Time Series Forecasting Using AR, MA, ARMA, and ARIMA Models in Python

Objective

To analyze and forecast time-dependent data using Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA) models.

Experiment Question

using the given dataset of daily temperature observations, develop AR, MA, ARMA, and ARIMA models to predict the next 7 days' temperature. Compare model accuracy using AIC and RMSE predict the next 7 days'. Compare model using AIC and RMSE

Dataset Description

Dataset Name:

daily_temperature.csv

	Description	**Data Type**	**Range / Example Values**	**Role**
Date	Observation date (daily interval)	DateTime	2023-01-01 → 2023-12-31	Independent variable (index)
Temperature	Average temperature (°C)	Float	10.0 to 45.0	Dependent variable (target)
Humidity (optional)	Relative humidity (%)	Float	30.0 to 95.0	Optional covariate

WindSpeed (optional)	Wind speed (km/h)	Float	0.0 to 25.0	Optional covariate

Records: 365 (one year of daily data)

Frequency: Daily (D)

Attribute Range Limits (for synthetic data generation)

Attribute	Minimum	Maximum	Distribution Type
Temperature	10.0	45.0	Seasonal + random noise
Humidity	30.0	95.0	Uniform random
WindSpeed	0.0	25.0	Normal (μ=12, σ=5)

Tasks to Perform

Step	Task	Expected Outcome
1	Import dataset and visualize the time series	Identify trend and seasonality
2	Perform stationarity test (ADF test)	Determine if differencing is required

3	Fit AR(p) model	Tune lag order p using AIC
4	Fit MA(q) model	Tune moving average order q
5	Fit ARMA(p,q) model	Compare AIC values
6	Fit ARIMA(p,d,q) model	Include differencing (d) to handle non-stationarity
7	Forecast next 7 days	Generate predictions
8	Evaluate model accuracy	Compare RMSE, AIC, and residual plots

Performance Metrics

Metric	Formula	Interpretation
AIC	Akaike Information Criterion	Lower is better
RMSE	\sqrt{\frac{1}{n}\sum(y_t - \hat{y_t})^2}	Lower means better fit
MAPE	$begin:math:text$\frac{1}{n}\sum	\frac{y_t - \hat{y_t}}{y_t}

Expected Learning Outcome

Understand time series stationarity and model selection.

Build and tune AR, MA, ARMA, and ARIMA models in Python.

Compare forecast performance using multiple metrics