### A Minor Project report entitled On

**TASK MANAGEMENT SYSYEM (EFFI TASKER)**

### In partial fulfillment of the requirements for the award of

## BACHELOR OF TECHNOLOGY

### In

## Computer Science and Engineering (Data Science)

### Submitted by

## GAGANA (21E51A6701)

## VIKAS (21E51A6705)

## K. CHETAN (21E51A6722)

## B. SHRIYA (21E51A6753)

### Under the Esteemed guidance of

## MS. ROHINI JADHAV

## Assistant Professor



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### (DATA SCIENCE)

## HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Gowdavelly (Village), Medchal, Hyderabad, Telangana, 501401

(UGC Autonomous, Affiliated to JNTUH, Accredited by NAAC (A+) and NBA)

## 2024-2025

## HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(UGC Autonomous, Affiliated to JNTUH, Accredited by NAAC (A+) and NBA)

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### (DATA SCIENCE)



## CERTIFICATE

This is to certify that the Major Project entitled “**Task Management System (EFFI TASKER)**" is being submitted by **A. Gagana** bearing hall ticket number **21E51A6701**, **B. Vikas** bearing hall ticket number **21E51A6705**, **K. Chetan** bearing hall ticket number **21E51A6722**, **B. Shriya** bearing hall ticket number **21E51A6753,** in partial fulfilment of the requirements for the degree **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)** by the Jawaharlal Nehru Technological University, Hyderabad, during the academic year 2024-2025. The matter contained in this document has not been submitted to any other University or institute for the award of any degree or diploma.

#### Under the Guidance of Head of the Department

#### Ms. Rohini Jadhav Dr. M.V.A Naidu

#### Assistant Professor Professor & HoD

#### Internal Examiner External Examiner

## HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(UGC Autonomous, Affiliated to JNTUH, Accredited by NAAC (A+) and NBA)

### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### (DATA SCIENCE)



## DECLARATION

We “**A. Gagana, B. Vikas, K. Chetan, B. Shriya**” students of ‘Bachelor of Technology in CSE (DATA SCIENCE)’, session: 2024- 2025, Hyderabad Institute of Technology and Management, Gowdavelly, Hyderabad, Telangana State, hereby declare that the work presented in this Minor Project entitled ‘Task Management System(EFFI TASKER)’ is the outcome of our bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of engineering ethics. It contains no material previously published or written by another person nor material that has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

## GAGANA (21E51A6701) B.VIKAS (21E51A6705)

## K. CHETAN (21E51A6722)

## B. SHRIYA (21E51A6753)

**ACKNOWLEDGEMENT**

An endeavour of a long period can be successful only with the advice of many well-wishers. We would like to thank our chairman, **SRI. ARUTLA PRASHANTH**, for providing all the facilities to carry out Project Work successfully. We would like to thank our Principal **DR. S. ARVIND**, who has inspired lot through their speeches and providing this opportunity to carry out our Minor Project successfully. We are very thankful to our Head of the Department, **DR.**

**M.V.A NAIDU** and B-Tech Project Coordinator **Ms. RICHA TIWARI**. We would like to specially thank my internal supervisor **Ms. ROHINI JADHAV,** our technical guidance constant encouragement and enormous support provided to us for carrying out our Minor Project. We wish to convey our gratitude and express sincere thanks to all **D.C (DEPARTMENTAL COMMITTEE)** and **P.R.C (PROJECT REVIEW COMMITTEE)** members, non-teaching staff for their support and Co-operation rendered for successful submission of our Minor Project work.

## A. GAGANA (21E51A6701)

## B. VIKAS (21E51A6705)

## K. CHETAN (21E51A6722)

## B. SHRIYA (21E51A6753)

**TABLE OF CONTENTS**

[LIST OF FIGURES i](#_TOC_250003)

[ABSTRACT ii](#_TOC_250002)

1. [INTRODUCTION. 1](#_TOC_250001)
2. LITERATURE SURVEY 2
3. [PURPOSE AND SCOPE 4](#_TOC_250000)
   1. OVERVIEW OF TASK MANAGEMENT SYSTEM
   2. IMPLICATIONS OF TASK MANAGEMENT
   3. PROPOSED SOLUTION

## METHODOLOGY 10

* 1. WHAT IS METHODOLOGY?
  2. METHODOLOGY TO BE USED
  3. TEST METHODOLOGY

## REQUIREMENTS 12

* 1. SOFTWARE REQUIREMENTS
  2. HARDWARE REQUIREMENTS
  3. OPERATING SYSTEM REQUIREMENTS
  4. INSTALLATION

## ARCHITECTURE 16

* 1. BLOCK DIAGRAM

## IMPLEMENTATION. 17

* 1. WORKING AND IMPLEMENATATION
  2. CODE

## RESULT 26

* 1. FINAL RESULT

## CONCULSION. 33

## REFERENCES 34

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **NAME OF THE FIGURE** | **PAGE NO.** |
| FIG 6.1 | Block Diagram | 16 |
| FIG 8.1.1 | Home Page | 26 |
| FIG 8.1.2 | User Login | 26 |
| FIG 8.1.3 | User Dashboard | 27 |
| FIG 8.1.4 | Update Task | 27 |
| FIG 8.1.5 | Apply Leave | 28 |
| FIG 8.1.6 | Leave Status | 28 |
| FIG 8.1.7 | User Registration | 29 |
| FIG 8.1.8 | Database | 29 |
| FIG 8.1.9 | Admin Login | 30 |
| FIG 8.2 | Admin Dashboard | 30 |
| FIG 8.2.1 | Create Task | 31 |
| FIG 8.2.2 | Manage Task | 31 |
| FIG 8.2.3 | Leave Application | 32 |

# ABSTRACT

Chronic health conditions such as cardiovascular diseases, neurological disorders like Parkinson's disease, and metabolic diseases like diabetes pose significant health risks. Early detection is crucial for effective treatment, but the increasing population and limited healthcare infrastructure often hinder timely diagnosis. As a result, many individuals remain unaware of their health conditions until symptoms become severe. Traditional diagnostic methods generally focus on one disease at a time, making comprehensive health assessments both time-consuming and less efficient.

The **Multi-Disease Prediction System** aims to address these challenges by enabling the simultaneous evaluation of multiple health conditions. Using advanced techniques such as Predictive Modeling, Deep Neural Networks, Bio-Inspired Optimization Algorithms, and Streamlit for interactive visualization, the platform offers users fast, holistic health assessments. This empowers individuals to proactively manage their health and seek timely medical consultation.

**Keywords:** Random Forest, SVM, XGboost, Predictive Analytics, Deep Neural Networks, Bio-Inspired Optimization, Chronic Disease Forecasting, Health Assessment, Early Detection, Streamlit.

# 1. INTRODUCTION

### **1.1 Coronary Artery Disease (CAD)**

Coronary artery disease, also known as coronary heart disease, is one of the most common cardiovascular disorders. It arises when the coronary arteries, responsible for supplying oxygen-rich blood to the heart muscle (myocardium), become narrowed or blocked due to a buildup of cholesterol, fats, and other substances, forming plaques. This plaque accumulation, a condition termed **atherosclerosis**, reduces or obstructs blood flow, leading to reduced oxygen delivery to the heart muscle.

* **Causes**: The primary cause of CAD is atherosclerosis, driven by factors like high cholesterol levels, high blood pressure, smoking, diabetes, obesity, and a sedentary lifestyle. Poor dietary habits and chronic stress also contribute to plaque formation.
* **Symptoms**: Symptoms may vary between men and women. Men typically experience angina, or chest pain, especially during physical exertion or stress. Women, however, may have symptoms such as nausea, fatigue, breathlessness, and back or jaw pain along with chest discomfort.
* **Complications**: If untreated, CAD can result in severe complications, including myocardial infarction (heart attack), arrhythmias (irregular heartbeats), heart failure, and, in extreme cases, sudden cardiac arrest. CAD is also a leading cause of disability due to its impact on daily functioning and activity levels.
* **Diagnosis and Treatment**: Diagnosis typically involves an electrocardiogram (ECG), stress tests, echocardiograms, coronary angiography, and blood tests. Treatment may include lifestyle modifications, medications like statins, beta-blockers, and in some cases, surgical procedures like angioplasty or coronary artery bypass grafting.

### **1.2 Parkinson’s Disease**

Parkinson’s disease is a chronic, progressive neurodegenerative disorder affecting the central nervous system, primarily impacting motor function. It results from the gradual degeneration of neurons in the **substantia nigra**, a region in the brain responsible for producing dopamine—a neurotransmitter essential for controlling movement and coordination.

* **Causes**: The exact cause of Parkinson’s remains unclear, but it is believed to involve a combination of genetic and environmental factors. Certain genetic mutations have been linked to early-onset Parkinson’s. Environmental factors, such as exposure to pesticides or head injuries, may also increase risk.
* **Symptoms**: Parkinson's disease typically begins with subtle symptoms that intensify over time. Early signs may include a mild tremor in one hand, stiffness, or difficulty in initiating movements. As the disease progresses, symptoms often include:
  + **Tremor**: Involuntary shaking, typically starting in one hand.
  + **Bradykinesia**: Slowed movements, making simple tasks difficult and time-consuming.
  + **Muscle Rigidity**: Stiffness that can affect any part of the body, limiting range of motion.
  + **Impaired Balance and Coordination**: Increased risk of falls and difficulty in maintaining balance.
  + Non-motor symptoms such as mood disorders, sleep disturbances, memory issues, and fatigue are also common.
* **Complications**: Complications of Parkinson’s can include mobility limitations, cognitive decline, swallowing difficulties, and susceptibility to infections. It also increases the risk of depression, anxiety, and isolation due to the challenges associated with movement and speech.
* **Diagnosis and Treatment**: Diagnosis is primarily based on clinical evaluation, neurological examination, and symptom history. Imaging tests like MRI and DAT scans may be used to rule out other conditions. While Parkinson’s has no cure, treatments like dopamine replacement therapy (e.g., levodopa), physical therapy, and, in advanced cases, deep brain stimulation (DBS) can help manage symptoms.

### **1.3 Diabetes Mellitus**

Diabetes mellitus is a group of metabolic disorders characterized by persistent high blood glucose levels, resulting from the body’s inability to produce or use insulin effectively. Insulin, a hormone produced by the pancreas, regulates blood sugar by facilitating glucose uptake by cells for energy production.

* **Types of Diabetes**:
  + **Type 1 Diabetes**: An autoimmune disorder where the body’s immune system attacks insulin-producing beta cells in the pancreas. This type often begins in childhood or adolescence and requires lifelong insulin therapy.
  + **Type 2 Diabetes**: The most common type, often developing in adulthood, where the body becomes resistant to insulin or fails to produce enough. Risk factors include obesity, a sedentary lifestyle, genetic predisposition, and poor dietary habits.
  + **Gestational Diabetes**: Occurs during pregnancy and may resolve after delivery, although it increases the risk of type 2 diabetes later in life.
  + **Prediabetes**: A condition where blood sugar levels are elevated but not yet high enough for a diabetes diagnosis. It serves as a warning sign for type 2 diabetes.
* **Symptoms**: Diabetes symptoms vary depending on the severity and type, but common signs include:
  + **Frequent urination** and excessive thirst due to high blood sugar levels.
  + **Extreme hunger** and fatigue, as glucose cannot enter cells effectively.
  + **Blurry vision**, weight loss, and slow healing of wounds.
  + In type 1 diabetes, symptoms can develop rapidly, whereas type 2 may progress more slowly over time.
* **Complications**: Chronic high blood glucose can lead to severe complications, including:
  + **Cardiovascular Disease**: Increased risk of heart attack, stroke, and atherosclerosis.
  + **Neuropathy**: Nerve damage leading to pain, numbness, or weakness, especially in the feet and hands.
  + **Retinopathy**: Damage to the retina, potentially causing blindness.
  + **Nephropathy**: Kidney damage that may lead to kidney failure.
  + **Foot Complications**: Poor circulation and nerve damage increase the risk of infections, potentially resulting in amputations.
* **Diagnosis and Treatment**: Diagnosis involves blood tests, such as fasting blood glucose, HbA1c, and oral glucose tolerance tests. Treatment varies depending on the type of diabetes.

Type 1 diabetes requires insulin therapy, while type 2 may be managed with lifestyle changes, oral medications (like metformin), and sometimes insulin. Gestational diabetes typically resolves postpartum but requires careful monitoring and diet control during pregnancy.

The **Multi-Disease Detection System** is developed to offer users a thorough evaluation of their susceptibility to various chronic ailments. With the increasing population and mounting pressure on healthcare resources—amplified by recent global health crises such as the pandemic—many individuals face delays in obtaining timely medical diagnoses. This often results in people remaining unaware of critical health conditions until their symptoms become severe. Such delays are particularly fatal for chronic diseases, where early detection is vital. This platform is designed to help users proactively manage their health by providing rapid risk assessments.

Utilizing Machine Learning (ML), Deep Learning (DL), and Bio-Inspired Optimization Algorithms, the system allows individuals to input their health data and receive timely predictions regarding their likelihood of developing specific diseases. By integrating Artificial Intelligence (AI), the platform harnesses advanced algorithms that simulate human cognition, enabling faster, more accurate evaluations compared to traditional methods.

The rise of Deep Learning (DL), a subfield of AI that mirrors the structure and function of the human brain, has had a transformative impact across industries, with healthcare being one of the primary beneficiaries. DL excels in analyzing large-scale, unstructured data such as medical imaging, genetic information, and clinical records, making it indispensable in areas like medical research, disease detection, and personalized treatment. This platform employs DL to identify patterns and anomalies in health data, enabling the early identification of chronic conditions like cardiovascular diseases, neurodegenerative disorders (e.g., Parkinson’s), and metabolic diseases (e.g., diabetes), potentially preventing the onset of life-threatening complications.

By leveraging cutting-edge AI and DL technologies, the platform addresses the critical need for faster, more accessible healthcare solutions in an era of growing medical challenges.

**2. LITERATURE SURVEY**

**Mallula et al. (2024)** introduced a predictive framework for diagnosing multiple chronic conditions, including metabolic diseases (diabetes), cardiovascular disorders, renal dysfunctions, neurodegenerative conditions (Parkinson’s), and oncological risks (breast cancer). The platform leverages models like Support Vector Machine (SVM), Logistic Regression, and deep learning frameworks via TensorFlow with Keras, deploying these through Streamlit Cloud for ease of user interaction and disease selection. The models achieved high precision for individual conditions (e.g., 97% for renal disease), showcasing the potential of these algorithms for single-condition diagnostics.

Despite these advancements, the system lacks the capability for integrated multi-disease assessments, which could provide deeper insights into comorbidities and interconnected health risks. The model selection primarily focuses on established algorithms without incorporating adaptive bio-inspired techniques, limiting the platform's ability to handle imbalanced datasets or dynamically adjust to diverse medical data characteristics. Additionally, the emphasis on singular disease prediction restricts the platform’s potential to deliver holistic health analyses for complex cases where multiple conditions coexist.

**Kallepalli et al. (2024)** presented a Multi-Disease Prediction System (MDPS) leveraging Logistic Regression and Support Vector Machines (SVM), deployed via a Streamlit interface, to forecast diseases such as diabetes, cardiovascular conditions, and Parkinson’s. With data sourced and refined from Kaggle, the system achieved accuracies of 78% for diabetes, 85% for cardiovascular risks, and 89% for Parkinson’s, underscoring machine learning’s role in early detection, personalized care, and efficient healthcare resource management.

However, this study primarily targets single-disease predictions, limiting its ability to account for complex comorbidities or interdependencies between diseases, which are critical for comprehensive health assessments. Additionally, the focus on traditional algorithms like SVM and Logistic Regression restricts scalability, as these may underperform with multi-layered, non-linear data relationships. The absence of advanced deep learning techniques also restricts adaptability and accuracy.

**Kumar et al. (2024)** developed a Multiple Disease Prediction System (MDPS) utilizing machine learning algorithms such as Logistic Regression for cardiovascular health, Convolutional Neural Networks (CNN) for cancer diagnosis, and Support Vector Machines (SVM) for diabetes prediction. This approach emphasizes early detection and personalization, aiming to identify risk patterns in medical data for diabetes, cancer, and heart disease through symptom and vitals analysis.

While effective in addressing individual disease risks, this framework has notable limitations. Primarily, it focuses on discrete predictions of each disease separately, limiting its utility in identifying interactions or combined effects among multiple health conditions. The reliance on traditional ML algorithms such as Logistic Regression and SVM further restricts adaptability to complex comorbid cases. Additionally, while CNN is employed for cancer imaging, the absence of other advanced deep learning methods across all conditions restricts holistic, interconnected disease predictions.

**Banoth et al. (2024)** developed an Automated Multi-Disease Prediction System that employs machine learning techniques, including decision trees, support vector machines (SVM), and random forests, to forecast the likelihood of various diseases based on patient symptoms and medical histories. With an accuracy rate exceeding 95%, the system is positioned as a valuable tool for early disease detection, personalized treatment, and improving healthcare outcomes through data-driven predictions.

Despite its effectiveness, the system has certain limitations. Primarily, it treats each disease prediction independently, neglecting the potential interactions or co-occurrence of multiple health conditions, which are critical for a more holistic understanding of patient risk. Furthermore, the reliance on conventional machine learning models such as decision trees, SVM, and random forests restricts the system’s capacity to capture intricate, non-linear relationships within complex healthcare datasets. Additionally, the framework does not address incorporating real-time patient data or continuous monitoring, which could further enhance prediction precision and enable timely interventions. To achieve a more robust and personalized healthcare solution, integrating advanced algorithms, such as deep learning models, is recommended to facilitate more accurate multi-disease prediction and more comprehensive insights into disease interactions.

**Mathew et al. (2024)** proposed a system for predicting multiple diseases using machine learning algorithms, emphasizing the integration of various data types and models for improved healthcare outcomes. Their approach highlights the importance of feature selection, model evaluation, and multi-data fusion, which enhances the accuracy of disease predictions. Additionally, the system's focus on improving patient-doctor interactions through remote consultations aims to make healthcare more accessible, efficient, and cost-effective.

However, several limitations in the framework remain. First, the system does not explicitly address the handling of complex and interrelated comorbidities, which is crucial for a more comprehensive understanding of patient health. By focusing primarily on isolated disease predictions, it misses the opportunity to explore how diseases may influence each other or share common risk factors. Moreover, while the integration of remote consultations enhances accessibility, the model’s prediction capabilities may not fully incorporate real-time patient data or continuous monitoring, which could improve the timeliness and precision of diagnoses. Further research could benefit from exploring advanced deep learning models or real-time data integration to better capture the intricate relationships among multiple diseases, providing a more holistic and personalized approach to healthcare.

# 3. PURPOSE AND SCOPE

#### 3.1 OVERVIEW OF INTEGRATED HEALTH RISK PREDICTION SYSTEM

Chronic diseases like cardiovascular conditions, neurological disorders (e.g., Parkinson's disease), and metabolic illnesses (e.g., diabetes) continue to represent a significant global health burden. Timely detection and diagnosis are essential to improving patient outcomes; however, increasing populations, limited healthcare resources, and traditional diagnostic systems focusing on single conditions hinder early detection. The **Multi-Disease Prediction System** aims to resolve these issues by offering a platform that provides a holistic assessment of multiple chronic diseases at once. By leveraging state-of-the-art techniques such as Predictive Analytics, Deep Neural Networks, Bio-Inspired Optimization Algorithms, and Streamlit for interactive data visualization, the system empowers users to take proactive control of their health. This system allows individuals to receive rapid, accurate evaluations, enabling early detection and timely medical intervention for multiple conditions, all within a user-friendly interface.

Key components of the system include:

1. **Predictive Modeling and Multi-Disease Forecasting:** Utilizing advanced algorithms, the system predicts the likelihood of individuals developing multiple health conditions based on their medical history, lifestyle factors, and other relevant data. This approach offers a comprehensive health risk assessment, identifying individuals who may benefit from further medical evaluation or lifestyle adjustments.
2. **Deep Neural Networks and Bio-Inspired Optimization:** The system uses sophisticated deep learning models to analyze vast amounts of health data, improving prediction accuracy. Bio-inspired optimization techniques are employed to fine-tune the model parameters, enhancing overall performance and providing more reliable forecasts.
3. **Interactive Data Visualization with Streamlit:** The platform provides users with an intuitive interface powered by **Streamlit**, making it easy for individuals to interact with the data, visualize their health predictions, and understand potential risks through graphical representations. This visualization feature aims to enhance user engagement and understanding of complex health predictions.
4. **Early Detection and Prevention:** The primary goal of the Integrated Health Risk Prediction System is to facilitate early detection of chronic diseases, enabling users to act before conditions become critical. By integrating multiple disease predictions into one platform, the system saves time and resources, allowing users to receive holistic health insights in one go.

#### 3.2 IMPLICATIONS OF THE INTEGRATED HEALTH RISK PREDICTION SYSTEM:

The **Multi-Disease Prediction System** has profound implications on both individual health management and broader healthcare practices. Below are the key aspects of its potential impact:

1. **Enhanced Preventive Healthcare:** By providing a holistic, multi-condition health risk prediction, the system encourages users to adopt preventive measures before the onset of chronic conditions. Users can receive alerts regarding potential health risks, prompting timely lifestyle changes and early medical interventions that may prevent the progression of diseases like cardiovascular conditions and diabetes.
2. **Improved Accuracy and Efficiency in Diagnostics:** Traditional diagnostic methods often rely on sequential testing for individual conditions. This system improves diagnostic efficiency by offering simultaneous predictions for multiple diseases, reducing the time spent on testing and enabling faster treatment. The use of deep learning models ensures higher accuracy, making predictions more reliable than conventional methods.
3. **Empowerment of Users for Proactive Health Management:** The system empowers users by providing real-time, actionable health insights that allow them to make informed decisions about their health. With clear visualizations and data-backed predictions, individuals can monitor their health risk and engage in proactive management, potentially reducing future healthcare costs and improving their quality of life.
4. **Scalability and Adaptability in Healthcare Settings:** The system can be scaled to accommodate larger datasets, enabling its use in various healthcare settings such as hospitals, clinics, and wellness centers. It is adaptable to different demographics, which is particularly useful for tailoring health assessments to specific population groups based on factors like age, gender, and geographical location.
5. **Data-Driven Decision Making for Healthcare Providers:** For healthcare professionals, this system provides a wealth of predictive data to guide decision-making and patient care strategies. By receiving data-backed health predictions, doctors can make better-informed recommendations for preventative care, early diagnosis, and personalized treatment plans for their patients.
6. **Cost-Effective Healthcare Solutions:** By improving early detection and prevention, the system can significantly reduce the burden on healthcare systems. Early intervention typically reduces the need for more expensive treatments for advanced stages of chronic diseases. This cost-saving aspect makes the system valuable for both individual users and healthcare organizations.
7. **Global Health Impact:** As the system is designed to be accessible and scalable, it has the potential to reach global populations, particularly in underserved regions with limited access to healthcare facilities. Its predictive capabilities can be a game-changer in countries where healthcare access is restricted, providing an affordable solution for improving public health outcomes.

## PROPOSED SOLUTION:

The Multi-Disease Prediction System is an innovative tool aimed at revolutionizing healthcare by forecasting the likelihood of three major health conditions: cardiovascular disorders, diabetes mellitus, and Parkinson's disease. Utilizing a variety of advanced machine learning techniques, this system evaluates and predicts the risk of these diseases, allowing for a comparative analysis of the models' performance and their prediction accuracies. Designed to be user-friendly, the platform enables individuals to enter their health information and receive predictions regarding their susceptibility to these chronic conditions, thus providing valuable early warnings for proactive health management.

### **3.3.1 Heart Disease Prediction Using Ensemble Learning and Ant Colony Optimization**

Cardiovascular diseases (CVDs), including Coronary Artery Disease (CAD), Myocardial Infarction (MI), Hypertension, Arrhythmias, and Heart Failure, remain leading causes of morbidity and mortality across the globe. Early diagnosis and accurate risk prediction play a vital role in reducing the burden of these conditions. In this project, Ensemble Learning algorithms are applied alongside **Ant Colony Optimization (ACO)** to predict the likelihood of CVD based on key clinical and demographic features.

#### Ensemble Learning Techniques for CVD Prediction

Ensemble learning refers to a method of combining multiple machine learning models to improve prediction accuracy and robustness. Here, we use five distinct ensemble algorithms to classify whether a patient is at risk of developing heart disease.

1. **Adaboost (Adaptive Boosting)**: Adaboost enhances the performance of weak classifiers by combining them to form a strong model. A weak classifier performs slightly better than random guessing. In this context, a decision stump (a decision tree with a depth of one) is used as the weak classifier. The goal of Adaboost is to iteratively focus on the misclassified instances from the previous iterations, making it an effective method for identifying subtle yet crucial patterns in cardiovascular health, such as atherosclerotic plaque build-up and coronary vessel occlusions.
2. **Bagging (Bootstrap Aggregating)**: Bagging involves training multiple models on random subsets of the data and combining their predictions. It reduces variance by averaging the outcomes of individual models. For cardiac conditions, this helps in mitigating overfitting, ensuring that the model generalizes well on new data, even when predicting conditions like hypertensive heart disease or valvular heart disease.
3. **Random Forest**: Random Forest is a powerful ensemble learning method that builds multiple decision trees during the training phase and aggregates their outputs. This technique works well for predicting complex cardiovascular conditions, such as heart failure or ischemic heart disease, as it captures intricate relationships among various risk factors, including age, serum lipid levels, family history of cardiovascular disease, and blood pressure.
4. **Gradient Boosting**: Gradient Boosting is an iterative method that adds new decision trees to correct the mistakes made by the previous trees. It is particularly useful in identifying cardiac arrhythmias or myocardial infarctions since it focuses on misclassified data points and builds progressively stronger models. This technique can effectively highlight critical indicators like heart rate variability and ECG anomalies.
5. **Extra Trees (Extremely Randomized Trees)**: Extra Trees constructs multiple trees like Random Forest but introduces more randomness during the construction of each tree. This randomness helps in reducing overfitting, making it ideal for predicting multifactorial diseases like coronary artery disease or dilated cardiomyopathy, where interactions between various clinical variables can be complex.

Then we combine these classifiers with Ant Colony Optimization

**Ant Colony Optimization (ACO)** is utilized to enhance ensemble learning models for feature selection on a Cardiovascular Disorder dataset (commonly known as Heart Disease). The key components involved in this optimization process are:

* **Performance Metric**: This class stores the efficacy of a solution derived from the algorithm. It tracks how well the selected attributes contribute to the model's overall performance.
* **Link**: This class represents a link in a graph, encapsulating details such as its origin, destination, weight, and pheromone concentration. The link represents the connection between two attributes in the dataset being considered for selection.
* **Network**: The Network class models the relationships between various attributes in the dataset. It holds the number of nodes (features), the links between them, and methods to add links, retrieve the weight of a link, access pheromone concentrations, set pheromone levels, and calculate the total weight of a path through the network.
* **Complete Network**: A subclass of the Network class, Complete Network is designed to generate a fully connected network based on a weight matrix. This matrix defines the intensity of relationships between all features, enabling the ACO to explore every potential feature combination.
* **Ant Individual**: The Ant Individual class simulates the behavior of a single ant in the ACO algorithm. Each ant has a current feature, a set of selected features (solution), and the cost and accuracy of that solution. The ant also has methods to obtain and set the current feature, retrieve its solution, and calculate the cost and accuracy of the solution.
* **ACO (Ant Colony Optimization):** This class implements the core logic of the ACO algorithm, applied for feature selection. It requires the input of several parameters: a network representing the dataset's features, the number of ants, alpha (pheromone influence), beta (heuristic influence), the number of iterations, evaporation rate (pheromone decay), and the number of feature subsets (FS). This class contains various methods:
  + **initialize\_ants**: Randomly assigns starting features to each ant.
  + **initialize\_pheromone:** Initializes the pheromone concentration of each link to a value based on the inverse of the number of nodes multiplied by the link's weight.
  + **update\_pheromone:** Updates the pheromone levels on the links after each iteration.
  + **select\_next\_feature:** Chooses the next feature for the ant to include in its solution, based on a combination of pheromone levels and heuristic data (such as cosine similarity between features).
  + **construct\_solution:** Constructs a feature selection solution for each ant by iteratively selecting features.
  + **run\_ACO:** The main method to execute the ACO algorithm for a specified number of iterations, building solutions for each ant and updating pheromone concentrations based on solution quality.
  + **print\_parameters**: Displays the ACO algorithm's configuration parameters.

The ACO algorithm operates by initializing a network where the number of nodes corresponds to the number of features in the dataset. Each ant starts at a random node (feature) and builds a solution by selecting the next feature (node) based on the pheromone concentrations and the heuristic data (i.e., cosine similarity between feature pairs). The heuristic function assesses the similarity between features, with highly similar features more likely to be selected together in the solution.

After each solution is constructed, the ACO algorithm evaluates the efficacy of the selected feature set in the ensemble learning model. The pheromone concentrations on the links are updated using the evaporation mechanism, where a portion of the pheromone dissipates over time. Additionally, pheromone is deposited by each ant on the links it traversed, with the amount of pheromone being proportional to the accuracy of the solution.

The process repeats for a given number of iterations, with each iteration enhancing the solutions as ants explore different feature subsets and modify the pheromone concentrations. The optimal feature subset is returned after all iterations, representing the best features for the dataset based on ACO optimization.

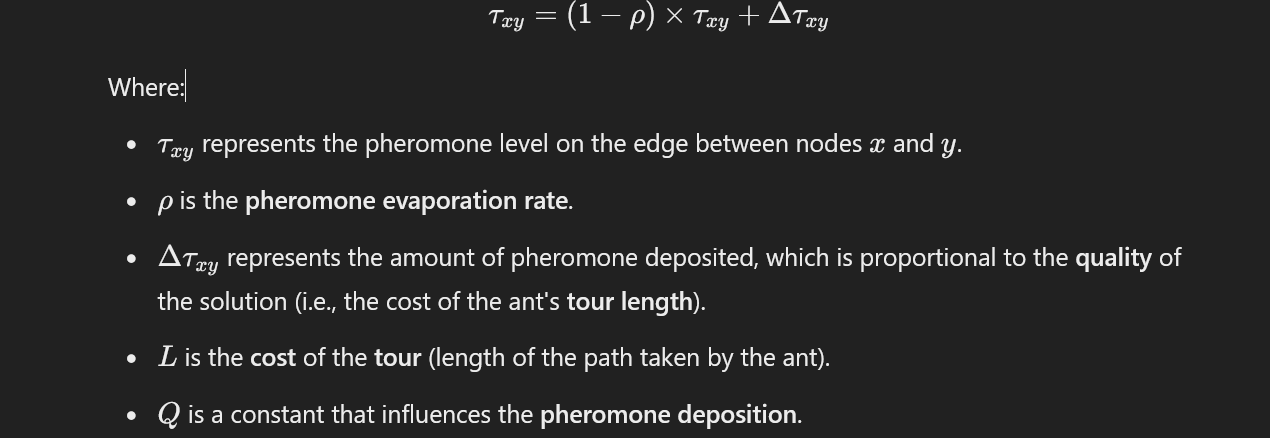
|  |  |  |
| --- | --- | --- |
| ***Parameter*** | ***Value*** | ***Importance*** |
| *Number of vertices of the graph* | *13* | *Number of features in dataset is13 for Heart Disease* |
| *Number of Ants* | *50* | *50-100Antsare ideal for this case.* |
| *Rate of Evaporation* | *0.2* | *High rate of evaporation can lead to faster evaporation of pheromone hence the following ants might lose their way to the food source. Hence an optimal rate of evaporation is 0.2-0.3* |
| *Alpha Heuristic* | *1* | *Importance of pheromone* |
| *Beta Heuristic* | *1* | *Importance of heuristic information* |
| *Number of iterations* | *100* | *100-200 iterations are optimal as a higher number of iterations means more exploration of paths and finding the most optimal shortest path* |

Ant Colony Optimization (ACO) is a metaheuristic optimization technique inspired by the foraging behavior of ants. Ants are renowned for their ability to find the most efficient route between their nest and a food source. They accomplish this by leaving a trail of pheromones as they move towards the food source, which attracts other ants to follow the same path. This process of laying and following pheromone trails forms the basis of the ACO algorithms. These algorithms have been extensively researched and applied to solve a wide range of optimization problems. They have demonstrated significant effectiveness in finding optimal solutions in various domains such as logistics, communications, and industrial engineering.

* Initially, there are two paths leading to the food from the colony. At the start, there is no pheromone laid on either path. Therefore, the probability of choosing one path over the other is equal, i.e., 50%. Suppose two ants choose two different paths towards the food source. The probability of selecting each path is equally likely.
* The distances of the two paths are not the same. The ant following the shorter path will reach the food faster than the other.
* Once the ant finds the food, it carries the food back to the colony. While retracing its steps, it deposits pheromones on the ground. The ant on the shorter path will return to the colony sooner.
* When the third ant sets out to search for food, it is more likely to follow the path with the higher pheromone concentration, which is the shorter path. Since the shorter path has more pheromone laid down than the longer path, the third ant is more likely to choose it.
* By the time the ant following the longer path returns to the colony, several ants would have already taken the shorter path, further increasing the pheromone concentration along it. Consequently, when another ant sets out to find food, it will see that both paths have similar pheromone levels. At this point, it randomly chooses one.
* With continued iterations, the shorter path accumulates a higher pheromone concentration, leading to a higher probability of ants following it. Eventually, after several rounds, the shorter path becomes the dominant path, and all ants will follow this path.

### **Pheromone Update Formula:**

The formula for **pheromone update** is given by:



### **Procedure of the ACO Metaheuristic:**

1. **Solution Generation**: In the first step, each artificial ant generates a solution.
2. **Comparison of Paths**: In the second step, the paths found by the different ants are compared for quality.
3. **Pheromone Update**: In the third step, the pheromone levels on the paths are updated based on the quality of the solutions found.

The **ACO\_MetaHeuristic** algorithm follows a repeated process to improve the solution:

*procedure ACO\_MetaHeuristic is*

*while not\_termination do*

*Generate Solutions()*

*daemonActions()*

*Pheromone Update()*

*repeat*

*end procedure*

### **Components of an Ant Colony Optimization (ACO) Algorithm:**

* **Problem Representation**: The problem is modeled as a graph or network, where the nodes represent the decision variables or problem parameters, and the edges represent the relationships or connections between them.
* **Solution Construction:** Ants build a solution by traversing the graph. At each node, an ant probabilistically selects the next node to visit, influenced by the pheromone trail that has been laid on the edges. The selection process is weighted by the amount of pheromone present, guiding the ants toward more promising solutions.
* **Pheromone Update**: As ants traverse the graph, they deposit pheromones on the edges they follow. Over time, the pheromone gradually evaporates, and the pheromone level is updated according to the quality of the solution discovered, reinforcing the more successful paths.
* **Local Search:** After constructing an initial solution, a local search procedure can be applied to refine and enhance the solution's quality further. This helps in improving the feasibility and optimality of the solution.
* **Termination:** The algorithm stops when a predefined stopping criterion is met, such as a maximum number of iterations or when the pheromone level reaches a specified threshold. This marks the completion of the optimization process.

### **Factors Affecting the Performance of Ant Colony Optimization Algorithm:**

* **Number of Ants:** Increasing the number of ants enhances the diversity of the solutions and improves the exploration of the search space, but it also raises computational costs due to the larger number of solutions to process.
* **Amount of Pheromones:** The quantity of pheromones deposited on the paths directly impacts the probability of ants selecting those paths. More pheromone increases the likelihood that ants will follow a particular path, guiding the search toward optimal or near-optimal solutions.
* **Evaporation Rate:** The evaporation rate of pheromone determines how quickly the deposited pheromone fades over time. A higher evaporation rate increases the exploration (searching new paths), while a lower rate promotes exploitation (sticking with known good paths), thus influencing the exploration-exploitation balance of the algorithm.
* **Local Search Procedure:** The effectiveness of the local search method used to refine the solutions significantly impacts the convergence of the algorithm. A well-designed local search can lead to higher-quality solutions by fine-tuning the paths or solutions found by the ants.

|  |  |  |
| --- | --- | --- |
| ***Parameter*** | ***Value*** | ***Importance*** |
| *Population Size* | *50* | *This parameter specifies the size of the population, i.e., the number of candidate solutions. A larger population size can increase the diversity of the solutions, but it may also lead to slower convergence and higher computational costs.* |
| *Awareness Probability* | *0.02* | *This parameter controls the probability of a crow selecting a specific feature for optimization. It ranges between0and1 and is used in the update\_position() function to decide whether a feature is updated based on the local best or randomly generated values.* |
| *Flight Length* | *2* | *This parameter controls the step size of the search process. It isused to balance between local and global search. A higher value of fL increases the global search capability of the algorithm, but it may also decrease the accuracy of the local search.* |
| *Number of Iterations* | *100* | *This parameter specifies the maximum number of iterations that the algorithm can run. It determines how long the algorithm will search for the optimal solution.* |
| *Target Function* | *Fitness* | *This parameter specifies the maximum number of iterations that the algorithm can run. It determines how long the algorithm will search for the optimal solution.* |
| *Minimum Value* | *[5,1]* | *This parameter isa list that specifies the minimum values of the search space for each feature. It ensures that the search does not go beyond the lower bounds of the problem domain.* |
| *Maximum Value* | *[120, 3]* | *This parameter isa list that specifies the maximum values of the search space for each feature. It ensures that the search does not go beyond the upper bounds of the problem domain.* |

**3.1.2 Diabetes Diagnosis via Supervised Learning Enhanced by Crow Search Optimization**

This model leverages three core supervised learning algorithms—K-Nearest Neighbors (KNN), Random Forest, and Logistic Regression—optimized using the Crow Search Algorithm (CSA) for accurate diabetes prediction. K-Nearest Neighbors is a basic yet powerful machine-learning technique commonly used for classification tasks, particularly in applications like pattern recognition, data mining, and intrusion detection. As a supervised model, KNN assigns a class to a data point based on the majority class of its nearest neighbors, making it a robust tool for various real-world problems.

Logistic Regression is another widely used supervised learning method, particularly for classification problems involving binary outcomes. It is designed to model the probability of an instance belonging to a particular class, establishing a relationship between independent variables and a binary dependent variable. Logistic regression is instrumental for decision-making tasks, such as identifying whether an email is spam or non-spam, based on certain features of the message.

Random Forest is a versatile and popular machine-learning algorithm that excels in both classification and regression problems. As an ensemble learning method, it combines multiple decision trees to make predictions, improving the model's robustness and accuracy in complex scenarios by aggregating the outputs of several trees.

The Crow Search Algorithm (CSA) is a nature-inspired optimization method based on the collective foraging behavior of crows. Introduced in 2014 by Yang and Deb, CSA is used to solve optimization problems across a wide array of fields, including engineering and data analysis. Optimization tasks aim to identify the most optimal solution from a vast search space, a challenge that becomes difficult due to the problem's complexity. Metaheuristic optimization algorithms like CSA are designed to address these challenges by efficiently exploring large solution spaces.

CSA operates by simulating the collaborative nature of crows when foraging for food. It begins by creating a population of crows, each representing a potential solution to the problem. The crows communicate by sharing both local and global information—local information comes from each crow’s individual experiences, while global information stems from the best solution found so far. Crows then adjust their positions in the search space based on this shared knowledge, striving to locate the optimal solution. The CSA utilizes three main operators: exploration, exploitation, and memorization. Exploration helps find new potential solutions by examining unexplored areas, while exploitation focuses on refining the best-found solutions by zeroing in on promising regions. Memorization involves each crow storing the best solution found to date in its memory, ensuring that the best solution is retained throughout the process.

The behavioral principles of CSA are inspired by the natural habits of crows:

1. Crows operate in groups or flocks.
2. They remember where they have hidden food.
3. They engage in “food stealing” by observing and following other crows.
4. They protect their food stores from other birds.

CSA has proven effective in a variety of domains, including optimization in engineering design, scheduling tasks in distributed systems, and fine-tuning parameters in image processing algorithms. For example, CSA has been used to enhance the design of hybrid energy storage systems, optimize task scheduling algorithms in computational systems, and adjust filter parameters for digital image processing.

The strengths of CSA include its simplicity in implementation, fast convergence to solutions, and ability to handle high-dimensional optimization problems. It is also highly resilient to noise and capable of solving multimodal optimization problems, where multiple local optima exist. However, CSA has some limitations, such as the possibility of getting stuck in local optima and experiencing slower convergence rates as the algorithm progresses.

Though CSA is a relatively new optimization technique, it offers substantial potential for further development. Future improvements could involve creating hybrid models that integrate CSA with other optimization methods, potentially enhancing performance. Moreover, the optimization efficiency of CSA could be increased through the use of dynamic control strategies and adaptive parameters. The potential applications of CSA are vast and can be extended to a variety of fields, including data science and machine learning, where optimization plays a critical role in problem-solving.

**Parkinson’s Disease Diagnosis via Voice Data Using Support Vector Machine Algorithm**

Approximately 90% of individuals with Parkinson’s disease (PD) experience speech impairments, making voice analysis one of the most widely utilized methods for diagnosing this condition. This technique involves employing a specialized algorithm to assess voice data for signs indicative of PD. The process begins by selecting the most relevant features from the pool of extracted voice characteristics. Subsequently, a Support Vector Machine (SVM)-based model is deployed for classification, distinguishing between healthy individuals and those with Parkinson's disease.

Support Vector Machines (SVMs) are supervised machine learning algorithms that can be applied to both classification and regression problems. The core concept of SVM is to identify a hyperplane that best separates the different classes within the training dataset. This separation is achieved by selecting a hyperplane with the maximum margin, which is defined as the distance between the hyperplane and the nearest data points from each class. Once the optimal hyperplane is established, new instances can be classified by determining which side of the hyperplane they fall on. SVMs are particularly effective when dealing with datasets that contain a large number of features or when there is a distinct margin separating the classes.

**4. METHODOLOGY**

## WHAT IS METHODOLOGY?

The methodology for a Multi-Disease Prediction System targeting cardiovascular disease (heart disease), diabetes mellitus, and Parkinson’s disease involves a systematic framework for the design, development, and verification of prediction models for each condition. This structured approach includes phases such as clinical requirement analysis, algorithm selection and design, data preprocessing, model training, performance evaluation, and deployment in a healthcare setting. This comprehensive methodology ensures a scientifically validated, user-friendly system that leverages machine learning to enable proactive patient care and intervention.

## METHODOLOGY TO BE USED

1. **Clinical Requirement Analysis**: In this initial phase, collaboration with medical professionals, researchers, and healthcare stakeholders will guide the understanding of each disease’s diagnostic requirements and clinical characteristics. A needs assessment, involving interviews, surveys, and literature reviews, will ensure that the system is designed to meet specific diagnostic and prognostic requirements for cardiovascular conditions, diabetes, and neurodegenerative disorders like Parkinson’s. This step ensures that the system is medically relevant and aligned with current healthcare practices.
2. **Planning and System Architecture**: After gathering requirements, a comprehensive project plan and technical blueprint will be developed, detailing timeframes, resource allocation, data sources, and key deliverables. The design phase will include constructing wireframes and architectural diagrams to outline the system components and user interfaces. Prototypes for disease-specific predictive models and data visualization tools will be developed for stakeholder feedback.
3. **Data Acquisition and Preprocessing**: Data for cardiovascular disease, diabetes mellitus, and Parkinson’s disease will be gathered from clinical datasets, healthcare databases, and open-source repositories. This phase involves data cleaning, normalization, and feature engineering to ensure accuracy in training the predictive models. Medical-specific preprocessing will include identifying critical biomarkers and other diagnostic indicators specific to each disease, ensuring that model training reflects clinical realities and variances in each condition.
4. **Machine Learning Model Development**: Disease-specific machine learning models will be developed using ensemble learning techniques such as Random Forest, XGBoost, and Gradient Boosting. For each disease:
   * **Cardiovascular Disease**: The model will be tailored to detect early signs of coronary artery disease and other cardiovascular disorders. Feature selection, guided by Ant Colony Optimization (ACO), will prioritize relevant indicators such as blood pressure, cholesterol levels, and ECG data.
   * **Diabetes Mellitus**: The model will focus on metabolic indicators and risk factors, including glucose levels, insulin resistance, BMI, and genetic predisposition.
   * **Parkinson’s Disease**: Feature selection will concentrate on neurological indicators, voice modulation characteristics, and motor function metrics, facilitating early prediction and monitoring of disease progression.
5. **Testing and Validation**: Rigorous testing is crucial to validate the predictive accuracy, reliability, and real-world performance of each model. The system will undergo:
   * **Unit Testing** for individual components.
   * **Integration Testing** to verify seamless interaction between data inputs, feature selection algorithms, and disease models.
   * **Validation Testing** to ensure each model’s predictive accuracy across varied datasets, covering cardiovascular, diabetic, and neurodegenerative profiles. Accuracy will be assessed through clinical metrics such as sensitivity, specificity, and area under the ROC curve.
6. **System Deployment and Training**: Upon successful testing, the system will be deployed in a controlled clinical environment, ensuring compatibility with healthcare workflows. Training sessions will be provided to healthcare providers, familiarizing them with system functionalities, result interpretation, and follow-up recommendations. This phase ensures ease of use, with guidance on incorporating the system into patient care protocols.
7. **Maintenance and System Updates**: After deployment, the system will be regularly monitored and updated based on user feedback, new medical research, and evolving clinical guidelines. Regular updates will refine model accuracy and adapt the system to new diagnostic indicators, ensuring sustained relevance and effectiveness.

## TEST METHODOLOGY

1. **Unit Testing**: Each disease-specific component, such as feature extraction and algorithm processing, will be tested independently to validate its functionality. Unit tests will confirm that each model operates correctly and produces accurate intermediate outputs for cardiovascular, diabetic, and Parkinson’s datasets.
2. **Integration Testing**: This phase will evaluate the interactions between system modules, ensuring a cohesive flow from data ingestion, feature selection, and disease prediction to results visualization. Integration testing verifies that all components function as a unified system.
3. **Functional Testing**: The system will be tested against specific medical requirements, such as accurate risk prediction for each disease. Test cases will simulate clinical scenarios, verifying that the model effectively distinguishes between healthy and at-risk individuals and produces appropriate predictions.
4. **User Acceptance Testing (UAT)**: Healthcare professionals will test the system in a simulated clinical environment to validate its practical applicability. UAT will provide feedback on usability and identify any usability challenges or clinical requirements, allowing for final adjustments prior to deployment.
5. **Performance Testing**: This phase assesses system stability, response times, and scalability to ensure efficient processing of large datasets and multiple disease predictions. Performance metrics, such as latency, resource utilization, and predictive response time, will be optimized.
6. **Security Testing**: Given the handling of sensitive patient information, security testing will assess the system’s defenses against data breaches and vulnerabilities. Measures such as secure data storage, encryption, and access control will be implemented to ensure compliance with data protection standards.
7. **Regression Testing**: As the system undergoes updates, regression testing will confirm that existing functionalities remain unaffected by new changes. This ensures ongoing accuracy and reliability.
8. **Documentation of Test Results**: Throughout the testing process, all test outcomes, including test cases, observed results, and identified issues, will be documented comprehensively. This documentation supports transparency and accountability in the testing process, serving as a valuable reference for system maintenance and future improvements.

**5. REQUIREMENTS AND INSTALLATION**

## SOFTWARE REQUIREMENTS:

1. **Development Technologies**
   * **Python**: Primary programming language for machine learning algorithms and data processing.
   * **Streamlit**: A Python-based web framework used to quickly build and deploy interactive web applications for the prediction system.
   * **Scikit-learn**: A library for machine learning, providing algorithms for classification and prediction.
   * **TensorFlow / Keras** (Optional): Deep learning libraries for advanced models.
   * **Pandas**: For data manipulation and analysis, handling large datasets.
   * **NumPy**: For numerical computing and array manipulation.
   * **Matplotlib / Seaborn**: Libraries for data visualization and generating charts/graphs.
2. **User Interface**
   * **Web-Based Interface**: A responsive, interactive web interface built using Streamlit, allowing users to input health data and view prediction results in real-time.
3. **Functional Features**
   * **Disease Prediction**: An AI-driven model that predicts the likelihood of various diseases based on input data (such as heart disease, diabetes, and liver disease).
   * **User Management**: Features like user registration, login, and password management.
   * **Data Input**: Forms for users to input their health-related information, such as symptoms, age, and medical history.
   * **Prediction Results**: Display of the predicted outcomes for the diseases based on the user input.
   * **Model Training**: Regular updates of the machine learning model using new data.

## HARDWARE REQUIREMENTS:

## The hardware requirements for the Task Management System are essential to ensure optimal performance, reliability, and scalability. Below are the key hardware specifications needed for both the server environment and end-user devices.

## Processor: A multi-core processor (e.g., Intel Xeon or AMD Ryzen) to handle concurrent requests efficiently.

## RAM: A minimum of 8 GB of RAM, with 16 GB recommended for handling higher loads and multiple simultaneous users.

## Storage: SSD (Solid State Drive) with at least 256 GB of storage capacity to ensure fast data access and retrieval. Additional storage may be required for backups and data archiving.

## Network Interface: A reliable network interface card (NIC) with at least 1 Gbps capacity for handling network traffic efficiently.

## OPERATING SYSTEM REQUIREMENTS:

For the Task Management System, the following operating system requirements are specified, focusing solely on Windows environments:

**Windows**:

* Windows 10 or later for desktop clients to ensure compatibility with modern web browsers.

**macOS**:

* Latest versions of macOS for Apple users, supporting common web browsers.

**Linux**:

* Popular distributions such as **Ubuntu** or **Fedora** for users who prefer open-source operating systems.
  1. **INSTALLATION:**

### **Jupyter Notebook Installation Steps**

#### Step 1. Install Python and pip

* Download Python from [python.org](https://www.python.org/downloads/), and check "Add Python to PATH" during installation.
* Verify with:

*python --version*

*pip --version*

#### Step 2. Install Jupyter Notebook

* **Via pip**:

*pip install notebook*

* **Via Anaconda** (recommended for data science):
  + Download and install Anaconda from anaconda.com.

#### Step 3. Launch Jupyter Notebook

* Open a terminal and type:

*jupyter notebook*

* This opens Jupyter in your default web browser.

#### Step 4. Optional: Install Additional Libraries

* Use pip to install commonly used libraries:

*pip install numpy pandas matplotlib scikit-learn seaborn*

**Install Python:** Download and install Python 3.x from the official Python website (<https://www.python.org/>).

#### Steps for python installation:

**Step 1:** Download the python installer from official website.

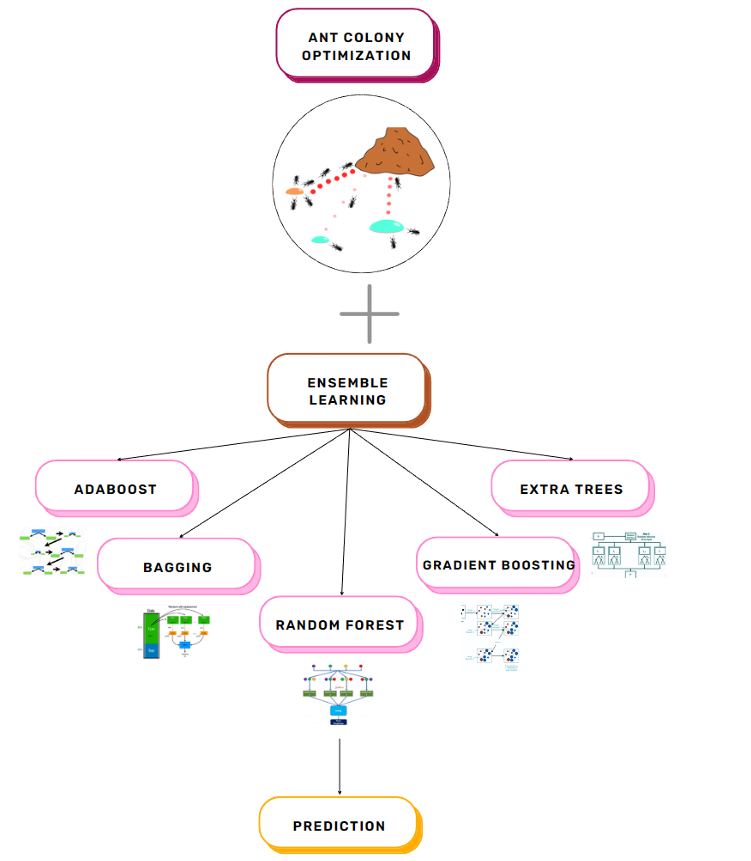
**Step 2:** Running the Executable installer. Click on the use two checkboxs below. Hit the “Install now” option.

**Step 3:** Then you come to the optional features page. Where you can select any of the four checkboxs according to your needs. Click on “Next”.

**Step 4:** Then you come to advanced options page. Where you can select any of the seven checkboxes and customize installation location. After selection, Click on “Install”.

**Step 5:** Setup is complete and python is installation is complete.

1. **ARCHITECTURE**

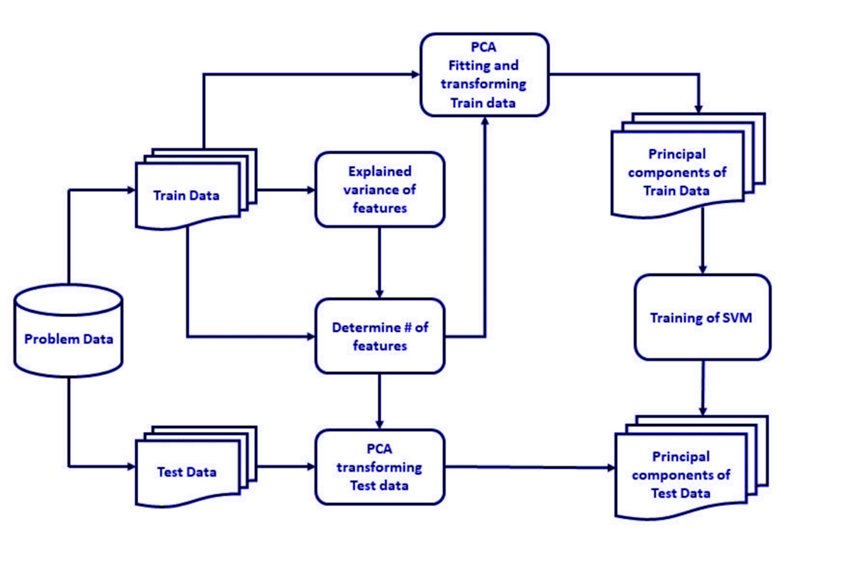


## FIG 6.1: HEART DISEASE PREDICTION- ENSEMBLE LEARNING WITH ANT COLONY OPTIMIZATION

## 

## 

## FIG 6.2: DIABETES PREDICTION- SUPERVISED LEARNING WITH CROW SEARCH OPTIMIZATION



## FIG 6.2: DIABETES PREDICTION- SUPERVISED LEARNING WITH CROW SEARCH OPTIMIZATION

**7. IMPLEMENTATION**

## WORKING AND IMPLEMENTATION

## System Overview:

## The Multi-Disease Prediction System is a diagnostic tool designed to predict the likelihood of several chronic diseases, including cardiovascular conditions, neurological disorders like Parkinson's, and diabetes. This system uses machine learning and optimization techniques to evaluate multiple health indicators simultaneously, providing a comprehensive health risk assessment. Targeted towards individuals and healthcare providers, it aims to facilitate early detection and proactive health management.

## System Architecture

## Backend: The backend is built using machine learning models, combining various classification algorithms and optimization techniques to enhance accuracy. Predictive models, including deep neural networks, evaluate user input parameters for each disease, and advanced bio-inspired optimization methods, such as Ant Colony Optimization and the Crow Search Algorithm, help refine model performance.

## Optimization Techniques: To improve prediction accuracy and computational efficiency, the system incorporates:

## Ant Colony Optimization: Used for feature selection to improve the accuracy and relevance of model inputs.

## Crow Search Algorithm: Applied to further optimize model parameters, enhancing prediction outcomes.

## Ensemble Learning: Combines outputs from multiple models to increase robustness and accuracy of predictions.

## Frontend: The frontend interface is developed with Streamlit, providing a user-friendly and interactive experience. Users input relevant health parameters (e.g., blood pressure, glucose levels, etc.), and the system predicts the probability of disease presence, displaying results immediately.

## No Database: The system does not require persistent storage, as it operates in a stateless manner. Data input is temporary, with predictions generated and displayed in real-time based on user inputs.

## Process Flow: Users provide specific health-related parameters, and the system instantly processes this data, applying the trained machine learning models to predict potential health conditions. Results indicate if the user is at risk for specific diseases, allowing for early health interventions.

## Parkison’s Disease:

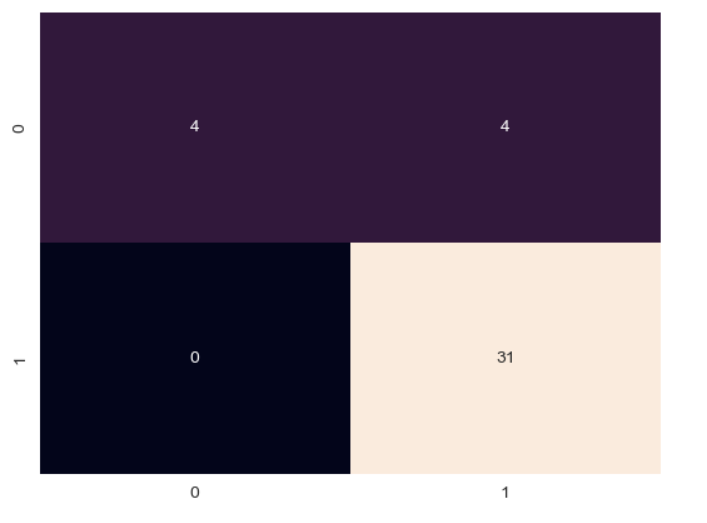
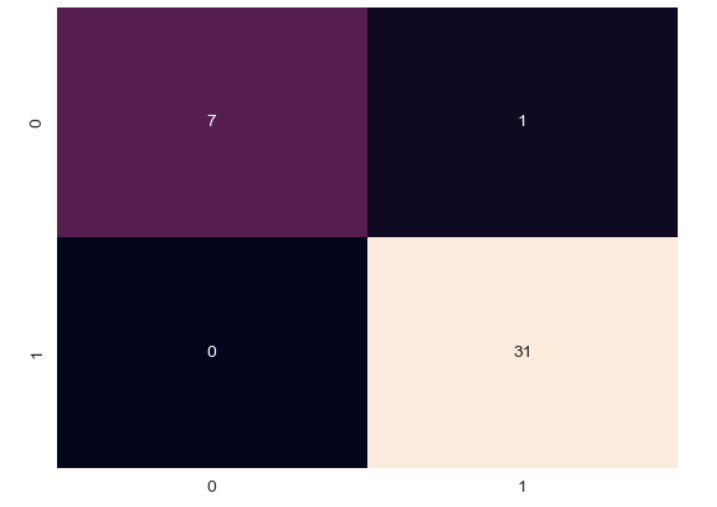
## Important Features

## Here we check the top 10 most important features that contribute to the prediction of parkinson’s

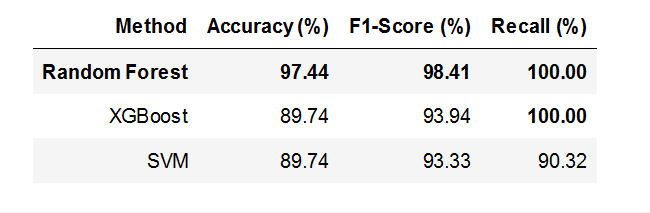
## 

**Confusion Matrix**

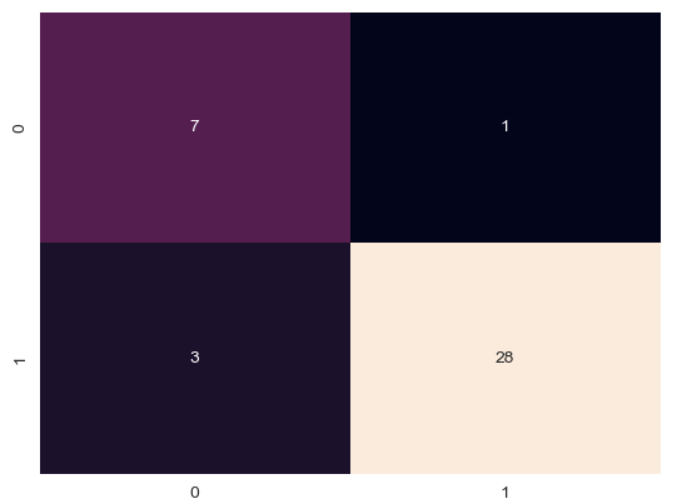
Random Forest: XGboost:



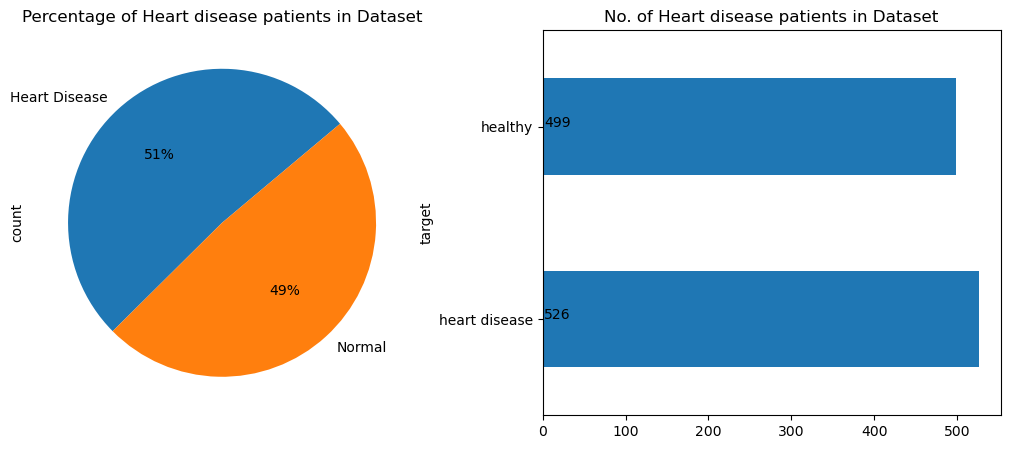
Comparison of Model Accuracy:

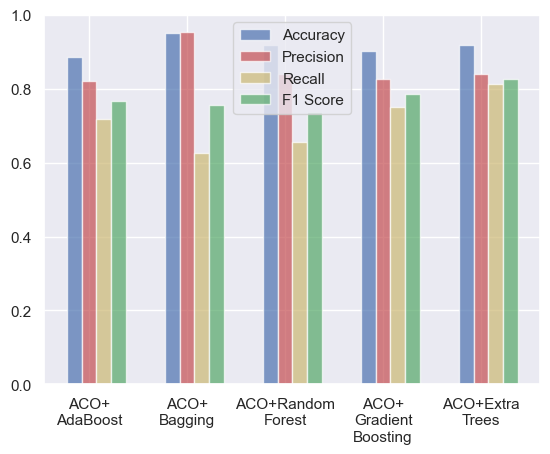


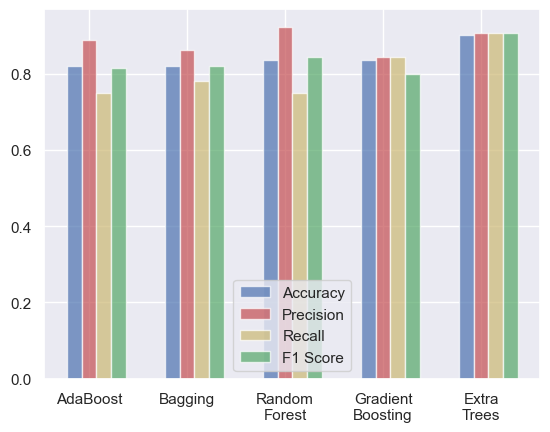
SVM:

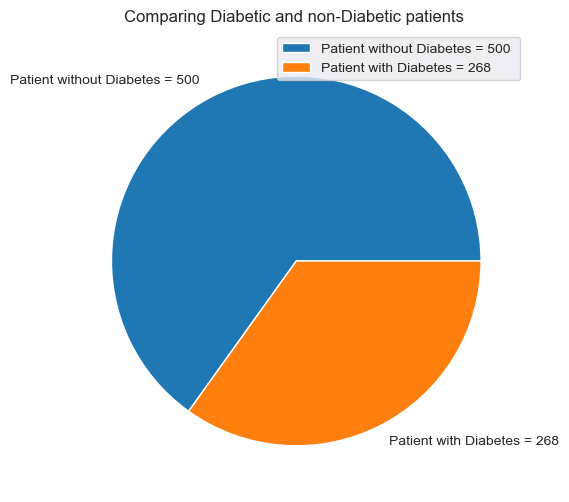


**Prediction Of Heart Disease Using Ensemble Learning & Ant Colony Optimization:**



Comparison of Accuracy, Precision, Recall and F1 score for all the above models:



**Diabetes Prediction- Supervised Learning with Crow Search Optimization:**



## CODE

## Main.Py

## # -\*- coding: utf-8 -\*-

## """

## Created on Fri Dec 29 08:22:49 2023

## @author: madhu

## """

## import pickle

## import streamlit as st

## from streamlit\_option\_menu import option\_menu

## # loading the saved models

## diabetes\_model = pickle.load(open('C:/Users/madhu/OneDrive/Desktop/Mini Project/Saved Models/diabetes\_model.sav', 'rb'))

## heart\_disease\_model = pickle.load(open('C:/Users/madhu/OneDrive/Desktop/Mini Project/Saved Models/heart\_disease\_model.sav','rb'))

## parkinsons\_model = pickle.load(open('C:/Users/madhu/OneDrive/Desktop/Mini Project/Saved Models/parkinsons\_model.sav', 'rb'))

## # sidebar for navigation

## with st.sidebar:

## 

## selected = option\_menu('Multiple Disease Prediction System',

## 

## ['Diabetes Prediction',

## 'Heart Disease Prediction',

## 'Parkinsons Prediction'],

## icons=['activity','heart','person'],

## default\_index=0)

## 

## st.title("Multiple Disease Prediction System")

## 

## # Diabetes Prediction Page

## if (selected == 'Diabetes Prediction'):

## 

## # page title

## st.subheader('Diabetes Prediction')

## 

## 

## # getting the input data from the user

## col1, col2, col3 = st.columns(3)

## 

## with col1:

## Pregnancies = st.text\_input('Number of Pregnancies')

## 

## with col2:

## Glucose = st.text\_input('Glucose Level')

## 

## with col3:

## BloodPressure = st.text\_input('Blood Pressure value')

## 

## with col1:

## SkinThickness = st.text\_input('Skin Thickness value')

## 

## with col2:

## Insulin = st.text\_input('Insulin Level')

## 

## with col3:

## BMI = st.text\_input('BMI value')

## 

## with col1:

## DiabetesPedigreeFunction = st.text\_input('Diabetes Pedigree Function value')

## 

## with col2:

## Age = st.text\_input('Age of the Person')

## 

## 

## # code for Prediction

## diab\_diagnosis = ''

## 

## # creating a button for Prediction

## 

## if st.button('Diabetes Test Result'):

## diab\_prediction = diabetes\_model.predict([[Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age]])

## 

## if (diab\_prediction[0] == 1):

## diab\_diagnosis = 'The person is diabetic'

## else:

## diab\_diagnosis = 'The person is not diabetic'

## 

## st.success(diab\_diagnosis)

## # Heart Disease Prediction Page

## if (selected == 'Heart Disease Prediction'):

## 

## # page title

## st.subheader('Heart Disease Prediction')

## 

## col1, col2, col3 = st.columns(3)

## 

## with col1:

## age = st.text\_input('Age')

## 

## with col2:

## sex = st.text\_input('Sex')

## 

## with col3:

## cp = st.text\_input('Chest Pain types')

## 

## with col1:

## trestbps = st.text\_input('Resting Blood Pressure')

## 

## with col2:

## chol = st.text\_input('Serum Cholestoral in mg/dl')

## 

## with col3:

## fbs = st.text\_input('Fasting Blood Sugar > 120 mg/dl')

## 

## with col1:

## restecg = st.text\_input('Resting Electrocardiographic results')

## 

## with col2:

## thalach = st.text\_input('Maximum Heart Rate achieved')

## 

## with col3:

## exang = st.text\_input('Exercise Induced Angina')

## 

## with col1:

## oldpeak = st.text\_input('ST depression induced by exercise')

## 

## with col2:

## slope = st.text\_input('Slope of the peak exercise ST segment')

## 

## with col3:

## ca = st.text\_input('Major vessels colored by flourosopy')

## 

## with col1:

## thal = st.text\_input('thal: 0 = normal; 1 = fixed defect; 2 = reversable defect')

## 

## # code for Prediction

## heart\_diagnosis = ''

## 

## # creating a button for Prediction

## 

## if st.button('Heart Disease Test Result'):

## heart\_prediction = heart\_disease\_model.predict([[age, sex, cp, trestbps, chol, fbs, restecg,thalach,exang,oldpeak,slope,ca,thal]])

## 

## if (heart\_prediction[0] == 1):

## heart\_diagnosis = 'The person is having heart disease'

## else:

## heart\_diagnosis = 'The person does not have any heart disease'

## 

## st.success(heart\_diagnosis)

## 

## # Parkinson's Prediction Page

## if (selected == "Parkinsons Prediction"):

## 

## # page title

## st.subheader("Parkinson's Disease Prediction")

## 

## col1, col2, col3, col4, col5 = st.columns(5)

## 

## with col1:

## fo = st.text\_input('MDVP:Fo(Hz)')

## 

## with col2:

## fhi = st.text\_input('MDVP:Fhi(Hz)')

## 

## with col3:

## flo = st.text\_input('MDVP:Flo(Hz)')

## 

## with col4:

## Jitter\_percent = st.text\_input('MDVP:Jitter(%)')

## 

## with col5:

## Jitter\_Abs = st.text\_input('MDVP:Jitter(Abs)')

## 

## with col1:

## RAP = st.text\_input('MDVP:RAP')

## 

## with col2:

## PPQ = st.text\_input('MDVP:PPQ')

## 

## with col3:

## DDP = st.text\_input('Jitter:DDP')

## 

## with col4:

## Shimmer = st.text\_input('MDVP:Shimmer')

## 

## with col5:

## Shimmer\_dB = st.text\_input('MDVP:Shimmer(dB)')

## 

## with col1:

## APQ3 = st.text\_input('Shimmer:APQ3')

## 

## with col2:

## APQ5 = st.text\_input('Shimmer:APQ5')

## 

## with col3:

## APQ = st.text\_input('MDVP:APQ')

## 

## with col4:

## DDA = st.text\_input('Shimmer:DDA')

## 

## with col5:

## NHR = st.text\_input('NHR')

## 

## with col1:

## HNR = st.text\_input('HNR')

## 

## with col2:

## RPDE = st.text\_input('RPDE')

## 

## with col3:

## DFA = st.text\_input('DFA')

## 

## with col4:

## spread1 = st.text\_input('spread1')

## 

## with col5:

## spread2 = st.text\_input('spread2')

## 

## with col1:

## D2 = st.text\_input('D2')

## 

## with col2:

## PPE = st.text\_input('PPE')

## 

## 

## 

## # code for Prediction

## parkinsons\_diagnosis = ''

## 

## # creating a button for Prediction

## if st.button("Parkinson's Test Result"):

## parkinsons\_prediction = parkinsons\_model.predict([[fo, fhi, flo, Jitter\_percent, Jitter\_Abs, RAP, PPQ,DDP,Shimmer,Shimmer\_dB,APQ3,APQ5,APQ,DDA,NHR,HNR,RPDE,DFA,spread1,spread2,D2,PPE]])

## 

## if (parkinsons\_prediction[0] == 1):

## parkinsons\_diagnosis = "The person has Parkinson's disease"

## else:

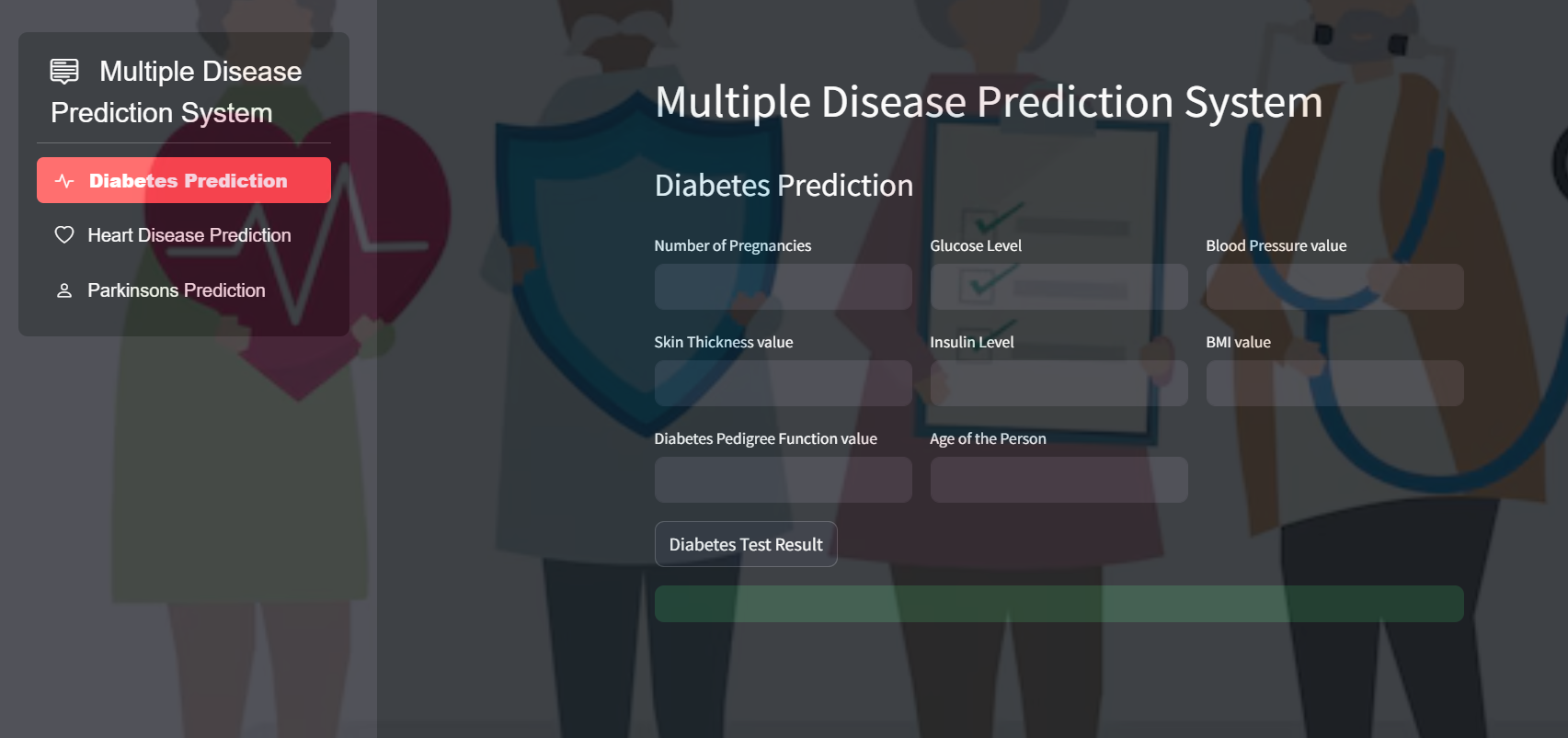
## parkinsons\_diagnosis = "The person does not have Parkinson's disease"

## 

## st.success(parkinsons\_diagnosis)

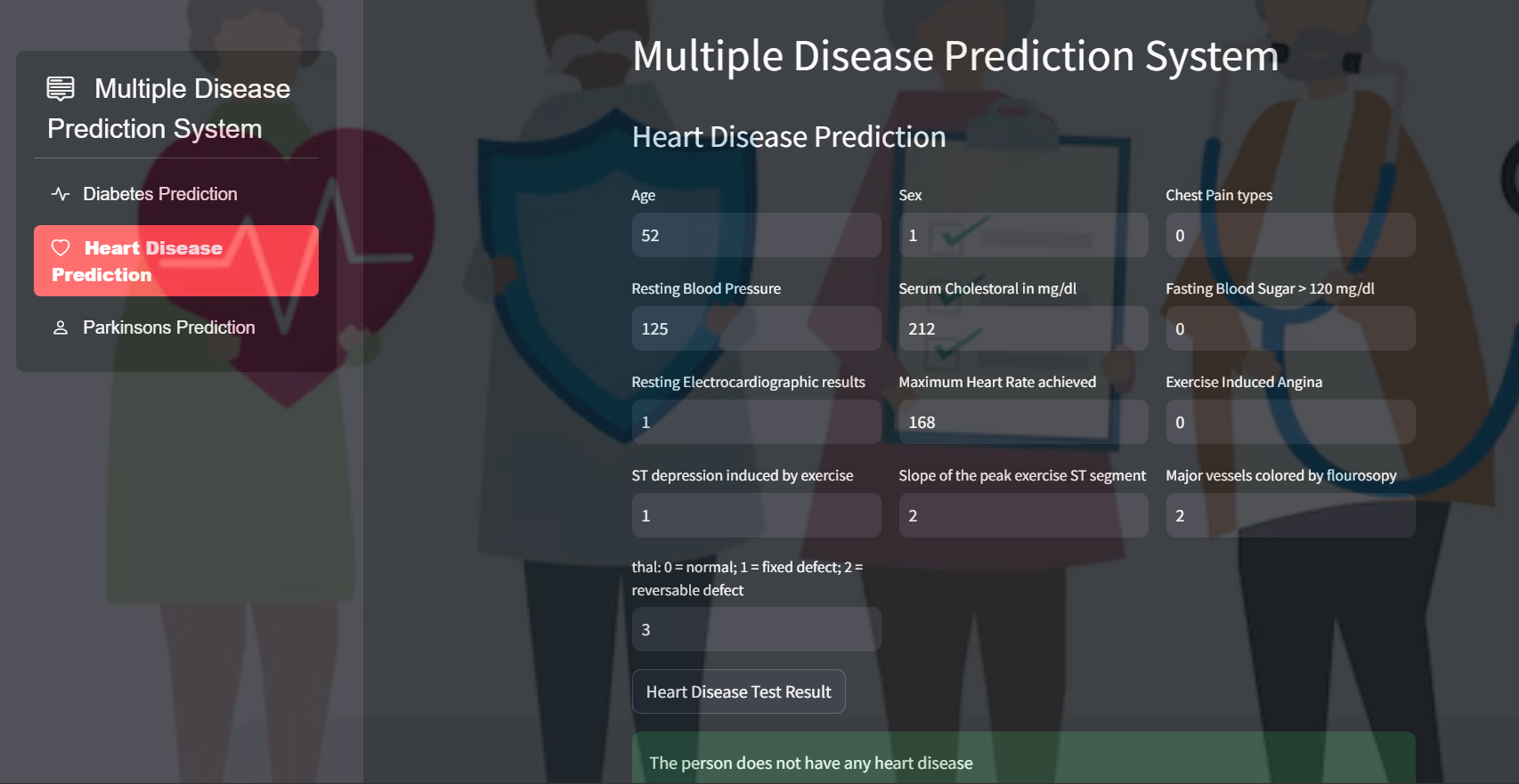
**8. RESULT**

**8.1 FINAL RESULT**

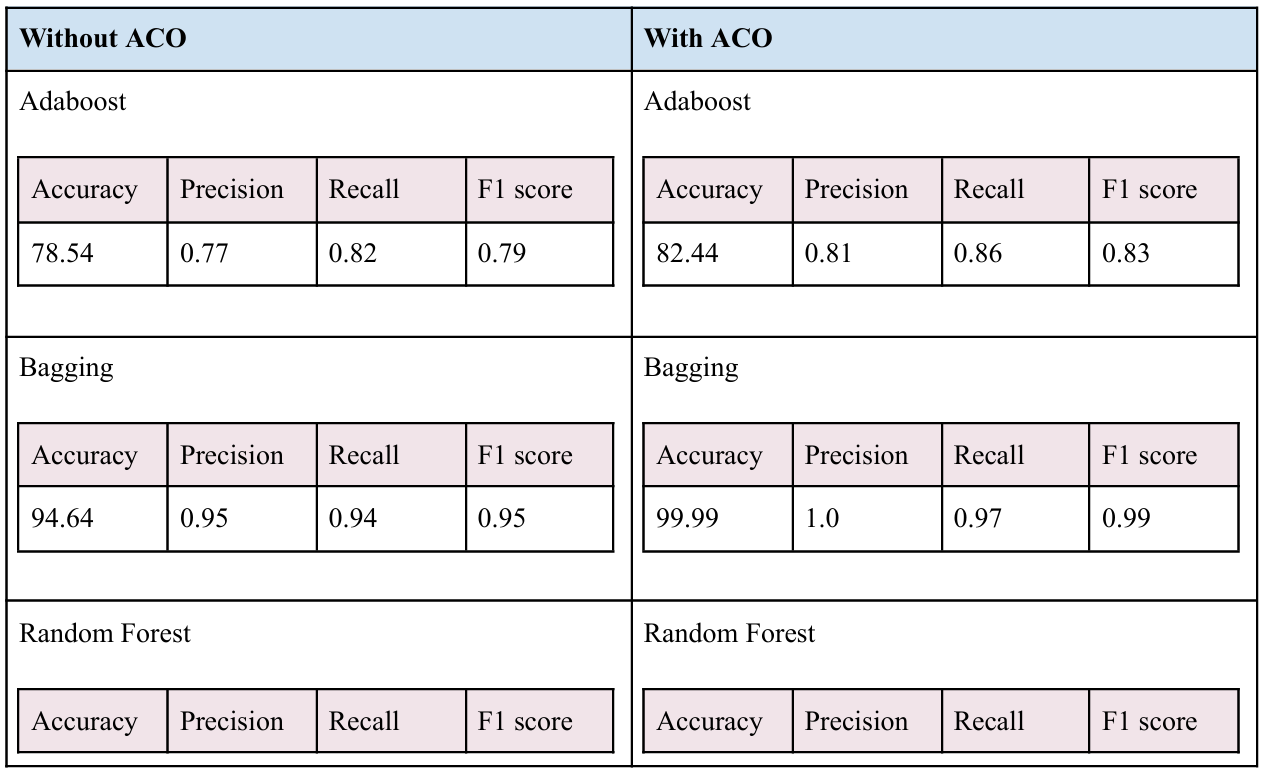


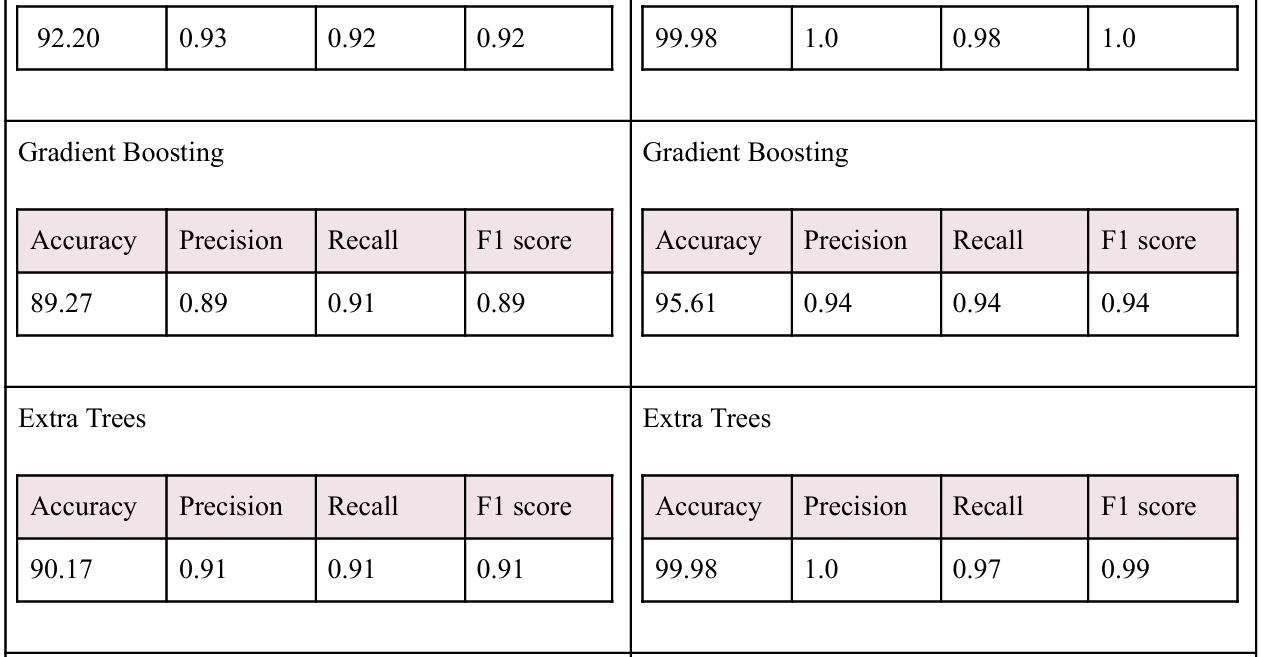
**Fig 8.1.1. HOME PAGE**

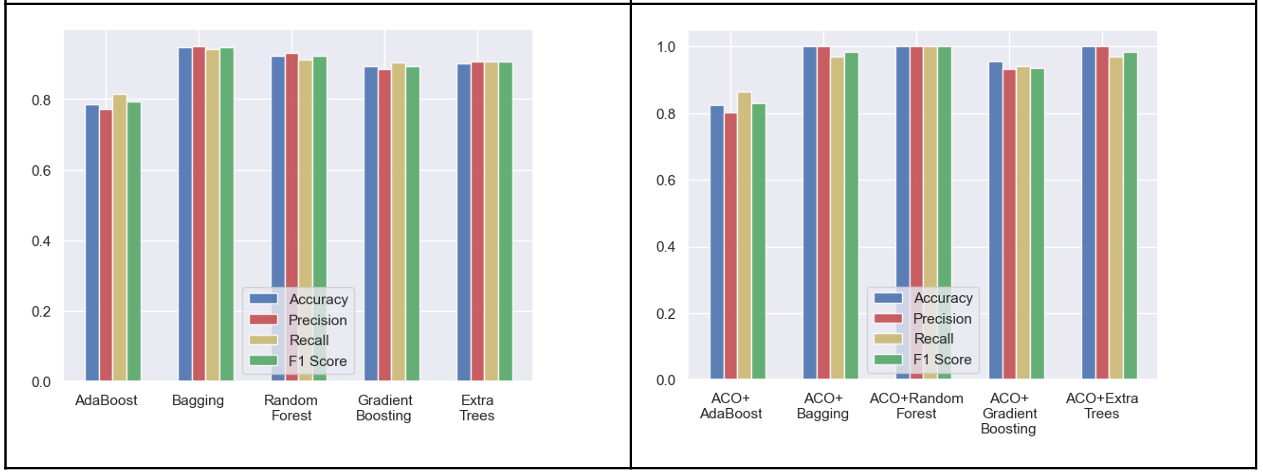
**Heart Disease Prediction- Ensemble Learning with Ant Colony Optimization:**

****

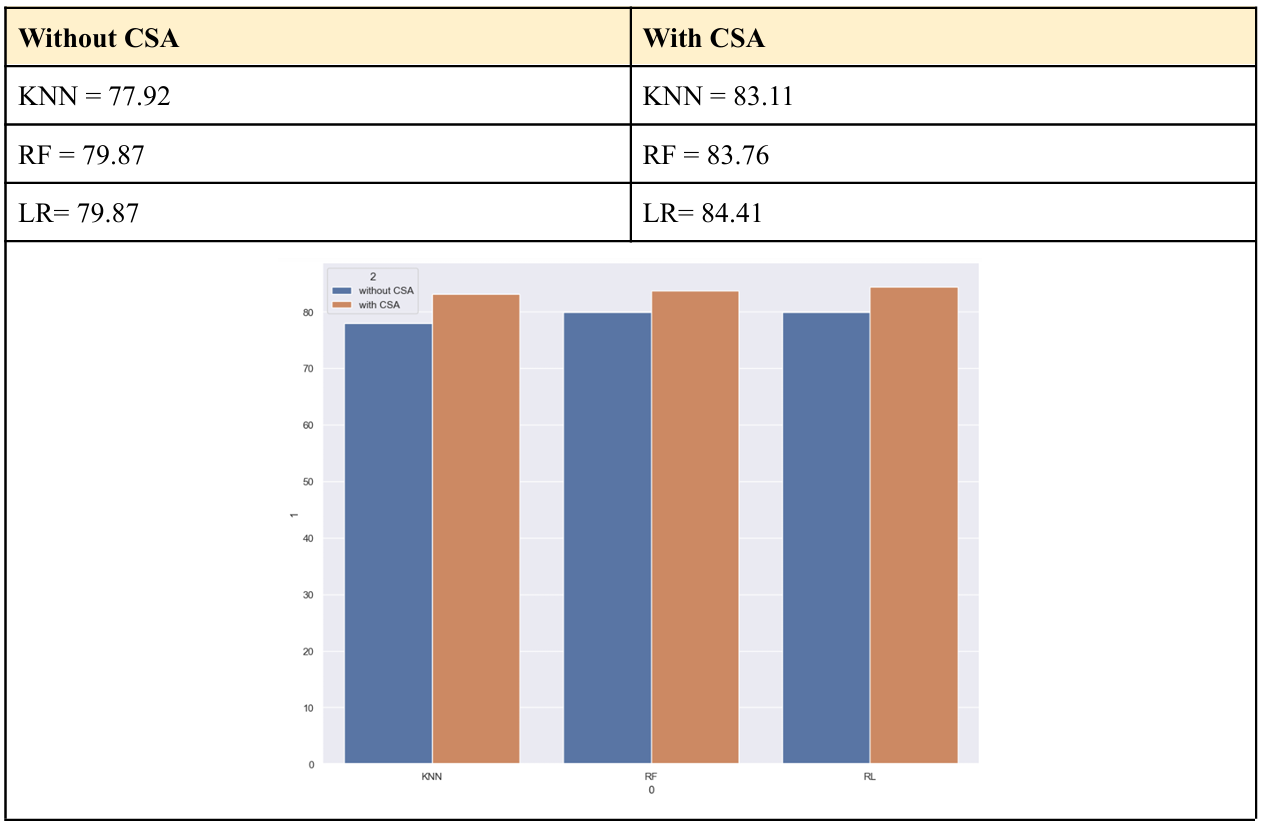
**Fig 8.1.2. HEART DISEASE PREDICTION**



****

****

**Diabetes Prediction- Supervised Learning with Crow Search Optimization:**

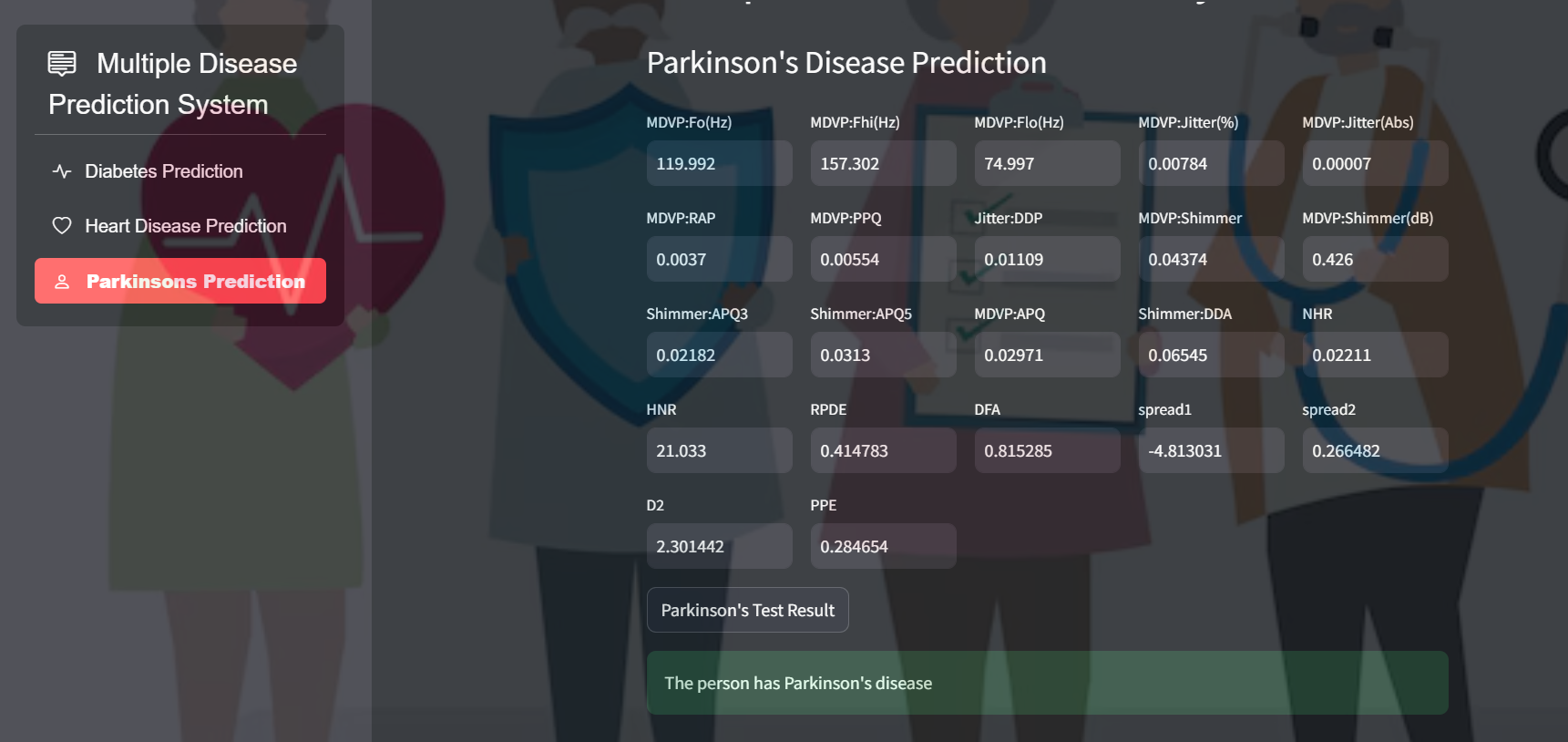
****

****

**Fig 8.1.3. DIABETES PREDICTION**

**Parkinson’s Disease Prediction- Voice data based Support Vector Machine Algorithm:**

Accuracy= 88%



**Fig 8.1.3. USER DASHBOARD**

**9. CONCLUSION**

**10. REFERENCES**