**RUNBOOK\_Guardian\_Alerts**

**1) Purpose**

This runbook explains how to execute the **Guardian Alerts** notebook end-to-end to generate patient risk alerts (Low/Medium/High) with reasons, using:

* a **sequence anomaly model** (LSTM autoencoder; IsolationForest fallback),
* a **behavioural anomaly classifier** (Random Forest or MLP),
* a **clinically anchored vitals overlay** (SpO₂, temperature °C, blood pressure, activity, meals skipped).

The final output is artifacts/alerts.csv with:  
user\_id, timestamp, anom\_score, clf\_prob, risk\_level, reason.

**2) What you need (once per machine)**

* **Python 3.9+**
* Packages:
  + Required: pandas, numpy, scikit-learn, joblib, matplotlib
  + Optional (enables LSTM): torch
* **Jupyter Notebook** (or JupyterLab)

Install with pip:

pip install pandas numpy scikit-learn joblib matplotlib

# Optional, for LSTM:

pip install torch

# Jupyter (if not already installed):

pip install notebook

(Conda users can conda install pandas numpy scikit-learn matplotlib and conda install pytorch -c pytorch.

**3) Folder setup (keep it simple)**

Place the notebook and dataset side-by-side in the **same folder** (e.g., Alerts/):

Alerts/

├─ Guardian\_Alerts.ipynb # the notebook you’ll run

├─ New AI spreadsheet - Sheet1.csv # the dataset (CSV)

└─ (auto-created after running)

└─ artifacts/

├─ alerts.csv

├─ scaler.pkl

├─ lstm.pt OR iforest.pkl

├─ clf.pkl

└─ thresholds.json

The notebook **auto-detects** the dataset in the current folder (prefers New AI spreadsheet - Sheet1.csv). If your CSV has a different name, keep it in the same folder — auto-discovery will still find it.

**4) How to run**

1. Open a terminal in the Alerts/ folder and launch Jupyter:

jupyter notebook

1. Open **Guardian\_Alerts.ipynb**.
2. Run **all cells** top → bottom (Kernel → Restart & Run All is fine).

That’s it — the pipeline reads the entire dataset, trains the models, applies the alert logic, saves artifacts, and shows visualizations.

**5) What you should see**

* Console prints like:
  + “Detected dataset: …”
  + “PyTorch available: True/False”
  + LSTM training progress (if PyTorch is installed)
  + Classifier evaluation summary (if labels are weak-labeled or provided)
  + “Saved alerts to: artifacts/alerts.csv (rows=####)”
* Plots:
  + **Anomaly score distribution** with p80/p95 markers
  + **anom\_score vs clf\_prob** (scatter) coloured by final risk
  + **Risk counts** (bar chart)
  + **Example patient timeline** (key features + risk overlay)

**6) Outputs (where to look)**

* **Primary file:** artifacts/alerts.csv with columns  
  user\_id, timestamp, anom\_score, clf\_prob, risk\_level, reason
* **Models & metadata:**
  + scaler.pkl — feature scaler
  + lstm.pt (if LSTM used) **or** iforest.pkl (fallback)
  + clf.pkl — behavioural anomaly classifier
  + thresholds.json — calibrated anomaly thresholds (p80, p95) and bounds

Quick sanity checks in a new cell:

import pandas as pd

alerts = pd.read\_csv("artifacts/alerts.csv")

alerts["risk\_level"].value\_counts()

alerts.sample(5)

**7) Configuration knobs (tune without editing much)**

At the top of the notebook you can change:

* ENGINE = "lstm" or "iforest"  
  (If torch isn’t installed, the code auto-falls back to IsolationForest.)
* CLF = "rf" or "mlp"
* SEQ\_LEN = 14 (sequence window in days; shorter = more reactive, longer = smoother)
* Classifier cutoffs used in reasons: **0.65** (Medium), **0.85** (High)
* Vital thresholds (SpO₂, Temp °C, BP, exercise/day, meals skipped) are in the vital\_risk\_and\_reasons() helper — edit if your clinicians prefer different limits.

**8) How the alert logic works (so you can verify)**

* **Model risk** from anomaly + classifier:
  + High if recon\_error ≥ p95 **or** clf\_prob ≥ 0.85
  + Medium if recon\_error ≥ p80 **or** clf\_prob ≥ 0.65 (and not High)
  + Low otherwise
* **Vitals risk** from clinical thresholds (direction-aware):
  + SpO₂ (Low ≥95, Med 90–94, High <90), Temp °C (Med ≥38.0, High ≥39.4),  
    BP (Stage 1/2 ranges), Exercise/day (Low ≥20, Med 10–19, High <10), Meals skipped (Low 0–1, Med 2, High ≥3)
* **Final** risk\_level = **max(model\_risk, vital\_risk)**
* **reason** lists all triggers for **Medium/High** rows (model + vitals).

**9) Troubleshooting (fast fixes)**

* **“No CSV/Excel found in current directory”**  
  → Ensure the dataset is in the same folder as the notebook, or rename to include “sheet1” or “new ai spreadsheet”.
* **Timestamp parse errors**  
  → The code prefers observationStart (falls back to observationEnd). Make sure one of them exists and is a valid date/time.
* **Missing columns / KeyError**  
  → The loader maps your schema to canonical names. If you changed header names, update the rename\_map in the loader cell.
* **Everything shows as High risk**  
  → This is usually thresholding. The notebook calibrates model thresholds from your dataset (p80/p95). Re-run after confirming the dataset isn’t trivially small or all-zeros; adjust thresholds if needed.
* **No PyTorch**  
  → The notebook will automatically use IsolationForest. If you want LSTM, install torch.