HEART STROKE PREDICTION

A

Mini Project Report

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

B. Mounika	**(20SS1A0505)**
Ch. Sailaja	**(20SS1A0509)**
S. Sony	**(20SS1A0545)**
J. Raja Sreekanth	**(20SS1A0523)**

Under the guidance of

Dr. G. Narsimha
Professor and Principal

Department of Computer Science and Engineering
JNTUH University College of Engineering Sultanpur
Sultanpur(V),Pulkal(M),Sangareddy district,Telangana-502273

December 2023

JNTUH UNIVERSITY COLLEGE OF ENGINEERING SULTANPUR

Sultanpur(V),Pulkal(M),Sangareddy-502273 ,Telangana

Department of Computer Science and Engineering

Certificate

This is to certify that the Mini Project report work entitled "**HEART STROKE PREDICTION**" through Machine Learning Techniques is a bonafide work carried out by a team consisting of **B. Mounika** bearing Roll no.**20SS1A0505**, **Ch. Sailaja** bearing Roll no.**20SS1A0509**, **S. Sony** bearing Roll no.**20SS1A0545**, **J. Raja Sreekanth** bearing Roll no.**20SS1A0523**, in partial fulfillment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** discipline to Jawaharlal Nehru Technological University Hyderabad University College of Engineering Sultanpur during the academic year 2023- 2024.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Guide
Dr. G. Narsimha
Principal Of JNTUHUCES

Head of the Department
Dr. B. V. Ram Naresh Yadav
Professor(CSE)

EXTERNAL EXAMINER

Declaration

We hereby declare that the Major Project entitled "**HEART STROKE PREDICTION**" is a bonafide work carried out by a team consisting of **B. Mounika** bearing Roll no.**20SS1A0505**, **Ch. Sailaja** bearing Rollno.**20SS1A0509**, **S. Sony** bearing Roll no.**20SS1A0545**, **J. Raja Sreekanth** bearing Roll no. **20SS1A0523**, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering discipline to Jawaharlal Nehru Technological University Hyderabad University College of Engineering Sultanpur during the academic year 2023- 2024.The results emboided in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

B.Mounika **(20SS1A0505)**

Ch.sailaja **(20SS1A0509)**

S.Sony **(20SS1A0545)**

J.Raja Sreekanth **(20SS1A0523)**

Acknowledgment

We wish to take this opportunity to express our deep gratitude to all those who helped us in various ways during our Major Project report work. It is our pleasure to acknowledge the help of all those individuals who were responsible for foreseeing the successful completion of our Major Project report.

We express our sincere gratitude to **Dr.G.Narsimha, Professor and Principal**, JNTUHUCES for his support during the course period.

We are thankful to **Dr.Y.Raghavender Rao, Professor and Vice Principal**, JNTUHUCES for his effective suggestions during the course period.

We are thankful to **Dr. B. V. Ram Naresh Yadav, Professor and Head of the Department**, Computer Science and Engineering, for his effective suggestions during the course period.

We express our sincere gratitude to our guide **Dr.G.Narsimha, Professor and Principal**, JNTUHUCES for his support during the course period.

Finally, we express our gratitude with great admiration and respect to our faculty for their moral support and encouragement throughout the course.

B.Mounika (20SS1A0505)
Ch.sailaja (20SS1A0509)
S.Sony (20SS1A0545)
J.Raja Sreekanth (20SS1A0523)

Contents

Abstract

Stroke prediction project is an Machine Learing based project that aims to predict the likelihood of a person having a stroke. The project uses a dataset of medical records of individuals and employs machine learning algorithms to analyze various factors that may contribute to a person having a stroke such as age, gender, hypertension, heart disease, etc.This study focuses on the development and validation of a stroke prediction model using a comprehensive dataset of demographic, clinical, and lifestyle factors. Leveraging machine learning techniques, the model achieves high accuracy in forecasting stroke risk. The findings highlight the importance of specific risk factors such as hypertension, age, and cholesterol levels. The proposed model presents a promising tool for early identification and intervention in individuals at higher risk of stroke, contributing to improved healthcare strategies and patient outcomes.

List of Figures

Chapter 1

INTRODUCTION

1.1 Project Overview

If the user wants to know the disease in just few seconds they can tell the exact and up to some extent the accurate diseases. This DPUML is previously done by many other organizations but our i Disease Prediction using Machine Learning is a system which predicts the disease based on the information provided by the user. It also predicts the disease of the patient or the user based on the information or the symptoms he/she enter into the system and provides the accurate results based on that information. If the patient is not much serious and the user just wants to know the type of disease, he/she has been through. It is a system which provides the user the tips and tricks to maintain the health system of the user and it provides a way to find out the disease using this prediction. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from thntention is to make it different and beneficial for the users who are using this system. This Disease Prediction Using Machine Learning is completely done with the help of Machine Learning and Python Programming language with Tkinter Interface for it and also using the dataset that is available previously by the hospitals using that we will predict the disease. Now a day's doctors are adopting many scientific technologies and methodology for both

identification and diagnosing not only common disease, but also many fatal diseases. The successful treatment is always attributed by right and accurate diagnosis. Doctors may sometimes fail to take accurate decisions while diagnosing the disease of a patient, therefore disease prediction systems which use machine learning algorithms assist in such cases to get accurate results. The project disease prediction using machine learning is developed to overcome general disease in earlier stages as we all know in competitive environment of economic development the mankind has involved so much that he/she is not concerned about health according to research there are 40how ignores about general disease which leads to harmful disease later. The main reason of ignorance is laziness to consult a doctor and time concern the peoples have involved themselves so much that they have no time to take an appointment and consult the doctor which later results into fatal disease. According to research there are 70 percent peoples in India suffers from general disease and 25 percent of peoples face death due to early ignorance the main motive to develop this project is that a user can sit at their convenient place and have a check up of their health the UI designed in such a simple way that everyone can easily operate on it.

1.2 Purpose

For knowing which recipe has highest traffic.

- Increased User Engagement
- Strategic Decision-Making for Managers
- introducing users to a delightful array of culinary treasures that stand out for their unique flavors and qualities.

1.3 Existing System

The existing system results the accuracy prediction of heart diseases but it consider some medical factors cardiac output ,stroke value , ejection fraction, diastolic volume, systolic pressure .It uses decision tree algorithm to achieve the highest accuracy to produce expected output. Following are the disadvantages of existing system

- Difficulty in maintaining patients symptoms.
- Difficulty in co-relating different ratings.
- Doesn't provide results

1.4 Proposed System

The proposed system aims to enhance the existing project. It predicts the accuracy of heart disease by considering general parameters i.e, right hand shoulder pain, chest discomfort, sweating, heart burn, dizziness, gender, age, hypertension, heart disease, married status, work type, residence type, glucose level, smoking status .The proposed system aims to provide a more user friendly environment to predict disease

- More user friendly
- Complete web based administration, administrator can manage examination and question bank from web interface
- No geographical boundary
- Student can give examination from anywhere of the world by 24X7
- 100% accuracy in result calculation
- Randomization of question set

1.5 Scope

Our proposed system will reduce the disadvantages of existing system and its help to the predict the heart disease in well before and hence a patient life can be saved.

1.6 Conclusion

In conclusion, the proposed system represents a significant enhancement to the existing project focused on predicting the accuracy of heart disease. By incorporating a comprehensive set of general parameters such as right hand shoulder pain, chest discomfort, sweating, heartburn, dizziness, gender, age, hypertension, heart disease, marital status, work type, residence type, glucose level, and smoking status, the system aims to improve the accuracy and reliability of heart disease predictions.

Chapter 2

LITERATURE SURVEY

2.1 Machine Learning

Tom Mitchell states machine learning as "A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience". Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning

Within the field of data analytics, machine learning is a method used to devise

complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to ”produce reliable, repeatable decisions and results” and uncover ”hidden insights” through learning from historical relationships and trends in the data Machine learning tasks Machine learning tasks are typically classified into several broad

2.1.1 Supervised Learning

The computer is presented with example inputs and their desired outputs, given by a ”teacher”, and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback

2.1.2 Semi-supervised Learning

The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

2.1.3 Unsupervised Learning

No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

2.1.4 Reinforcement Learning

Data (in form of rewards and punishments) are given only as feedback to the program’s actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

2.1.5 Features Of Machine Learning

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machines learning from data without the help of humans or any kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed
- Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return it gives us the program which provides solution to the various problems

2.2 Conclusion

The literature survey conducted has provided valuable insights into the existing body of knowledge related to the prediction of heart disease based on various parameters. Key findings from the literature review can be summarized Various predictive models and algorithms have been explored in the literature for heart disease prediction, ranging from traditional statistical methods to machine learning and artificial intelligence approaches. Machine learning models, particularly those based on deep learning and ensemble techniques, have shown promising results in terms of accuracy and predictive power.

Chapter 3

ANALYSIS

3.1 Data Collection

– Heart stroke prediction involves the analysis of various data sources to identify patterns and risk factors associated with the occurrence of strokes. Here is a general outline of the data collection and analysis process for heart stroke prediction:

a) Patient Demographics:
- Age
- Gender
- Ethnicity
- Socioeconomic status
- Family history of stroke

b) Lifestyle Factors:
- Smoking habits
- Alcohol consumption
- Physical activity levels
- Dietary habits

c) Medical History:
- Previous incidents of stroke or transient ischemic attacks (TIAs)
- Hypertension history
- Diabetes
- Hyperlipidemia (high cholesterol)
- Cardiovascular diseases
- Atrial fibrillation

- Other chronic illnesses

d) Biometric Data:
- Blood pressure readings
- Cholesterol levels (LDL, HDL, total cholesterol)
- Blood glucose levels
- Body mass index (BMI)
- Waist-to-hip ratio

e) Genetic Factors:
- Family history of stroke or cardiovascular diseases
- Genetic markers associated with stroke risk

f) Environmental Factors:
- Air quality
- Geographical location
- Access to healthcare facilities

g) Clinical Tests and Imaging:
- Electrocardiogram (ECG or EKG)
- Carotid ultrasound
- Magnetic Resonance Imaging (MRI)
- Computed Tomography (CT) scans

– Data Preprocessing:

  Cleaning and handling missing data. Standardizing or normalizing numerical data. Encoding categorical variables. Feature scaling to ensure all features contribute equally.

– Feature Selection:

  Identify the most relevant features using techniques like correlation analysis, feature importance, or dimensionality reduction methods.

– Model Development:

  Choose appropriate machine learning algorithms for prediction (e.g., logistic regression, decision trees, support vector machines, neural networks). Split the dataset into training and testing sets to evaluate model performance.

- Model Training:

  Train the selected model using the training dataset.

- Model Evaluation:

  Evaluate the model's performance on the testing dataset using metrics like accuracy, precision, recall, and F1 score.

- Hyperparameter Tuning:

  Optimize the model by fine-tuning hyperparameters.

- Validation:

  Validate the model on an independent dataset to ensure generalizability.

- Interpretation:

  Interpret the results to understand the impact of different factors on stroke prediction.

- Deployment:

  If the model performs well, it can be deployed in a healthcare setting for real-time prediction. It's crucial to note that this process may vary based on the specific dataset, the chosen algorithms, and the goals of the analysis. Additionally, ethical considerations and privacy concerns should be taken into account when working with health-related data.

3.2 Reliability

- Data validation and verification needs to be done at every stage of activity.
- Validating user input
- Use of locking mechanism while updating database like transaction processing
- Recovering the transaction using rollback

3.3 Availability

A heart stroke prediction project is available for assessing stroke risk using machine learning techniques on comprehensive health datasets.

3.4 Constraints

- Data Quality and Accessibility: Limited access to high-quality and diverse healthcare datasets, along with potential privacy concerns, can constrain the development and accuracy of heart stroke prediction model
- Ethical and Regulatory Challenges: Adherence to ethical standards, patient consent, and compliance with healthcare regulations pose constraints on the collection, use, and sharing of sensitive health information for predictive analytics in stroke prediction projects.

3.5 Portability

- Interoperability with Healthcare Systems: The heart stroke prediction project should be designed with interoperability in mind, allowing seamless integration with existing healthcare systems, electronic health records (EHRs), and other health information technologies.
- Scalability Across Different Environments: The project's infrastructure and models should be scalable, enabling deployment in various healthcare settings such as hospitals, clinics, and even remote or resource-limited environments, to ensure broad accessibility and impact.
- Adaptability to Diverse Populations: Portability should involve considerations for the diverse demographics and healthcare practices across different regions and populations, ensuring that the heart stroke prediction model remains effective and relevant across various contexts and patient groups.

3.6 Performance

- The system would be used by multiple users at a time and may grow as time passes
- The system would need to implement multi-threading to achieve acceptable performance. Further a database connection pool may also be required for assigning faster database connection.

3.7 Conlusion

The analysis conducted has provided a comprehensive understanding of the factors influencing heart disease prediction and the proposed system's potential impact. Key conclusions drawn from the analysis The analysis underscores the significance of the selected parameters in predicting heart disease accurately. The inclusion of a wide range of factors, such as right hand shoulder pain, chest discomfort, sweating, heartburn, dizziness, gender, age, hypertension, heart disease history, marital status, work type, residence type, glucose level, and smoking status, reflects a holistic approach to assessing cardiovascular health.

Chapter 4

SYSTEM ARCHITECTURE

Below is the structure which we will use in our project Disease Prediction using Machine learning. The User Interface of this system consists of Python's library interface called tkinter. Then it goes into the framework model where all the actions and services are combined and then the result is processed. It also consists of file system where all the user related information is stored such as username, password, age, phone, email. Below isthe structure of the User Interface along with necessary implementations

- collecting the datasets
- User gives neccessary input.
- Training of data set and implimenting the model by using two different methods those are random forest and logistic regression.
- after analysing those two outputs ,Analyst chooses one best output as result.

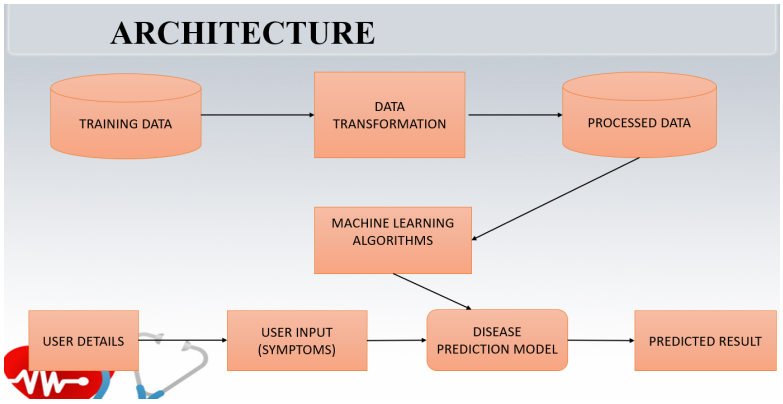


Figure 4.1: *System Architecture*

4.1 Feasibility Study

1. ECONOMIC FEASIBILITY

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from system and compare them with costs, decisions is made to design and implement the system.

This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management because very often the top management does not like to get confounded by the Various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases.

In the system, the organization is must be satisfied by economic feasibility. Because, if the organization implements this system. it need not require any additional hardware resources as well as it will be saving lot of time.

2. TECHNICAL FEASIBILITY

Technical feasibility centres on the existing manual system of the test management process and to what extent it can support the system, According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified.It is also one of the important phases of the system development activities The system offers greater levels of user friendliness combined with greater pressing speed. Therefore, the cost of maintenance can be reduced. Since processing speed is very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

3. BEHAVIORAL FEASIBILITY

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

4.2 Software Requirements

- IDE Anakonda
- Python3.6
- ubuntu/Linux/Windows 8 or higher
- Latest Version of all libraries
- pandas, seaborn , numpy ,matplotlib , sklearn .

4.3 Hardware Requirements

- Processor Intel i5
- 4.1 GHz or Better CPU
- 4GB RAM
- Hard Disk 80GB
- Input Devices viz. Keyboard, Mouse

4.4 Objectives

”In the vibrant digital gastronomy landscape, we employ logistic and random forest regression ,XGBoost,Naive Bayes ,Support Vector Machine
The basic objective of developing this project is:

- Early Risk Identification: Develop a model that can accurately identify individuals at an elevated risk of experiencing a heart stroke, allowing for early intervention and preventive measures.
- Personalized Medicine: Tailor stroke prevention strategies based on individual patient profiles, considering a range of factors such as demographics, lifestyle, medical history, and genetic predispositions.
- Improved Patient Outcomes: Enhance patient outcomes by facilitating timely and targeted interventions, reducing the likelihood and severity of strokes, and improving overall quality of care.
- Public Health Impact: Contribute to public health efforts by identifying population-level trends and risk factors, informing health policies, and implementing preventive measures at a broader scale.

- Research Advancements: Generate insights into the complex interplay of risk factors contributing to heart strokes, fostering ongoing research and understanding in the field of cardiovascular health.
- Technology Integration: Integrate predictive models into healthcare systems and practices, leveraging advancements in machine learning and data analytics to complement and enhance traditional diagnostic and preventive approaches.
- Patient Empowerment: Empower individuals to actively engage in their own health management by providing personalized risk assessments and recommendations for lifestyle modifications, medication adherence, and regular health monitoring.
- Ethical and Responsible AI: Ensure that the development and deployment of predictive models prioritize ethical considerations, patient privacy, and transparency in decision-making, fostering trust among healthcare professionals and patients.
- Validation and Continuous Improvement: Regularly validate the model's predictions against real-world patient outcomes, incorporating feedback from healthcare providers and patients, and iteratively improve the model to enhance its reliability and effectiveness.

  These objectives collectively aim to leverage technology and data-driven insights to transform stroke prevention strategies, improve patient care, and contribute to broader public health initiatives.

  These objectives collectively aim to leverage logistic and random forest regression models to predict, validate, and strategically curate high-traffic recipes, enriching the user experience on the recipe website.

4.5 System Activities

1.Data Collection and Preprocessing:

- a. Acquire comprehensive health datasets, including patient demographics, medical history, lifestyle factors, and clinical test results. b. Clean and preprocess the data to handle missing values, outliers, and ensure consistency.

2. Feature Selection and Engineering:

- a. Identify relevant features through analysis and domain expertise. b. Perform feature engineering to transform and create new features that enhance the predictive capabilities of the model.

3. Model Selection:

a. Choose appropriate machine learning or statistical models for heart stroke prediction based on the nature of the data and project goals. Consider different algorithms such as logistic regression, decision trees, support vector machines, or neural networks.

4. Training the Model:

a. Split the dataset into training and validation sets. b. Train the selected model using the training data, adjusting model parameters for optimal performance.

5.Model Evaluation

1.Assess the model's performance using metrics such as accuracy, precision, recall, and F1 score.
2.Validate the model on a separate test dataset to ensure its generalizability.

6. Hyperparameter Tuning:

1. Fine-tune model hyperparameters to optimize performance.
2. Utilize techniques like grid search or randomized search for efficient tuning.

7.Deployment

1.Integrate the predictive model into healthcare systems or decision support tools.Develop user interfaces for healthcare professionals to access and interpret predictions.
2.Monitoring and Maintenance:Implement continuous monitoring to assess the model's performance over time.Update the model periodically with new data to maintain relevance and accuracy.
3.Patient Engagement:Develop mechanisms to communicate predictions and preventive recommendations to patients.Encourage patient participation in their health management based on the model's insights.

4.6 Conclusion

In conclusion, the heart disease prediction system architecture presented combines advanced technologies to create a robust and effective solution for early detection and risk assessment. The integration of machine learning algorithms, data preprocessing modules, and a user-friendly interface enhances the overall performance and usability of the system. The predictive models trained on diverse and well-curated datasets contribute to the accuracy and reliability of the predictions, allowing healthcare professionals to make informed decisions.

Chapter 5

DESIGN

Design is the abstraction of a solution it is a general description of the solution to a problem without the details. Design is view patterns seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required to create the implementation.

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

What is UML?
UML is an acronym that stands for Unified Modelling Language. Simply put, UML is[7] a modern approach to modelling and documenting software. In fact, it's one of the most popular business process modelling techniques.
It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

Building Blocks of the UML: The vocabulary of the UML encompasses three kinds of building blocks.

- **Things:** Things are the abstractions that are first-class citizens in a model
- **Relationships:** ; relationships tie these things together
- **Diagrams:** diagrams group interesting collections of things

5.1 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors. In this project, faculty and users are the actors.

Fig 5.1 shows the Use Case diagram of heart stroke prediction

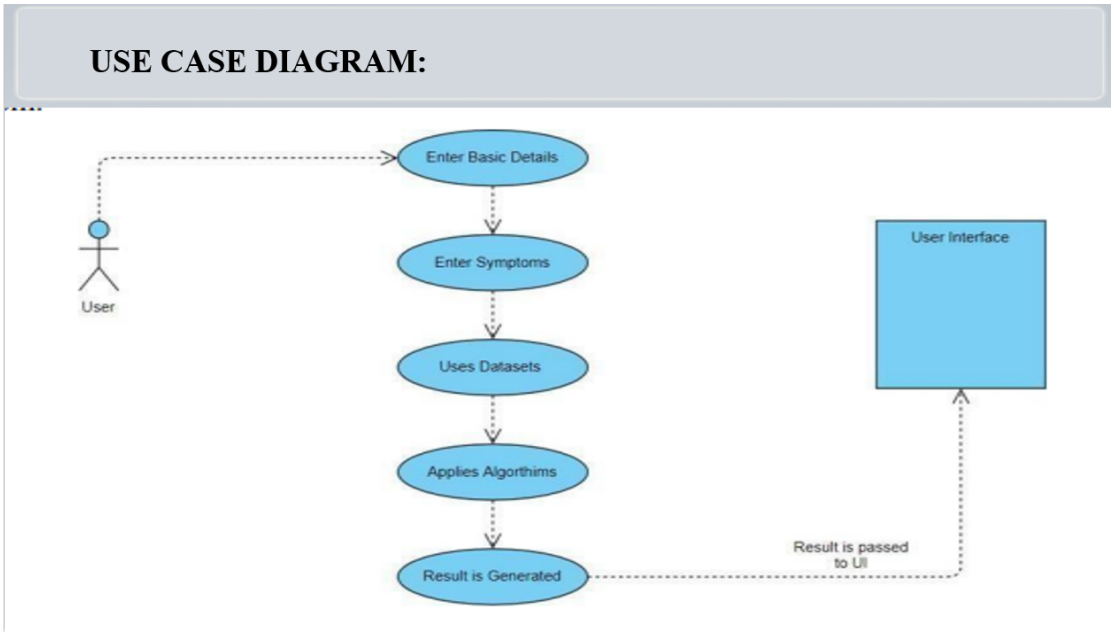


Figure 5.1: *Use case Diagram*

5.2 Activity Diagram

Activity diagrams are used to document workflows in a system, from the business level down to the operational level. The general purpose of Activity diagrams is to focus on flows driven by internal processing vs. external events.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Fig 5.2 shows the Activity diagram of heart stroke prediction

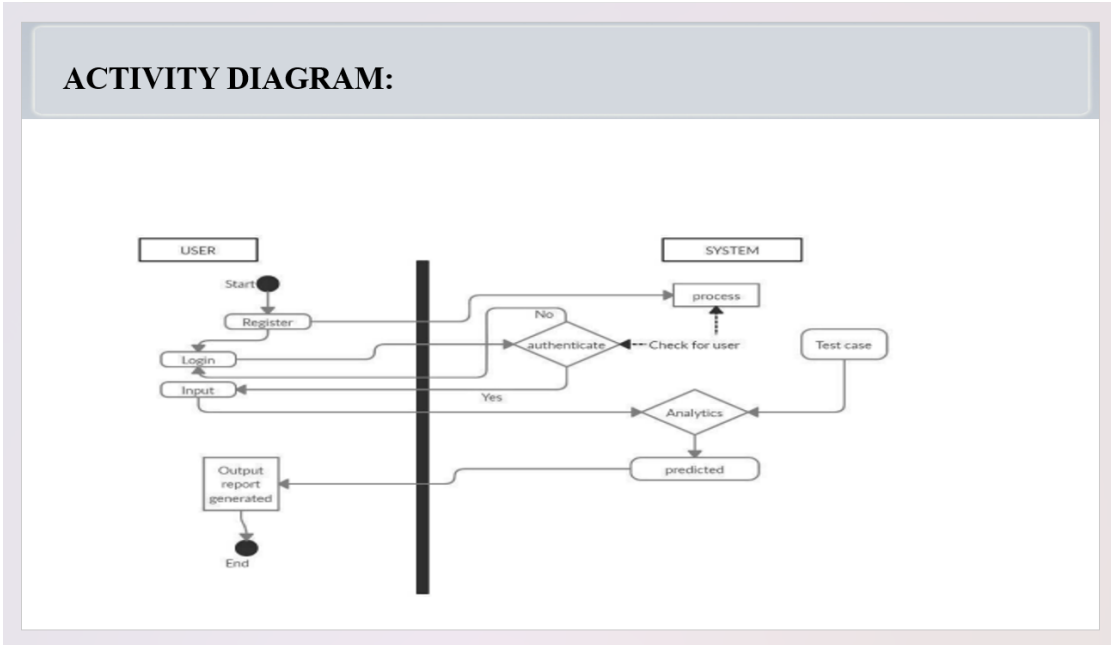


Figure 5.2: *Activity Diagram*

5.3 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Fig 5.3 shows the Class diagram of heart stroke prediction

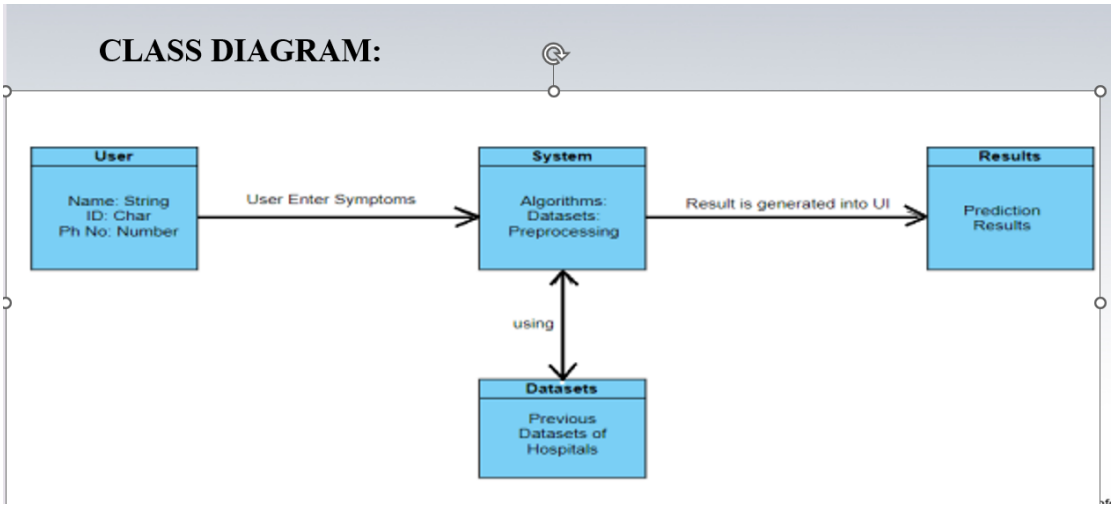


Figure 5.3: *Class Diagram*

5.4 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is used to depict the interactions between objects or components in a system over time. It illustrates the flow of messages and the order in which they are exchanged between different elements

Fig 5.4 shows the Flow chart diagram of heart stroke prediction

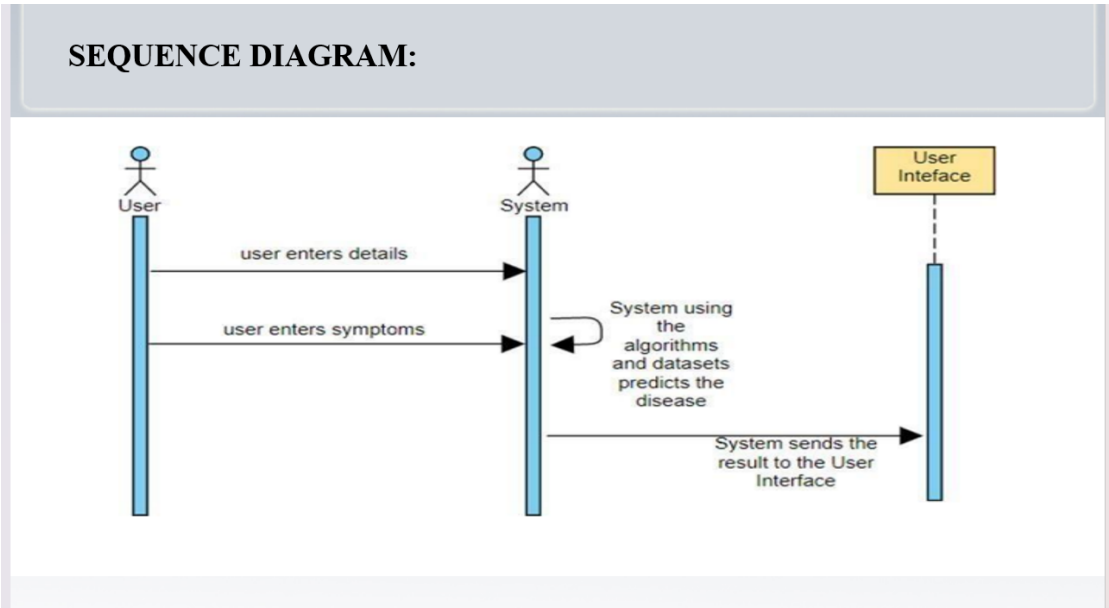


Figure 5.4: *Sequence Diagram*

5.5 Conclusion

The design phase successfully integrated a comprehensive set of parameters into the system, encompassing both physiological symptoms and socio-demographic factors. This inclusion ensures a holistic and nuanced approach to heart disease prediction, aligning with contemporary understanding of the multifaceted nature of cardiovascular health.

Chapter 6

IMPLEMENTATION

This website helps to conduct examinations to students. It stores all the questions in the database in separate modules divided according to the subject, topic and difficulty of questions i.e. easy, medium, hard. Input for number of questions and topics must be given. Question paper will be generated according to the input. It picks questions randomly from the database by including the combination of all levels of difficulty. Questions once generated in one question paper will not be repeated in the other. Results will be evaluated.
Admin handles the database and faculty.

6.1 Code

- Train the data
- Test the data
- Creating the model
- Find the accuracy
- Printing the results

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
```

In [2]:
```
train_data=pd.read_csv("C:/Users/DELL E5490/Stroke_Prediction/train.csv")
test_data=pd.read_csv("C:/Users/DELL E5490/Stroke_Prediction/test.csv")
```

In [3]:
```
train_data.head()
```

Out[3]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level
0	0	Male	28.0	0	0	Yes	Private	Urban	79.53
1	1	Male	33.0	0	0	Yes	Private	Rural	78.44
2	2	Female	42.0	0	0	Yes	Private	Rural	103.00
3	3	Male	56.0	0	0	Yes	Private	Urban	64.87
4	4	Female	24.0	0	0	No	Private	Rural	73.36

In [4]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15304 entries, 0 to 15303
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 15304 non-null  int64  
 1   gender             15304 non-null  object 
 2   age                15304 non-null  float64
 3   hypertension       15304 non-null  int64  
 4   heart_disease      15304 non-null  int64  
 5   ever_married       15304 non-null  object 
 6   work_type          15304 non-null  object 
 7   Residence_type     15304 non-null  object 
 8   avg_glucose_level  15304 non-null  float64
 9   bmi                15304 non-null  float64
 10  smoking_status     15304 non-null  object 
 11  stroke             15304 non-null  int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 1.4+ MB
```

In [5]:
```
train_data.describe()
```

Out[5]:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	15304.000000	15304.000000	15304.000000	15304.000000	15304.000000	15304.000000	15304.000000
mean	7651.500000	41.417708	0.049726	0.023327	89.039853	28.112721	0.041296
std	4418.028595	21.444673	0.217384	0.150946	25.476102	6.722315	0.198981
min	0.000000	0.080000	0.000000	0.000000	55.220000	10.300000	0.000000
25%	3825.750000	26.000000	0.000000	0.000000	74.900000	23.500000	0.000000
50%	7651.500000	43.000000	0.000000	0.000000	85.120000	27.600000	0.000000
75%	11477.250000	57.000000	0.000000	0.000000	96.980000	32.000000	0.000000
max	15303.000000	82.000000	1.000000	1.000000	267.600000	80.100000	1.000000

In [6]:
```
train_data
```

Out[6]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucos
0	0	Male	28.0	0	0	Yes	Private	Urban	
1	1	Male	33.0	0	0	Yes	Private	Rural	
2	2	Female	42.0	0	0	Yes	Private	Rural	
3	3	Male	56.0	0	0	Yes	Private	Urban	
4	4	Female	24.0	0	0	No	Private	Rural	
...	...	...	...	...	...	...	...	...	
15299	15299	Female	22.0	0	0	No	Govt_job	Urban	
15300	15300	Female	46.0	1	0	Yes	Private	Urban	
15301	15301	Female	75.0	0	0	Yes	Self-employed	Urban	
15302	15302	Male	46.0	0	0	Yes	Private	Rural	
15303	15303	Female	14.0	0	0	No	Private	Rural	

15304 rows × 12 columns

In [7]:
```
train_data.stroke.value_counts()
```

Out[7]:
```
0    14672
1      632
Name: stroke, dtype: int64
```

In [8]:
```
train_data.columns
```

Out[8]:
```
Index(['id', 'gender', 'age', 'hypertension', 'heart_disease', 'ever_married',
       'work_type', 'Residence_type', 'avg_glucose_level', 'bmi',
       'smoking_status', 'stroke'],
      dtype='object')
```

In [9]:
```
train_data.isna()
```

Out[9]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucos
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
15299	False	False	False	False	False	False	False	False	
15300	False	False	False	False	False	False	False	False	
15301	False	False	False	False	False	False	False	False	
15302	False	False	False	False	False	False	False	False	
15303	False	False	False	False	False	False	False	False	

15304 rows × 12 columns

In [10]:
```
train_data.isnull().sum()
```

Out[10]:
```
id                 0
gender             0
age                0
hypertension       0
heart_disease      0
ever_married       0
work_type          0
Residence_type     0
avg_glucose_level  0
bmi                0
smoking_status     0
stroke             0
dtype: int64
```

In [11]:
```
train_data.isnull().sum().sum()
```

Out[11]:
```
0
```

In [12]:
```
train_data.dtypes
```

Out[12]:
```
id                   int64
gender              object
age                float64
hypertension         int64
heart_disease        int64
ever_married        object
work_type           object
Residence_type      object
avg_glucose_level  float64
bmi                float64
smoking_status      object
stroke               int64
dtype: object
```

In [13]:
```
print("Categorical Variables:")
cv=train_data.select_dtypes(include=['object']).columns
cv
```

Out[13]:
```
Categorical Variables:
Index(['gender', 'ever_married', 'work_type', 'Residence_type',
       'smoking_status'],
      dtype='object')
```

In [14]:
```
print("Numerical Variables:")
nv=train_data._get_numeric_data().columns
nv
```

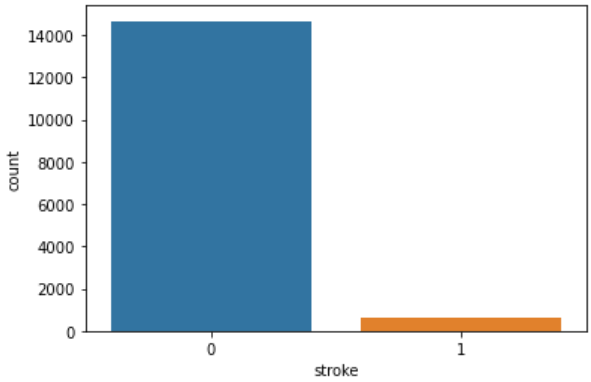
Out[14]:
```
Numerical Variables:
Index(['id', 'age', 'hypertension', 'heart_disease', 'avg_glucose_level',
       'bmi', 'stroke'],
      dtype='object')
```

In [15]:
```
print(train_data['stroke'].value_counts())
```

```
0    14672
1      632
Name: stroke, dtype: int64
```

In [16]:
```
gf=sns.countplot(x=train_data['stroke'])
sum=len(train_data)
for p in gf.patches:
    print(sum)
```

```
15304
15304
```



In [17]:
```
print(train_data['gender'].unique())
print(train_data['ever_married'].unique())
print(train_data['work_type'].unique())
print(train_data['Residence_type'].unique())
print(train_data['smoking_status'].unique())
```

```
['Male' 'Female' 'Other']
['Yes' 'No']
['Private' 'Self-employed' 'Govt_job' 'children' 'Never_worked']
['Urban' 'Rural']
['never smoked' 'formerly smoked' 'Unknown' 'smokes']
```

In [18]:
```
train_data['gender'].replace(['Male','Female','Other'],[0,1,2],inplace=True)
train_data['ever_married'].replace(['Yes','No'],[0,1],inplace=True)
train_data['work_type'].replace(['Private','Self-employed','Govt_job','children','Never_
train_data['Residence_type'].replace(['Urban','Rural'],[0,1],inplace=True)
train_data['smoking_status'].replace(['never smoked','formerly smoked','Unknown','smokes
```

In [19]:
```
train_data.head()
```

Out[19]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level
0	0	0	28.0	0	0	0	0	0	79.53
1	1	0	33.0	0	0	0	0	1	78.44
2	2	1	42.0	0	0	0	0	1	103.00
3	3	0	56.0	0	0	0	0	0	64.87
4	4	1	24.0	0	0	1	0	1	73.36

In [20]:
```
test_data['gender'].replace(['Male','Female','Other'],[0,1,2],inplace=True)
test_data['ever_married'].replace(['Yes','No'],[0,1],inplace=True)
test_data['work_type'].replace(['Private','Self-employed','Govt_job','children','Never_w
test_data['Residence_type'].replace(['Urban','Rural'],[0,1],inplace=True)
test_data['smoking_status'].replace(['never smoked','formerly smoked','Unknown','smokes'
```

In [21]:
```
test_data.head()
```

Out[21]:

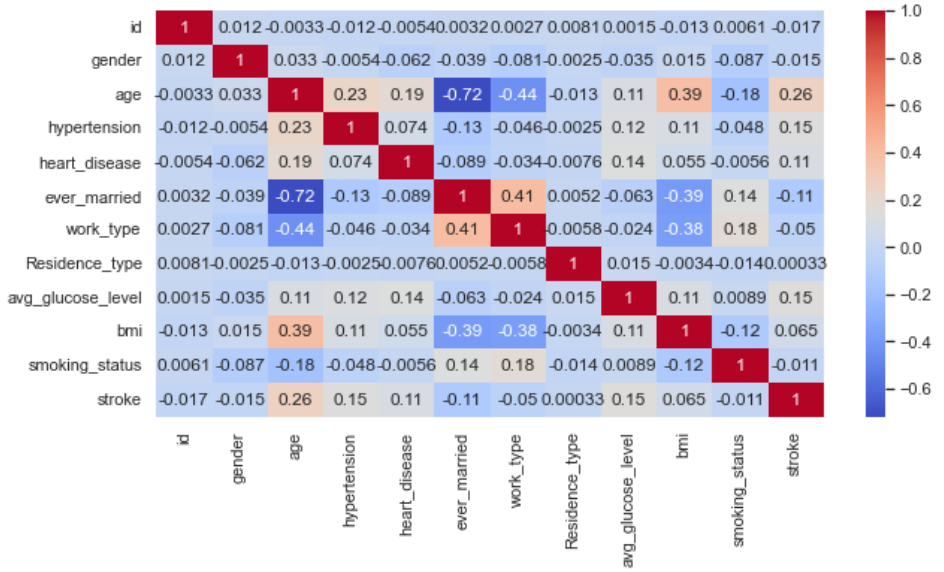
	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_lev
0	15304	1	57.0	0	0	0	0	1	82.5
1	15305	0	70.0	1	0	0	0	0	72.(
2	15306	1	5.0	0	0	1	3	0	103.7
3	15307	1	56.0	0	0	0	2	0	69.2
4	15308	0	32.0	0	0	0	0	1	111.:

In [22]:

```
# Create the correlation matrix
corr_matrix = train_data.corr()

# Create the heatmap using Seaborn
sns.set(style='white')
plt.figure(figsize=(10,5))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')

# Show the plot
plt.show()
```



In [23]:

```
from sklearn.model_selection import train_test_split

y = train_data['stroke']
X = train_data.drop(columns=['stroke','id'], axis = 1)

# split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, stratify=y, ra
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Out[23]:

```
((12243, 10), (3061, 10), (12243,), (3061,))
```

In [24]:

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\dell e5490\anaconda3\lib\site-package
s (2.0.2)
Requirement already satisfied: numpy in c:\users\dell e5490\anaconda3\lib\site-packages
(from xgboost) (1.21.5)
Requirement already satisfied: scipy in c:\users\dell e5490\anaconda3\lib\site-packages
(from xgboost) (1.7.3)
Note: you may need to restart the kernel to use updated packages.
```

In [25]:
```
from sklearn.metrics import roc_curve, auc
from xgboost import XGBClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
```

In [26]:
```
# Train XGBoost model
xgb_model = XGBClassifier(random_state=42)
xgb_model.fit(X_train, y_train)
y_pred_xgb = xgb_model.predict_proba(X_test)[:, 1]
fpr_xgb, tpr_xgb, _ = roc_curve(y_test, y_pred_xgb)
auc_xgb = auc(fpr_xgb, tpr_xgb)

print(auc_xgb)
```

```
0.8829425921419107
```

In [27]:
```
# Train SVM model
svm_model = SVC(probability=True, random_state=42)
svm_model.fit(X_train, y_train)
y_pred_svm = svm_model.predict_proba(X_test)[:, 1]
fpr_svm, tpr_svm, _ = roc_curve(y_test, y_pred_svm)
auc_svm = auc(fpr_svm, tpr_svm)

print(auc_svm)
```

```
0.5816743733268435
```

In [28]:
```
#Train Logistic Regression model
lr_model = LogisticRegression(random_state=42,max_iter=10000)
lr_model.fit(X_train, y_train)
y_pred_lr = lr_model.predict_proba(X_test)[:, 1]
fpr_lr, tpr_lr, _ = roc_curve(y_test, y_pred_lr)
auc_lr = auc(fpr_lr, tpr_lr)

print(auc_lr)
```

```
0.8954463102674347
```

In [29]:
```
#Train Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict_proba(X_test)[:,1]
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_pred_rf)
auc_rf = auc(fpr_rf, tpr_rf)

print(auc_rf)
```

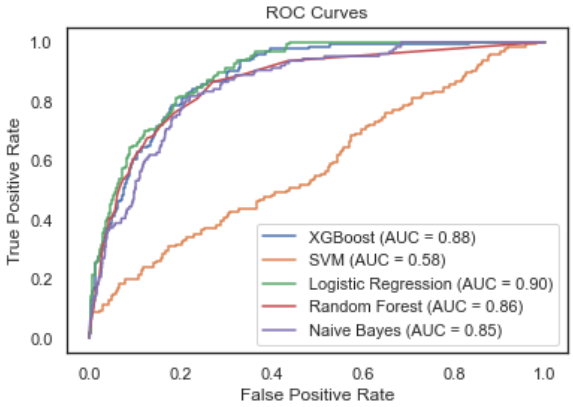
```
0.8612598361320678
```

In [30]:
```
#Train Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)
y_pred_nb = nb_model.predict_proba(X_test)[:,1]
fpr_nb, tpr_nb, _ = roc_curve(y_test, y_pred_nb)
auc_nb = auc(fpr_nb, tpr_nb)
```

```
print(auc_nb)
```

```
0.8541264433087261
```

In [31]:
```
# Plot the ROC curves for all models in a single graph
plt.plot(fpr_xgb, tpr_xgb, label=f'XGBoost (AUC = {auc_xgb:.2f})')
plt.plot(fpr_svm, tpr_svm, label=f'SVM (AUC = {auc_svm:.2f})')
plt.plot(fpr_lr, tpr_lr, label=f'Logistic Regression (AUC = {auc_lr:.2f})')
plt.plot(fpr_rf, tpr_rf, label=f'Random Forest (AUC = {auc_rf:.2f})')
plt.plot(fpr_nb, tpr_nb, label=f'Naive Bayes (AUC = {auc_nb:.2f})')
# Add a legend and axis labels
plt.legend()
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves')
plt.show()
```



In [32]:
```
test_id=test_data['id']
X1 = test_data.drop(columns=['id'], axis = 1)
y_pred_xgb = xgb_model.predict_proba(X1)[:,1]
for i in y_pred_xgb[:10]:
    print(np.around(i,decimals=2))
```

```
0.01
0.13
0.0
0.18
0.0
0.02
0.0
0.18
0.0
0.01
```

In [33]:
```
y_pred_lr = lr_model.predict_proba(X1)[:,1]
for i in y_pred_lr[:20]:
    print(np.around(i,decimals=2))
```

```
0.04
0.17
0.0
0.04
0.01
0.01
0.01
0.02
0.0
0.02
0.02
0.19
0.0
0.01
0.02
0.0
0.0
0.0
0.03
0.03
```

In [34]:
```
y_pred_lr = lr_model.predict_proba(X1)[:,1]
y_pred_lr=y_pred_lr .round(2)
prob_df = pd.DataFrame(y_pred_lr)
prob_df
result = pd.merge(test_id,prob_df,left_index = True, right_index = True, how = 'left')
result.head()
datatoexcel = pd.ExcelWriter('submission.xlsx')
result.to_excel(datatoexcel)
datatoexcel.close()
```

In [36]:
```
pip install gradio
```

```
Requirement already satisfied: gradio in c:\users\dell e5490\anaconda3\lib\site-packages
(4.3.0)
Requirement already satisfied: jinja2<4.0 in c:\users\dell e5490\anaconda3\lib\site-pack
ages (from gradio) (2.11.3)
Requirement already satisfied: numpy~=1.0 in c:\users\dell e5490\anaconda3\lib\site-pack
ages (from gradio) (1.21.5)
Requirement already satisfied: importlib-resources<7.0,>=1.3 in c:\users\dell e5490\anac
onda3\lib\site-packages (from gradio) (6.1.1)
Requirement already satisfied: markupsafe~=2.0 in c:\users\dell e5490\anaconda3\lib\site
-packages (from gradio) (2.0.1)
Requirement already satisfied: typer[all]<1.0,>=0.9 in c:\users\dell e5490\anaconda3\lib
\site-packages (from gradio) (0.9.0)
Requirement already satisfied: semantic-version~=2.0 in c:\users\dell e5490\anaconda3\li
b\site-packages (from gradio) (2.10.0)
Requirement already satisfied: pyyaml<7.0,>=5.0 in c:\users\dell e5490\anaconda3\lib\sit
e-packages (from gradio) (6.0)
Requirement already satisfied: requests~=2.0 in c:\users\dell e5490\anaconda3\lib\site-p
ackages (from gradio) (2.27.1)
Requirement already satisfied: ffmpy in c:\users\dell e5490\anaconda3\lib\site-packages
(from gradio) (0.3.1)
Requirement already satisfied: httpx in c:\users\dell e5490\anaconda3\lib\site-packages
(from gradio) (0.25.1)
Requirement already satisfied: huggingface-hub>=0.14.0 in c:\users\dell e5490\anaconda3
\lib\site-packages (from gradio) (0.19.3)
Requirement already satisfied: packaging in c:\users\dell e5490\anaconda3\lib\site-packa
ges (from gradio) (21.3)
Requirement already satisfied: fastapi in c:\users\dell e5490\anaconda3\lib\site-package
s (from gradio) (0.104.1)
Requirement already satisfied: matplotlib~=3.0 in c:\users\dell e5490\anaconda3\lib\site
-packages (from gradio) (3.5.1)
Requirement already satisfied: pandas<3.0,>=1.0 in c:\users\dell e5490\anaconda3\lib\sit
e-packages (from gradio) (1.4.2)
Requirement already satisfied: uvicorn>=0.14.0 in c:\users\dell e5490\anaconda3\lib\site
-packages (from gradio) (0.24.0.post1)
Requirement already satisfied: pillow<11.0,>=8.0 in c:\users\dell e5490\anaconda3\lib\si
te-packages (from gradio) (9.0.1)
Requirement already satisfied: pydantic>=2.0 in c:\users\dell e5490\anaconda3\lib\site-p
ackages (from gradio) (2.5.1)
Requirement already satisfied: python-multipart in c:\users\dell e5490\anaconda3\lib\sit
e-packages (from gradio) (0.0.6)
Requirement already satisfied: tomlkit==0.12.0 in c:\users\dell e5490\anaconda3\lib\site
-packages (from gradio) (0.12.0)
Requirement already satisfied: gradio-client==0.7.0 in c:\users\dell e5490\anaconda3\lib
\site-packages (from gradio) (0.7.0)
Requirement already satisfied: typing-extensions~=4.0 in c:\users\dell e5490\anaconda3\l
ib\site-packages (from gradio) (4.8.0)
Requirement already satisfied: pydub in c:\users\dell e5490\anaconda3\lib\site-packages
(from gradio) (0.25.1)
Requirement already satisfied: altair<6.0,>=4.2.0 in c:\users\dell e5490\anaconda3\lib\s
ite-packages (from gradio) (5.1.2)
Requirement already satisfied: orjson~=3.0 in c:\users\dell e5490\anaconda3\lib\site-pac
kages (from gradio) (3.9.10)
Requirement already satisfied: aiofiles<24.0,>=22.0 in c:\users\dell e5490\anaconda3\lib
\site-packages (from gradio) (23.2.1)
Requirement already satisfied: fsspec in c:\users\dell e5490\anaconda3\lib\site-packages
(from gradio-client==0.7.0->gradio) (2023.10.0)
Requirement already satisfied: websockets<12.0,>=10.0 in c:\users\dell e5490\anaconda3\l
ib\site-packages (from gradio-client==0.7.0->gradio) (11.0.3)
Requirement already satisfied: jsonschema>=3.0 in c:\users\dell e5490\anaconda3\lib\site
-packages (from altair<6.0,>=4.2.0->gradio) (4.4.0)
Requirement already satisfied: toolz in c:\users\dell e5490\anaconda3\lib\site-packages
(from altair<6.0,>=4.2.0->gradio) (0.11.2)
Requirement already satisfied: tqdm>=4.42.1 in c:\users\dell e5490\anaconda3\lib\site-pa
ckages (from huggingface-hub>=0.14.0->gradio) (4.64.0)
```

```
Requirement already satisfied: filelock in c:\users\dell e5490\anaconda3\lib\site-packag
es (from huggingface-hub>=0.14.0->gradio) (3.6.0)
Requirement already satisfied: zipp>=3.1.0 in c:\users\dell e5490\anaconda3\lib\site-pac
kages (from importlib-resources<7.0,>=1.3->gradio) (3.7.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\users
\dell e5490\anaconda3\lib\site-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradi
o) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\dell e5490\anaconda3\lib\site-p
ackages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (21.4.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell e5490\anaconda3\lib\si
te-packages (from matplotlib~=3.0->gradio) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell e5490\anaconda3\lib
\site-packages (from matplotlib~=3.0->gradio) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\dell e5490\anaconda3\lib\site-pa
ckages (from matplotlib~=3.0->gradio) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell e5490\anaconda3\lib\si
te-packages (from matplotlib~=3.0->gradio) (1.3.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell e5490\anaconda3\lib\sit
e-packages (from matplotlib~=3.0->gradio) (3.0.4)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell e5490\anaconda3\lib\site-pa
ckages (from pandas<3.0,>=1.0->gradio) (2021.3)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\dell e5490\anaconda3\l
ib\site-packages (from pydantic>=2.0->gradio) (0.6.0)
Requirement already satisfied: pydantic-core==2.14.3 in c:\users\dell e5490\anaconda3\li
b\site-packages (from pydantic>=2.0->gradio) (2.14.3)
Requirement already satisfied: six>=1.5 in c:\users\dell e5490\anaconda3\lib\site-packag
es (from python-dateutil>=2.7->matplotlib~=3.0->gradio) (1.16.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell e5490\anaconda3\lib\s
ite-packages (from requests~=2.0->gradio) (2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dell e5490\anaconda3\li
b\site-packages (from requests~=2.0->gradio) (1.26.9)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\dell e5490\anaconda
3\lib\site-packages (from requests~=2.0->gradio) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell e5490\anaconda3\lib\site-pa
ckages (from requests~=2.0->gradio) (3.3)
Requirement already satisfied: colorama in c:\users\dell e5490\anaconda3\lib\site-packag
es (from tqdm>=4.42.1->huggingface-hub>=0.14.0->gradio) (0.4.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in c:\users\dell e5490\anaconda3\lib
\site-packages (from typer[all]<1.0,>=0.9->gradio) (8.0.4)
Requirement already satisfied: shellingham<2.0.0,>=1.3.0 in c:\users\dell e5490\anaconda
3\lib\site-packages (from typer[all]<1.0,>=0.9->gradio) (1.5.4)
Requirement already satisfied: rich<14.0.0,>=10.11.0 in c:\users\dell e5490\anaconda3\li
b\site-packages (from typer[all]<1.0,>=0.9->gradio) (13.7.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\dell e5490\anaconda3
\lib\site-packages (from rich<14.0.0,>=10.11.0->typer[all]<1.0,>=0.9->gradio) (2.16.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\dell e5490\anaconda3\li
b\site-packages (from rich<14.0.0,>=10.11.0->typer[all]<1.0,>=0.9->gradio) (3.0.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\dell e5490\anaconda3\lib\site-pack
ages (from markdown-it-py>=2.2.0->rich<14.0.0,>=10.11.0->typer[all]<1.0,>=0.9->gradio)
(0.1.2)
Requirement already satisfied: h11>=0.8 in c:\users\dell e5490\anaconda3\lib\site-packag
es (from uvicorn>=0.14.0->gradio) (0.14.0)
Requirement already satisfied: anyio<4.0.0,>=3.7.1 in c:\users\dell e5490\anaconda3\lib
\site-packages (from fastapi->gradio) (3.7.1)
Requirement already satisfied: starlette<0.28.0,>=0.27.0 in c:\users\dell e5490\anaconda
3\lib\site-packages (from fastapi->gradio) (0.27.0)
Requirement already satisfied: exceptiongroup in c:\users\dell e5490\anaconda3\lib\site-
packages (from anyio<4.0.0,>=3.7.1->fastapi->gradio) (1.1.3)
Requirement already satisfied: sniffio>=1.1 in c:\users\dell e5490\anaconda3\lib\site-pa
ckages (from anyio<4.0.0,>=3.7.1->fastapi->gradio) (1.2.0)
Requirement already satisfied: httpcore in c:\users\dell e5490\anaconda3\lib\site-packag
es (from httpx->gradio) (1.0.2)
Note: you may need to restart the kernel to use updated packages.
```

In [37]:
```
import gradio as gr
from gradio.components import *
```

In [38]:
```
def make_prediction( gender, age, hypertension, heart_disease,ever_married, work_type, R
       smoking_status):
    # Create input dataframe
    input_df = pd.DataFrame([[gender, age, hypertension, heart_disease, ever_married, wo
smoking_status]],columns=[ 'gender','age', 'hypertension', 'heart_disease', 'ever_marrie
       'smoking_status'])
    lr_pred = lr_model.predict_proba(input_df)[:, 1][0]
    return {'Logistic Regression': lr_pred}
# Define input components for Gradio interface
gender = gr.components.Number(label='Gender')
age = gr.components.Number(label='Age')
hypertension = gr.components.Number(label='Hypertension')
heart_disease = gr.components.Number(label='Heart Disease')
ever_married=gr.components.Number(label='Ever_married')
work_type=gr.components.Number(label='Work_type')
Residence_type=gr.components.Number(label='Residence_type')
avg_glucose_level = gr.components.Number(label='Average Glucose Level')
bmi = gr.components.Number(label='BMI')
smoking_status=gr.components.Number(label='Smoking_status')

css_code='''
        .gradio-container
        {
            background-image: url("C:\\Users\\DELL E5490\\Pictures\\Saved Pictures\\cert
            height: 100%;
            background-position: center;
            background-repeat: no-repeat;
            background-size: cover;
        }
        '''
# Create Gradio interface
gr.Interface(fn=make_prediction,
             inputs=[gender,age, hypertension, heart_disease,  ever_married, work_type,
             outputs='label',
             title='Stroke Prediction',
             description='Predict the probability of stroke ',
             css=css_code
            ).launch()
```

```
Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
```

Out[38]:

In [39]:
```
test_data.head()
```

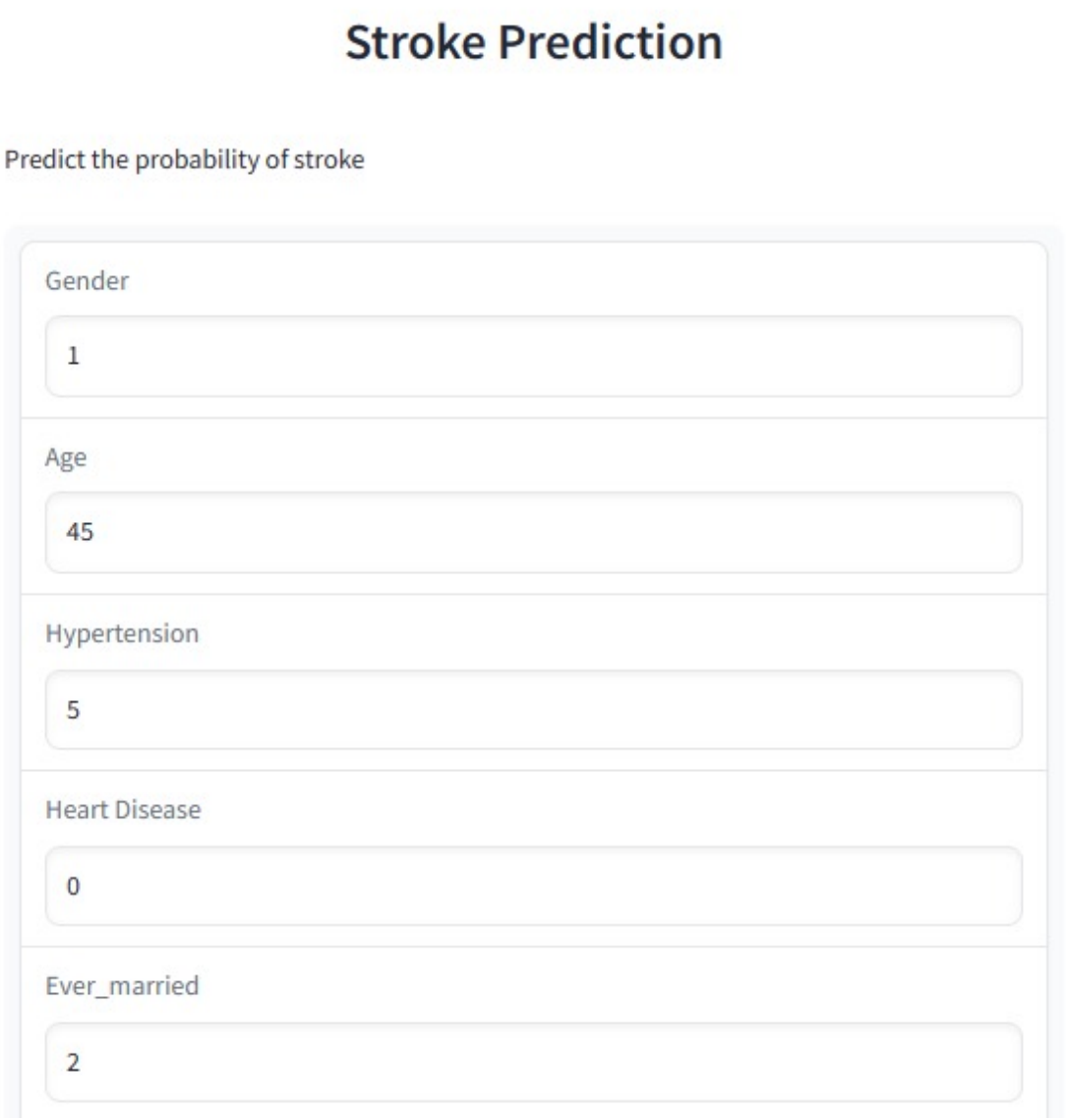
Out[39]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_lev
0	15304	1	57.0	0	0	0	0	1	82.5
1	15305	0	70.0	1	0	0	0	0	72.(
2	15306	1	5.0	0	0	1	3	0	103.7
3	15307	1	56.0	0	0	0	2	0	69.2
4	15308	0	32.0	0	0	0	0	1	111.3

In [ ]:

In [ ]:

6.2 Input


Stroke Prediction
Predict the probability of stroke
Gender
1
Age
45
Hypertension
5
Heart Disease
0
Ever_married
2

Work_type

4

Residence_type

1

Average Glucose Level

72

BMI

87

Smoking_status

1

Clear

Submit

6.3 Output

6.4 Testing

We all have to agree that in today's ever-changing and competitive world, the internet has become an integral part of our lives. Most of us make our decisions by searching the information on the internet these days, hence hosting a website is no[9] longer optional but mandatory for all kind of businesses. It is the first step in becoming and staying relevant in the market.

Just having a website is not enough. An organization is needed to develop a website that is informative, accessible and user-friendly. To maintain all these qualities, the website should be well tested, and this process of testing a website is known as web testing.

What Is Web Testing?
Web testing is a software testing practice to test websites or web applications for potential bugs. It's a complete testing of web-based applications before making live. A web-based system needs to be checked completely from end-to-end before it goes live for end users.By performing website testing, an organization can make sure that the web-based system is functioning properly and can be accepted by real-time users. The UI design and functionality are the captains of website testing.

Web Testing Checklists
1) Functionality testing
2) Usability testing
3) Interface testing
4) Security testing

6.4.1 Functionality Testing

Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages.

Check all the links:

- Test the outgoing links from all the pages to the specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send email to admin or other users from web pages.

- Test to check if there are any orphan pages.
- Finally, link checking includes, check for broken links in all the above-mentioned links.

Test forms on all pages:

Forms are an integral part of any website. Forms are used for receiving information from users and to interact with them. So what should be checked in these forms?

- First, check all the validations on each field.
- Check for default values of the fields.
- Wrong inputs in the forms to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.

Validate your HTML/CSS:

If you are optimizing your site for Search engines then HTML/CSS validation is the most important one. Mainly validate the site for HTML syntax errors.

Database Testing:

Data consistency is also very important in a web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality. Check if all the database queries are executing correctly, data is retrieved and also updated correctly.

6.4.2 Usability Testing

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction. Usability Testing includes the following:

- The website should be easy to use.
- The instructions provided should be very clear.
- Check if the instructions provided are perfect to satisfy its purpose.
- The main menu should be provided on each page.
- It should be consistent enough.

6.4.3 Interface Testing

In web testing, the server-side interface should be tested. This is done by verifying that communication is done properly. Compatibility of the server with software, hardware, network, and the database should be tested.
The main interfaces are:

- Web server and application server interface
- Application server and Database server interface.

Check if all the interactions between these servers are executed and errors are handled properly. If the database or web server returns an error message for any query by application server then the application server should catch and display these error messages appropriately to the users.

6.4.4 Security Testing

The primary reason for testing the security of a web is to identify potential vulnerabilities and subsequently repair them.
Following are some of the test cases for web security testing:

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- Try some invalid inputs in input fields like login username, password, input text boxes, etc. Check the system's reaction to all invalid inputs.

CONCLUSION

This project Heart Stroke prediction,using machine learning is very much useful in everyone's day to day life and it is mainly more important for the healthcare sector, because they are the one that daily uses these systems to predict the diseases of the patients based on their general information and their symptoms that they are been through. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. If health industry adopts this project, then the work of the doctors can be reduced and they can easily predict the disease of the patient. The Disease prediction is to provide prediction for the various and generally occurring diseases that when unchecked and sometimes ignored can turns into fatal disease and cause lot of problem to the patient and as well as their family members

REFERENCES

[1] *Prediction of Heart Stroke Using Support Vector Machine Algorithm:*

[2] *Heart Stroke Prediction Using Machine Learning Models:*

[3] *An enhanced approach for analyzing the performance of heart stroke prediction with machine learning techniques:*

[4] *Heart Disease and Stroke Statistics - 2020 Update:*

[5] *Deep Learning for Stroke Prediction:*

[6] *Stroke Risk Prediction Using Electronic Health Records and Machine Learning:*

[7] *A Comprehensive Review of Stroke Prediction Models:*