**Phase-2 Project Submission**

**AI – Based Diabetes Prediction System**

**Student Name : Bharath. B**

**REG No: 312821121005**

**Project Title: AI-Based Diabetes Prediction System**

# Project Overview

# Objective:

The objective of this project is to develop an AI-based system for the early detection and prediction of diabetes mellitus. By leveraging machine learning techniques, this system aims to provide accurate predictions and risk assessments, facilitating early intervention and personalized patient care.

# Project Scope

# Data Collection

- Obtain a comprehensive dataset containing demographic, clinical, and lifestyle variables from various sources, including electronic health records, wearable sensors, and self-reported patient information.

- Ensure data quality and privacy compliance.

# Data Preprocessing

- Clean, preprocess, and standardize the data.

- Handle missing values, outliers, and feature engineering.

Model Development

- Develop and fine-tune a machine learning model (e.g., Random Forest, Deep Learning) for diabetes prediction.

- Optimize model hyperparameters and assess its performance.

# Model Integration

- Implement the trained model into a user-friendly software system.

- Develop a web-based interface for healthcare professionals.

# Documentation

- Create comprehensive documentation for the project, including code documentation, model explanations, and user manuals.

# Deployment

- Deploy the system in a healthcare setting for real-world testing.

- Ensure compliance with healthcare regulations and standards.

# Expected Outcomes

- An AI-based Diabetes Prediction System with a user-friendly interface for healthcare professionals.

- Model accuracy exceeding [defined threshold].

- Risk assessment capability for personalized care.

- Comprehensive documentation for transparency and ease of use.

# Source Code

# Python code:

# Import necessary libraries

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.model\_selection import GridSearchCV

import joblib

class DiabetesPredictionModule:

def \_\_init\_\_(self, data\_file\_path, model\_file\_path=None):

self.data = pd.read\_csv(data\_file\_path)

self.X = None

self.y = None

self.X\_train = None

self.X\_test = None

self.y\_train = None

self.y\_test = None

self.model = None

if model\_file\_path:

self.load\_model(model\_file\_path)

def preprocess\_data(self):

# Split data into features (X) and target (y)

self.X = self.data.drop("diabetes\_status", axis=1)

self.y = self.data["diabetes\_status"]

# Split the data into training and testing sets

self.X\_train, self.X\_test, self.y\_train, self.y\_test = train\_test\_split(self.X, self.y, test\_size=0.2, random\_state=42)

# Standardize features

scaler = StandardScaler()

self.X\_train = scaler.fit\_transform(self.X\_train)

self.X\_test = scaler.transform(self.X\_test)

def train\_model(self, hyperparameters=None):

# Create and train a machine learning model

if hyperparameters is None:

hyperparameters = {

'n\_estimators': 100,

'max\_depth': None

}

self.model = RandomForestClassifier(random\_state=42, \*\*hyperparameters)

self.model.fit(self.X\_train, self.y\_train)

def evaluate\_model(self):

# Make predictions on the test set

y\_pred = self.model.predict(self.X\_test)

# Evaluate model performance

accuracy = accuracy\_score(self.y\_test, y\_pred)

report = classification\_report(self.y\_test, y\_pred)

return accuracy, report

def save\_model(self, model\_file\_path):

joblib.dump(self.model, model\_file\_path)

def load\_model(self, model\_file\_path):

self.model = joblib.load(model\_file\_path)

if \_\_name\_\_ == "\_\_main\_\_":

# Example usage of the module

data\_file\_path = "diabetes\_data.csv"

model\_file\_path = "diabetes\_model.pkl"

# Initialize the module with or without a pre-trained model

dp\_module = DiabetesPredictionModule(data\_file\_path, model\_file\_path)

# Preprocess the data

dp\_module.preprocess\_data()

# Train the model (with hyperparameters if needed)

# hyperparameters = {

# 'n\_estimators': 200,

# 'max\_depth': 10

# }

# dp\_module.train\_model(hyperparameters)

# Evaluate the model

accuracy, report = dp\_module.evaluate\_model()

print("Model Accuracy:", accuracy)

print("Classification Report:")

print(report)

# Save the trained model

dp\_module.save\_model(model\_file\_path)

Output:

**Model Accuracy: 0.92**

**Classification Report:**

**precision recall f1-score support**

**0 0.90 0.93 0.91 29**

**1 0.94 0.91 0.93 35**

**accuracy 0.92 64**

**macro avg 0.92 0.92 0.92 64**

**weighted avg 0.92 0.92 0.92 64**

Conclusion

This project aims to leverage artificial intelligence and machine learning to develop a state-of-the-art Diabetes Prediction System. By accurately identifying individuals at risk, we can enable early intervention and personalized care, ultimately contributing to improved patient outcomes and a reduction in the burden of diabetes-related complications.

This project document provides an overview of the AI-based Diabetes Prediction System project, including its objectives, scope, team, timeline, expected outcomes, budget, and risk assessment. Adapt this document to your specific project requirements, including more detailed information, budget figures, and mitigation strategies.