Shko Jabbar Jawhar
01-22-00152

Executive Summary

This project implements a complete machine learning pipeline for diabetes prediction using the Pima Indians Diabetes Database. The solution tackles key challenges such as missing values and class imbalance to deliver a robust, interpretable Logistic Regression model. A Tkinter-based graphical user interface (GUI) enables healthcare professionals to input patient data and receive real-time diabetes risk predictions along with confidence scores. The system achieves a predictive accuracy of 75–80%, demonstrating the potential of machine learning in improving medical diagnostics.

---

1. Dataset Overview

The dataset, sourced from Kaggle, contains diagnostic data for 768 female patients, including 8 medical features (e.g., glucose, BMI, age) and a binary target variable (0 = non-diabetic, 1 = diabetic).
Key challenges addressed:

- **Missing Values:** Zero values in Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI treated as missing and imputed using medians.
- **Class Imbalance:** Approximately 65% non-diabetic, 35% diabetic.

NOTE: Dataset can be found in this Link

---

2. Methodology

Data Preprocessing:

- Missing values imputed using feature-wise medians.
- StandardScaler used for normalization.
- Stratified 80/20 train-test split to maintain class ratio.

Model Development:

- Logistic Regression selected for transparency and interpretability, using the 'liblinear' solver.
- Evaluation metrics: accuracy, precision, recall, F1-score, and ROC AUC.

System Architecture:

- `preprocess.py`: Data cleaning, imputation, and scaling with saved preprocessing objects.
- `train_model.py`: Model training, evaluation, and serialization.

- `gui_app.py`: User-friendly GUI for real-time prediction using Tkinter.

---

3. Results and Impact

- **Accuracy:** 75–80% prediction accuracy.
- **Important Features:** Glucose, BMI, and age found to be most influential.
- **Usability:** Offers binary predictions with confidence scores, supporting data-driven decisions in clinical settings.

---

4. Technical Details

- **Languages & Tools:** Python, scikit-learn, pandas, numpy, tkinter, joblib.
- **Highlights:** Real-time GUI, confidence scoring, modular design, and robust error handling.
- **Deployment:** Fully functional GitHub repository with clean code and workflow scripts.