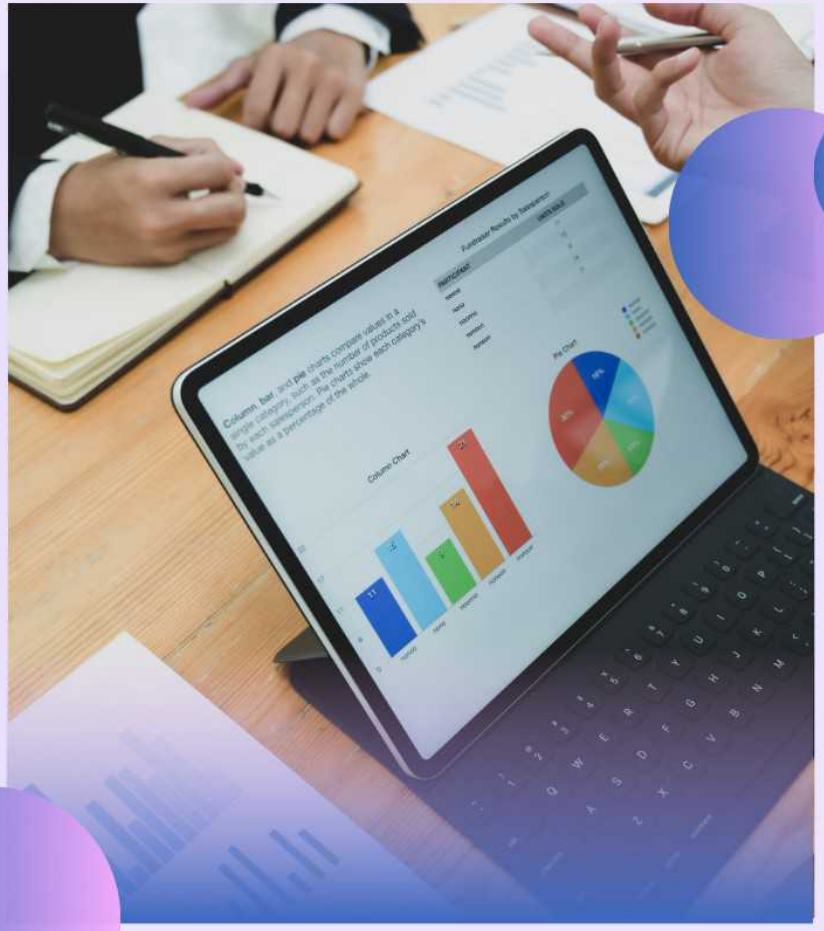
Health Metrics Forecasting Service

Problem Statement

- Healthcare providers need early predictions of health trends.
- Manual tracking of metrics → time-consuming & reactive.
- Lack of accessible API-driven forecasting tools for health data.
- Aim → Build a simple, API-first forecasting service for health data.



Objectives

- **Develop a FastAPI-based service for forecasting.**
- **Provide REST endpoints for:**
  - Health check (/health)
  - Forecasting (/forecast)
- **Enable integration with frontends or apps via API**
- **Make it accessible globally using Ngrok.**

Libraries & Tools Used

- FastAPI → to build RESTful API endpoints (/health, /forecast)
- Requests → to test API endpoints from client side
- Ngrok → to expose local FastAPI server to the internet
- Uvicorn → ASGI server to run the FastAPI app

(Optional, if you actually imported/used them in notebook)

- Pydantic → for request/response validation

Target Audience

🏥 Healthcare Analysts – quick forecast reports.

👩‍⚕️ Doctors & Hospitals – anticipate patient surges.

📊 Data Scientists – integrate forecasting API into pipelines.

👨‍💻 Developers & Students – learning API-based health projects

API Endpoints

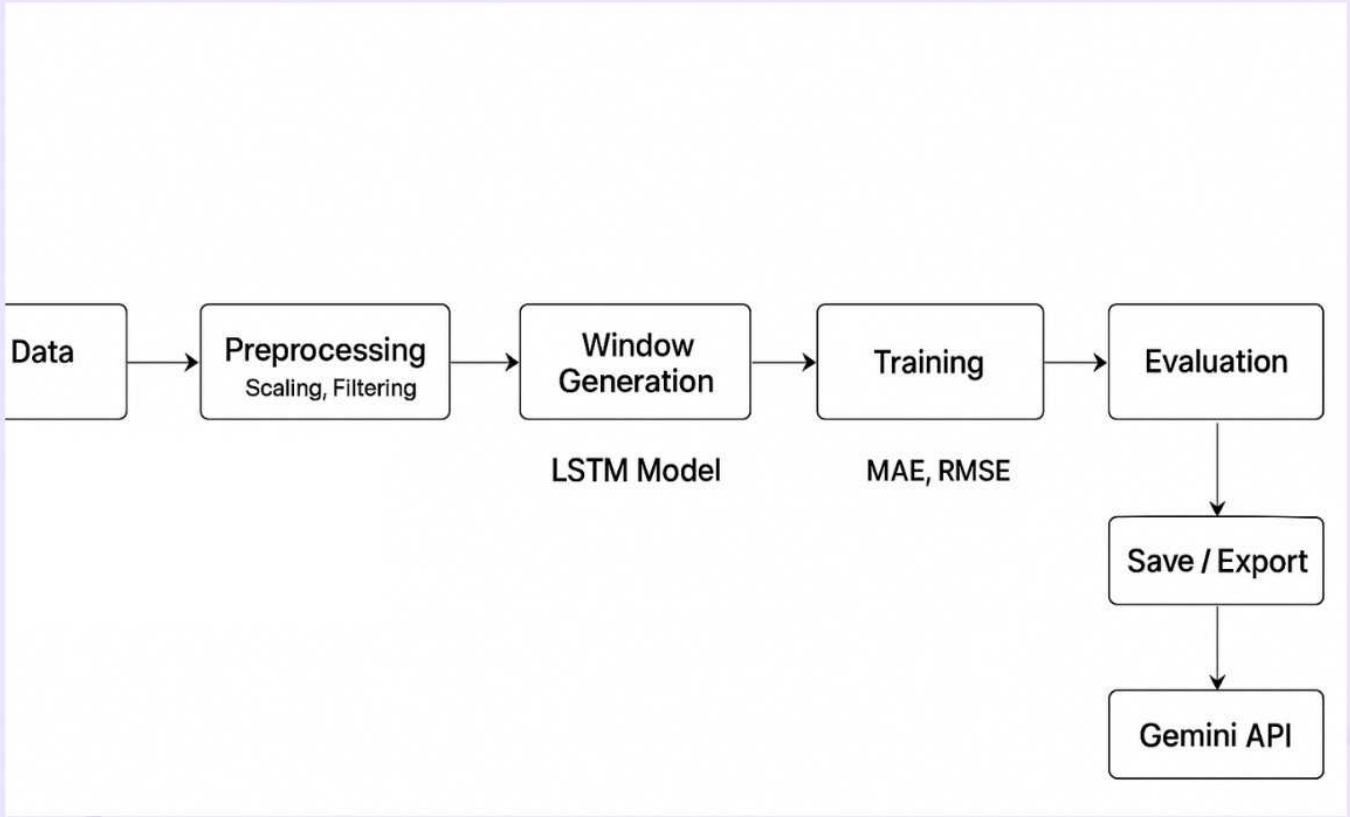
GET /health:

Returns system status ("Service is up and running")

POST /forecast:

Input: {"horizon": 5} (integer forecast horizon)
Output: Simulated forecast (JSON list of future values)

Architecture & Dataflow



Benefits & Impact

⚡ **Lightweight & Fast (built on FastAPI).**

🌍 **Accessible anywhere with Ngrok.**

🔗 **Easy integration into mobile/web apps.**

📈 **Helps proactive healthcare decisions (resource planning, patient care).**

🔒 **Secure API structure (only exposed endpoints).**

Future Evolution

- 📌 Advanced Models – integrate ML/DL forecasting (Prophet, LSTMs).
- 📌 Database Integration – store patient/metric history.
- 📌 Visualization Dashboard – real-time graphs for doctors.
- 📌 Scalability – deploy on cloud (AWS/GCP).
- 📌 Security Enhancements – authentication & role-based access.

Thank You.