Pedagogical Report

Time Series Forecasting with ARIMA in Healthcare: Predicting Daily ED Patient Volumes

INFO 7390 – Advanced Data Science and Architecture

Author: Kanishk Tawde

Teaching Philosophy

Target Audience

The target learners for this instructional module are:

- Graduate students in Data Science, Information Systems, or related fields
- Individuals with foundational Python experience (Pandas, NumPy, Matplotlib)
- Students who understand basic statistics (mean, variance, correlation)
- Beginners to intermediate learners in time series analysis
- Healthcare analytics professionals transitioning into predictive modeling

Assumptions of prior knowledge:

- Ability to write and run Python code in Jupyter Notebook
- Understanding of regression analysis
- Familiarity with supervised learning concepts
- Basic statistical reasoning

However, no prior experience with ARIMA, stationarity, or time-series forecasting is assumed. The instructional materials teach these from the ground up.

Learning Objectives

By the end of this module, learners will be able to:

1. Define what a time series is and identify its components (trend, seasonality, noise).
2. Explain the importance of stationarity and evaluate it using rolling statistics and ADF tests.
3. Interpret ACF and PACF plots to select appropriate ARIMA parameters.
4. Fit ARIMA models using Python's statsmodels library.

5. Generate and interpret forecasts, including confidence intervals.
6. Evaluate model performance using MAE and RMSE.
7. Perform residual diagnostics to assess model adequacy.
8. Interpret forecasting results within a healthcare operational context, particularly for ED staffing and resource planning.

Pedagogical Approach and Rationale

This project follows a constructivist, inquiry-driven, and applied learning design, supported by:

Explain → Show → Try Instructional Model

1. Explain
    - The theoretical documentation introduces core concepts slowly and clearly, using diagrams, analogies, and healthcare examples.
    - This builds conceptual scaffolding before code implementation.
2. Show
    - A fully commented Jupyter notebook demonstrates the entire ARIMA workflow end-to-end.
    - Each cell includes pedagogical explanations, not just code, so learners see not only *how* to do something but *why* it is done.
3. Try
    - A student-facing starter template provides TODO sections that mirror the instructor steps.
    - This ensures active experimentation and reinforces deep learning.

Why This Approach Works

- Learners retain more when they apply theory immediately in code.
- A healthcare scenario provides real-world meaning, anchoring abstract math to actionable insight.
- Breaking the process into small, digestible steps supports cognitive load management.
- Multiple representations (plots, equations, interpretations) strengthen comprehension.

Concept Deep Dive

Technical / Mathematical Foundations

What is a Time Series?

A time series is a sequence of observations measured at consistent intervals over time. In healthcare:

Example	Frequency	Operational Value
ED visits	Daily	Staffing, triage readiness
ICU census	Hourly	Bed allocation
Operating room usage	Daily	Scheduling
Infection incidence	Weekly	Public health response

Time Series Components

1. Trend – long-term direction (e.g., increasing ED visits due to population growth).
2. Seasonality – periodic patterns (e.g., weekends vs weekdays; winter flu season).
3. Noise – unpredictable residual variation.

Mathematically, decomposition follows:

Additive model:

$$Y_t = T_t + S_t + R_t$$

Multiplicative model:

$$Y_t = T_t \cdot S_t \cdot R_t$$

Stationarity

A series is stationary if:

- Mean is constant
- Variance is constant
- Autocorrelation does not vary over time

Why stationarity matters:
ARIMA models rely on past behavior predicting future behavior. If the system's underlying distribution changes over time, the model becomes unstable.

We use:

- Visual checks (rolling mean & variance)
- ADF (Augmented Dickey–Fuller) statistical test
- Differencing to transform non-stationary series into stationary ones

Differencing formula:

$$Y_t' = Y_t - Y_{t-1}$$

ACF & PACF

- ACF (Autocorrelation Function): correlation of series with its lagged versions
  - Guides selection of q (MA component)
- PACF (Partial Autocorrelation Function): correlation controlling for intermediate lags
  - Guides selection of p (AR component)

Interpretation rules commonly used:

Observation	Implication
Sharp PACF cutoff at lag k	AR(k)
Sharp ACF cutoff at lag k	MA(k)
Slow decay in both	Mixed ARMA

ARIMA Model Structure

ARIMA(p, d, q) stands for:

1. AR(p) – Autoregressive

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

2. I(d) – Integrated (d-th differencing)

3. MA(q) – Moving Average

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

The combined ARIMA model is flexible, interpretable, and suitable for operational forecasting tasks in healthcare where interpretability is key.

Connection to INFO 7390 Course Themes

GIGO: Garbage In, Garbage Out

- Time series forecasting is extremely sensitive to missing values, outliers, and irregular sampling.
- Preprocessing and imputation steps ensure high-quality input.

Botspeak

- The Explain → Show → Try instructional flow simulates structured AI-assisted reasoning.
- Code comments model “good inner monologue” or “traceable chain-of-thought” for students.

Computational Skepticism

- Residual diagnostics encourage learners to question model adequacy rather than accepting outputs blindly.
- Forecast uncertainty bands reinforce humility in predictive modeling.

Relationship to Real-World Analytics Workflows

The ARIMA workflow in this project mirrors real hospital analytics:

Workflow Step	Real Hospital Use Case
Exploratory analysis	Historical ED load analysis
Stationarity testing	Assessing data predictability
Model fitting	Forecasting 7-day staffing needs
Forecast evaluation	Comparing model to operational tolerances
Deployment	Integrating daily forecast into hospital dashboards

This reinforces learner readiness for industry roles in:

- Healthcare operations
- Hospital command centers
- Predictive analytics teams

Implementation Analysis

Architecture & Design Decisions

Why Jupyter Notebook?

- Interactive execution supports teaching
- Immediate visualization encourages exploration
- Markdown cells add theoretical context alongside code

Workflow Architecture

1. Data loading
2. Visualization
3. Stationarity testing
4. Differencing
5. ACF/PACF diagnostics
6. ARIMA model fitting
7. Forecast generation
8. Evaluation
9. Residual diagnostics

This modular pipeline allows learners to understand the dependencies between steps.

Tools & Libraries Chosen

Library	Purpose
pandas	Time series manipulation
numpy	Numerical operations
matplotlib / seaborn	Visualizations
statsmodels	ARIMA, ADF tests, ACF/PACF
scikit-learn	Error metrics
jupyter	Interactive teaching environment

Why statsmodels instead of pmdarima or Prophet?

- ARIMA in statsmodels is fully transparent and mathematically explicit
- Students learn model structures and parameters instead of black-box automation
- Interpretability matters in healthcare forecasting

Performance Considerations

- ARIMA works well for short-term forecasts (1–14 days out)
- For longer horizons or complex seasonality, SARIMA or Prophet might be required
- Differencing lowers computational complexity compared to seasonal models
- Synthetic dataset avoids privacy concerns, improves reproducibility

---

Edge Cases & Limitations

Limitation	Explanation
ARIMA assumes linear relationships	Non-linear surges may require LSTMs
ARIMA handles seasonality poorly unless SARIMA	Weekly patterns may not be fully captured
Sensitive to outliers	ED spikes during major events degrade accuracy

Limitation	Explanation
Requires stationarity	Some hospital data exhibits structural breaks

---

Assessment & Effectiveness

Evaluating Student Understanding

Assessment strategy includes:

- Embedded exercises in documentation (“Try it yourself” prompts)
- Starter template with graded difficulty TODO tasks
- Comparison between student forecasts & benchmark ARIMA(1,1,1) model
- Reflection questions focused on healthcare interpretation

Students also submit:

- An updated notebook
- Answers to conceptual questions
- A short forecast interpretation memo

Common Challenges Students Face

Challenge	How Material Addresses It
Understanding stationarity	Visual + statistical + intuitive explanations
Confusing ACF vs PACF	Side-by-side plots + interpretation guide
ARIMA parameter selection	Heuristic rules + experimentation
Residual diagnostics	Clear white-noise explanation
Healthcare interpretation	Realistic ED examples, operational insights

---

Supporting Multiple Learning Styles

Learning Style	Support Provided
Visual	Plots, diagrams, color-coded ACF/PACF
Analytical	Mathematical notation + stats tests
Hands-on	Jupyter coding, starter templates
Reflective	Interpretation prompts
Applied / practical	Operational healthcare use cases

Future Improvements & Extensions

Possible extensions include:

- Introducing SARIMA for weekly seasonality
- Comparing ARIMA to LSTM neural networks
- Adding outlier detection for major hospital events
- Incorporating exogenous variables (ARIMAX):
    - Weather (temperature, snowfall)
    - Local flu incidence
    - Holidays

Each extension would elevate the project toward research-grade forecasting.