Lecture 6: Stock Forecasting

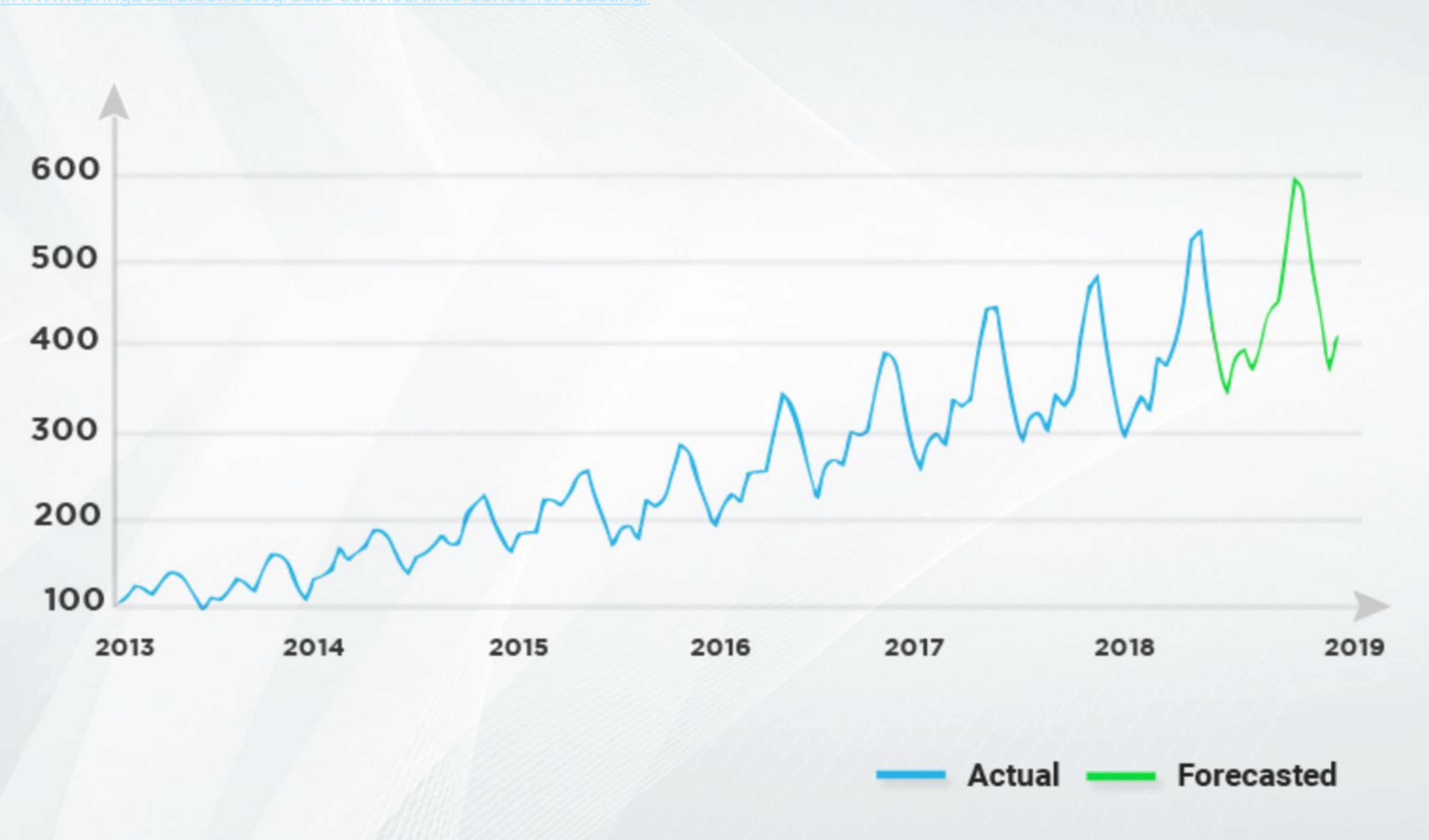
https://github.com/kaopanboonyuen/SC310005_ArtificialIntelligence_2025s1

Teerapong Panboonyuen
https://kaopanboonyuen.github.io

What is Time Series (TS)?

A time series is a sequence of data points collected or recorded at successive equally spaced points in time. These data capture how something evolves over time — like stock prices, weather measurements, or monthly sales. Analyzing time series helps us understand patterns, trends, seasonality, and make forecasts.

https://www.springboard.com/blog/data-science/time-series-forecasting/

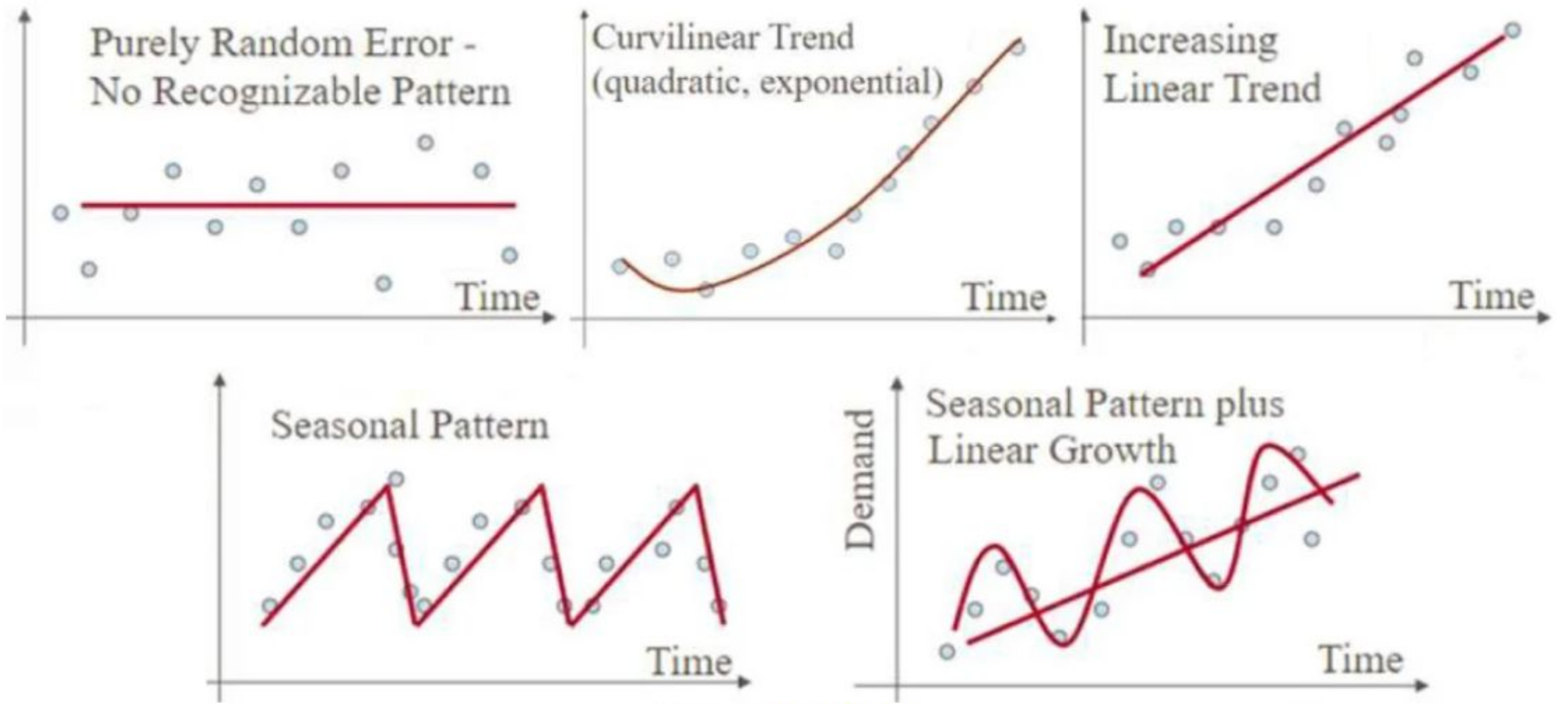


https://www.springboard.com/blog/data-science/time-series-forecasting/

When Should You Use Time Series Forecasting?

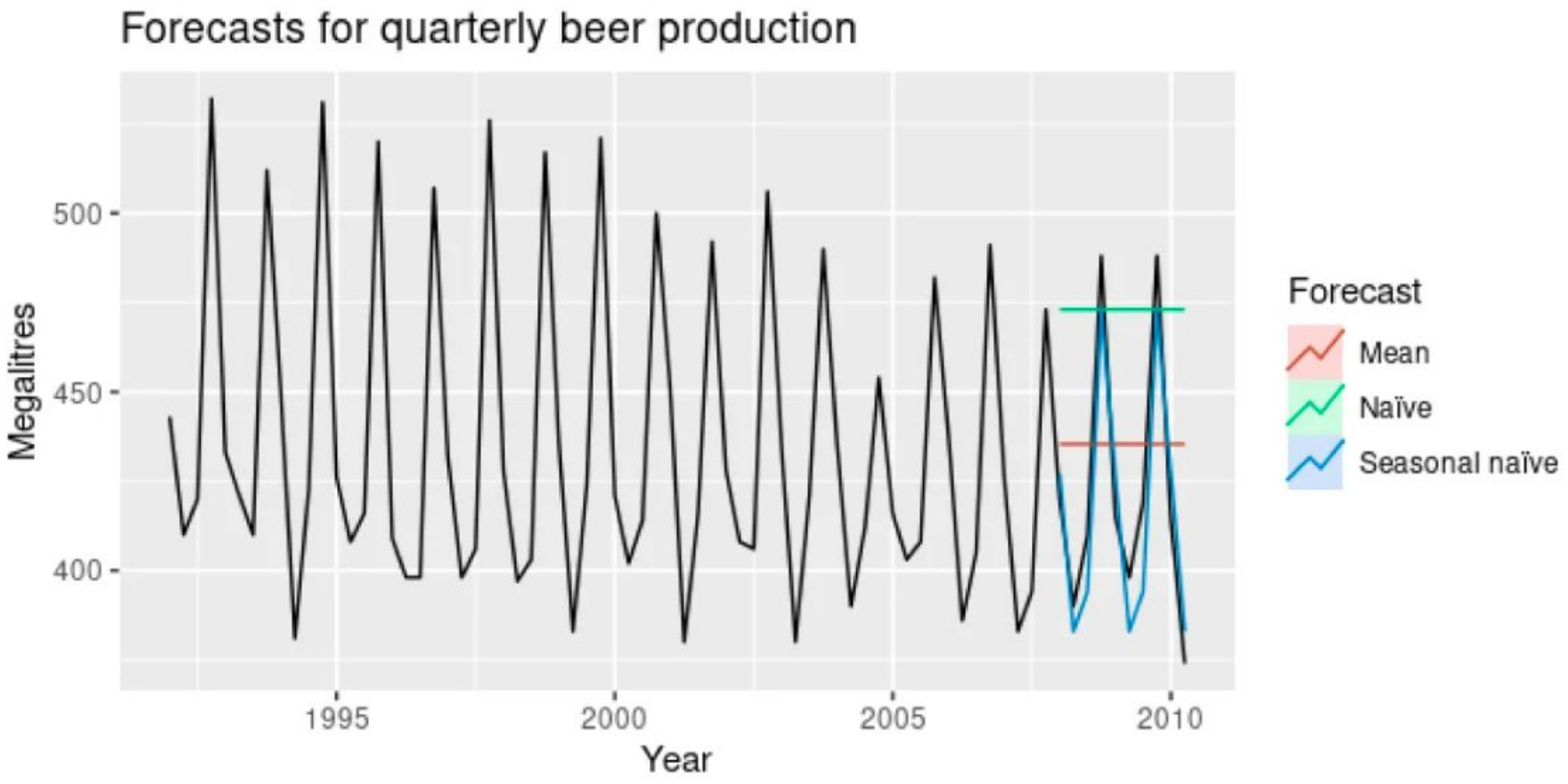
https://www.springboard.com/blog/data-science/time-series-forecasting/

Defining the Business Problem



Source: Medium

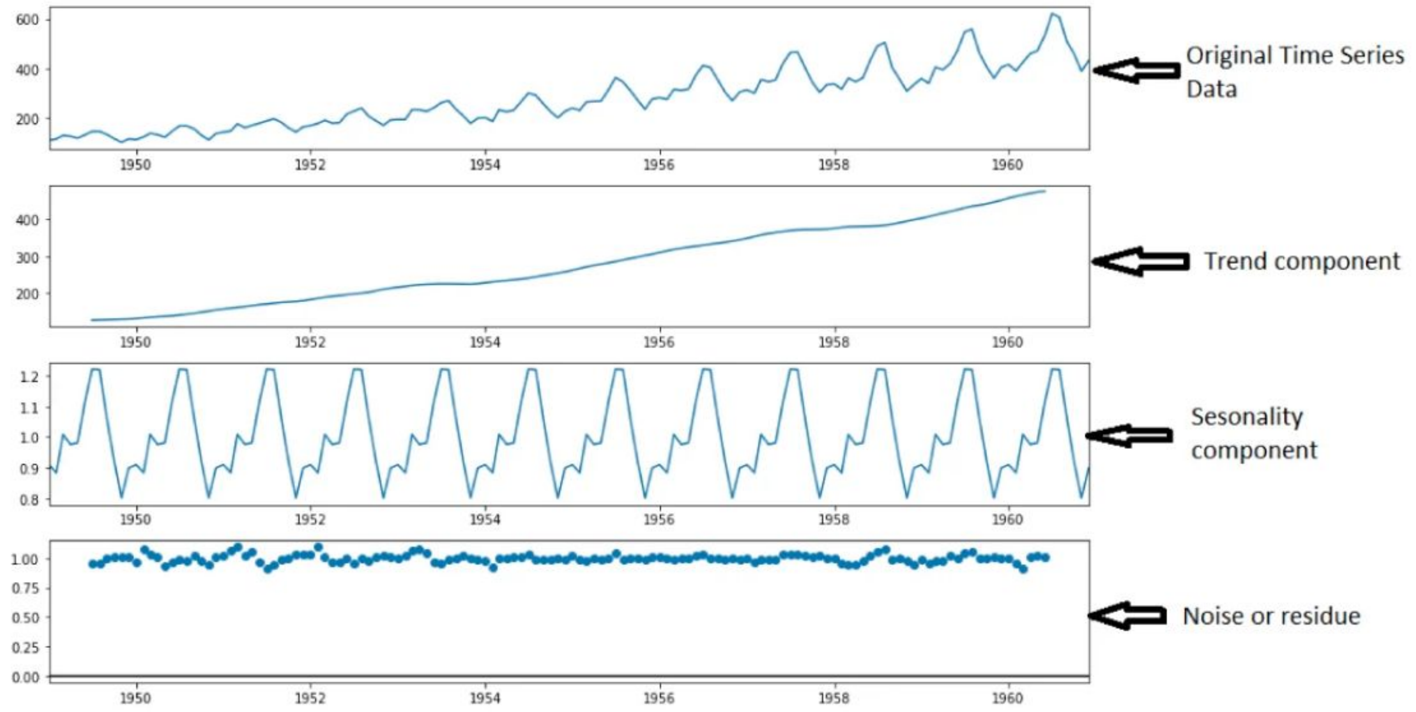
https://www.springboard.com/blog/data-science/time-series-forecasting/



Source: OTexts

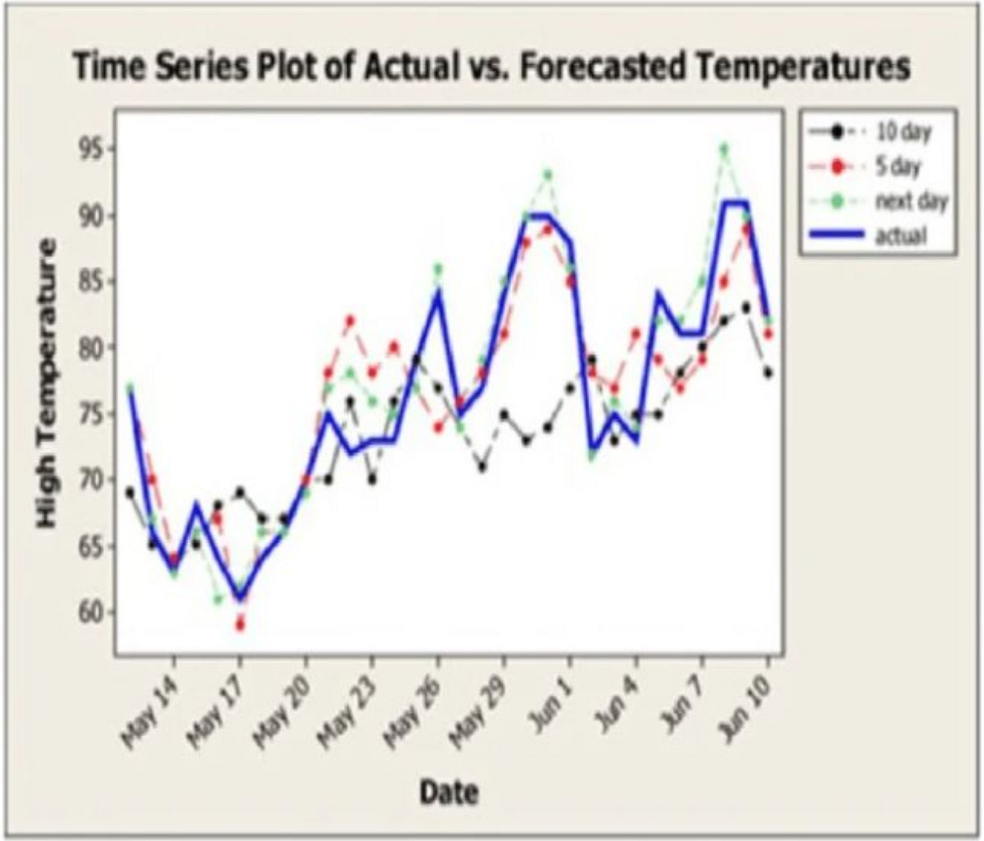
https://www.springboard.com/blog/data-science/time-series-forecasting/

Determining How Often Model Outputs Are Needed



Source: Medium

https://www.springboard.com/blog/data-science/time-series-forecasting/



Source: ResearchGate

- Air temperature
- Time of day
- Wind speed
- Wind direction
- Atmospheric pressure

https://www.springboard.com/blog/data-science/time-series-forecasting/

Forecasting Stock Price Changes

Source: Unsplash

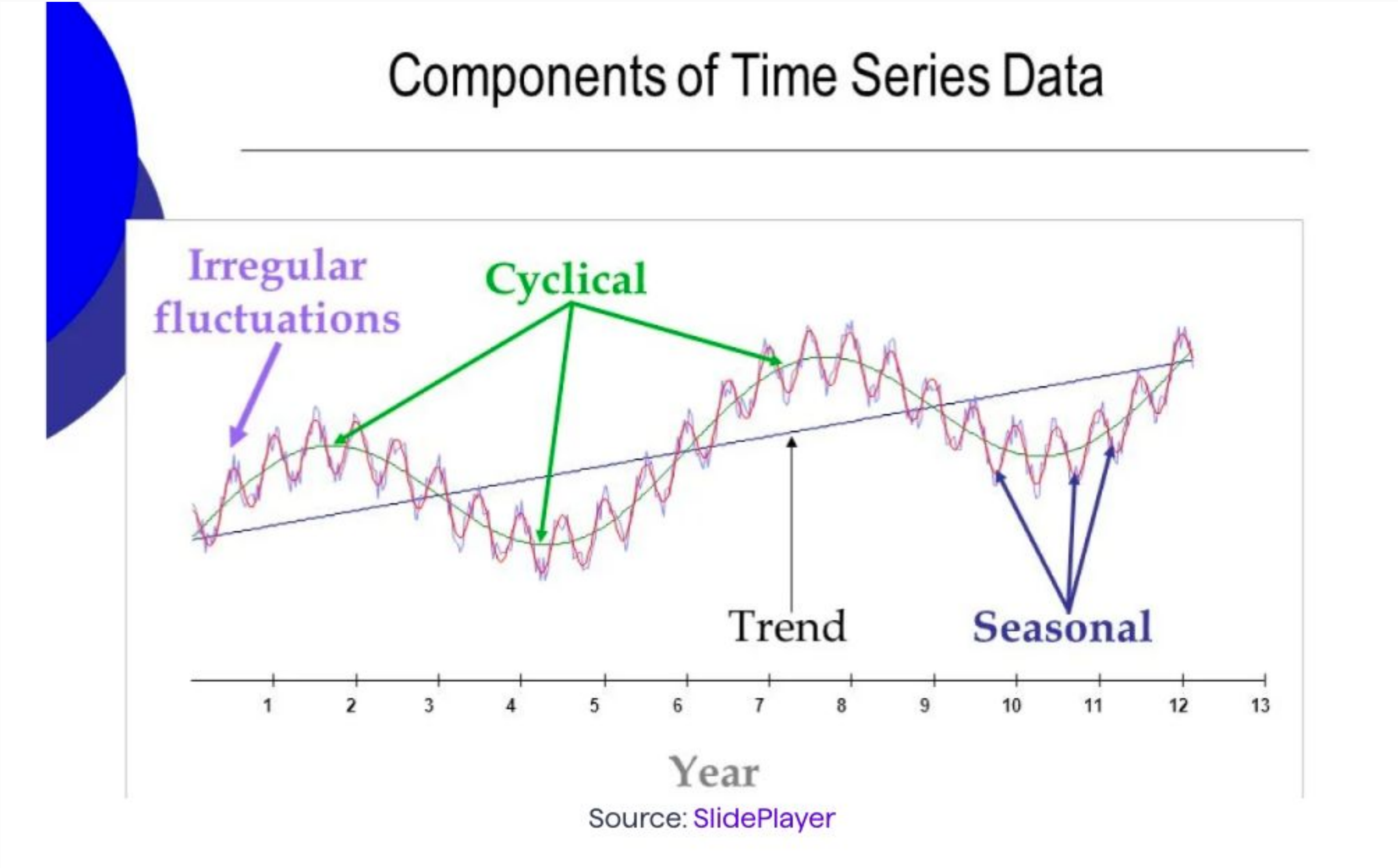
⚠️ Disclaimer

This content is intended strictly for academic and educational purposes.

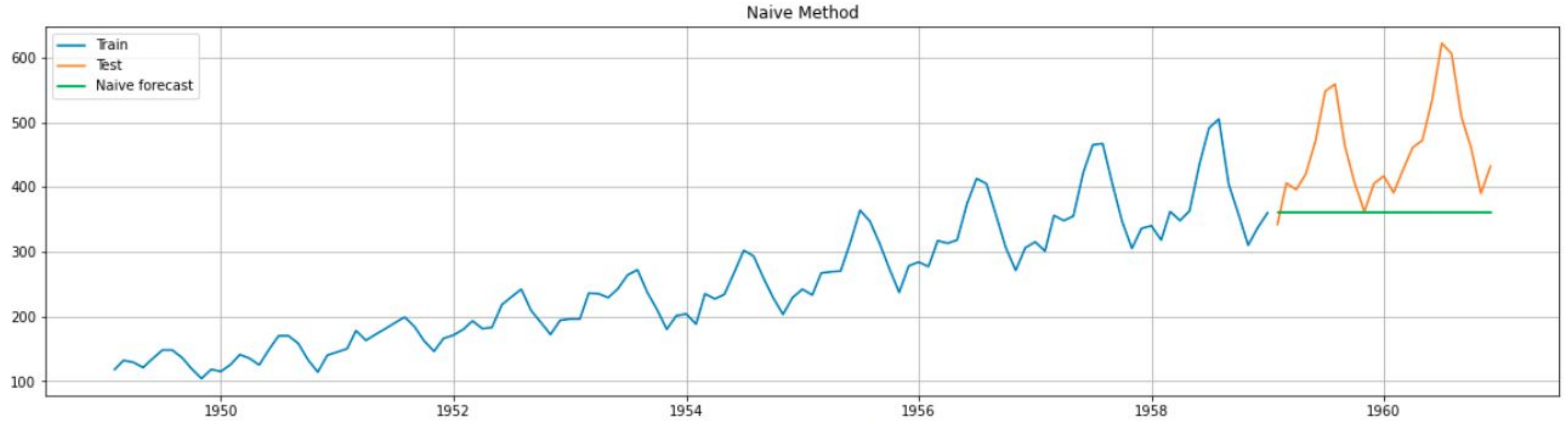
- All stock data used in this lesson — including historical prices, trends, and forecasts — are for learning, experimentation, and demonstration only.
- Nothing presented here should be interpreted as financial advice, investment guidance, or stock recommendation.
- Models used (e.g., MA, ARIMA, LSTM) are simplified for teaching and do not account for real-world financial risks or market volatility.
- Always consult a certified financial advisor before making investment decisions.

📊 **Dataset Source:** Publicly available Thai stock data
📘 **Purpose:** Teach fundamentals of time series forecasting in AI and data science

https://www.springboard.com/blog/data-science/time-series-forecasting/

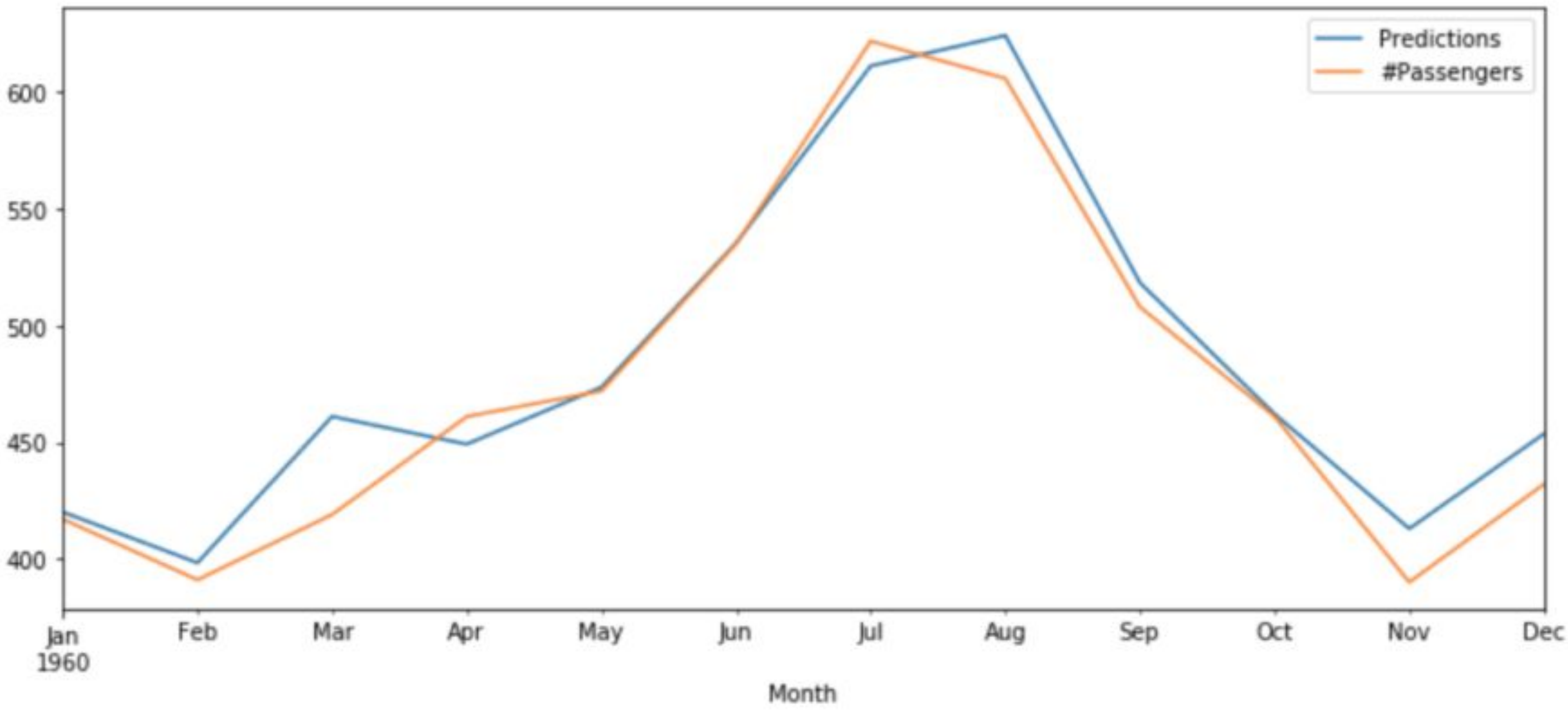


https://www.springboard.com/blog/data-science/time-series-forecasting/



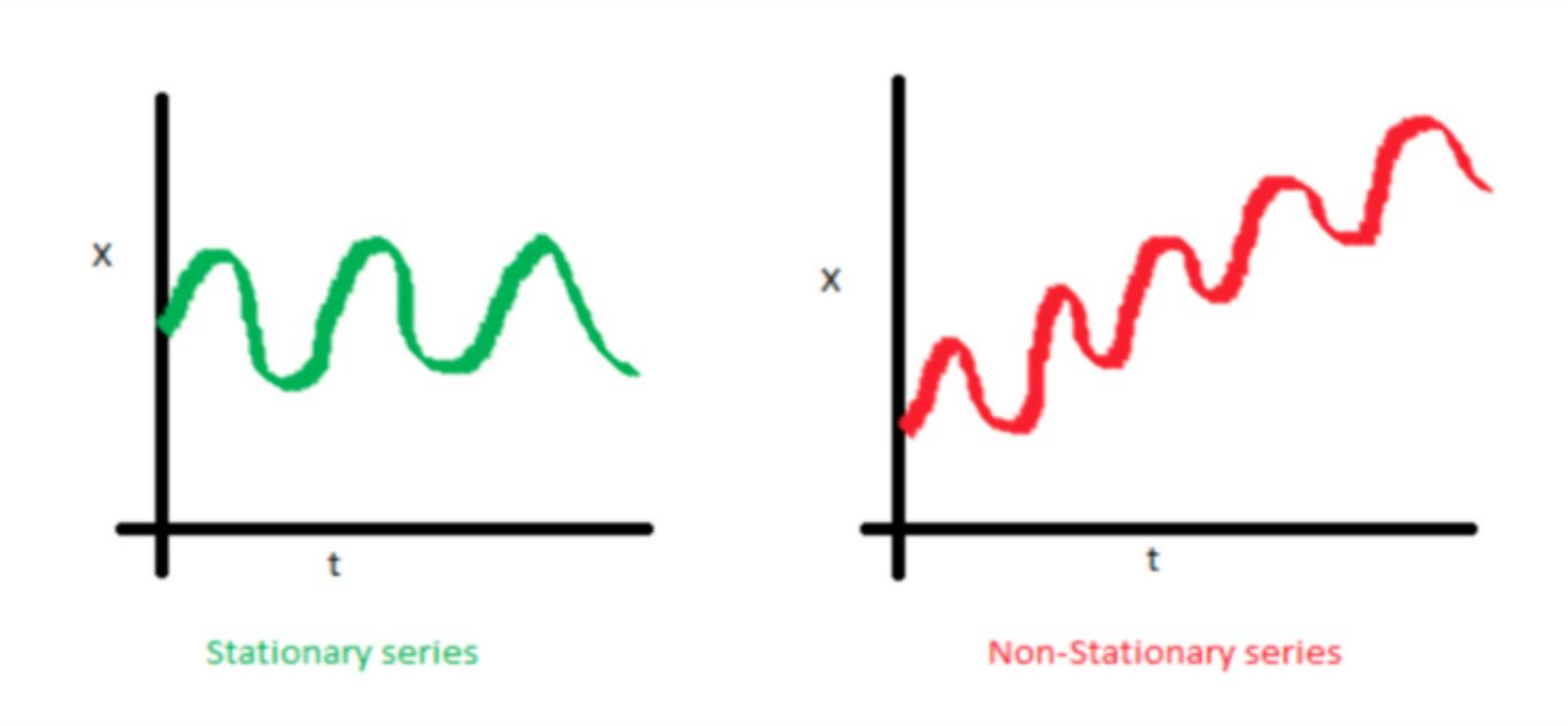
Source: Kaggle

https://www.springboard.com/blog/data-science/time-series-forecasting/

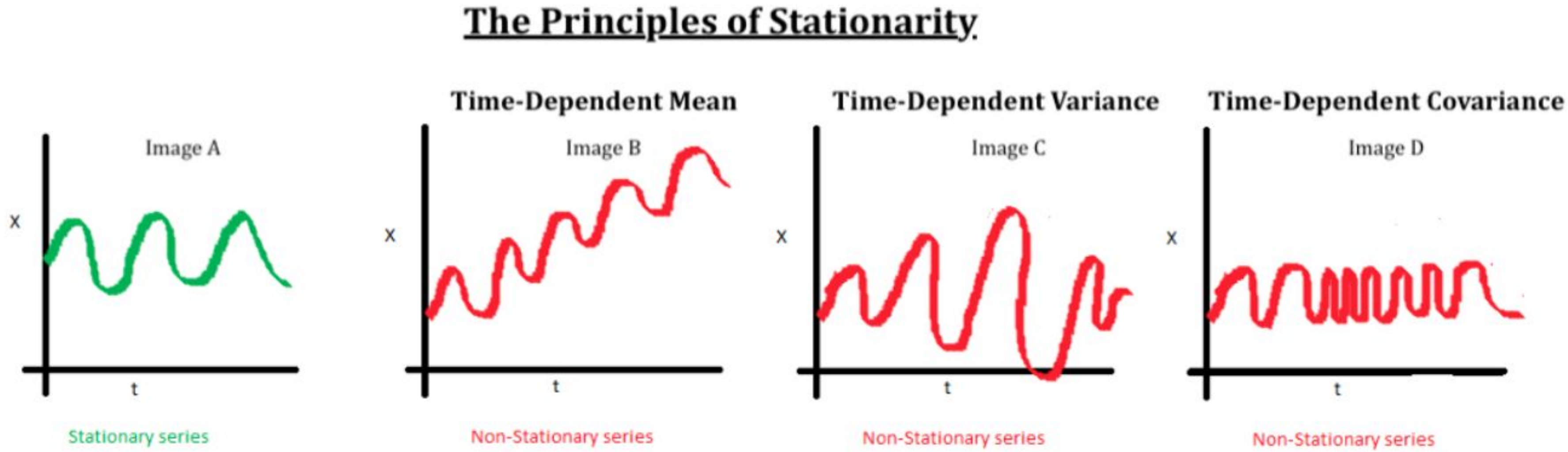


Source: Geeksforgeeks

https://www.datalytyx.com/choosing-the-right-forecast-model-for-time-series-data/



What is a stationary series and how important is it?



Stationary vs non-stationary series. Source: Mitrani 2020.*

Basics You Should Know About Time Series

- **Stationary vs Non-Stationary**

  Stationary series have constant mean, variance, and covariance over time — easier to model.

  Non-stationary series show trends, changing variance, or seasonal effects — often need transformations like differencing to make stationary.

- **Trend**

  The long-term upward or downward movement in data.

- **Seasonality**

  Regular repeating patterns or cycles over fixed periods.

- **Noise**

  Random variation or “leftover” data unexplained by trends or seasonality.

Sample Dataset: Thai Stock Prices (Jan 2024 - July 2025)

We will work with 5 stocks:

- **PTT** (PTT.BK): Thailand's largest oil and gas company — key energy sector player.
- **ADVANC** (ADVANC.BK): Leading telecom provider — essential for digital economy.
- **SCB** (SCB.BK): Siam Commercial Bank — one of Thailand's biggest banks.
- **CPALL** (CPALL.BK): Operates 7-Eleven stores across Thailand — retail powerhouse.
- **KBANK** (KBANK.BK): Kasikornbank — major banking institution.

Each data record contains:

- **Date**: The trading date
- **Close**: Closing stock price for the day
- **Volume**: Number of shares traded
- **Stock**: Stock symbol/name

Exploratory Data Analysis (EDA)

- Visualize price trends over time for each stock
- Check volume fluctuations and trading activity
- Identify patterns like trends, seasonality, outliers
- Calculate moving averages (MA) to smooth data

Techniques for Time Series Forecasting

- **Moving Average (MA)**

  Smoothes data by averaging past values — helps reveal trends.

- **ARIMA (AutoRegressive Integrated Moving Average)**

  Captures different aspects of time series (auto-regression, differencing, moving averages).

- **Linear Extrapolation**

  Extends current trend linearly into the future — simple but limited.

- **Exponential Smoothing**

  Weighs recent data more heavily to adapt quickly to changes.

- **Naive MA Extension + Noise**

  Flat forecast using moving average with added random noise for uncertainty.

- **Facebook Prophet**

  Flexible forecasting tool handling seasonality, holidays, and trend changes.

- **Deep Learning: LSTM (Long Short-Term Memory networks)**

  Powerful RNN model capturing complex time dependencies in sequential data.

Moving Average (MA)

- *What?* Smooths out short-term fluctuations by averaging past data points.
- *Why?* Helps reveal the underlying trend and reduce noise.
- *Example:*
  For PTT stock prices, a 5-day moving average shows the general price direction by smoothing daily ups and downs.

Moving Average (MA)

What is it?

A simple technique to smooth time series data by averaging a fixed number of recent observations, which helps reveal underlying trends by reducing noise.

Formula (Simple Moving Average with window size k):

$$MA_t = \frac{1}{k} \sum_{i=0}^{k-1} y_{t-i}$$

- MA_t: Moving average at time t
- y_{t-i}: Actual observed value at time $t - i$
- k: Window size (number of periods to average)

Example:

Given stock closing prices over 5 days:

$100, 102, 101, 105, 107$

Calculate 3-day Moving Average on day 5 ($t = 5$):

$$MA_5 = \frac{y_5 + y_4 + y_3}{3} = \frac{107 + 105 + 101}{3} = \frac{313}{3} = 104.33$$

This smooths the price at day 5 to 104.33, showing the short-term trend.

ARIMA (AutoRegressive Integrated Moving Average)

- *What?* Combines three parts:
    - AutoRegression (uses past values),
    - Integration (differencing to make data stationary),
    - Moving Average (modeling error terms).
- *Why?* Good for capturing complex patterns like trends and seasonality.
- *Example:*
  Modeling SCB bank's stock prices by differencing to remove trend, then fitting AR and MA terms to predict future prices.

ARIMA (AutoRegressive Integrated Moving Average)

What is it?

ARIMA is a powerful and flexible time series forecasting method that models three key components:

- **AR (AutoRegression):** Uses the relationship between an observation and some number of lagged observations (past values).
- **I (Integrated):** Differencing the data to make it **stationary** (remove trends/seasonality).
- **MA (Moving Average):** Models the relationship between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA Model Notation: $\mathrm{ARIMA}(p, d, q)$

- p = number of lag observations in the model (AR order)
- d = degree of differencing (number of times data is differenced to make it stationary)
- q = size of the moving average window (MA order)

Mathematical Formulation:

1. **Differencing (to get stationary series):**

$$y'_t = y_t - y_{t-1} \quad \text{(1st order differencing)}$$

Repeat differencing d times if needed.

2. **AR part (order p):**

$$y'_t = c + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + \cdots + \phi_p y'_{t-p} + \epsilon_t$$

- c: constant
- ϕ_i: coefficients for lagged terms
- ϵ_t: white noise error

3. **MA part (order q):**

$$y'_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

- θ_i: coefficients for lagged errors

Step 1: Check if differencing is needed (stationarity)

Calculate first difference $y'_t = y_t - y_{t-1}$:

Day	Price y_t	Difference $y'_t = y_t - y_{t-1}$
1	100	- (no previous day)
2	102	102 - 100 = 2
3	101	101 - 102 = -1
4	105	105 - 101 = 4
5	107	107 - 105 = 2

So differenced series y' = **2, -1, 4, 2**

Step 2: Assume an AR(1) model on differenced data

$$y'_t = \phi_1 y'_{t-1} + \epsilon_t$$

Let's pick $\phi_1 = 0.5$ (example coefficient).

Step 3: Forecast y'_5 (the difference at day 5) using y'_4

Given:

- $y'_4 = 4$
- Assume noise $\epsilon_5 = 0$ (for simplicity)

Calculate:

$$\hat{y}'_5 = 0.5 \times y'_4 + \epsilon_5 = 0.5 \times 4 + 0 = 2$$

Step 4: Get predicted price for day 6

Recall:

$$y_6 = y_5 + \hat{y}'_5$$

Given $y_5 = 107$, predicted y_6 is:

$$y_6 = 107 + 2 = 109$$

Summary:

Day	Price y_t	Difference y'_t	Forecast $\hat{y}'_t$
1	100	-	-
2	102	2	-
3	101	-1	-
4	105	4	-
5	107	2	-
6	?	?	2

Forecasted price for day 6 is **109**.

Key Points:

- **Differencing** makes the series stationary (constant mean & variance) — critical before AR or MA.
- **AR** captures momentum from past values.
- **MA** captures shock effects from past errors.
- Combine all for robust forecasting.

Linear Extrapolation

- *What?* Extends the current linear trend forward into the future.
- *Why?* Simple baseline forecast when trends are fairly consistent.
- *Example:*
  If ADVANC stock shows a steady upward trend, linear extrapolation projects that same slope into the next 30 days.

Exponential Smoothing

- *What?* Weights recent observations more heavily than older ones for forecasting.
- *Why?* Adapts quickly to changes in trend or level.
- *Example:*
  CPALL's retail stock price reacting quickly to market events, where exponential smoothing captures sudden shifts better than MA.

Exponential Smoothing

What is it?

A smoothing technique that gives more weight to recent observations, allowing the model to respond faster to recent changes in the data.

Formula:

$$S_t = \alpha \cdot y_t + (1 - \alpha) \cdot S_{t-1}$$

- S_t: Smoothed value at time t
- y_t: Actual value at time t
- S_{t-1}: Previous smoothed value
- α: Smoothing factor between 0 and 1 (higher α means more weight on recent data)

Example:

Given stock closing prices over 5 days:

$100, 102, 101, 105, 107$

Example:

Using the same prices:

Day 1 price: 100 (Initialize $S_1 = 100$)

Let $\alpha = 0.5$

Calculate S_2 to S_5:

$$S_2 = 0.5 \times 102 + 0.5 \times 100 = 101$$

$$S_3 = 0.5 \times 101 + 0.5 \times 101 = 101$$

$$S_4 = 0.5 \times 105 + 0.5 \times 101 = 103$$

$$S_5 = 0.5 \times 107 + 0.5 \times 103 = 105$$

Exponential smoothing adapts smoothly, reacting quicker to new price changes compared to moving average.

Naive MA Extension + Noise

- *What?* Forecast assumes flat price equal to last moving average value plus some random noise.
- *Why?* Provides a simple baseline that incorporates uncertainty.
- *Example:*
  KBANK's forecast set as last 5-day MA plus small random variation to mimic daily fluctuations.

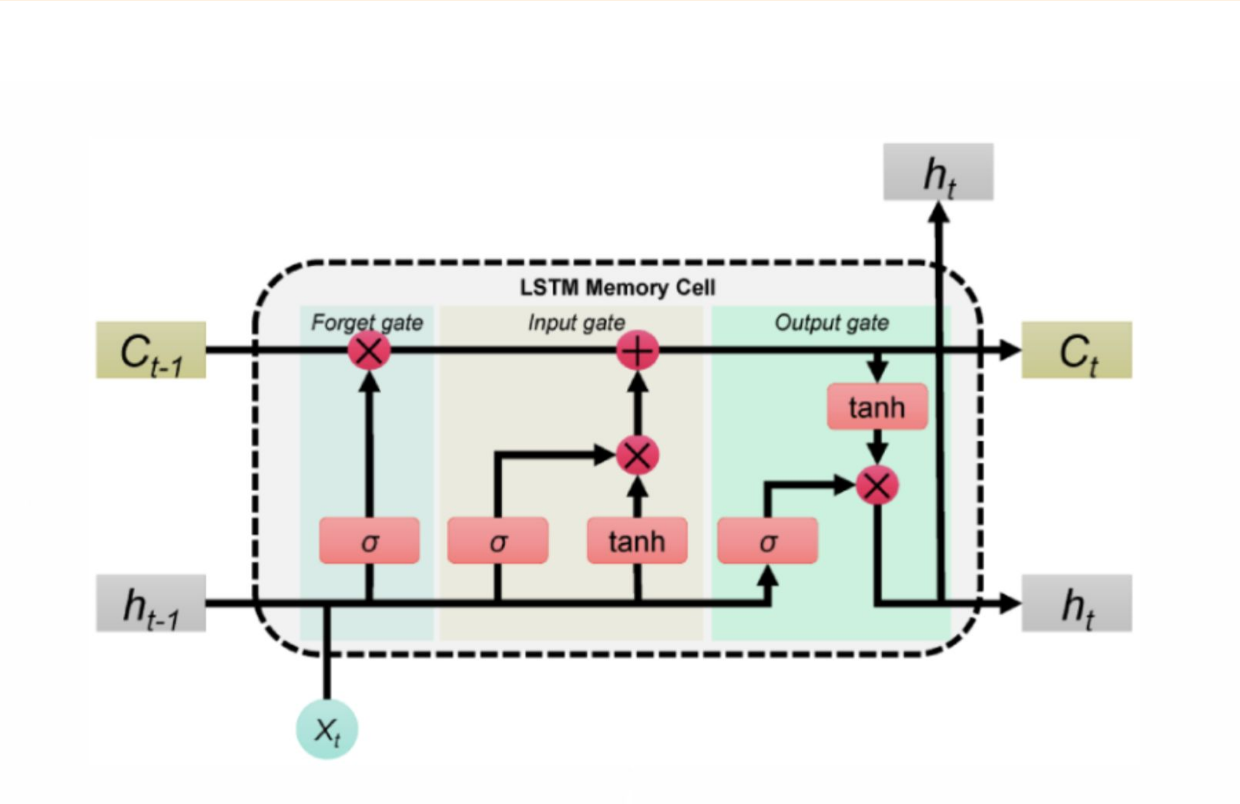
Facebook Prophet

- *What?* Advanced model designed for time series with multiple seasonalities and holiday effects.
- *Why?* Easy to use, handles missing data, trend changes, and special events.
- *Example:*
  Forecasting stock prices during Thai holidays where trading volume dips — Prophet models holiday impact explicitly.

Deep Learning: LSTM (Long Short-Term Memory networks)

- *What?* Neural network designed to learn long-range dependencies in sequential data.
- *Why?* Captures complex temporal patterns beyond linear or statistical models.
- *Example:*
  Predicting next-day price for PTT stock by feeding the past 10 days' prices into LSTM, learning hidden nonlinear relations.

https://dida.do/what-is-an-lstm-neural-network



Evaluation Metrics: How Do We Measure Forecast Quality?

- **RMSE (Root Mean Squared Error)**

  Measures average magnitude of errors between predicted and actual values — sensitive to large errors.

- **MAE (Mean Absolute Error)**

  Average absolute difference between forecast and actual — easier to interpret.

Calculating RMSE and MAE helps you compare model accuracy on stock predictions.

Evaluation Metrics: How Do We Measure Forecast Quality?

Root Mean Squared Error (RMSE)

Measures the average magnitude of prediction errors — penalizes large errors more heavily.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- y_i = actual value
- $\hat{y}_i$ = predicted value
- n = number of observations

Example:

Actual prices: [100, 105, 110, 115]

Predicted prices: [102, 107, 108, 120]

Calculate squared errors:

$$(100 - 102)^2 = 4, \quad (105 - 107)^2 = 4, \quad (110 - 108)^2 = 4, \quad (115 - 120)^2 = 25$$

Mean squared error:

$$\frac{4 + 4 + 4 + 25}{4} = \frac{37}{4} = 9.25$$

RMSE:

$$\sqrt{9.25} \approx 3.04$$

Mean Absolute Error (MAE)

Measures average absolute differences — easier to interpret as "average error."

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Using the same example:

Absolute errors:

$$|100 - 102| = 2, \quad |105 - 107| = 2, \quad |110 - 108| = 2, \quad |115 - 120| = 5$$

MAE:

$$\frac{2+2+2+5}{4} = \frac{11}{4} = 2.75$$

Why use both?

- RMSE penalizes large errors more, good for sensitive applications

- MAE gives intuitive average error magnitude

Both help evaluate and compare forecasting models on stock prices or any time series data.

Today's Lab

We will explore the 5 stocks:

`PTT.BK`, `ADVANC.BK`, `SCB.BK`, `CPALL.BK`, `KBANK.BK`

- Visualize their historical prices
- Perform basic EDA and compute moving averages
- Implement forecasting techniques like MA and ARIMA
- Evaluate your models using RMSE and MAE

Homework

Repeat the analysis and forecasting for the following stocks:

`AOT.BK`, `BDMS.BK`, `BAY.BK`, `ESSO.BK`, `HMPRO.BK`

Use any technique (MA, ARIMA, Prophet, or LSTM) to achieve the lowest RMSE possible.