AI BASED DIABETES PREDICTION SYSTEM

PHASE 2

TEAM 3

COLLEGE CODE: 3128

🡺REXLIN FELIX S

🡺G GOWTHAM

🡺RAMYA DEVI

🡺M NETHAJI

🡺SIVA RANJINI

Creating an AI-based diabetes prediction system involves several steps, including data preparation, model training, and deployment. Below is a simple implementation in Python using popular libraries such as scikit-learn and Flask for creating a web-based interface. Keep in mind that this is a basic example, and in a real-world scenario, you would need a more extensive dataset and potentially a more sophisticated model.

Step 1: Install Dependencies

bash

pip install flask scikit-learn pandas

Step 2: Create a Python Script

Create a Python script (e.g., diabetes\_prediction.py) with the following code:

python

# Import necessary libraries

from flask import Flask, request, render\_template

import pickle

import numpy as np

# Create a Flask web application

app = Flask(\_\_name\_\_)

# Load the trained machine learning model

with open('diabetes\_model.pkl', 'rb') as model\_file:

model = pickle.load(model\_file)

# Define a route for the home page

@app.route('/')

def home():

return render\_template('index.html')

# Define a route for predicting diabetes

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

def predict():

# Get input values from the form

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

# Convert input values to a NumPy array

input\_data = np.array(features).reshape(1, -1)

# Make a prediction using the trained model

prediction = model.predict(input\_data)

# Display the prediction on a new page

return render\_template('result.html', prediction=prediction[0])

# Run the Flask application

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

Step 3: Create HTML Templates

Create two HTML templates in a folder named templates:

1. index.html:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>Diabetes Prediction</title>

</head>

<body>

<h2>Diabetes Prediction System</h2>

<form action="/predict" method="post">

<label>Pregnancies:</label>

<input type="text" name="Pregnancies" required><br>

<label>Glucose:</label>

<input type="text" name="Glucose" required><br>

<label>BloodPressure:</label>

<input type="text" name="BloodPressure" required><br>

<label>SkinThickness:</label>

<input type="text" name="SkinThickness" required><br>

<label>Insulin:</label>

<input type="text" name="Insulin" required><br>

<label>BMI:</label>

<input type="text" name="BMI" required><br>

<label>DiabetesPedigreeFunction:</label>

<input type="text" name="DiabetesPedigreeFunction" required><br>

<label>Age:</label>

<input type="text" name="Age" required><br>

<button type="submit">Predict</button>

</form>

</body>

</html>

2. result.html:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>Diabetes Prediction Result</title>

</head>

<body>

<h2>Diabetes Prediction Result</h2>

<p>The predicted outcome is: {{ prediction }}</p>

</body>

</html>

Step 4: Train and Save the Model

Train a machine learning model using a suitable dataset and save it using the pickle library. Replace this with your actual machine learning model and dataset.

Step 5: Run the Application

Run your Flask application:

bash

python diabetes\_prediction.py

Remember, this is a basic example, and in a real-world scenario, you'd need to address security, model performance, and other considerations. Additionally, the quality of predictions heavily depends on the quality and representativeness of the training data and the chosen machine learning model. THANK YOU