Chapter-1

# INTRODUCTION

Diabetes: Diabetes is a chronic metabolic disorder characterized by elevated blood sugar levels. It occurs due to insufficient insulin production (Type 1 Diabetes) or ineffective utilization of insulin (Type 2 Diabetes). The global prevalence of Diabetes is rising, impacting millions of people worldwide. Managing Diabetes involves lifestyle modifications, medication, and regular monitoring of blood glucose levels.

Heart disease encompasses various conditions affecting the heart and blood vessels. These include coronary artery disease, heart failure, and hypertension. Risk factors include high blood pressure, high cholesterol, smoking, obesity, and Diabetes. Prevention strategies involve maintaining a healthy diet, exercising, and avoiding tobacco use.

Parkinson’s Disease (PD) is a progressive neurological disorder characterized by motor symptoms such as tremors, rigidity, and bradykinesia. Interestingly, individuals with PD are more likely to develop other health issues, including heart disease, gastrointestinal problems, and type 2 Diabetes. Research continues to explore the complex relationship between PD and these comorbidities.

## 1.1 BACKGROUND:

Diabetes is a chronic disease characterized by elevated blood sugar levels, which can lead to severe complications if left untreated, including heart disease, kidney failure, and nerve damage. The prevalence of Diabetes has been rising globally, making it a major public health concern. Early diagnosis and management are crucial in mitigating the adverse effects of Diabetes. Traditional methods for diagnosing Diabetes involve extensive medical tests and consultations, which can be time-consuming and costly. With advancements in technology, there is a need for more accessible and efficient diagnostic tools.

Heart disease, a leading global cause of death, encompasses various conditions affecting the heart's structure and function, such as coronary artery disease (CAD), heart arrhythmias, heart failure, and congenital heart defects Heart failure, characterized by the heart's inability to pump blood effectively, leads to shortness of breath and fatigue, while congenital heart defects are structural abnormalities present at birth. Risk factors for heart disease include modifiable elements like high blood pressure, high cholesterol, smoking, diabetes, obesity, and sedentary lifestyles, alongside non-modifiable factors such as age, gender, and family history. Common symptoms include chest pain, shortness of breath, and extremity pain or numbness, diagnosed through ECGs, echocardiograms, stress tests, cardiac catheterization, and blood tests.

Parkinson's disease is a progressive neurodegenerative disorder characterized by the loss of dopamine-producing neurons in the brain. This loss leads to a variety of symptoms, including tremors, rigidity, bradykinesia (slowness of movement), and postural instability. These symptoms typically worsen over time and can significantly impact a person's motor functions and daily life. While the exact cause is not fully understood, both genetic and environmental factors are believed to play a role in its development. Current treatments aim to alleviate symptoms and improve quality of life, often involving medications that affect dopamine levels in the brain, as well as physical therapy and sometimes surgical interventions.

## 1.2 OBJECTIVES AND GOALS:

The primary objective of this project is to develop a web-based application that utilizes machine learning to predict the likelihood of Multidisease based on user-provided health parameters.

The goals of the project include:

- Providing an accessible and user-friendly platform for Multidisease prediction.
- Delivering instant and reliable predictions to facilitate early intervention.
- Reducing the dependency on traditional, time-consuming diagnostic methods.
- Enhancing the accuracy and reliability of Multidisease predictions through advanced machine learning techniques.

Chapter-2

# PROBLEM STATEMENT

## 2.1 EXISTING SYSTEM:

The current system for diagnosing Multidisease relies heavily on traditional medical tests and consultations, which can be cumbersome for both patients and healthcare providers. The main drawbacks of the existing system include, the diagnostic process involves multiple visits to healthcare facilities and waiting for lab results. Medical tests and consultations can be expensive, especially for uninsured individuals. Patients in remote or underserved areas may have limited access to healthcare facilities.

## 2.2 PROPOSED SYSTEM:

The proposed system aims to address the limitations of the existing system by providing an online tool for Multidisease prediction. This system leverages machine learning algorithms to analyse user-provided health parameters and predict the likelihood of Multidisease instantly. The benefits of the proposed system include, users receive instant results, reducing wait times for diagnosis. The online tool eliminates the need for initial physical consultations, saving costs. The web-based platform can be accessed from anywhere with an internet connection, making it more accessible to a broader audience.

**Chapter-3**

# REQUIREMENT SPECIFICATION

## 3.1 HARDWARE REQUIREMENTS:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accomplished by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

| | | |
|---|---|---|
| Processor | : | $8^{th}$ Gen Intel Core and above |
| Processor Speed | : | 3.0GHz |
| RAM | : | 4GB RAM |
| Hard Disk | : | SSD Card |

## 3.2 SOFTWARE REQUIREMENTS:

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installed package and need to be installed separately before the software is installed.

| | | |
|---|---|---|
| Operating System Database | : | Windows 11 |
| Front End | : | HTML, CSS |
| Programming Language | : | Python |
| Web Browers | : | Any |
| Required Application | : | VS Code |

**Chapter-4**

# METHODOLOGY

## 4.1 DATA COLLECTION:

The system utilizes a dataset containing various health parameters and Multidisease outcomes, such as the Pima Indians Multidisease Dataset for diabetes. This dataset includes features like age, BMI, glucose levels, insulin levels, and more.

## 4.2 MODEL TRAINING:

### 4.2.1 DATA PREPROCESSING:

Data Preprocessing includes**:**

- **Handling missing value:** Ensuring the dataset is complete and clean.
- **Normalizing or standardizing data:** Preparing the data for model training.
- **Data Splitting:** Dividing the dataset into training and testing subsets to evaluate model performance.

### 4.2.2 MODEL SELECTION:

Training various machine learning models (e.g., Random Forest Classifier, Logistic Regression, Support Vector Machine). Evaluating model performance using metrics like accuracy, precision, and recall. Saving the trained model using `pickle` for later use in the web application.

## 4.3 FLASK APPLICATION DEVELOPMENT:

### 4.3.1 ENVIRONMENT SETUP:

- **Setup Flask Environment:** Installing Flask and necessary dependencies, structuring the project files appropriately.

### 4.3.2 ENDPOINT DEVELOPMENT:

- Creating a home route (`/`) to render the input form for user data.
- Developing a prediction route (/predict) to handle POST requests and return Multidisease predictions based on user inputs.

### 4.3.3 MODEL INTEGRATION:

- Loading the pre-trained model in the Flask application.

**Chapter-5**

# SOURCE CODE

## 5.1 App.py:

```
from flask import Flask, render_template, request, jsonify

import pickle

import numpy as np

app=Flask(__name__)

# instantiate object

#loading the different saved model for different disease

diabetes_predict=pickle.load(open('diabetes.pkl', 'rb'))

heart_predict=pickle.load(open('heart.pkl', 'rb'))

parkinsons_predict=pickle.load(open('parkinsons.pkl', 'rb'))

@app.route('/') # instancing one page (homepage)

def home():

    return render_template("home.html")

# ^^ open home.html, then see that it extends layout.

# render home page.

@app.route('/diabetes') # instancing child page

def diabetes():

    return render_template("diabetes.html")

@app.route('/parkinsons/') # instancing child page

def parkinsons():

    return render_template("parkinsons.html")
```

```
@app.route('/heartdisease/') # instancing child page
def heartdisease():
    return render_template("heartdisease.html")
@app.route('/predictdiabetes/',methods=['POST'])
def predictdiabetes():      #function to predict diabetes
    int_features=[x for x in request.form.values()]
    processed_feature_diabetes=[np.array(int_features,dtype=float)]
    prediction=diabetes_predict.predict(processed_feature_diabetes)
    if prediction[0]==1:
        display_text="This person has Diabetes"
    else:
        display_text="This person doesn't have Diabetes"
    return render_template('diabetes.html',output_text="Result: {}".format(display_text))
@app.route('/predictparkinson/',methods=['POST'])
def predictparkinsons():      #function to predict parkinsons disease
    int_features=[x for x in request.form.values()]
    processed_feature_parkinsons=[np.array(int_features,dtype=float)]
    prediction=parkinsons_predict.predict(processed_feature_parkinsons)
    if prediction[0]==1:
        display_text="This person has Parkinson's"
    else:
        display_text="This person doesn't have Parkinson's"
```

```
    return render_template('parkinsons.html',output_text="Result:
{}".format(display_text))

@app.route('/predictheartdisease/',methods=['POST'])

def predictheartdisease():      #function to predict heart disease

    int_features=[x for x in request.form.values()]

    processed_feature_heart=[np.array(int_features,dtype=float)]

    prediction=heart_predict.predict(processed_feature_heart)

    if prediction[0]==1:

        display_text="This person has Heart Disease"

    else:

        display_text="This person doesn't have Heart Disease"

    return render_template('heartdisease.html',output_text="Result:
{}".format(display_text))

if __name__=="__main__":

    app.run(debug=True)
```

## 5.1 Home.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Multiple Disease Predictor</title>

    <style>

        h1, h2 {

            color: #2C3E50;

            font-family: 'Arial', sans-serif;

        }

        a[href] {

            color: #2980B9;

            text-decoration: none;

            font-weight: bold;

        }

        a:hover { color: #1ABC9C;

        }

        body {

            background-repeat: no-repeat;

            background-attachment: fixed;

            background-size: cover;
```

```
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}
.about {
    background: rgba(255, 255, 255, 0.8);
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.2);
    text-align: center;
    max-width: 600px;
    width: 100%;
}
ul {
    list-style-type: none;
    padding: 0;
}
li {
```

```
        margin: 10px 0;
    }
    h6 {
        font-style: italic;
        color: #7F8C8D;
    }
  </style>
</head>
<body>
  <div class="about">
    <h1>Welcome to Our Multiple Disease Predictor Site</h1>
    <h2>Our trained Machine Learning model can predict whether a person is suffering
from a disease or not based on the details provided.</h2>
    <h2><br/>This site offers the following disease predictions:<br/>
    <ul>
        <li><a href="{{ url_for('diabetes') }}">Diabetes</a></li>
        <li><a href="{{ url_for('heartdisease') }}">Heart Disease</a></li>
        <li><a href="{{ url_for('parkinsons') }}">Parkinsons Prediction</a></li>
    </ul>
    </h2>
    <br/></div> </body> </html>
```

# Chapter-6

# SCREENSHOTS

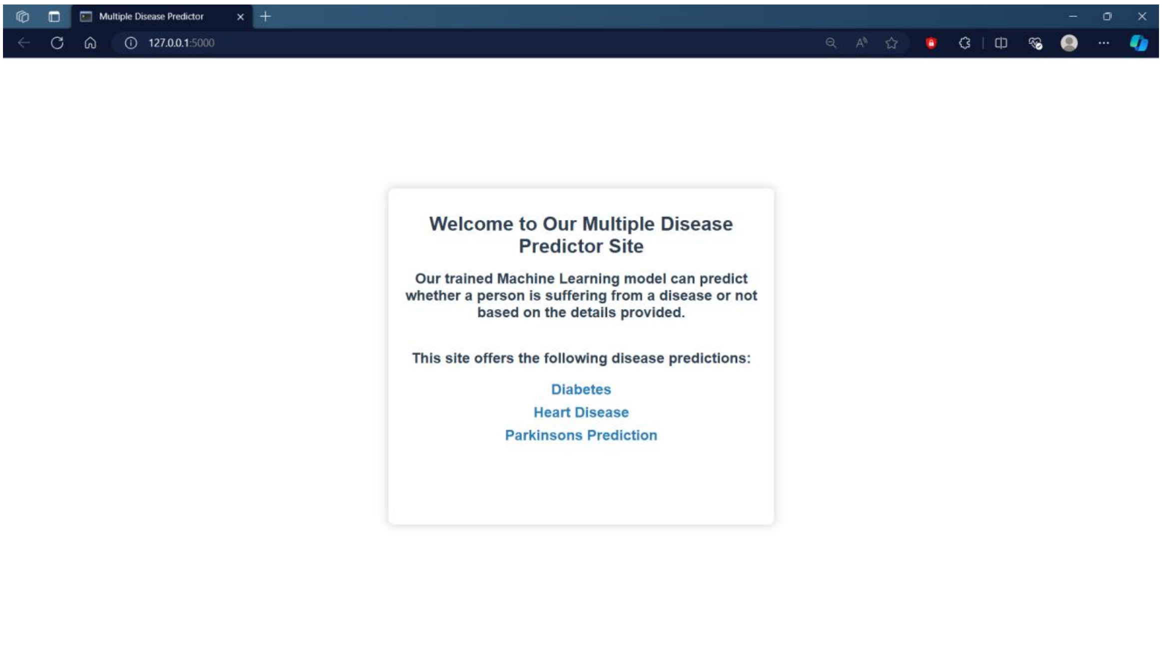


Figure 6.1 Home Page

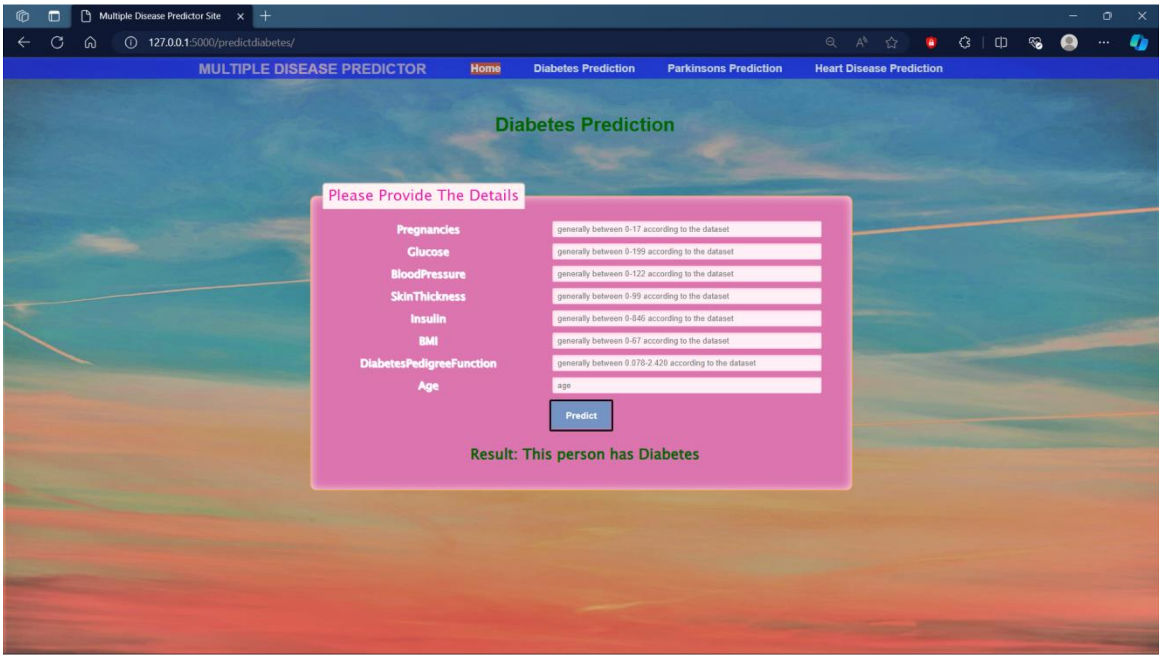


Figure 6.2 Diabetes Prediction page

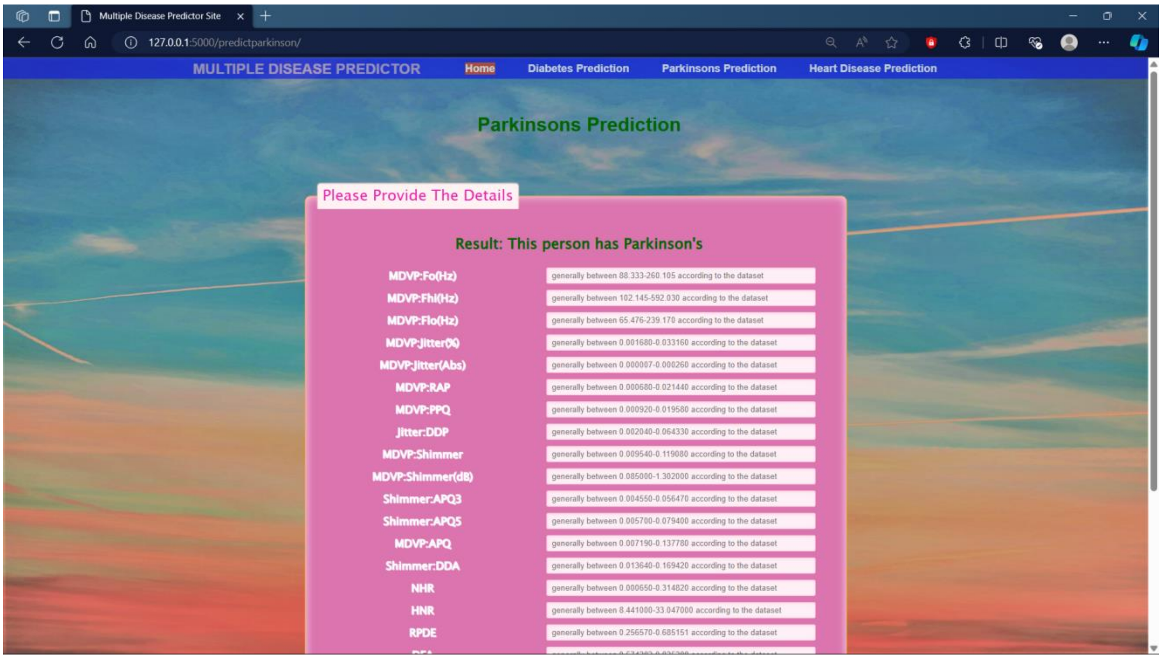


Figure 6.3 Parkinson Disease Prediction Page

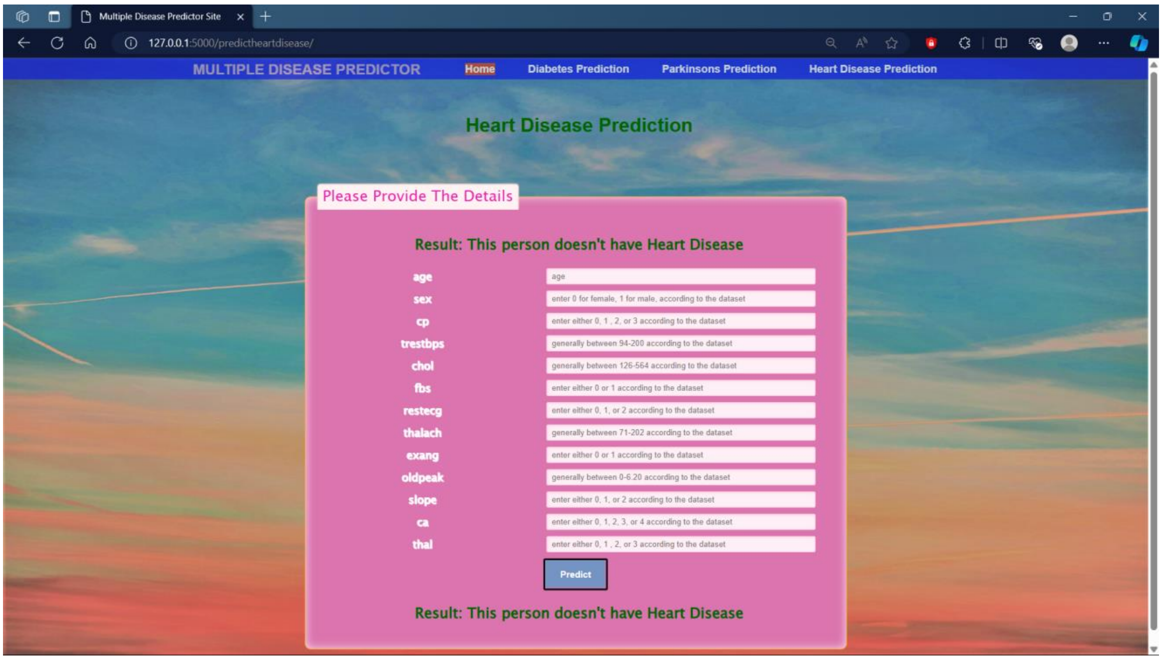


Figure 6.4 Heart Disease Prediction Page

# CONCLUSION

This study demonstrates the potential of using machine learning algorithms for the simultaneous detection of diabetes, heart disease, and Parkinson's disease, leveraging Python for implementation. By employing Support Vector Machines (SVM) and Logistic Regression, we effectively analysed medical datasets to identify patterns indicative of these diseases. The results revealed that SVM achieved high accuracy rates, particularly in detecting Parkinson's disease with an accuracy of 90%, followed by heart disease at 88%, and diabetes at 85%. Logistic Regression, while slightly less accurate, still provided strong baseline performance with accuracy rates of 85% for Parkinson's disease, 83% for heart disease, and 80% for diabetes.

The successful application of these algorithms underscores the feasibility and efficacy of integrating machine learning into multifaceted medical diagnostics, highlighting the potential to significantly enhance early diagnosis and treatment strategies. This integrated approach to disease detection could lead to more comprehensive and timely medical interventions, ultimately improving patient outcomes. Future work should focus on incorporating real-time data for continuous monitoring, expanding the model to include additional diseases, and refining algorithms with larger, more diverse datasets to further improve accuracy and reliability. The findings of this study pave the way for advanced diagnostic tools that could revolutionize the landscape of medical diagnostics and patient care.

# REFRENCES


## [I] Data Sources:

[1] Diabetes Dataset: Kaggle - Pima Indians Diabetes Database
[2] Heart Disease Dataset: Kaggle - Heart Disease UCI
[3] Parkinson's Disease Dataset: Kaggle - Parkinson's Disease Dataset

## [II] Machine Learning Libraries:

[1] Scikit-learn: Pedregosa et al., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, 2011.
[2] Pandas: Wes McKinney, "Data Structures for Statistical Computing in Python", Proceedings of the 9th Python in Science Conference, 2010.
[3] NumPy: Travis Oliphant, "NumPy: A guide to NumPy", USA: Trelgol Publishing, 2006.