**A Project Report**

Submitted in partial fulfillment of the Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (COMPUTER SCIENCE) (hons.)**

**Specialization – Artificial Intelligence and Machine Learning**

**By**


**Ms. Devika Sanjay Jonjale**

Roll Number: 22


**Under the esteemed guidance of**

**Prof. Ankit Javeri**

**MALAD KANDIVALI EDUCATION SOCIETY'S**

**DEPARTMENT OF COMPUTER SCIENCE OF**

**NAGINDAS KHANDWALA COLLEGE**

**(EMPOWERED AUTONOMOUS)**

**(Reaccredited 'A' Grade by NAAC)**
**(AFFILIATED TO UNIVERSITY OF MUMBAI)**
**(ISO 9001:2015)**


**2023-2024**

**MALAD KANDIVALI EDUCATION SOCIETY'S**

**NAGINDAS KHANDWALA COLLEGE OF COMMERCE,**
**ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS**
**KHANDWALA COLLEGE OF SCIENCE**
**MALAD [W], MUMBAI – 64**

**(EMPOWERED AUTONOMOUS)**

**(Reaccredited 'A' Grade by NAAC)**
**(AFFILIATED TO UNIVERSITY OF MUMBAI)**
**(ISO 9001:2015)**

# CERTIFICATE

This is to certify that the project entitled, **"Health Assistant"**, is bonafide work of **Ms. DEVIKA SANJAY JONJALE** bearing **Roll. No: 22** for the course **FINAL PROJECT/INTERNSHIP AND VIVA** (Course Code: **2365UHAIPR**) submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in COMPUTER SCIENCE (hons.) specialization in ARTIFICIAL INTELLIGENCE and MACHINE LEARNING from University of Mumbai.

**Internal Guide** **Coordinator** **External Examiner**

**Date:** **College Seal**

# DECLARATION

I hereby declare that the project entitled, **"Health Assistant"** done at **NAGINDAS KHANDWALA COLLEGE**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of

**BACHELOR OF SCIENCE (COMPUTER SCIENCE) (hons.) specialization in ARTIFICIAL INTELLIGENCE and MACHINE LEARNING** to be submitted as a final semester project as part of our curriculum.

Ms. **Devika Sanjay Jonjale**

**Name and Signature of the Student**

# ACKNOWLEDGEMENT

I would like to place my sincere gratitude to those who have contributed to the successful completion of this project directly or indirectly.

I would like to thank **Dr. (Mrs.) Ancy Jose, Director - MKES INSTITUTIONS, Prof. Dr. Moushumi Datta, Principal, Prof. Dr. Mona Mehta, Vice Principal & IQAC Coordinator and Prof. Rashmi Tiwari, H.O.D** as well as the Management for their kind co - operation in the completion of my project and for their great support throughout my graduation years.

I take this opportunity to express my profound gratitude and deep regards to my guide **Prof. Ankit Javeri**, without whose guidance & critical appreciation, this project would have been incomplete. Right from its inception, this project has been shaped by his expert opinions and he has helped me improve this project in all manners and achieve the level that it has acquired. I am extremely thankful to them for providing such nice support and guidance, although they had a busy schedule managing the corporate affairs.

# ABSTRACT


## Health Assistant: A Multi-Disease Prediction Web Application

This project called "Health Assistant" is a web application that empowers individuals to take a more proactive role in their health by leveraging machine learning for disease risk assessment. Developed using Streamlit, this user-friendly platform caters to individuals with no prior technical experience. Users simply provide basic data points, and "Health Assistant" generates predictions on the likelihood of developing three common diseases: Diabetes, Heart Disease, and Parkinson's Disease. These insights can serve as a springboard for further investigation and potentially lead to earlier diagnoses. Early detection is paramount for effective disease management, and "Health Assistant" aspires to contribute to improved health outcomes by promoting preventative measures and timely intervention.

"Health Assistant" addresses a growing need for accessible and informative healthcare tools. By democratizing health knowledge through user-friendly interfaces and clear visualizations, the application empowers individuals to make informed decisions about their well-being. It is important to emphasize that "Health Assistant" does not replace professional medical advice. Instead, it serves as a complementary resource, prompting users to seek confirmation and guidance from qualified healthcare providers. Furthermore, "Health Assistant" can be a valuable tool for medical professionals, providing preliminary insights that can inform further testing and diagnosis. Ultimately, "Health Assistant" aspires to bridge the gap between individuals and vital health information, fostering a more proactive approach to healthcare and empowering users to collaborate with medical professionals in managing their health.

# TABLE OF CONTENTS


| Sr.No | Index | Page No |
|---|---|---|
| **1** | **CHAPTER 1: INTRODUCTION** | |
| **1.1** | Background of the Project | **8** |
| **1.2** | Objectives of the Project | **9** |
| **1.3** | Scope of the Project | **10** |
| **2** | **CHAPTER 2: LITERATURE REVIEW** | |
| **2.1** | List of Technologies | **11** |
| **2.2** | Comparative Study | **12** |
| **2.3** | Literature Review | **13** |
| **3** | **CHAPTER 3: REQUIREMENT & ANALYSIS** | |
| **3.1** | Problem Definition | **14** |
| **3.2** | Dataset Description | **16** |
| **3.3** | Feasibility Study | **20** |
| **3.4** | Planning and scheduling | **22** |
| **4** | **CHAPTER 4: SYSTEM DESIGN** | |
| **4.1** | Model Workflow | **23** |
| **4.2** | User Interface Design | **30** |
| **5** | **CHAPTER 5: SYSTEM DEPLOYMENT** | |
| **5.1** | App Deployment | **36** |
| **5.2** | Test Approach and Test Cases | **38** |
| **6** | **CHAPTER 6: CONCLUSION** | |
| **6.1** | Conclusion | **41** |
| **6.2** | Limitations | **42** |
| **6.3** | Future Scope | **42** |
| | **REFERENCES** | **43** |

# List of Figures


| Sr. No | Figure Name | Page No |
|---|---|---|
| 1 | Gantt Chart | 22 |
| 2 | Model Workflow Diagram [1] | 23 |
| 3 | SVM Model [1] | 23 |
| 4 | Model Training [1] | 24 |
| 5 | Model Evaluation [1] | 24 |
| 6 | Model Workflow Diagram [2] | 25 |
| 7 | Logistic Regression Model | 25 |
| 8 | Model Training [2] | 26 |
| 9 | Model Evaluation [2] | 26 |
| 10 | Model Workflow Diagram [3] | 27 |
| 11 | SVM Model [2] | 27 |
| 12 | Model Training [3] | 28 |
| 13 | Model Evaluation [3] | 28 |
| 14 | Diabetes Prediction Page (Deployment) | 35 |
| 15 | Heart Disease Prediction Page (Deployment) | 35 |
| 16 | Parkinson's Disease Prediction Page (Deployment) | 36 |
| 17 | Extra Options Page (Deployment) | 36 |
| 18 | Diabetes Prediction Page (Test) | 37 |
| 19 | Heart Disease Prediction Page (Test) | 37 |
| 20 | Parkinson's Disease Prediction Page (Test) | 38 |
| 21 | Print Result Page (Test) | 38 |
| 22 | Record Page (Test) | 39 |

# CHAPTER 1: INTRODUCTION

## 1.1 BACKGROUND OF THE PROJECT:

The rise of chronic diseases poses a significant global health challenge. Early detection and preventive measures are crucial for effective disease management. Here, we explore the specific background and growing concern for each disease targeted by the application:

**1. Diabetes:**

Diabetes is a chronic condition characterized by persistently high blood sugar levels. It occurs when the body either doesn't produce enough insulin, or cells become resistant to its effects, leading to an impaired ability to utilize glucose for energy. The International Diabetes Federation estimates that 463 million people globally have diabetes in 2019, with a projected rise to 700 million by 2045.Diabetes can lead to severe complications, including heart disease, stroke, kidney failure, blindness, and limb amputation.Early detection allows for improved blood sugar management and can significantly reduce the risk of complications.

**2. Heart Disease:**

Heart disease, encompassing various conditions affecting the heart and blood vessels, is the leading cause of death globally. Risk factors for heart disease include high blood pressure, high cholesterol, diabetes, smoking, obesity, and physical inactivity. Early detection and risk assessment are crucial for preventing heart disease or mitigating its severity. Early intervention can involve lifestyle modifications, medication, and potentially procedures like angioplasty or bypass surgery.

**3. Parkinson's Disease:**

Parkinson's disease is a neurodegenerative disorder that affects movement. Symptoms typically develop gradually and include tremors, stiffness, slowness of movement, and balance problems. Although the exact cause remains unknown, age is the biggest risk factor. There is currently no cure for Parkinson's disease, but medications and therapies can manage symptoms and improve quality of life.Early diagnosis allows for timely intervention and improved disease management strategies.

# 1.2 OBJECTIVE OF THE PROJECT:

The primary objective of the "Health Assistant" web application is to develop a user-friendly platform for multi-disease prediction, specifically focusing on three prevalent and concerning conditions: diabetes, heart disease, and Parkinson's disease.

**1. Diabetes Prediction:**

- Develop a machine learning model that utilizes user-provided data to predict the likelihood of developing type 2 diabetes.
- Identify key features and risk factors associated with diabetes, such as age, weight, family history, and lifestyle habits.
- Train the model on a robust dataset of labeled diabetes cases to achieve a high degree of accuracy in predicting the risk of the disease.

**2. Heart Disease Prediction:**

- Develop a machine learning model that predicts the user's susceptibility to heart disease based on their input data.
- Integrate relevant risk factors including blood pressure, cholesterol levels, smoking history, family history, and physical activity levels.
- Train the model on a comprehensive dataset of heart disease cases to generate reliable predictions of potential heart issues.

**3. Parkinson's Disease Prediction:**

- Design a machine learning model that analyses user-provided data to assess the risk of developing Parkinson's disease.
- Explore the potential of incorporating voice recordings, handwriting analysis, or user-reported symptoms as input features.
- Train the model on a dataset of Parkinson's disease diagnoses to identify patterns that might indicate an increased risk for the disease.

# 1.3 SCOPE OF THE PROJECT:

"Health Assistant" focuses on developing a web application for multi-disease prediction, specifically targeting diabetes, heart disease, and Parkinson's disease. The project scope encompasses the following key elements:

- **Machine Learning Model Development:**
    - Constructing separate machine learning models for each targeted disease, leveraging user-provided data to predict the likelihood of disease development.
    - Training the models on comprehensive datasets of labelled disease cases to ensure accurate predictions.
- **User Interface Design:**
    - Developing a user-friendly and intuitive web interface using Streamlit for ease of use by individuals with no prior technical experience.
    - Ensuring clear data input fields, informative visualizations of results, and accessible explanations of predicted risks.

**Existing System vs. Proposed System:**

- **Existing Systems:**
    - Traditional methods for disease risk assessment often involve consultations with healthcare professionals and potentially invasive diagnostic tests.
    - Limited availability of user-friendly tools for self-assessment of disease risks can hinder early detection efforts.
- **Proposed System:**
    - "Health Assistant" offers a readily accessible and non-invasive approach for individuals to assess their potential risk for developing specific diseases.
    - The user-friendly interface empowers individuals to take a more proactive role in managing their health.

# CHAPTER 2: SURVEY OF TECHNOLOGIES

## 2.1 LIST OF TECHNOLOGIES:

**1. Python:**

Python is a general-purpose, high-level programming language known for its readability and ease of use. It's widely used in various fields, including data science, web development, machine learning, and scientific computing. Python's popularity stems from its clear syntax, extensive libraries, and large supporting community.

**2. Google Colab:**

Google Colab is a free cloud-based Jupyter notebook environment offered by Google Research. It allows you to write and execute Python code directly in your web browser, eliminating the need for local software installation. Colab provides free access to powerful hardware resources like GPUs, making it ideal for computationally intensive tasks like machine learning model training.

**3. Spyder on Anaconda:**

Anaconda is a free and open-source distribution of Python that includes essential scientific computing libraries pre-installed. Spyder is a scientific IDE (Integrated Development Environment) included in the Anaconda distribution. IDEs provide a user-friendly interface for writing, editing, and running Python code, along with features like code completion, debugging tools, and variable inspection. Spyder offers a more traditional desktop application experience for working with Python compared to the web-based Google Colab.

**4. Streamlit:**

Streamlit is a Python framework specifically designed for creating web applications. It allows you to quickly build data apps with minimal coding compared to traditional web development frameworks. Streamlit integrates seamlessly with Python libraries, enabling you to leverage existing code and data analysis results to create interactive web interfaces.

# 2.2 COMPARATIVE STUDY:

This analysis compares Python, Google Colab, Spyder on Anaconda, and Streamlit for their suitability in developing the "Health Assistant" web application. We'll focus on three key areas: Development and Model Training, Web Application Development, and IDE.

## 1. Development and Model Training:

- **Python:** The core programming language. Python excels in data science and machine learning tasks. Libraries like NumPy, Pandas, and scikit-learn provide tools for data manipulation, analysis, and model development.
- **Google Colab:** A cloud-based Python environment. It eliminates the need for local software installation and offers free access to powerful hardware (GPUs) ideal for training computationally intensive machine learning models. However, it requires internet connectivity and has limited storage for long-term projects.

## 2. Web Application Development:

- **Streamlit:** A Python framework for creating web applications. Streamlit simplifies development compared to traditional frameworks. Leverage existing Python code and data analysis results to build interactive data apps with minimal coding. However, customization options are limited compared to full-fledged frameworks, and the community is smaller.

## 3. IDE (Integrated Development Environment):

- **Spyder on Anaconda:** A user-friendly IDE specifically designed for scientific computing with Python. Anaconda pre-installs essential scientific computing libraries. Spyder offers features like code completion, debugging tools, and variable inspection, streamlining your development process. However, it requires local software installation.

# 2.3 LITERATURE REVIEW:

1. Aishwarya Mujumdar and Dr. V Vaidehi (2019) this research focuses on improving diabetes prediction using big data analysis. Traditional methods rely on various tests and may not be very accurate. This study proposes a new model that considers additional factors beyond the usual ones (glucose, BMI, age, etc.) for better classification of diabetes. By using a bigger dataset and a new model, the researchers were able to improve the accuracy of predicting diabetes.

2. Mitushi Soni and Dr. Sunita Varma (2020) this research investigates using machine learning to improve early prediction of diabetes. They highlight the dangers of untreated diabetes and the importance of early detection. The project explores various machine learning techniques (KNN, Logistic Regression, Decision Tree, etc.) to build models for predicting diabetes from patient data. Their findings suggest that Random Forest outperforms other techniques in achieving the most accurate predictions.

3. Mohammed Khalid Hossen (2022) this paper explores using machine learning to detect heart disease. Researchers compared different machine learning algorithms on a dataset of patient information. Logistic regression achieved the highest accuracy (95%) in detecting heart disease compared to other algorithms tested (Support Vector Machine, KNN, Random Forest, Gradient Boosting Classifier). The study acknowledges the challenge of improving accuracy even further to near-perfect levels (97-100%).

4. Aditi Govindua and Sushila Palweb (2023) this research explores using machine learning in telemedicine to remotely detect PD early on. By analyzing voice data from patients, they found a Random Forest machine learning model achieved the highest accuracy (91.83%) in detecting PD. This approach has the potential to improve early detection and treatment for Parkinson's patients.

# CHAPTER 3: REQUIREMENT & ANALYSIS

## 3.1 PROBLEM DEFINITION:

Chronic diseases like diabetes, heart disease, and Parkinson's disease pose a significant global health burden. Early detection is crucial for effective treatment and improved patient outcomes. However, traditional methods of disease diagnosis often involve physical examinations and various tests, leading to inconvenience and potential delays. Additionally, limited access to healthcare facilities and specialists can further hinder early detection, particularly in remote areas or for patients with mobility challenges.

This project aims to address these challenges by developing a web application called "Health Assistant." Health Assistant will be a user-friendly, web-based platform utilizing machine learning algorithms for the early prediction of these prevalent diseases.

Here's a detailed breakdown of the **problems addressed by Health Assistant**:

- Limited Accessibility to Early Disease Detection: Traditional methods for disease diagnosis often rely on physical visits and specialized tests. This can be inconvenient and impractical for some patients, especially those in remote locations or with mobility issues.
- Delays in Diagnosis: Traditional methods may involve waiting times for appointments and test results, leading to delays in diagnosis and treatment initiation.
- Lack of Comprehensive Screening Tools: Current screening tools may not comprehensively assess risk factors for various diseases.

### How Health Assistant Addresses these Problems:

- Convenience and Accessibility: Health Assistant will be a web-based application accessible from any internet-connected device, eliminating geographical and mobility barriers.
- Early Disease Prediction: By leveraging machine learning algorithms, Health Assistant will analyze user-provided data to predict the risk of developing diabetes, heart disease,

and Parkinson's disease. This early prediction can prompt users to seek professional medical evaluation.

- Comprehensive Risk Assessment: Health Assistant will incorporate a wider range of data points beyond traditional risk factors, potentially improving the accuracy of early prediction.

## Overall Objective:

The primary objective of Health Assistant is to empower individuals to take a proactive approach to their health through early disease prediction. By providing a convenient, accessible, and informative tool, Health Assistant aims to:

- Increase awareness of potential health risks.
- Encourage early detection and intervention.
- Bridge the gap between individuals and healthcare professionals.

This project contributes to the advancement of telemedicine by leveraging machine learning for preventative healthcare and promoting early diagnosis of chronic diseases.

# 3.1 DATASET DESCRIPTION:

## 1. Diabetes Prediction Dataset:

This dataset focuses on predicting diabetes in Pima Indian women using diagnostic measurements.

**Key Points:**

- **Origin:** National Institute of Diabetes and Digestive and Kidney Diseases
- **Objective:** Predict diabetes based on diagnostic tests
- **Target Population:** Pima Indian females at least 21 years old
- **Number of Instances:** 768
- **Number of Attributes:** 8 (features) + 1 (class label) - all numeric
- **Features:**
    - Number of times pregnant
    - Plasma glucose concentration (2-hour oral glucose tolerance test)
    - Diastolic blood pressure (mm Hg)
    - Triceps skin fold thickness (mm)
    - 2-hour serum insulin level (mu U/ml)
    - Body mass index (BMI)
    - Diabetes pedigree function
    - Age (years)
- **Class Label:** Outcome (0 = negative test, 1 = positive test for diabetes)

# 2. Heart Disease Prediction Dataset:

This dataset, compiled in 1988, combines information from four medical centers (Cleveland, Hungary, Switzerland, and Long Beach V) to study heart disease. While it contains 76 attributes, most research focuses on a specific subset of 14. The key characteristic of this dataset is the "target" field, indicating the presence or absence of heart disease (0 = no disease, 1 = disease).

## Keypoints of the relevant attributes:

- **age:** Patient's age in years
- **sex:** Patient's sex (likely encoded as 1 = male, 0 = female)
- **chest pain type (4 values):** Categorical variable describing the type of chest pain experienced by the patient (specific values not provided)
- **resting blood pressure:** Patient's blood pressure reading at rest
- **serum cholesterol in mg/dl:** Level of cholesterol in the patient's blood
- **fasting blood sugar > 120 mg/dl:** Binary indicator (1 = yes, 0 = no) of whether the patient's fasting blood sugar level exceeded 120 mg/dl
- **resting electrocardiographic results (values 0,1,2):** Categorical variable representing the results of the patient's resting electrocardiogram (ECG) test (specific value meanings not provided)
- **maximum heart rate achieved:** Highest heart rate reached by the patient during exercise testing
- **exercise induced angina:** Binary indicator (1 = yes, 0 = no) of whether the patient experienced chest pain during exercise testing
- **oldpeak = ST depression induced by exercise relative to rest:** Measurement of the change in ST segment on the ECG during exercise compared to rest
- **the slope of the peak exercise ST segment:** Categorical variable describing the slope of the ST segment on the ECG during peak exercise (specific value meanings not provided)

- **number of major vessels (0-3) colored by flourosopy:** Number of major coronary arteries with significant blockage as identified by a fluoroscopy procedure
- **thal: 0 = normal; 1 = fixed defect; 2 = reversable defect:** Categorical variable indicating the results of a thallium stress test, which can reveal abnormalities in blood flow to the heart muscle (0 = normal, 1 = fixed blockage, 2 = potentially reversible blockage)

## 3. Parkinson's Disease Prediction Dataset:

This dataset focuses on voice characteristics of individuals with and without Parkinson's disease (PD). It contains recordings from 31 people, 23 with PD and 8 healthy.

**Key Points:**

- **Source:** Voice recordings
- **Number of Individuals:** 31 (23 with PD, 8 healthy)
- **Number of Recordings:** 195 (around 6 recordings per person)
- **Format:** ASCII CSV
- **Objective:** Differentiate between healthy and Parkinson's disease based on voice features

**Data Attributes:**

- **name:** Subject name and recording number (might be helpful for linking recordings to a specific person)
- **MDVP Features (14):** These features capture various aspects of vocal fundamental frequency (pitch):
    - **Average, Maximum, Minimum Frequency (Hz):** Overall pitch characteristics
    - **Jitter (% & Abs):** Quantify cycle-to-cycle variations in pitch period
    - **RAP, PPQ, DDP:** Additional measures related to pitch variation/disturbance

- **Shimmer Features (6):** These features analyze variations in voice amplitude:
    - **Shimmer, Shimmer(dB):** Measure the degree of amplitude variations
    - **Shimmer APQ3, APQ5, MDVP:APQ:** Different calculations related to amplitude perturbation
    - **Shimmer:DDA:** Captures the average absolute difference between consecutive amplitude values
- **NHR & HNR:** Represent the ratio of noise to tonal components in the voice, potentially indicating breathiness or voice quality changes.
- **Status (0 or 1):** Indicates the health status of the subject (0 = healthy, 1 = Parkinson's disease)
- **RPDE & D2:** Nonlinear complexity measures, potentially reflecting changes in voice dynamics in PD.
- **DFA:** Signal fractal scaling exponent, another measure of vocal complexity.
- **spread1, spread2, PPE:** Capture non-linear variations in fundamental frequency.

# 3.2 Feasibility Study:

Feasibility study is checking if a project or idea is doable and makes sense. It involves looking at the resources, costs, and potential risks to see if the project is possible and if it's worth doing. Doing a feasibility study helps us decide if the project is a good idea and if we can make it work before we start.

The Feasibility study can be further divided into following:

## 1. Technical Feasibility:

- Streamlit simplifies web app development.
- Existing machine learning libraries (e.g., scikit-learn) offer pre-built algorithms

- Technical feasibility is high. Existing tools and libraries can be leveraged to develop and deploy the application. However, careful data analysis and model optimization are crucial for achieving reliable predictions.

## 2. Behavioural Feasibility:

- The application addresses a growing concern - early disease detection.
- User-friendly interface can encourage preventative healthcare practices.
- Accessibility through web browsers eliminates geographical and mobility barriers.

- Behavioural feasibility is moderate. User adoption depends on effective communication about the app's limitations and its role as a potential early warning system.

## 3. Economic Feasibility:

- Open-source libraries and cloud platforms offer cost-effective development and deployment options.

- Economic feasibility is promising. Utilizing cost-effective tools and a free model can ensure financial sustainability.

# Software and Hardware Requirements:

| Process | Software Requirements |
|---|---|
| Programming Language | Python |
| Model Development | Google Colab |
| IDE | Spyder on Anaconda environment |
| User Interface | Streamlit |

| Hardware Requirements | Description |
|---|---|
| Operating System | Windows 10 or higher |
| RAM | 4GB or Above |
| System Type | 64-bit operating system, x64-based processor |
| Monitor | Color screen |

# Specific Library Requirements:

- numpy==1.26.3
- scikit-learn==1.3.2
- streamlit==1.29.0
- streamlit-option-menu==0.3.6

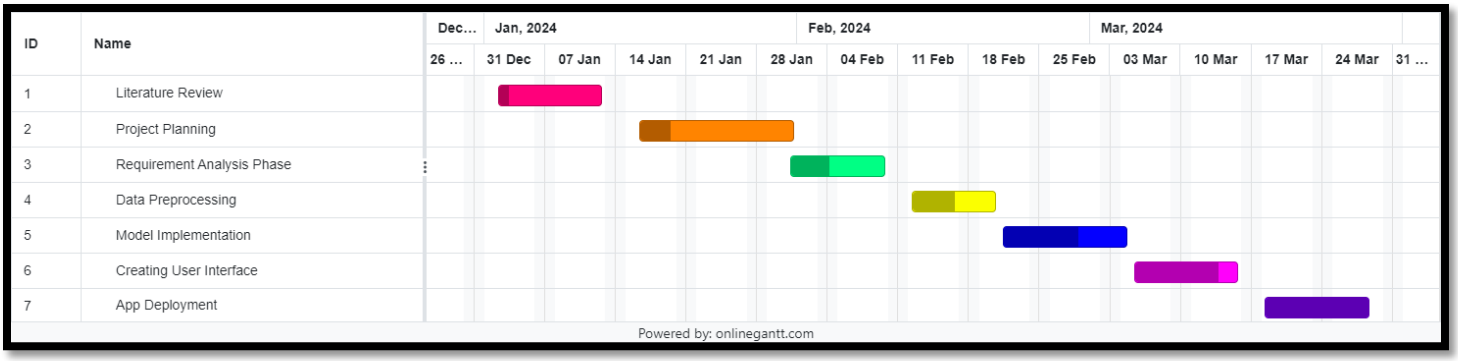
# 3.3 Planning and Scheduling:

## Gantt Chart:

Gantt chart is a bar chart used to illustrate project schedules, showing effort, resources, milestones, and deliveries. It allows project managers to track overall project progress and individual tasks. Invented by Henry Gantt in 1910, Gantt charts have evolved into sophisticated software tools, enabling easy project management.

**Advantages:**

- Provides an overview of project status and tasks, facilitating efficient project tracking.
- Software-based Gantt charts show task dependencies, helping identify and maintain critical project paths.
- Suitable for managing small projects as a single entity.

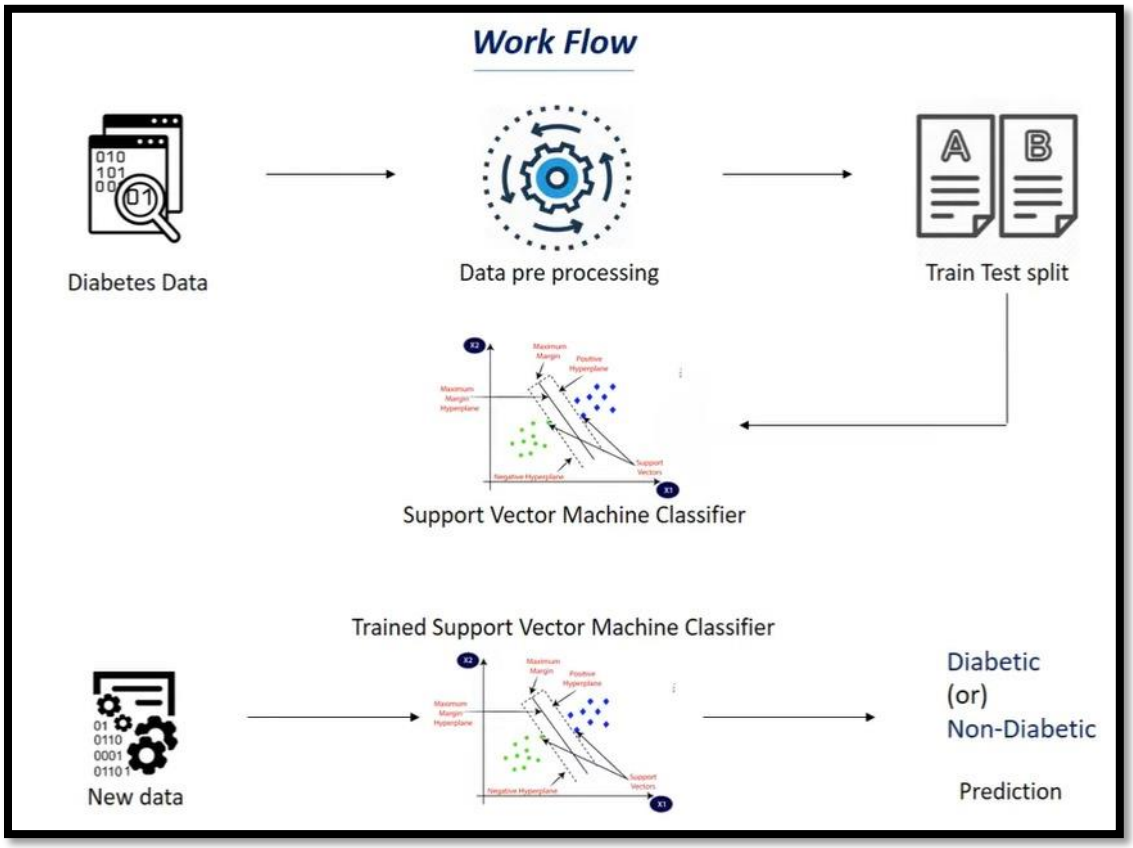


# CHAPTER 4: SYSTEM DESIGN

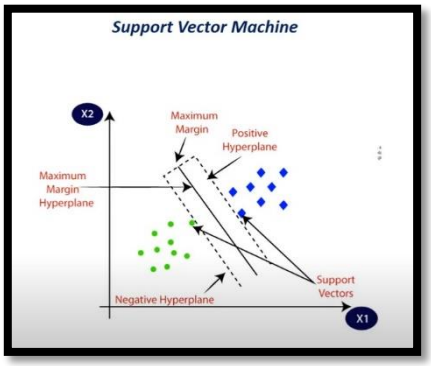
## 4.1 Model Workflow:

### 1. DIABETES PREDICTION MODEL:

**Model Workflow Diagram:**




- **Support Vector Machine:**

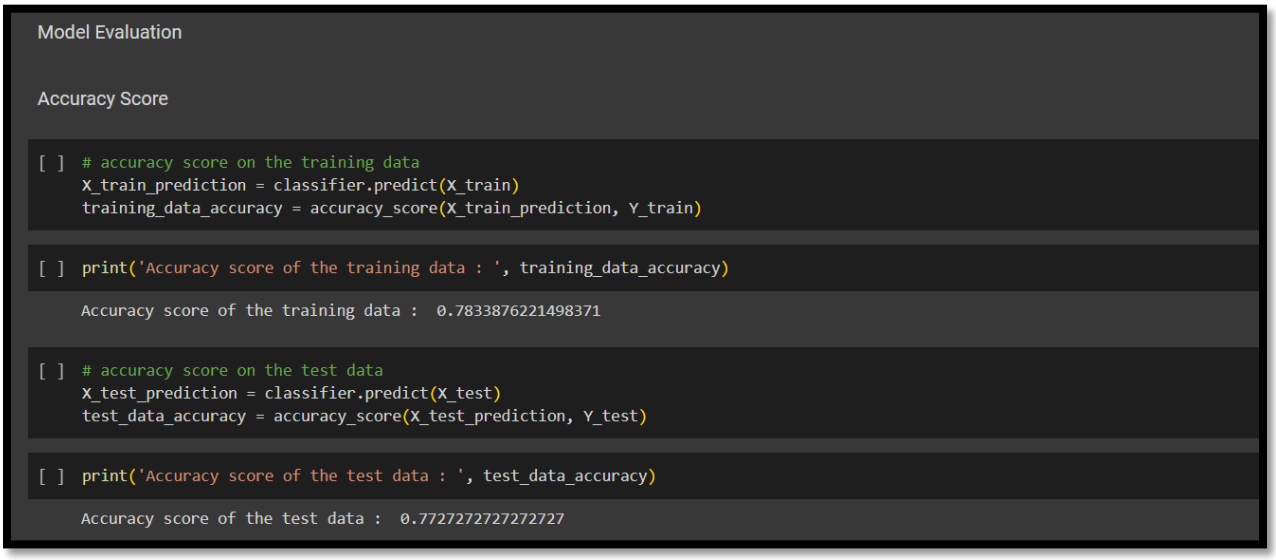



Support Vector Machines (SVMs) are a powerful tool used for classification tasks in machine learning. They work by finding a dividing line (or hyperplane in higher dimensions) that best separates data points belonging to different categories. The algorithm maximizes the margin between this line and the closest data points from each class, ensuring a clear separation. SVMs are particularly useful for high-dimensional data and can work well even with smaller datasets. However, training SVMs can be computationally

demanding, and choosing the right settings for the algorithm is crucial for optimal performance.

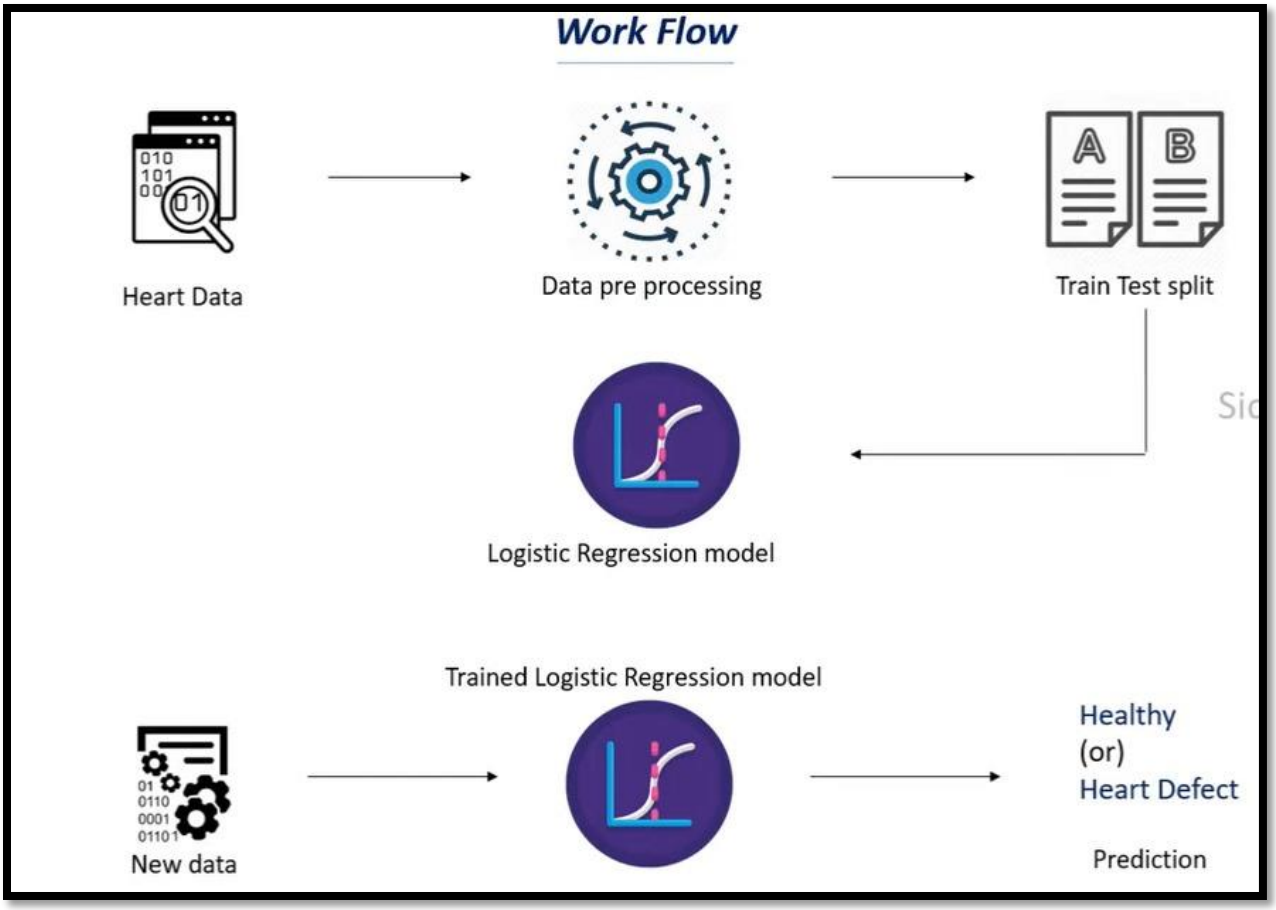
## Model Training:

## Model Evaluation:



## 2. HEART DISEASE PREDICTION MODEL:

### Model Workflow Diagram:



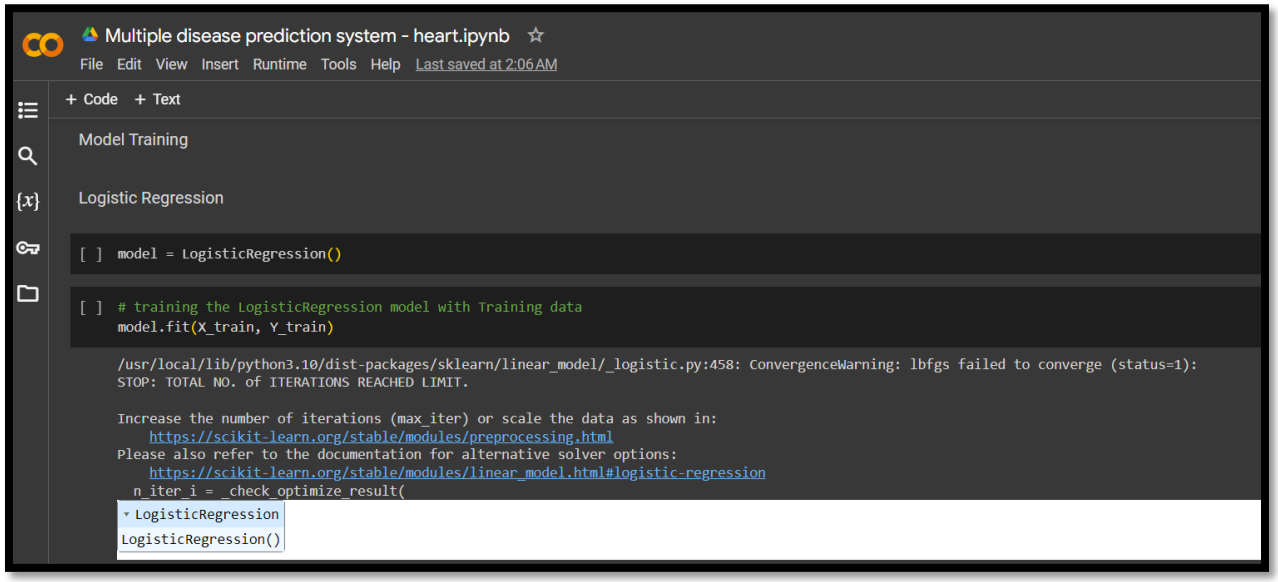

- **Logistic Regression Model:**

Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. It measures the relationship between the dependent variable and the one or more independent variables (risk factors) by estimating probabilities using underlying logistic function (sigmoid function). Sigmoid function is used as a cost functionto limit the hypothesis of logistic regression between 0 and 1 (squashing) i.e. $0 \leq h\theta(x) \leq 1$.

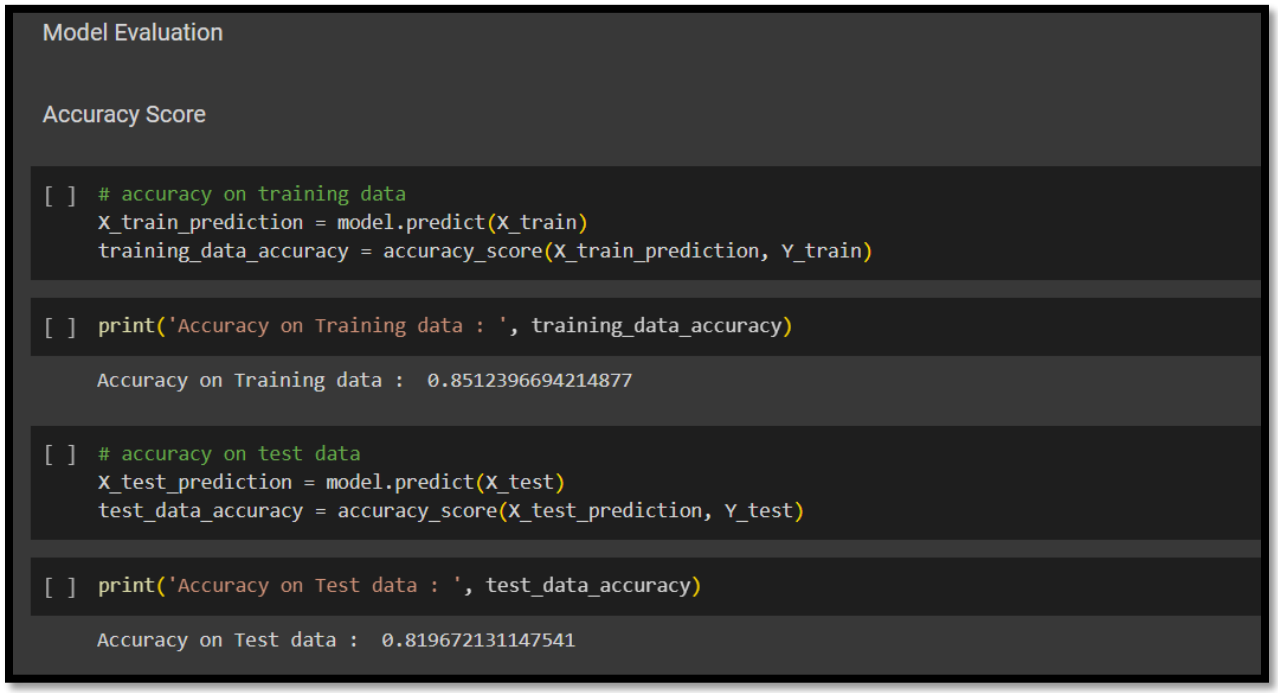
In logistic regression cost function is defined as:

$$Cost(h\theta(x), y) = \begin{cases} -\log(h\theta(x)) & if\ y = 1 \\ -\log(1 - h\theta(x)) & if\ y = 0 \end{cases}$$

# Model Training:

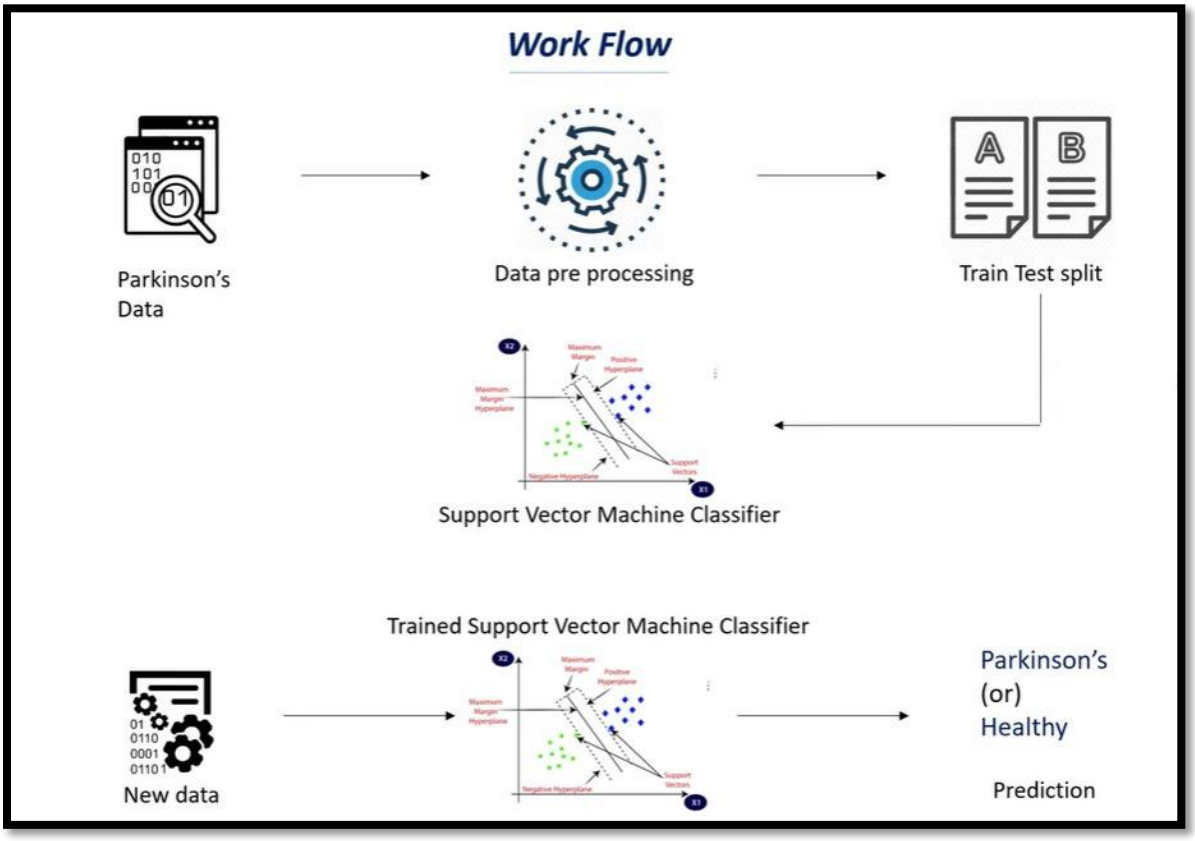



# Model Evaluation:



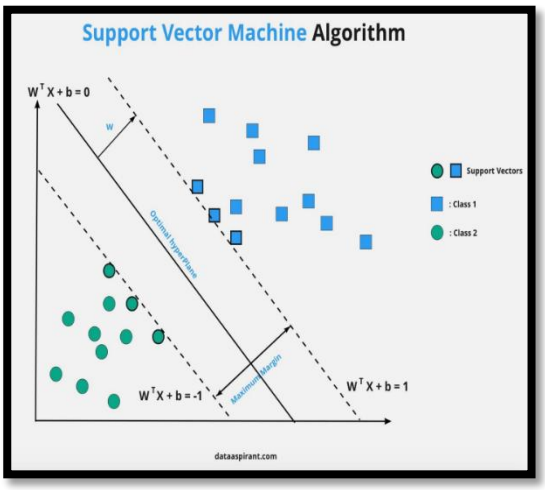
## 3. PARKINSON'S DISEASE PREDICTION MODEL:

### Model Workflow Diagram:

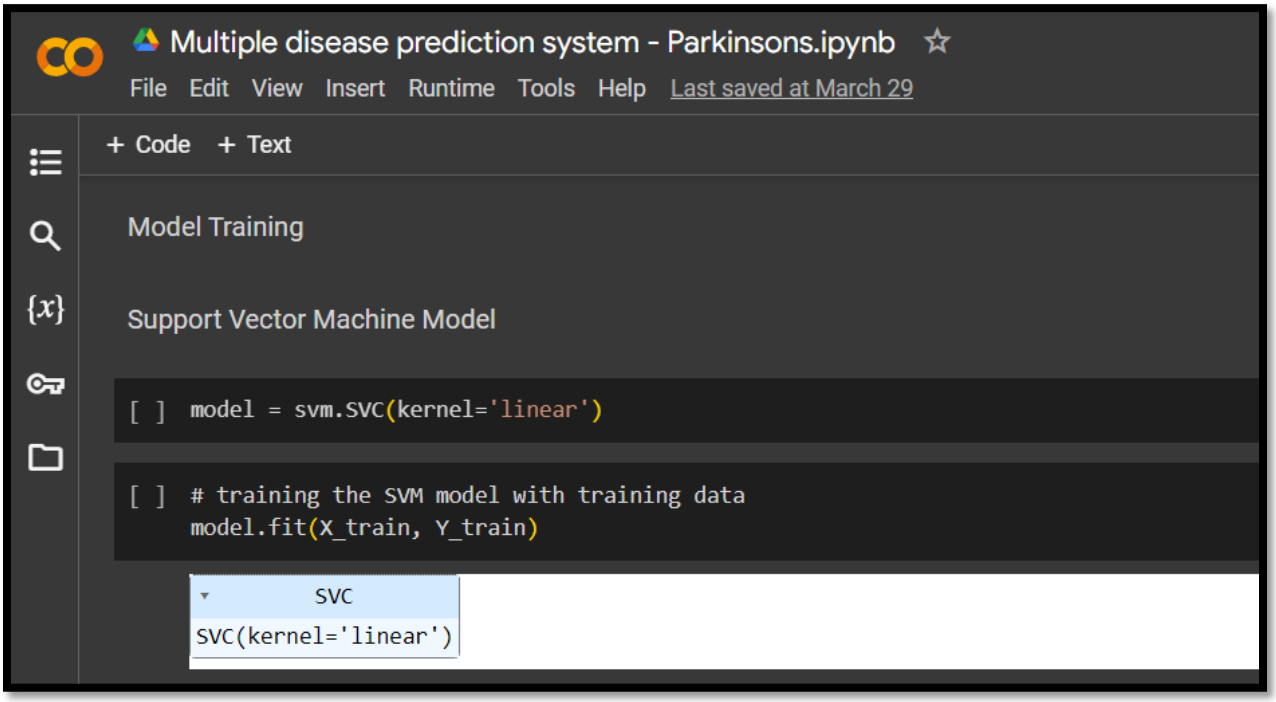



- **Support Vector Machine:**

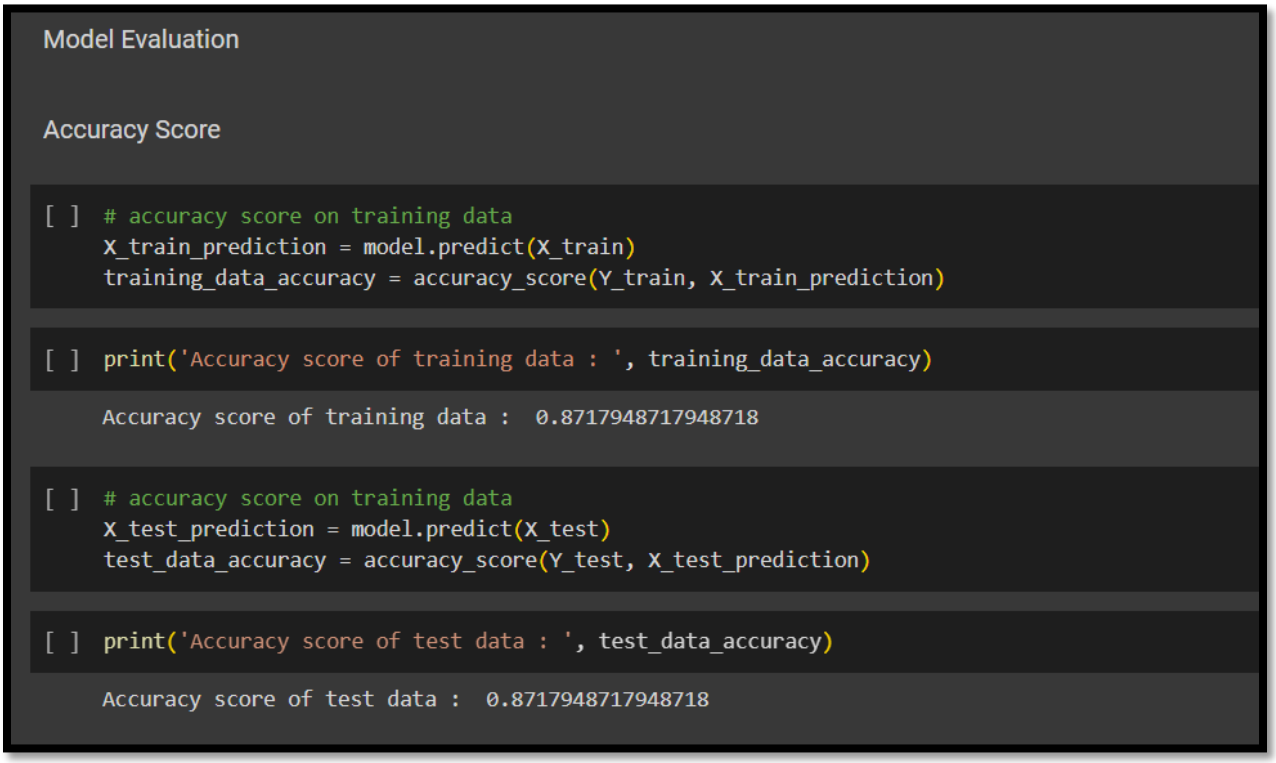
The vector support machine is a supervised learning algorithm for regression and classification questions and scenarios. Creates a decision boundary that can divide N dimensional spaces into classes; this decision boundary is called a hyperplane. It selects excess points to create hyperplanes and is called support vectors. We have two types of vertical and indirect vector learning machines. Is Kim creating multiple decision boundaries to segregate data but we need to find out the best decision boundary to classify our data The best decision boundary is known as the hyperplane supporting vectors are the data points that are in the most proximity to the hyperplane.



## Model Training:




## Model Evaluation:



# 4.2 User Interface Design:

## Code Executed on Anaconda's Sypder IDE:

```
import os

import pickle

import streamlit as st

from streamlit_option_menu import option_menu


# Set page configuration

st.set_page_config(page_title="Health Assistant",

                   layout="wide",

                   page_icon="□⚕□")


# getting the working directory of the main.py

working_dir = os.path.dirname(os.path.abspath(__file__))


# loading the saved models

diabetes_model = pickle.load(open(f'D:/Users/Devika
Jonjale/Desktop/Multiple_Disease_Prediction_System/saved models/diabetes_model.sav', 'rb'))

heart_disease_model = pickle.load(open(f'D:/Users/Devika
Jonjale/Desktop/Multiple_Disease_Prediction_System/saved models/heart_disease_model.sav', 'rb'))

parkinsons_model = pickle.load(open(f'D:/Users/Devika
Jonjale/Desktop/Multiple_Disease_Prediction_System/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'],

                           menu_icon='hospital-fill',

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

```
                    default_index=0)


# Diabetes Prediction Page
if selected == 'Diabetes Prediction':
    # page title
    st.title('Diabetes Prediction using ML')
    # 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'):
        user_input = [Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin,
```

```
                BMI, DiabetesPedigreeFunction, Age]
        user_input = [float(x) for x in user_input]
        diab_prediction = diabetes_model.predict([user_input])
        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.title('Heart Disease Prediction using ML')
    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'):
        user_input = [age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal]
        user_input = [float(x) for x in user_input]
        heart_prediction = heart_disease_model.predict([user_input])
        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.title("Parkinson's Disease Prediction using ML")
    col1, col2, col3, col4, col5 = st.columns(5)
```

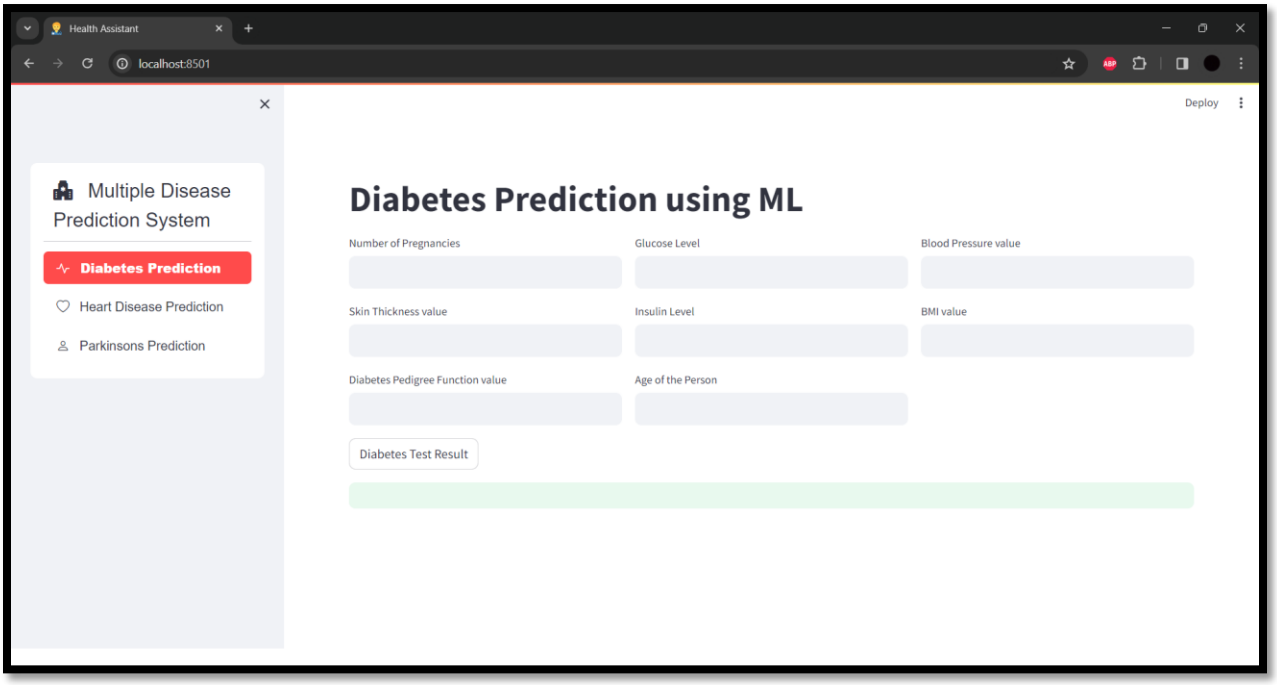
```
with col1:
    fo = st.text_input('MDVP:Fo(Hz)')
with col2:
    fhi = st.text_input('MDVP:Fhi(Hz)')
with col3:
    flo = st.text_input('MDVP:Flo(Hz)')
with col4:
    Jitter_percent = st.text_input('MDVP:Jitter(%)')
with col5:
    Jitter_Abs = st.text_input('MDVP:Jitter(Abs)')
with col1:
    RAP = 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"):
    user_input = [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]
    user_input = [float(x) for x in user_input]
    parkinsons_prediction = parkinsons_model.predict([user_input])
    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)
```

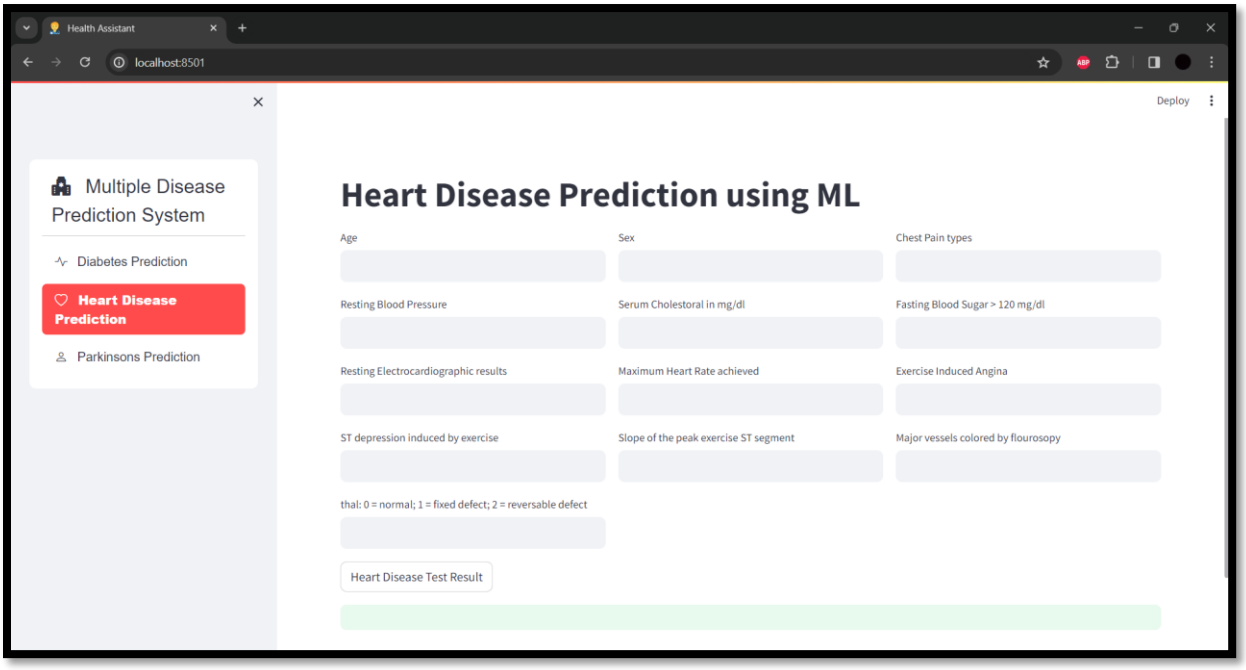
# CHAPTER 5: SYSTEM DEPLOYMENT

## 5.1 App Deployment:

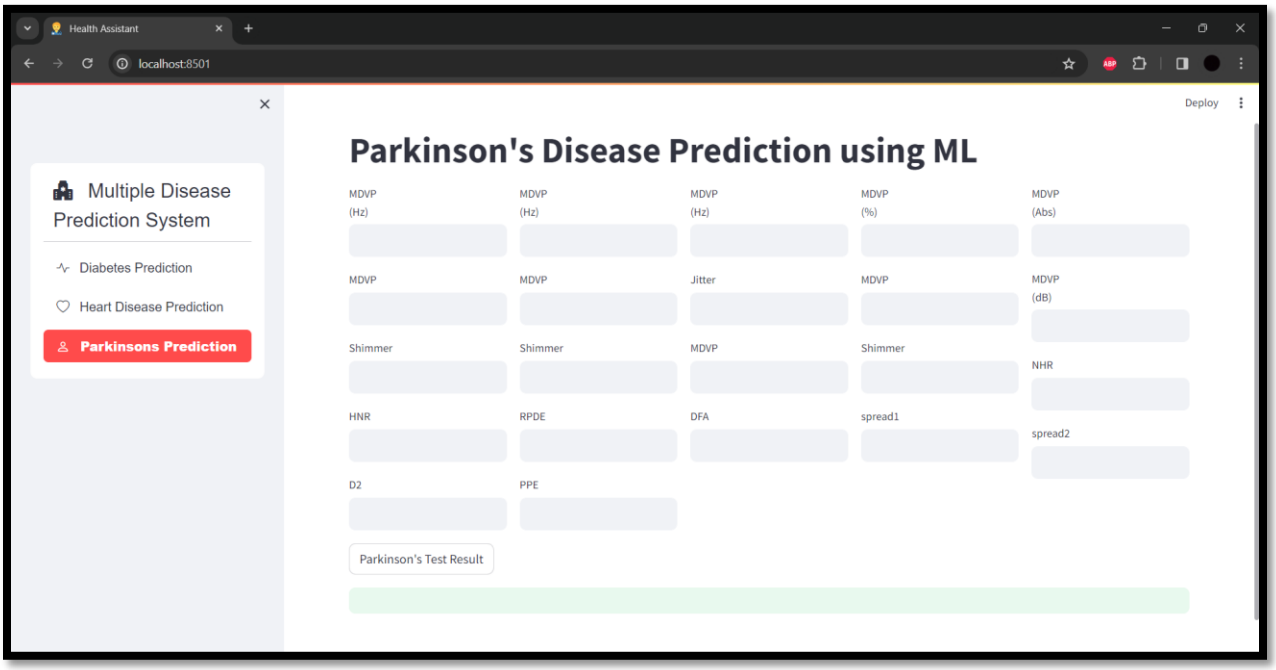
### Diabetes Prediction Page:



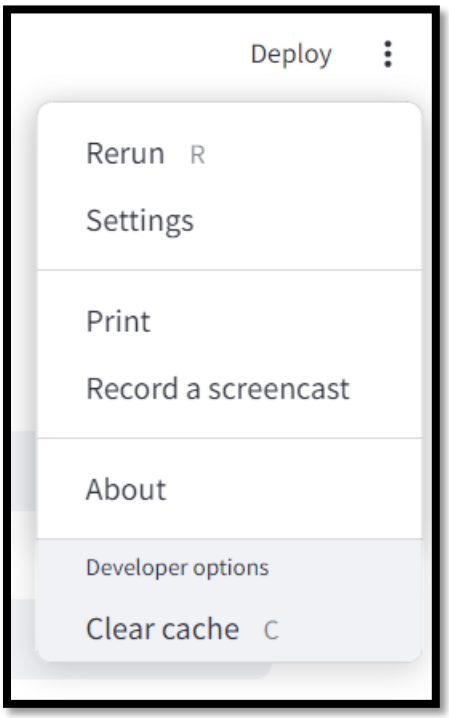

### Heart Disease Prediction Page:



## Parkinson's Disease Prediction Page:

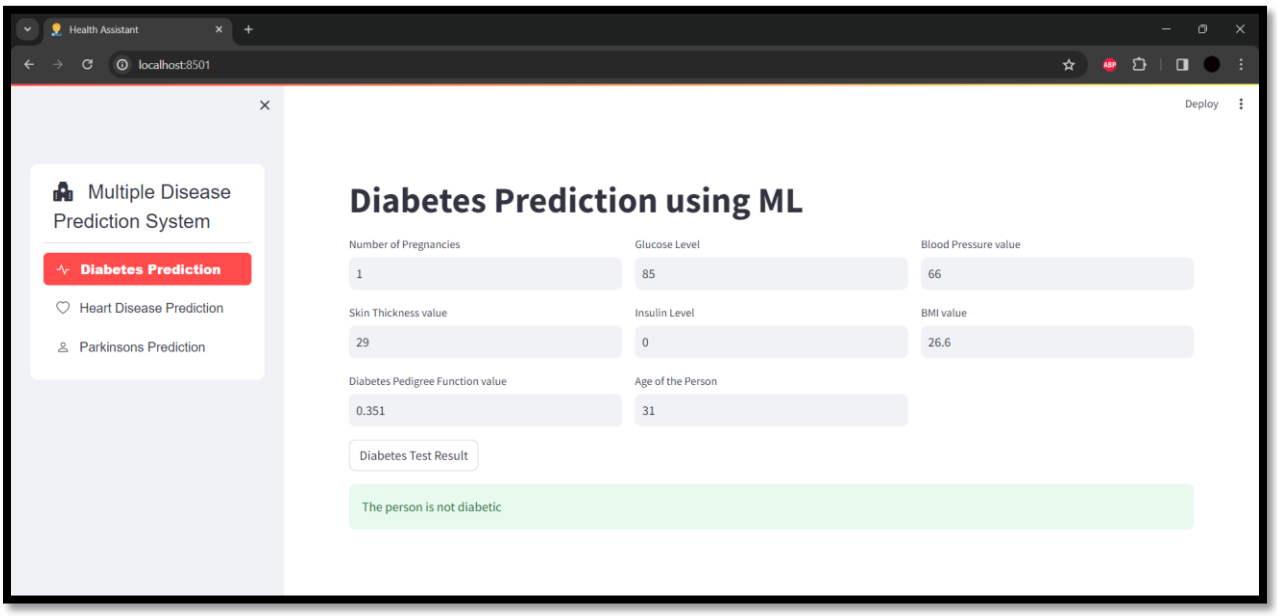



## Extra Options:

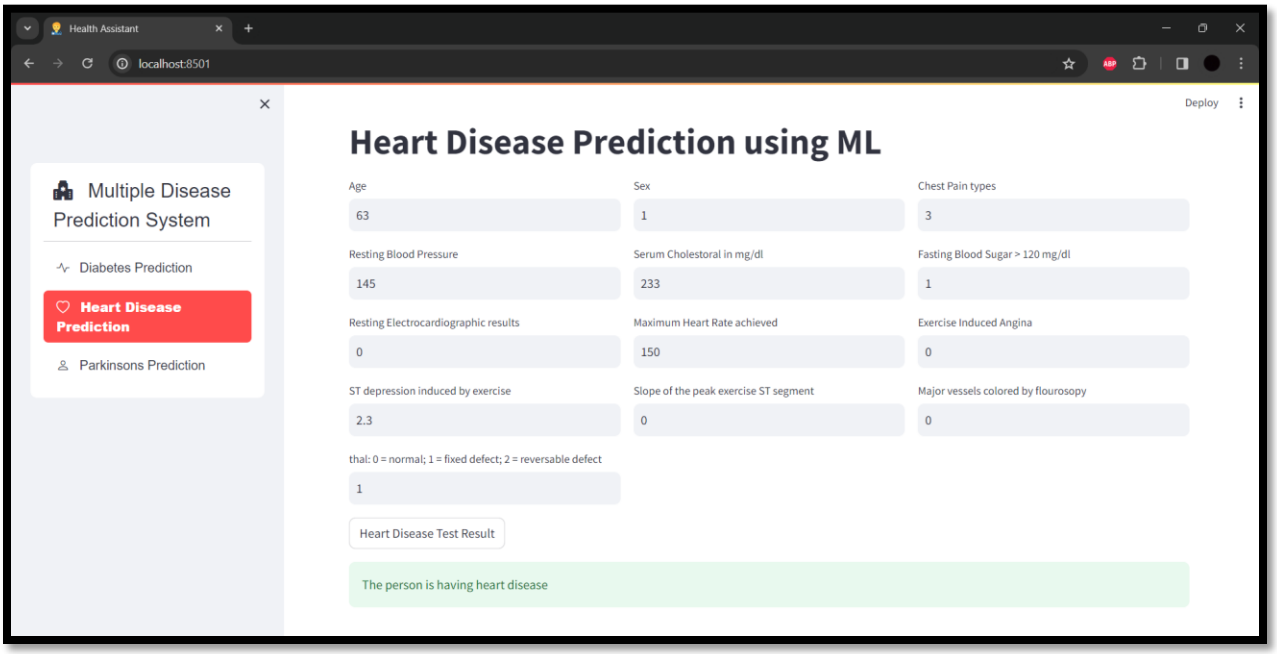


# 5.2 Test Approach and Test Cases:

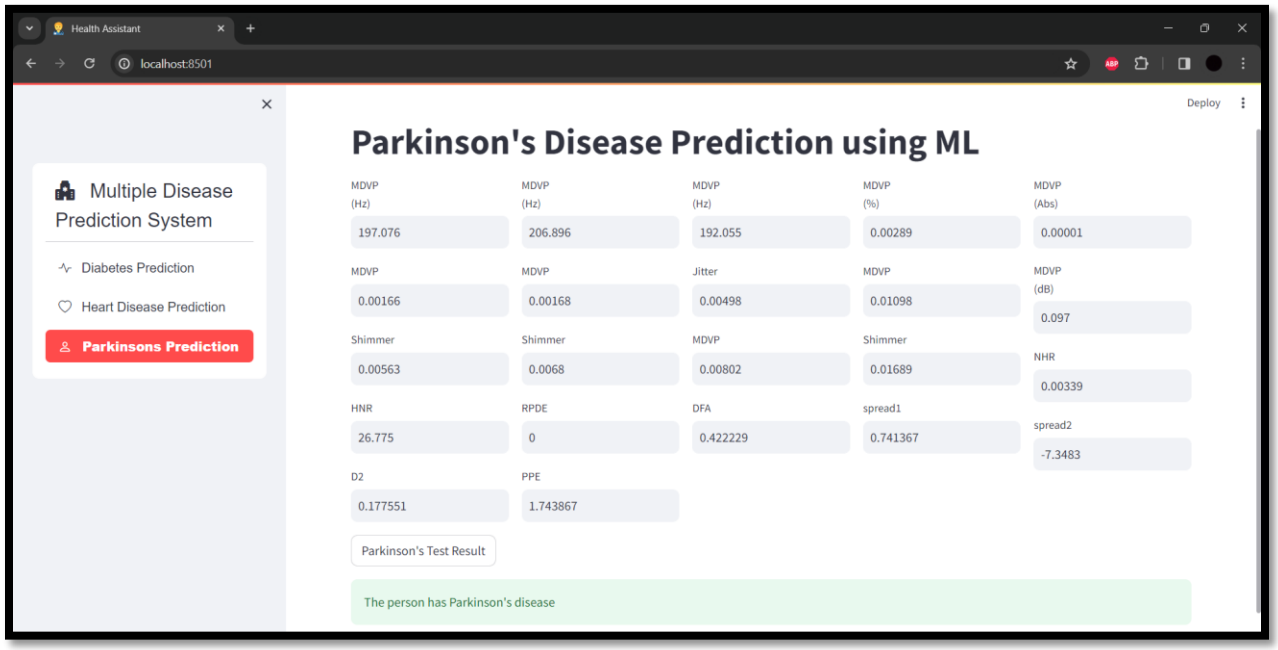
## Diabetes Prediction Page:



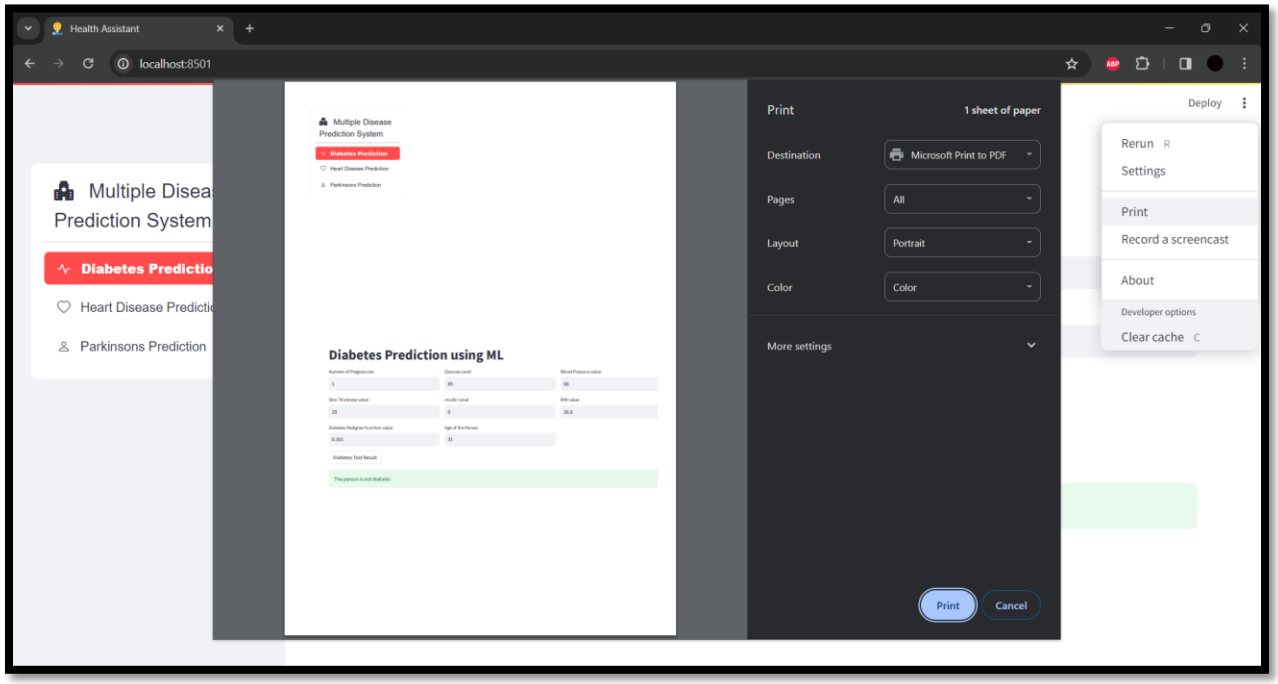

## Heart Disease Prediction Page:



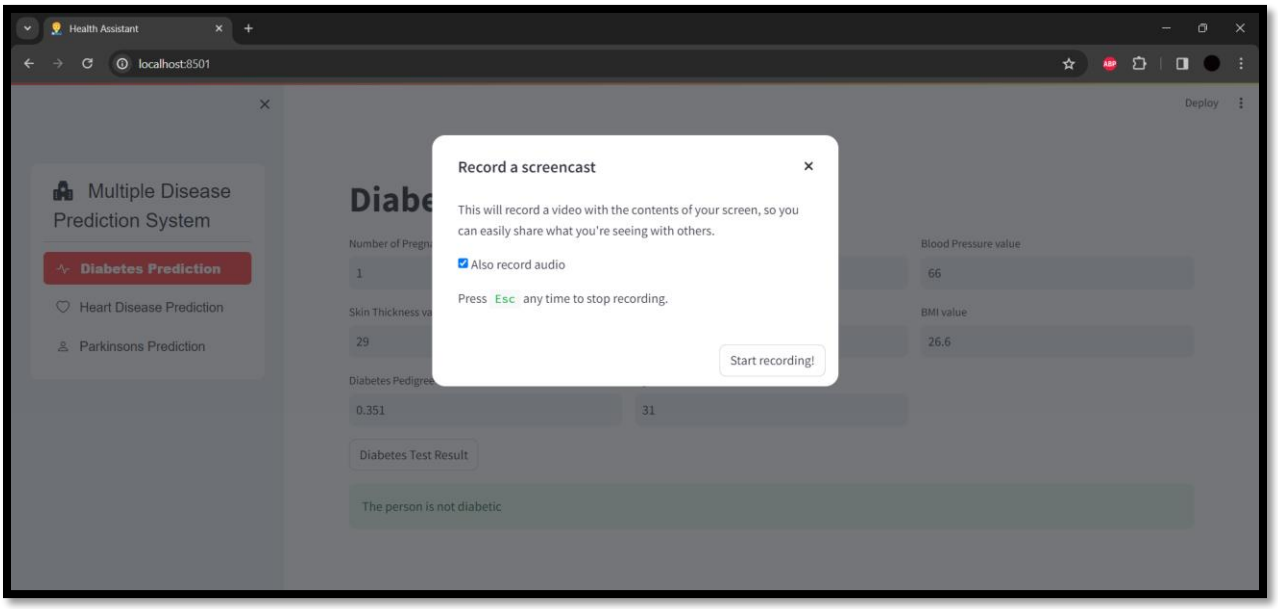
## Parkinson’s Disease Prediction Page:




## You can also print the results:



## You can record a screencast:



# CHAPTER 6: CONCLUSION

## 6.1 Conclusion:

### Health Assistant: A Promising Tool for Early Disease Detection

- The Health Assistant web application has the potential to be a valuable asset in promoting preventative healthcare and early disease detection.

### Key Strengths:

- Early Disease Detection: Machine learning models can analyze user data to predict potential risks for diabetes, heart disease, and Parkinson's disease, prompting users to seek professional medical evaluation.
- Accessibility and Convenience: Web-based accessibility eliminates geographical and mobility barriers, allowing users to conveniently assess their health risks.
- Cost-Effectiveness: Development and deployment can utilize open-source tools and cloud platforms, making it a financially sustainable solution.

### Addressing Challenges:

- Data Quality and Model Accuracy: Careful data analysis, model optimization, and addressing potential biases in existing datasets are crucial for ensuring reliable predictions.
- User Education and Behavior Change: Clear communication is essential to manage user expectations, emphasizing the app's role as a preliminary screening tool, not a definitive diagnosis.
- Data Privacy and Security: Implementing robust security measures and a comprehensive data privacy policy is paramount to gaining user trust.

# 6.2 Limitations:

- **Accuracy and Reliability:**

  While machine learning models offer promising results, the accuracy of disease prediction is not perfect. Dependence on user-reported data and potential biases in existing datasets can impact reliability.

- **Not a Diagnostic Tool:**

  Health Assistant is intended for early detection, not definitive diagnosis. It should always prompt users to seek professional medical evaluation for confirmation and treatment.

- **Data Privacy and Security:**

  User trust hinges on robust data security measures and a transparent data privacy policy. Breaches or misuse of data can have severe consequences.

- **Limited Scope:**

  The current focus is on diabetes, heart disease, and Parkinson's disease. Expanding the application's scope to encompass other diseases requires additional data and model development.

# 6.3 Future Scope:

- **Improved Model Accuracy:**

  Techniques like data augmentation and hyperparameter tuning can be employed to refine models and enhance prediction accuracy.

- **Integration with Wearable Devices:**

  Connecting with wearable devices for real-time data collection (e.g., blood pressure, heart rate) could provide more comprehensive insights.

- **AI-powered Chatbot Integration:**

  A chatbot assistant can guide users through the app, answer questions, and provide educational resources about preventative healthcare.

- **Integration with Electronic Health Records (EHR):**

  Potential future integration with EHR systems could offer a more holistic view of user health data, with user consent of course.

# References:


[1] A. Govindu and S. P. , "Early detection of Parkinson's disease using machine learning," in *Procedia Computer Science*, Pune, 2023.

[2] F. S. M. A.-S. M. A.-M. A. E. W. B. M. A. and F. G. , "Enhancing Parkinson's Disease Prediction Using Machine Learning and," *Tech Science Press,* 2021.

[3] K. K. S. . D. D. D. A. P. G. K. and . D. , "AN EFFECTIVE PARKINSON'S DISEASE PREDICTION USING LOGISTIC DECISION REGRESSION AND MACHINE LEARNING WITH BIG DATA," *Turkish Journal of Physiotherapy and Rehabilitation,* 2019.

[4] A. M. and D. V. V. , "Diabetes Prediction using Machine Learning Algorithms," in *Procedia Computer Science*, 2019.

[5] M. S. and D. S. V. , "Diabetes Prediction using Machine Learning Techniques," *International Journal of Engineering Research & Technology (IJERT),* vol. 9, no. 09, 2020.

[6] K. J. Rani, "Diabetes Prediction Using Machine Learning," *International Journal of Scientific Research in Computer Science Engineering and Information Technology,* vol. 6, no. 4, pp. 294-305, 2020.

[7] M. K. Hossen, "Heart Disease Prediction Using Machine Learning Techniques," *American Journal of Computer Science and Technology,* vol. 5, no. 3, pp. 146-154, 2022.

[8] V. R. . A. D. and M. K. R. , "Heart disease prediction using machine learning techniques: a survey," *International Journal of Engineering & Technology,* vol. 7, no. 2.8, pp. 684-687, 2018.

[9] Python Documentation: https://docs.python.org/3.10/tutorial/index.html

[10] Streamlit Documentaion:https://docs.streamlit.io/get-started/tutorials/create-an-app