**Overview**

This project is a **full-stack machine learning application** designed to predict the likelihood of a stroke based on patient health metrics. It consists of:

* A **Flask backend** serving a trained machine learning model via a REST API
* A **React.js frontend** providing an interactive form for user input
* A **data preprocessing and model training pipeline** built with scikit-learn

The system integrates **medical data analysis, predictive modeling, and web development** to deliver a user-friendly stroke risk assessment tool.

**Key Components**

**1. Backend (Flask API -**app.py**)**

* **REST API** built with Flask to handle prediction requests
* **CORS-enabled** for frontend communication
* **Endpoint (**/predict**)** that accepts JSON input, processes it with the trained model, and returns predictions
* **Error handling** for robust API responses
* Hosted on localhost:5000 for local development

**2. Machine Learning Pipeline (**training.py**)**

* **Data Loading & Preprocessing:**
  + Reads stroke dataset (healthcare-dataset-stroke-data.csv)
  + Drops irrelevant columns (id)
  + Separates **categorical** (e.g., hypertension, smoking\_status) and **numerical** (e.g., age, bmi) features
* **Feature Engineering:**
  + **OneHotEncoder** for categorical variables
  + **SimpleImputer** (median strategy) for missing values
  + **PowerTransformer (Yeo-Johnson)** for numerical feature normalization
* **Handling Class Imbalance:**
  + Uses **SMOTE (Synthetic Minority Oversampling Technique)** to balance the dataset
* **Model Training & Evaluation:**
  + **Linear Discriminant Analysis (LDA)** as the classifier
  + **Repeated Stratified K-Fold Cross-Validation (10 splits, 3 repeats)**
  + Evaluated using **ROC-AUC score**
  + **Mean ROC-AUC: 0.84** (indicating strong predictive performance)
* **Model Persistence:**
  + Saved as stroke\_prediction\_model.joblib for deployment

**3. Frontend (React.js -**App.js**)**

* **Interactive Form** collecting:
  + Demographic data (gender, age)
  + Medical history (hypertension, heart\_disease)
  + Lifestyle factors (smoking\_status, work\_type)
  + Health metrics (avg\_glucose\_level, bmi)
* **Dynamic UI** with animated background elements
* **API Integration:**
  + Submits form data to Flask backend
  + Displays prediction result via alert
* **Responsive Design** with CSS styling