Diabetes Prediction System using Machine Learning

Project Overview

I developed a desktop application that uses machine learning to predict diabetes based on medical measurements. The goal was creating a practical tool for quick health assessments that anyone can use.

Objective

The main idea was making diabetes screening more accessible. Users can get a preliminary assessment by entering basic health information, and the application keeps track of previous tests for health monitoring over time.

Machine Learning Model

I used the Random Forest algorithm trained on the Pima Indians Diabetes Database (768 samples). The model analyzes 8 health factors:

- Pregnancies, Glucose level, Blood pressure, Skin thickness
- Insulin levels, BMI, Diabetes pedigree function, Age

The model achieved approximately 85% accuracy on the test set.

Technologies Used

Machine Learning & Data:

- scikit-learn (Random Forest classifier)
- pandas (data processing)
- joblib (model serialization)

Application Interface:

- Tkinter (desktop GUI)
- matplotlib (data visualization)

Database:

- MySQL via XAMPP (local server)

- mysql-connector-python (database integration)

Key Features

Authentication System: Secure login and registration with input validation

Prediction Interface: User-friendly form with tooltips, real-time validation, and instant AI-powered results with health advice

Health History Dashboard: Complete test history with timestamps, statistics summary, and bar chart visualization

User Profile: Personalized welcome message and logout functionality

---

Technical Implementation

The application flow is straightforward:

- User authentication checked against MySQL database
- Health data validated and processed by the ML model
- Predictions saved to database with timestamps
- History retrieved and visualized on demand

I used XAMPP for local MySQL database management, which made development and testing easier.

Challenges & Learning

The most challenging part was creating a responsive GUI with Tkinter, especially handling scrolling and window management between login, register, and main screens.

This project taught me how to integrate different components: connecting a trained ML model with a database-backed GUI application. It's quite different from training models in notebooks - you need to handle user input, data persistence, and error handling properly.

Future Improvements

- Implement password encryption for better security
- Compare multiple ML algorithms (SVM, Neural Networks)
- Add PDF export functionality for health reports

- Consider web-based version for easier accessibility

Conclusion

This project combines machine learning with practical software development, demonstrating how AI can be applied to real-world healthcare problems. Building it provided hands-on experience with Python development, database design, GUI creation, and machine learning integration.

Developer: Baha Taki Eddine Ben Abdallah
Tools: Python, scikit-learn, MySQL (XAMPP), Tkinter