Fusemachines
AI School

Introduction to Time Series

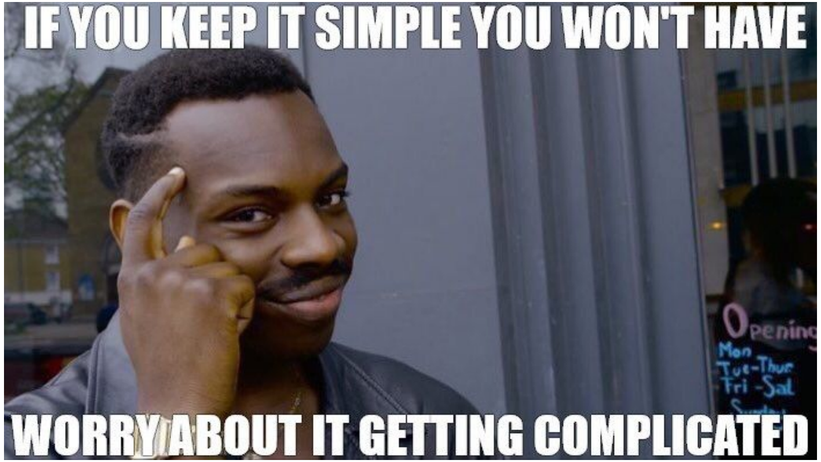
Sijan Shrestha
Sr. ML Engineer, Fusemachines

TODAY’S PLAN

TODAY’S PLAN

TODAY'S PLAN

KEEP IT
SIMPLE
makeameme.org


IF YOU KEEP IT SIMPLE YOU WON'T HAVE
WORRY ABOUT IT GETTING COMPLICATED
Opening
Mon
Tue-Thur
Fri -Sat

BASICS

1 Introduction to Time Series

2 Components of Time Series

3 Time Series Forecasting Models

4 Evaluation of Time Series Models

1 Introduction to Time Series

2 Components of Time Series

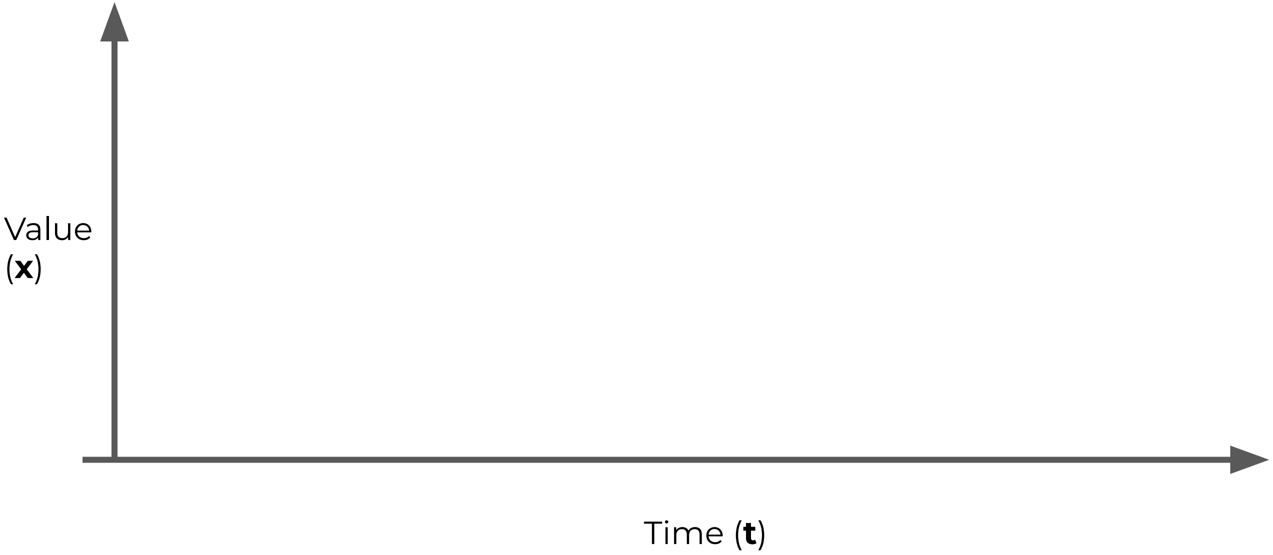
3 Time Series Forecasting Models

4 Evaluation of Time Series Models

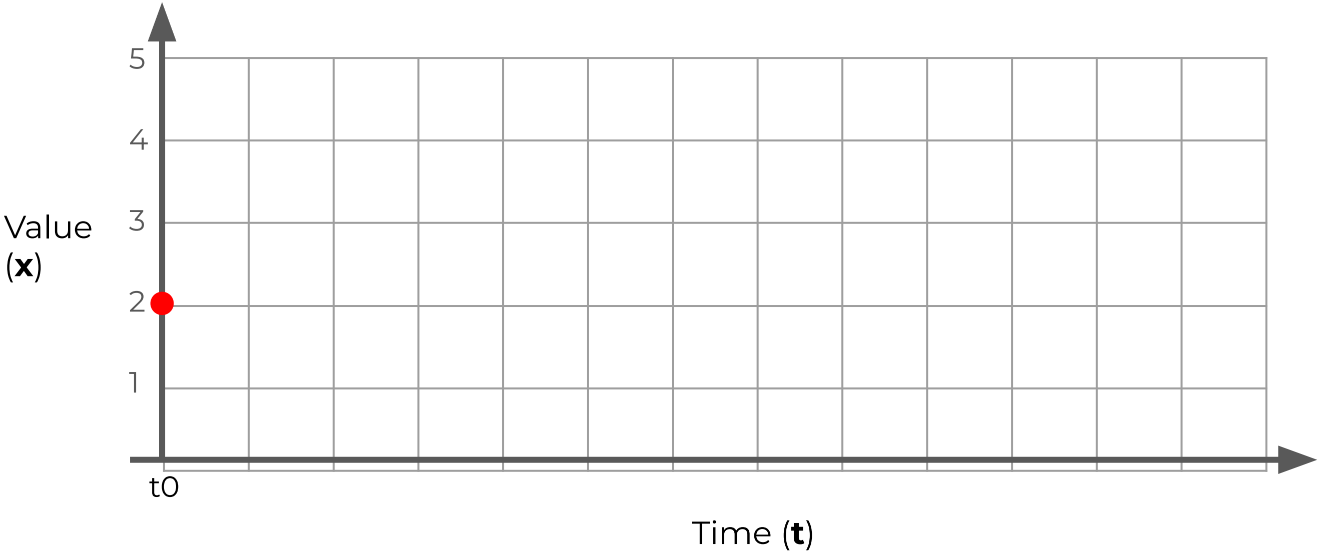
Introduction to Time Series

Time (**t**)

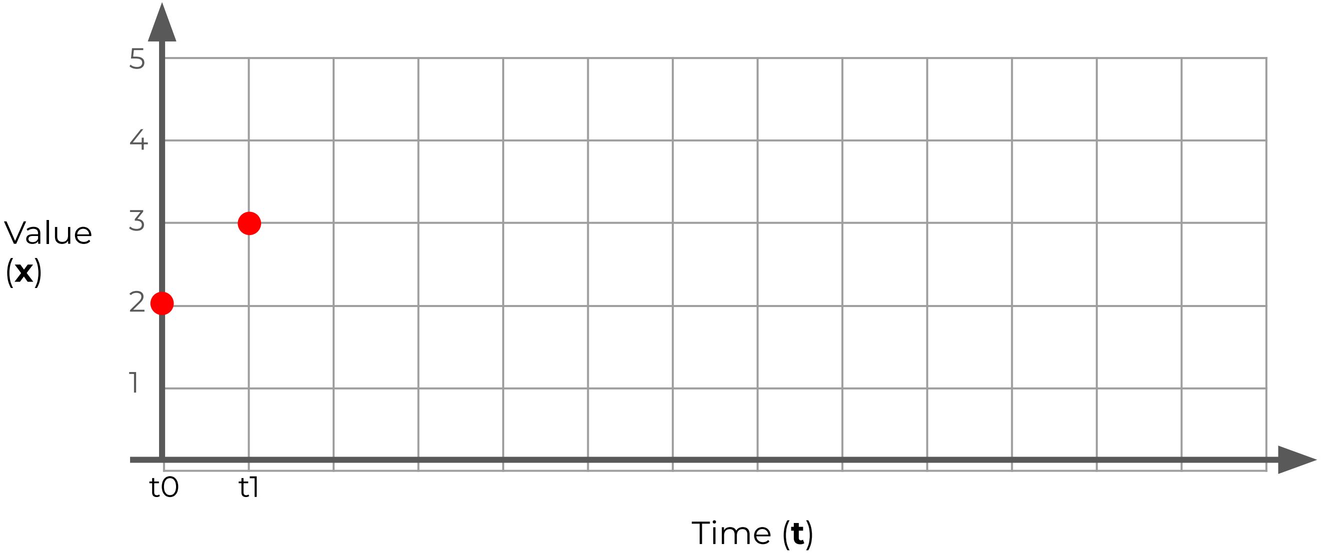
Introduction to Time Series



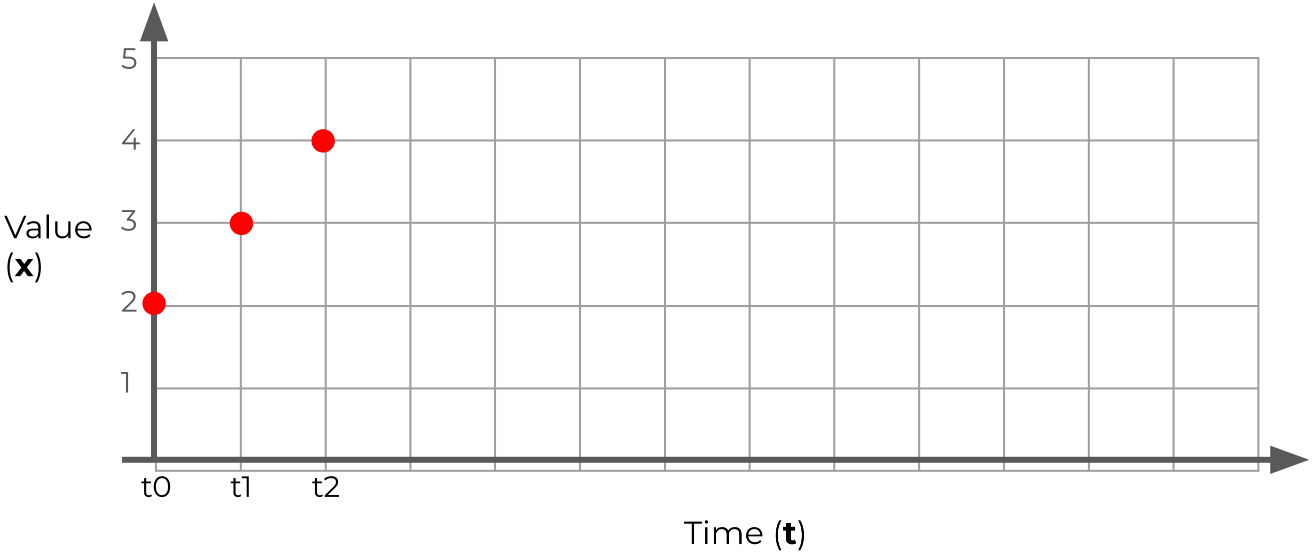
Introduction to Time Series



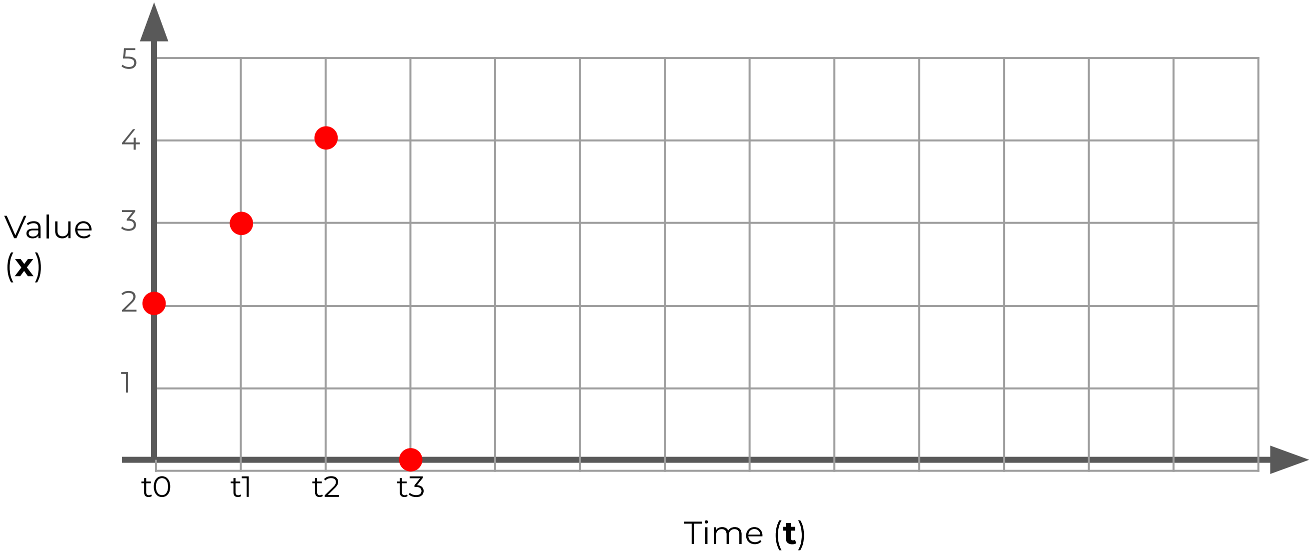
Introduction to Time Series



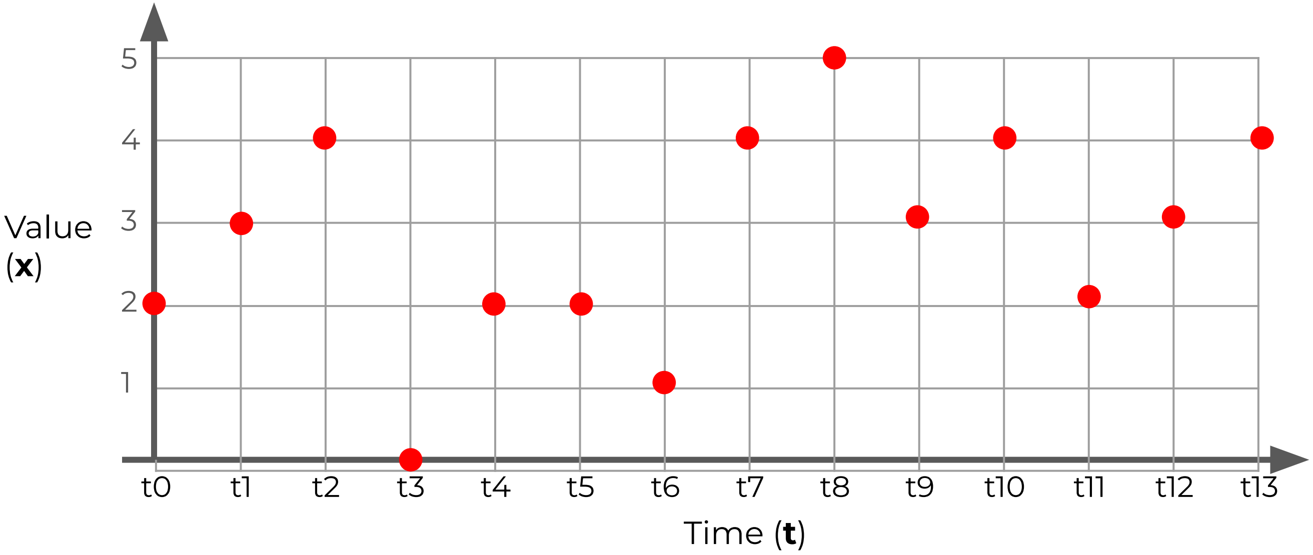
Introduction to Time Series



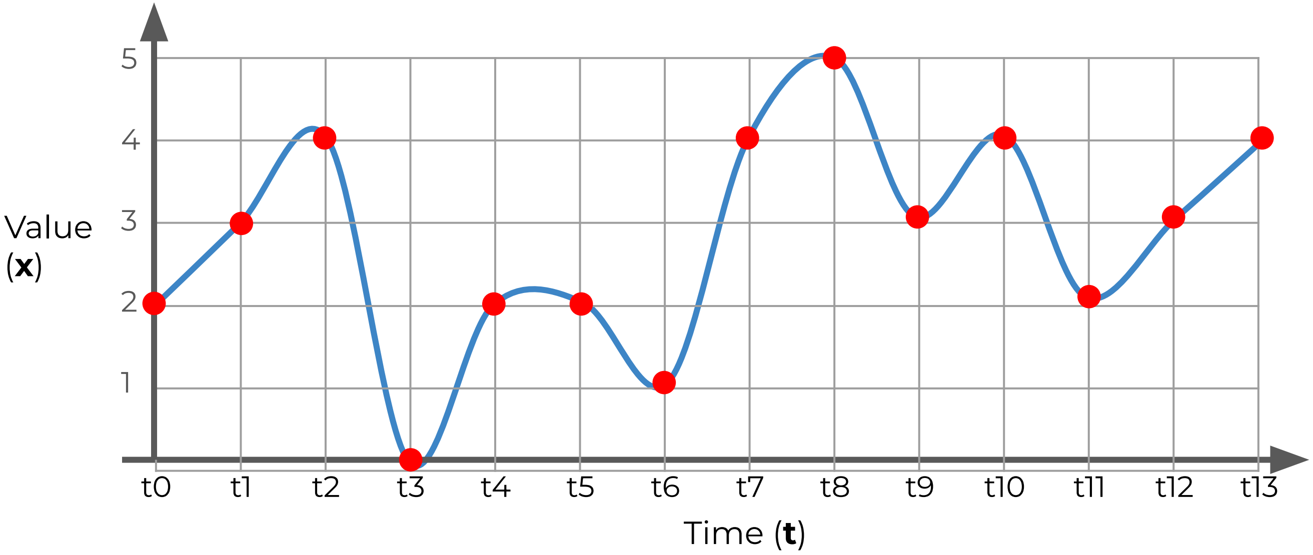
Introduction to Time Series



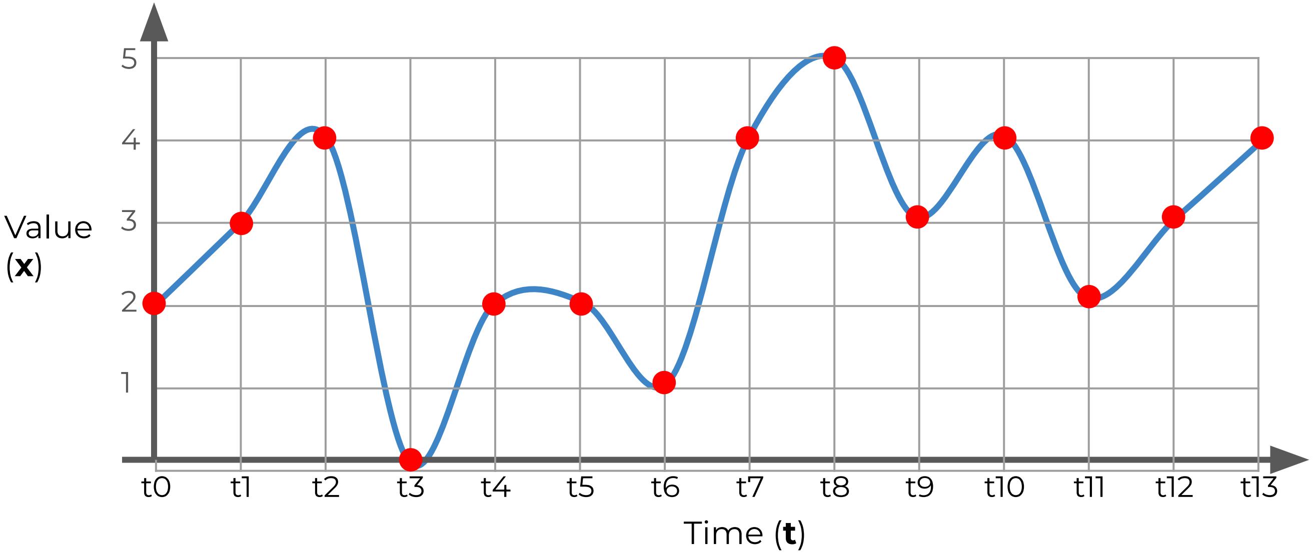
Introduction to Time Series



Introduction to Time Series

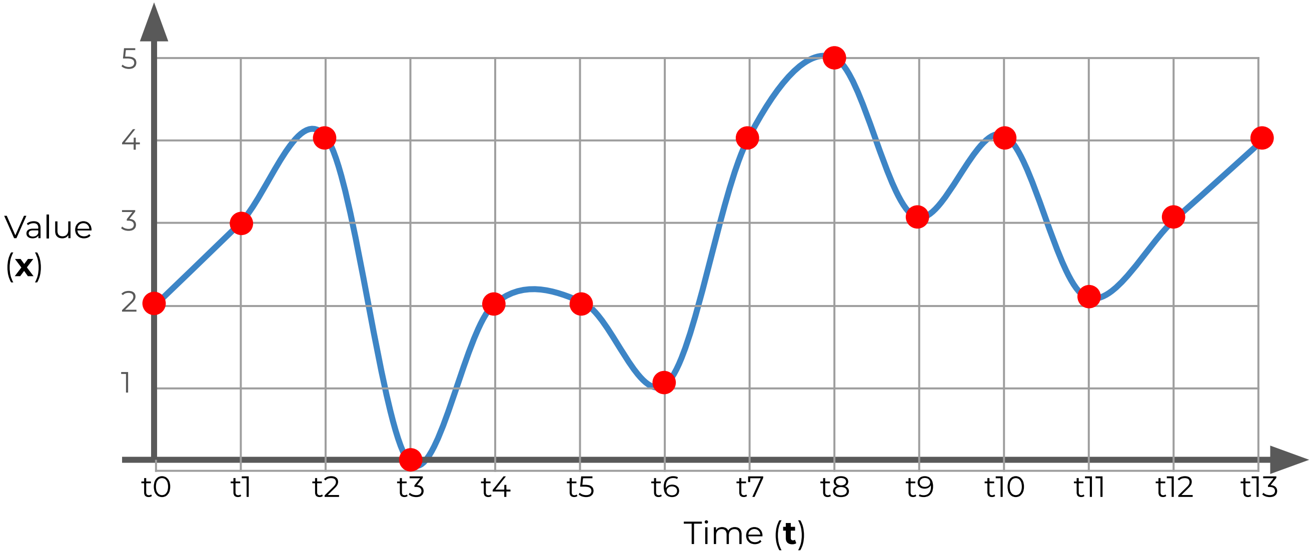


Introduction to Time Series



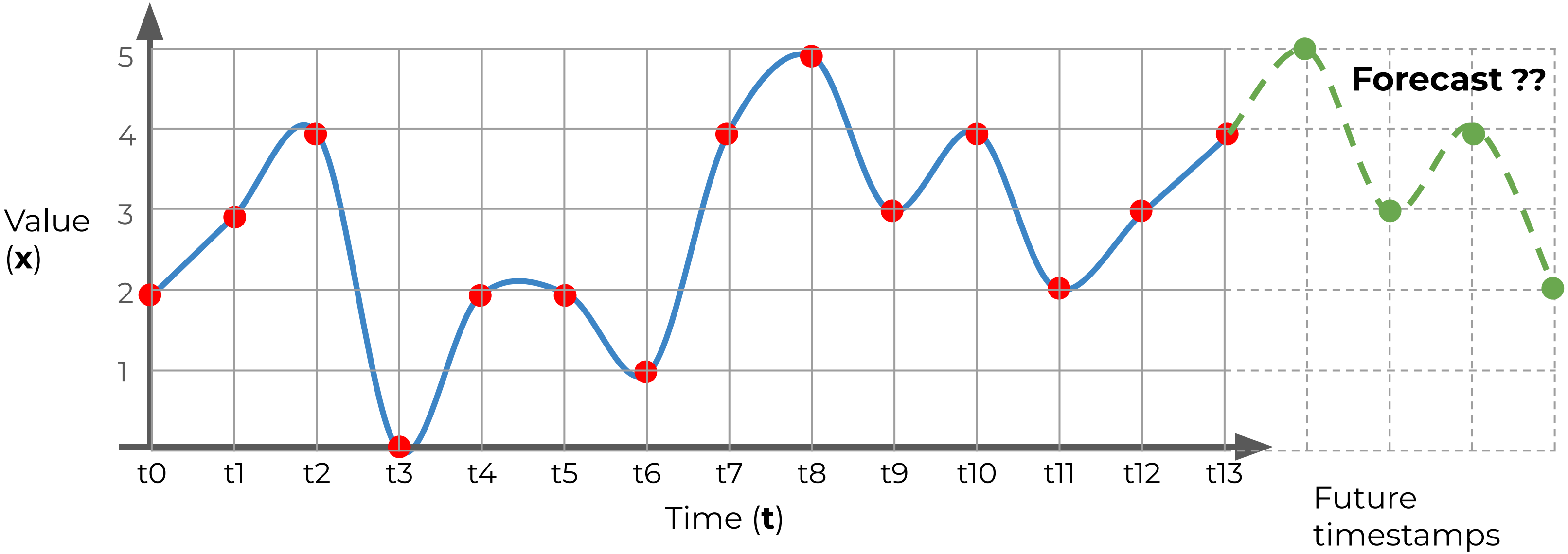
$$\boxed{X = \{x_t\}_{t=0}^{T}}$$

Time Series Forecasting

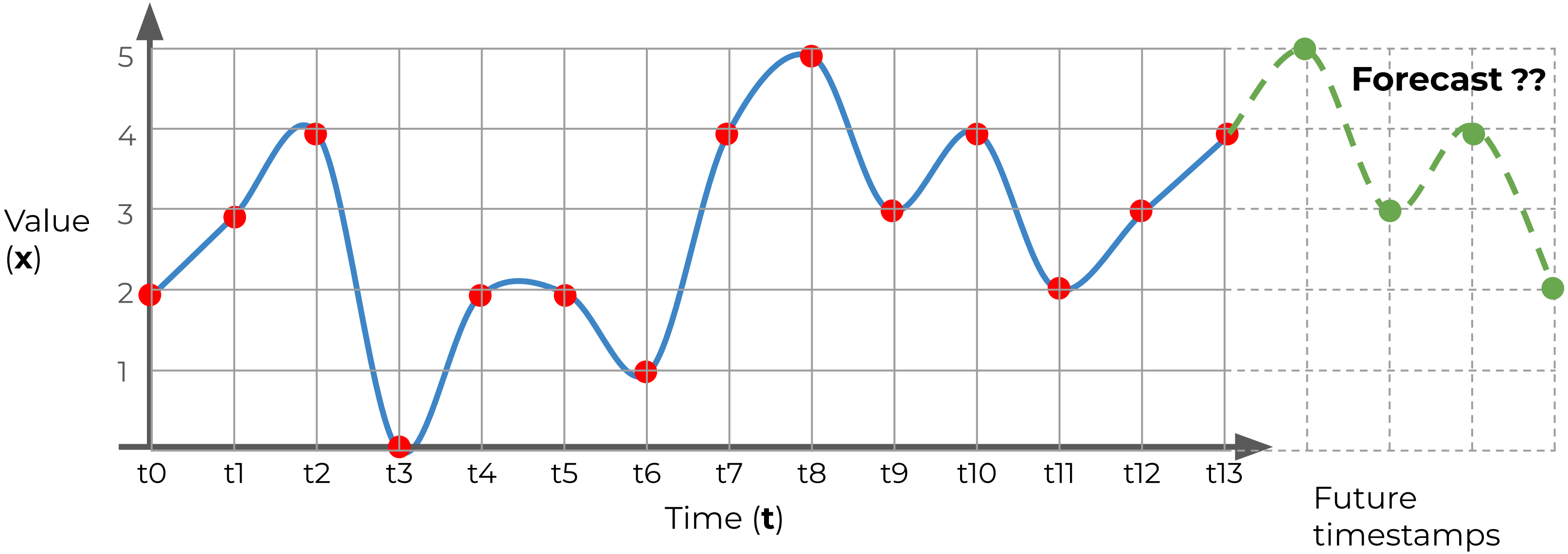


$$\boxed{X = \{x_t\}_{t=0}^{T}}$$

Time Series Forecasting



Time Series Forecasting

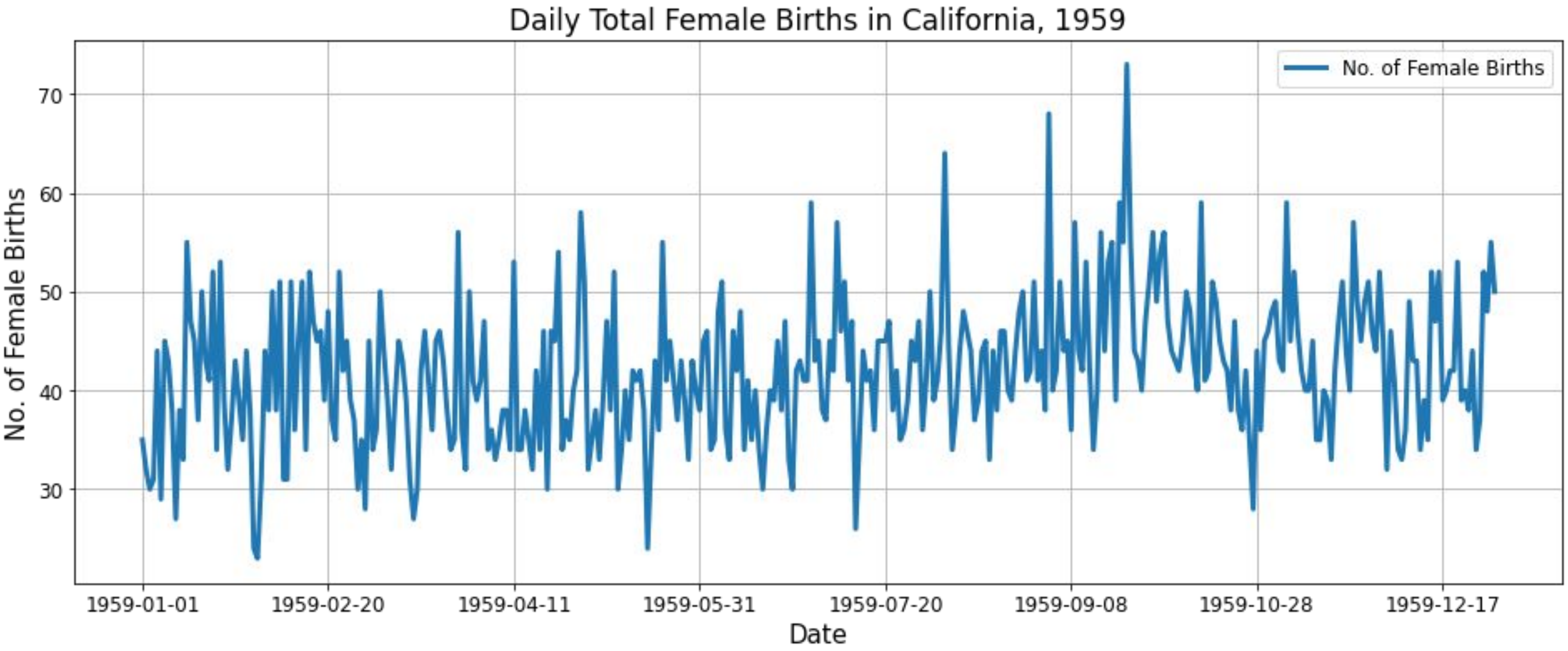


Why time-series forecasting?

- Forecasting future
- Real world applications (**profit**)
    - Finance (predicting stock market price)
    - Economics (GDP, unemployment rates)
    - Retail (sales forecasting)
    - Weather

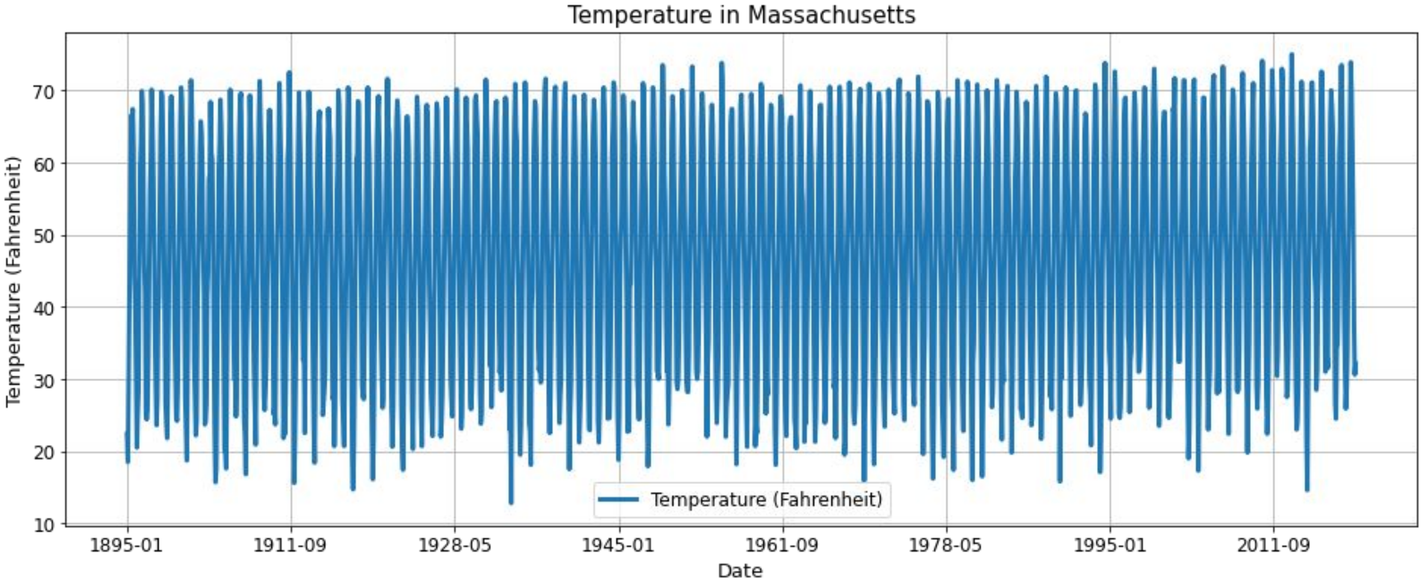
Time Series Examples

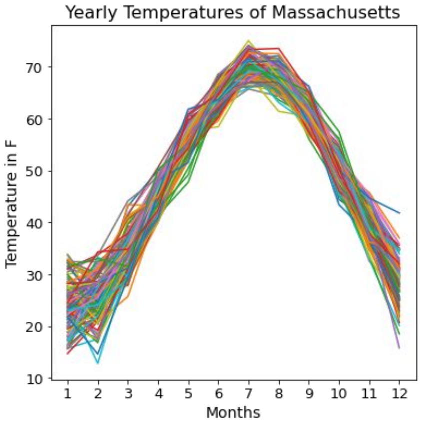
Daily Female Births



Time Series Examples

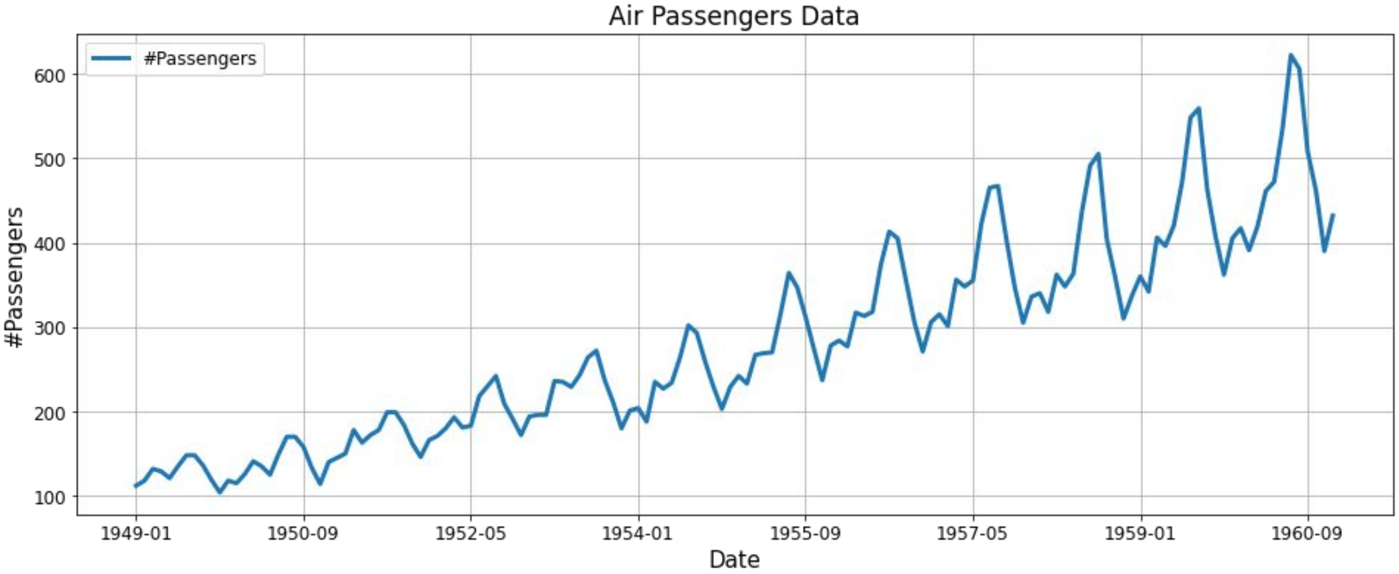
Temperature Data

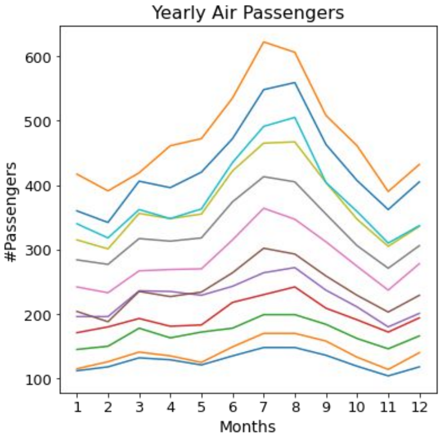


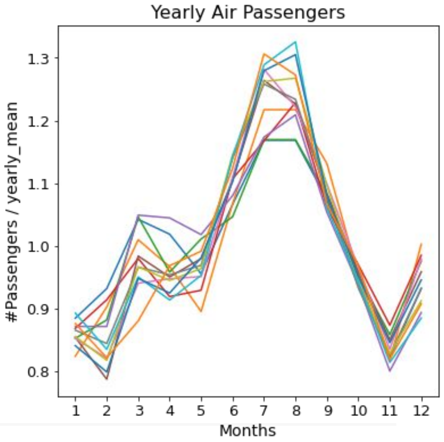


Time Series Examples

Air Passengers

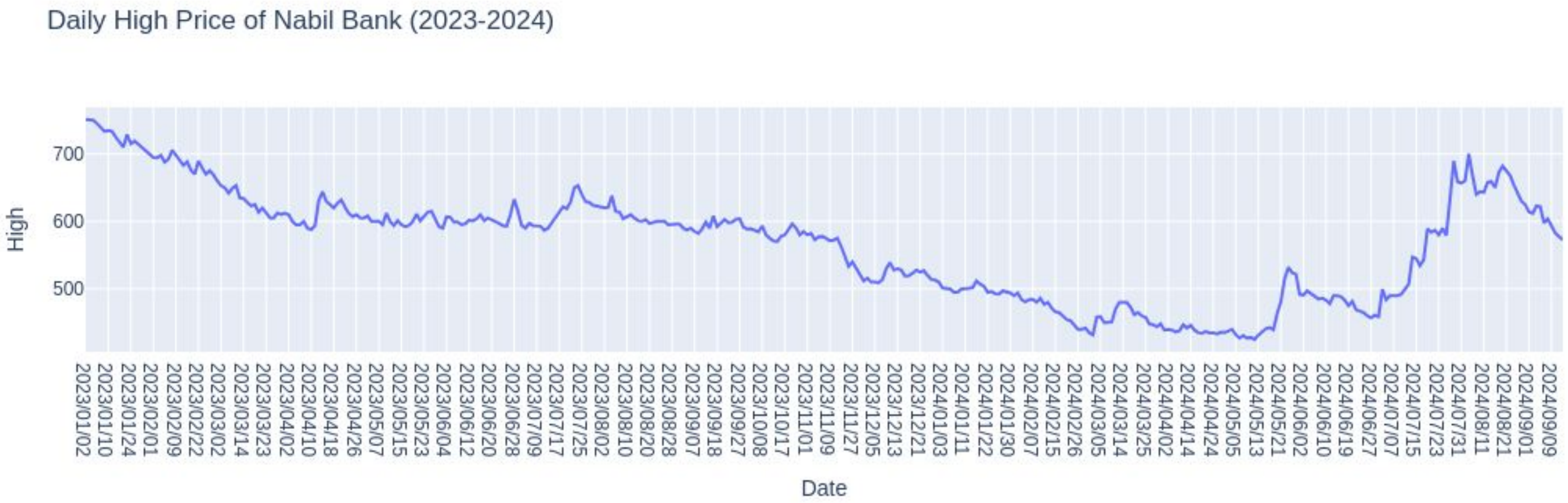






Time Series Examples

Daily closing price of Nabil Bank (NEPSE)



1 Introduction to Time Series

2 Components of Time Series

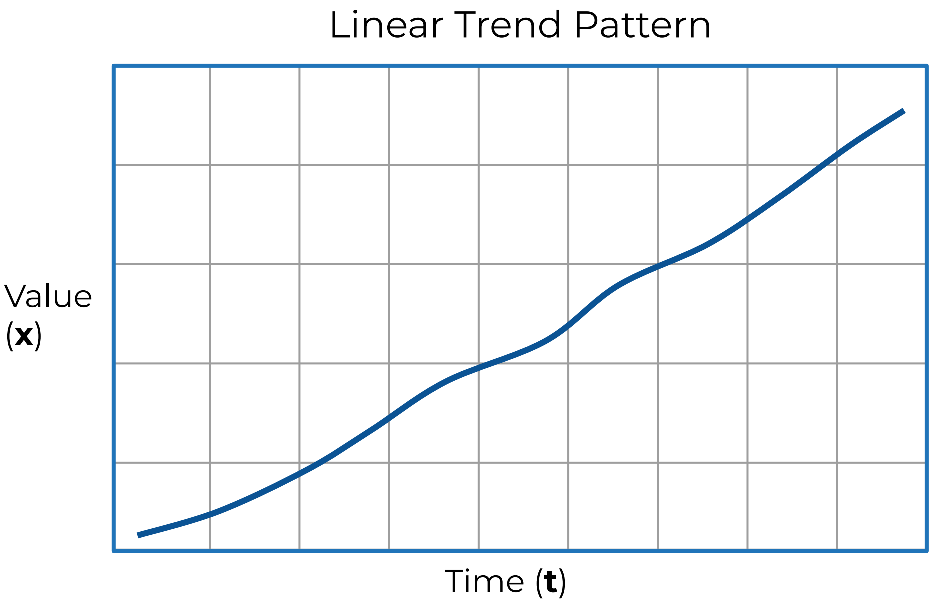
3 Time Series Examples

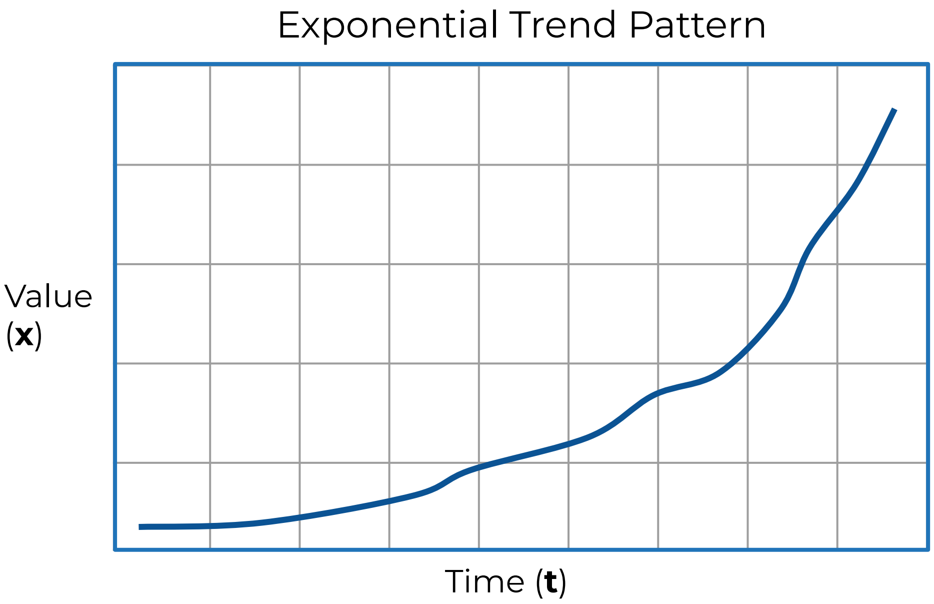
4 Statistical Time Series Model

Components of Time Series

- Trend
- Seasonality
- Error / Noise

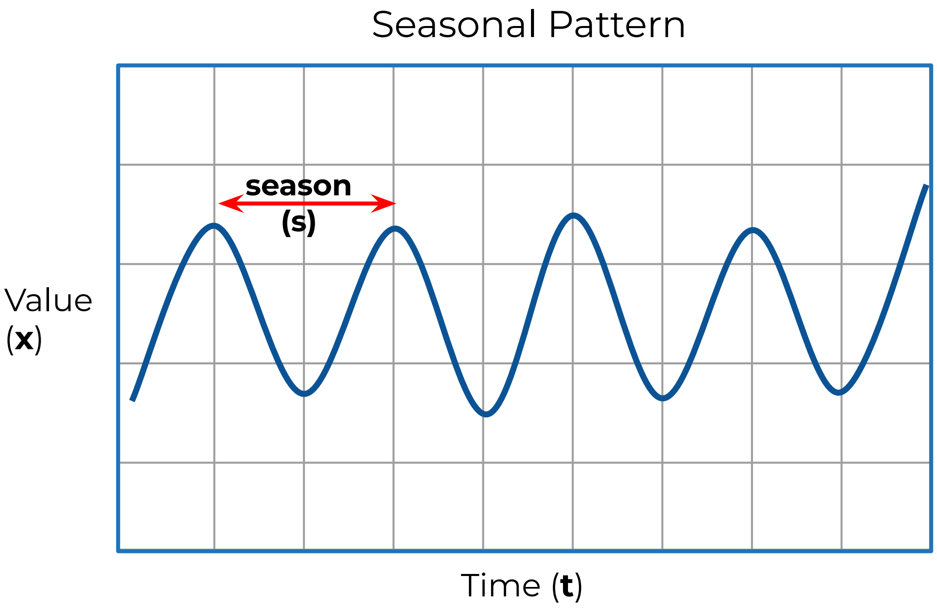
Trend





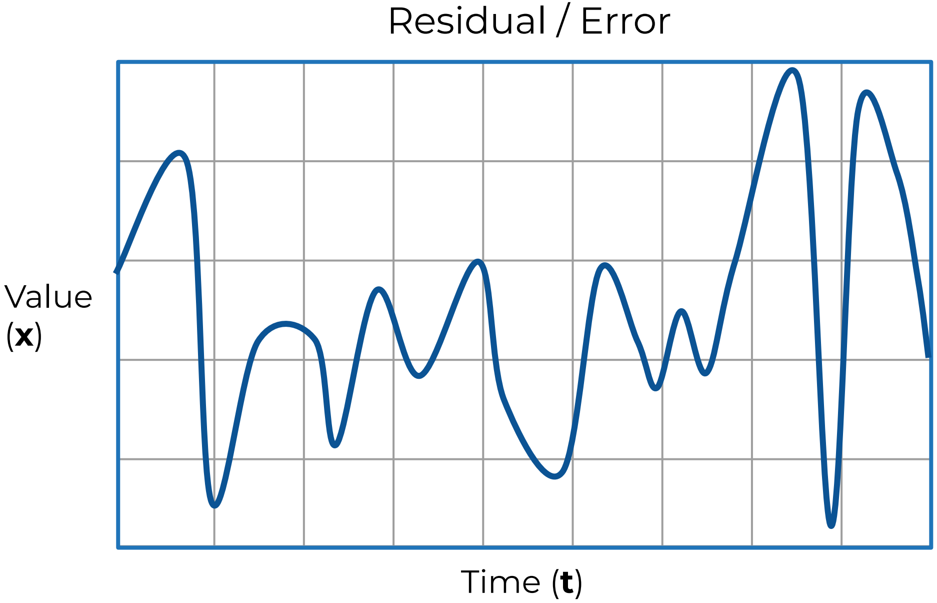
- General upward or downward direction of data over time

Seasonality



- Regular patterns that occur at fixed intervals (weekly, monthly, etc.)

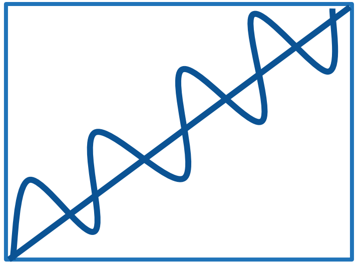
Error / Noise



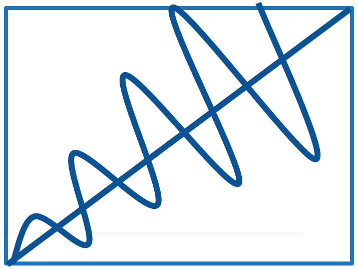
- Random fluctuations in data

Time Series Decomposition

- Breaking down of time series into individual components.
- Types of Decomposition
  - Additive Decomposition
    - y(t) = Trend(T) + Seasonality(S) + Error (E)
    - Used when seasonal variation is constant over time



  - Multiplicative Decomposition
    - y(t) = Trend(T) * Seasonality(S) * Error (E)
    - Used when seasonal variation grows or shrinks proportionally with trend.

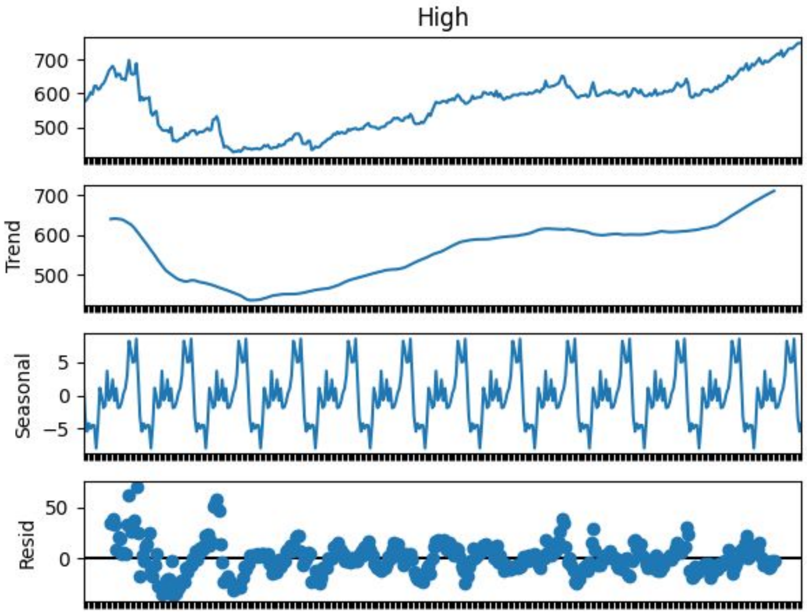


Time Series Decomposition

- Additive Decomposition of **Daily High of Nabil Bank**

```
from statsmodels.tsa.seasonal import seasonal_decompose

# Additive Decomposition
additive_decomposition = seasonal_decompose(df['High'], model='additive', period=30)
additive_decomposition.plot()
plt.show()
```



1 Introduction to Time Series

2 Components of Time Series

3 **Time Series Forecasting Models**

4 Evaluation of Time Series Models

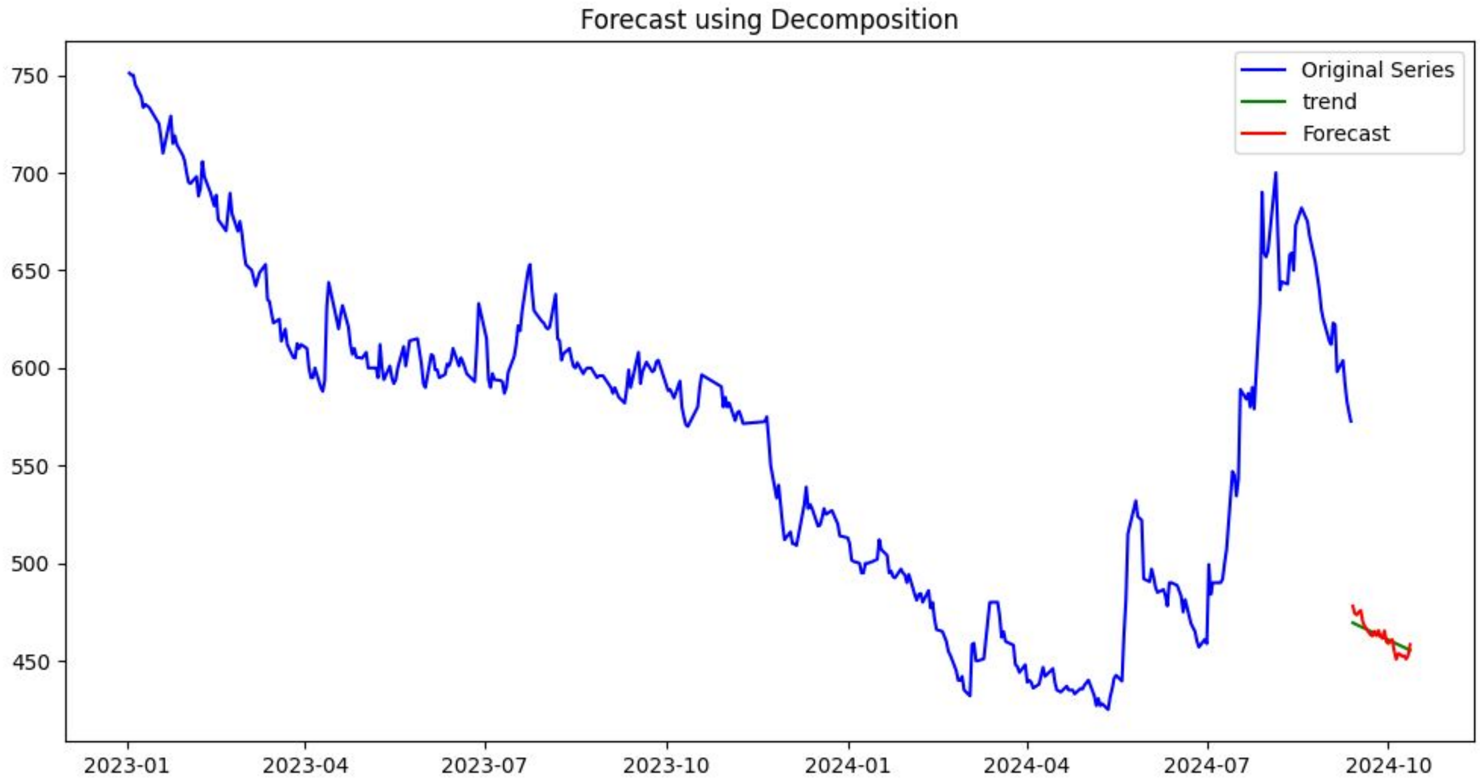
Time Series Forecasting Models

- Simple Forecasting Models
- Exponential Smoothing Models
- Decomposition Based Models
- Autoregressive Models
- Different versions of ARIMA
- Machine Learning Models
- Hybrid Models
- LLM Based Foundation Models

Decomposition Based Models

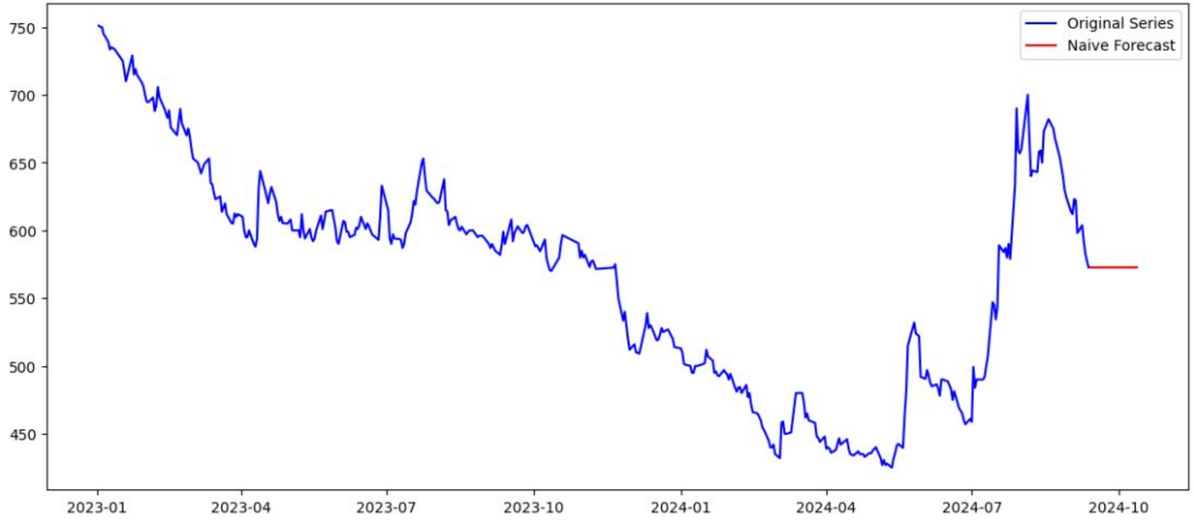
- Decompose the time-series
- Forecast trend using regression or smoothing techniques
- Use the seasonal component identified during decomposition (repeat seasonal pattern of last n periods into the future)
- Assume residuals are random and assume their forecast is 0 (or small random error)
- Recombine the components

Decomposition Based Models



Simple Forecasting Models

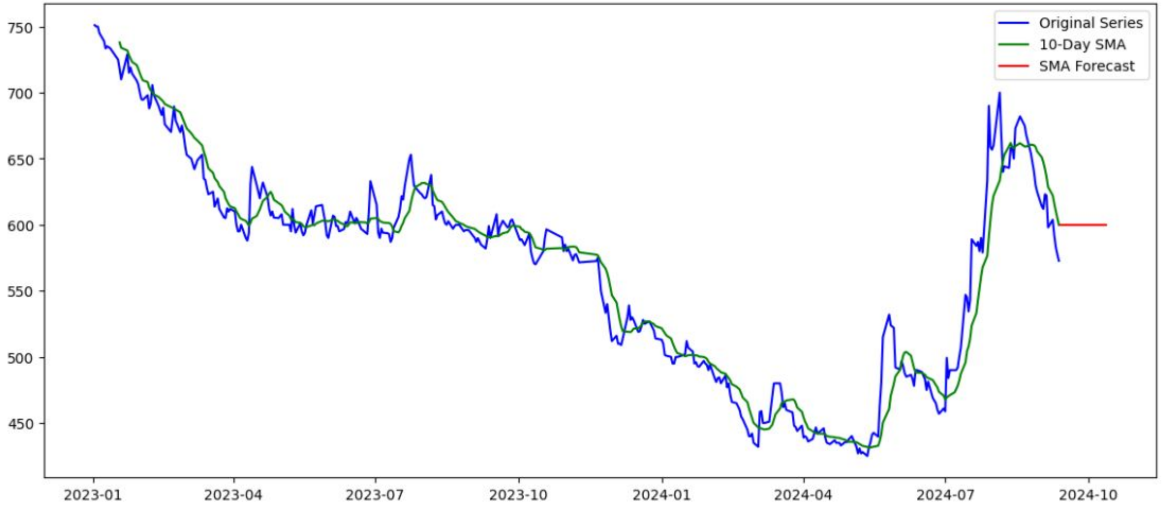
- Naive Forecast
  - Uses the most recent observation as forecast for all future periods



Simple Forecasting Models

- Simple Moving Average
  - Forecast is average of last n observations

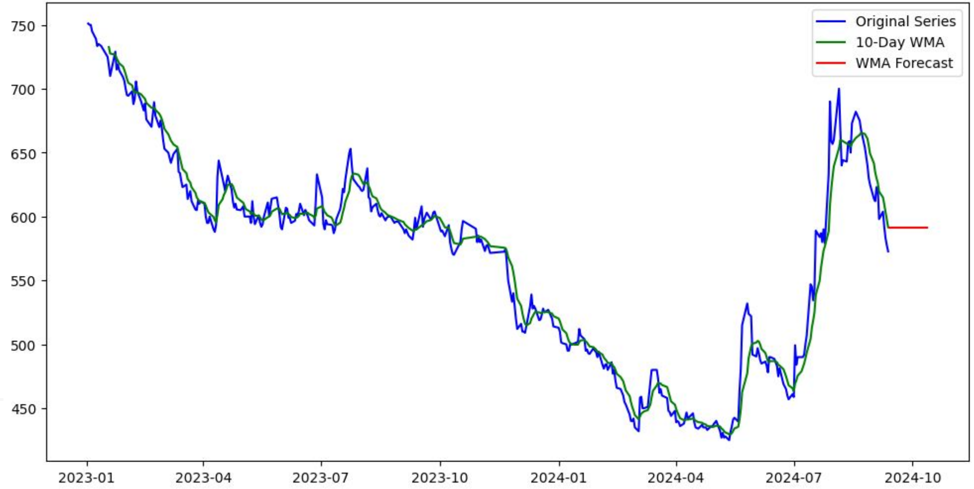
$$\hat{y}_{t+1} = \frac{y_t + y_{t-1} + \cdots + y_{t-n+1}}{n}$$



Simple Forecasting Models

- Weighted Moving Average
  - Forecast is average of last n observations, but weights recent data more heavily

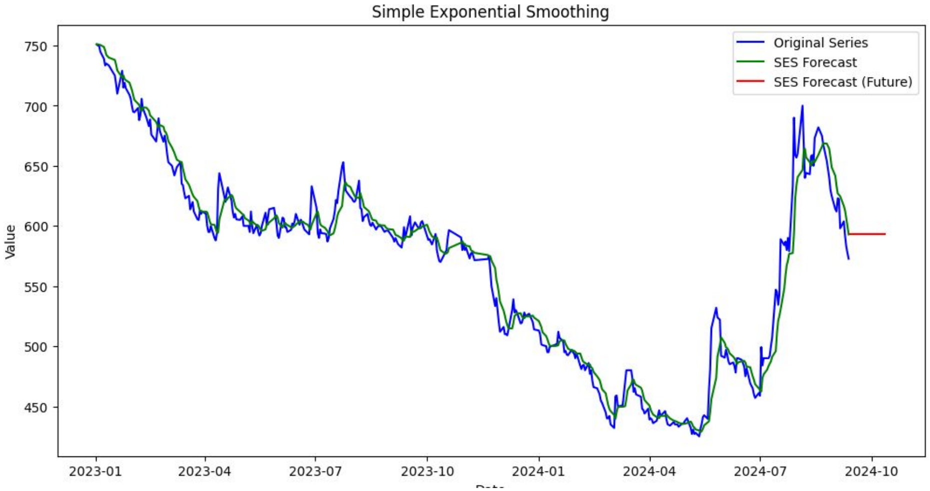
$$\hat{y}_{t+1} = w_1 y_t + w_2 y_{t-1} + \cdots + w_n y_{t-n+1}, \quad \text{where} \quad w_1 > w_2 > \cdots > w_n$$



Exponential Smoothing Models

- Simple Exponential Smoothing
  - Weighted moving average with exponentially decreasing weights (determined by smoothing parameter **α**)

$$\hat{y}_{t+k} = \alpha y_t + \alpha(1-\alpha) y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots + \alpha(1-\alpha)^{n-1} y_{t-n+1} + (1-\alpha)^n \hat{y}_{t-n+1}$$



Exponential Smoothing Models

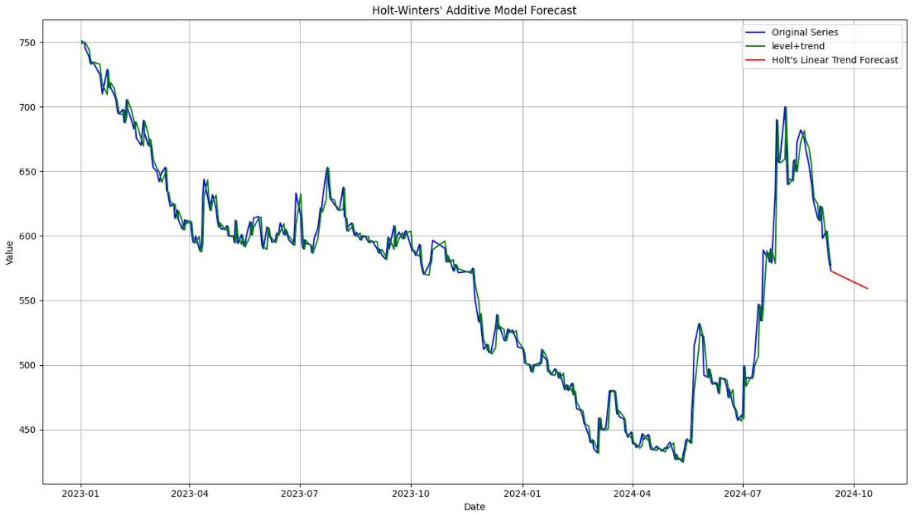
- Holt’s Linear Trend Model
  - Extension of SES that accounts for linear trends in data

$$\hat{y}_{t+1} = \ell_t + b_t$$

where:

$$\text{level}(\ell_t) = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$$

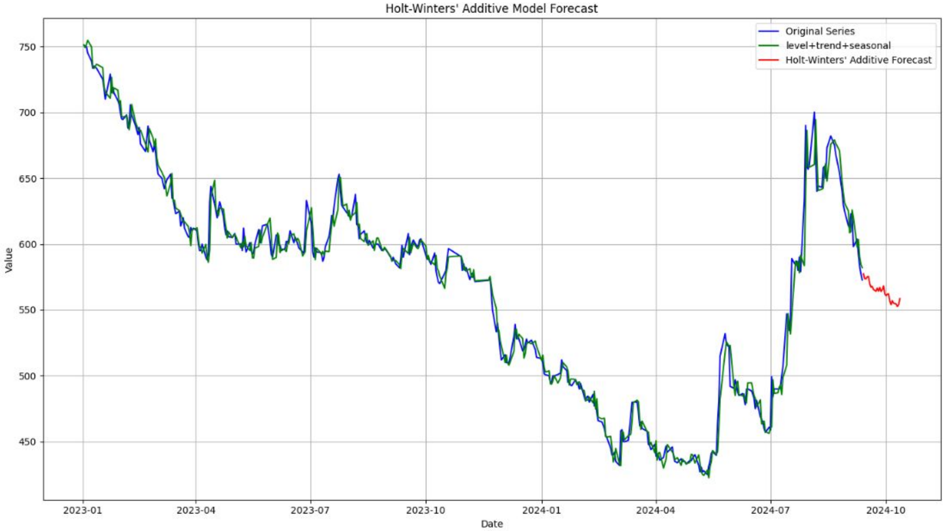
$$\text{trend}(b_t) = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$$



Exponential Smoothing Models

- Holt-Winters Seasonal Model (ETS)
    - Extension of Holt’s Linear Trend model that includes seasonal effects (additive and multiplicative

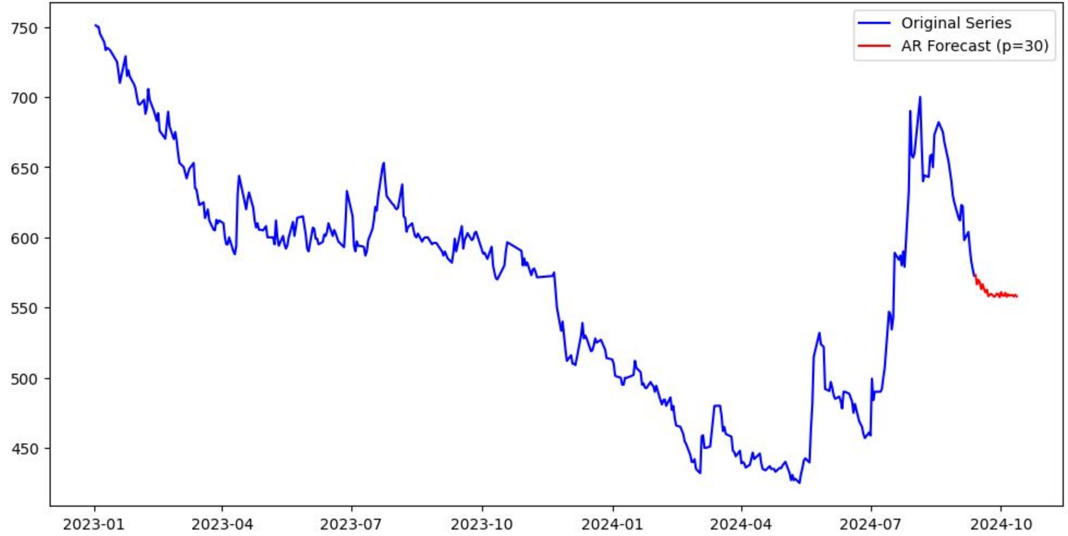
$$
\begin{aligned}
\hat{y}_{t+1} &= \ell_t + b_t + s_{t+1} \\
\text{where:} & \\
\text{level}(\ell_t) &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{trend}(b_t) &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
\text{seasonal component}(s_t) &= \gamma(y_t - l_t) + (1-\gamma)s_{t-m}
\end{aligned}
$$



Auto-regressive models

- AR models
  - Time series value is regressed on it’s past values

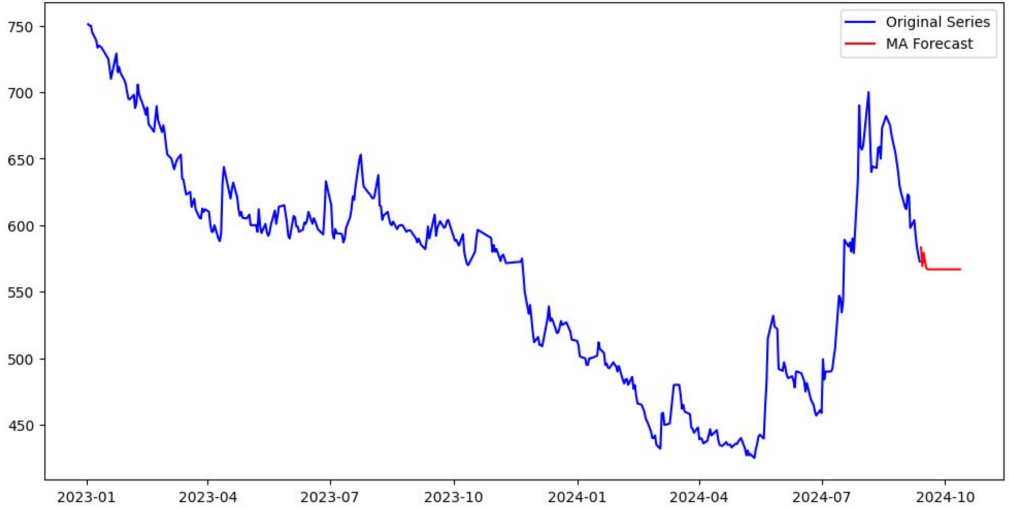
$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$



Auto-regressive models

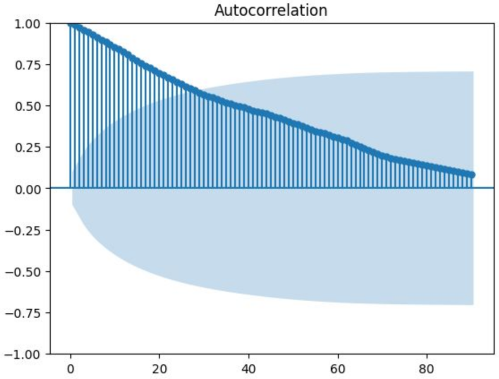
- MA models
    - Uses past forecast errors to predict future values

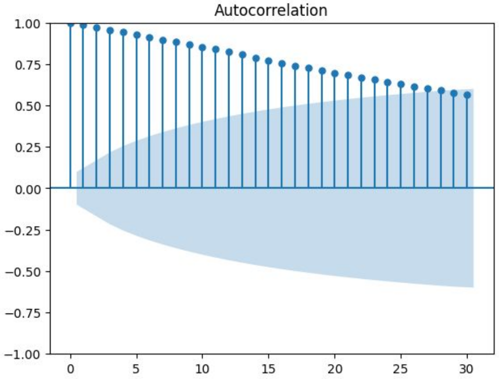
$$y_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$



Auto-regressive models

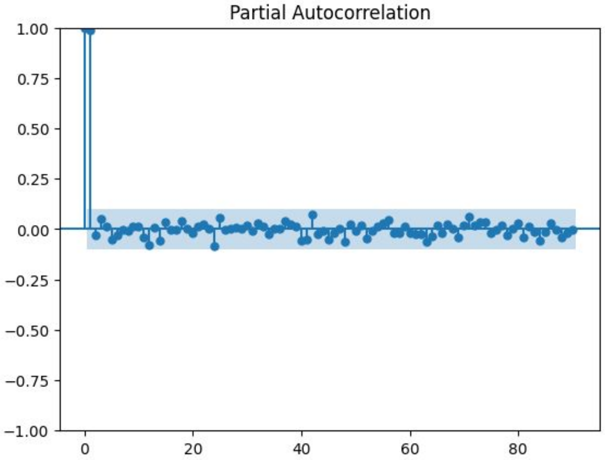
- How do we know what lag to use?
    - Autocorrelation Function (ACF)
        - Correlation of time series with itself at different lags
        - Used to determine value of q in MA models





Auto-regressive models

- Partial Autocorrelation Function (ACF)
  - Correlation of time series with it’s past values after removing influence of earlier lags
  - Used to determine value of p in AR models

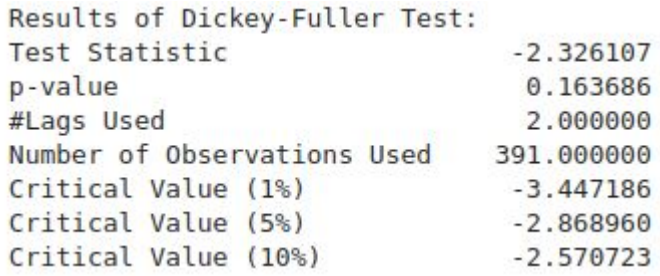


Auto-regressive models

- Stationarity
    - A series is stationary when it’s statistical properties like mean, variance and autocorrelation remains constant
    - Why stationarity?
        - A non-stationary series is considered difficult to forecast
    - Stationary test
        - Augmented Dickey-Fuller (ADF) Test
        - Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

Auto-regressive models

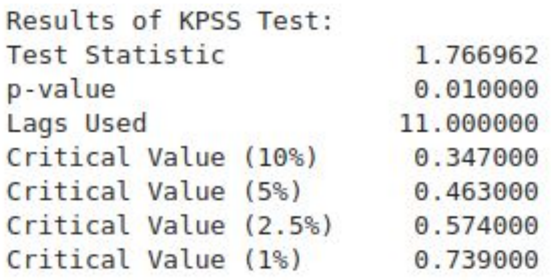
- Stationarity test
    - Augmented Dickey-Fuller (ADF) Test
        - **Null hypothesis**: Series is not stationary
        - ADF test gives you a test statistic, a p-value, and critical values at different significance levels (like 1% or 5%).
        - p-value < significance level (0.05) means we reject null hypothesis (series is stationary)
        - Test statistic < critical value (5%) means we reject null hypothesis (series is stationary)



```
Results of Dickey-Fuller Test:
Test Statistic                 -2.326107
p-value                         0.163686
#Lags Used                      2.000000
Number of Observations Used   391.000000
Critical Value (1%)            -3.447186
Critical Value (5%)            -2.868960
Critical Value (10%)           -2.570723
```

Auto-regressive models

- Stationarity test
  - (KPSS) Test
    - **Null hypothesis**: Series is trend stationary
    - ADF test gives you a test statistic, a p-value, and critical values at different significance levels (like 1% or 5%).
    - p-value < significance level (0.05) means reject null hypothesis (series is non-stationary)
    - Test statistic > critical value (5%) means reject null hypothesis (series is non-stationary)



```
Results of KPSS Test:
Test Statistic            1.766962
p-value                   0.010000
Lags Used                11.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
```

Auto-regressive models

- How to make series stationary?
  - Differencing ($y_t = y_t - y_{t-1}$)
  - Seasonal differencing ($y_t = y_t - y_{t-m}$)
  - Log transformation ($y_t = \log(y_t)$)
  - Decomposition

Auto-regressive models

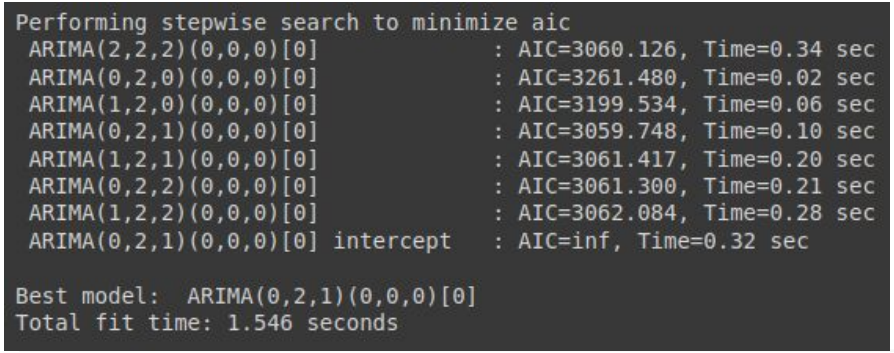
- ARIMA model
  - AutoRegressive Integrated Moving Average
    - AR (**p**)
    - I (differencing) (**d**=degree of differencing))
    - MA (**q**)

Auto-regressive models

- How do we find p, d, q?
- By minimizing AIC or BIC
  - Akaike Information Criteria (AIC)
    - AIC = 2k - 2l,
    - k is number of parameters and l is log likelihood
  - Bayesian Information Criteria (BIC)
    - BIC = klogn - 2l
    - n is number of samples used
- Library **pmdarima** has a function **auto_arima** to optimize values of p, d, q

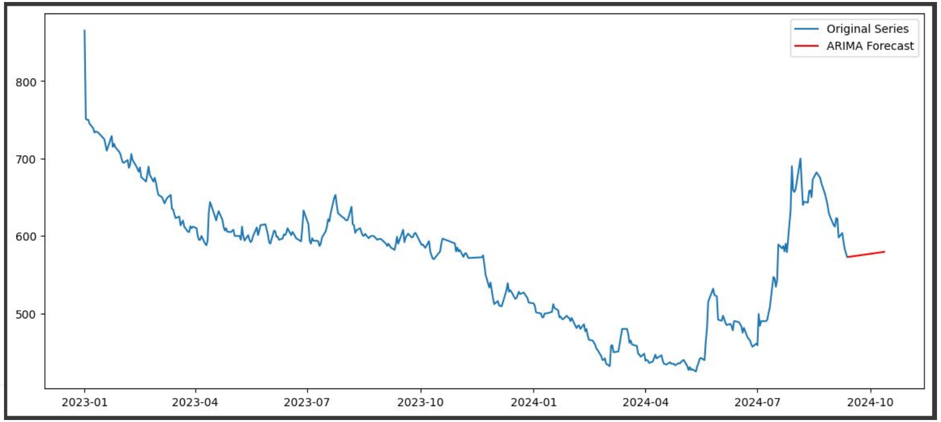
Different variants of ARIMA

- SARIMA model
  - Added seasonal component to ARIMA
  - Four more parameters **(P, D, Q, m):** Seasonal parameters
  - Optimize p, d, q, P, D, Q, m using **auto_arima**



```
Performing stepwise search to minimize aic
 ARIMA(2,2,2)(0,0,0)[0]             : AIC=3060.126, Time=0.34 sec
 ARIMA(0,2,0)(0,0,0)[0]             : AIC=3261.480, Time=0.02 sec
 ARIMA(1,2,0)(0,0,0)[0]             : AIC=3199.534, Time=0.06 sec
 ARIMA(0,2,1)(0,0,0)[0]             : AIC=3059.748, Time=0.10 sec
 ARIMA(1,2,1)(0,0,0)[0]             : AIC=3061.417, Time=0.20 sec
 ARIMA(0,2,2)(0,0,0)[0]             : AIC=3061.300, Time=0.21 sec
 ARIMA(1,2,2)(0,0,0)[0]             : AIC=3062.084, Time=0.28 sec
 ARIMA(0,2,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.32 sec

Best model:  ARIMA(0,2,1)(0,0,0)[0]
Total fit time: 1.546 seconds
```



Different variants of ARIMA

- SARIMAX model
  - SARIMA model with Exogenous variables (factors external to time series but believed to influence it)
  - The model is represented as SARIMA(p, d, q)(P, D, Q, m) + X
  - X -> one or more variables
  - Exogenous variable can be holiday, time features, lag features, other factors that can affect the target.

Different variants of ARIMA

- VARIMA (VECTOR ARIMA)
  - Multiple time-series that are related
  - One model to forecast multiple time-series
  - Example: Model the relationship between **GDP growth, inflation rates**, and **interest rates** to forecast future trends

Univariate vs Multivariate Time Series

- **Univariate**: Single variable which is our target.
- **Multivariate**: Multiple variables that are interdependent, target can be one or more.

Univariate vs Multivariate Time Series

- Variates that can be used as features
  - **Time features**: day_of_week, month_of_year, etc.
    - Better to use sine, cosine transformation because of periodic nature
  - **Lag features**: Exact or aggregated lag over time
  - **Holiday or calendar events:** Can be used as categorical feature or distance to the event can be numerical feature
  - **Other related features** to the use-case

Machine Learning Models

- Regression and Random Forests
    - **Regression Trees:** Use decision trees to model relationships between time series features and future values.
    - **Random Forests:** Ensemble of decision trees that improves accuracy and reduces overfitting by averaging predictions.

Machine Learning Models

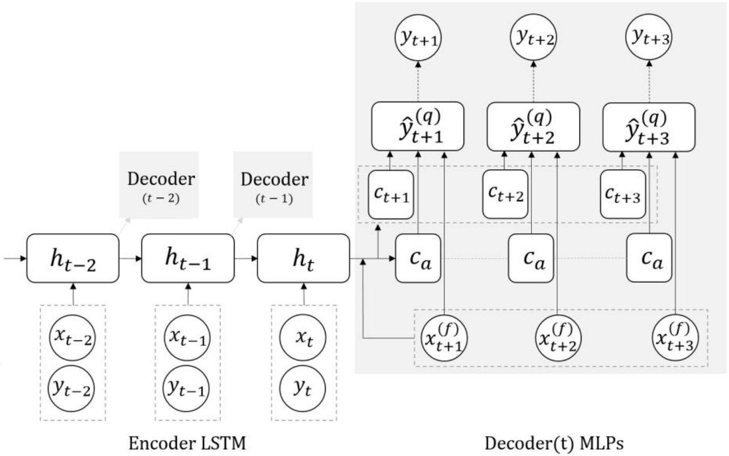
- Gradient Boosting Models
  - Sequentially build trees where each tree corrects errors of the previous ones, useful for capturing complex patterns in data.

Machine Learning Models

- Neural Networks
  - Can model non-linear relationships in time-series data
- Recurrent Neural Network
  - Specifically designed for sequential data, so proved useful for most cases.
- Convolutional Neural Networks
  - Applied to time series data to extract features through convolutional layers

Machine Learning Models

- Encoder-Decoder Model
  - Past known features and target are used as features for encoder
  - Future known features are used as features for decoder
  - Decoder takes input from encoder as well as future features



PROPHET

- Model developed by Facebook for business
- Allows incorporation of Holidays
- Handle gaps in the data
- Models non-linear trends with automatic changepoint detection

Hybrid Models

- SARIMAX + Random Forest
    - SARIMAX used to fit the past values and generate base forecast value
    - Random Forest takes the base forecast value and uses external features to predict the residual between true value and base forecast.
    - **Final forecast = Base forecast + Predicted residual**

Hybrid Models

- SARIMAX + Neural Network
  - SARIMAX used to capture linear relationship in data
  - Neural networks’ non-linear capability to increase accuracy
- Ensemble methods
  - Combine forecast from various models using ensemble techniques like averaging and weighted averaging

Hierarchical Forecasting

- In many scenarios, there is hierarchy, and we might need forecast in different hierarchies
- We take data in lowest hierarchy
- Use **Aggregator** to aggregate data in higher hierarchies.
- Use Individual **Forecaster** to generate forecast in different levels
- Use **Reconciler** to reconcile the forecasts in different levels.

LLM based foundation models

- Pre-trained on huge datasets
- Can be directly used for inference or fine-tuned on our data.
- Some promising foundation model for time-series
    - Foundation Models for Time Series Analysis: A Tutorial and Survey
    - **Time-LLM** (https://github.com/KimMeen/Time-LLM)
    - **TIMESFM** by google (https://huggingface.co/google/timesfm-1.0-200m)
    - **Lag-llama** - include demo colabs (https://huggingface.co/time-series-foundation-models/Lag-Llama)

1 Introduction to Time Series

2 Components of Time Series

3 Time Series Forecasting Models

4 **Evaluation of Time Series Models**

Evaluation of time series models

- Evaluation metrics
  - Mean Absolute Error
    - $\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
    - Usually, absolute error is proportional to true value
    - If true value is small, error is small and same for large values
  - Mean Absolute Percentage Error
    - $\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$
    - Higher percentage error for smaller true values can be misleading
    - If true=1, pred=2, percentage error = 100%

Evaluation of time series models

- Evaluation metrics
  - Mean Square Error
    - $$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
    - Penalizes higher error more
  - Root Mean Square Error
    - $$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$
    - Easier to visualize than MSE because of relatively smaller values

Evaluation of time series models

- Evaluation metrics to check balance in over and under forecast
    - Mean Error
        - $$\mathrm{ME} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)$$
        - Has to be near to 0 for balanced over and under forecast
    - Forecast Ratio
        - $$\text{Forecast Ratio} = \frac{\text{Total Overforecast}}{\text{Total Underforecast}}$$
        - Has to be near to 1 for balanced over and under forecast
    - Distribution of error and percentage error

Thank you for your attention
The end of the presentation