Intelligent Systems Design Report

Comparative Analysis of Stroke Prediction Models

Date: November 27, 2025

Subject: Comparison of Rule-Based, Classical ML, and Deep Learning Approaches for Stroke Prediction

1. Dataset Summary and Problem Framing

Dataset Overview The analysis utilized the healthcare-dataset-stroke-data.csv, a clinical dataset containing patient health records. The dataset includes demographic details (Age, Gender, Residence), physiological measurements (BMI, Average Glucose Level), and medical history (Hypertension, Heart Disease).

- **Input Features:** 11 Features (Mix of categorical and numerical).
- **Target Variable:** stroke (Binary: 1 for Stroke, 0 for No Stroke).
- **Data Characteristics:** The dataset contains missing values in the BMI column and is essentially **imbalanced**, meaning the number of negative cases (No Stroke) significantly outweighs the positive cases.

Problem Framing The objective is **Binary Classification**. We aim to predict the likelihood of a patient suffering a stroke based on their health profile. The core challenge is **Sensitivity (Recall)**. In medical diagnostics, the cost of a False Negative (missing a stroke) is catastrophic compared to a False Positive (unnecessary check-up). A model with 95% accuracy is useless if it achieves that by simply predicting "No Stroke" for everyone.

2. Overview of Rule-Based Logic (Symbolic AI)

The first approach implemented was a **Knowledge-Based System**. Instead of learning from data, this system utilized explicit "If-Then" rules derived from general medical heuristics.

Logic Implemented: The system applied a deterministic logic flow. A patient was flagged as "High Risk" (1) only if they met specific criteria; otherwise, they were defaulted to "Low Risk" (0).

- **Rule 1 (Age & Hypertension):** IF Age > 75 AND Hypertension = Yes → *Stroke*.
- **Rule 2 (Diabetic Risk):** IF Heart Disease = Yes AND Glucose > 200 → *Stroke*.
- **Rule 3 (Lifestyle):** IF BMI > 40 (Obesity Class III) AND Smoker → *Stroke*.
- **Rule 4 (Comorbidities):** IF Age > 60 AND Hypertension AND Heart Disease → *Stroke*.
- **Rule 5 (Safety):** IF Age < 30 AND No History → *No Stroke*.

Observed Behavior: The sample outputs show the rigidity of this approach.

Success: Correctly flagged row 0 (Age 67, Heart Disease, High Glucose).

Failure: Missed row 2 (Age 80, Heart Disease). Despite high age and heart issues, the patient didn't meet the strict threshold of Rule 1 (requires Hypertension) or Rule 2 (requires Glucose > 200), leading to a False Negative.

Critique of Logic: This approach represents "White Box" reasoning. The decision path is perfectly transparent. However, it is rigid. A 74-year-old with hypertension would be classified as "Safe" by Rule 1 simply because they missed the age cutoff by one year, demonstrating the brittleness of hard-coded boundaries.

3. Architectures and Results

A. Machine Learning Model (Random Forest)

Architecture:

- **Model:** Random Forest Classifier (Ensemble of Decision Trees).
- **Preprocessing:**
    - *Imputation:* Median filling for missing BMI.
    - *Encoding:* One-Hot Encoding for categorical variables (e.g., Work Type, Gender).
    - *Scaling:* StandardScaler applied to Age, BMI, and Glucose.
    - *Balancing:* class_weight='balanced' was used to penalize the model more for missing stroke cases.
- **Performance:**
    - **Accuracy:** 94%
    - **Precision:** 100%
    - **Recall: 2% (0.02)**

Analysis: The model fell into the **Accuracy Paradox**. Despite using class weights, the Random Forest overwhelmingly favored the majority class. It achieved high accuracy by predicting "No Stroke" for almost everyone. It is "safe" but clinically useless, as it only identified 2% of actual stroke victims.

Performance: The Random Forest demonstrated robust performance. By aggregating the votes of multiple decision trees, it captured non-linear relationships (e.g., the interaction between Age and BMI) better than linear models. It achieved high accuracy (~95%) but struggled slightly with Recall due to the extreme class imbalance, a common issue in medical datasets.

B. Deep Learning Model (Feed-Forward Neural Network)

Architecture:

- **Structure:** A Multi-Layer Perceptron (MLP) built with TensorFlow/Keras.
    - *Input Layer:* Matches processed feature shape.
    - *Hidden Layers:* Dense (64 neurons, ReLU) → Dropout (0.3) → Dense (32 neurons, ReLU).

    - *Output Layer:* Dense (1 neuron, Sigmoid activation).
- **Optimization:** Adam optimizer with Binary Cross-entropy loss.
- **Performance:**
    - **Accuracy:** 84.15%
    - **Precision:** 19.14%
    - **Recall: 50.00%**

Analysis: The Deep Learning model sacrificed overall accuracy to capture more positive cases. It successfully identified **half (50%)** of the stroke patients. However, the low precision (19%) means it generates many False Alarms—for every 5 patients it flags, only 1 actually has a stroke.

Performance: The Deep Learning model showed performance metrics very similar to the Random Forest. The loss curve converged quickly (within 10-15 epochs). However, the neural network required significantly more data preprocessing (normalization is strictly required for convergence) and was more sensitive to hyperparameter changes (learning rate, batch size).

4. Comparative Discussion

Accuracy vs. Interpretability

- **Rule-Based: Lowest Accuracy, Highest Interpretability.**
    - It is easy to explain *why* a decision was made ("Patient is over 75 and has hypertension"), which is valuable in clinical settings. However, it fails to capture complex, hidden patterns, resulting in poor predictive power.
- **Machine Learning (Random Forest): High Accuracy, Moderate Interpretability.**
    - While we cannot look at every single tree, Random Forests provide "Feature Importance" scores, allowing us to see that Age and Glucose are the biggest drivers of risk. This strikes a good balance.
- **Deep Learning: High Accuracy, Low Interpretability.**
    - The Neural Network acts as a "Black Box." It provides a probability output based on complex weight matrices that are difficult to map back to biological reasoning.

Effort to Design and Train

- **Rule-Based:** High *manual* effort. requires domain experts to define rules. Zero training time.
- **Machine Learning:** Low effort. Scikit-learn pipelines handle preprocessing and training in few lines of code. Fast training (seconds).
- **Deep Learning:** High effort. Requires defining architecture (layers, neurons), choosing activation functions, and tuning optimizers. Slower training time.

Robustness

- **Rule-Based: Brittle.** It fails completely on "edge cases" that don't fit the exact hard-coded rules.
- **Machine Learning: Robust.** The Random Forest is resistant to noise and overfitting due to its ensemble nature (averaging out errors).
- **Deep Learning: Variable.** Without techniques like Dropout (which we used), Neural Networks are prone to overfitting on small datasets, memorizing the training data rather than learning patterns.

5. Reflection: The Suitable Approach

Conclusion: Contrary to typical expectations where Random Forests usually win on tabular data, **the Deep Learning approach proved more suitable in this specific experiment.**

Why?

1. **The Recall Criticality:** A medical model with 2% Recall (Random Forest) is dangerous. It provides a false sense of security. The Deep Learning model, with 50% Recall, is a far better starting point for a screening tool.
2. **Learning Capabilities:** The Neural Network's continuous activation functions (Sigmoid/ReLU) likely allowed it to model a "risk gradient" better than the Random Forest's discrete splits, preventing it from completely ignoring the rare stroke cases.
3. **Trade-off Acceptance:** In this domain, we prefer the Deep Learning error profile (Low Precision, Moderate Recall) over the Random Forest profile (High Precision, Near-Zero Recall). It is better to check a healthy patient than to ignore a sick one.

Final Recommendation: While the Deep Learning model performed better here, the ideal solution would likely be a **Hybrid Approach**:

- Use the **Deep Learning model** to generate risk probabilities.
- Apply **Rule-Based post-processing** to filter out obvious False Positives from the DL output (e.g., "If DL predicts stroke but Age < 20, override to No").