Time Series Analysis

Machine Learning – Comp 479

Snigdha Upadhyay
Md Sharuf Hossain

Introduction

What is time series analysis -

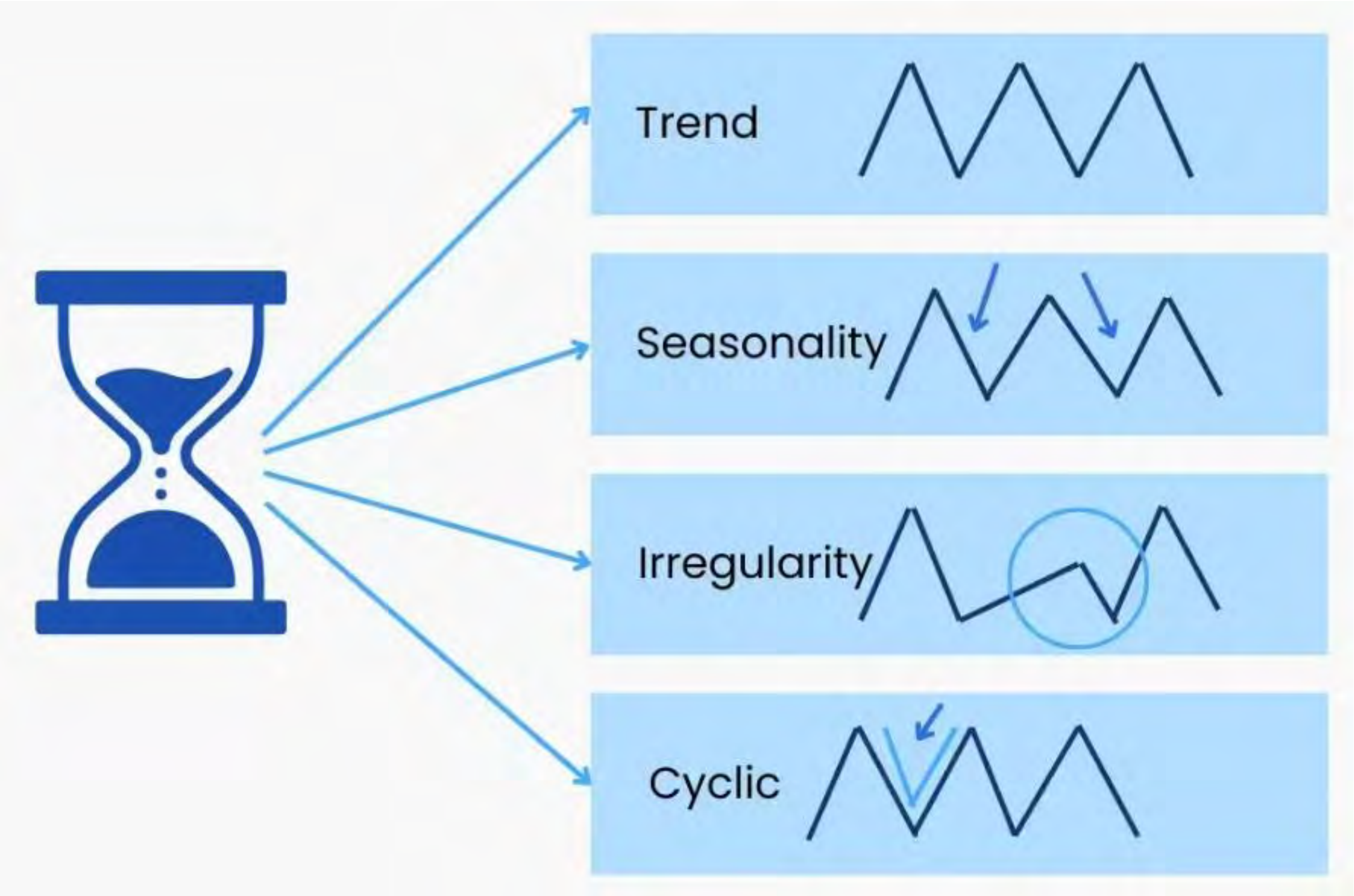
- Sequence of data points collected, recorded, or measured at successive, evenly-spaced time intervals.
- Data point represents observations or measurements taken over time .
- Time series data is represented graphically with time on the horizontal axis and the variable of interest on the vertical axis

Usage -

- Analyze data that is collected over time.
- It helps in understanding patterns like trends, seasonality, and cycles
- Used in fields like finance, economics, weather forecasting
- Aids in planning, budgeting, strategizing and decision making eventually driving efficiency and competitiveness.

Importance of Time Series Analysis

- **Predict Future Trends**
  - Allowing businesses to anticipate key variables like market demand , stock price
  - Enables proactive decision-making.
- **Detect Patterns and Anomalies**
  - By examining sequential data points, helps detect recurring patterns and anomalies
  - Providing insights into underlying behaviors and potential outliers.
- **Risk Mitigation**
  - Spotting potential risks, businesses can develop strategies to mitigate them.
  - enhance overall risk management.
- **Strategic Planning:**
  - Time series insights inform long-term strategic planning, guiding decision-making across sectors
- **Competitive Edge:**
  - Enables businesses to optimize resource allocation and plan ahead.
  - Staying ahead of market trends, responding to changes, and making data-driven decisions, businesses gain a competitive edge.



Components of Time Series Data

Data consist of observations recorded over time at regular intervals

Analyzed by breaking it down into four primary components-

- **Trend** – Long term movement of data over time.
- **Seasonality** – Periodic fluctuation or pattern occur at a regular interval within time.
- **Cyclic Variation** - longer-term fluctuations that do not have a fixed period like seasonality.
- **Irregularity (Noise)** - Unpredicted or random fluctuation n data.

These components help identify patterns, trends, and irregularities in the data.

Components of Time Series Data

Trend

- A long-term upward or downward movement in the data,
- Indicates a general increase or decrease over time.
  - **Upward Trend:** shows a general increase over time
  - **Downward Trend:** shows a general decrease over time
  - **Horizontal Trend:** shows no significant change over time
  - **Non-linear Trend:** shows a more complex pattern of change, including upward or downward trends

Cyclicity

- Refers to the repeated patterns or periodic fluctuations that occur in the data over an unspecified time interval.
- Unlike seasonality , cyclicity is not limited to a fixed time interval and can be of different frequencies.

Seasonality

- Patterns that repeat over a regular time period like a day, a week, a month, or a year.
- These patterns arise due to regular events like holidays, weekends, or the changing of seasons.
- There are several types of seasonality in time series data, including-.
  - **Weekly Seasonality:** repeats over a 7-day period , seen in data such as sales or energy usage.
  - **Monthly Seasonality:** repeats over a 30- or 31-day period , seen in data such as sales or weather patterns.
  - **Annual Seasonality:** repeats over a 365- or 366-day period, seen in data such as sales, agriculture, or tourism patterns.
  - **Holiday Seasonality:** caused by special events such as holidays, festivals, or events, seen in data such as sales, traffic, or entertainment patterns.

Ir-regularity

- Refers to unexpected or unusual fluctuations in the data that do not follow the general pattern of the data.
- These fluctuations can occur for various reasons, such as measurement errors, unexpected events, or other sources of noise.
- Irregularities can have a significant impact on the accuracy of time series models and forecasting.
- They can obscure underlying trends and seasonality patterns in the data.

Data Preparation

Data Preprocessing

- Clean, transform, and prepare the data.
- The goal is to ensure that the data is in a suitable format for subsequent analysis or modeling.

  - **Handling Missing Values :** to ensure continuity and reliability in analysis.
  - **Dealing with Outliers:** Identifying outliers, which can distort analysis results.
  - **Stationarity and Transformation:** Ensuring that the statistical properties such as mean and variance, remain constant over time. Techniques like differencing, detrending, and deseasonalizing are used to achieve stationarity.

Data Decomposition

- Process of separating a time series into its constituent components - Trend, seasonality, and noise.

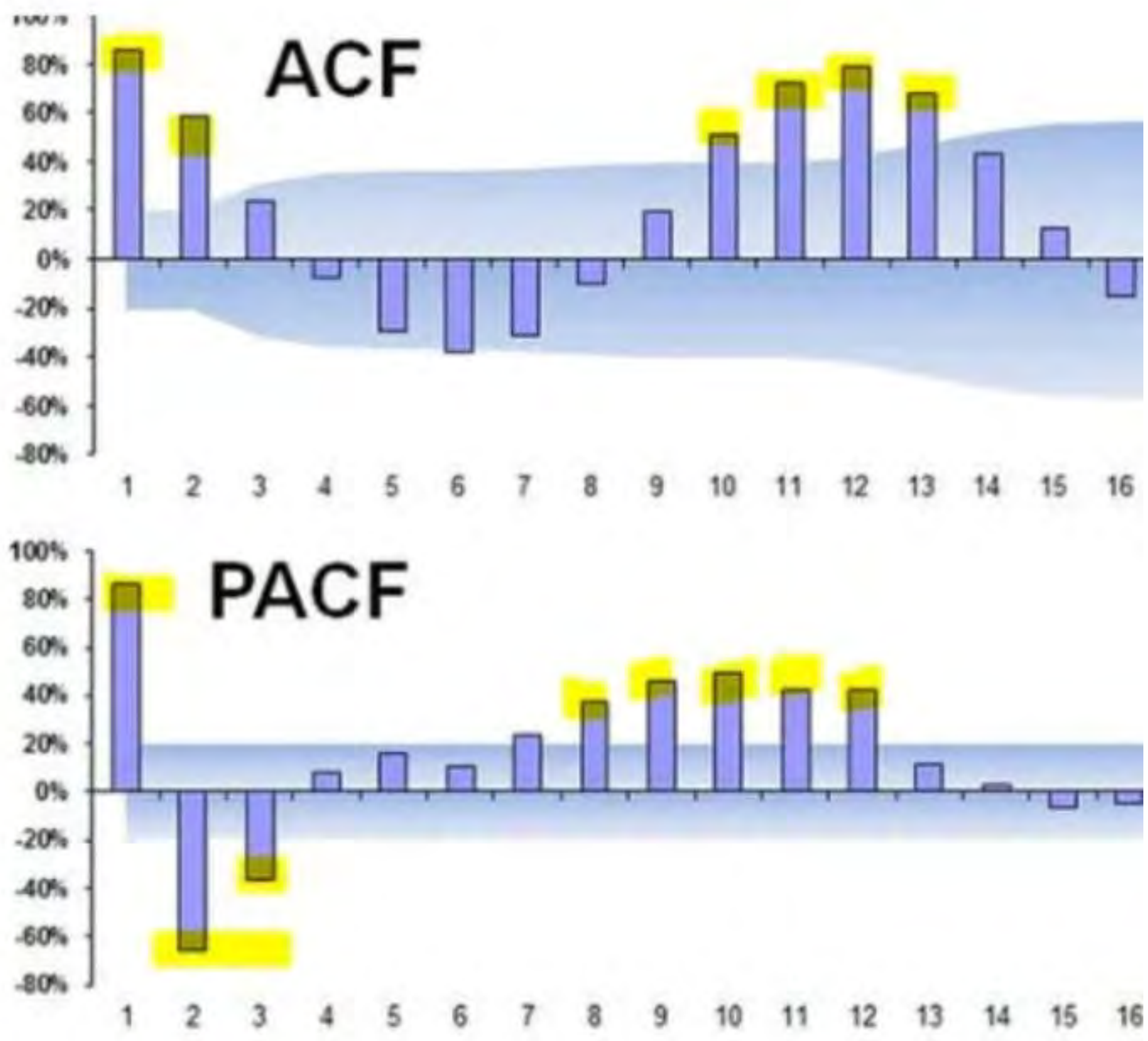
Types of Time Series Decomposition Techniques

Additive Decomposition:

- The time series is expressed as the sum of its components
- $Y(t)=Trend(t)+Seasonal(t)+Residual(t)$

Multiplicative Decomposition:

- The time series is expressed as the product of its components:
- $Y(t)=Trend(t)*Seasonal(t)*Residual(t)$



Key Concepts

- **Autocorrelation Analysis(ACF)**
  - Statistical concept that assesses the degree
    of correlation/Similarity between the values of a variable at different time points.
  - Detects repeating patterns and trends in time series data.
- **Partial Autocorrelation Functions (PACF)**
  - Measures the partial correlation between a stationary time series and its own past values, considering and accounting for the values at all shorter lags.
  - It is different from the autocorrelation function, which does not control other lags.

Time Series Forecasting

Statistical technique used to predict future values of a time series based on past observations.

It's like looking into the future of data points plotted over time.

By analyzing patterns and trends in historical data, Time Series Forecasting helps make informed predictions about what may happen next

Assisting in decision-making and planning for the future.

Techniques -

- Autoregressive (AR) Model.
- Autoregressive Integrated Moving Average (ARIMA).
- ARIMAX.
- Seasonal Autoregressive Integrated Moving Average (SARIMA).
- SARIMAX.
- Vector Autoregression (VAR) Models.
- Theta Method.
- Exponential Smoothing Methods.
- Gaussian Processes Regression.
- Generalized Additive Models (GAM).
- Random Forests.
- Gradient Boosting Machines (GBM).
- State Space Models.
- Dynamic Linear Models (DLMs).
- Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) Networks.
- Hidden Markov Model (HMM).

Autoregressive(AR)

- Predicts future values based on linear combinations of past values of the same time series.
- Current value is modeled as a linear function of its previous(past) p values, plus a random error term.
- The order of the autoregressive model (p) determines how many past values are used in the prediction.
- **Notation**: AR(p), where p is the number of lagged observations used.
- **Assumes stationarity.**
- Partial autocorrelation(PACF) plot helps determine optimal p.

Use case

- Good for data with significant autocorrelation in recent lags.
- Short term Prediction , Smaller data set

Example

- Predicting today's stock price based on the last few days' prices.

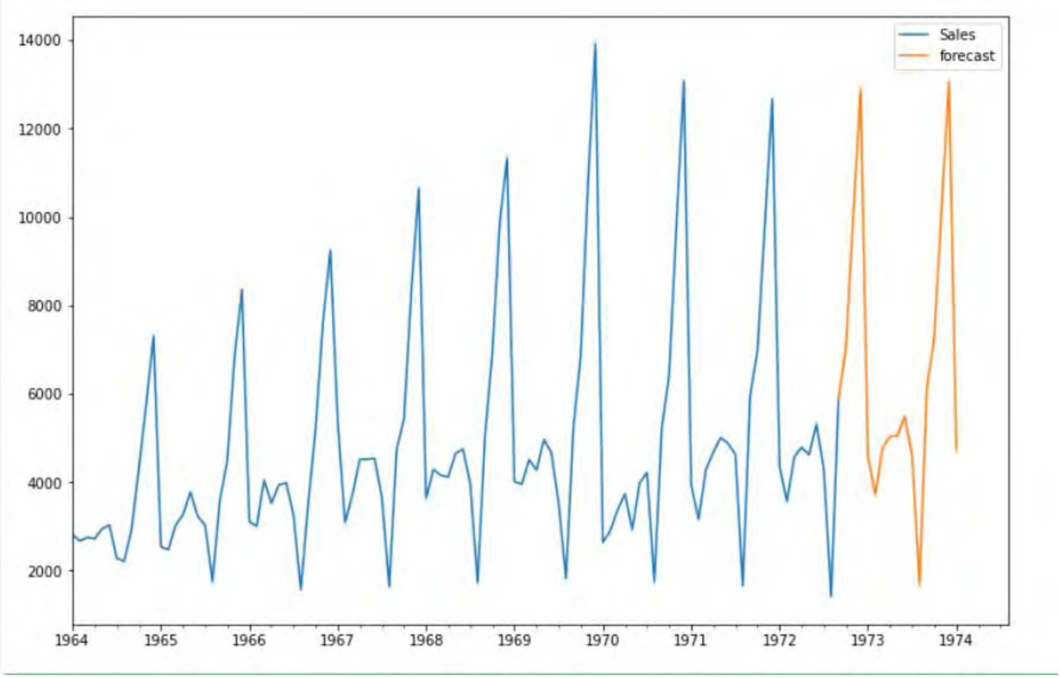
- Implementation - https://colab.research.google.com/drive/11DWBQQxlYFbAm5cnI3GNMnbGQwAsG7nZ#scrollTo=9STjl16i20eE

Moving Average (MA)

- Present value of the time series depends on the linear combination of the past white noise error.
- Denoted by the letter "q" which represents the order of the moving average model
- Current value of the time series will depend on the past q error terms.
- **Notation**: MA(q), where q is the number of lagged forecast errors.
- **Assumes stationarity.**
- Autocorrelation(ACF) plot helps determine optimal q.
- **Use case**
  - Effective when recent forecast errors influence current values.
- **Example**
  - Predicting tomorrow’s temperature using today's and yesterday's forecast errors.

ARIMA- Autoregressive Integrated Moving Averages

- Combines AR, I (differencing), and MA models for **non-stationary** time series.
- Specified by three order parameters: *(p, d, q).*
- **Notation**: ARIMA(p, d, q).
- **AR(*p*) Autoregression.**
  - Refers to relationship/ influence of past value on current value
- **I(*d*) Integration.**
  - Differencing of observations in order to make the time series stationary.
  - Involves the subtraction of the current values of a series with its previous values d number of times.
- **MA(*q*) Moving Average.**
  - Depicts the error of the model as a combination of previous error terms. The order *q* represents the number of terms to be included in the model.
- **ACF and PACF** used to determine p and q.
- **Use case.**
  - Forecasting stock prices, economic indicators, etc.



Implementation -
https://colab.research.google.com/drive/1ji14tHdh97EjbfVT5G_av4-omahsvKI4#scrollTo=wT40rotKE4uC

ARIMA - Pros and Cons

Pros

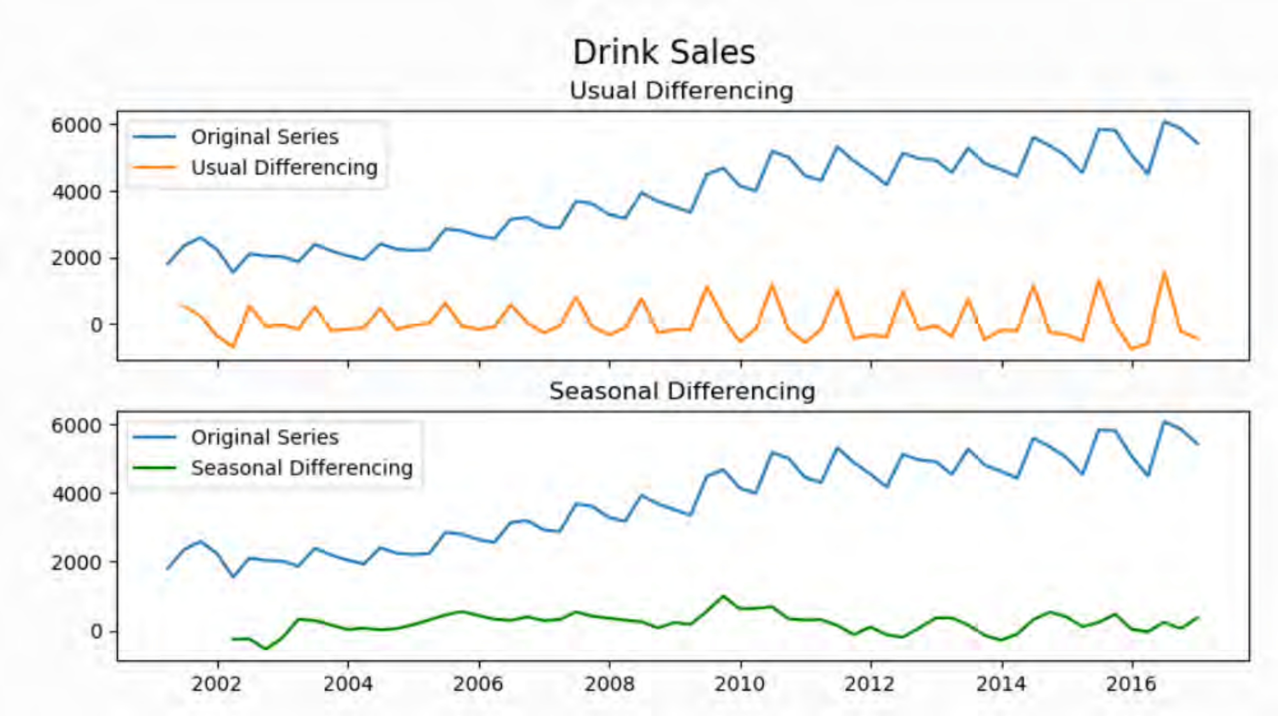
- Simple and Easy to Use
  - Straightforward to understand and implement.
- Interpretable
  - The coefficients show how past values influence future predictions.
- Good for Stationary Data
  - AR works well when the data has stable patterns over time.
- Fast and Efficient
  - They run quickly and are ideal for smaller datasets.

Cons

- Requires Stationarity
  - Most real-world data needs preprocessing
- Limited to Recent History
  - AR models can't capture long-term dependencies well.
- Sensitive to Noise
  - Random fluctuations can lead to inaccurate forecasts.
- Not Ideal for Long-Term Forecasting
  - *Performance drops over longer prediction horizons.*

SARIMA - Seasonal Autoregressive Integrated Moving Average

- It's an extension of the non-seasonal ARIMA model
- Designed to handle data with **seasonal patterns.**
- Captures both short-term and long-term dependencies within the data, making it a robust tool for forecasting.
- It combines the concepts of autoregressive (AR), integrated (I), and moving average (MA) models with seasonal components.
- **Notation**: ARIMA(p, d, q)(P, D, Q, s)
- **Seasonal Component:**
  - "S" represents seasonality, refers to repeating patterns in the data.
  - This could be daily, monthly, yearly, or any other regular interval.
  - Identifying and modelling the seasonal component is a key strength of SARIMA.
- **AR(*p*) Autoregression**
  - Refers to relationship/ influence of past value on current value
- **I(*d*) Integration**
  - Differencing of observations in order to make the time series stationary.
  - Involves the subtraction of the current values of a series with its previous values d number of times.



- **MA(q) Moving Average**
  - Depicts the error of the model as a combination of previous error terms. The order q represents the number of terms to be included in the model.
- Captures Trend + seasonality + short-term dependencies.
- Example: Modeling monthly retail sales with holiday effects every December.

SARIMA - Pros and Cons

Pros

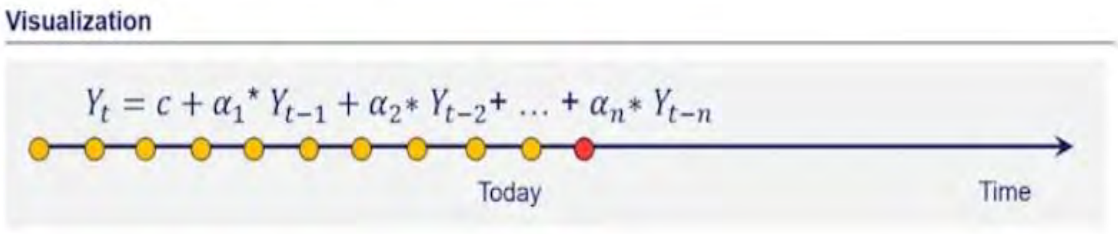
- Handles both trend and seasonality (seasonal differencing).
- More flexible for real-world cyclical data.
- Captures longer-term periodic patterns.
- Extends ARIMA without changing core logic — just adds seasonal terms.

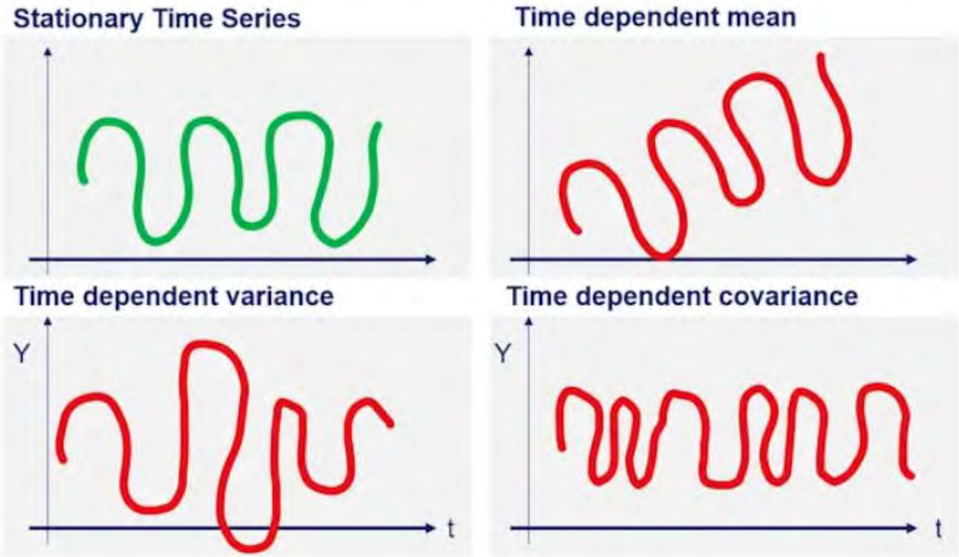
Cons

- More complex .
- Longer training time due to additional parameters.
- Can be overfit if seasonality is weak or not well understood.
- Requires enough seasonal cycles in the data to learn patterns.

ARIMAX

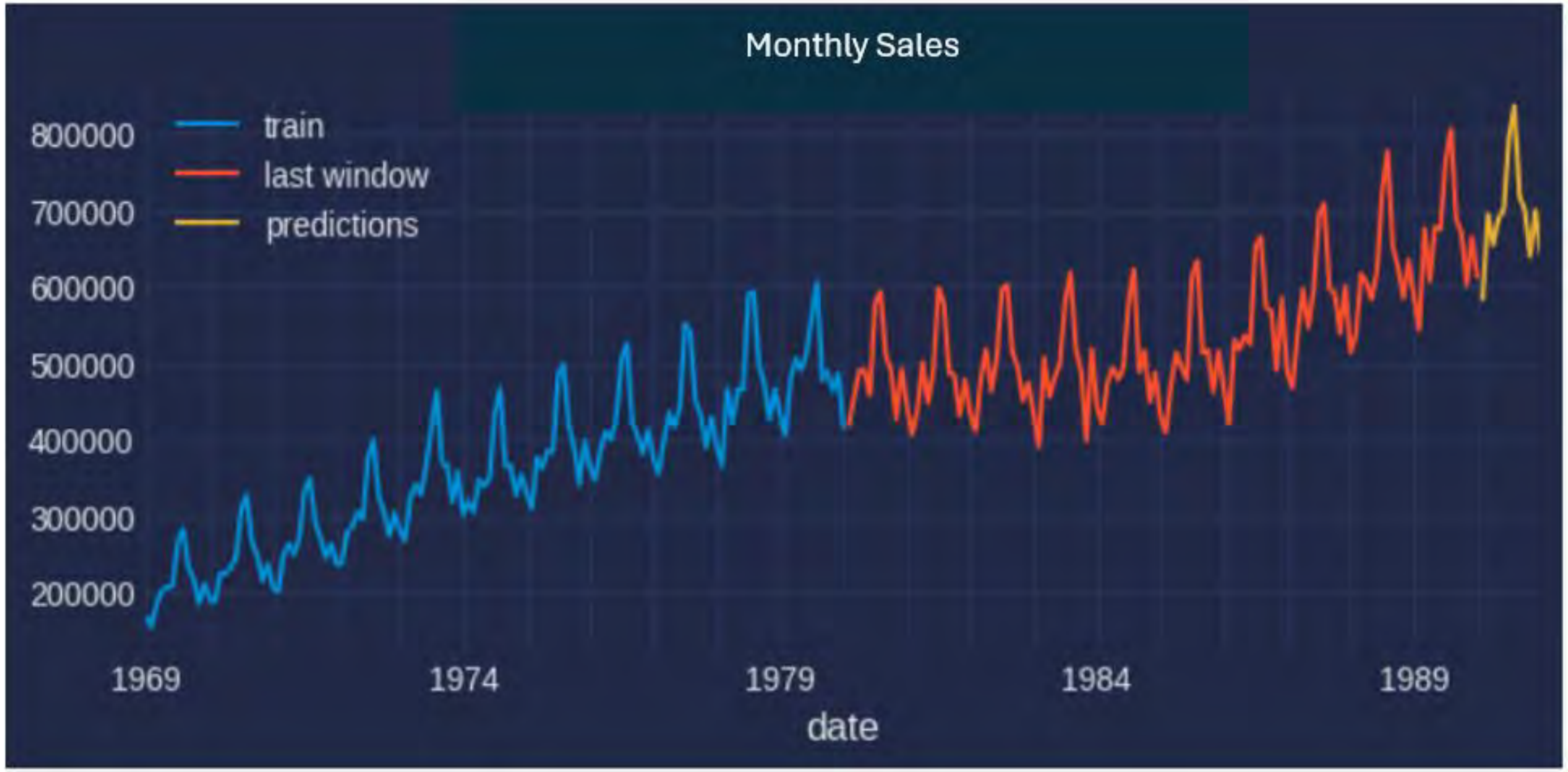
- AutoRegressive Integrated Moving Average with eXogenous variables.
- Four main parts of ARIMAX
  - AutoRegressive
    - *Understanding influence of past values on current value.*
  - Integrated
    - Achieve stationarity
  - Moving Average
    - Dependency between observation and residual error (past error)
  - Exogenous Variables.
    - External factors that influence target variable





ARIMAX

- When to use ARIMAX
  - More eXogenous variables
- Why to use eXogenous variable?
  - Improve Accuracy
  - Understanding Casual Relationship
  - Scenario Analysis
- Example of eXegoneous variable
  - Weather
  - Economic Indicator



ARIMAX - Pros and Cons

Pros

- Improved *Forecasting* Accuracy.
- Better Capture of External Factors
- Flexibility

Cons

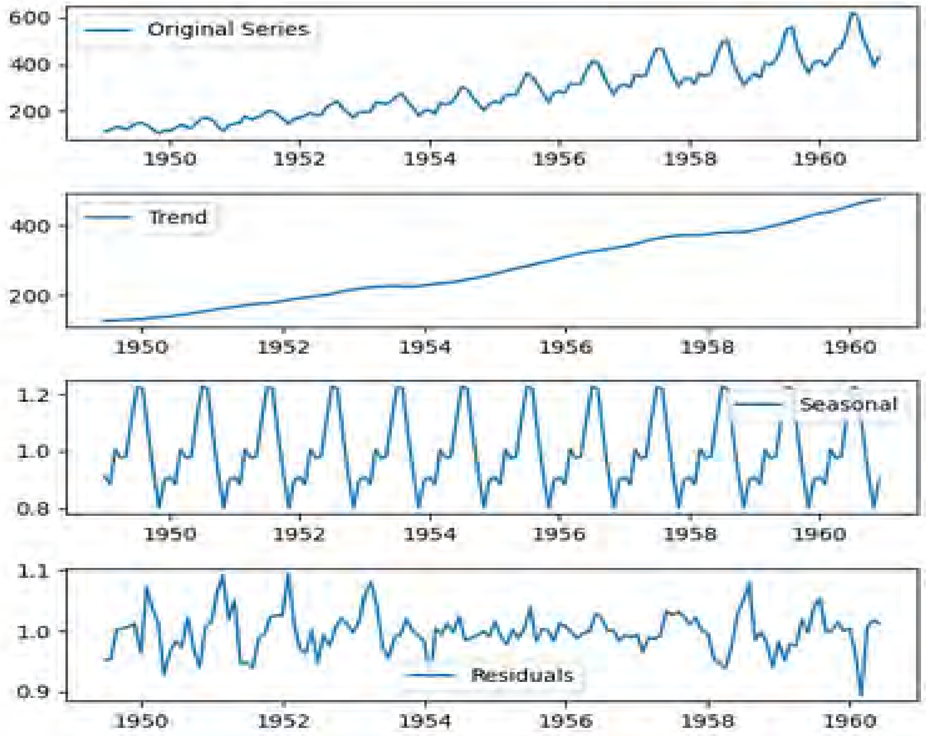
- *Variable Selection*
- *Increase Complexity*
- *Leads to overfitting*

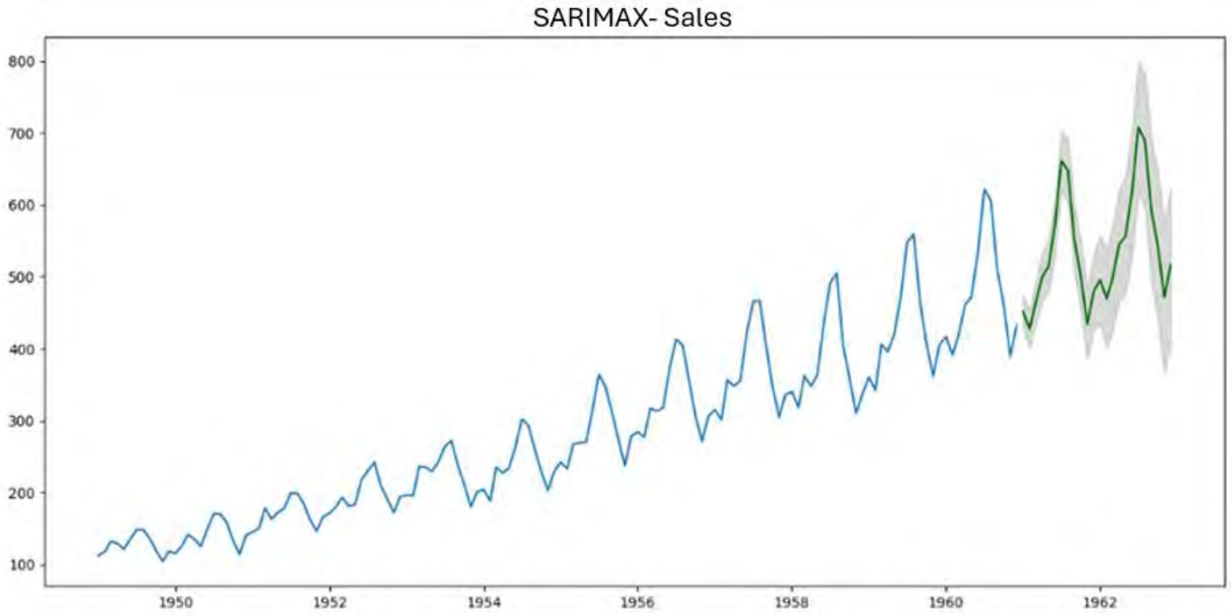
SARIMAX

- Seasonal Autoregressive Integrated Moving Average + exogenous variables.
- Uses both trends and seasonal variations including eXogenous variables.
- Components of SARIMAX
  - Seasonal component (S)
    - Capture Variation at regular intervals
  - Two major elements:
    - Covariates(X)
    - Covariate Component(Z)

Why seasonality?

- Capture recurring and predictable patterns
- Control external variables to make accurate prediction.





SARIMAX - Pros and Cons

Pros:

- The implementation is easy to set up and use
- The above structure is the same for other forecasting models
- Even though SARIMAX is an old methodology, it still gets super good results

Cons:

- Not good for long duration Time Series
- Difficult to deal with multi-colinearity
- It is not ideal for more than one seasonal data

Key Differences and comparison Table

Feature / Model	ARIMA	SARIMA	ARIMAX	SARIMAX
Full Form	AutoRegressive Integrated Moving Average	Seasonal ARIMA	ARIMA with eXogenous variables	Seasonal ARIMA with eXogenous variables
Trend Handling	✅ Yes (via differencing)	✅ Yes (via differencing)	✅ Yes	✅ Yes
Seasonality Handling	❌ No	✅ Yes (via seasonal differencing)	❌ No	✅ Yes
Exogenous Variables	❌ Not supported	❌ Not supported	✅ Supported	✅ Supported
Parameters	(p, d, q)	(p, d, q)(P, D, Q, s)	(p, d, q) + exog	(p, d, q)(P, D, Q, s) + exog
Use Case	Non-seasonal time series	Seasonal time series	Time series influenced by other variables	Seasonal + influenced by other variables
Complexity	⚪ Medium	🔺 High	🔺 High	🔺 🔺 Very High
Examples	Stock prices, short-term trends	Monthly sales, weather patterns	Ad sales affected by promotions	Monthly sales affected by holidays, weather

Tools for Time Series Analysis

Python-Based Tools

- **statsmodels**
  - Traditional statistical models (ARIMA, SARIMA, ARIMAX, SARIMAX etc.) Good for interpretability & hypothesis testing
  - Key functions: ARIMA, SARIMAX, seasonal_decompose
- **pmdarima**
  - Simplifies model selection (auto ARIMA) Automatically selects best (p,d,q) parameters
  - Great for beginners & fast prototyping
- scikit-learn
  - General ML framework; not focused on time series
  - Useful for time series with feature engineering
- TensorFlow / PyTorch
  - Deep learning for time series (RNN, LSTM, GRU)
  - Powerful for complex sequences & nonlinear patterns
  - **Visualization**
    - Matplotlib, Seaborn for Python

R-Based Tools

Forecast

Includes ARIMA, ETS, Holt-Winters

Easy to use for traditional time series models

Tseries / TSA

Statistical tests: ADF, KPSS

Classical decomposition & diagnostics

Challenges in Time Series Analysis

- **Data Quality**
  - Handling missing values and outliers.
- **Model Selection**
  - Choosing the right model for the data.
- **Computational Complexity**
  - Efficiently processing large datasets
- Overfitting and Underfitting
- Handling External (Exogenous) Variable

Conclusion

- Analyzes data over time to uncover trends, patterns, and seasonality
- Essential for forecasting future values, detecting anomalies and decision making across various industries

Key benefits -

- Improved forecasting accuracy
- Early anomaly detection
- Better resource planning and optimization

Widely used in -

- Finance ,Retail, Healthcare , Energy ,Web traffic, IoT, and more

Classical Model -

- ARIMA,SARIMA, ARIMAX,SARIMAX

Supervise ML Model -

- Linear Regression. Random Forest, XGBoost, SVR

Future Trends -

- Growth of real-time analytics and AutoML for time series
- Hybrid models combining statistical + ML approaches

Relevance -

- Used in LSTM, RNN, Transformers for predictive modeling
- Core to behavior modeling and dynamic systems

References

https://www.geeksforgeeks.org/time-series-analysis-and-forecasting/

https://www.geeksforgeeks.org/sarima-seasonal-autoregressive-integrated-moving-average/

https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/

https://ming-zhao.github.io/Business-Analytics/html/docs/time_series/sarima.html

https://www.analyticsvidhya.com/blog/2020/10/how-to-create-an-arima-model-for-time-series-forecasting-in-python/

https://www.analyticsvidhya.com/blog/2022/05/a-comprehensive-guide-to-time-series-analysis-and-forecasting/

Thankyou