Time Series Forecasting

UA DataLab

Shravan Aras, Ph.D.

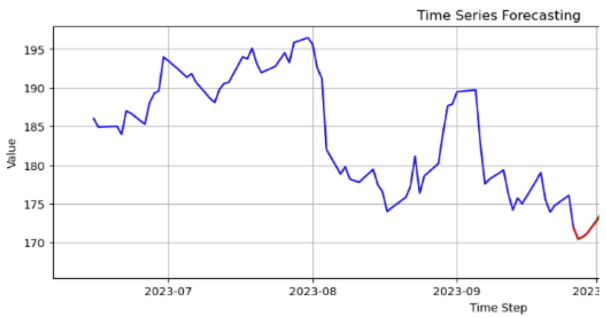
(3-18-2024)

Outline

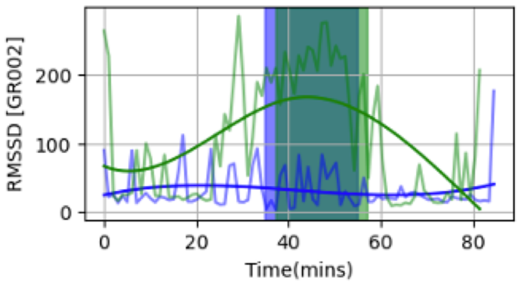
- What is timeseries data
- Different types of timeseries data
- Forecasting – Definition and Illustration
- Forecasting methods for stationary
- Forecasting methods for non-stationary
- Using RNN for forecasting
- Using LSTM for forecasting

What is timeseries data?

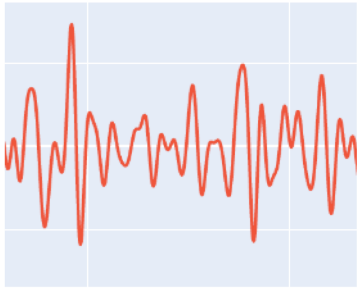
- Sequence of data points
- Successive points in time
- Preferably with a regular period / regular interval



Stock Market



HRV



BVP

Characteristics of time series data

- **Temporal Dependence** –
  - Values are not independent. Y(t) is related to a previous value Y(t-n).
- **Seasonality** –
  - Periodicity or having a time periods – seconds, minutes, hours, years or decades.
- **Trend** -
  - DC component or mean, upward or downward trend over time.
- **Noise** -
  - Variability in the time series.

Is this time series data?

- Tossing a coin multiple times?
  - H T H H T T
- Dice roll
  - 3 2 1 1 1 6 2
- Random Walk
  - Y(t) = Y(t-1) + Random

Is this time series data?

- Tossing a coin multiple times?
  - H T H H T T
- Dice roll
  - 3 2 1 1 1 6 2
- Random Walk
  - Y(t) = Y(t-1) + Random

Types of Timeseries

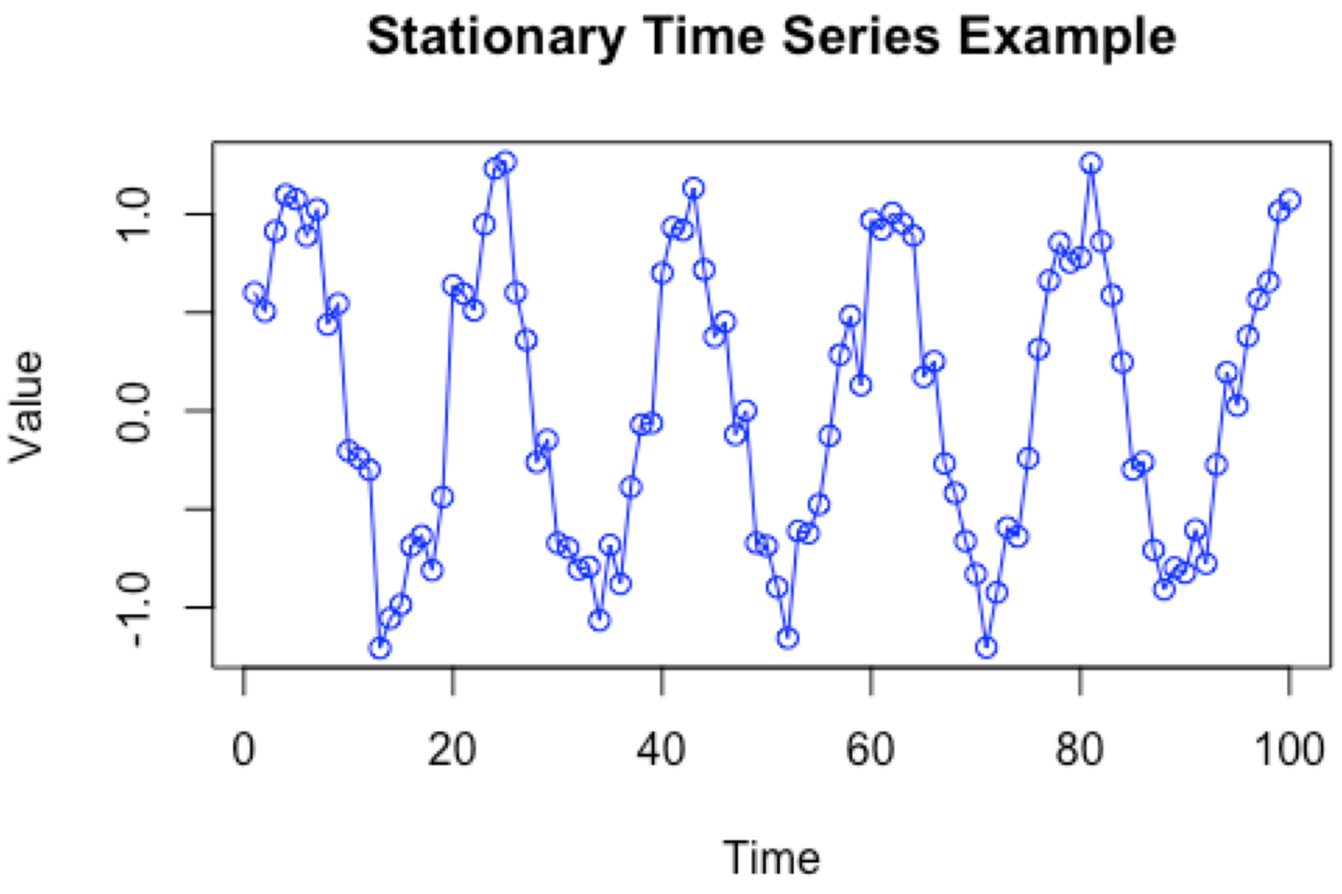
- Stationary Time Series
  - Mean, variance and autocorrelation are constant
  - Fluctuates around a constant mean and variance

- Non-Stationary Time Series
  - Statistical properties change over time.
  - They have –
    - Trending -> Mean changes
    - Seasonality -> Periodic change

Examples of the 2 Types

- Stationary
  - Periodicity from a sine wave
  - Fluctuates around the mean of 0 and std = 0.2
  - Mean and variance remains same throughout

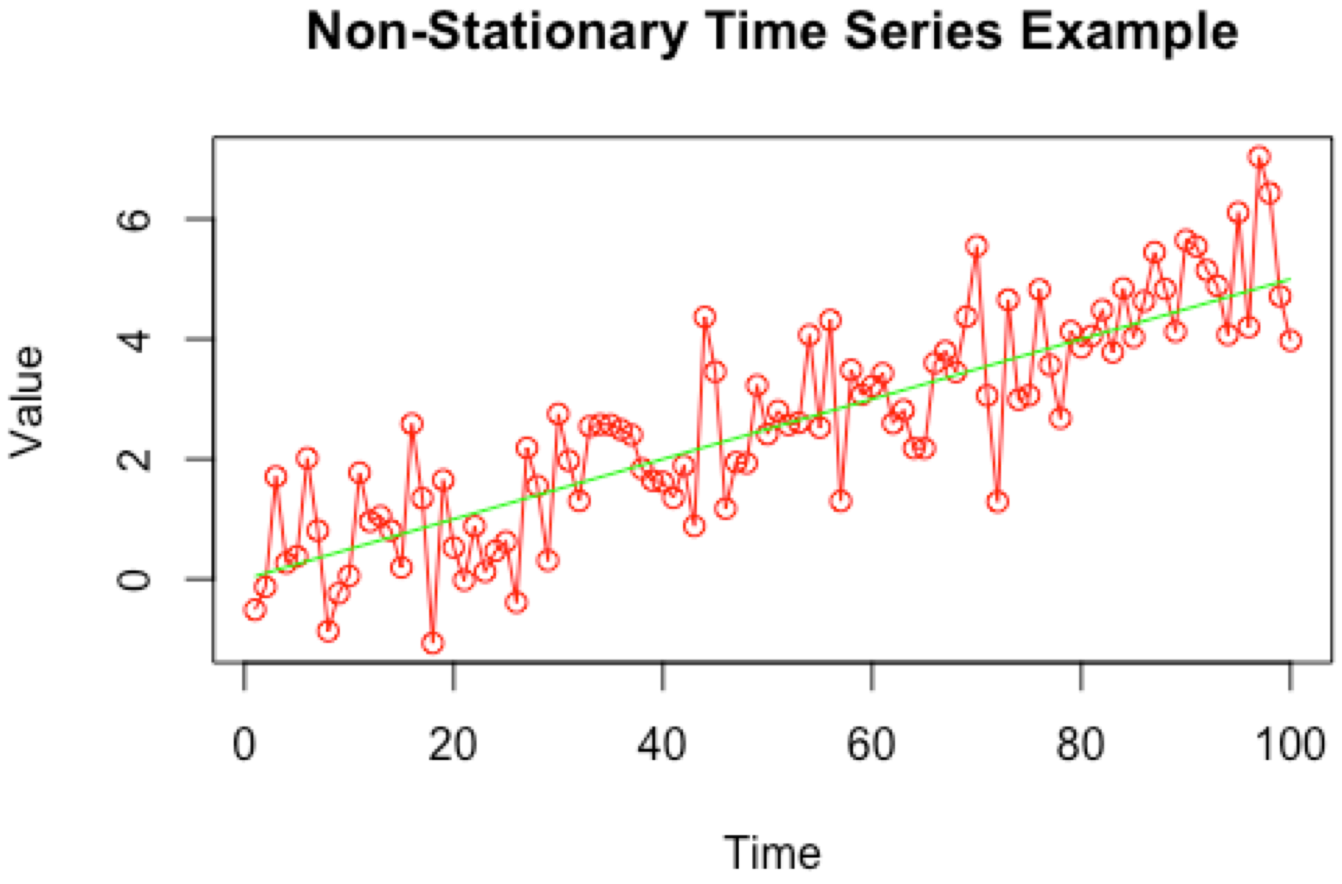
Observe we don't see any DC component or trend overtime.



Generated in R – code/ stationary-example.R

Example of the 2 Types

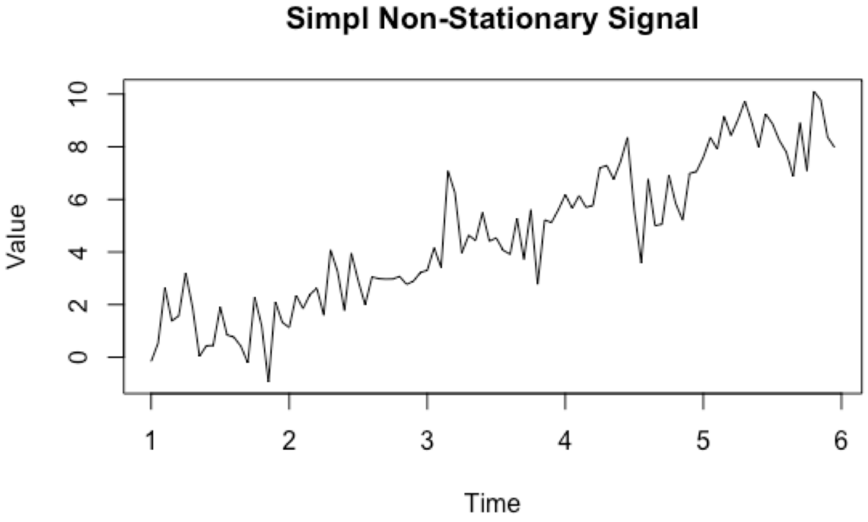
- Non-Stationary
  - Has a linear trend with slope of 0.05
  - Combines the linear trend with noise data
  - Observe how mean / DC component of the signal changes over time.
  - Makes it harder to build a forecasting model since the mean changes.
  - Most signals in real-life.

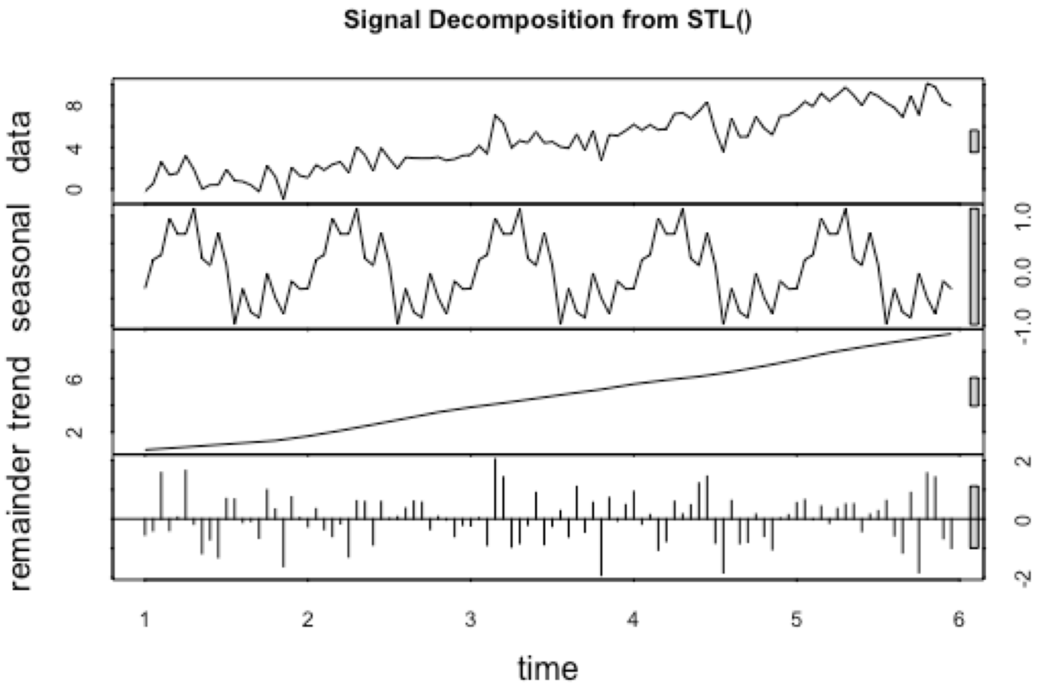


Generated in R – code/non-stationary-example.R

Decompose the time series.

- Using the STL() method in R to isolate seasonality, trend and irregular





Generated in R – code/decompose.R

Testing stationarity / lack of.

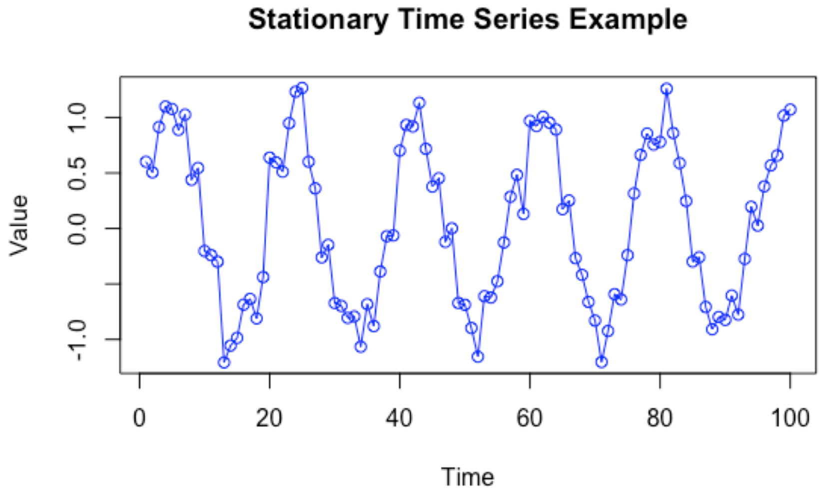
- Do a visual inspection → Plot your data and stare at it.
- Decompose the signal as shown earlier.
- Use domain knowledge, is the unkown system generating this likely to be stationary or not.
- Use statistical tests -
  - ADF (Augmented Dickey-Fuller) Test
  - KPSS (Kwiatkowski-Phillips-Schmidt-Shin) Test

Statistical Tests

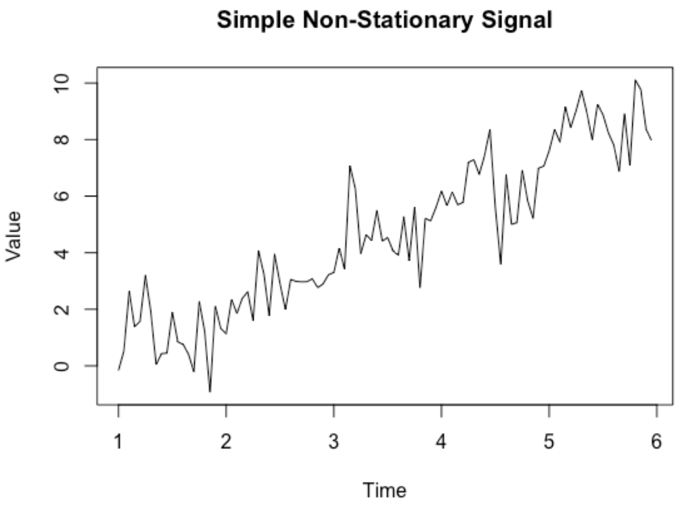
- ADF
  - Null : Unit root is present in AR model → implies non-stationarity
  - $P < 0.05$ → We reject the null hypothesis → signal is stationary
  - R package : “tseries”, adf.test()

- KPSS
  - Null : The signal is stationary
  - $P < 0.05$ → We reject the null hypothesis → signal is non-stationary
  - R package : “tseries”, kpss.test()

Examples



Augmented Dickey-Fuller Test
p-value < 0.01
alternative hypothesis: stationary
KPSS Test for Level Stationarity
p-value > 0.1

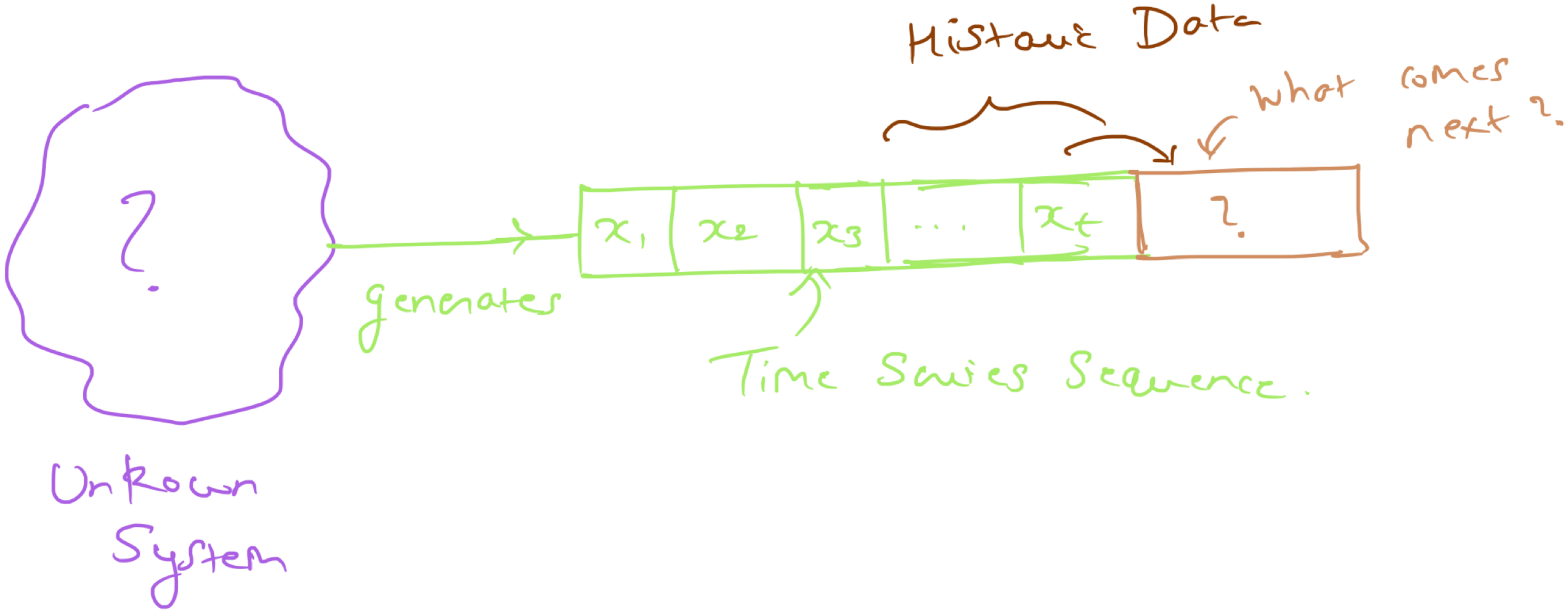


Augmented Dickey-Fuller Test
p-value < 0.03
alternative hypothesis: stationary
KPSS Test for Level Stationarity
p-value < 0.01

Generated in R – code/statisticaltest.R

Problem Illustration

- Given a sequence of data-points, we must predict what comes next.



Problem Formalization

X_t : $\{x_t, x_{t-1}, x_{t-2} \cdots x_{t-l}\}$. Past Values.

h: Forecasting Horizon. How far into the future.

l: Lag. How many historic points to consider.

Y_{t+h} : $\{y_t, y_{t+1}, \cdots y_{t+h}\}$. Future values.

f_h : Forecasting Function

U_{t+h} : Penalization OR bias vector.

$$\boxed{Y_{t+h} = f_h(X_t) + U_{t+h}}$$

Formulation of prediction accuracy

$f_h \rightarrow$ Is unkown.

$\hat{f}_h \rightarrow$ Estimate of f_h from data.

$$\underbrace{\Delta(x_t)}_{\text{Prediction Error}} = |\underbrace{\hat{f}_h(x_t)}_{\text{Predicted Value}} - \underbrace{f_h(x_t)}_{\text{Actual value.}}|$$

Common Errors – L1 and L2 norm

Mean Absolute Prediction Error (MAPE)

$$\Delta(x_t) = \frac{\sum \left| \hat{f}_h(x_t) - f_h(x) \right|}{N}$$

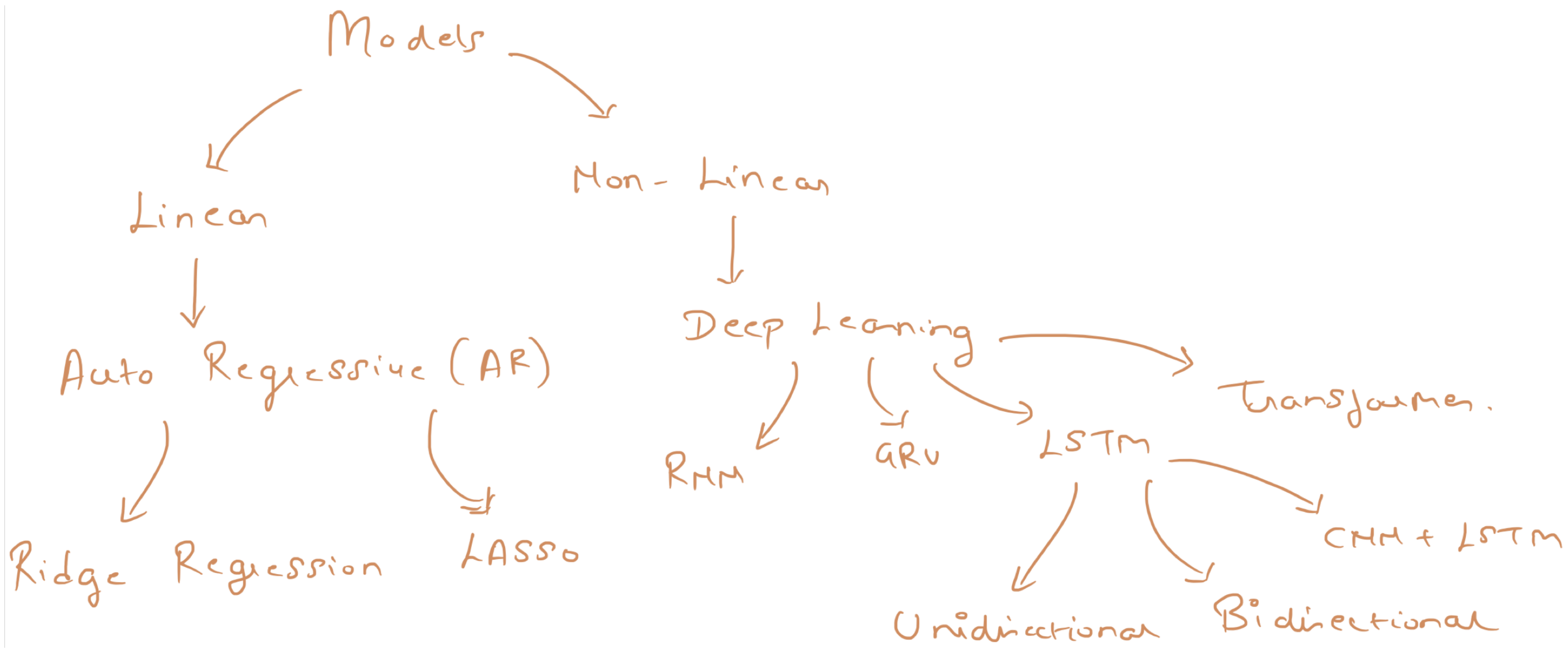
L1

Mean Squared Prediction Error (MSPE)

$$\Delta(x_t) = \frac{\sum \left(\hat{f}_h(x_t) - f_h(x) \right)^2}{N}$$

L2

Forecasting options



AR – For Stationary Signals

- Fundamental Auto-Regressive Model
  - Regresses on itself.

- AR(p) -
  - P is the order of the model, how far back
  - AR(1) – Look one timestep behind
  - R can find the best order using a metric like AIC

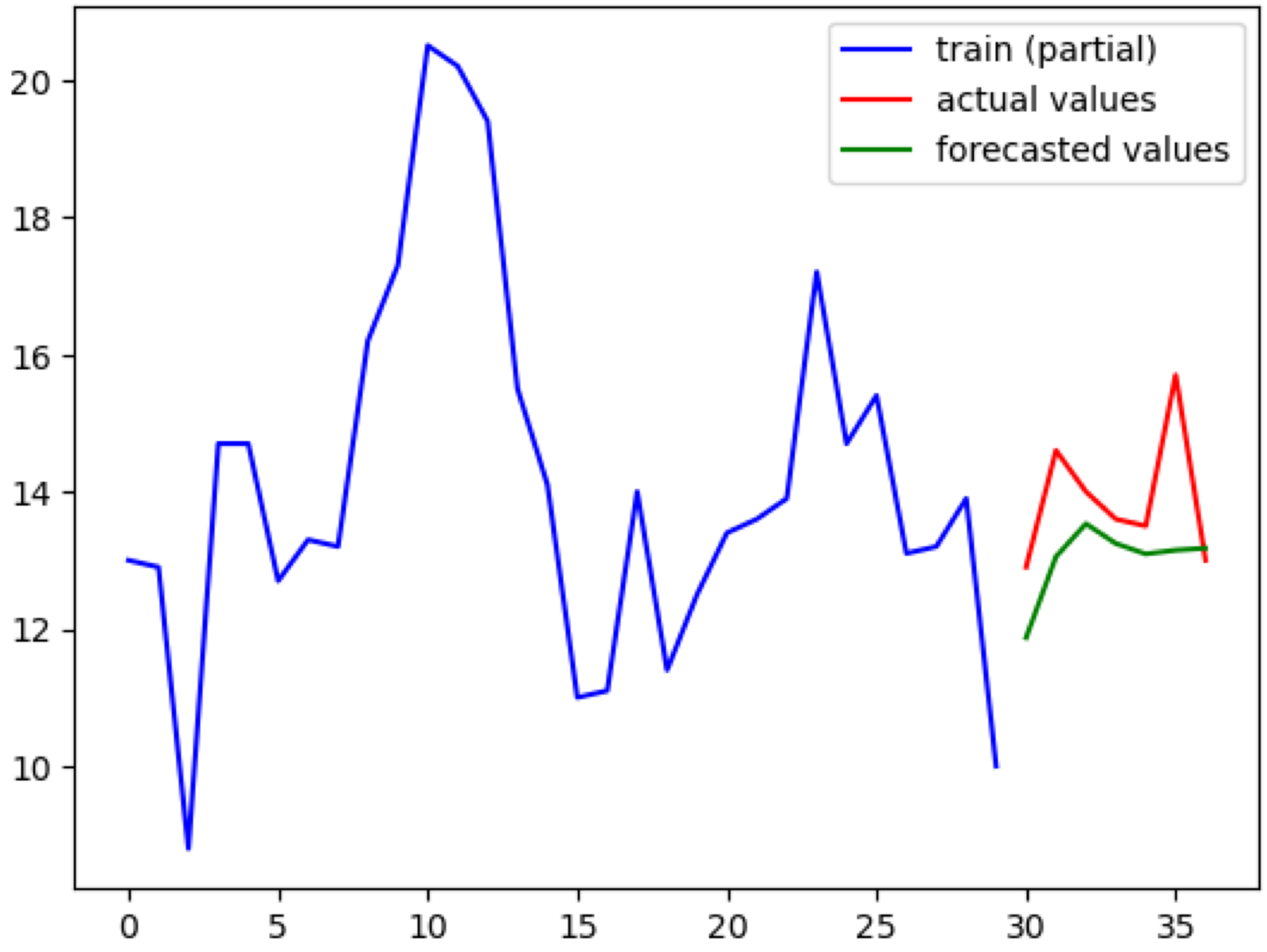
- R Package : forecast
- Method : ar() to fit the model. Or auto.arima()

ARIMA – For Non-Stationary Signals

- AR(p) – Auto Regressive
- I(d) – Integrated
- MA(q) – Moving Average
- ARIMA(p, d, q)
- Use auto.arima() to fit and then forecast() to generate

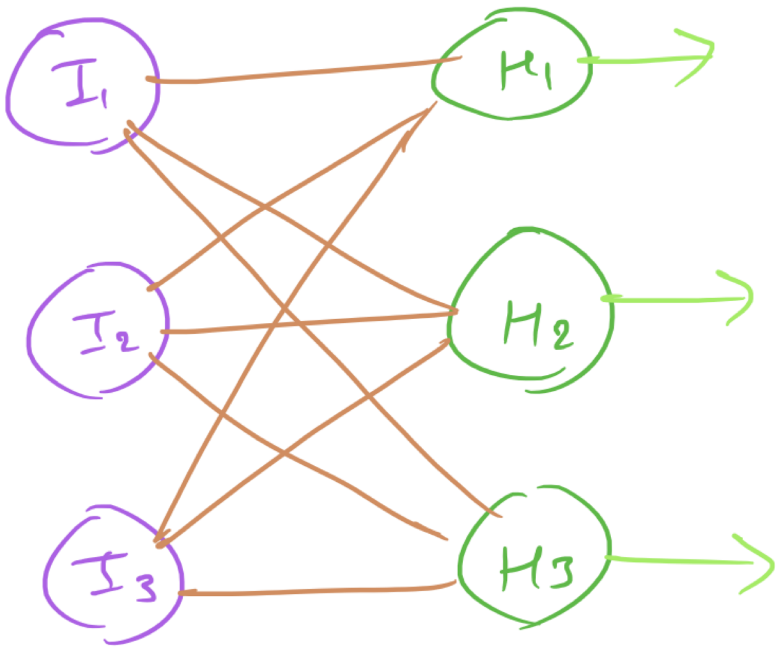
Python Example for AR

- Refer to https://github.com/nextgensh/datalab-teaching-timeseries/blob/main/autoregression.ipynb

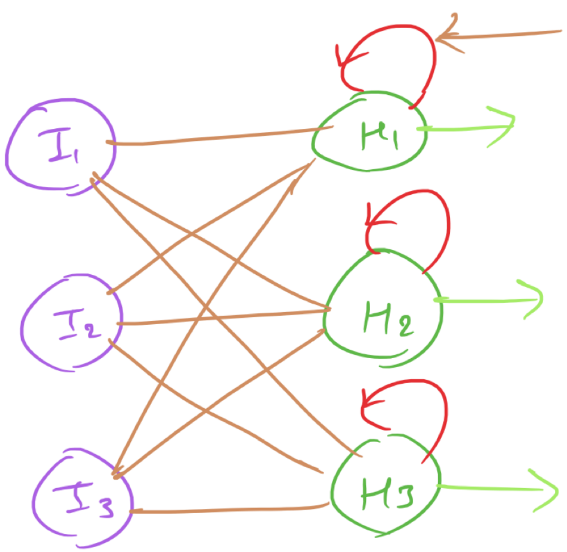


RNN – Recurrent Neural Networks

Feed forward neural networks

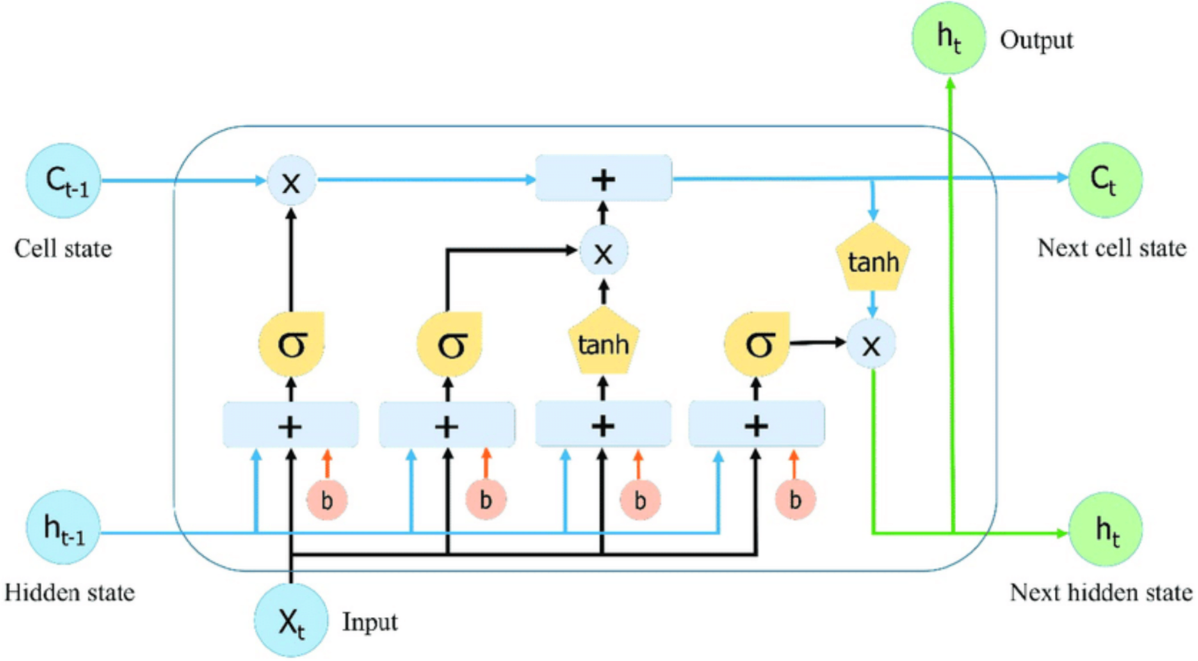


Recurrent neural networks

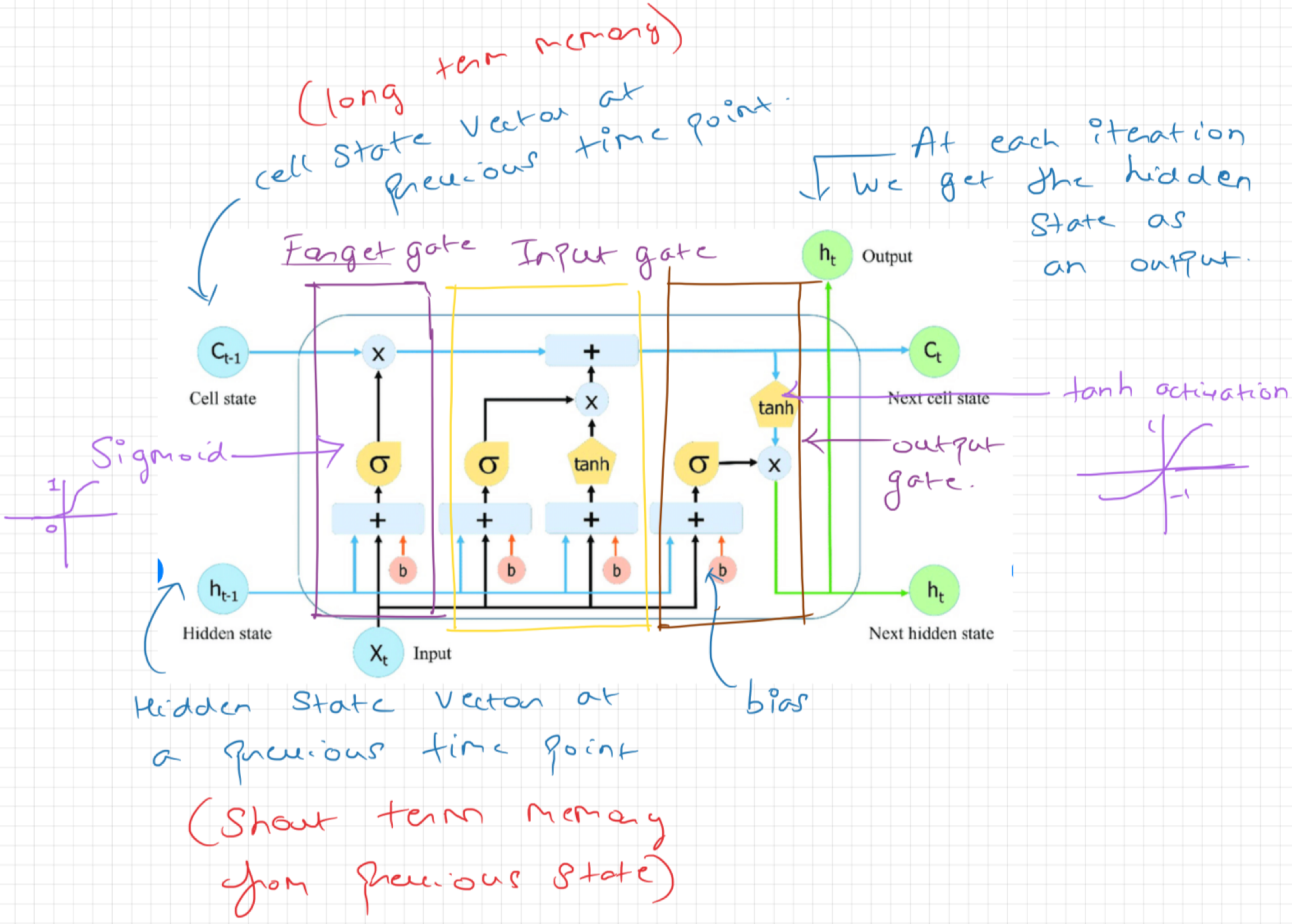


Using information from a previous iteration.

LSTM (Long-Short Term Memory)



LSTM Annotated



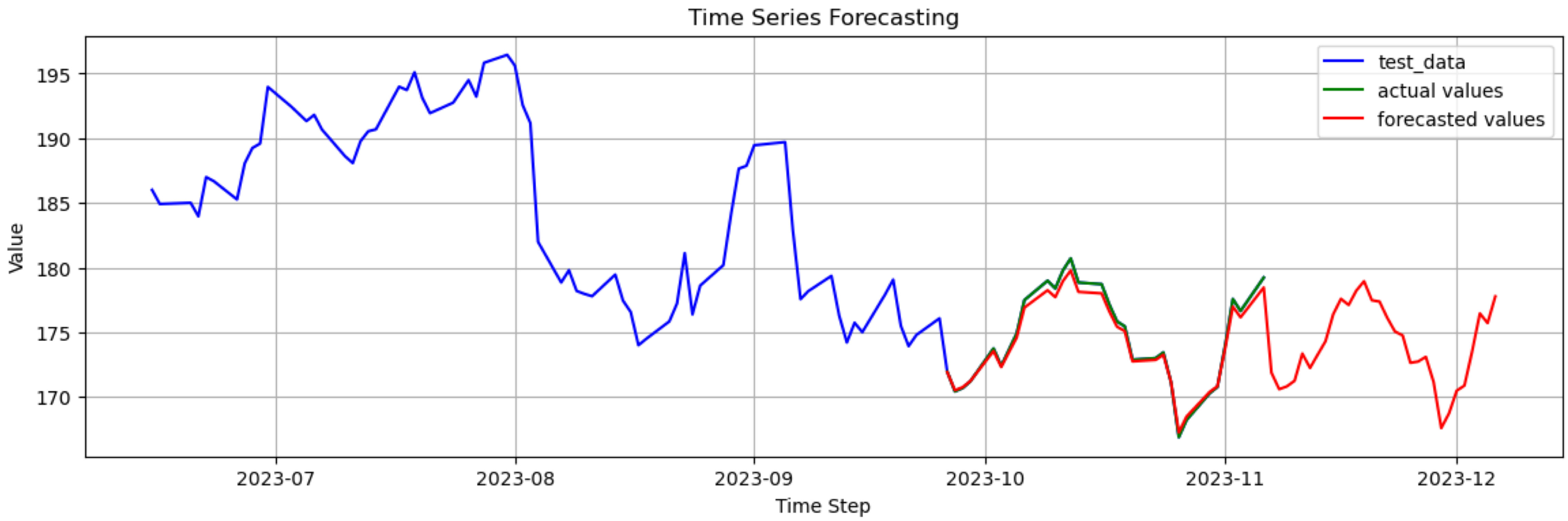
σ : Sigmoids are gate keeper OR they help scale.

gate : $C_{t-1} \cdot \boxed{\sigma(W x_t)}$ (0,1) contribution OR scaling.

internal state

Python Example

- Refer to https://github.com/nextgensh/datalab-teaching-timeseries/blob/main/basicLSTM.ipynb



Thank You

- Check out the workshop page for notebooks and notes.
- Contact me if you have any questions – shravanaras@arizona.edu