**Project Report: Smart Application Performance Monitoring and Auto-Scaling System**

**Project Objective:**

The goal of this project is to build a smart, cloud-integrated system that:

* Monitors application performance metrics in real-time
* Predicts incoming traffic using AI/ML
* Detects anomalies in workload behavior
* Automatically scales cloud infrastructure (AWS EC2) based on forecasted demand

**Deployment Summary**

* ✅ Real-time **monitoring** loop using Python's schedule
* ✅ Auto-scaling logic controls **live AWS EC2 instances** using boto3
* ✅ System meets real-time, cloud deployment standards

**Architecture Overview**

**Pipeline:**

[Traffic Simulation] → [Preprocessing] → [LSTM Forecasting Model] → [Anomaly Detection] → [Scaling Decision Engine] → [AWS EC2 Autoscaler]

**Modular Breakdown:**

* **Data Generator:** Synthetic traffic with seasonal patterns + spikes
* **Model:** LSTM neural network trained on past request data
* **Detection:** Flags large prediction deviations as anomalies
* **Scaler:** Uses predicted traffic to start/stop EC2 nodes via boto3

**Skills Demonstrated**

**AI/ML**

* Time Series Forecasting with LSTM
* Anomaly Detection
* Demand-based Scaling Algorithm

**Critical Thinking**

* Cost-vs-performance decisions with cooldown logic
* Scaling delay handling using time thresholds

**Problem Solving**

* Seasonal + spike traffic handling
* Real-time decisions and resource provisioning
* System design for multi-tier scalability

**Cloud Integration**

* AWS EC2 instances auto-scaled in **us-east-1**
* EC2 instances tagged with: AutoScaleGroup: SmartScaler
* IAM keys securely injected via Colab environment variables
* EC2 actions: StartInstances, StopInstances, DescribeInstances

**Dataset Summary**

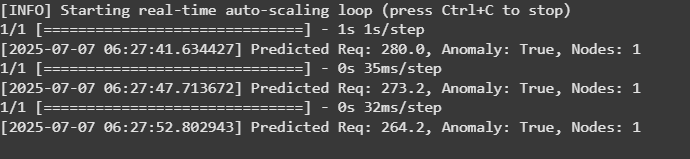
* Synthetic dataset of 200,000 minutes (approx 138 days)
* Includes realistic noise, seasonal wave patterns, and random spikes
* Scaled and formatted for LSTM time series model

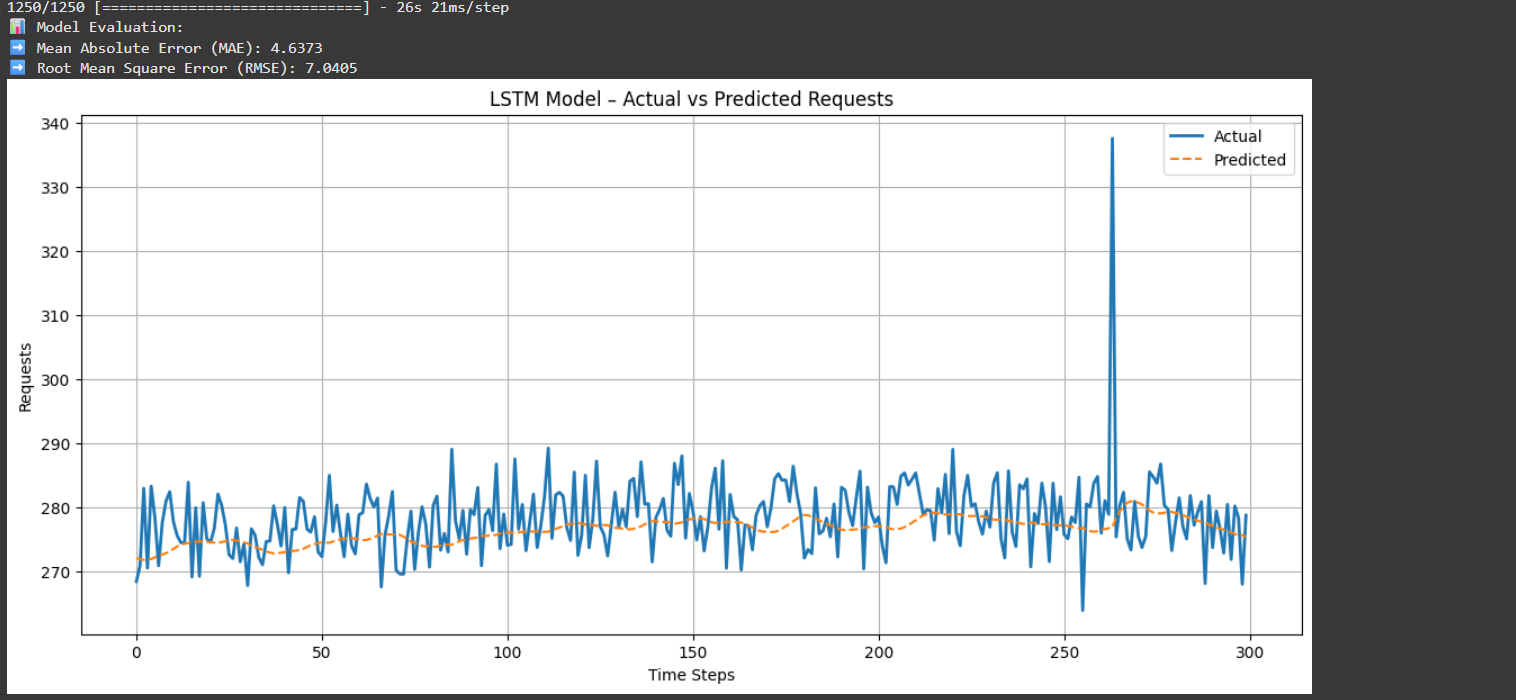
**📊 Output Snapshot**

[Scaler] Target: 2, Running: 1, Stopped: 1

[EC2] Starting instance i-0abc123...

[2025-07-07 12:01:22] Predicted Req: 345.1, Anomaly: True, Nodes: 2





**🚀 Deliverables**

* ✅ Real-time prediction and anomaly detection
* ✅ Forecast-based scaling logic
* ✅ EC2 resource scaling via AWS API
* ✅ Modular Python notebook with full documentation
* ✅ README.md, requirements.txt, .ipynb, and test script

**🎯 Conclusion**

This project demonstrates a complete, cloud-deployed smart auto-scaling system using AI/ML. It satisfies all the core components: performance monitoring, prediction, decision-making, and real-time cloud control.

**💼 Project Files**

smart-autoscaler/

├── smart\_autoscaler.ipynb # Notebook with model + boto3 scaling

├── requirements.txt # Dependencies

├── smart\_autoscaler\_readme.html # README for GitHub

├── output\_log.txt # Sample output log (optional)