Assignment 3: Integrated ISD Prototype (Stroke Risk Assistant)

1. System Overview
The system integrates perception, reasoning, action, and feedback for stroke-risk prediction using the stroke dataset from Assignment 2.

Perception:
- Module: StrokePerception (perception.py)
- Simulates tabular sensor input by sampling patient records from the dataset.
- Adds mild Gaussian noise to numeric attributes (age, avg_glucose_level, bmi) to emulate measurement uncertainty.

Reasoning:
- Module: StrokeReasoner (reasoning_integration.py)
- Reuses the rule-based stroke predictor from Assignment 2 (rules_python.py).
- Trains a RandomForestClassifier on the same dataset as the ML component.
- Combines rule-based and ML outputs; a patient is flagged high-risk if either component predicts stroke above a configurable probability threshold.

Action:
- Module: action_feedback.py
- If the final decision is high risk, the system triggers an alert to notify a clinician; otherwise, it logs the patient as low-risk.
- Actions are observable via console output and can be easily mapped to a GUI indicator in a future capstone.

Feedback:
- The loop evaluates predictions against ground-truth labels from the dataset.
- If too many false negatives are observed, the ML probability threshold is lowered slightly. This makes the system more conservative and improves safety.

2. Evaluation (Example Run)
- Number of simulated samples: 30–50 (configurable).
- Metrics reported:
  * Accuracy: fraction of correctly classified stroke vs. non-stroke cases.
  * Latency: average time per Perceive -> Reason -> Act cycle (in milliseconds).
  * Robustness: fraction of correct classifications under noisy input.
- The RandomForest model typically achieves good validation accuracy, while the rule-based component adds interpretability and a safety layer.

3. Reflection and Future Work
- Strengths:
  * Clear, modular Perceive -> Reason -> Act -> Feedback pipeline.
  * Combination of symbolic rules and ML improves both interpretability and predictive performance.
  * Feedback mechanism demonstrates how ISD can adapt to observed errors.
- Limitations:
  * Perception currently uses a static dataset rather than a live sensor feed.
  * Feedback only adjusts a single threshold; more advanced online learning (e.g., incremental retraining) could be explored.
  * The action module is console-based; a deployed system would require a UI and proper integration with clinical workflows.

This prototype is ready to be extended into the capstone by adding a simple web-based front end and more advanced feedback/learning strategies.