Brain Stoke Prediction: Predicting Stoke Likeliness

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering (Information Technology)

By

Devansh Wadhwani

Roll No – 64

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY,

Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

April 2024

BRAIN STOKE PREDICTION

Name of student	Devansh Wadhwani
Class_Roll no	D15A_64
D.O.P	20/03/25
D.O.S	27/03/25
Sign and Grade	

Title : Brain Stoke Prediction

Project Description :

The **Brain Stroke Prediction** project is a machine learning-based web application designed to assess the likelihood of a person experiencing a stroke based on user-provided medical information. It utilizes a reliable stroke prediction dataset and employs classification algorithms such as **Decision Tree** or **K-Nearest Neighbors** to ensure accurate and efficient risk prediction.

The backend is developed using **Flask in Python**, integrating a pre-trained model built with **scikit-learn**. The system is supported by a clean and interactive frontend that allows users to input relevant health features—such as age, hypertension, heart disease status, glucose level, BMI, and smoking history—and receive instant risk assessments.

The application provides **real-time inference with minimal latency**, making it suitable for **educational demonstrations**, **clinical decision support**, and **preventive healthcare systems**. The architecture follows **modular design principles**, ensuring scalability, maintainability, and ease of integration with larger healthcare platforms. Furthermore, the prediction logic is exposed through **RESTful APIs**, enabling smooth and reliable communication between the frontend and backend systems.

Requirement gathering : Requirement gathering is a crucial phase in the development of the Brain Stroke Prediction system. It involves understanding the goals of the application, identifying the end-users, and specifying the functional and non-functional requirements of the system. Below are the components of the requirement gathering process:

1. Objective

To develop a machine learning-based web application that predicts the risk of brain stroke in individuals based on their medical and lifestyle data.

3. Functional Requirements

- Users should be able to input personal health data (e.g., age, gender, hypertension, heart disease, glucose level, BMI, smoking status).
- The system should validate the input for correctness.
- The trained machine learning model should process the input and predict the probability of stroke.
- The prediction result (e.g., "High Risk" or "Low Risk") should be displayed instantly.
- RESTful APIs should be implemented for frontend-backend communication.
- Admins should be able to update or retrain the model if needed.

5. Dataset Requirements

- Use a publicly available and trusted dataset (e.g., from Kaggle) containing fields such as:
    - Age
    - Gender
    - Hypertension (Yes/No)
    - Heart Disease (Yes/No)
    - Ever Married
    - Work Type
    - Residence Type (Urban/Rural)
    - Average Glucose Level
    - BMI
    - Smoking Status
    - Stroke (target variable)

6. Tools & Technologies

- **Frontend**: HTML, CSS, JavaScript (optional)
- **Backend**: Flask (Python)
- **Machine Learning**: scikit-learn, pandas, numpy
- **Model Serialization**: joblib or pickle
- **Version Control**: Git
- **Deployment (Optional)**: Heroku, AWS, or local server

System Requirements :

1. Hardware Requirements:

- **Processor:** Intel Core i5 / AMD Ryzen 5 or higher (dual-core, 2.0 GHz or faster)
- **RAM:** Minimum 8GB (16GB recommended)
- **Storage:** At least 1GB free space (256GB SSD recommended)
- **Network:** Stable internet connection (especially for MongoDB Atlas users)

2. Software Requirements:

- **Operating System:** Windows 10/11, macOS 10.15+, or Ubuntu 20.04+
- **Code Editor:** Visual Studio Code or compatible IDE
- **Version Control:** Git 2.25+
- **Python:** Version 3.8 or higher

Technologies Used :

Development	VS Code , Postman , Git
Frontend	HTML/CSS/Typescript(or Streamlit/Flask Templates)
Backend	Flask (Python 3.8+)
ML Model	Scikit-learn
Styling	CSS
APIs	RESTful Flask APIs

Setup Instructions :

1. **Python 3.8+:** To set up the Flower Prediction project, first install Python 3.8 or higher from the official Python website (python.org). During installation (especially on Windows), make sure to check the option "Add Python to PATH." After installation, verify it using python --version and pip --version in the terminal. For better environment management, you can create a virtual environment using python -m venv venv, and activate it with venv\Scripts\activate on Windows or source venv/bin/activate on macOS/Linux.
2. **Flask and Dependencies:** Once Python is set up, install the necessary packages by running pip install -r requirements.txt in your project directory. This will install Flask, scikit-learn, and other required libraries. Make sure the requirements.txt file is up to date with your project dependencies.
3. **ML Model Setup**: Ensure that your trained ML model (e.g., a .pkl file) is saved in the appropriate directory (e.g., a model/ folder). The Flask backend will load this model at runtime to perform predictions based on user inputs.

Backend Setup :

1. **Navigate to backend folder:**

cd project

2. **(Optional) Create a virtual environment:**

python -m venv venv

venv\Scripts\activate # For Windows

3. **Install dependencies:**

pip install -r requirements.txt

4. **Start the Flask server:**

cd api
python app.py

Backend will run at: http://localhost:5000

Frontend Setup

1. **Navigate to frontend folder:**

cd project

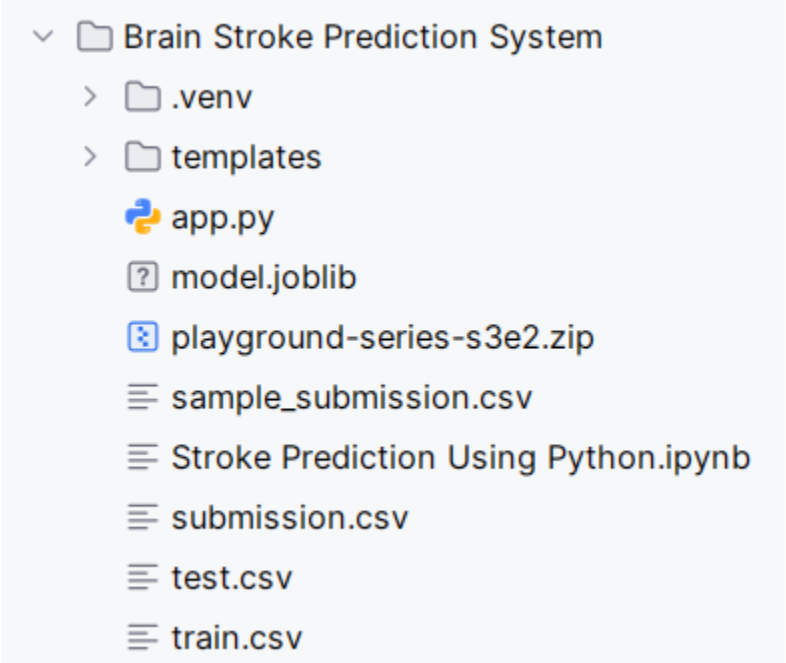
2. **Install dependencies:**

npm install

3. **Npm run dev**

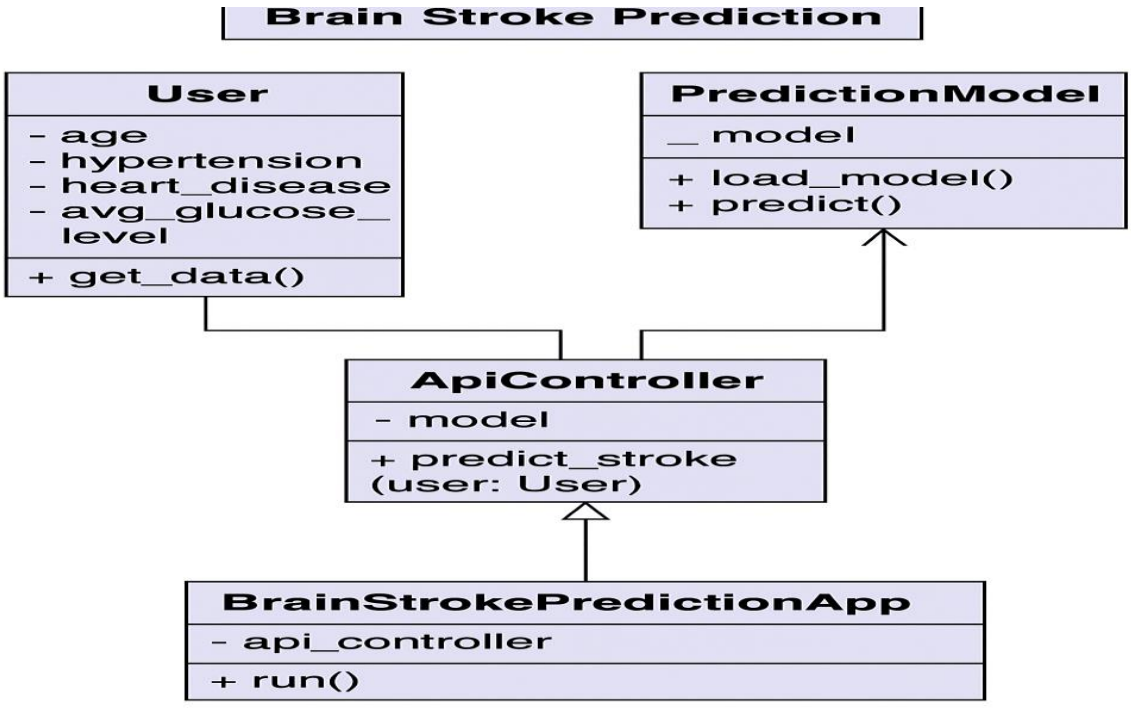
Frontend will run at: http://localhost:8081

Project Structure:



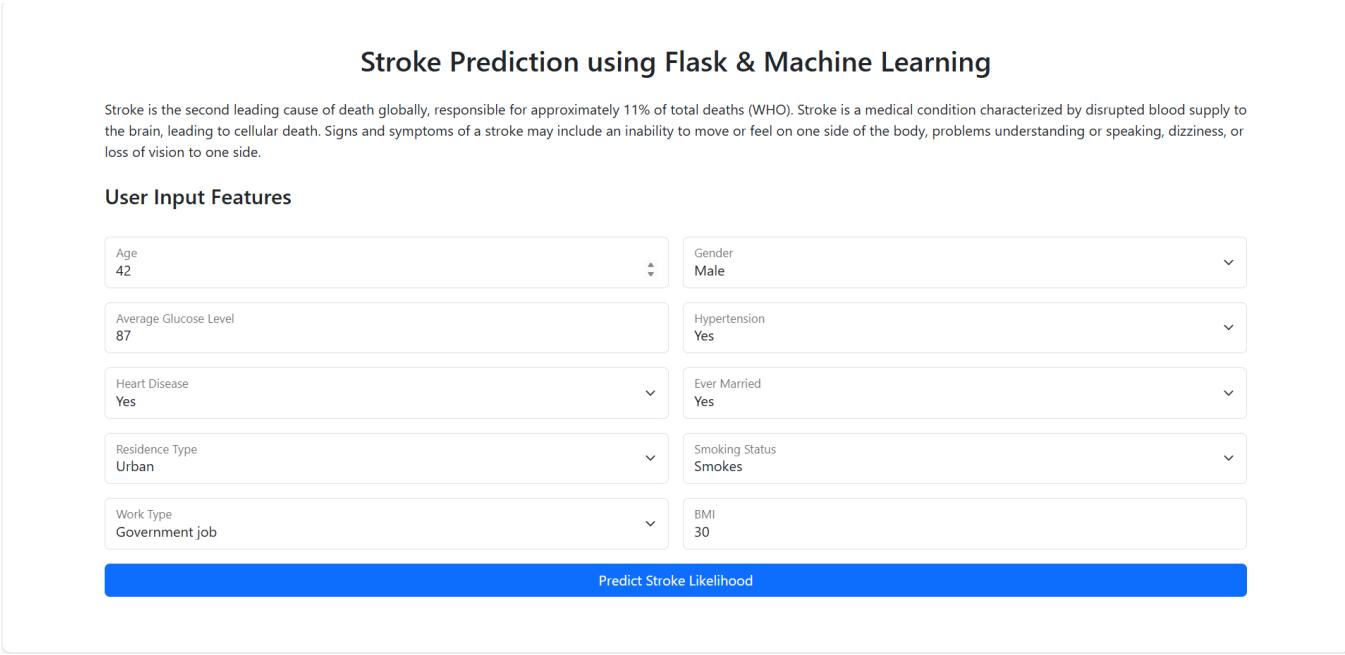
Architectural Diagrams :

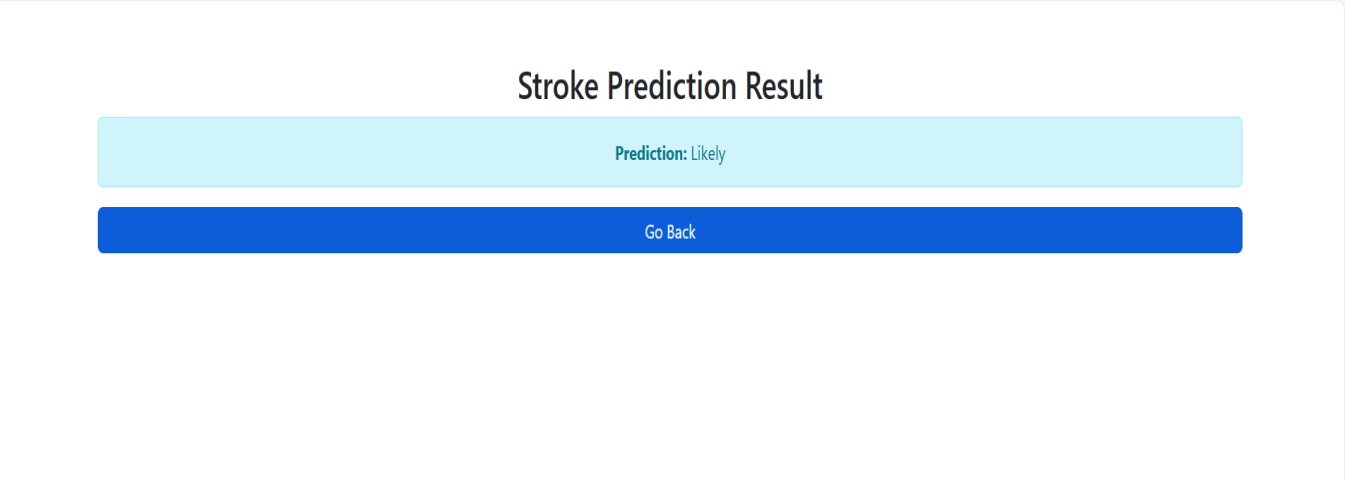
a) Class Based Diagram -



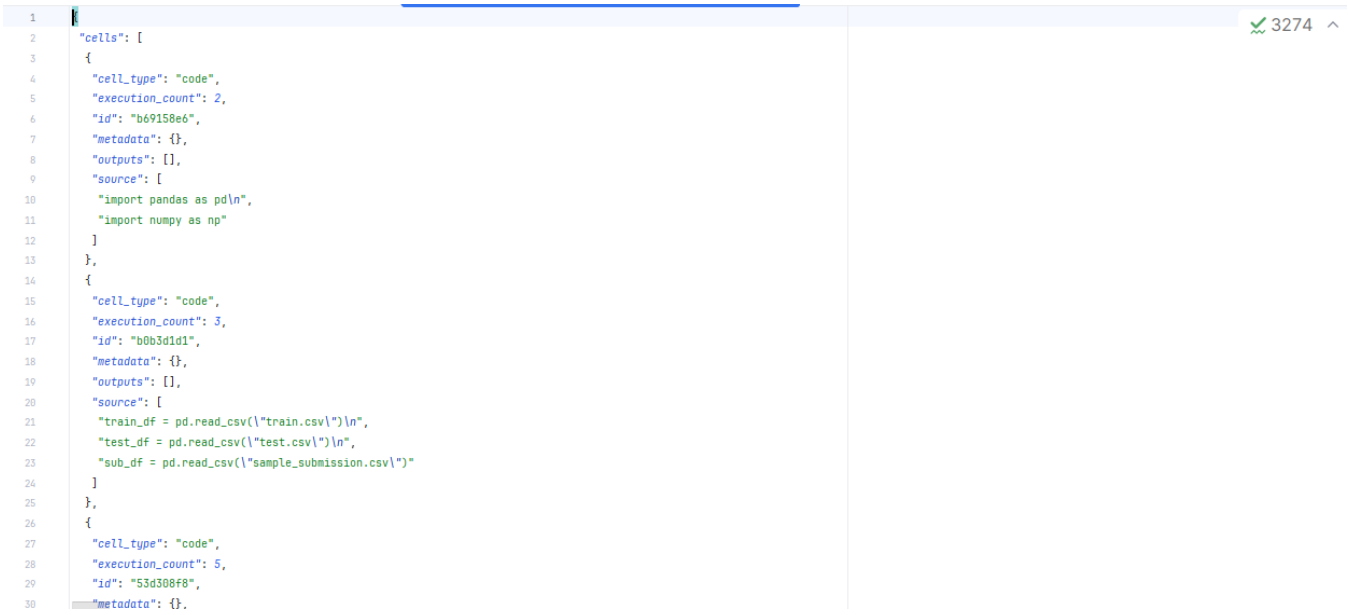
Screenshots of implementation:

Prediction Page:





Dataset Page:



```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "b69158e6",
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "b0b3d1d1",
   "metadata": {},
   "outputs": [],
   "source": [
    "train_df = pd.read_csv(\"train.csv\")\n",
    "test_df = pd.read_csv(\"test.csv\")\n",
    "sub_df = pd.read_csv(\"sample_submission.csv\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "id": "53d308f8",
   "metadata": {},
   "metadata": {},
```

1	0
2	0
3	1
4	0
5	1
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	1
14	0
15	0
16	0
17	0
18	0
19	0

Future Scope :

The future of brain stroke care is rapidly evolving with advancements in AI for early detection, improved neuroimaging techniques, and personalized treatments through genomics. Telemedicine and wearable devices are enhancing remote monitoring and emergency response. Emerging therapies like stem cell treatment and robotic rehabilitation are improving recovery outcomes. Continued research and public health initiatives will play a key role in stroke prevention and management

Github Link : devanshwadhwani2004/Brain-Stoke-Prediction-App

Conclusion:

In conclusion, brain stroke is a critical medical condition that requires urgent attention and long-term care. Despite its severity, advancements in medical science, technology, and artificial intelligence are paving the way for better prevention, early detection, and effective treatment strategies. Continued efforts in research, public awareness, and healthcare innovation are essential to reduce stroke-related mortality and improve the quality of life for survivors. With a collaborative approach, the burden of brain stroke can be significantly minimized in the future.