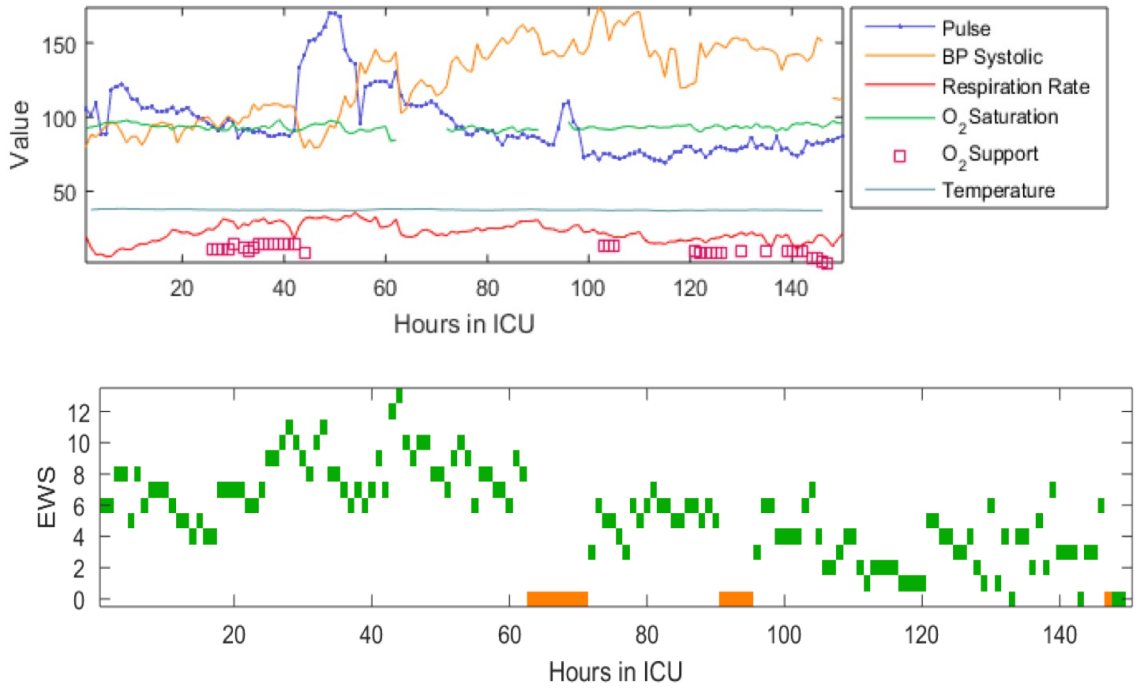
Respiratory rate

Temperature

…

Medical Time Series Forecasting

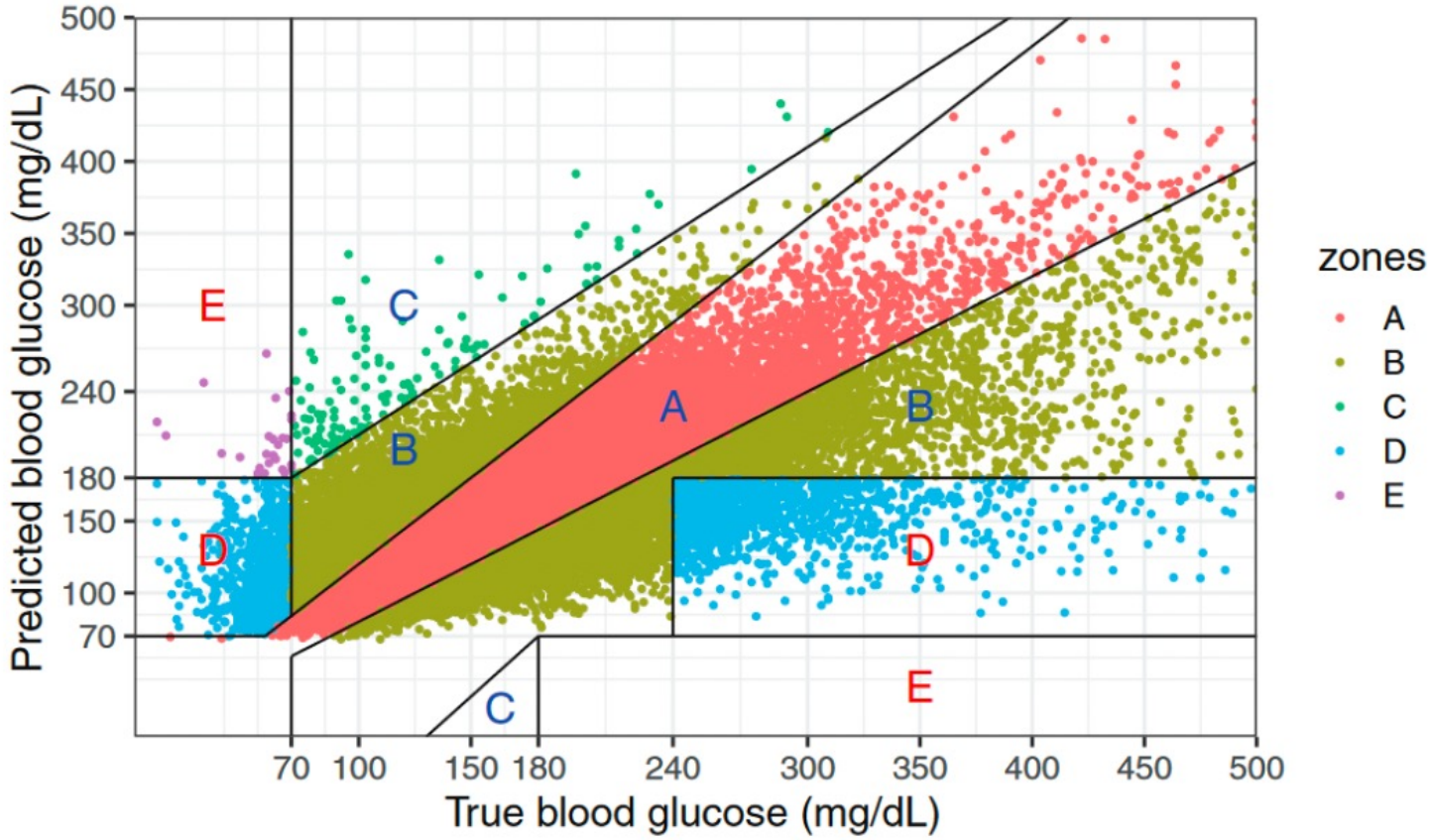
- Intensive Care Unit (ICU) patient's progression
- Vital signs deterioration prediction
- At each precise time step *t* in the admission period, we aim to predict the values for the same variables in the forward time of *h* hours using the records of *p* past hours in the ICU



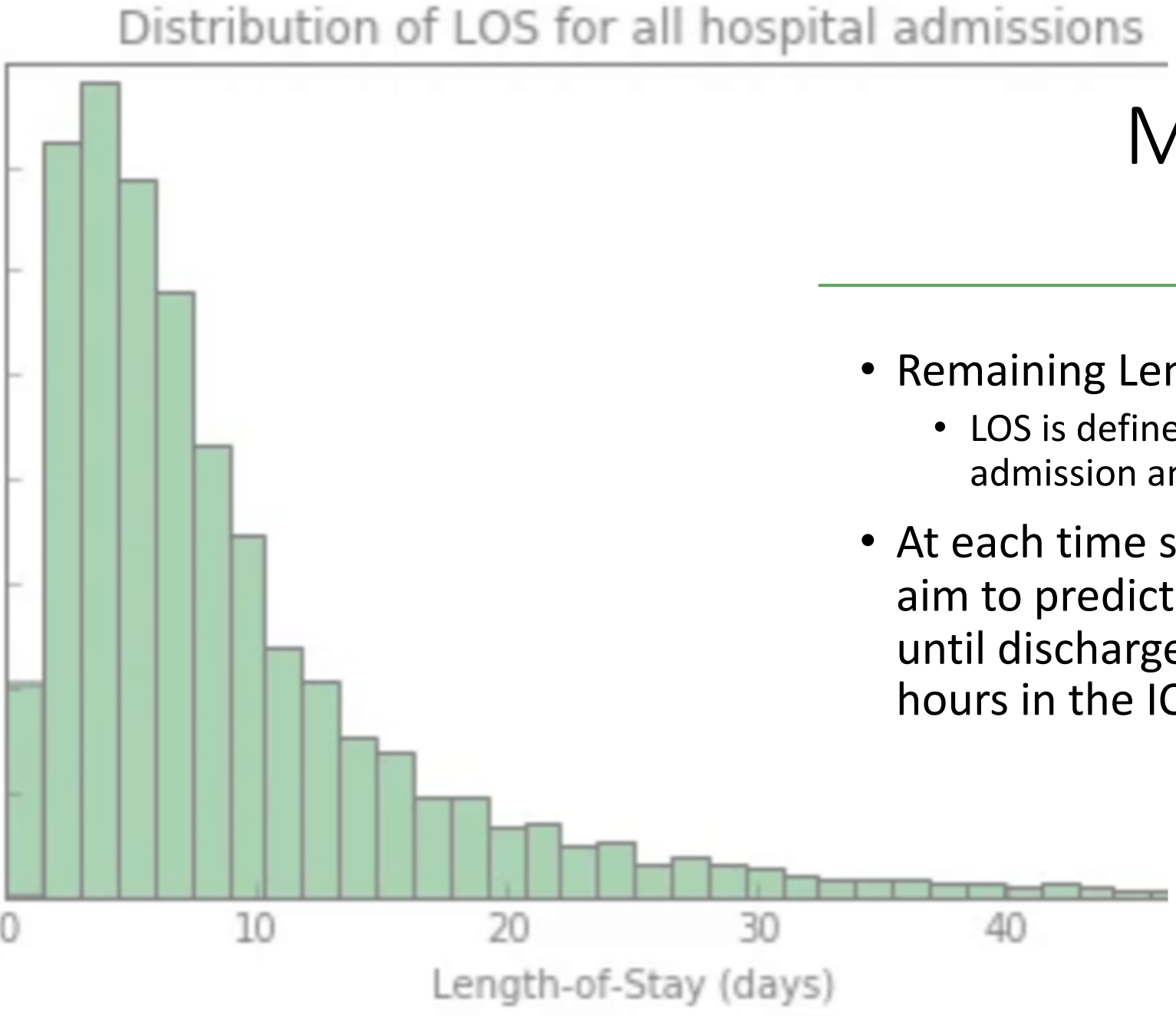
Vilic et. al, **Simplifying EHR Overview of Critically Ill Patients Through Vital Signs Monitoring**, IEEE Journal of Biomedical and Health Informatics, 2016

Medical Time Series Forecasting

Blood glucose in ICU for early warning & detection of dysglycaemia



Fitzgerald et. Al., **Incorporating real-world evidence into the development of patient blood glucose prediction algorithms for the ICU**, Journal of the American Medical Informatics Association, 2021



Medical Time Series Forecasting

- Remaining Length Of Stay (LoS) Prediction
  - LOS is defined as the time between hospital admission and discharge measured in days
- At each time step *t* at the admission period, we aim to predict the remaining number of days until discharge, using the records of the *p* past hours in the ICU

Formulating the problem

In a typical scenario, we have an outcome measurement, usually categorical (such as heart attack/no heart attack) or quantitative (such as a temperature measurement), that we wish to predict based on a set of features

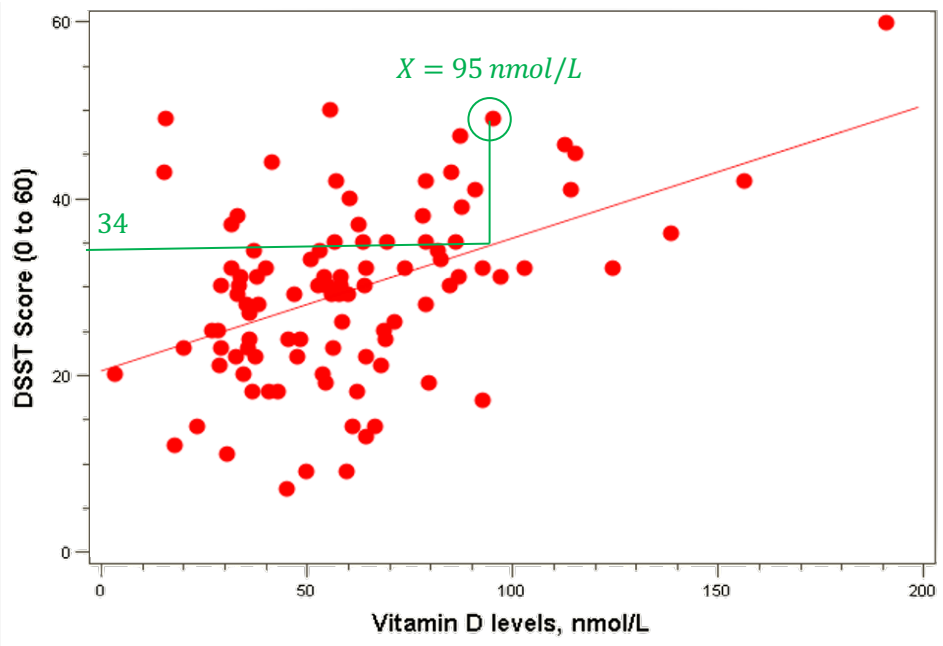
- **Classification**: when we predict categorical (qualitative) outputs
- **Regression**: when we predict quantitative outputs

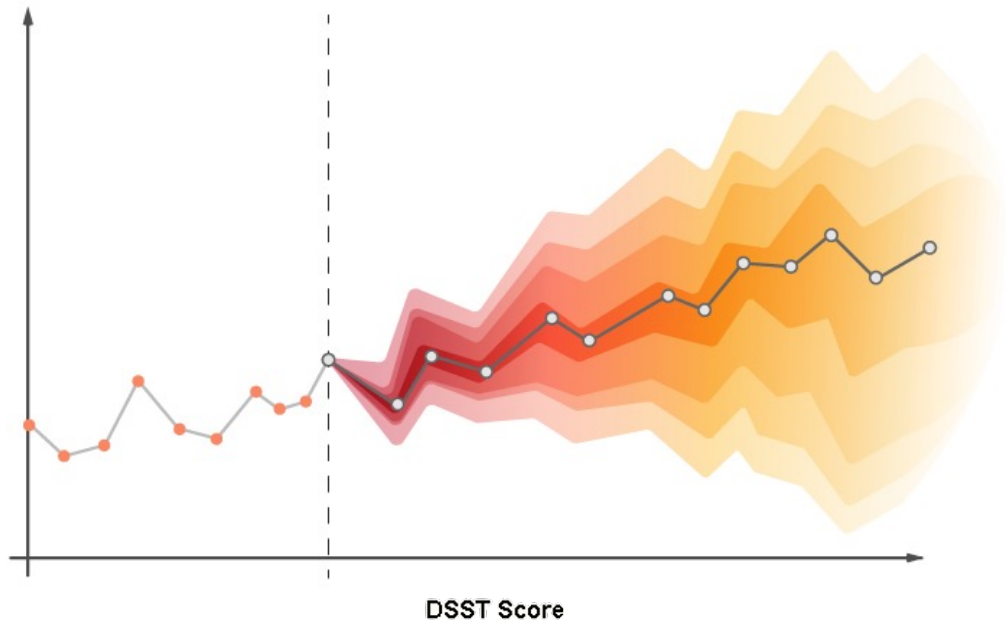
Regression

- Statistical method that attempts to determine the relationship between the dependent variable Y and a set of independent variables X
  - Dependent or response variable: the variable whose values change as a consequence of changes in other values in the system
  - Independent or predictors or explanatory variables: regarded as inputs to the system and may take on different values freely
- Widely used for predicting the next value or values of the dependent variable Y from the values of the independent variables X

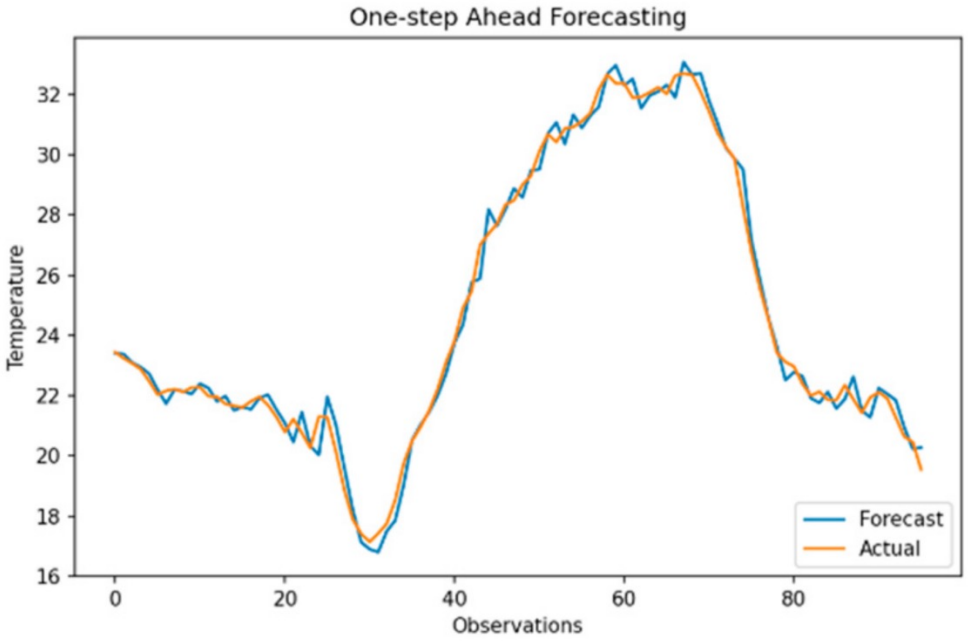
Regression vs Forecasting

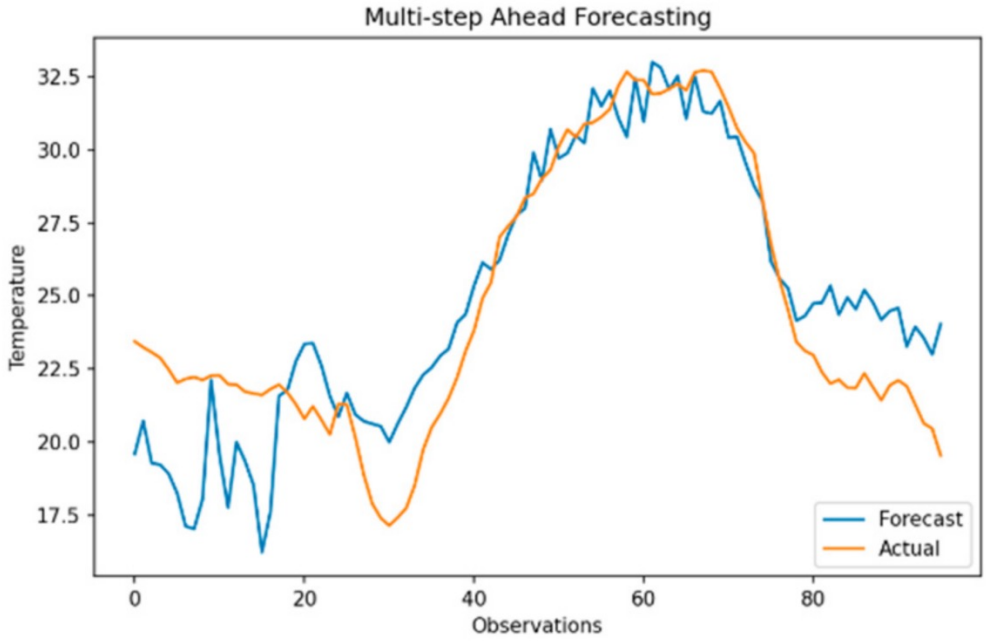
- **Regression**: determine the relationship between two time series
- **Forecasting**: predict the next value of a time series from the values of one or multiple time series in a dataset using the learned relationship



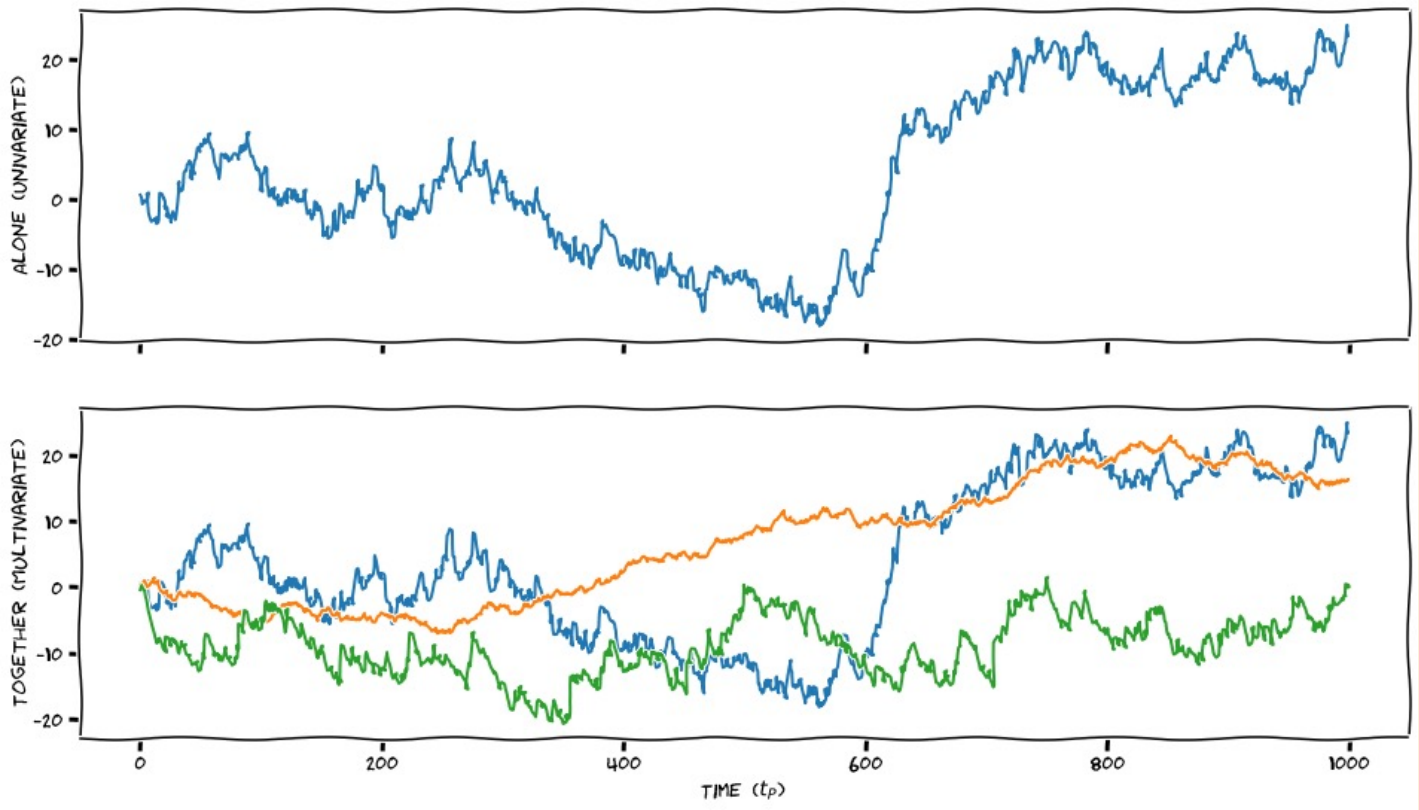


Time Series Forecasting





Univariate vs Multivariate TS



- Time series can be classified as univariate and multivariate
- It is an important distinction as the selection of the forecasting method depends, among other factors, on this
- Univariate time series is observation of one variable over time
- Multivariate time series is a collection of several univariate time series where a series can impact others

Methods for Time Series Forecasting

- **Statistical Methods**
  - extract relevant information from historical data
  - inference about the relationships between variables and their significance
  - highly interpretable
- **Traditional Machine Learning Methods**
  - accurate predictions
  - sacrifice interpretability for predictive power
- **Deep Learning Methods**
  - accurate predictions
  - added complexity – sacrifice interpretability
  - learn features and dynamics from the data – no need for feature engineering

Outline

- Intro to Time Series Forecasting
- Statistical Forecast Methods
- Machine Learning Methods
- Deep Learning Models
- Evaluation Metrics
- Nowcasting

Statistical Models

- AutoRegression (AR)
- Moving Average (MA)
- AR(I)MA
- ARIMAX
- SARIMA(X)
- VARMA(X)
- Exponential Smoothing
- Holt Winter's Exponential Smoothing
- …

Autoregression (AR)

- AR states that the next observation depends on the past observations with some lag p which represents the maximum lag
- Linear function of its past values plus a random noise/error
- We assume that X_t is stationary
- *AR(p):* autoregressive model of order p

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are constants and ε_t is white noise

Moving Average (MA)

- MA assumes that the current value is dependent on the error terms (including the current error)
- It uses the last t time points in order to predict time point $t + 1$
- Values are generated by a normal distribution with 0 mean
- *MA(q):* linear combination of the past values (of order q)

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

where $\theta_1, \theta_2, \ldots, \theta_q$ are constants

ARIMA

- At first, we have the autoregression model $AR(p)$
- Then, we add the moving average model $MA(q)$
- After, we add the order of integration $I(d)$
  - The parameter d represents the number of differences required to make the series stationary, because a model cannot forecast on non-stationary time series data

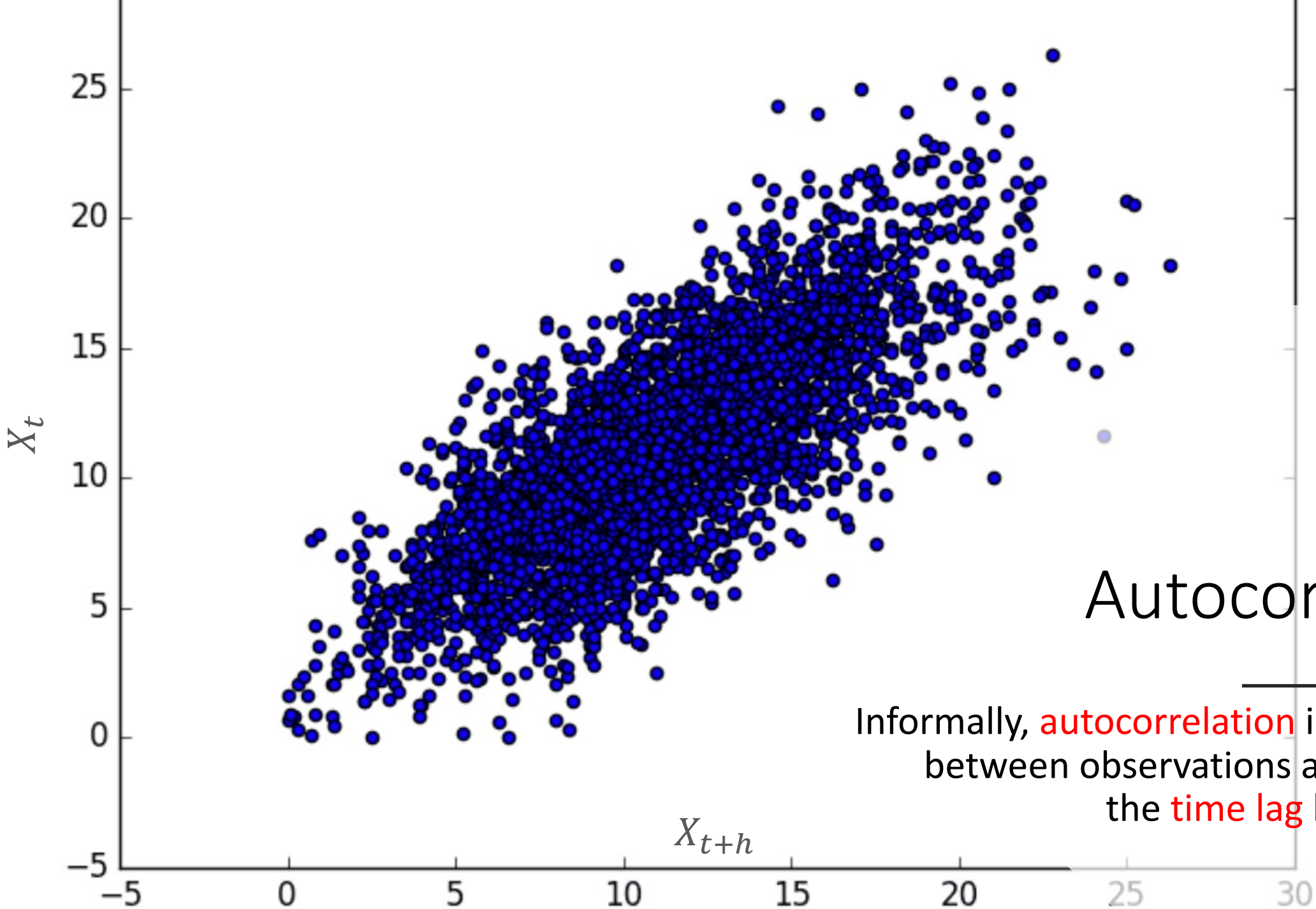
ARIMA

- At first, we have the autoregression model *AR(p)*
- Then, we add the moving average model *MA(q)*
- After, we add the order of integration *I(d)*

$$X = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

Choosing the order

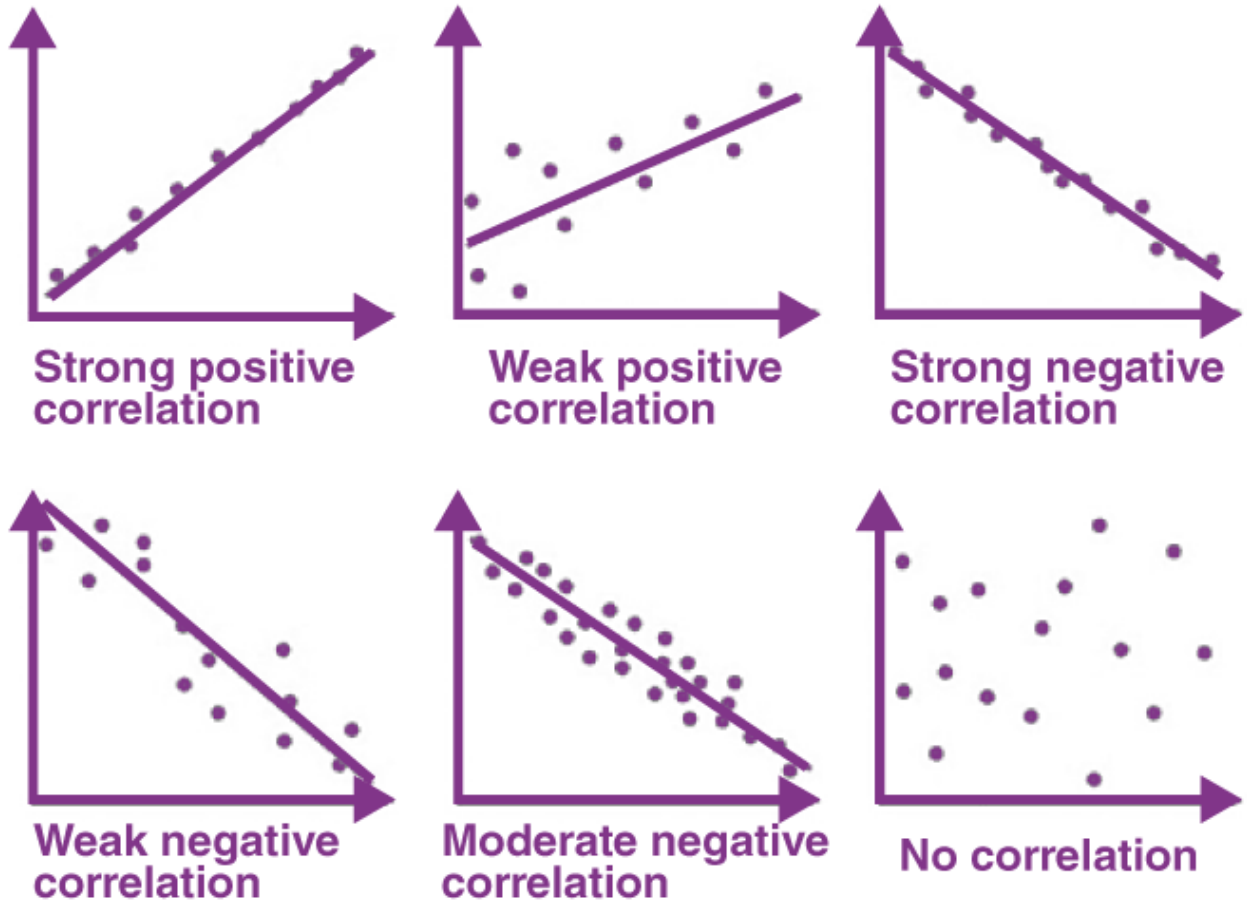
- When working with time series, it is often useful to analyze the autocorrelation of the data to understand the patterns and dependencies between time steps
- Two common tools for this analysis are the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF)



Autocorrelation

Informally, autocorrelation is the similarity between observations as a function of the time lag between them

Correlation



Correlation

- Given two time series X and Y, how do they correlate?

- Pearson Correlation is a measure of the linear dependence between two variables during period $[t_1, t_n]$:

$$r = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(X_i - \bar{X})}{\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})}}$$

Lag

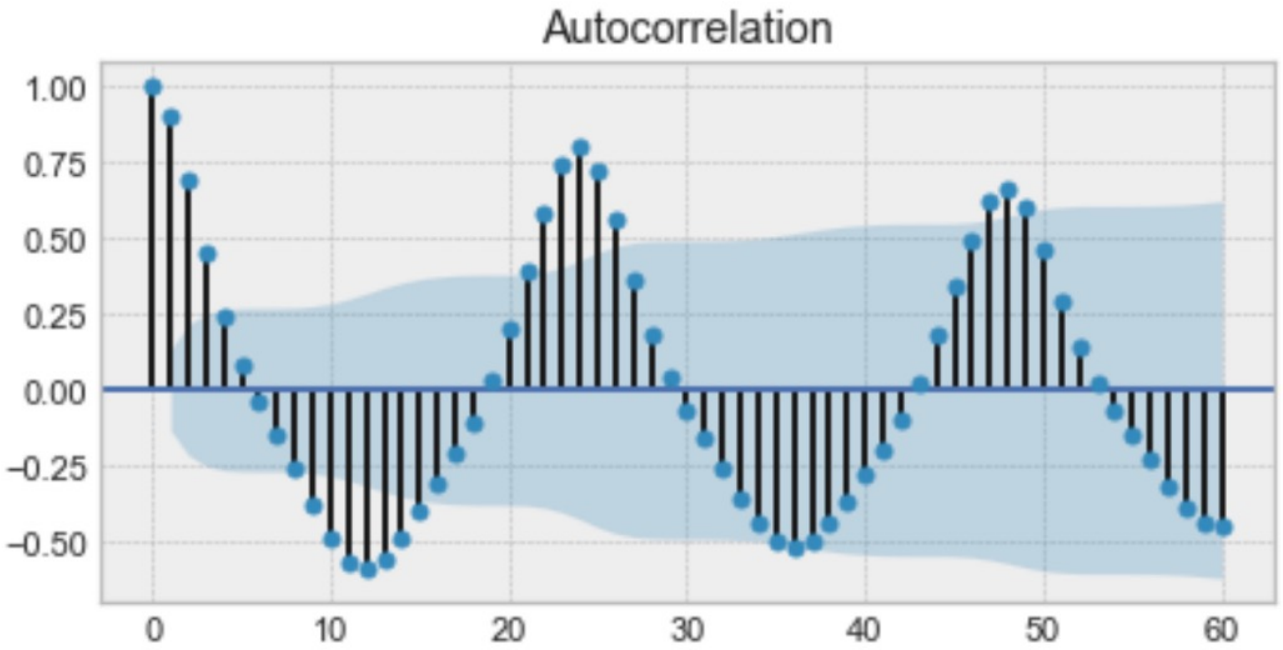
- To compute correlation, we need two variables
- But if we only have a time series, we need to compute the correlation of the time series with a k-th lagged version of itself
- Lag: for a specific time point t, the observation X_{t-k} is called the k-th lag of X_t
- A time series with lag $k = 1$ is a version of the original time series that is 1 period behind in time

Autocorrelation Function

- The autocorrelation of a time series is the Pearson correlation between values of the time series at different times, as a function of the two times or of the time lag
- Autocorrelation function (ACF): $\rho_X(k) = Corr(X_{t-k}, X_t)$
- The ACF plot can provide answers to the following questions:
  - Is the observed time series **white noise/random**?
  - Is an observation related to an adjacent observation, an observation twice-removed, and so on?
  - Can the observed time series be modeled with an **MA model**? If yes, what is the order?

ACF plot

The ACF plot can be used to identify the parameter q which represents the biggest lag after which other lags are not significant on the autocorrelation plot



The plot looks like a sinusoidal function

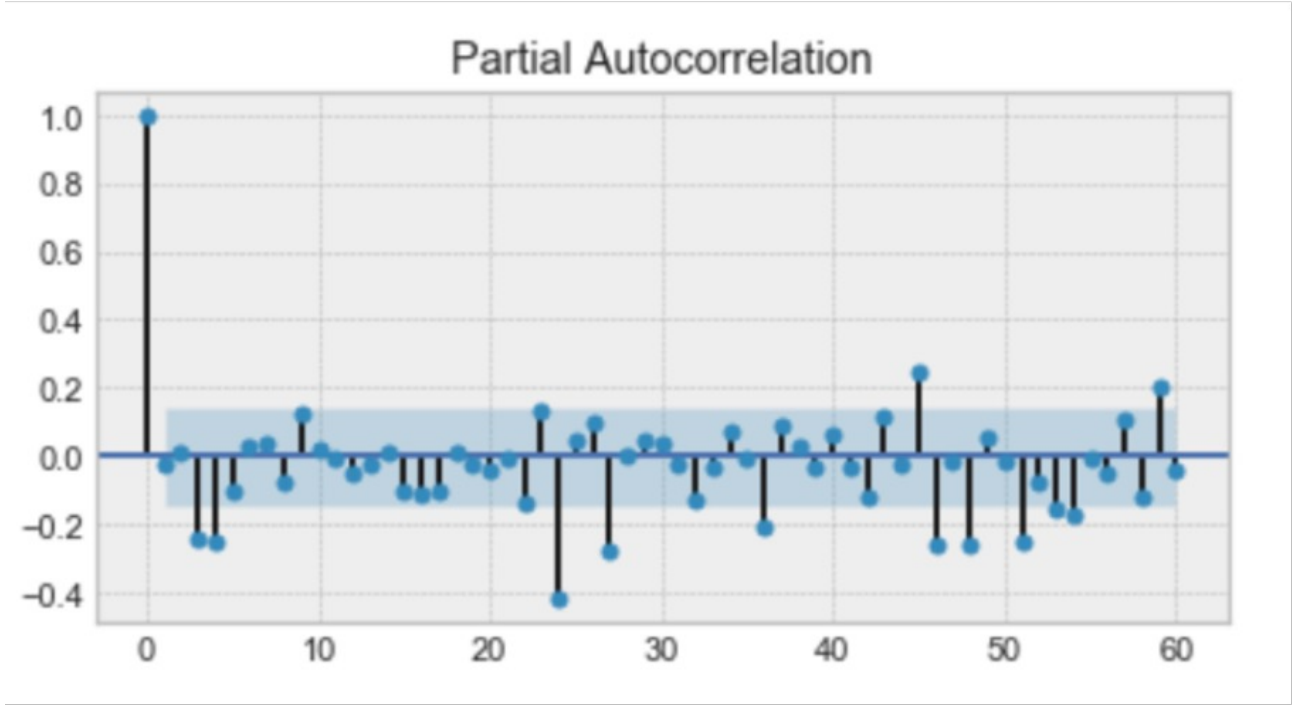
- This is a hint of **seasonality**
- The first value and the 24th value have a high **autocorrelation**
- The 12th and 36th observations are highly **correlated**
- We will find a very similar value at every 24 units of time

PACF plot

- Partial autocorrelation function (PACF): the correlation between the time series and its lag after excluding the contributions from the intermediate lags
  - The partial autocorrelation at lag k is the autocorrelation between X_t and X_{t-k} that is not accounted for by lags 1 through $k-1$
- The PACF plot can provide answers to the following question:
  - Can the observed time series be modeled with an **AR model**? If yes, what is the order?
- The difference between ACF and PACF is the inclusion or exclusion of indirect correlations in the calculation

PACF plot

The PACF plot can be used to identify the parameter p which represents the maximum lag after which most lags are not significant



SARIMA

- At first, we have the autoregression model $AR(p)$
- Then, we add the moving average model $MA(q)$
- After, we add the order of integration $I(d)$
- Finally, we add the final component: seasonality $S(P, D, Q, s)$, where s is simply the season's length
  - Furthermore, this component requires the parameters P and Q which are the same as p and q, but for the seasonal component
  - Finally, D is the order of seasonal integration representing the number of differences required to remove seasonality from the series
- $\boldsymbol{SARIMA(p, d, q)(P, D, Q, s)}$ model

(S)ARIMAX

- But ARIMA models do not fully model reality
- There are often other exogenous variables that affect the time series
- Another class of models that incorporated ARIMA with the explanatory variables approach of standard econometrics
- ARIMAX: ARIMA with additional explanatory variables provided by economic theory
  - X: exogenous variable(s)

M-competitions

- Time series forecast competitions that were held starting from 1982 organized by Spyros Markidakis
- The results of the M4 competition (2018) showed that the traditional statistical approaches largely outperformed the pure ML methods
- However, in the most recent M5 competition (2020), with a more creative dataset, the top spot submissions featured only Machine Learning (ML) and Deep Learning (DL) methods
  - This competition saw the rise of LightGBM (used for time series forecasting) and the debut of Amazon's DeepAR and N-BEATS

Limitations

- The observations from the M4 competition hold but with serious limitations:
  - The amount of data should be limited
  - The time series should be univariate
  - The time series should be stationary or at least not too volatile due to external factors
- But in reality, in the era of Big Data, we rather face large datasets than small, time series in most real world cases are multivariate and external factors influence time series more often than not influence (pandemic, economic crisis, etc.)
- Therefore, the future is with ML and DL time series forecasts methods with classical methods being organically integrated rather than opposed to it

Outline

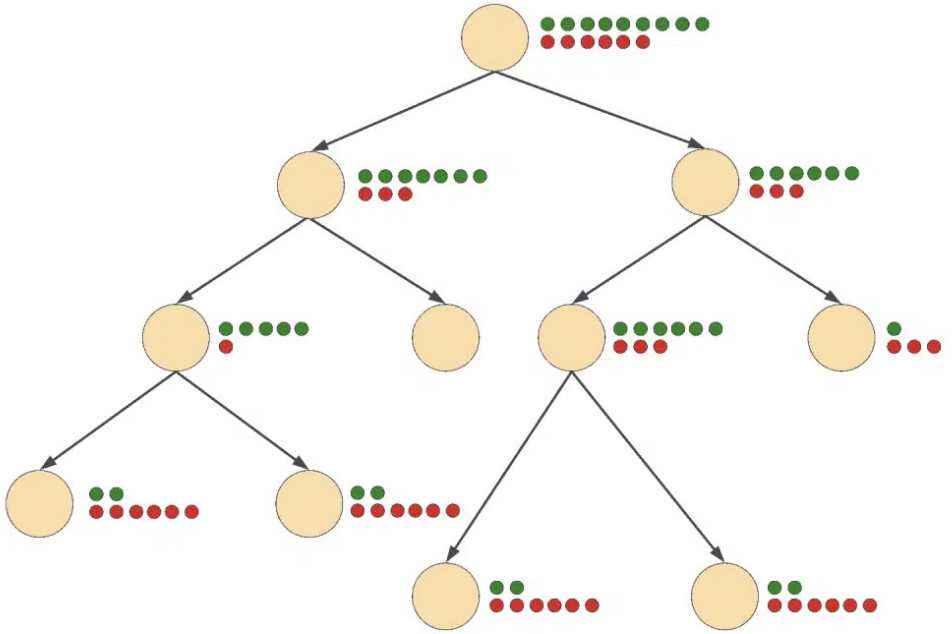
- Intro to Time Series Forecasting
- Statistical Forecast Methods
- Machine Learning Methods
- Deep Learning Models
- Evaluation Metrics
- Nowcasting

Machine Learning Models

- Decision Tree
- Random Forest
- XGBoost
- LightGBM
- Support Vector Regression (SVR)
- Linear Regression
- Ridge Regression
- Lasso Regression
- Elastic Net
- …

Decision Tree

- Decision tree builds regression or classification models in the form of a tree structure
- A regression tree is basically a decision tree that is used for predicting continuous valued outputs instead of discrete outputs
- Due to tree-like structure, decision trees can capture non-linear relationship between variables and therefore can provide good results in forecasting complex data



Decision Tree

- To split the nodes at the most informative features, we need to define an objective function that we want to optimize via the tree learning algorithm
- We want to maximize the information gain at each split, which we define as follows:

$$IG(D_p, f) = I(D_p) - \left(\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right}) \right)$$

- Need for an impurity metric I that is suitable for continuous variables, so we define the impurity measure using the mean squared error (MSE) of the children's nodes:

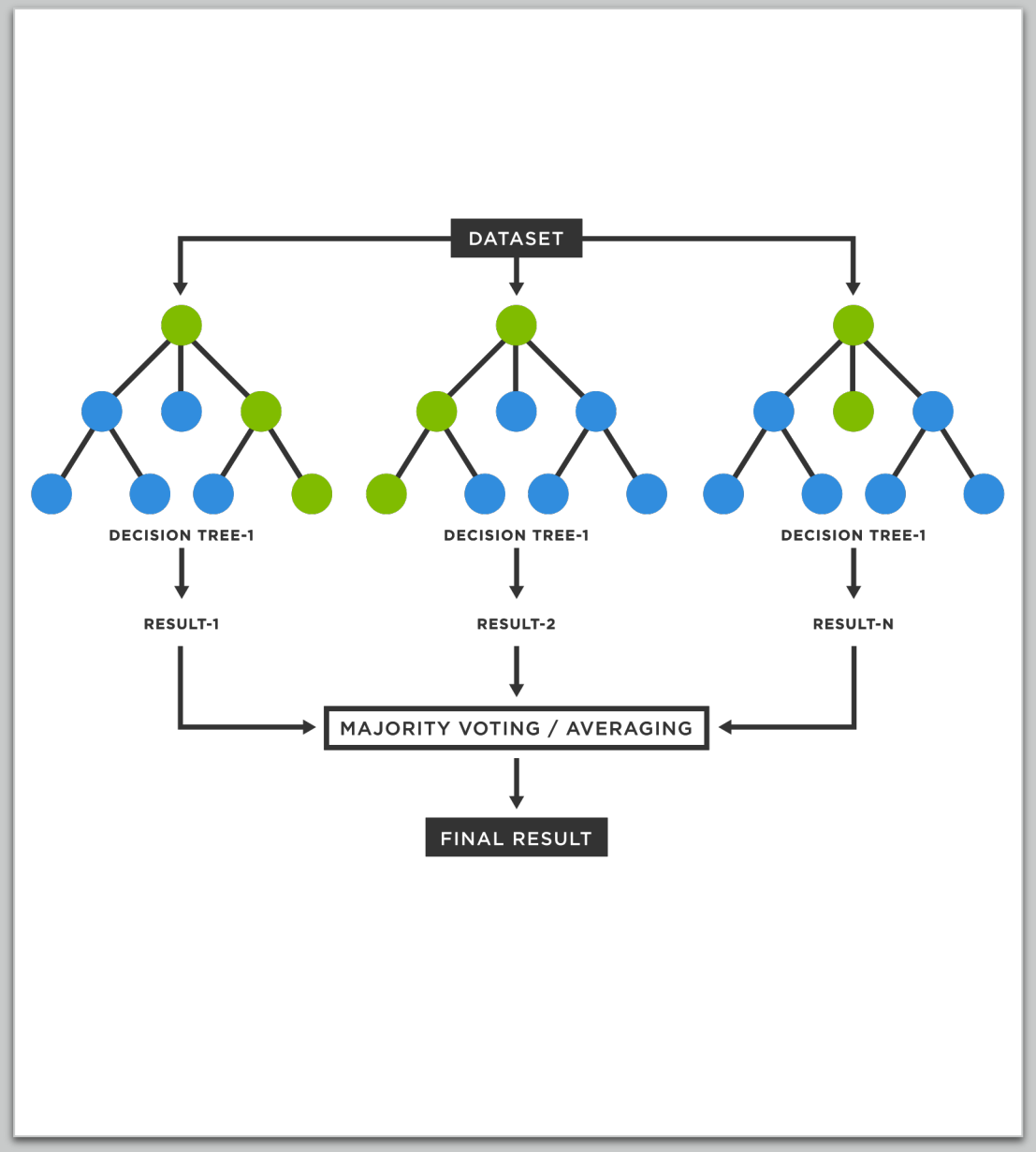
$$MSE(t) = \frac{1}{N_t} \sum_{i \in D_t} (Y^{(i)} - \hat{Y}_t)^2$$

Decision Tree

- Decision trees are very easy to understand and interpret because they can be visualized
- However, they are prone to overfitting
- A decision tree can be considered as a weak learner, which means that the model is simple and has limited predictive power

Random Forest

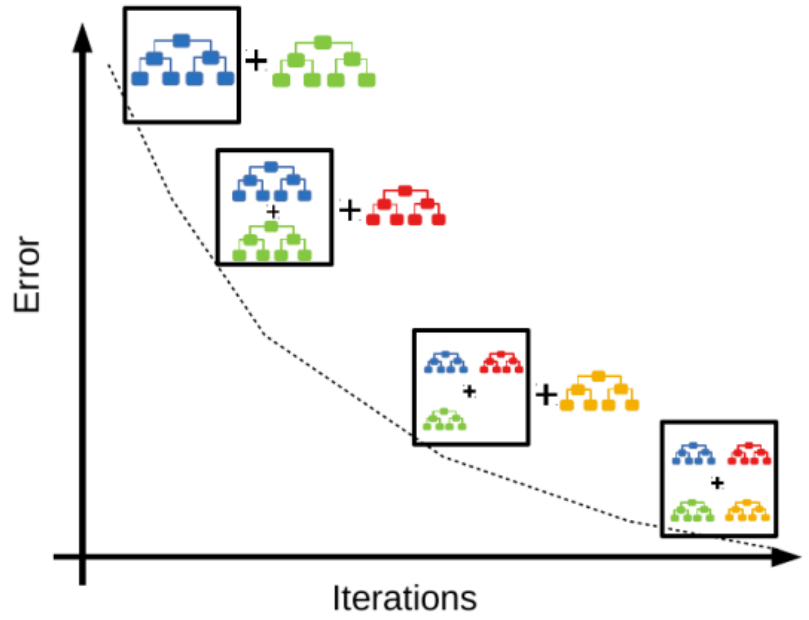
- Ensemble methods in ML are based on combination of outcomes of many weak learners
- Random Forest is called a forest because it grows a forest of decision trees
- Thus, many decision trees can be built using the same dataset, and the final outcome is the average of the predictions of all trees



Random Forest

- Bootstrapping: Random Forest randomly performs row sampling and feature sampling from the dataset forming sample datasets for every model
- Random Forest is very versatile
  - It can used for both classification and regression tasks
  - It can easily handle binary and numerical features as well as categorical ones, with no need for transformation or rescaling
  - Unlike other models, it is incredibly efficient with all types of data
- Random forest prevents overfitting by building different sized trees from subsets and combining the results
- Similarly to Decision Trees, the loss function for the model can be MSE

Gradient Boosting



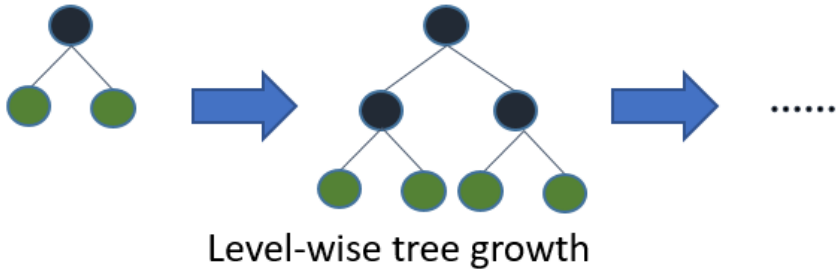
- It is an ML methodology where an ensemble of weak learners is used to improve the model performance in terms of efficiency, accuracy, and interpretability
  - These learners are defined as having better performance than random chance
  - Such models are typically decision trees and their outputs are combined for better overall results
- The hypothesis is to filter out instances that are difficult to accurately predict and develop new weak learners to handle them

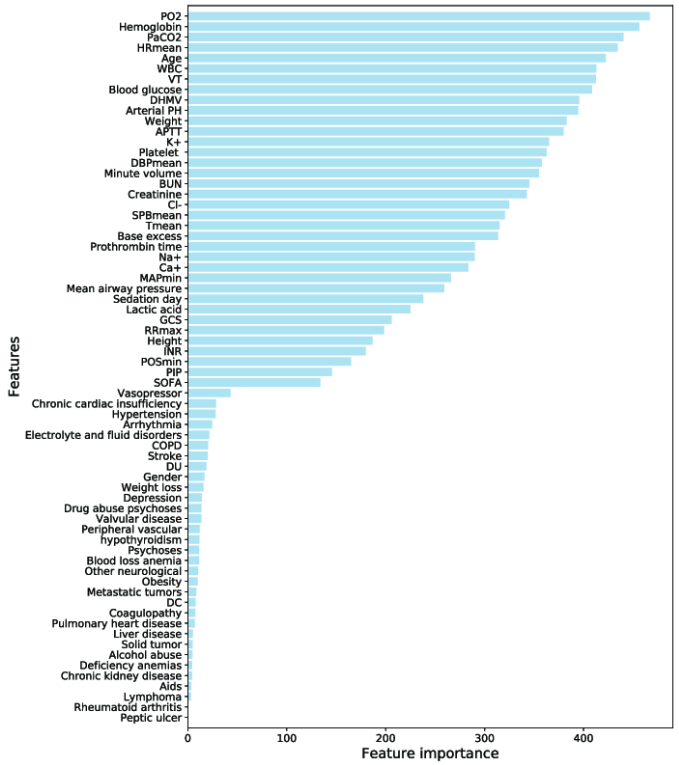
eXrteme Gradient Boosting (XGBoost)

- XGBoost is built on top of gradient boosted ensemble of decision trees
- XGBoost is often used for time series analysis due to its ability to handle non-linear relationship between features
- It also includes regularisation techniques such as L1 and L2 to prevent overfitting
- Tree pruning using depth-first approach
- Utilization of all the available cores of a CPU during tree construction for parallelization

eXrteme Gradient Boosting (XGBoost)

- Level-wise (horizontal) growth





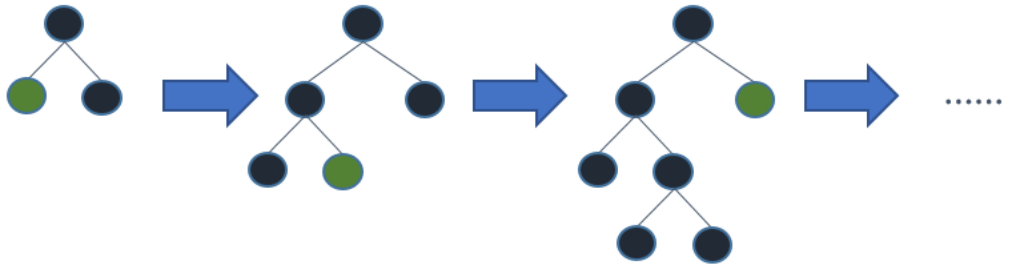
- XGBoost can calculate the importance of every feature, which can be used for feature engineering and explainability of the model

LightGBM

- LightGBM by Microsoft is a distributed high-performance framework that uses decision trees for ranking, classification, and regression tasks
- Histogram-based algorithm that performs bucketing of values, that also requires lesser memory
- Support for both parallel learning and GPU learning
- Faster rate of execution along with being able to maintain good accuracy levels primarily due to the utilization of two novel techniques:
  - **Gradient-Based One-Side Sampling (GOSS)**: data instances with larger gradients contribute more towards information gain and and in addition, it performs random sampling on instances with smaller gradient
  - **Exclusive Feature Bundling (EFB)**: a near lossless method to reduce the number of effective features

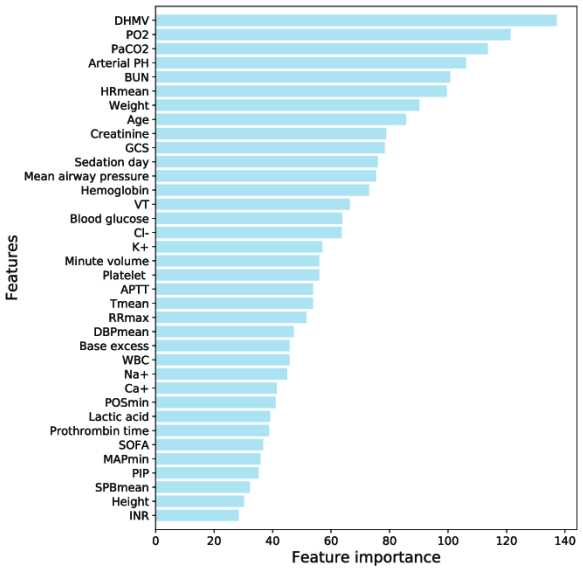
LightGBM

- In contrast to the level-wise (horizontal) growth in XGBoost, LightGBM carries out leaf-wise (vertical) growth that results in more loss reduction and in turn higher accuracy while being faster



Leaf-wise tree growth

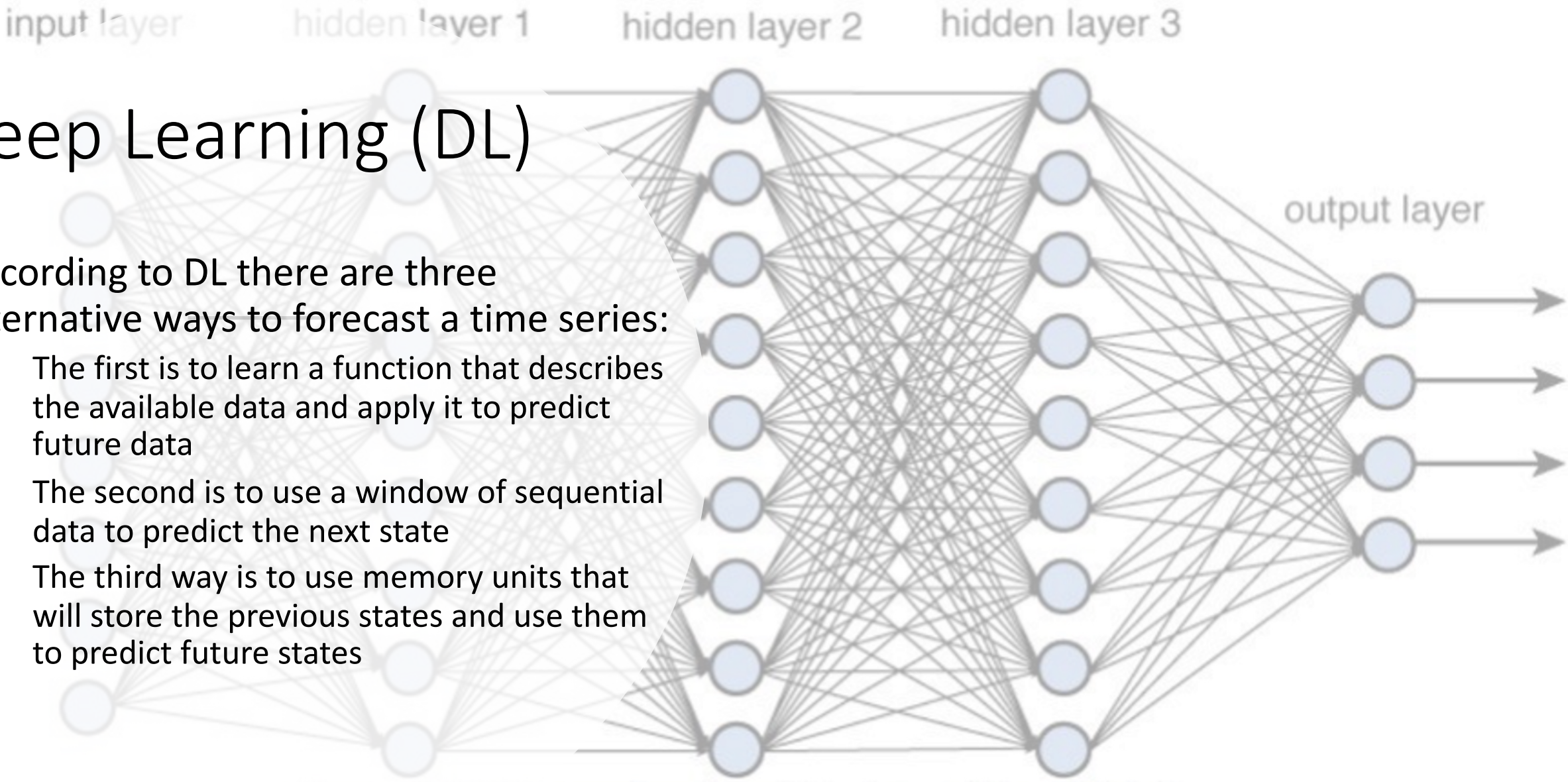
- LightGBM can also calculate the importance of every feature for making the predictions



Outline

- Intro to Time Series Forecasting
- Statistical Forecast Methods
- Machine Learning Methods
- Deep Learning Models
- Evaluation Metrics
- Nowcasting

Deep Learning (DL)



According to DL there are three alternative ways to forecast a time series:

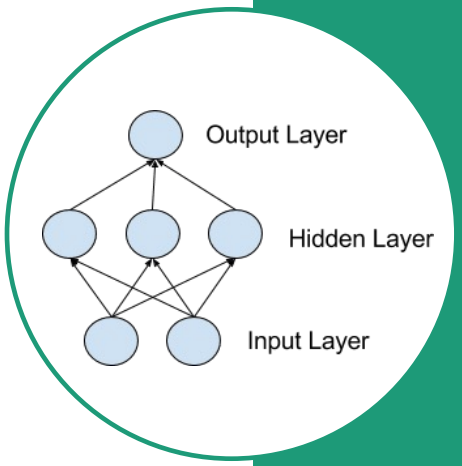
- The first is to learn a function that describes the available data and apply it to predict future data
- The second is to use a window of sequential data to predict the next state
- The third way is to use memory units that will store the previous states and use them to predict future states

Deep Learning Models

- Multilayer Perceptrons (MLPs)
- Convolutional NNs (CNNs)
- Recurrent NNs (RNNs)
  - LSTM, GRU
- N-BEATS
- DeepAR
- Encoder-Decoder Model
- Attention Mechanism
- …

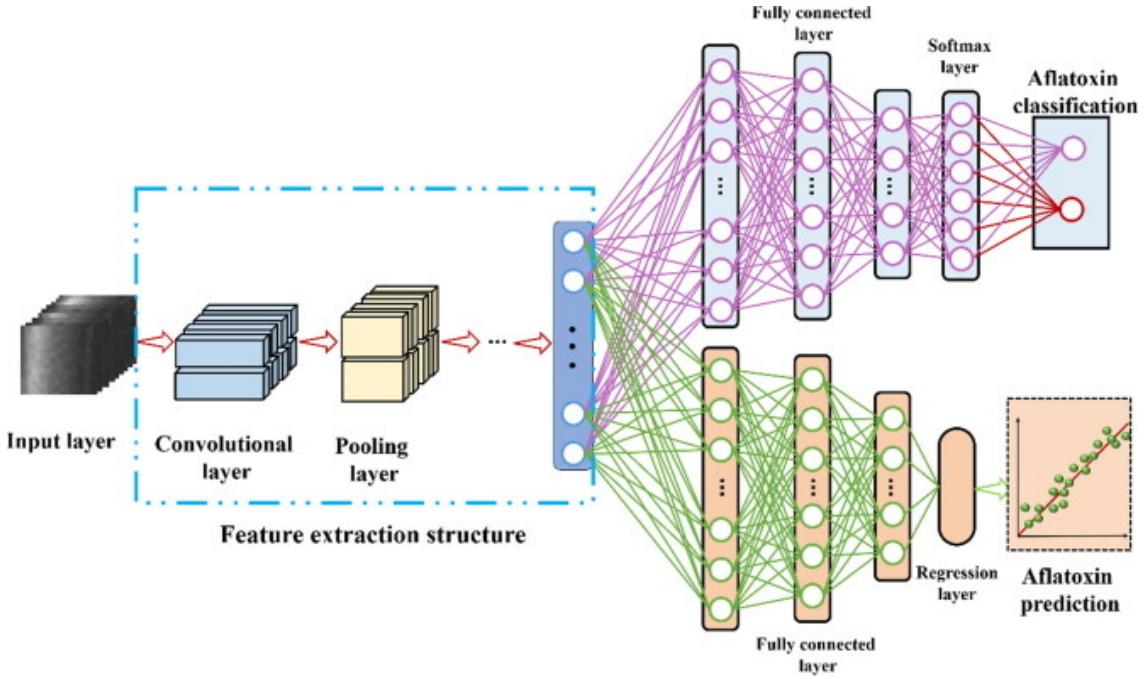
Multilayer Perceptrons (MLPs)

- MLPs are the classical type of Neural Network
- MLP models are memory-less, and they use the feed forward neural network architecture, which applies the back propagation algorithm for training the neural network
- Usually fully-connected networks, each neuron in one layer is connected to all neurons of the next layer
  - “full connectivity” makes them prone to overfitting
- MSE as the regression loss



Convolutional Neural Networks (CNNs)

- CNNs are regularized versions of multilayer perceptrons
- A CNN is a neural network with convolutional layers
  - A convolutional layer has a number of filters that perform the operation of convolution (an integral that expresses the amount of overlap of one function as it is shifted over another function)
- They take advantage of local and hierarchical patterns in the data



Implementing a CNN for regression prediction is as simple as:

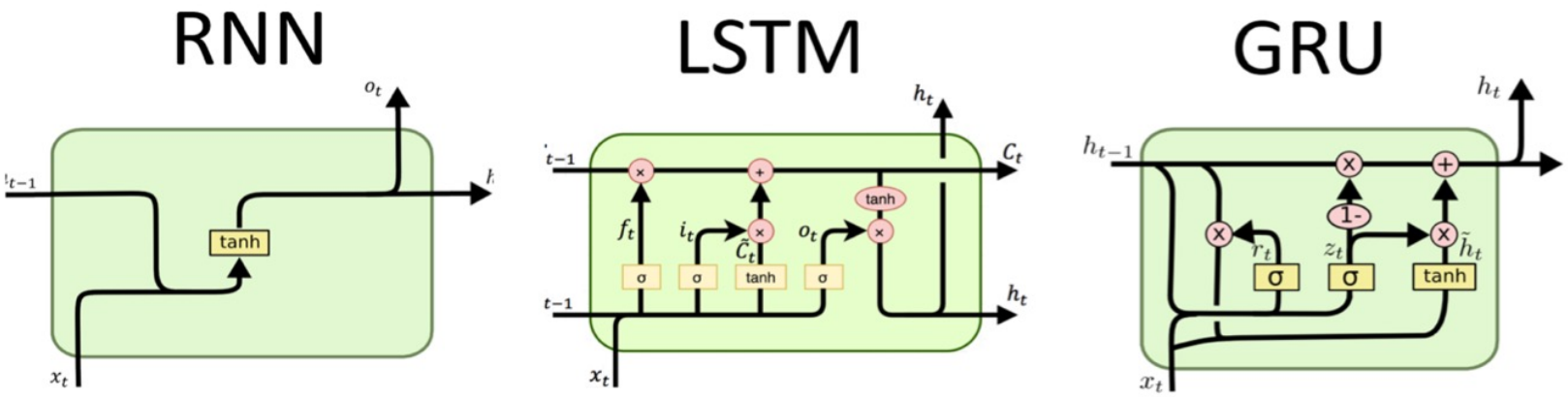
- Replacing the fully-connected softmax classifier layer (classification) with a fully-connected layer with a single node along with a linear activation function
- Training the model with a continuous value prediction loss function such as MSE

Recurrent Neural Networks (RNNs)

- RNNs are deep learning models, typically used to solve problems with sequential input data, such as time series
- They can have multiple hidden layers to solve more complex sequential problems
- They retain a memory of what it has already processed and, therefore, a sense of time
  - Achieved by implementing several neurons receiving as input the output of one of the hidden layers and injecting their output again in that layer
- They share the same weights across several time steps

Gated RNNs

- The most effective RNNs are the Gated RNNs:
  - **Long Short-Term Memory (LSTM)**: more accurate with longer sequences
  - **Gated Recurrent Unit (GRU)**: faster than LSTM with shorter sequences
- They allow the network to accumulate information over a long duration
- Once this information has been used the neural network can forget the old state
- Instead of manually deciding when to clear the state, we want the neural network to learn to decide when to do it



Long Short-Term Memory (LSTM)

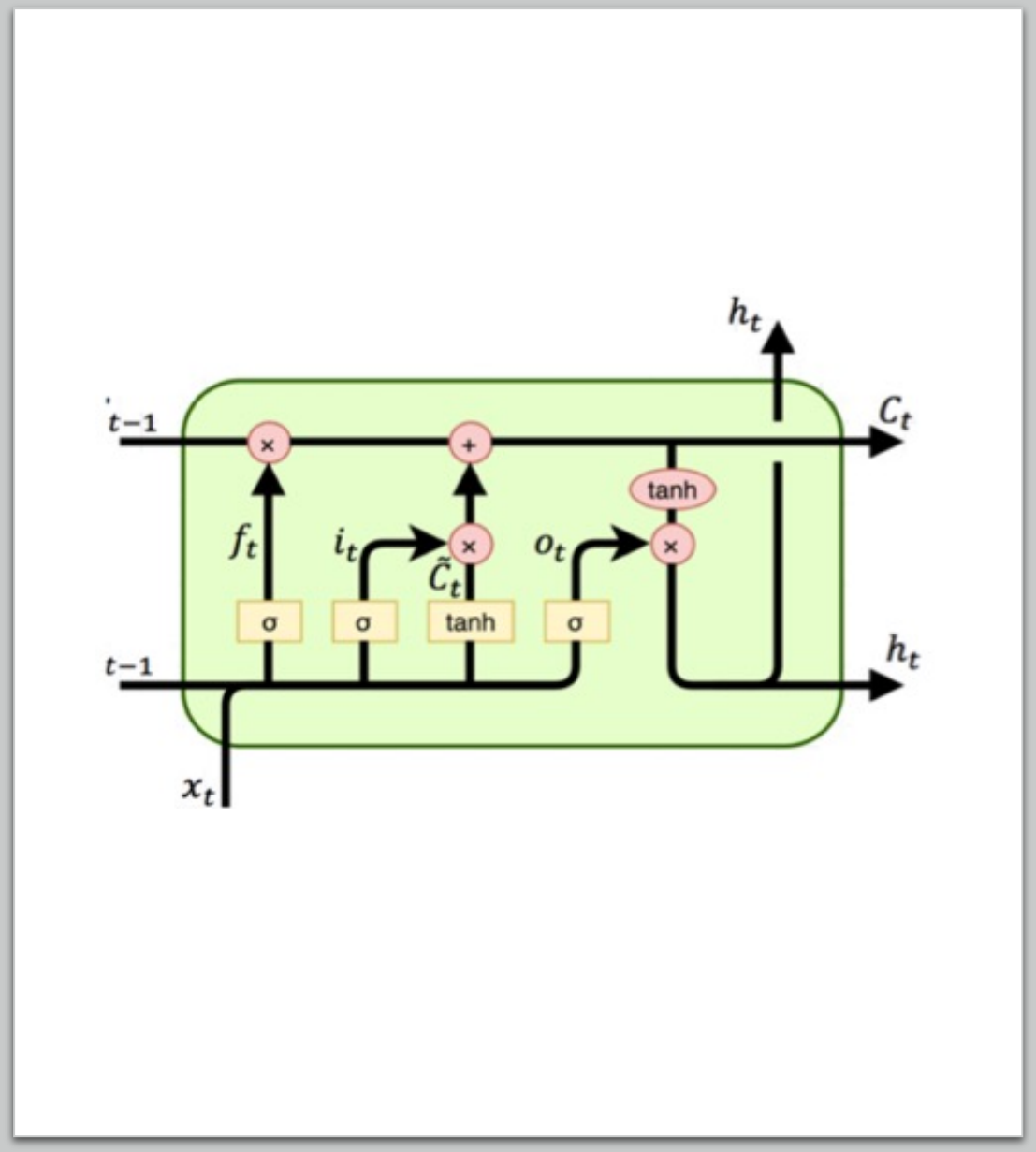
- LSTM is a type of recurrent neural network that can learn the order dependence between items in a sequence
- LSTM architecture allows it to be effective for time series forecasting due to its ability to capture short term and long-term dependencies and complex non-linear relationships
- It makes predictions according to the data of previous times
- LSTM works well for multivariate time series and can handle missing data
- Pre-processing data is a very important step for the performance of an LSTM

LSTM

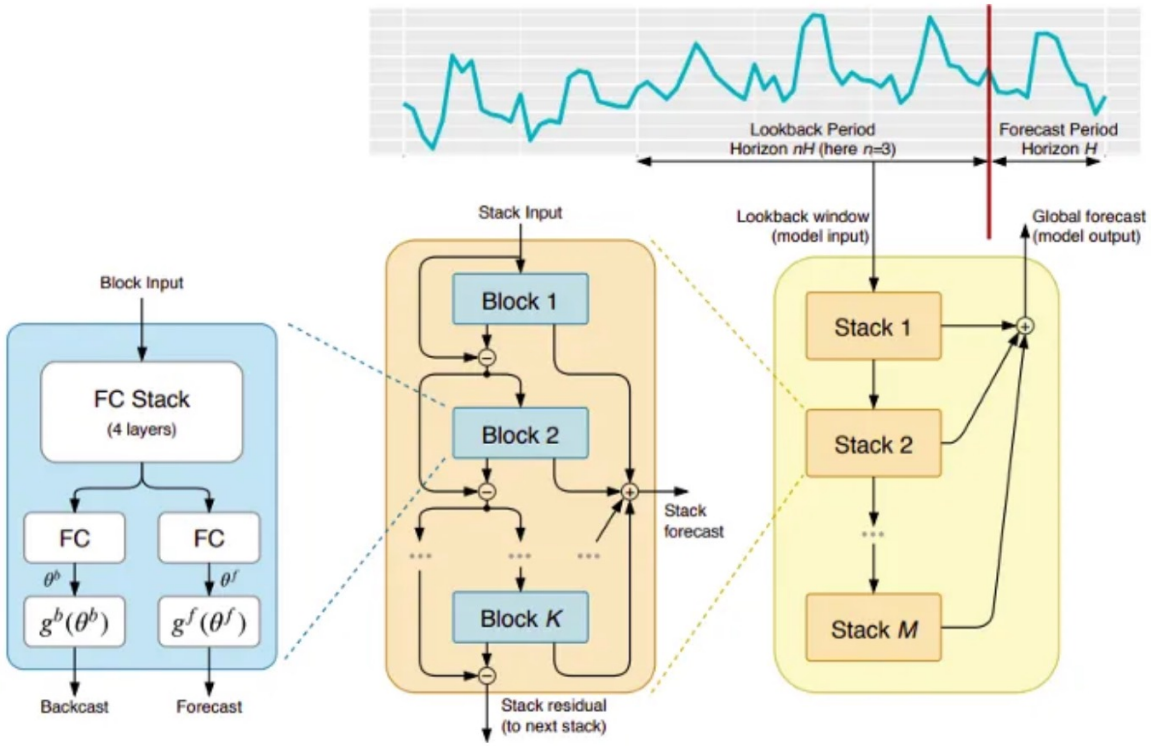
LSTM consists of three gates that control flow of information and maintain long-term dependencies in the data

- **Input gate**: responsible for the information that can enter the cell state
- **Forget gate**: controls how much of previous information should be ignored and how much should be kept
- **Output gate**: decides the information that should be sent to the next state

The combination of the gates allows LSTM to remember information over longer periods and therefore makes LSTM suitable for processing time series data



N-BEATS



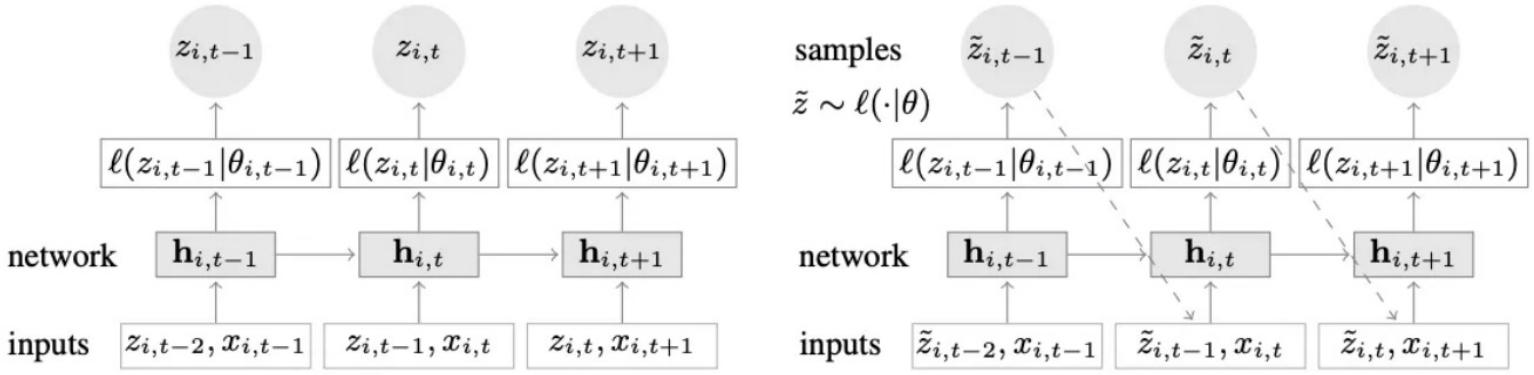
- N-BEATS is a custom DL algorithm that is based on backward and forward residual links for univariate time series point forecasting
- Essentially, N-BEATS is a pure DL architecture based on a deep stack of ensembled feed-forward networks that are also stacked by interconnecting backcast and forecast links
- Each successive block models only the residual error due to the reconstruction of the backcast from the previous block and then updates the forecast based on that error

N-BEATS

- Simple to understand and has a modular structure (blocks and stacks)
- It has the ability to generalize on multiple time-series through meta-learning
  - Inner learning: takes place inside blocks and helps the model capture local temporal characteristics
  - Outer learning: takes place inside stacks and helps the model learn global characteristics across all time-series
- The model has two variants:
  - General: the final weights in the fully-connected layers of each block are learned by the network arbitrarily
  - Interpretable: the last layer of each block is removed, then the backcast and forecast branches are multiplied by specific matrices that mimic trend (monotonic function) and seasonality (periodic cyclical function)

DeepAR

- DeepAR developed by Amazon is a novel probabilistic forecasting model based on autoregressive RNNs
- It combines both DL and autoregressive characteristics
- No need to scale the time sequence using a normalization or standardization technique as in other DL models, since DeepAR scales the autoregressive input z of each time series i with a scaling factor v_i



DeepAR

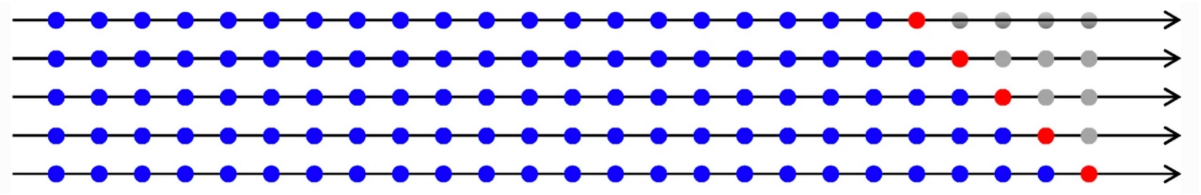
- DeepAR works really well with multiple time series
  - A global model is built by using multiple time series with slightly different distributions
- Apart from historical data, DeepAR also allows the use of known future time sequences (a characteristic of auto-regressive models) and extra static attributes for series
- It makes probabilistic forecasts instead of directly outputting the future values
  - in the form of Monte Carlo samples
  - these forecasts are used to compute quantile forecasts, by using the quantile loss function, used to calculate not only an estimate, but also a prediction interval around that value

Outline

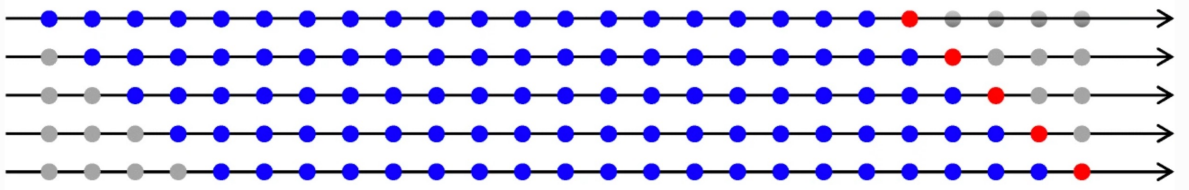
- Intro to Time Series Forecasting
- Statistical Forecast Methods
- Machine Learning Methods
- Deep Learning Models
- Evaluation Metrics
- Nowcasting

Rolling Forecast: Cross validation & Predictions

1. **Minimum number of observations** required to train the model
2. **Sliding window**:
   - the model will be trained on all the data it has available



   - the model will be trained only on the most recent observations



Evaluation Metrics

- We cannot calculate accuracy for a forecasting model. The performance of the model must be reported as an error for the predictions
- Four commonly used error metrics for evaluating and reporting the performance of a forecasting model:
  - **Mean Squared Error (MSE)**
  - **Root Mean Squared Error (RMSE)**
  - **Mean Absolute Error (MAE)**
  - **Mean Absolute Percentage Error (MAPE)**
- In addition, we calculate the correlation between the predictions and the real values:
  - **Pearson's Correlation**
  - **Spearman's Rank Correlation**

Error Metrics

Acronym	Full Name	Formula	Units?	Residual Operation?	Robust to Outliers?
MSE	Mean Squared Error	$\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}_i)^2$	Squared units	Square	No
RMSE	Root Mean Squared Error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Yi - \hat{Y}_i)^2}$	Same as target	Square	No
MAE	Mean Absolute Error	$\frac{1}{n}\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert$	Same as target	Absolute Value	Yes
MAPE	Mean Absolute Percentage Error	$\frac{100\%}{n}\sum_{i=1}^{n}\frac{\lvert Y_i - \hat{Y}_i\rvert}{Y_i}$	Percentage	Absolute Value	Yes

Pearson's Correlation Coefficient

- It attempts to establish a line of best fit between two variables
- The resulting Pearson's correlation coefficient indicates how far away the actual values are from the mean values:

$$r_{XY} = corr(X, Y)$$

- The value of r_{XY} ranges between −1 and +1
  - the closer it is to either −1 or 1, the stronger the correlation between the variables
  - as it approaches 0 there is less of a relationship
- If X and Y independent: $r_{XY} = 0$
  - X and Y independent and uncorrelated

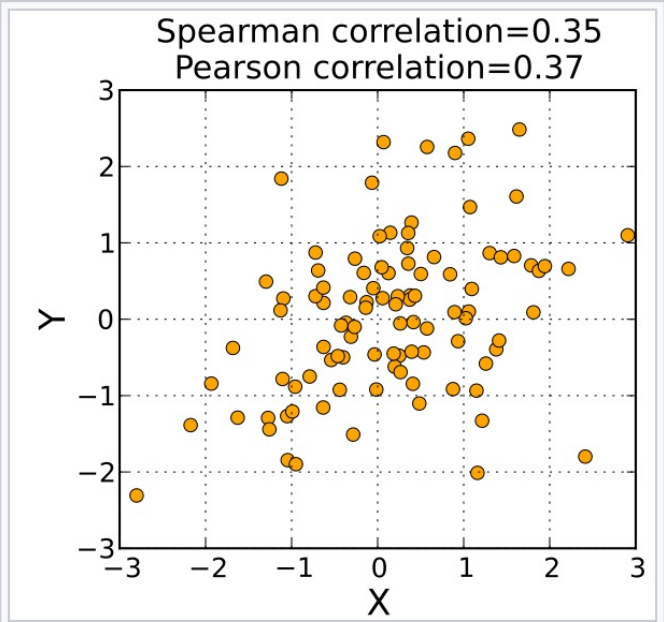
Correlation Coefficient Value	Indication
$\pm 0.8\ to\ \pm\ 1.00$	High correlation
$\pm 0.6\ to\ \pm\ 0.79$	Moderately high correlation
$\pm 0.4\ to\ \pm\ 0.59$	Moderately correlation
$\pm 0.2\ to\ \pm\ 0.39$	Low correlation
$\pm 0.1\ to\ \pm\ 0.19$	Negligible correlation

Spearman's Rank Correlation Coefficient

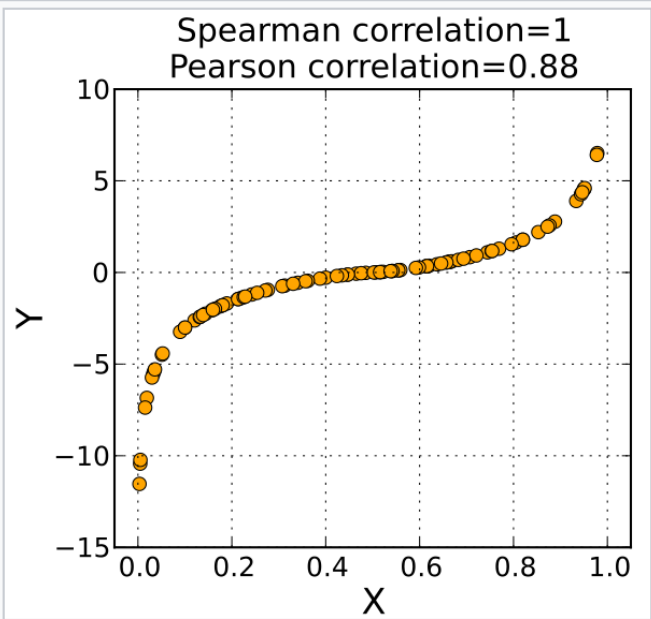
- It measures the extent to which, as one variable increases, the other variable tends to increase, without requiring that increase to be represented by a linear relationship
- The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the rank variables $R(X)$ and $R(Y)$:

$$r_s = r_{R(X),R(Y)}$$

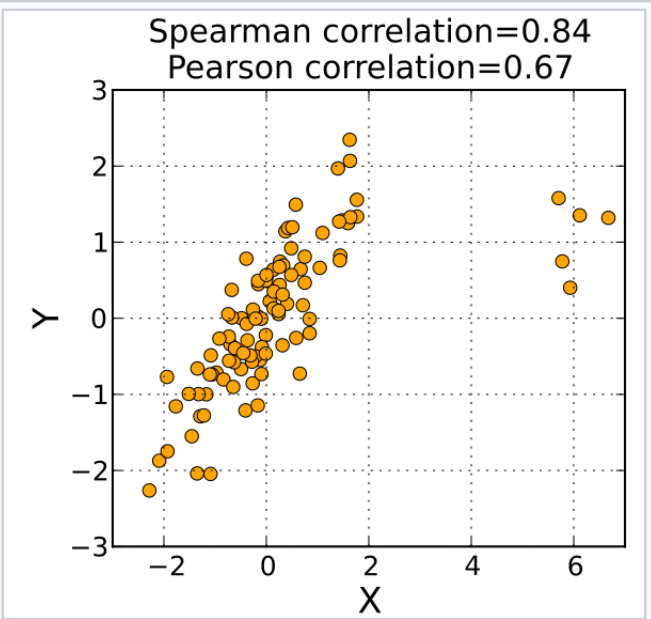
- If, as the one variable increases, the other decreases, the rank correlation coefficient will be negative



When the data are roughly elliptically distributed and there are no prominent outliers, the Spearman correlation and Pearson correlation give similar values.



A Spearman correlation of 1 results when the two variables being compared are monotonically related, even if their relationship is not linear. This means that all data points with greater *x* values than that of a given data point will have greater *y* values as well. In contrast, this does not give a perfect Pearson correlation.



The Spearman correlation is less sensitive than the Pearson correlation to strong outliers that are in the tails of both samples. That is because Spearman's ρ limits the outlier to the value of its rank.

Pearson's Vs Spearman's

Outline

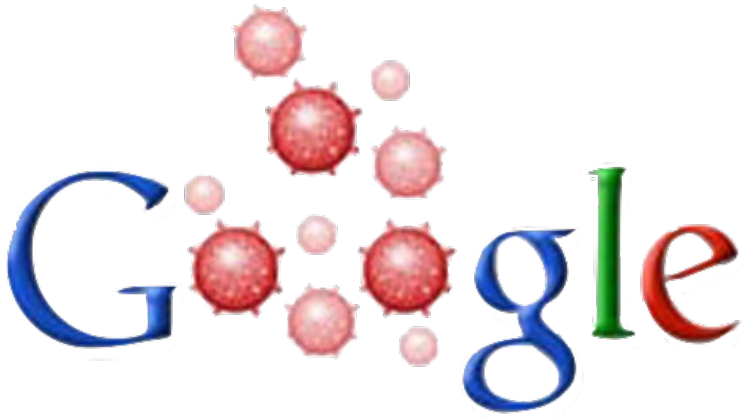
- Intro to Time Series Forecasting
- Statistical Forecast Methods
- Machine Learning Methods
- Deep Learning Models
- Evaluation Metrics
- Nowcasting

Nowcasting

- Predict the short-term future values given past observations of several indicators
- Early detection of a critical event
- Widely used by epidemiologists since they are interested in timelier forecasts of disease spreading

Nowcasting problem

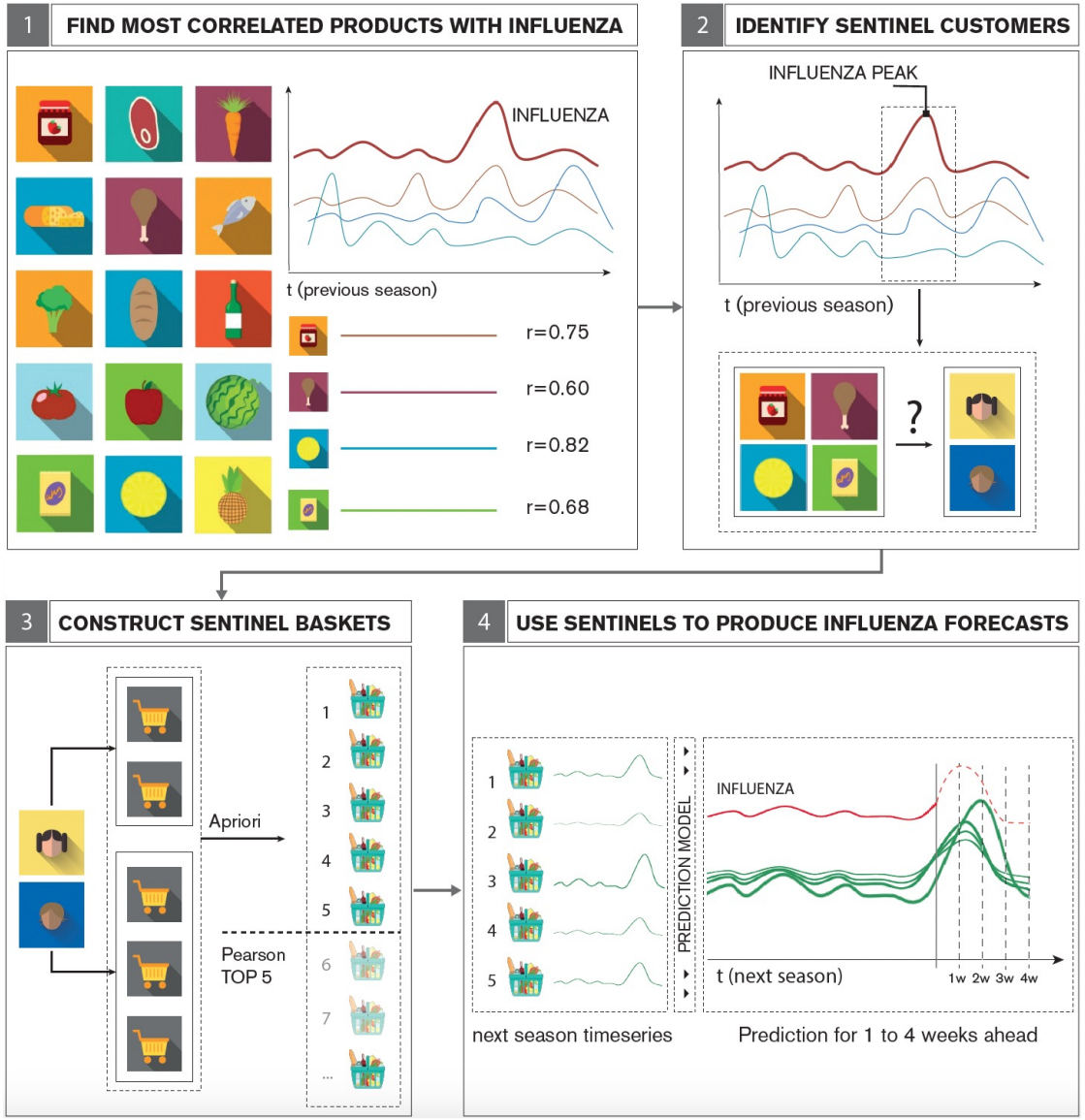
- Time **dependent** variable *Y(t)* that cannot be measured instantaneously
- Value of *Y(t)* available at time *t+d*, where *d* is the delay
- We approximate the value of *Y(t)* at time *t*, based on a set of *n* **independent** variables (**predictors**) $X_1(t), X_2(t), \ldots X_n(t),$ which can be measured at time *t* or time $t+d_{X_{1,2,\ldots n}}$ where $d_{1,2,\ldots n} < d$



Google Flu Trends

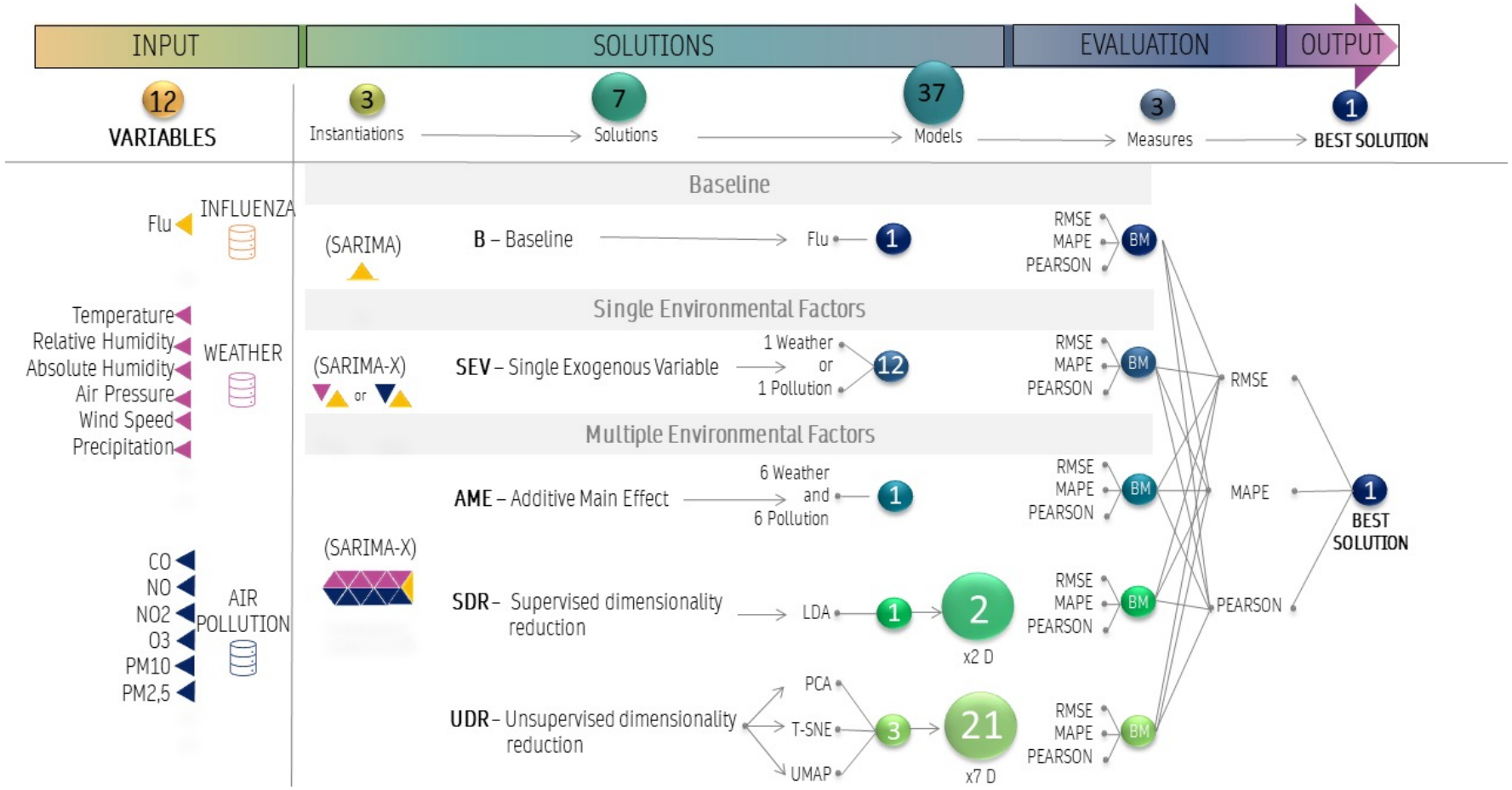
- Search data can help predict the incidence of influenza-like diseases
- Close relationship between number of people searching for flu-related topics and how many people have symptoms
- Prediction models compared to real-world cases of flu

Predicting seasonal influenza using supermarket retail records



Miliou et. al., **Predicting seasonal influenza using supermarket retail records**, PLOS Computational Biology, 2021

Impact of dimensionality on nowcasting seasonal influenza with environmental factors



Guarnizo et. al., **Impact of dimensionality on nowcasting seasonal influenza with environmental factors**, IDA, 2022

Questions?

Thank you

Ioanna Miliou, PhD

ioanna.miliou@dsv.su.se

Stockholm University