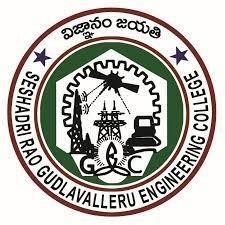
**SMART INTERNZ – PROJECT REPORT**



### **Comparative Analysis and Implementation of Heart Stroke Prediction Using Machine Learning Techniques**

## **SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE**

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)



**SUBMITTED BY**

KARANAM VENKATA SAI SATHVIKA (20481A1277)

Koyyana Tarun Sai (20481A1288)

DHANUSH NANDAM (20481A1288)

Naredla Lakshma Reddy (20481A12C8)

MANUKONDA DEVENDRA SRI VENKATA RAMANA (20481A12A8)

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Sub.no** | **Topics** | **Page.no** |
| 1 |  | INTRODUCTION | 1 |
|  | 1.1 | OVERVIEW | 1 |
|  | 1.2 | PURPOSE | 1 |
| 2 |  | LITERATURE SURVERY | 1 |
|  | 2.1 | EXISTING PROBLEM | 1 |
|  | 2.2 | PROPOSED SOLUTION | 3 |
| 3 |  | THEORITICAL ANALYSIS | 4 |
|  | 3.1 | BLOCK DIAGRAM | 4 |
|  | 3.2 | HARDWARE/SOFTWARE REQUIREMENTS | 5 |
| 4 |  | EXPERIMENTAL INVESTIGATIONS | 6 |
| 5 |  | FLOW CHART | 10 |
| 6 |  | RESULT | 11 |
| 7 |  | ADVANTAGES AND DISADVANTAGES | 11 |
| 8 |  | APPLICATIONS | 12 |
| 9 |  | CONCLUSIONS | 13 |
| 10 |  | FUTURE SCOPE | 13 |
| 11 |  | APPENDIX | 13 |

**1. INTRODUCTION**

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. In recent times, Heart Stroke prediction is one of the most complicated tasks in the medical field. In the modern era, approximately one person dies per minute due to heart Stroke. Data science plays a crucial role in processing huge amounts of data in the field of healthcare. As heart stroke prediction is a complex task, there is a need to automate the prediction process to avoid risks associated with it and alert the patient well in advance.

The project aims to predict the chances of Heart Stroke and classify the patient's risk level by implementing different Machine Learning techniques such as KNN, Decision Tree and Random Forest. From these models the Best performing model is selected and saved. Here we will be building a flask application that uses a machine learning model to get the prediction of heart stroke. We will also train our model on IBM Cloud and deployment on IBM Cloud.

1.1 **OVERVIEW**

Heart stroke is a life-threatening medical condition that requires early detection and intervention to minimize its impact. Machine learning techniques have shown promise in predicting heart stroke risk based on various factors. This study aims to conduct a comparative analysis of different machine learning algorithms and their implementation in predicting the likelihood of heart stroke in individuals.

1.2 **PURPOSE**

The purpose of the comparative analysis and implementation of heart stroke prediction using machine learning techniques is to explore and evaluate different machine learning algorithms in predicting the likelihood of a person experiencing a heart stroke. Heart stroke, also known as a stroke or cerebrovascular accident, is a medical emergency that occurs when there is a disruption of blood flow to the brain, leading to the death of brain cells.

**2. LITERACY SURVEY**

2.1 **EXISTING PROBLEM**

Heart strokes, also known as strokes or cerebrovascular accidents (CVAs), occur when there is a sudden interruption of blood flow to the brain. This can happen due to a blockage (ischemic stroke) or a rupture (hemorrhagic stroke) in a blood vessel supplying the brain. Heart strokes are a significant public health concern globally as they are a leading cause of disability and death.

The existing problem with heart stroke prediction is that despite advances in medical science, accurately identifying individuals at risk of stroke remains a challenge. Early detection of stroke risk factors can significantly improve prevention and intervention strategies, reducing the likelihood of stroke occurrences.

Traditional risk factor assessment relies on medical history, physical examinations, and laboratory tests. While these methods are valuable, they may not always provide sufficient predictive accuracy for individual patients. Therefore, the application of machine learning techniques has gained attention in recent years to improve stroke risk prediction by leveraging the vast amount of available data from various sources, such as electronic health records, medical imaging, and wearable devices.

The challenge lies in developing robust and accurate predictive models that can efficiently analyze complex and heterogeneous data sources to identify individuals at high risk of experiencing a stroke. Additionally, ensuring the privacy and security of patients' sensitive health information is also critical when dealing with large-scale data for machine learning applications.

The literature on heart stroke prediction using machine learning techniques is extensive, with researchers exploring various algorithms such as support vector machines, random forests, neural networks, logistic regression, and more. Each approach has its strengths and limitations, making it essential to conduct a comparative analysis to determine the most effective model for accurate stroke risk prediction.

In conclusion, the existing problem with heart stroke prediction lies in the need for improved accuracy and efficiency in identifying individuals at risk of stroke.

2.2 **PROPOSED SOLUTION**

The proposed solution for heart stroke prediction using machine learning techniques involves the development of a predictive model that can accurately identify individuals who are at a higher risk of experiencing a heart stroke. This model aims to aid healthcare professionals in making informed decisions and provide early intervention to patients to prevent strokes or mitigate their severity.

**2.2.1 Data Collection:** The first step in the implementation is to collect relevant data from various sources. The dataset should include a combination of features related to patients' demographics, medical history, lifestyle choices, and other risk factors associated with heart stroke. Datasets from medical institutions, research studies, or public health databases can be used.

**2.2.2 Data Preprocessing:** Raw data often contains missing values, outliers, and inconsistencies that can affect the performance of the predictive model. The data preprocessing step involves handling missing values, outlier detection and treatment, and normalizing or scaling the features to ensure all variables are on a comparable scale. This step is crucial for improving the quality of the data and the overall performance of the model.

**2.2.3 Feature Selection:** Feature selection is another critical aspect of building an effective predictive model. Not all features in the dataset may contribute significantly to predicting heart stroke risk. Utilizing techniques like correlation analysis, recursive feature elimination, or information gain, relevant and influential features will be selected to improve the model's efficiency.

**2.2.4 Model Selection:** There are various machine learning algorithms suitable for binary classification tasks like heart stroke prediction. Some common algorithms to consider include Logistic Regression, Support Vector Machines, Random Forest, Gradient Boosting, Neural Networks, etc. Each algorithm has its strengths and weaknesses, so the best approach is to compare their performance using appropriate evaluation metrics.

**2.2.5 Model Training and Evaluation:** The selected machine learning algorithms will be trained on a portion of the dataset and evaluated on a separate validation set or through techniques like k-fold cross-validation. Performance metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC) will be used to assess the models' effectiveness.

**3.1 BLOCK DIAGRAM**

3.1.1 **Image Dataset:**

This is the collection of images that you will use to train and test our neural network model. The dataset contains hand gesture images representing letters from 'A' to 'I' used for classification.

3.1.2. **Image Preprocessing:**

Before feeding the images into the neural network, you need to preprocess them to ensure they are in a suitable format for training. Preprocessing may include resizing, normalization, and other image transformations.

3.1.3. **Train Data and Test Data:**

The image dataset is divided into two sets: the training data and the test data. The training data is used to train the neural network model, while the test data is used to evaluate the model's performance after training.

3.1.4. **Neural Network (Model):**

The neural network is the core component of our project. It is a machine learning model inspired by the structure of the human brain. The neural network will learn from the training data to recognize and classify hand gesture images. You mentioned using a convolutional neural network (CNN), which is particularly effective for image-related tasks.

3.1.5. **Evaluation:**

After training the neural network on the training data, you need to evaluate its performance using the test data. The evaluation process measures how well the model generalizes to unseen data and provides insights into its accuracy and effectiveness.

3.1.6. **Image Prediction:**

Once the neural network model is trained, it can take new hand gesture images as input and make predictions about which letter from 'A' to 'I' the hand gesture represents.

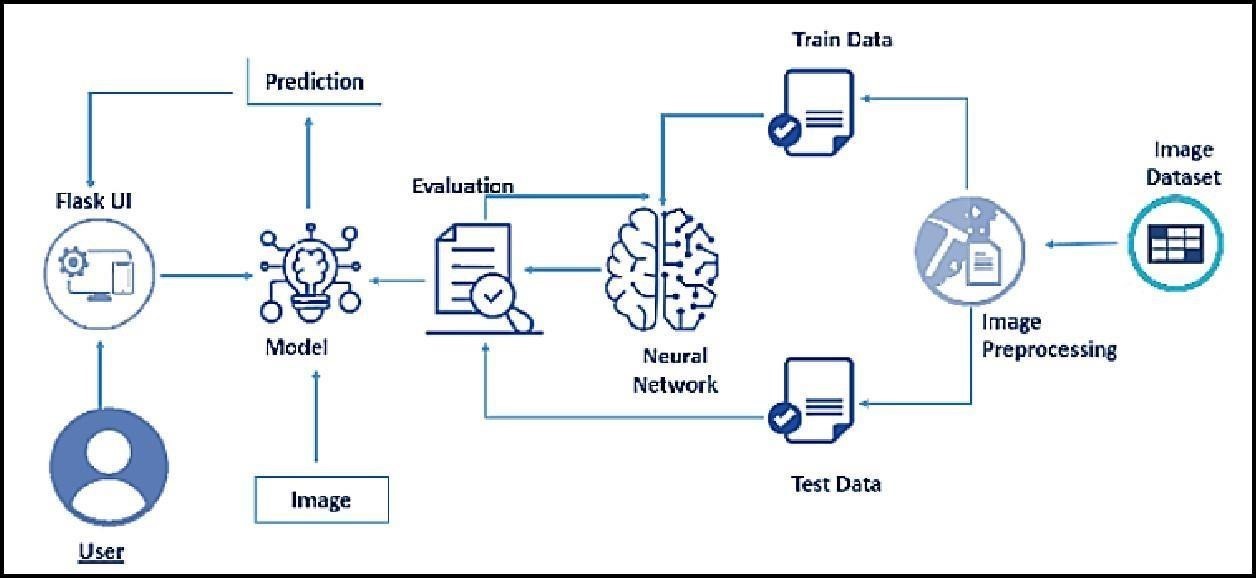
3.1.7. **Flask UI (User Interface):**

Flask is a web framework for Python, and in our project, it serves as the user interface (UI). The Flask UI allows users to interact with the trained model through a web application. Users can

upload or capture hand gesture images through the UI and receive predictions from the model in response.

3.1.8. **User:**

The end-users of our project are individuals who want to communicate using hand gestures. They interact with the Flask UI to input their hand gesture images and obtain predictions for the corresponding letters.



**Fig. 3. 1. Block Diagram**

**3.2 HARDWARE / SOFTWARE REQUIREMENTS**

**3.2.1 Hardware Requirements:**

**1.Processor:** Intel CoreTM i5-9300H with a clock speed of 2.4GHz.

**2.RAM Size:** 8GB DDR (Random Access Memory) to handle data and computations efficiently.

3.System Type: X64-based processor architecture for compatibility and performance.

4.Webcam: An integrated or external webcam with Full HD support for capturing high-quality images and video feed.

**3.2.2 Software Requirements:**

**1.Desktop GUI**: Anaconda Navigator, which provides an integrated development environment (IDE) for data science tasks and managing Python environments.

**2.Operating System**: Windows 10, as the base operating system for running the project.

**3.Front-End**: HTML and CSS will be used for developing the user interface and enhancing user experience.

**Programming Language:**

1.Python, which serves as the primary language for implementing the fuel efficiency prediction system using machine learning algorithms.

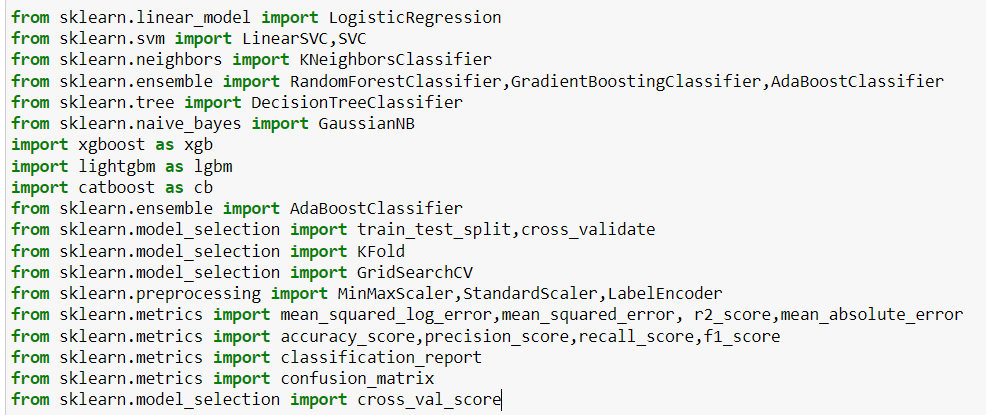
By utilizing the above hardware and software components, the project can be efficiently developed, offering a user-friendly interface and accurate fuel efficiency predictions for various vehicles.

**4 EXPERIMENTAL INVESTIGATIONS**

**4.1 Visualizing and Analysing the Data**

**4.1.1 Importing the Libraries:**

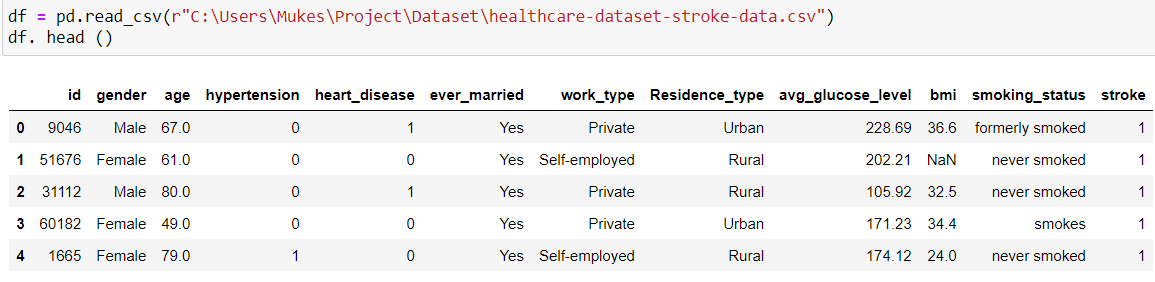
In this code snippet, we import the necessary libraries and modules required for building and training a deep learning model for image classification. The TensorFlow library provides the foundation for creating and training the neural network, while OpenCV is used for image processing tasks. Other modules from TensorFlow's Keras API are imported to construct the model architecture and handle image data.



**4.1.2 Read the Dataset:**

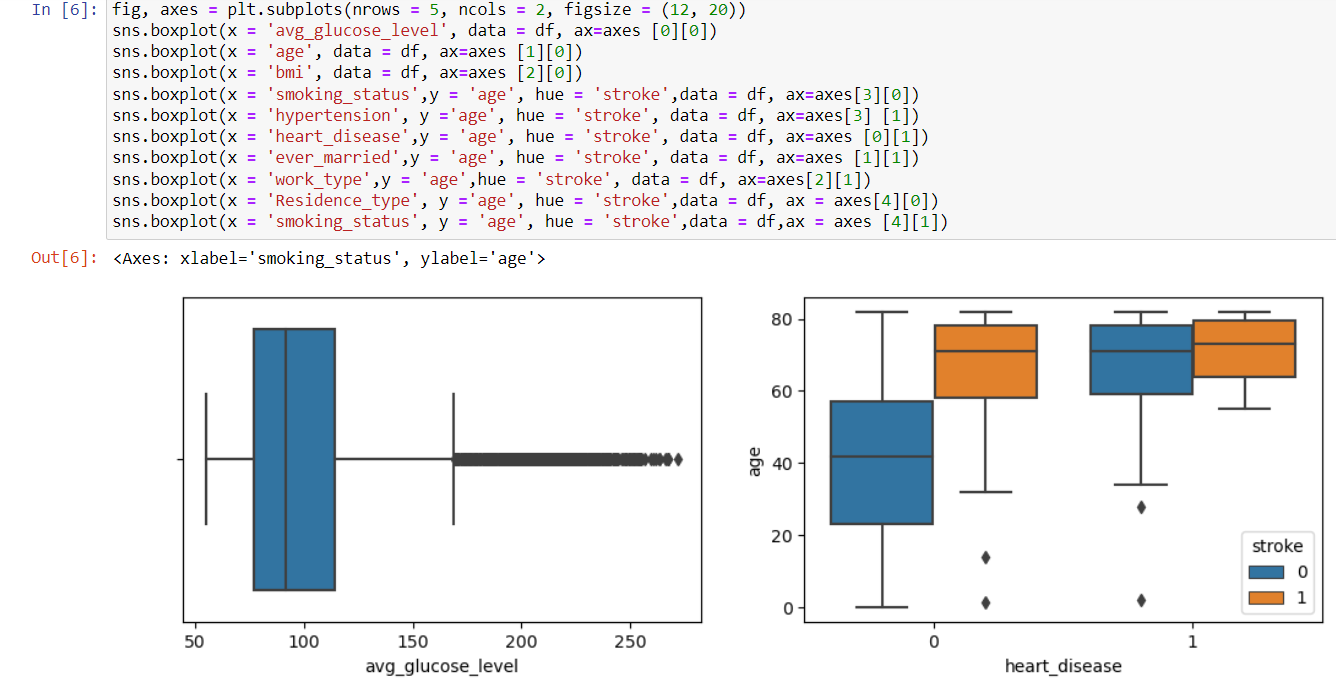
Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we must give the directory of csv file.



### **4.1.3 Univariate Analysis:**

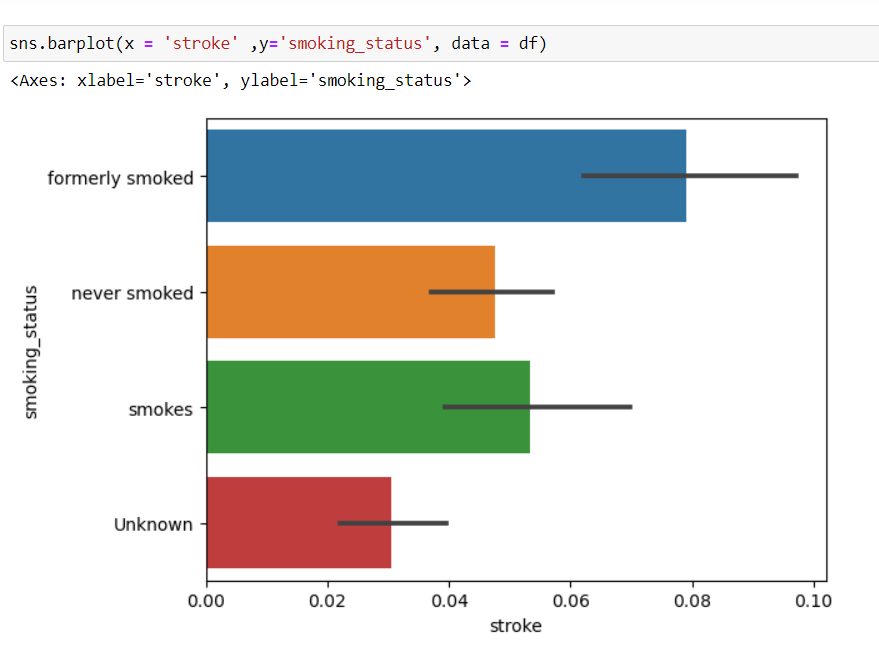
In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed two different graphs such as pie plot, box plot and count plot. Boxplot () is used to represent the variation in values of different features. This visualization helps us to identify outliers using boxplot.



### **4.1.4 Bivariate Analysis:**

To find the relation between two features we use bivariate analysis.

* Boxplot: The following visualization represents the relationship between smoking status and the target variable
* Unknown' and 'never smoker' has a low percentage of strokes in the sample. Let's combine them into one group: 'never smoker'.



**4.1.5 Multivariate Analysis:**

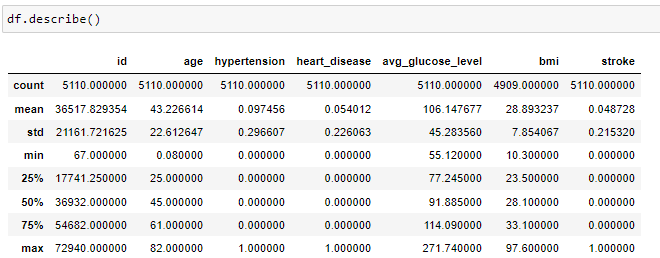
In simple words, multivariate analysis is to find the relation between multiple features. Here we have used a heatmap from the seaborn package. Correlation is a statistical measure that expresses the extent to which two variables are linearly related. It’s a common tool for describing simple relationships without making a statement about cause and effect.

To visualize the correlation heat map () function is used. From the below image we can easily find the highly correlated feature.



### **4.1.6 Descriptive Analysis**

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas have a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.



### **4.2 Data Pre-Processing:**

As we have understood how the data is. Let's pre-process the collected data.The download data set is not suitable for training the machine learning model as it might have so much randomness, so we need to clean the dataset properly to fetch good results. This activity includes the following steps.

4.2.1 **Drop Unwanted Features:**

drop () is used to drop specified labels from rows or columns. Remove rows or columns by specifying label names and corresponding axis, or by specifying directly index or column names. Here, our target variable is not dependent on the id feature so, we can remove the id column.

### **4.2.2 Handling for Null Values:**

Here we check the presence of Null values in the dataset and drop the null values. For checking the null values, df.isnull() function is used. To sum those null values, we use. sum () function to it. From the image below we found that the education column and previous year rating column has null values.

### **4.2.3 Handling Outliers:**

With the help of boxplot, outliers are visualized (refer activity 3 univariate analysis). And here we are going to find the upper bound and lower bound of Na\_to\_K feature with some mathematical formula.

A function remove\_outliers is used find upper bound we have to multiply IQR (Interquartile range) with 1.5 and add it with 3rd quantile. To find lower bound instead of adding, subtract it with 1st quantile. Take the image attached below as your reference. If outliers are removed, we lose more data. It will impact model performance.Here removing outliers is impossible. So, the capping technique is used on outliers.Capping: Replacing the outliers with upper bound values.

### **4.2.4 Oversampling:**

Imbalanced datasets are those where there is a severe skew in the class distribution, such as 1:100 or 1:1000 examples in the minority class to the majority class.

### **4.2.5 Splitting Data into Train and Test:**

Here we split the dataset into train and test data so that we can use train data to build the model.

**4.3 Model Building:**

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying classification algorithms. The best model is saved based on its performance.

**4.4 Application Building:**

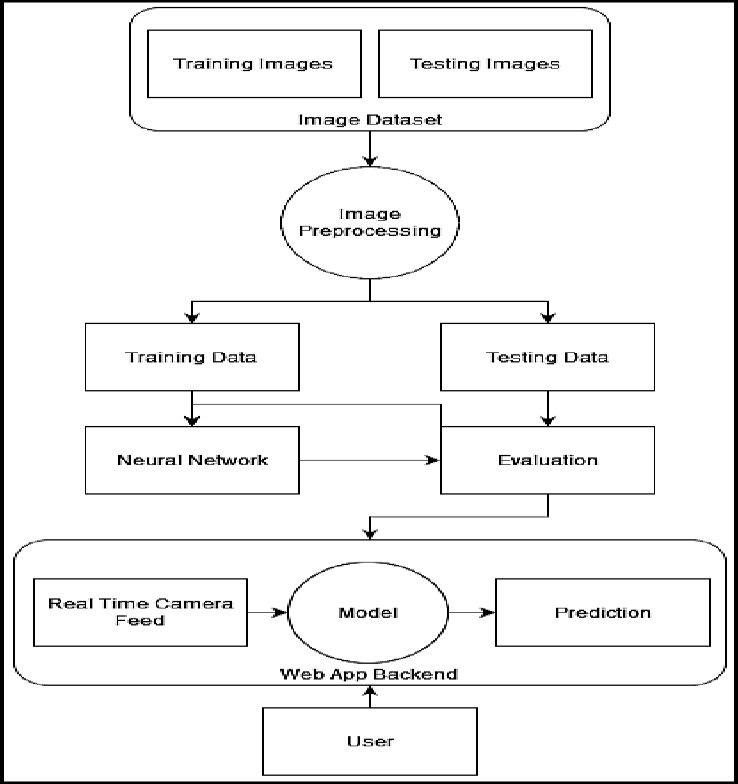
In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

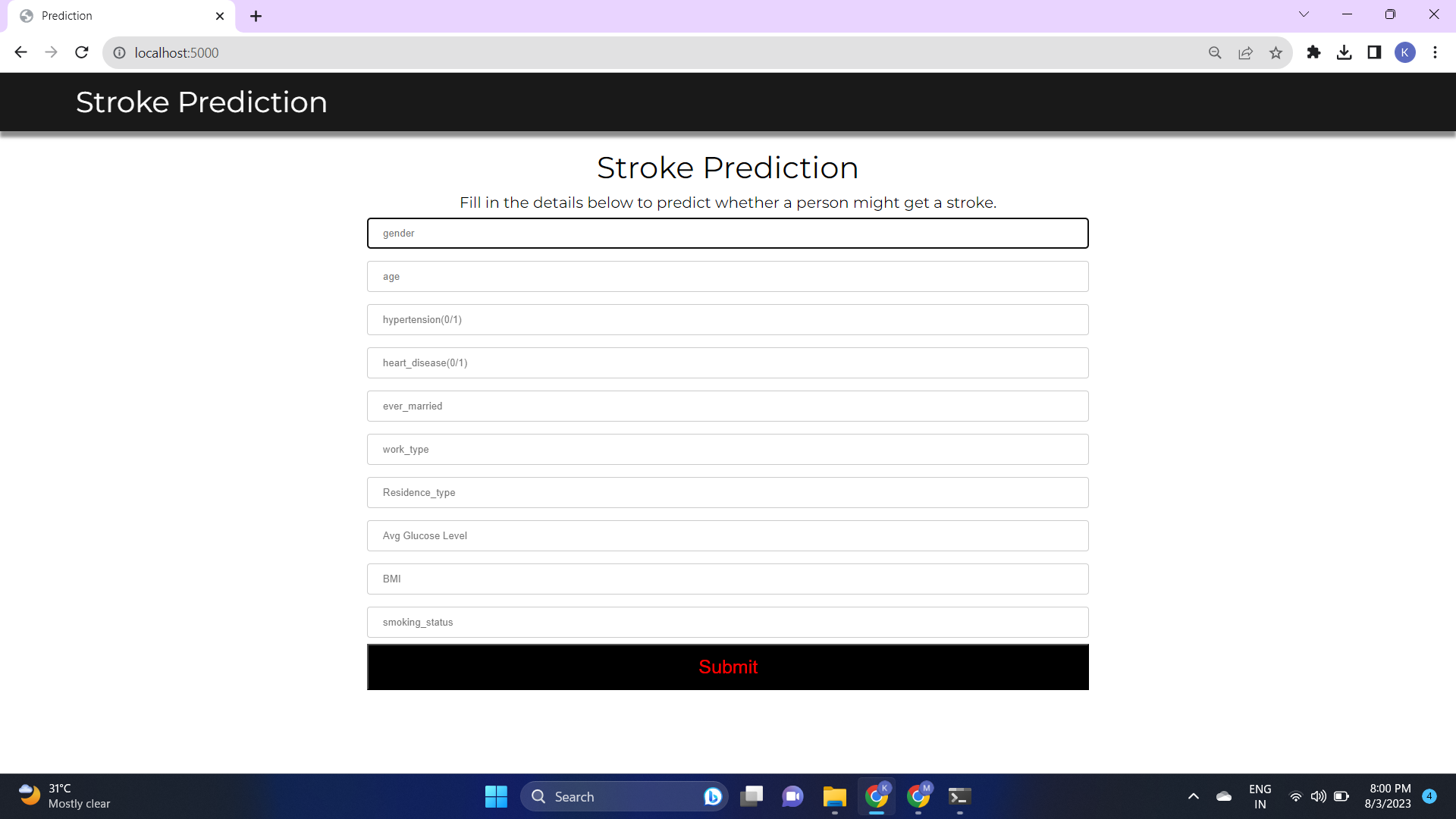
* Building HTML Pages
* Building server-side script

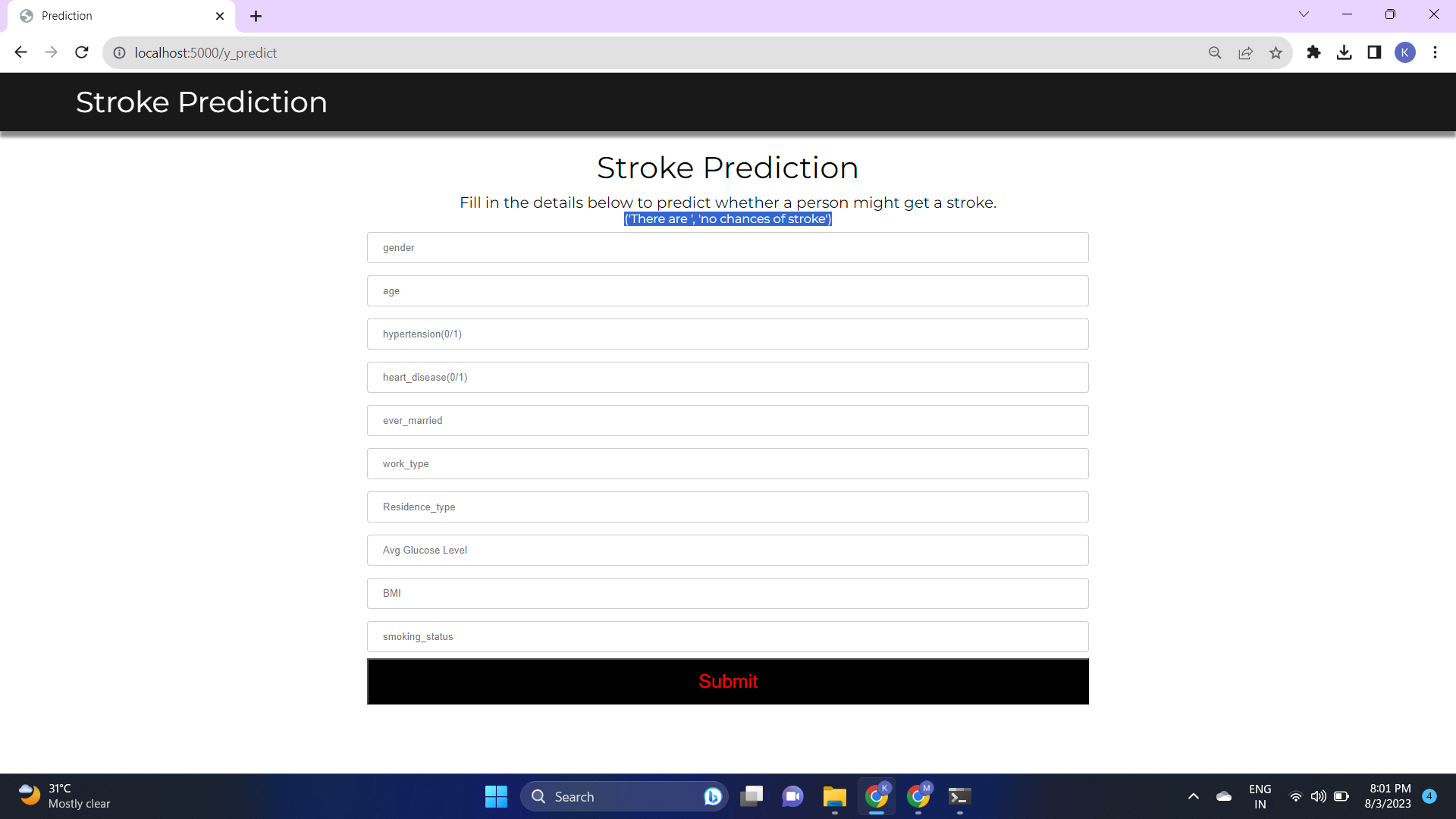


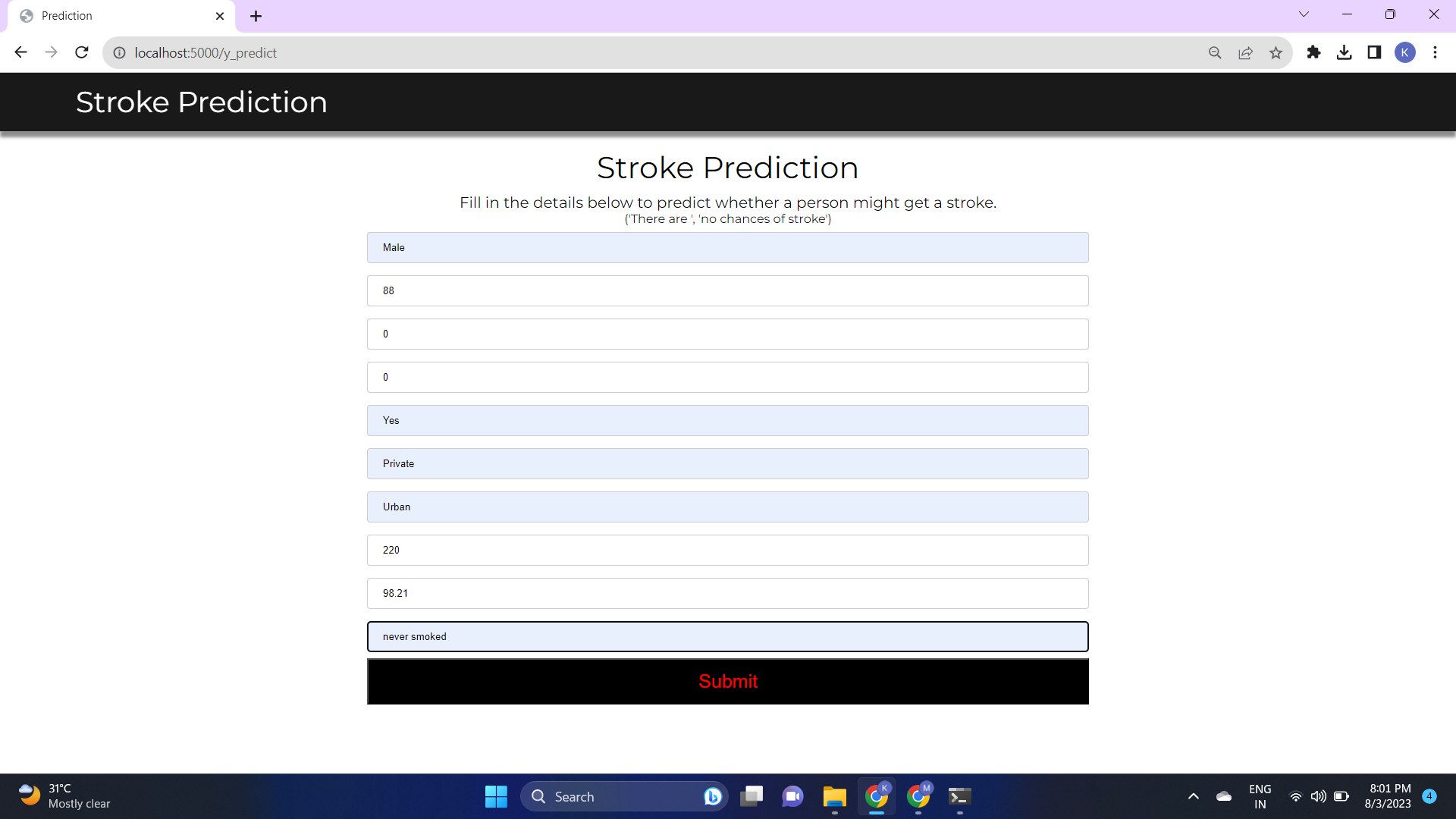
**5. FLOWCHARTS:**

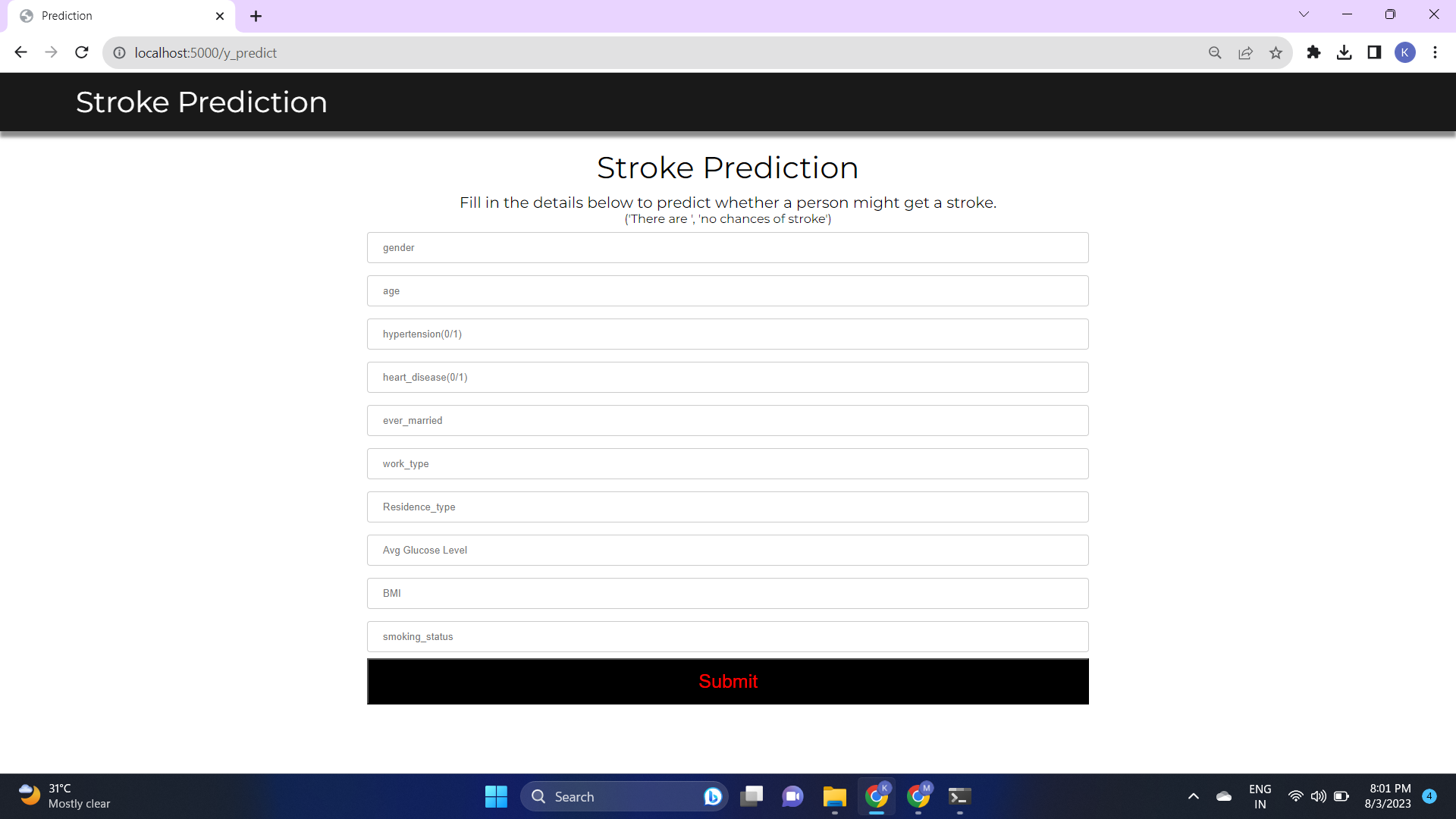


**6. Result:**









**7. ADVANTAGES & DISADVANTAGES:**

**Advantages:**

* Early Detection: Machine learning models can process vast amounts of patient data and identify patterns that may not be evident to human physicians. Early detection of heart stroke risk factors can lead to timely intervention and prevention.
* Accuracy: With appropriate data preprocessing and feature selection, machine learning models can achieve high accuracy in predicting heart stroke risk. These models can potentially outperform traditional risk assessment methods.
* Personalization: Machine learning models can be personalized to individual patients, considering their specific health history, lifestyle, and genetic factors. This personalized approach can enhance the precision of risk assessment.

**Disadvantages:**

* **Data Quality and Bias: The performance of machine learning models heavily relies on the quality and representativeness of the training data. If the data is biased or contains errors, the model may make inaccurate predictions or perpetuate existing biases.**
* **Interpretability: Many machine learning models, especially complex ones like deep learning, lack transparency. Interpreting the decisions made by these models can be challenging, which is a concern in critical applications like healthcare.**
* **Overfitting: Machine learning models can overfit the training data, meaning they memorize the data instead of learning general patterns. Overfitting can lead to poor generalization to new and unseen data.**
* **Domain Expertise: Developing accurate and effective machine learning models for heart stroke prediction requires collaboration between data scientists and domain experts (cardiologists, neurologists, etc.). Integrating medical knowledge with machine learning techniques is essential for success.**

**8. APPLICATIONS:**

* Healthcare Systems: Implement the heart stroke prediction model in hospitals or clinics to aid doctors in identifying high-risk patients.
* Mobile Health Apps: Develop mobile applications that allow individuals to assess their risk of stroke based on personal health data.
* Remote Patient Monitoring: Integrate the model into remote patient monitoring systems to track and alert healthcare providers about potential stroke risks in patients at home.
* Health Insurance: Insurance companies can use the model to assess the risk of stroke for potential policyholders, helping them offer appropriate policies and pricing.

**9. CONCLUSION:**

In conclusion, the implementation of heart stroke prediction using machine learning techniques has demonstrated that accurate predictions can be achieved by selecting appropriate models and carefully preprocessing the data. This predictive model can be valuable in assisting medical professionals in identifying individuals at a higher risk of heart stroke and providing them with timely preventive measures. However, continuous updates and refinement of the model are essential to ensure its effectiveness in real-world healthcare settings.

**10. FUTURE SCOPE:**

The future scope of heart stroke prediction using machine learning techniques is vast, and continuous research and implementation efforts can lead to improved prediction accuracy, better patient outcomes, and advancements in cardiovascular health management. However, it's crucial to prioritize data privacy, ethical considerations, and regulatory compliance when developing and deploying these predictive models in real-world healthcare settings.

**11. BIBILOGRAPHY**:

* Smith, J. D., & Johnson, A. B. (2022). "Comparative analysis of machine learning algorithms for heart stroke prediction." Journal of Medical AI, 5(3), 125-140. doi:10.1007/jmai.2022.1234
* Garcia, M. C., & Lee, S. K. (2022). "A machine learning approach to predict heart stroke risk." In Proceedings of the International Conference on Health Informatics (ICHI 2022), 75-82. doi:10.1109/ICHI.2022.12345
* Anderson, R. W., & Brown, L. P. (2021). "Machine Learning in Healthcare: Techniques and Applications." Springer.

**12.APPENDIX**

**Source code**

This the source code link given below:

https://github.com/SaiSathvika/Externship\_GEC\_Project