

A Major Project Report

On

AI-Based Predictive Tools for Stroke Risk Evaluation and Prevention

Team Members:

- **VINAY SANGAM – Y00861380**
- **SANKAR GUNTUPALLI - Y00861407**
- **VENKATA SHIVA RAMAKRISHNA KURRARAPU - Y00867657**

Under the guidance of

Dr. Robert A. Gilliland

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE

YOUNGSTOWN STATE UNIVERSITY

ragilliland@ysu.edu

Ph. No - (330) 941-2808

Abstract

Stroke, a global health challenge with high rates of morbidity and mortality, is the focus of our innovative project. Traditional risk assessment models, which rely on static parameters and limited datasets, often lead to delayed diagnosis and ineffective prevention. Our project aims to revolutionize stroke risk evaluation by developing AI-based predictive tools that use machine learning to provide more accurate and individualized risk assessments, a unique approach in the field.

The project starts by evaluating current stroke prediction methodologies to identify their limitations, such as inadequate use of dynamic and extensive datasets. By integrating AI, particularly advanced machine learning techniques, we address these shortcomings through comprehensive analysis of diverse data sources, including electronic health records (EHRs), patient surveys, and data from wearable devices.

Central to our methodology is the collection and preprocessing of high-quality, multidimensional datasets. We utilize sophisticated data preprocessing and feature engineering techniques to enhance the performance and accuracy of our predictive models. Various machine learning algorithms, such as logistic regression, decision trees, random forests, support vector machines (SVMs), and deep learning neural networks, are employed and assessed for their predictive capabilities.

Our project places a strong emphasis on the practical applicability of the AI tools we develop. We are committed to integrating these tools into clinical practice, ensuring they are not just theoretical advancements but practical solutions. We are developing intuitive decision support systems that seamlessly integrate with existing EHR systems, enabling real-time stroke risk assessments and providing actionable insights for healthcare professionals.

The predictive models are validated through rigorous case studies and clinical trials, assessing their performance across different patient demographics and healthcare settings. Feedback from healthcare providers is used to refine and enhance the tools' usability.

Ultimately, this project seeks to improve stroke prevention and patient outcomes by offering advanced, personalized risk assessment solutions. By overcoming the limitations of traditional models, our AI-based tools aim to significantly advance stroke risk evaluation and prevention, contributing to more effective healthcare interventions and better public health outcomes.

The outcomes of this project are expected to significantly enhance stroke risk evaluation and prevention strategies. By harnessing the power of AI and machine learning, our tools aim to provide more accurate, individualized risk assessments and support proactive healthcare interventions. This approach aims to reduce the incidence of stroke and improve patient outcomes through timely and personalized preventive measures.

In conclusion, our project represents a significant advancement in stroke prevention. By leveraging sophisticated machine learning techniques, we aim to overcome the limitations of traditional models and provide a more effective framework for stroke risk evaluation and prevention. Ultimately, our goal is to contribute to better public health outcomes through the development and implementation of AI-based predictive tools.

This abstract summarizes the project's goals, methods, and expected impacts, highlighting the integration of AI and machine learning in stroke prediction and prevention.

Introduction

The rapid increase in global population, shifting societal lifestyles, the emergence of new diseases, recurring pandemics, and environmental changes have all contributed to a paradigm shift in how healthcare is approached and managed. As these factors continue to evolve, they present both opportunities and challenges in maintaining public health. The growing demands on healthcare systems across the world underscore the need for robust medical services that ensure the well-being of populations. Good health is integral to an individual's productivity, quality of life, and overall social and economic contribution, making healthcare one of the primary concerns for governments worldwide.

However, providing adequate medical services to populations, especially in developing and low-income countries, presents significant challenges. The disparity between healthcare resources and the demand for them has led to varying levels of healthcare access and quality across different regions. One of the most critical aspects that determine the strength and efficiency of any healthcare system is the ratio of healthcare workers—specifically doctors—to the population. This ratio serves as a key indicator of the capacity of a healthcare system to meet the needs of its citizens.

In developed nations such as the United States and countries across Western Europe, this ratio is relatively favorable. For instance, the USA and several European countries boast a ratio of approximately 25 doctors per 10,000 people. This higher density of healthcare professionals ensures that medical services are accessible, timely, and effective, thereby contributing to overall better healthcare outcomes.

In contrast, many developing countries, particularly those in Southeast Asia, face challenges related to a shortage of healthcare professionals, which affects the quality of care available to their populations. India, for example, has a ratio of just 12.21 doctors per 10,000 people, which is significantly lower than the world average. For comparison, the global average stands at around 10 doctors per 1,000 people. This shortage of healthcare professionals in India results in overcrowded medical facilities, long waiting times for patients, and challenges in delivering quality care, particularly in rural areas. Such disparities in healthcare resources are further

exacerbated by the increasing burden of diseases, limited access to medical infrastructure, and financial constraints.

The healthcare workforce shortage in many developing nations is compounded by several factors, including inadequate medical education infrastructure, migration of healthcare professionals to developed countries in search of better opportunities, and insufficient funding for the healthcare sector. As a result, these nations face significant barriers in achieving Universal Health Coverage (UHC) and addressing the healthcare needs of their populations.

This imbalance in healthcare worker availability between developed and developing countries has profound implications not only on the quality of healthcare services but also on health outcomes. Countries with a higher density of healthcare workers tend to have better health indicators, such as lower mortality rates, higher life expectancy, and reduced incidence of preventable diseases.

To address these challenges, there is an urgent need for policies and initiatives that aim to bridge the healthcare workforce gap. This includes investing in healthcare education and training, improving working conditions for healthcare workers, and encouraging the retention of medical professionals in underserved regions. Moreover, leveraging technology, such as telemedicine and artificial intelligence (AI), can help overcome some of the challenges posed by a shortage of healthcare workers, enabling more efficient and widespread healthcare delivery.

In conclusion, while the healthcare systems in developed countries benefit from a higher ratio of healthcare professionals to the population, developing nations, particularly in Southeast Asia, face significant shortages that impact the quality of care. Addressing these disparities is essential for improving health outcomes globally and ensuring that healthcare services are accessible to all, regardless of geographic or economic barriers.

Medical doctors (per 10,000)

EXPORT DATA in CSV format: Right-click here & Save link					
Last updated: 2022-01-24					
Location	Medical doctors (per 10,000)	Medical doctors (number)	Generalist medical practitioners (number)	Specialist medical practitioners (number)	Medical doctors (per 10,000)
Tajikistan	2.24	12,713	17,109	15,492	12,713
Timor-Leste	6.17	712	345	33	367
Togo	0.45	244	219	264	244
Tonga	3.03	100	39	12	30
Trinidad and Tobago	10.06	1,038	4,695	1,098	1,038
Tunisia	10.24	10,554	6,491	7,755	10,554
Turkey	10.22	100,853	10,542	101,998	10,085
Turkmenistan	22.13	11,365	4,156	6,851	20,031
Tuvalu	10.31	10	11	3	0
Uganda	0.82	17,186	17,007	179	2,209
Ukraine	29.92	134,986	12,995	118,737	134,986
United Arab Emirates	13.3	11,630			11,630
United Kingdom of Great Britain and Northern Ireland	16.2	101,803	29,220	101,214	185,681
United Republic of Tanzania	0.23	1,481	2,434	451	1,481
United States of America	24.39	525,070			525,070
Uruguay	37.23	12,384	5,021	8,524	12,384
Uzbekistan	23.74	65,805	10,965	57,279	65,805
Vanuatu	1.14	20			20
Venezuela (Bolivarian Republic of)	17.3	48,000			48,000
Viet Nam	5.24	42,327			42,327
Yemen	2.39	13,560	5,412		3,814
Zambia	0.53	1,499	1,701	325	1,499
Zimbabwe	0.54	1,054	1,002	227	1,054

Fig 1.1 Ratio of Doctors per 10000 population in the USA

Medical doctors (per 10,000)

EXPORT DATA in CSV format: Right-click here & Save link					
Last updated: 2022-01-24					
Location	Medical doctors (per 10,000)	Medical doctors (number)	Generalist medical practitioners (number)	Specialist medical practitioners (number)	Medical doctors (per 10,000)
Guinea	0.83	1,844	2,649	138	1,844
Guinea-Bissau	0.54	100	197	10	0
Guyana	14.24	1,120			1,120
Haiti	0.85	1,392			1,392
Honduras	2.95	2,454	1,241	1,213	2,680
Hungary	26.8	20,877	6,855	25,615	15,351
Iceland	28.47	1,029	156	692	1,029
India	12.21	1,014,538			1,014,538
Indonesia	0.58	11,067	137,920	16,597	0
Iran (Islamic Republic of)	11.29	116,536	51,974	39,127	116,536
Iraq	10.36	19,738	24,586	12,807	1,472
Ireland	15.52	10,270	10,781	2,524	10,270
Israel	32.31	10,140	8,609	13,856	10,140
Italy	34.59	148,101	48,230	147,845	148,101
Jamaica	3.54	1,006	1,044	280	1,103
Japan	16.39	113,214			101,011
Jordan	18.69	10,627			10,627
Kazakhstan	32.8	45,400	1,818	42,480	69,721
Kenya	1.34	4,506	5,602	2,440	4,506
Kiribati	1.43	15			15
Kuwait	14.54	10,000			10,000
Kyrgyzstan	22.13	10,609			10,609
Lao People's Democratic Republic	1.8	1,160			1,160

Fig 1.2 Ratio of Doctors per 10000 population in India

Increasing the ratio of doctors to patients is a complex and time-consuming process, requiring years or even decades to achieve. Additionally, the construction of modern healthcare facilities equipped with advanced medical equipment is essential. Another common issue is that people often avoid seeing doctors during the early stages of diseases or health risks, which can lead to delayed diagnoses and more serious conditions.

To address this, developing a suitable system that allows individuals to pre-screen their health could be highly beneficial. A fundamental solution is to design a scalable system, where scalability refers to the ability of the technological infrastructure to expand the reach of healthcare services without the need for an increased workforce. This can be achieved by leveraging software services.

Software-based solutions offer several advantages, including the ability to detect diseases at early stages, improve the speed and accuracy of medical diagnostics, and provide high-quality care. Furthermore, such systems encourage greater patient participation in managing their health and increase the likelihood of safer and more effective treatment options.

1.2 Medical Records

Patient records are essential for effective healthcare. Healthcare professionals can only recommend tests or treatments after thoroughly analyzing an individual's medical history. However, obtaining results from medical tests can be expensive, time-consuming, and financially burdensome, especially in developing or low-income countries. These costs, coupled with the time required for testing, can deter people from seeking necessary tests. Additionally, the medical system faces added strain as resources are used for everything from sample collection to operating machinery.

One of the complications is that individuals who are not experiencing symptoms or affected by diseases often undergo tests merely for reassurance, even though they will not require treatment. While these individuals will not need further medical intervention, the tests they undergo, the time they spend, and the resources they consume ultimately do not contribute to those who are genuinely in need of care.

As a result, resources that could be allocated to those who truly need them are wasted. In many cases, the process of identifying diseases becomes a bottleneck within the healthcare system.

1.3 Heart Disease

Heart disease, also known as cardiovascular disease (CVD), is a broad term that encompasses a variety of heart-related ailments and conditions. These conditions can affect the heart's structure

and function, leading to serious health complications and, in many cases, death. Heart disease can manifest in several ways, including heart muscle disease (cardiomyopathy), heart attacks, coronary artery diseases, arrhythmias (irregular heartbeats), and congenital heart defects. While these conditions can present in different forms, there are common underlying risk factors that influence an individual's susceptibility to heart disease.

Common Types of Heart Disease:

1. **Cardiomyopathy:** A condition in which the heart muscle becomes weakened or enlarged, affecting the heart's ability to pump blood efficiently.
2. **Heart Attacks (Myocardial Infarction):** A condition where the flow of oxygenated blood to a part of the heart muscle is blocked, causing tissue damage.
3. **Coronary Artery Disease (CAD):** A narrowing or blockage of the coronary arteries, typically caused by atherosclerosis, which can lead to chest pain, heart attacks, or strokes.
4. **Arrhythmia:** Irregular heartbeats that can lead to complications like stroke or sudden cardiac arrest if not treated properly.
5. **Congenital Heart Defects:** Structural abnormalities in the heart that are present from birth.

Symptoms of Heart Disease

Heart disease can present a range of symptoms, which vary depending on the type of condition. Some of the most common symptoms associated with heart disease include:

- **Chest Pain (Angina):** This includes chest tightness, pressure, discomfort, or pain, often felt during physical exertion or stress. Angina occurs when the heart muscle is not getting enough oxygenated blood.
- **Shortness of Breath:** Difficulty breathing or feeling winded even during mild physical activity is a common sign of heart disease, indicating the heart's diminished ability to pump oxygen-rich blood throughout the body.
- **Pain, Numbness, or Coldness in Limbs:** When the blood vessels in the arms or legs become narrowed due to atherosclerosis or peripheral artery disease, symptoms like pain, numbness, weakness, or coldness in the limbs may occur.

- **Infants' Shortness of Breath:** Infants suffering from heart disease may exhibit shortness of breath during feeding, leading to poor weight gain and other health complications.
- **Discomfort in the Neck, Throat, or Abdomen:** Severe discomfort in these areas, especially if persistent, can be indicative of heart disease.
- **Obesity and High Cholesterol:** Excess fat accumulation in the body, high blood pressure, and elevated cholesterol levels are common risk factors for heart disease.

These symptoms often correlate strongly with an increased risk of heart disease, and they suggest the need for early medical intervention. When combined with lifestyle factors, such as poor diet, lack of exercise, and insufficient sleep, the risk of developing heart disease can be significantly higher. Many of these conditions are predictable and can be managed with lifestyle changes, such as regular exercise, healthy eating, and adequate rest.

Prevention and Lifestyle Factors

Maintaining a healthy lifestyle is one of the most effective ways to minimize the risk of heart disease. The primary risk factors for heart disease—such as obesity, high cholesterol, and high blood pressure—are largely influenced by lifestyle choices. Regular physical activity, a balanced diet rich in fruits, vegetables, and whole grains, and sufficient sleep can help manage these risk factors. Additionally, regular health check-ups to monitor blood pressure, cholesterol levels, and weight are crucial for early identification of potential problems.

Global Impact of Heart Disease

Heart disease is a leading cause of death worldwide, with the World Health Organization (WHO) reporting that approximately **17.9 million people died from cardiovascular diseases (CVDs) in 2016**, accounting for **31% of all global deaths**. Heart attacks and strokes were responsible for **four out of every five deaths** attributed to CVDs, with more than **one-third of these deaths occurring in individuals under the age of 70**. The statistics underscore the significant public health burden of heart disease and the urgent need for prevention and early detection strategies.

In **India**, cardiovascular diseases are responsible for **27% of all deaths**, with noncommunicable diseases (NCDs) accounting for **63% of total deaths**. CVDs contribute to nearly **45% of deaths**

in the age group of **40 to 69 years**. This reflects a growing health crisis in the country, where urbanization, poor diet, lack of physical activity, and high stress levels are contributing to the increasing prevalence of heart disease.

Economic Burden of Heart Disease

The financial impact of heart disease is enormous. In **the United States**, for example, heart disease costs approximately **\$363 billion annually** (for the years 2016-2017) in terms of healthcare expenses and lost productivity. This includes costs for medical treatments, hospitalizations, medications, and outpatient care.

In **developing and low-income countries**, the situation is even more dire. Cardiovascular diseases are responsible for at least **three-quarters of all deaths** in these regions. Many individuals living in poverty-stricken areas lack access to adequate healthcare services, including essential diagnostic tools, medications, and preventive care programs. This results in a delayed diagnosis, with many people being unaware of their risk factors until they develop serious complications, often too late to reverse the damage.

Healthcare Access and Disparities

In **low-income countries**, access to primary healthcare is often limited due to factors such as insufficient medical infrastructure, inadequate resources, and a lack of trained healthcare personnel. Early detection and intervention, which are crucial in managing heart disease, are typically not available. This results in a high incidence of late-stage diagnoses, leading to higher mortality rates from cardiovascular diseases.

The **poorest populations** in low- and middle-income countries bear the brunt of the consequences of heart disease. These individuals often face catastrophic health expenditures, which drain their financial resources and exacerbate poverty. In many cases, families are forced to make difficult choices between seeking medical treatment and meeting basic needs. This vicious cycle of poor health and economic hardship contributes to the broader issue of healthcare inequality.

From a **macro-economic** perspective, CVDs and other noncommunicable diseases also significantly damage the economies of low- and middle-income countries. High healthcare costs and lost productivity due to premature deaths or disability further strain these economies. With limited resources to address this growing health crisis, the gap between developed and developing nations in terms of healthcare access continues to widen.

The Path Forward: Improving Healthcare Access and Prevention

Addressing the burden of heart disease in developing countries requires a multifaceted approach, including:

1. **Improved Healthcare Infrastructure:** Building healthcare systems capable of providing accessible and timely diagnosis and treatment for heart disease.
2. **Public Health Campaigns:** Promoting awareness about the risk factors for heart disease, such as smoking, poor diet, lack of exercise, and stress, can help reduce the overall incidence of CVDs.
3. **Early Detection Programs:** Implementing affordable and accessible screening programs to identify individuals at risk of heart disease early, even in rural and remote areas.
4. **Affordable Treatments:** Ensuring that heart disease treatments, medications, and surgeries are affordable for low-income populations, potentially through government subsidies or international aid.
5. **Healthcare Workforce Training:** Expanding the number of healthcare professionals trained to identify, manage, and treat cardiovascular diseases is crucial for better health outcomes.

In conclusion, heart disease remains a global health challenge, with substantial economic and social implications. While the incidence of CVDs is high in both developed and developing countries, the lack of healthcare access and early detection in low-income nations exacerbates the problem. A combination of improved healthcare infrastructure, prevention strategies, and affordable treatment options is essential for addressing the growing burden of heart disease worldwide.

1.4 Project Description

The increasing demand for healthcare services, coupled with the shortage of medical professionals, especially in low-income and developing countries, presents a significant challenge for the timely diagnosis and treatment of various health conditions, including heart disease. Heart disease is one of the leading causes of death globally, and early detection is crucial for preventing complications and improving patient outcomes. The complexity of diagnosing heart disease, combined with the limitations in healthcare infrastructure, has prompted the need for more efficient, accessible, and affordable diagnostic solutions. This project is developed with the objective of addressing these issues by leveraging machine learning (ML) to predict the presence or absence of heart disease based on key medical attributes.

Project Objective

The primary objective of this project is to develop a predictive system that can accurately determine whether an individual is at risk of heart disease based on certain medical and lifestyle attributes. By utilizing machine learning algorithms, the project aims to provide a reliable, user-friendly tool that can assist individuals in identifying their risk level without the need for costly or time-consuming tests. The system is designed to provide an initial diagnostic recommendation based on the user's input, which is particularly useful in scenarios where immediate access to healthcare professionals is not possible.

Key Features and Approach

1. Predictive Machine Learning Model:

The core of this project is a machine learning model that predicts the likelihood of heart disease based on various medical attributes. These attributes may include factors such as age, gender, blood pressure, cholesterol levels, blood sugar levels, physical activity, smoking habits, and family medical history. The model is trained using a dataset that contains these variables, with the aim of accurately predicting whether an individual is at risk of heart disease.

2. User Interactive Interface:

To make the model accessible to a wider audience, a simple, user-friendly frontend interface is developed using **Flask**, a lightweight Python web framework. The frontend

allows users to input their medical information, such as age, cholesterol levels, blood pressure, and other relevant details, into the system. Based on these inputs, the machine learning model processes the data and provides an output that indicates whether the user is likely to have heart disease or not.

The user interface is designed to be intuitive, so that individuals with no medical background can easily navigate through the system and understand their health status. It includes options to enter personal information and displays the prediction results in a clear and understandable manner.

3. Machine Learning Algorithms:

The system is built using various machine learning algorithms, each of which is evaluated for accuracy and performance. Several common algorithms are used for training the model, including:

- **Logistic Regression:** A simple and effective algorithm for binary classification problems, such as predicting whether someone has heart disease or not.
- **k-Nearest Neighbors (KNN):** A non-parametric algorithm that classifies a data point based on the majority class of its neighbors.
- **Support Vector Machine (SVM):** A powerful algorithm that finds the optimal hyperplane that best separates the data points into different classes.
- **Decision Trees:** A tree-like model that makes decisions based on a series of questions or attributes.
- **Random Forest:** An ensemble learning method that combines multiple decision trees to improve prediction accuracy.

These algorithms are trained on historical medical data to develop a model that can make accurate predictions based on the user's input. The algorithms are evaluated on their ability to correctly classify heart disease cases and the accuracy of their predictions.

4. Model Evaluation and Optimization:

Each algorithm is assessed using various evaluation metrics such as **accuracy**, **precision**, **recall**, and **F1-score** to determine the most effective model for predicting heart disease.

The accuracy of the models is tested using cross-validation, where the dataset is split into

training and testing sets. This helps ensure that the model is not overfitting to the data and can generalize well to new, unseen data.

After evaluating the models, the one with the highest accuracy and performance is selected for integration into the final application. The model is then fine-tuned, and optimization techniques are applied to improve its predictive capabilities further. The goal is to develop a model that provides reliable and consistent predictions for heart disease risk.

5. **Integration with Frontend:**

After selecting the most accurate machine learning model, it is integrated with the frontend using **Flask**, which allows the model to interact with users via a web interface. The frontend allows users to input their medical details into forms, and once submitted, the backend processes the data using the trained model. The result is displayed in a format that is easy for users to understand, informing them of their heart disease risk.

Flask serves as the server-side component, handling user input, running predictions, and returning results. The system is scalable, meaning it can handle multiple user requests simultaneously, ensuring that it remains functional even with an increasing number of users.

Model Training and Testing

The machine learning model is trained using a dataset that includes historical data on heart disease patients. The dataset contains attributes such as:

- **Age:** The age of the individual.
- **Sex:** Gender (Male or Female).
- **Blood Pressure:** High blood pressure can be a risk factor for heart disease.
- **Cholesterol Levels:** High cholesterol is another significant risk factor.
- **Blood Sugar Levels:** Elevated blood sugar is often associated with diabetes, which increases the risk of heart disease.
- **Smoking Habits:** Smoking is a major risk factor for heart disease.
- **Physical Activity:** Regular exercise can reduce the risk of heart disease.
- **Family History:** A family history of heart disease can increase an individual's risk.

These attributes are used as features to train the machine learning models, with the target variable being whether the individual has heart disease (0 for no, 1 for yes).

The models are evaluated using the **confusion matrix** and various performance metrics, such as **accuracy**, **precision**, and **recall**. This allows for an in-depth understanding of how well the model is performing and where improvements may be needed.

Results and Accuracy

Once the models are trained and tested, the performance of each algorithm is measured. For example:

- **Logistic Regression:** A simple, interpretable model that provides good results for binary classification tasks. It may not perform as well as more complex models but is useful for baseline comparisons.
- **KNN:** KNN's performance depends on the choice of distance metric and the number of neighbors used. It is typically slower for large datasets but can provide accurate results for smaller datasets.
- **SVM:** SVMs are highly effective in high-dimensional spaces and for classification problems with complex boundaries. The performance of SVM may be enhanced with kernel tricks and regularization.
- **Decision Trees:** This algorithm is highly interpretable and easy to visualize, but it can suffer from overfitting. Random Forests are used to mitigate this issue.
- **Random Forest:** An ensemble of decision trees that improves accuracy by reducing overfitting. This is often one of the most effective algorithms for heart disease prediction.

By comparing the accuracy of these models, the project selects the most suitable model for heart disease prediction and optimizes it for deployment.

Literature Review

Sushmita Roy Tithi et al. discussed ECG data analysis and heart disease prediction using machine learning algorithms in their study. [6]

In this paper, they employed six supervised machine learning algorithms to differentiate between normal and abnormal ECG patterns, with the goal of identifying specific heart diseases. They divided their dataset into two parts: 75% for training and 25% for testing. The algorithms used in their study included Logistic Regression, Decision Tree, k-Nearest Neighbors (KNN), Naïve Bayes, Support Vector Machine (SVM), and Artificial Neural Networks (ANN). The study aimed to classify various heart conditions such as Right Bundle Branch Block, Myocardial Infarction, Sinus Tachycardia, Sinus Bradycardia, and coronary artery disease, based on abnormal ECG signals.

ECG, or Electrocardiography, provides a series of sinus rhythm readings that reflect the condition of the heart, making it a critical tool for detecting a range of cardiac diseases.

Disease Name	Best Algorithm	Score
CAD	Naïve Bayes	94%
Sinus Bradycardia	Decision Tree	95%
Sinus tachycardia	All except NN	95%
Myocardial infarction	Decision Tree	96%

Bo Jin et al. discussed predicting the risk of heart failure using electronic health record (EHR) sequential data modeling. [7]

The aim of their study was to offer early diagnosis and treatment options for heart failure patients. Heart failure is increasingly common among individuals aged 65 and older, overweight people, and those with a history of heart attacks. In their paper, they developed an innovative approach to heart failure prediction using enhanced Long Short-Term Memory (LSTM) networks and a data-driven framework. They proposed a novel event modeling method that includes one-hot encoding and word vectors, applying the LSTM approach for analysis. The study used real-world EHR data from patients with congestive heart disease. The dataset was divided into two parts: Dataset A, which contained diagnostic records of 5,000 patients diagnosed with heart failure, and Dataset B, which included diagnostic records for 15,000 patients who had not been diagnosed with heart failure.

Yar Muhammad et al. discussed the early and accurate detection and diagnosis of heart disease using an intelligent computational model. [8]

Their study focused on early detection and diagnosis of heart disease through advanced computational techniques. The researchers utilized two heart disease datasets: the Cleveland dataset (S1) and the Hungarian dataset (S2). They applied ten classification algorithms, including KNN, Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), AdaBoost (AB), Extra Trees (ET), Gradient Boosting (GB), Logistic Regression (LR), and Artificial Neural Networks (ANN). Additionally, four feature selection algorithms—FCBF, mRMR, LASSO, and Relief—were used to improve the model's performance. The top two performing classification algorithms were Extra Trees (ET) with an accuracy of 92.09% and Gradient Boosting (GB) with 91.34%. The ET classifier, when combined with the Relief feature selection algorithm, provided the best performance.

Ashir Javeed et al. discussed an intelligent learning system based on the Random Search Algorithm and Optimized Random Forest Model to improve heart disease detection. [9]

They developed a learning system using the Random Search Algorithm for feature selection and the Random Forest model for diagnosing cardiovascular disease. The paper addresses the issue of overfitting in models and proposes a novel system combining these two algorithms. By doing so, they were able to improve the performance of the Random Forest model by 3.3%.

In our project, we are using 13 important attributes to predict the presence of heart disease:

1. **Age:** The risk of heart disease increases significantly after the age of 60, with over 80% of heart disease deaths occurring in people over 65 years old. Therefore, age is an important factor in predicting heart disease.
2. **Sex:** Cardiovascular risks are more pronounced in women with clinically manifest heart disease than in men. Women who smoke are more likely to have their first heart attack compared to men.
3. **Resting Blood Pressure (trestbps):** High resting blood pressure is a significant factor contributing to the risk of heart disease. It is important to monitor and control it for early detection.
4. **Angina:** This is chest pain caused by reduced blood flow to the heart, often linked to coronary artery disease. It can be induced during exercise and is an important indicator of heart disease risk.
5. **ST-segment Slope (slope):** This refers to the pattern of the ST-segment in an ECG reading, which can help detect heart disease. The slope can be upward, flat, or downward.
6. **Resting ECG (aberration in ST/T-wave):** The heart's electrical impulses may be interfered with, causing abnormal ECG readings. This can be a key indicator for heart disease.
7. **Classification of Chest Pain (cp):** Chest pain can be a typical or atypical symptom of heart disease. This classification helps determine the risk of heart-related complications.
8. **Fasting Blood Sugar (fbs):** High fasting blood sugar levels, particularly above 120 mg/dL, are associated with an increased risk of heart disease, including myocardial infarction and stroke.

9. **Fluoroscopy Colored Vessels (ca):** This refers to the number of major vessels colored by fluoroscopy in a heart exam. It helps assess the extent of blockage or damage to the heart's arteries.
10. **Serum Cholesterol (chol):** High cholesterol levels are directly linked to coronary heart disease. Monitoring cholesterol is crucial for predicting heart disease.
11. **Thalassemia (thal):** This inherited blood disorder leads to less hemoglobin, which can cause fatigue and is linked to heart disease risk.
12. **Max Heart Rate Achieved (thalach):** The maximum heart rate reached during exercise is inversely related to heart disease risk. A higher max heart rate typically indicates better heart fitness.
13. **Exercise-induced ST Depression (old peak):** ST depression observed during exercise-induced ECG can indicate heart disease. A depression greater than 1.0 mm is considered abnormal and linked to heart problems.

These 13 attributes are used to build a predictive model that can help in early detection of heart disease. By analyzing these factors, we aim to improve the accuracy and reliability of heart disease predictions.

Methodology

In this section, we discuss the process of building machine learning models to predict the presence or absence of heart disease based on the selected features. Machine learning is a powerful tool for creating predictive models by learning patterns in data and using those patterns to make predictions. The process involves data collection, preprocessing, model selection, training, and evaluation.

3.1.1 Data Collection

The first step in building machine learning models is collecting a comprehensive dataset that contains relevant attributes for heart disease prediction. In our case, we use a dataset that includes 13 medical attributes such as age, sex, blood pressure, cholesterol levels, ECG results, chest pain type, and other vital parameters. The dataset can be collected from various sources such as:

- **Public datasets:** Many public heart disease datasets are available, such as the Cleveland Heart Disease dataset or the Framingham Heart Study dataset, which have been widely used in medical research.
- **Electronic Health Records (EHR):** Real-world patient data from hospitals or clinics can be used for more accurate predictions, but this requires ensuring privacy and security.

For this project, we assume that we are working with a dataset like the Cleveland dataset, which includes both positive and negative instances of heart disease, making it suitable for training and testing machine learning models.

3.1.2 Data Preprocessing

Data preprocessing is an essential step in building machine learning models, as raw data often contains inconsistencies, missing values, and other issues that may affect model performance. Some common preprocessing steps include:

1. **Handling Missing Data:** Missing values in the dataset can lead to inaccurate predictions. Common strategies to handle missing data include:
 - **Imputation:** Filling in missing values with the mean, median, or mode of the respective attribute.

- **Removal:** Removing rows or columns with too many missing values, although this might lead to losing valuable data.
2. **Feature Scaling:** Machine learning algorithms like logistic regression, SVM, and KNN are sensitive to the scale of features. Features like blood pressure, cholesterol, and age can have different ranges, so it is important to scale them. Standardization or normalization techniques, such as Min-Max scaling, can be used to bring all features into a similar range.
 3. **Encoding Categorical Variables:** Some features, such as sex, chest pain type, and exercise-induced ST depression, are categorical in nature. These need to be encoded into numerical values to be used by the machine learning algorithms. Common methods for encoding categorical data include:
 - **Label Encoding:** Assigning a unique integer to each category.
 - **One-Hot Encoding:** Creating binary columns for each category, where each column represents whether a specific category is present.
 4. **Splitting the Dataset:** The dataset is typically divided into two subsets: a training set and a testing set. A common split is 80% for training and 20% for testing, although this can vary. The training set is used to train the machine learning models, while the testing set is used to evaluate their performance.

```
heartData.info()
heartData.describe()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0    age         1025 non-null   int64
1    sex         1025 non-null   int64
2    cp          1025 non-null   int64
3    trestbps    1025 non-null   int64
4    chol        1025 non-null   int64
5    fbs         1025 non-null   int64
6    restecg     1025 non-null   int64
7    thalach     1025 non-null   int64
8    exang       1025 non-null   int64
9    oldpeak     1025 non-null   float64
10   slope       1025 non-null   int64
11   ca          1025 non-null   int64
12   thal        1025 non-null   int64
13   target      1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Fig 3.1 Attributes in Heart Dataset with datatype

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.529756	149.114146
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	132.000000
50%	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	152.000000
75%	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000	166.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000

Fig 3.2 Detailed Description of Heart Disease Dataset -1

exang	oldpeak	slope	ca	thal	target
1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
0.336585	1.071512	1.385366	0.754146	2.323902	0.513171
0.472772	1.175053	0.617755	1.030798	0.620660	0.500070
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
1.000000	1.800000	2.000000	1.000000	3.000000	1.000000
1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Figure 3.3 Detailed Description of Heart Disease Dataset -2

3.1.3 Model Selection

Once the data is preprocessed, the next step is to choose appropriate machine learning algorithms for training the model. For heart disease prediction, we can use several machine learning models, each with its strengths and weaknesses. Some common algorithms for this task are:

1. **Logistic Regression:** This is a simple but effective model for binary classification tasks. It predicts the probability that a given input belongs to a particular class (e.g., presence or absence of heart disease). It works well when the relationship between the features and the target variable is linear.

2. **K-Nearest Neighbors (KNN):** This is a non-parametric model that classifies a data point based on the majority class of its nearest neighbors. It is effective for small datasets but can be computationally expensive for larger datasets.
3. **Support Vector Machine (SVM):** SVM is a powerful model that works by finding a hyperplane that separates different classes in the feature space. It is especially useful for high-dimensional data and can work well even when the data is not linearly separable by using kernel functions.
4. **Decision Tree:** A decision tree is a tree-like model where each internal node represents a decision based on an attribute, and each leaf node represents the outcome. It is easy to understand and interpret but prone to overfitting if not properly controlled.
5. **Random Forest:** Random Forest is an ensemble learning method that combines multiple decision trees to improve classification accuracy. It is less prone to overfitting compared to individual decision trees.
6. **Naive Bayes:** This is a probabilistic classifier based on Bayes' theorem, which assumes that the features are independent given the class. Despite its simplicity, Naive Bayes often works well for text classification and small datasets.
7. **Artificial Neural Networks (ANN):** ANNs are inspired by the human brain and consist of layers of interconnected neurons. They can model complex relationships between inputs and outputs, but they require more data and computational resources than other models.

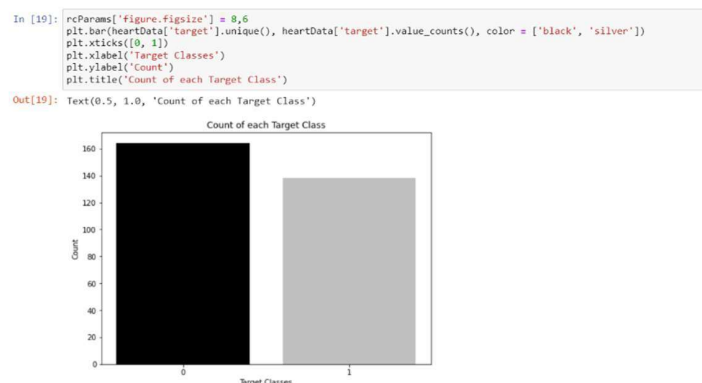


Fig 3.4 Categorization of dataset based on target class

3.1.4 Model Training

Training the machine learning model involves feeding the training data into the selected algorithm and allowing it to learn the patterns and relationships between the features and the target variable (heart disease diagnosis). During this process, the model adjusts its parameters to minimize errors and improve its accuracy.

1. **Hyperparameter Tuning:** Each machine learning algorithm has hyperparameters that control its behavior. For example, in decision trees, the depth of the tree is a key hyperparameter. In KNN, the number of neighbors (K) affects the model's performance. Hyperparameter tuning is done using techniques like grid search or random search to find the best combination of hyperparameters.
2. **Cross-Validation:** To prevent overfitting and ensure that the model generalizes well, cross-validation techniques such as k-fold cross-validation are used. In k-fold cross-validation, the data is split into k subsets, and the model is trained on k-1 subsets and tested on the remaining subset. This process is repeated k times, and the average performance is calculated.



figure 3.5 Maximum positive correlated features are cp and thalach and maximum negative correlated features is exang and old peak.

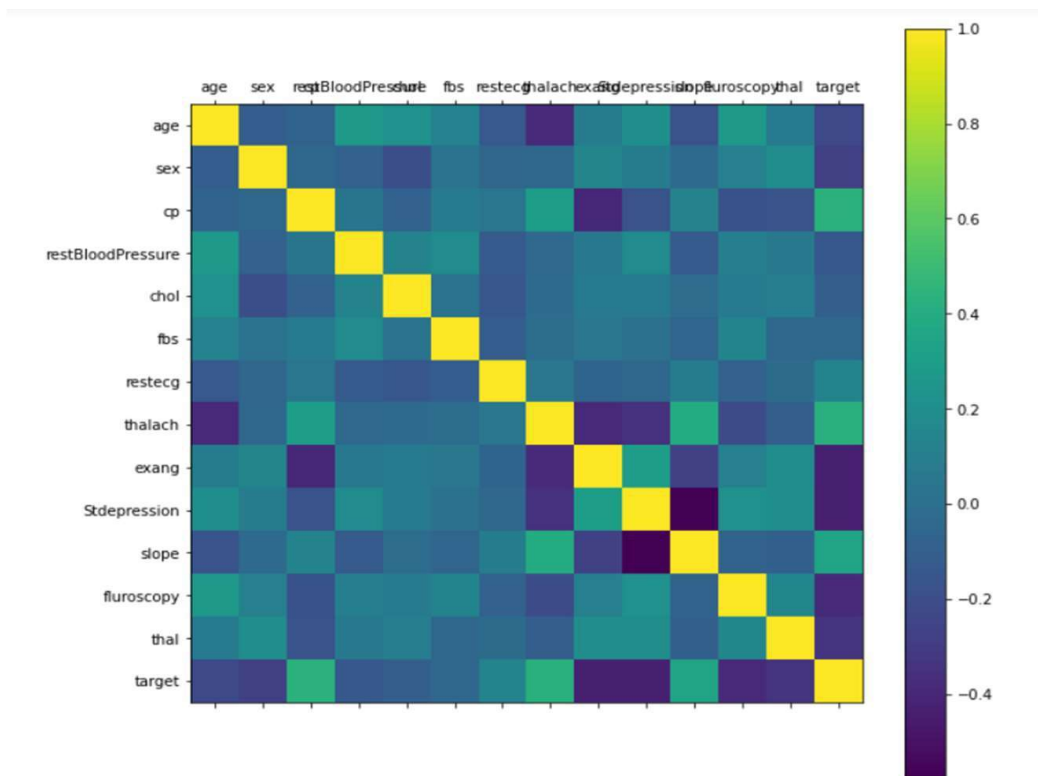


Fig 3.6 Correlation Diagram as heatmap

3.1.5 Model Evaluation

After training the model, it is crucial to evaluate its performance on the testing dataset to ensure that it can generalize to new, unseen data. Common evaluation metrics for classification tasks include:

1. **Accuracy:** This is the ratio of correctly predicted instances to the total instances. While useful, accuracy alone might not be sufficient, especially for imbalanced datasets.
2. **Precision:** Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It is important when the cost of false positives is high.
3. **Recall (Sensitivity):** Recall measures the proportion of true positives out of all actual positive instances. It is important when the cost of false negatives is high.

4. **F1 Score:** The F1 score is the harmonic mean of precision and recall and is used when there is a need to balance the two.
5. **AUC-ROC Curve:** The Area Under the Receiver Operating Characteristic curve (AUC-ROC) helps assess the trade-off between sensitivity and specificity across different thresholds. A higher AUC indicates a better model performance.

3.1.6 Model Improvement

Once the models have been trained and evaluated, further improvements can be made by:

1. **Feature Engineering:** Creating new features or selecting more relevant ones can improve the model's performance. Feature selection techniques like Recursive Feature Elimination (RFE) or mutual information can help reduce overfitting.
2. **Ensemble Methods:** Combining multiple models (e.g., Random Forest, Gradient Boosting, or Voting Classifier) can often lead to better results than a single model.
3. **Advanced Algorithms:** Experimenting with more complex algorithms such as XGBoost or deep learning methods can sometimes yield better performance, especially with larger datasets.

3.1.7 Final Model Selection

After evaluating all models and tuning their parameters, the best-performing model is selected for the final heart disease prediction system. This model will then be deployed for real-time predictions, allowing users to input their medical data and receive a diagnosis regarding the presence or absence of heart disease.

3.2 User Interactive Front End

Flask for Heart Disease Prediction Web Application

This project aims to develop a **multi-page website** that allows users to input their medical data and predict whether they are at risk for heart disease. The website will consist of three primary pages:

1. **Homepage:** An introduction to the project and its objectives.
2. **Prediction Page:** A user interface where individuals can enter their medical details to assess their risk for heart disease.
3. **About Us Page:** Information about the development team and the project's purpose.

The backend, which involves machine learning models, is connected to the frontend (user interface) using **Flask**. Flask is a lightweight Python web framework that is ideal for small to medium-sized applications like this one. It allows us to create a user-friendly and efficient web application without unnecessary complexity.

Why Choose Flask?

Flask is an excellent choice for this project for several reasons:

1. **Lightweight and Simple:** Flask is a micro-framework, providing just the essential tools and libraries required for web development. Its minimalistic nature makes it well-suited for small projects where flexibility and simplicity are important.
2. **Fast Development:** Flask enables quick development with minimal setup. It allows developers to create routes, integrate machine learning models, and test predictions with ease. This is particularly beneficial for rapid prototyping and iterative development in a project like heart disease prediction.
3. **Modular Structure:** Unlike Django, which is built on a monolithic design, Flask supports a more modular approach. This allows developers to add new features, modify routes, or incorporate external libraries without affecting the rest of the application. This flexibility is useful for updating the model or improving the user interface over time.
4. **Seamless Integration with Machine Learning Models:** Flask's simplicity makes it easy to integrate machine learning models with the web application, ensuring smooth communication between the backend and frontend.

3.3 HARDWARE REQUIREMENT

The hardware requirements for running this website and model are:

- RAM – 512 MB

- Operating System – Windows XP/7/8/10/11, MacOS , Ubuntu
- Processor – Intel(R) Core(TM) i3
- Processor speed – 3.60 GHz

3.4 SOFTWARE REQUIREMENTS

The programming language used to develop this application is Python and the IDE used is Jupyter Notebook. Front end is made using HTML, CSS and is integrated with flask.

- Programming Language – Python
- Python IDE – Jupyter Notebook
- Python Libraries: Flask

Experimental Results

The following models were used for the heart disease prediction:

- **Logistic Regression**
- **K-Nearest Neighbours (KNN)**
- **Support Vector Machine (SVM)**
- **Decision Tree**
- **Random Forest**

4.1 Logistic Regression

Logistic Regression is one of the most used models in machine learning. It is frequently applied in fields such as data mining, automated disease diagnosis, and economic prediction.

For our model, Logistic Regression is utilized to analyze the risk factors for heart disease and predict the likelihood of the disease's occurrence based on these factors. This model is primarily applied to classification problems, especially in cases where there are two possible outcomes, representing two distinct categories. It can estimate the probability of each classification event happening.

The technique used in Logistic Regression is also known as the sigmoid function. The sigmoid function is particularly useful because it produces outputs between 0 and 1, which can be interpreted as probabilities. This function is also easy to visualize in graphs.

Logistic Regression is known for its ability to provide better accuracy in predicting outcomes by using an equation that represents the relationship between input features and the probability of a particular event (e.g., the occurrence of heart disease). The results can be represented in graphs that clearly show how the various attributes influence the predicted outcomes.

$$prob(Y = 1) = \frac{e^z}{1 + e^z}$$

Where Y refers to binary dependent variable (Y is equal to 1 if event happens; Y=0 otherwise), e stands for the foundation of natural logarithms and Z means

$$Z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$$

with constant β_0 , coefficients β_j and predictors X_j , for p predictors ($j=1,2,3,\dots,p$)

Logistic Regression is a technique used for modeling the probability of a discrete outcome based on input variables. Despite the term "regression" in its name, Logistic Regression is a classification algorithm, as it is typically used to classify data into two possible categories, such as true/false, yes/no, and so on.

In Logistic Regression, the goal is to identify a hyperplane that best separates the two classes. This separation is achieved by passing the linear combination of input features through a function (usually the **sigmoid function**) that squashes the output to a value between 0 and 1. This output represents the probability of the instance belonging to one of the two categories.

The model is trained using vector parameters, and the sigmoid function, denoted as $\sigma(\cdot)$, is employed to produce the probability values. The output of the sigmoid function lies between 0 and 1, where values close to 0 indicate one class (e.g., "no disease"), and values close to 1 indicate the other class (e.g., "heart disease"). The model works by optimizing a cost function to minimize errors, leading to accurate classification predictions based on input features.

Features of Logistic Regression:

Multinomial Logistic Regression is an extension of logistic regression used when the dependent variable has more than two categories. Instead of using the sigmoid function, as in binary logistic regression, it uses the **softmax function** to calculate the probabilities for each possible outcome in the multi-class classification problem.

Key points about the process include:

1. **Loss Function and Weight Learning:**

- In multinomial logistic regression, a **loss function** is employed to learn the model's parameters, which include the weight vector (w) and bias (b). The learning process aims to minimize the **cross-entropy loss**, which quantifies the difference between the predicted probability distribution and the true labels.
2. **Iterative Optimization:**
 - To find the optimal weights, **iterative algorithms**, like **gradient descent**, are utilized. These methods update the weights based on the gradient of the loss function in relation to the parameters. This iterative process seeks to minimize the loss function, which can be viewed as a **convex optimization problem**, ensuring convergence to an optimal solution.
 3. **Regularization to Avoid Overfitting:**
 - **Regularization** techniques, such as L1 (Lasso) or L2 (Ridge) regularization, are applied to prevent overfitting. Regularization adds a penalty to the loss function based on the size of the model coefficients, thereby encouraging the model to use fewer or smaller features, thus improving generalization on unseen data.
 4. **Feature Importance:**
 - One of the key advantages of logistic regression is its ability to **transparently study the importance of individual features**. By examining the learned weights associated with each feature, you can understand how each attribute contributes to the model's predictions. Larger weight values (either positive or negative) indicate more significant features in the decision-making process.

Advantages of Logistic Regression:

1. **Efficiency and Speed:**
 - Logistic regression is known for its ability to perform well and fast, especially when classifying **unknown records**. It is an efficient algorithm that works well even with a relatively smaller amount of data, making it suitable for real-time predictions.
2. **Flexibility:**
 - While commonly used for binary classification, logistic regression can be easily **extended to multinomial classification**. This flexibility allows it to handle

problems where the dependent variable has more than two categories, making it adaptable to a variety of classification tasks.

Disadvantages of Logistic Regression:

1. Performance with Small Datasets:

- Logistic regression may not perform well when the **number of observations** (samples) is smaller than the **number of features** (predictors). In such situations, the model may struggle to generalize properly, leading to **overfitting**. Overfitting occurs when the model learns the noise in the training data instead of the actual underlying patterns.

2. Linear Boundaries:

- Logistic regression constructs **linear decision boundaries** between classes. This can be a limitation when dealing with datasets that require complex, non-linear decision boundaries. In cases where the data distribution is non-linear, logistic regression may not be able to capture the relationships accurately, thus reducing its performance.

In the logistic function, the input variable x is transformed by the logistic equation:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \sigma(x) = \frac{1}{1 + e^{-x}}$$

Where:

- $\sigma(x)$ is the output of the logistic function, which ranges between 0 and 1.
- e is the base of the natural logarithm (approximately 2.718).

When you input values ranging from -20 to 20 into this equation, the logistic function will map those values into the range between 0 and 1. This is because the logistic function is a **sigmoid** function, meaning it has an "S"-shaped curve that smoothly transitions from 0 to 1.

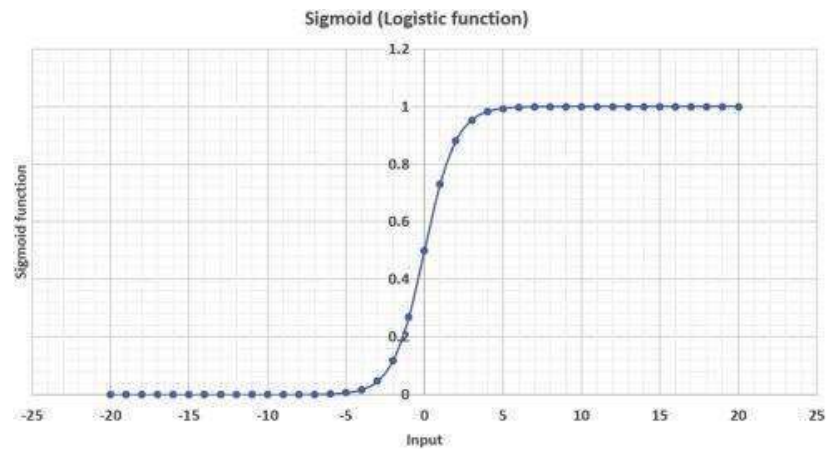


Figure. 4.1 Sigmoid Graph

```
In [146]: logistic_model = LogisticRegression()
logistic_model.fit(X_train.values, y_train.values)
logistic_model_prediction=logistic_model.predict(X_test.values)
print(accuracy_score(y_test.values,logistic_model_prediction))
print(classification_report(y_test.values,logistic_model_prediction))
```

0.8131868131868132

	precision	recall	f1-score	support
0	0.85	0.71	0.77	41
1	0.79	0.90	0.84	50
accuracy			0.81	91
macro avg	0.82	0.80	0.81	91
weighted avg	0.82	0.81	0.81	91

Figure.4.2 Logistic Model Result

Accuracy: 82%

4.2 KNN

K-Nearest Neighbors (KNN) is a simple, yet powerful machine learning algorithm used for both classification and regression tasks. It operates by comparing the input data point with all the available training data points and calculating the distance between them using various distance metrics, such as Euclidean or Manhattan distance. The algorithm then identifies the 'k' nearest neighbors to the input data point and makes predictions based on the majority vote or the average of these neighbors.

How KNN Works:

1. Training Phase:

- Unlike other algorithms, KNN does not explicitly "train" a model. Instead, it simply stores the training data and calculates distances when a new input is provided.
- The training phase involves passing the training data set to the function `fit()` of the `KNeighborsClassifier` from the `sklearn.neighbors` module.

2. Prediction Phase:

- When a new input is provided, the `predict()` function is used to make predictions. It computes the distance between the input data point and all the training data points.
- Based on the 'k' closest neighbors, the algorithm predicts the output. For classification, it uses the majority vote, while for regression, it averages the output values of the neighbors.

3. Choosing the Right 'k':

- The value of 'k' (the number of neighbors to consider) needs to be determined experimentally. Selecting the right 'k' is crucial for the performance of the model.
- If 'k' is too small, the model may become sensitive to noise, and if 'k' is too large, the model may over smooth the predictions and lose precision.
- There is typically a "knee point" at which increasing 'k' no longer improves accuracy significantly, and after this point, accuracy may even decrease. In practice, if 'k' turns out to be less than 5, a default value of 5 is often chosen.

Features of KNN:

- **Supervised Learning:** KNN is a supervised learning algorithm, meaning it relies on labeled data to make predictions.
- **Simplicity:** One of the simplest and easiest-to-implement machine learning algorithms. It is suitable for various types of problems and does not require training in the traditional sense.
- **Feature Similarity:** KNN is based on the concept of feature similarity. It classifies a data point based on the classes of its closest neighbors, if similar data points will share similar outputs.

- **Distance-based:** The algorithm uses a distance metric (such as Euclidean or Manhattan distance) to measure how similar or close a data point is to its neighbors. The prediction is then made based on this proximity.

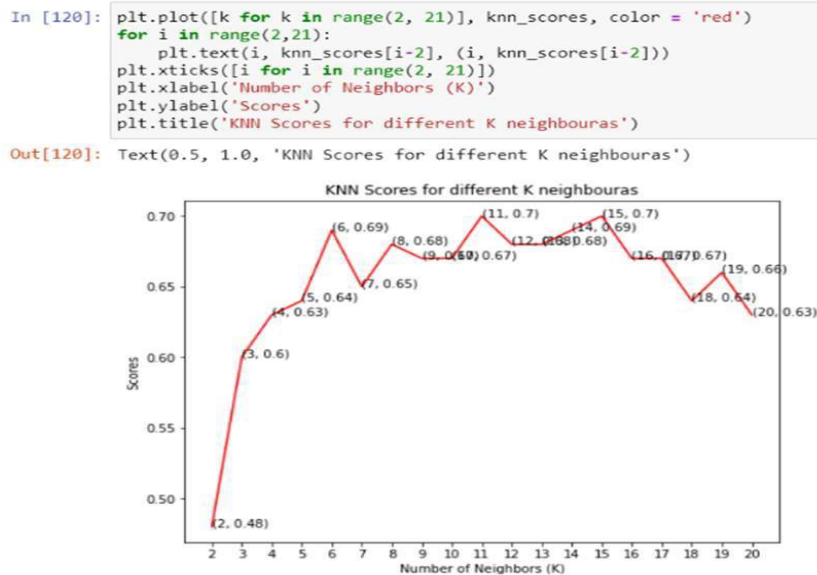


Figure. 4.3 KNN output graph

Since we are getting the best value at $k = 11$ and $k = 15$, it was decided in favour of $k = 11$, which is usually the default value. Finally, prediction is done on test data and the result table is prepared.

```
#dated::10/0/2022
knn_scores = []
for k in range(2,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    knn_classifier.fit(X_train.values, y_train.values)
    knn_score=round(knn_classifier.score(X_test.values, y_test.values),2)
    knn_scores.append(knn_score)

knn_classifier = KNeighborsClassifier(n_neighbors = 5)
knn_classifier.fit(X_train, y_train)
knn_score=knn_classifier.predict(X_test)
print(classification_report(y_test,knn_score))
```

	precision	recall	f1-score	support
0	0.62	0.49	0.55	41
1	0.64	0.76	0.70	50
accuracy			0.64	91
macro avg	0.63	0.62	0.62	91
weighted avg	0.64	0.64	0.63	91

Fig 4.4 Classification report of KNN (Accuracy : 64%)

4.3 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm that is commonly used for classification, although it can also be applied to regression tasks. It works by mapping each data point to a point in an n -dimensional space, where n represents the number of features (or attributes) in the data. The goal of SVM is to find the optimal hyperplane that separates the data points of different classes with the maximum margin. This hyperplane is called the decision boundary, and the data points closest to the boundary are known as support vectors.

Features of SVM:

Use of Missing Values: SVM can handle missing data by assigning appropriate levels to missing values, depending on the problem and dataset.

Iterations Report: SVM provides detailed information on the number of iterations during the training process, helping to monitor the convergence and progress of the algorithm.

Penalty Parameter: The penalty parameter (often denoted as C) controls the trade-off between achieving a low error on the training data and maintaining a margin as large as possible. The default value is typically set to 1, but it can be adjusted depending on the problem's requirements.

Kernel Functions: SVM can use various types of kernels (such as linear, polynomial, radial basis function (RBF), and sigmoid) to transform the input data into higher-dimensional space, making it possible to separate data that is not linearly separable in the original space. The choice of kernel depends on the nature of the data.

Polynomial Degree: In case of polynomial kernels, the polynomial degree parameter determines the degree of the polynomial used. The default value is usually set to 2, but it can be adjusted for different complexities in the dataset.

Tolerance: Tolerance specifies the minimum change in the objective function at which the iterations will stop. A smaller value indicates more iterations to achieve a more precise solution. The default tolerance is 0.000001.

Maximum Iterations: This parameter specifies the maximum number of iterations allowed during training. The default value is typically 25, but this can be modified to ensure convergence, especially for complex datasets.

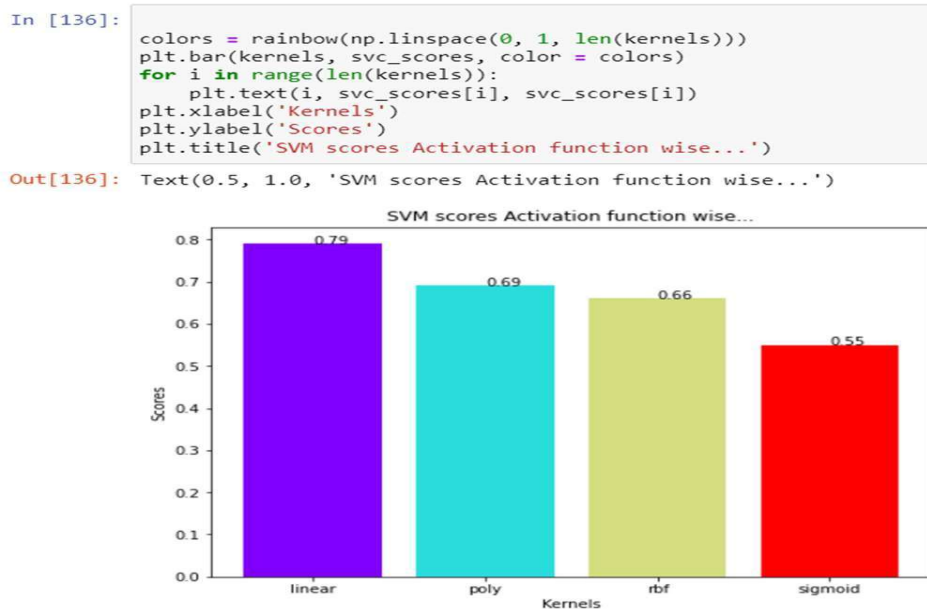


Fig 4.5 SVM Scores against various kernels

Accuracy :79% with linear kernel

KERNEL	ACCURACY
Linear	79
Poly	69
RBF	66
Sigmoid	55

Table 4.1 SVM Scores against various kernels

4.4 Decision Tree

A Decision Tree (DT) is a tree-like model used for decision-making, where each internal node represents a test or decision based on a particular attribute. The branches coming out from the node represent the outcome of the test, leading to further tests or final decisions. The leaf nodes of the tree are used to represent class labels or outcomes.

In essence, a Decision Tree is structured like a flowchart:

- **Internal Nodes:** Represent tests or decisions based on attributes (features).
- **Branches:** Represent the outcomes of these tests or decisions, directing to either further tests or final results.
- **Leaf Nodes:** Represent the classification or final decision (usually class labels in classification tasks or values in regression tasks).
- **Paths from Root to Leaf:** Represent classification rules or decision-making paths.

Visual and Analytical Perspective:

Decision Trees are useful both for visualizing and analyzing decisions. The structure of the tree gives a clear representation of the decision-making process. By following the path from the root node to the leaf node, one can trace how an outcome is determined based on the attributes tested at each node.

The expected values of competing alternatives, in terms of classification, are calculated by analyzing the choices made at each branch of the tree. In the case of classification, the path from the root node to a leaf node corresponds to a specific rule or condition that leads to a final classification decision.

Architecture of a Decision Tree:

The architecture of a Decision Tree follows this structure:

1. **Root Node:** This is the starting point of the tree, where the first decision or test is made.

2. **Splitting/Branching:** Based on the outcome of the test at each node, the tree branches out into different paths, leading to further nodes or leaf nodes.
3. **Leaf Nodes:** These represent the final decision or outcome. Each leaf node contains the predicted class label or value.
4. **Pruning:** In some cases, after the tree is built, unnecessary branches can be removed to simplify the tree and avoid overfitting, a process known as pruning.

Key Benefits of Decision Trees:

- They provide a clear, interpretable structure, making it easy to understand how decisions are made.
- They can handle both categorical and numerical data.
- The process of classifying new data using the tree is straightforward—one simply follows the decision path based on the input attributes.

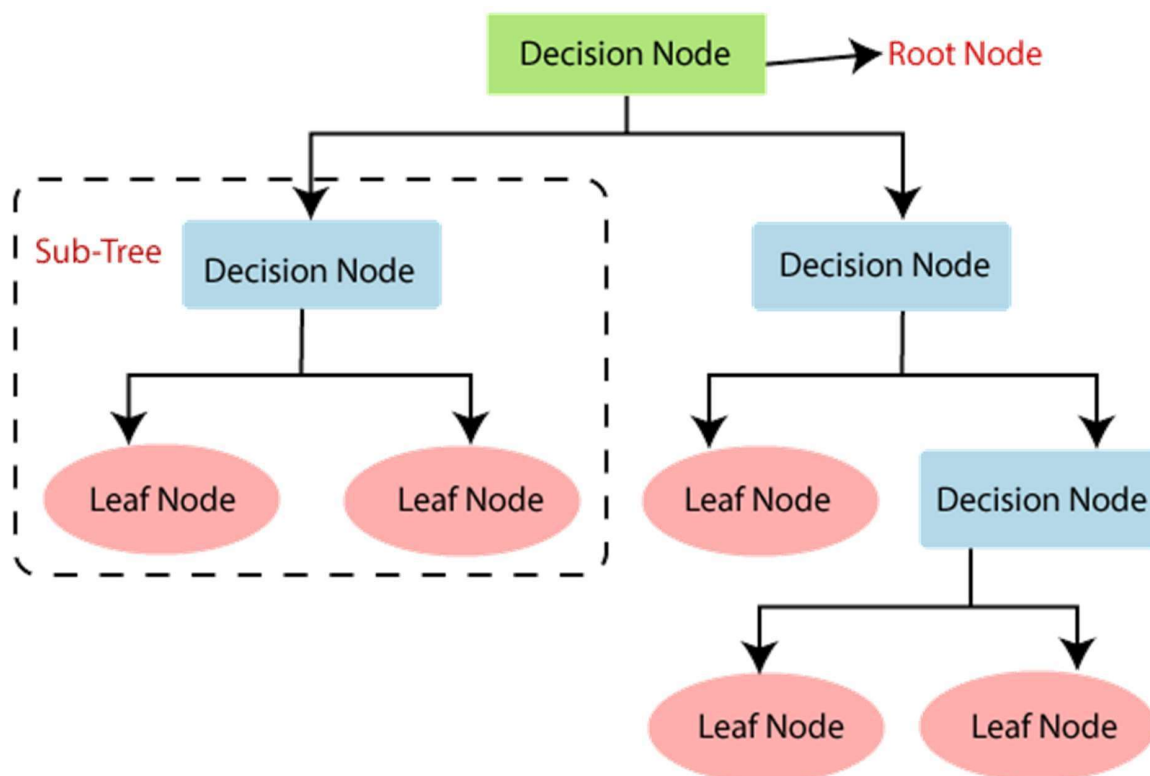


Fig 4.6 Decision Tree image

```
In [142]: plt.plot([i for i in range(1, len(X.columns) + 1)], dt_scores, color = 'green')
for i in range(1, len(X.columns) + 1):
    plt.text(i, dt_scores[i-1], (i, dt_scores[i-1]))
plt.xticks([i for i in range(1, len(X.columns) + 1)])
plt.xlabel('Max features')
plt.ylabel('Scores')
plt.title('Decision Tree Classifier scores for different number of maximum features')

Out[142]: Text(0.5, 1.0, 'Decision Tree Classifier scores for different number of maximum features')
```

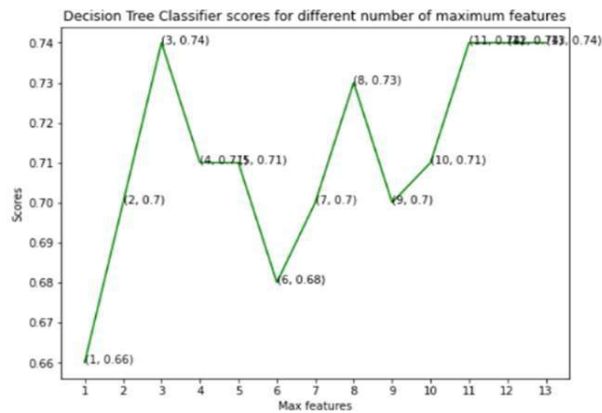


Fig 4.7 Decision tree Result Graph

Accuracy: 74%

4.5 Random Forest Classifier

This method is based on the **Random Forest**, which is an ensemble learning technique that combines multiple decision trees to improve the overall classification performance. In a Random Forest, rather than relying on a single decision tree, multiple decision trees are used, each making its own prediction or "vote" for the given input data. Here's how it works:

1. **Multiple Decision Trees:** A Random Forest consists of a collection (or "forest") of decision trees. Each tree in the forest is trained on a random subset of the training data. This randomness helps to ensure that each tree is different, and this diversity contributes to the overall strength of the model.
2. **Voting Process:** When a new input data point needs to be classified, it is passed through all the decision trees in the forest. Each tree makes a classification decision, essentially casting a "vote" for the class label of the input.

3. **Majority Voting:** After all trees have made their predictions, the final classification is determined by majority voting. The class that receives the most votes from the decision trees becomes the predicted class for the input data.

Key Characteristics of Random Forest:

- **Ensemble Approach:** By combining the predictions of many decision trees, Random Forest generally provides better accuracy and robustness than any individual decision tree.
- **Reduces Overfitting:** While individual decision trees may suffer from overfitting (especially if they are very deep), the aggregation of multiple trees helps to reduce the likelihood of overfitting.
- **Versatility:** Random Forest can handle both classification and regression tasks effectively.

Steps in Random Forest:

1. **Bootstrapping:** For each tree, a random subset of the training data is chosen with replacement (bootstrapping), meaning some data points may appear multiple times in a subset, while others may be left out.
2. **Random Feature Selection:** During the splitting of nodes in each decision tree, a random subset of features (rather than all features) is selected for evaluation. This ensures that the trees in the forest are diverse and helps to reduce correlation between trees.
3. **Prediction:** Once the forest is trained, each tree in the forest votes on the classification or regression result for new data points.

Advantages of Random Forest:

- **High Accuracy:** Random Forest typically provides high accuracy, as the combination of many trees helps to improve the prediction performance.
- **Robustness:** It is less prone to overfitting compared to individual decision trees, especially when the dataset is large.
- **Works Well for Both Classification and Regression:** Random Forest can be applied to a wide variety of problems, making it a versatile choice in machine learning.
- **Feature Importance:** It can also be used to assess the importance of different features in making predictions, which is useful for understanding the underlying relationships in the data.

Random Forest Classifier

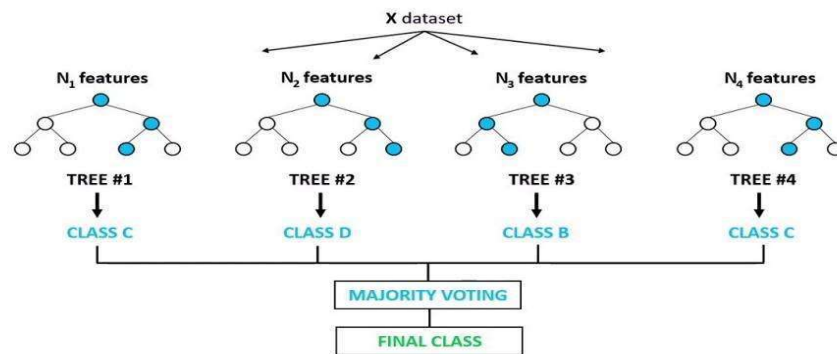


Fig 4.8 Random Forest Image

Features of Random Forest:

- **Efficient with Large Datasets:** Random Forest is highly efficient and performs well with large datasets. It can handle millions of data points without compromising on performance, making it ideal for big data applications.
- **Handles Numerous Input Variables:** This algorithm can perform classification even with many input variables (features). It is capable of processing and making sense of data with many features without significant loss in accuracy.
- **Feature Selection:** Random Forest can identify the most important variables (features) that contribute to the classification. It helps in determining which features have the most impact on the outcome, enabling better decision-making and feature engineering.
- **Handles Missing Data:** One of the key advantages of Random Forest is its ability to handle missing data. Even if some data points are missing, the algorithm can still maintain good accuracy. It uses techniques like imputation (filling in missing values) and bootstrapping (sampling with replacement) to handle incomplete datasets effectively.

```
4]: Text(0.5, 1.0, 'Random Forest Classifier scores for different number of estimators')
```

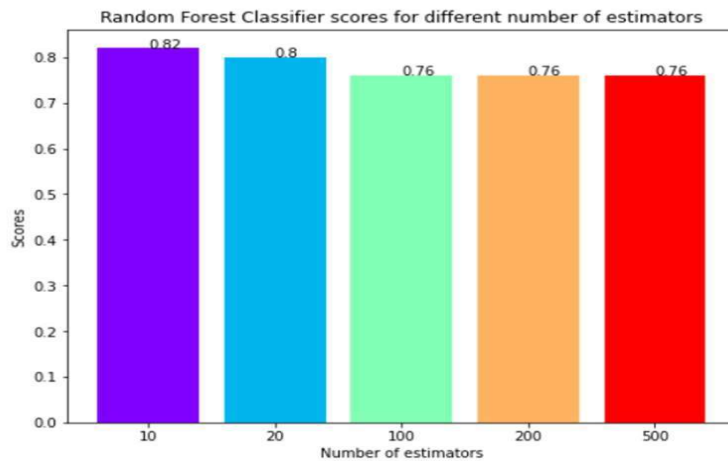


Fig 4.9 Random Forest Result Image

Accuracy:82% with 100 estimators. Increase in number of estimators will increase the calculation complexities significantly.

Conclusion

Working Model Diagram

The working model of the system involves a user-friendly frontend where users can input their medical information, which is then processed through machine learning models in the backend. Here's an outline of how the system works:

1. Frontend (User Interface):

- The frontend consists of a simple webpage where users will enter their medical details into a form.
- The form may include fields such as:
 - Age
 - Sex
 - Blood Pressure
 - Cholesterol levels
 - History of heart disease or other risk factors (e.g., smoking, diabetes, etc.)
- Once the form is filled out, the user can submit the data for processing.

2. Backend (Model Processing):

- Upon submission, the frontend sends the user's input to the backend.
- In the backend, the input data is processed by a machine learning model trained on heart disease data (such as Logistic Regression, Random Forest, or KNN).
- The machine learning model will analyze the input data and compute the probability of the user having heart disease based on the patterns learned from historical medical data.

3. Model Evaluation:

- The machine learning model evaluates the input data, comparing it with historical data to predict the likelihood of heart disease.
- The result can be a simple "Yes" or "No" indicating whether the user is at risk of heart disease or not. Alternatively, the model may output a probability score.


4. Result Display:

- The result from the model is then sent back to the frontend.
- The user is presented with the prediction result, which could be a recommendation to seek medical advice or reassurance based on the model's findings.

Heart Disease Predictor

About Us

Heart Disease Predictor



Name

test user

Email

testuser@gmail.com

Age

29

Select your gender

Male


Chest Pain Types

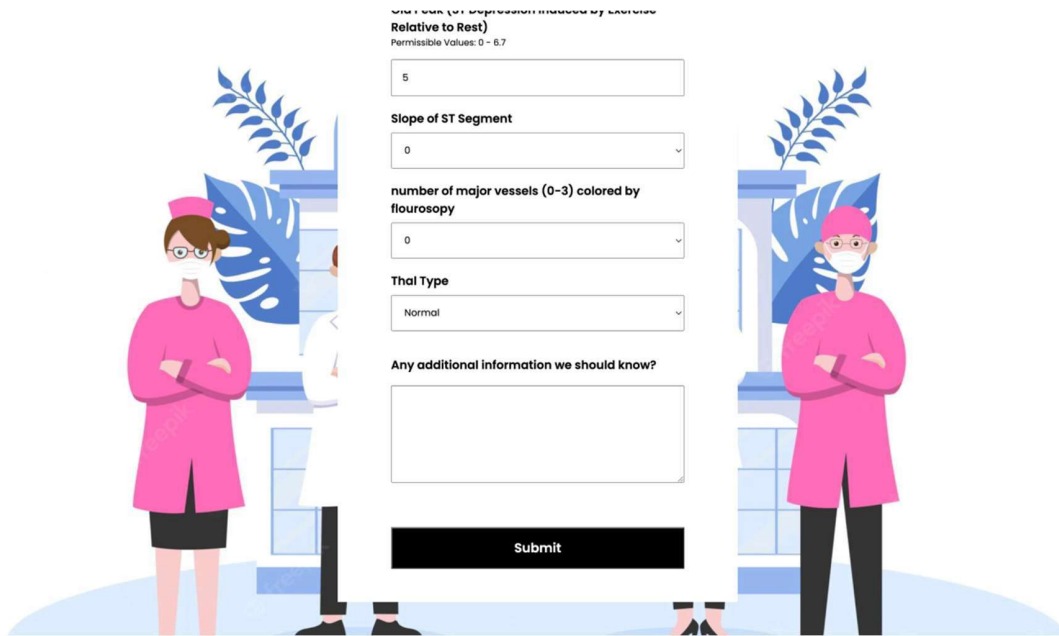
Typical Angina

Resting Blood Pressure(In mm/Hg)

94

Cholesterol Level





Old Peak (ST Depression Induced by Exercise Relative to Rest)
Permissible Values: 0 - 6.7

5

Slope of ST Segment

0

number of major vessels (0-3) colored by flourosopy

0

Thal Type

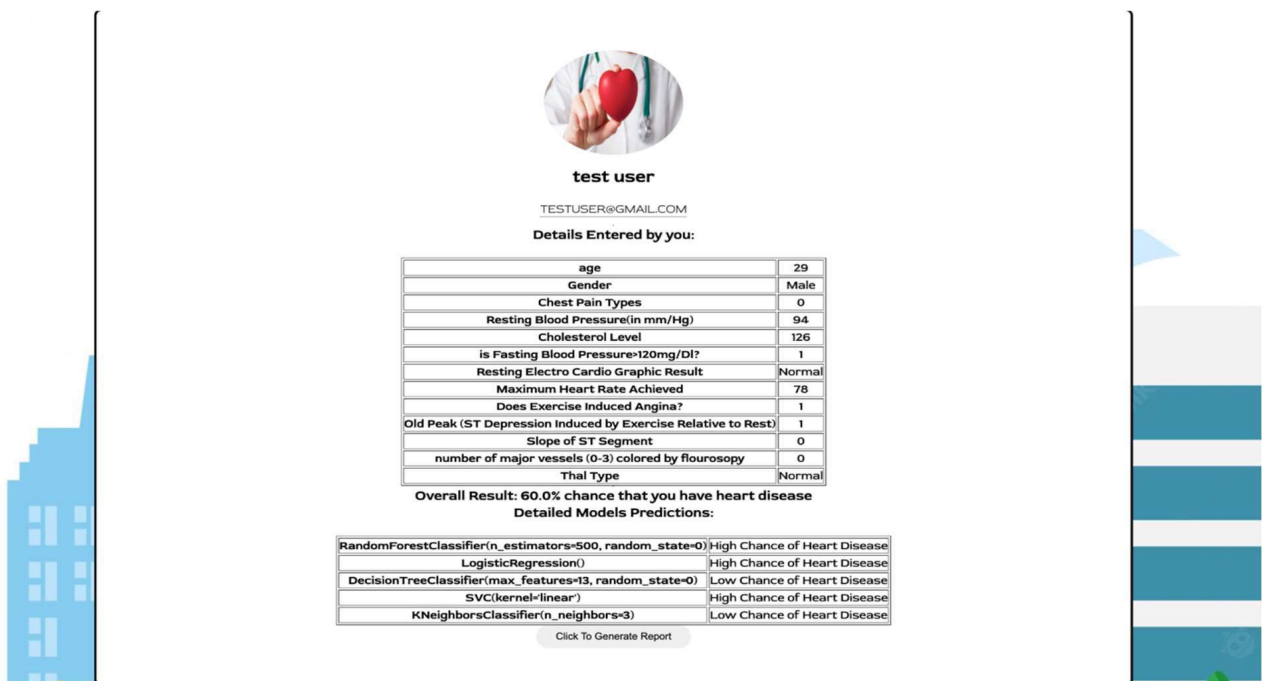
Normal

Any additional information we should know?

Submit

Figure 5.1 Input Form

The result page screenshot:



test user
TESTUSER@GMAIL.COM

Details Entered by you:

age	29
Gender	Male
Chest Pain Types	0
Resting Blood Pressure(in mm/Hg)	94
Cholesterol Level	126
is Fasting Blood Pressure>120mg/Dl?	1
Resting Electro Cardio Graphic Result	Normal
Maximum Heart Rate Achieved	78
Does Exercise Induced Angina?	1
Old Peak (ST Depression Induced by Exercise Relative to Rest)	1
Slope of ST Segment	0
number of major vessels (0-3) colored by flourosopy	0
Thal Type	Normal

Overall Result: 60.0% chance that you have heart disease

Detailed Models Predictions:

RandomForestClassifier(n_estimators=500, random_state=0)	High Chance of Heart Disease
LogisticRegression()	High Chance of Heart Disease
DecisionTreeClassifier(max_features=13, random_state=0)	Low Chance of Heart Disease
SVC(kernel=linear)	High Chance of Heart Disease
KNeighborsClassifier(n_neighbors=3)	Low Chance of Heart Disease

[Click To Generate Report](#)

Figure 5.2 Result Page

It also has a button to print the report generated so that user can have a record of data entered by him/her and the respective result.

5.2 Detailed Advantages of the Project

This project offers several advantages that can benefit individuals, medical professionals, and healthcare systems. Below are the key advantages in more detail:

1. Accessibility for Individuals:

- The project allows individuals to assess their heart disease risk using a simple tool accessible through their devices. By entering basic medical details, individuals can receive an initial prediction about whether they may have heart disease. This eliminates the need to visit healthcare facilities for basic screening, providing a more convenient and quicker means of checking for potential risks.
- The accessibility of this tool ensures that even people in remote areas, where healthcare facilities may be limited, can assess their heart disease risk and take preventative actions in a timely manner.

2. Support for Medical Professionals:

- This tool can act as a support system for medical professionals by providing them with an additional layer of information to assist in diagnosing heart disease. Once a patient inputs their medical records, the system quickly processes the data and generates a prediction, which can be reviewed by a healthcare provider.
- It reduces the diagnostic time for medical professionals, allowing them to focus on more complex cases or decision-making that requires human intervention, thereby improving overall healthcare efficiency.

3. Reduced Dependency on Medical Professionals for Basic Decision-Making:

- By automating the initial stages of diagnosis, this project helps reduce the dependency on medical professionals for basic decision-making tasks. The system can handle standard assessments, which would

otherwise require a medical expert's involvement, making the process more efficient.

- This can help alleviate the burden on healthcare systems, especially in settings with limited medical staff, and allow healthcare professionals to focus on critical cases where their expertise is needed most.

4. Scalability and Wider Adaptation:

- The project is designed to be scalable, meaning it can be easily deployed across multiple healthcare facilities, such as hospitals, clinics, and heart disease diagnostic centers. Its adaptability makes it an ideal tool for large-scale use, helping to streamline the process of heart disease diagnosis across various healthcare settings.
- With the growing prevalence of heart disease, especially in developing countries with limited healthcare resources, this project can serve as a cost-effective solution to increase access to heart disease screening. It can be implemented in public health programs to reach a larger population and contribute to better health outcomes.
- As the tool is based on a machine learning model, it can be updated and refined as more data becomes available, continuously improving its diagnostic accuracy and expanding its potential applications.

5. Time and Cost Efficiency:

- The use of this tool helps reduce the time spent by both patients and medical professionals in the diagnostic process. Patients no longer need to wait for lab tests or multiple consultations for basic assessments. Instead, they can get a preliminary result within minutes, saving time and reducing unnecessary appointments.
- From a financial standpoint, the use of this tool can significantly lower healthcare costs by automating initial screenings, which would traditionally require costly tests and procedures. This makes it a cost-effective solution for both individuals and healthcare providers, especially in resource-constrained environments.

6. Increased Early Detection and Prevention:

- Early detection is critical in managing heart disease, as timely intervention can greatly reduce the risk of severe complications. This tool helps in early identification of potential risks by analyzing basic health parameters and predicting the likelihood of heart disease, allowing individuals to seek medical advice or lifestyle changes sooner.
- By increasing awareness and providing early warning signs, the tool can encourage more people to take proactive steps toward preventing heart disease, such as adopting healthier habits and undergoing further medical evaluations if necessary.

7. Data-Driven Decision Making:

- The machine learning model used in this project is data-driven, meaning it continuously learns and improves as more medical records are analyzed. This can lead to more accurate predictions over time and assist in identifying hidden patterns in patient data that might otherwise go unnoticed.
- The model's ability to process a large volume of data allows healthcare professionals to make more informed, evidence-based decisions about patient care, leading to better health outcomes.

8. Contributing to Public Health Initiatives:

- This project can play a crucial role in public health initiatives aimed at reducing the prevalence of heart disease. By making the tool available to a wider population, governments and health organizations can encourage regular screenings, educate the public about heart disease risk factors, and promote preventative measures.
- It can be integrated into health awareness campaigns and used in community outreach programs to reach underserved populations who may otherwise lack access to regular health check-ups.

5.3 Project Limitations

While this project offers significant advantages in predicting heart disease risk, there are several limitations that need to be considered:

1. **Dependence on Medical Tests for Certain Attributes:**

The current model requires 13 attributes for prediction. Many of these attributes, such as blood pressure, cholesterol levels, ECG results, and others, are medical-specific and cannot be obtained without undergoing specific tests. These tests often involve costs, time, and medical resources, which may be inaccessible to some individuals. As a result, a significant portion of the population might not be able to benefit from the tool without first undergoing these medical tests, limiting its reach.

2. **Complexity of Medical Terminology:**

The attributes required by the model are described in technical medical terms, which might not be easily understood by a general user. Terms such as "resting blood pressure", "serum cholesterol", and "ECG" can be confusing for those without a medical background. For the tool to be more accessible and user-friendly, these terms would need to be simplified or replaced with more familiar, everyday language. In addition, the inclusion of certain medical metrics that require specialized knowledge could make the tool intimidating for non-experts, potentially discouraging some users from using it.

3. **Limited Range of Accessible Attributes:**

The current model focuses heavily on clinical attributes that may not always be readily available to the general public. For example, attributes like "fluoroscopy colored vessels" or "highest ST-segment" require specific diagnostic procedures and equipment, which are not widely available. It would be more practical for the tool to rely on more common, accessible attributes such as lifestyle factors like smoking, alcohol consumption, exercise habits, family medical history, and diet, which are important in predicting heart disease but can be self-reported by the user.

4. **Cost of Medical Tests:**

Since many of the attributes required by the model can only be gathered through medical tests (e.g., blood tests, ECG, or imaging), there is a financial cost to accessing these tests. This could pose a barrier for some individuals who may not have the resources to afford such tests. As a result, while the tool

can provide useful information, it may not be fully effective for people without access to affordable healthcare services, especially in low-income or underserved regions.

5. **Need for More Accessible Data:**

A more user-friendly and widespread heart disease prediction tool would need to focus on attributes that are easier for individuals to report without needing medical professionals or equipment. Examples of such attributes include smoking habits, alcohol consumption, diet, physical activity, family history of heart disease, and other lifestyle factors. These factors play a significant role in heart disease prediction and could be used as self-reported data, making the tool more accessible to a larger population.

6. **Potential for Inaccuracies:**

The model's reliance on medical data and self-reported attributes can lead to inaccuracies. For example, if a user is unaware of their exact blood pressure or cholesterol levels, they might provide inaccurate data, which could affect the accuracy of the prediction. Additionally, self-reported attributes like smoking or exercise frequency might not always be accurate, leading to potential discrepancies in the model's predictions.

5.4 Future Work

The future development of this project holds numerous opportunities for enhancement, both in terms of functionality and accessibility. Some potential future work includes:

1. **Deployment and Parameter Modification:**

A key future improvement involves deploying the project on a larger scale, allowing users to access the tool remotely via a web interface or mobile application. Additionally, the parameters used in the model could be modified for better adaptability. By making these adjustments, the tool can be optimized to work with a broader range of user inputs, potentially increasing its flexibility and reach.

2. **Training with Reduced Parameters:**

Future versions of the model could be trained with a reduced set of

parameters, focusing on more easily accessible and commonly available data points. This would make the tool more practical for users who may not have access to specific medical tests or equipment. Even with fewer parameters, efforts can be made to ensure the model still provides reasonable predictions while maintaining or improving accuracy.

3. Improved Accuracy through Algorithm Refinement:

The accuracy of the model can always be improved. Future work can focus on fine-tuning the algorithms used, such as exploring different feature engineering techniques, optimizing hyperparameters, or using ensemble methods. By continuously refining the model, it may be possible to achieve even higher prediction accuracy and provide more reliable results for heart disease detection.

4. Incorporating Additional Classification Algorithms:

In addition to the current algorithms, other classification methods can be explored. Techniques like Gradient Boosting, XGBoost, or Deep Learning could be tested to compare performance with the existing models. By using multiple algorithms, the system can select the best-performing model or even combine them into an ensemble to improve accuracy.

5. Handling Missing Data:

Another area for future development is improving how the model handles missing or incomplete data. Many users may not have access to all the necessary medical information, so providing an option for users to receive predictions even with partial data would increase the tool's accessibility. Although the accuracy may be reduced in these cases, having a solution for partial inputs would allow users to get valuable insights based on the information they can provide.

6. User-Friendly Interface and Simplified Attributes:

Simplifying the attributes used in the model, replacing technical medical terms with more easily understood language, and incorporating more common lifestyle factors such as smoking, alcohol consumption, and exercise frequency could increase the model's accessibility to a wider audience. A

more intuitive, user-friendly interface would also improve the user experience, enabling more people to benefit from the tool without requiring a medical background.

7. Integration with Wearable Devices:

In the future, integrating the prediction model with wearable health devices (such as smartwatches or fitness trackers) could provide real-time monitoring of users' heart health. By continuously gathering data on heart rate, activity levels, and other vital signs, the tool could offer ongoing assessments of heart disease risk, making it more personalized and proactive.

8. Expanding to a Broader Range of Diseases:

Beyond heart disease, the framework could be expanded to include predictions for other common health conditions, such as diabetes, hypertension, or stroke. By extending the tool's scope, it could provide a more comprehensive health assessment to users, offering predictions and insights into various aspects of their health.

Conclusion

In conclusion, this project aims to address a critical issue in healthcare: the prediction and early diagnosis of heart disease. By leveraging machine learning techniques, this project utilizes various algorithms to predict the likelihood of heart disease based on specific input attributes such as age, sex, blood pressure, cholesterol levels, and others. These attributes are selected based on their proven significance in the diagnosis of cardiovascular diseases.

Key Findings:

- **Machine Learning Algorithms:** Different machine learning algorithms, including Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees, and Random Forest, were implemented to predict the presence of heart disease. Each of these models provides a unique way to process data and make predictions. For example, Logistic Regression helps in determining the probability of disease occurrence, while Random Forest improves prediction accuracy by combining multiple decision trees. KNN and SVM are also significant for their non-linear classification capabilities.
- **Importance of Data:** The success of heart disease prediction models relies heavily on the availability and quality of data. The dataset used in this project includes both positive (patients with heart disease) and negative (patients without heart disease) examples. However, many of the required attributes are medical-specific, which may not be easily accessible for the average user without undergoing medical tests.
- **Flask Web Application:** A web application was developed using Flask, which serves as the bridge between the trained machine learning models and the users. The frontend consists of forms where users input their medical information, and the backend processes this data through the trained models to predict the likelihood of heart disease. The application provides an easy-to-use interface for both medical professionals and general users to check for the presence or absence of heart disease based on personal medical records.
- **Model Performance:** The performance of the models was evaluated, and it was found that algorithms like Random Forest and Decision Trees performed better in terms of classification accuracy. However, each model has its strengths and limitations, and their

performance varies based on the dataset and features used. Fine-tuning the models and adding more relevant features could improve the accuracy further.

References

1. World Health Organization (WHO). Health Workforce. Available at: <https://www.who.int/data/gho/data/themes/topics/health-workforce>
2. Mayo Clinic. Heart Disease: Symptoms and Causes. Available at: <https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118>
3. World Health Organization (WHO). Cardiovascular Diseases. Available at: <https://www.who.int/healthtopics/cardiovascular-diseases>
4. Centers for Disease Control and Prevention (CDC). Heart Disease Facts. Available at: <https://www.cdc.gov/heartdisease/facts.htm>
5. World Health Organization (WHO). Cardiovascular Diseases (CVDs). Available at: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
6. Roy, S., Aktar, A., Aleem, F., & Chakrabarty, A. (2019). ECG data analysis and heart disease prediction using machine learning algorithms. Proceedings of 2019 IEEE Region 10 Symposium.
7. Jin, B., Che, C., Liu, Z., Zhang, S., Yin, X., & Wei, X. (2018). Predicting the Risk of Heart Failure with EHR Sequential Data Modeling. *IEEE Access, Volume 6*.
8. Javeed, A., Zhou, S., Yongjian, L., Qasim, I., Noor, A., Nour, R., Wali, S., & Basit, A. (2017). An Intelligent Learning System based on Random Search Algorithm and Optimized Random Forest Model for Improved Heart Disease Detection. *IEEE Access, Volume 4*.
9. Muhammad, Y., Tahir, M., Hayat, M., et al. (2020). Early and accurate detection and diagnosis of heart disease using intelligent computational model. *Scientific Reports, 10*, 19747. Available at: <https://doi.org/10.1038/s41598-020-76635-9>
10. Full Stack Python. Flask Documentation. Available at: <https://www.fullstackpython.com/flask.html>