Diabetes Prediction Model

Project Submission Report

# Introduction

The goal of this project is to develop a diabetes prediction model using machine learning techniques. The dataset used for training and evaluation contains various features related to individuals' health, such as gender, age, hypertension, smoking history, heart disease, BMI, HbA1c level, and blood glucose level. The Diabetes Prediction Project aims to develop a predictive model that can accurately identify individuals at risk of developing diabetes based on various demographic and health-related factors. Early detection of diabetes can significantly contribute to timely intervention and management of the disease, ultimately improving the overall health outcomes for affected individuals.

# Dataset

The project utilizes a comprehensive dataset that includes information on individuals' demographic characteristics, medical history, and key health indicators. The dataset contains the following columns:

* Gender: This categorical variable represents the gender of the individual and includes values such as 'Female', 'Male', and 'Other'.
* Age: This numeric variable denotes the age of the individual, providing insights into the age-related risk factors associated with diabetes.
* Hypertension: This binary variable indicates whether the individual has been diagnosed with hypertension, with '0' representing the absence of hypertension and '1' indicating its presence.
* Heart Disease: Similarly, this binary variable signifies the presence or absence of heart disease in the individual.
* Smoking History: This categorical variable captures the individual's smoking history and includes values such as 'No Info', 'current', 'ever', 'former', 'never', and 'not current'.
* BMI (Body Mass Index): This numeric variable quantifies the individual's body mass index, a measure of body fat based on height and weight.
* HbA1c Level: This numeric variable represents the HbA1c level, which provides an indication of long-term blood sugar control.
* Blood Glucose Level: This numeric variable captures the individual's blood glucose level, which is an essential marker for diabetes risk assessment.
* Diabetes (Target Variable): This binary variable serves as the target variable, with '0' denoting individuals without diabetes and '1' representing individuals with diabetes.

# Methodology

## 1. Data Loading and Preprocessing

The project starts by loading the dataset from a CSV file using the Pandas library. The dataset is then examined for missing values, and it is confirmed that there are no missing values in any of the columns.

## 2. Exploratory Data Analysis

To gain insights into the dataset, the first few rows are displayed using the **head()** function. Additionally, the distribution of the target classes (diabetes/non-diabetes) is checked to understand the class imbalance. It is observed that there is a class imbalance, with a higher number of non-diabetes records compared to diabetes records.

## 3. Data Preprocessing

The dataset is preprocessed to prepare it for model training. The selected columns for prediction are extracted from the dataset, and the corresponding target variable is separated.

Handling Categorical Features

The categorical features in the dataset, such as gender and smoking history, are encoded as numerical values. The values are converted into numerical labels using the **np.where** function, assigning unique numeric values to each category.

Handling Class Imbalance

To address the class imbalance issue, the Synthetic Minority Over-sampling Technique (SMOTE) is applied. SMOTE generates synthetic samples for the minority class (diabetes) to balance the dataset. The **imblearn** library is used to perform the oversampling, and the number of diabetes records is verified after oversampling.

## 4. Data Normalization

Before training the machine learning model, the input data is normalized using TensorFlow's **Normalization** layer. The **Normalization** layer learns the mean and variance of the input data to normalize it effectively. The normalized data is then obtained using the **adapt** and **numpy** functions.

## 5. Model Training and Evaluation

The diabetes prediction model is trained using the XGBoost algorithm. Here's a step-by-step breakdown of the training process:

1. Splitting the Data: The preprocessed dataset is split into training and testing sets using the **train\_test\_split** function from the **sklearn.model\_selection** module. The dataset is divided into input features (X) and target variable (Y), with 33% of the data allocated for testing (**test\_size=0.33**).
2. Model Initialization: An instance of the **XGBClassifier** class from the **xgboost** library is created. This class provides an implementation of the XGBoost algorithm for classification tasks.
3. Model Fitting: The XGBoost classifier is trained on the training data using the **fit** method. During training, the model learns to find patterns and relationships between the input features (X) and the corresponding target variable (Y).

## 6. Evaluation Metrics

The model's performance is evaluated using several metrics:

* F1 Score: A measure of the model's accuracy, considering both precision and recall.
* Precision: The proportion of correctly predicted positive samples (diabetes) out of all predicted positive samples.
* Recall: The proportion of correctly predicted positive samples (diabetes) out of all actual positive samples.
* Accuracy: The proportion of correctly predicted samples out of all samples.

# Results

After training and evaluating the model, the following results were obtained:

* F1 Score: 0.982637668056323
* Precision: 0.9970779110461758
* Recall: 0.9686097174544495
* Accuracy: 0.9828282828282828

These results indicate that the developed diabetes prediction model achieved high accuracy and performed well in terms of precision and recall, indicating a good balance between correctly identified positive and negative samples.

# User Interface

To provide an accessible and user-friendly interface, the project incorporates a graphical user interface (GUI) developed using the Tkinter library. The GUI allows users to input the relevant information, such as gender, age, medical history, and health indicators. Upon submission, the model predicts the likelihood of diabetes and presents the prediction results.

# Conclusion

In conclusion, this project successfully developed a diabetes prediction model using machine learning techniques. The model was trained on a dataset containing various health-related features and achieved high accuracy and performance in predicting diabetes. The model's performance was evaluated using commonly used metrics, including F1 score, precision, recall, and accuracy. The results demonstrate the model's effectiveness in identifying individuals with diabetes based on the provided features.

The trained model (**diabetes\_model1**) and the normalization layer (**Normalized\_model**) were saved for future use. However, it is important to note that further validation and refinement may be required before deploying the model in real-world settings.

This project contributes to the ongoing efforts in utilizing machine learning and data analysis to improve disease prevention, early detection, and personalized healthcare. The developed model can assist healthcare professionals in identifying individuals at high risk of diabetes and facilitate timely interventions for better patient outcomes.

Further improvements and optimizations can be explored, such as feature selection, hyperparameter tuning, and utilizing other machine learning algorithms to compare performance. Additionally, the model can be deployed in real-world scenarios to assist in diabetes prediction and prevention efforts.