Brief Report

Glucobuddy: Detecting Diabetes Risk Using Machine Learning

Md Sultanul Arefin Afnan *

Posted Date: 28 May 2025

doi: 10.20944/preprints202505.2269.v1

Keywords: Diabetes Prediction; Machine Learning; Early Detection; Risk Classification

 doi:10.20944/preprints202505.2269.v1

Brief Report

Glucobuddy: Detecting Diabetes Risk Using Machine Learning

Subtitle: A Machine Learning and AI Chatbot-Based Approach for Early Diabetes Risk Prediction

Md Sultanul Arefin Afnan

Dalian Polytechnic University Graduation Project/Thesis, Major: Computer Science & Technology; 219j10@xy.dlpu.edu.cn

Abstract: Diabetes mellitus is a chronic disease affecting over 420 million people globally, contributing significantly to mortality, disability, and healthcare costs. Early detection and risk assessment are critical in preventing severe complications such as cardiovascular disease, kidney failure, and neuropathy. Traditional diagnostic approaches, including fasting glucose and HbA1c testing, require medical infrastructure and trained personnel, making them difficult to access in resource-limited areas. This thesis presents Glucobuddy, an intelligent system designed to predict diabetes risk levels using machine learning models and to enhance user interaction through an integrated AI chatbot. The system analyzes key health indicators including age, glucose levels, and body mass index (BMI) to classify individuals into low-risk or high-risk categories. Three machine learning algorithms—Logistic Regression, Random Forest, and Support Vector Machines (SVM)—are evaluated and compared using performance metrics such as accuracy, precision, recall, and F1-score. In addition to automated risk classification, Glucobuddy incorporates an AI-powered chatbot designed to communicate results, provide general diabetes education, answer common queries, and suggest preventive actions based on the user's risk profile. This interactive approach aims to enhance user understanding and engagement. The proposed system offers a cost-effective, accessible, and scalable solution for early diabetes risk screening, with particular focus on underserved communities. It provides healthcare professionals and individuals with a practical tool for early intervention, contributing to improved health outcomes and reduced healthcare burdens.

Keywords: diabetes prediction; machine learning; early detection; risk classification

1. Introduction

1.1. Background

Diabetes mellitus is a chronic, progressive metabolic disorder characterized by high levels of blood glucose, which, if left unmanaged, can lead to severe and irreversible complications. The World Health Organization (WHO) estimates that over 420 million people worldwide currently live with diabetes, and this number is expected to rise to 578 million by 2030, making it one of the most significant global public health challenges. [1] Type 2 diabetes accounts for over 90% of all cases and is strongly associated with lifestyle-related factors such as obesity, poor diet, sedentary behavior, and genetic predisposition. It is also a leading cause of morbidity and mortality, responsible for an estimated 1.5 million deaths globally in 2019. [2,3]. Early detection and intervention are critical to preventing the progression of diabetes and mitigating associated complications such as **cardiovascular disease, kidney failure, retinopathy, neuropathy, and premature death.** Clinical evidence strongly suggests that lifestyle interventions, including weight loss, dietary improvements, and increased physical activity, can delay or even prevent the onset of Type 2 diabetes in high-risk individuals. However, current diagnostic methods—including **fasting plasma glucose**, **oral glucose tolerance tests**, **and glycated hemoglobin (HbA1c)** measurements—require clinical laboratory

 doi:10.20944/preprints202505.2269.v1

facilities and trained healthcare professionals. The cost and complexity of these procedures create barriers to accessibility, especially in low-resource and rural communities where healthcare infrastructure is often inadequate.

Technological advancements over the past decade have introduced innovative ways to tackle this global epidemic. Machine learning (ML), a branch of artificial intelligence, has shown tremendous potential in healthcare applications by processing large datasets and identifying hidden patterns that are difficult to discern through traditional statistical methods. In recent years, numerous studies have explored the application of machine learning algorithms for the prediction and diagnosis of diabetes. For example, researchers have successfully utilized models such **as Decision Trees, Support Vector Machines (SVM), Random Forest, and Logistic Regression** on medical datasets like the PIMA Indian Diabetes dataset to achieve high prediction accuracies. A study by Sisodia and Sisodia (2018) demonstrated the effectiveness of Decision Tree and SVM models in diabetes prediction, achieving accuracy rates of over 78%. Similarly, the work of Kavakiotis et al. (2017) provided a comprehensive review of data mining and machine learning techniques in diabetes research, further highlighting the promising capabilities of these methods. [5,6]. Despite these advances, many studies focus primarily on binary classification (diabetic vs. non-diabetic), with fewer efforts directed at developing models for early risk prediction. Risk prediction—estimating an individual's probability of developing diabetes in the near future—is an emerging research area that could empower individuals to take preventive action before clinical diagnosis becomes necessary. A system designed for risk prediction could serve as a cost-effective, non-invasive, and scalable screening tool that complements existing diagnostic methods, particularly in resource-constrained settings. [7].

This research proposes Glucobuddy, a novel system designed to harness the power of machine learning to predict diabetes risk levels based on easily obtainable health indicators, specifically age, blood glucose levels, and body mass index (BMI). The simplicity of these input variables makes the system widely applicable and highly practical, especially for primary care providers and community health workers in low-resource environments. The core of Glucobuddy lies in its integration of predictive ML algorithms—**Logistic Regression, Random Forest, and SVM**—to classify individuals into low-risk and high-risk categories, thereby facilitating early interventions.

Furthermore, Glucobuddy distinguishes itself by incorporating an AI-powered chatbot to enhance user engagement and accessibility. The chatbot serves as a digital health assistant, capable of explaining risk assessments, providing diabetes education, answering user queries, and recommending lifestyle modifications based on individual risk profiles. This conversational interface is intended to bridge the gap between complex algorithmic predictions and user understanding, empowering individuals to make informed decisions about their health [8].

The fusion of machine learning and conversational AI in Glucobuddy presents an innovative, user-friendly solution aimed at democratizing diabetes risk assessment. By addressing barriers such as cost, access, and patient engagement, this system aspires to contribute meaningfully to global diabetes prevention efforts. Its design specifically targets underserved populations, offering a scalable tool that can be deployed in both clinical and community settings to improve early detection and support proactive health management.

1.2. Overview of the Project

Diabetes remains one of the most urgent health challenges of modern times, with incidence and mortality rates continuing to rise globally. Despite substantial advancements in medical diagnostics and treatment, many individuals in low-resource settings still lack access to timely screening and early intervention. The Glucobuddy project aims to bridge this critical healthcare gap by leveraging machine learning and artificial intelligence technologies to create an affordable, non-invasive, and scalable solution for early diabetes risk prediction and personalized health guidance.

The core objective of the Glucobuddy project is to develop an intelligent system capable of classifying individuals as either low-risk or high-risk for developing diabetes. Unlike traditional

 doi:10.20944/preprints202505.2269.v1

diagnosis, which relies on laboratory tests and medical supervision, Glucobuddy is designed to provide predictive risk assessment using easily obtainable personal health indicators. The three primary features targeted for prediction are **age, blood glucose level, and body mass index (BMI)**. These parameters were chosen due to their strong association with diabetes risk and their availability in standard health screening environments. [6,9,10].

The Glucobuddy system is composed of two main components: the predictive model and the conversational AI chatbot. The predictive model leverages supervised machine learning algorithms trained on historical health datasets. Three well-established models—Logistic Regression, Random Forest, and Support Vector Machines (SVM)—have been selected due to their proven effectiveness and interpretability in healthcare classification tasks. Each model will be trained and evaluated using performance metrics such as accuracy, precision, recall, and F1-score to determine the optimal algorithm for risk classification. The model with the highest balanced performance will be integrated into the final system.

In parallel, the AI chatbot component serves as the interactive layer of Glucobuddy. While the machine learning model provides a risk score, the chatbot contextualizes and communicates this information to the user. Using **natural language understanding (NLU)**, the chatbot can explain the predicted risk, offer personalized advice on lifestyle changes, answer common questions about diabetes, and suggest follow-up actions. The chatbot is envisioned to operate 24/7, providing immediate access to health information, especially in areas where professional medical consultation may not be available. This human-computer interaction element is a significant enhancement over existing prediction systems, promoting user engagement, understanding, and empowerment.

The data pipeline for Glucobuddy includes data collection, preprocessing, model training, evaluation, and deployment. Data preprocessing will involve handling **missing values, outliers, normalization, and addressing class imbalance** using techniques such as Synthetic Minority Over-sampling Technique (SMOTE). Once the data is cleaned and transformed, the machine learning models will be trained and cross-validated to prevent overfitting and to ensure generalizability.

Following model deployment, the system will undergo rigorous testing to validate its real-world usability and robustness. The AI chatbot will be integrated with the prediction model through API communication, allowing real-time risk prediction and conversational feedback. User acceptance testing (UAT) and simulated patient cases will be conducted to assess both the predictive accuracy and the quality of the chatbot interactions.

The Glucobuddy project also emphasizes accessibility and scalability. The system is intended to be lightweight, enabling it to run on basic mobile and web platforms. This design decision makes it suitable for use in community clinics, rural health centers, and even at-home personal health monitoring. The ultimate vision for Glucobuddy is to serve as a preventive healthcare companion that complements existing clinical practices, reduces the diagnostic burden on healthcare systems, and empowers individuals to take proactive control of their health.

By combining the analytical power of machine learning with the conversational capabilities of AI, Glucobuddy represents an innovative approach to early diabetes risk screening. This research expects to contribute to the emerging field of AI-assisted preventive healthcare by demonstrating the feasibility and impact of integrated predictive and interactive systems. Upon successful implementation, Glucobuddy could become a model framework for similar disease risk assessment applications beyond diabetes, laying the groundwork for the next generation of accessible digital health solutions.

1.3. Current System

Diabetes diagnosis and risk assessment are traditionally conducted within clinical settings, relying on established medical protocols and laboratory-based testing. The most widely used diagnostic methods include fasting plasma glucose tests, oral glucose tolerance tests (OGTT), and glycated hemoglobin (HbA1c) tests. While these methods are considered the gold standard by organizations such as the **World Health Organization (WHO)** and the American Diabetes

 doi:10.20944/preprints202505.2269.v1

Association (ADA), they share several inherent limitations that restrict their accessibility and effectiveness, particularly in underserved regions. [1,2]

Firstly, these traditional testing methods require specialized equipment, controlled environments, and trained healthcare professionals to ensure accuracy and safety. This creates a barrier for early screening in rural and low-income areas where medical infrastructure and resources are limited. Patients in such areas often face challenges including long travel distances to health facilities, high costs of diagnostic services, and long wait times for appointments and test results. Consequently, many individuals remain undiagnosed until the disease has progressed to more advanced stages, when complications are harder and more expensive to manage.

In addition to logistical barriers, traditional diagnostic methods are invasive and time-consuming. Blood samples must be collected, processed, and analyzed in a laboratory. For example, the **OGTT** requires patients to fast overnight, consume a glucose solution, and undergo multiple blood draws over a two-hour period. These factors contribute to patient discomfort and poor adherence to regular screening schedules, further exacerbating the risk of delayed diagnosis.

To address some of these limitations, a number of mobile applications and health monitoring devices have emerged in recent years to provide diabetes management and tracking solutions. However, most of these technologies focus on post-diagnosis monitoring rather than early detection or risk prediction. Common applications offer features such as blood glucose logging, medication reminders, and diet tracking, but they do not typically integrate predictive algorithms capable of assessing an individual's future risk of developing diabetes based on health indicators. [11,12]

Moreover, the few predictive tools that do exist are often designed for research or academic purposes and lack the user-centered design and conversational interface needed for real-world deployment in community or primary care settings. They are usually static and offer limited interactivity, providing risk scores without adequate explanation, context, or actionable recommendations for users.

As a result, there remains a significant unmet need for a comprehensive system that not only predicts diabetes risk based on accessible health data but also actively engages users in understanding and managing their health status. There is currently no widely adopted system that combines advanced machine learning models with an AI-driven conversational interface to guide users through the risk assessment process, explain their results, and recommend personalized preventive measures.

This gap presents the opportunity for Glucobuddy to offer a transformative solution. By integrating machine learning-based risk prediction with an AI chatbot capable of delivering real-time explanations and guidance, Glucobuddy addresses the shortcomings of both traditional diagnostic methods and current mobile health applications. It aims to empower individuals to take proactive steps toward diabetes prevention, especially in areas where access to professional medical consultation is limited.

1.4. Proposed System

The proposed system, Glucobuddy, aims to address the limitations of current diabetes diagnostic and monitoring approaches by providing an intelligent, accessible, and user-friendly early risk assessment tool. Glucobuddy combines machine learning-based risk prediction with an integrated AI-powered chatbot to deliver a complete and engaging user experience. It is designed to be easily accessible through mobile devices or web platforms, making it suitable for deployment in both clinical and non-clinical environments, including remote and resource-limited communities. [13,14]

The system operates in two main phases: data-driven risk prediction and conversational feedback.

In the first phase, users are prompted to enter basic health information including age, body mass index (BMI), and blood glucose levels. These variables were selected based on their proven correlation with Type 2 diabetes risk and their widespread availability through basic health

 doi:10.20944/preprints202505.2269.v1

screenings. After data input, the information is preprocessed and analyzed using trained machine learning algorithms. Glucobuddy leverages three well-established models—Logistic Regression, Random Forest, and Support Vector Machines (SVM)—to classify individuals into two categories: low-risk and high-risk for developing diabetes.

The selection of multiple algorithms allows for performance comparison and ensures that the most accurate and reliable model can be integrated into the final system. The model training phase includes data cleaning, handling of missing values, normalization, and mitigation of class imbalance using Synthetic Minority Over-sampling Technique (SMOTE). These techniques improve the model's robustness and predictive capability across diverse patient data.

Once the machine learning model generates a prediction, the second phase of the system is activated. The AI-powered chatbot takes over as the user interface, delivering the prediction results in an understandable and friendly conversational format. Rather than presenting raw numerical risk scores, the chatbot explains the result, outlines possible health implications, and provides recommendations for lifestyle changes or encourages the user to consult a healthcare professional for further evaluation.

The chatbot is designed using natural language understanding (NLU) technology, enabling it to understand user queries and respond to them contextually. Users can ask follow-up questions about diabetes, healthy diets, exercise routines, or specific explanations about their risk level. The chatbot serves as a virtual health assistant available 24/7, providing reliable information and support at any time.

One of the key strengths of Glucobuddy is its focus on accessibility and scalability. The system is intentionally lightweight and designed to function efficiently on basic smartphones and computers, minimizing technological barriers to adoption. Its intuitive design ensures that users with limited technical literacy can easily navigate the interface and receive valuable health guidance.

Another important feature is privacy and data protection. Glucobuddy is designed to operate with anonymized or consent-based data to ensure compliance with healthcare data privacy regulations. No personal identifiers are stored or transmitted without user approval.

In summary, the proposed Glucobuddy system introduces an innovative combination of machine learning and conversational AI to create a user-centered solution for early diabetes risk assessment. By simplifying the prediction process and improving user engagement through interactive dialogue, Glucobuddy has the potential to complement traditional healthcare systems, reduce the diagnostic burden on clinics, and empower individuals to take a proactive role in managing their health.

This dual system approach not only predicts potential diabetes risk but also closes the communication gap by providing understandable, actionable insights directly to users. It represents a step forward in making predictive healthcare technologies accessible, interactive, and practical for widespread use, particularly in settings where healthcare access is limited.

1.5. Scope of the Project

The scope of this project focuses on the design, development, and evaluation of an intelligent system for early diabetes risk prediction using machine learning techniques, integrated with an AI-powered chatbot for user interaction and feedback. The project is structured around creating a proof-of-concept application that demonstrates the feasibility and potential of combining predictive analytics with conversational AI to enhance preventive healthcare services.

The primary objective is to develop a system that accepts basic health indicators—age, body mass index (BMI), and blood glucose levels—as inputs to predict an individual's likelihood of developing Type 2 diabetes. The machine learning component will be limited to three widely recognized classification algorithms: Logistic Regression, Random Forest, and Support Vector Machines (SVM). These models will be trained and tested using existing publicly available health datasets. The project will evaluate each model's performance based on standard classification metrics such as accuracy, precision, recall, and F1-score to determine the most effective predictive model.

 doi:10.20944/preprints202505.2269.v1

The AI chatbot component is designed to enhance the user experience by communicating the risk assessment results and providing general lifestyle recommendations. The chatbot will use natural language processing (NLP) to understand and respond to user queries. However, the chatbot's advice is strictly informational and not intended to replace professional medical consultation or diagnosis.

This project is intended as a research prototype and does not aim to build a fully operational commercial product. The scope does not include integration with wearable devices, continuous glucose monitors, or electronic health record (EHR) systems. Additionally, while the system will prioritize user privacy and data security, this project will not conduct formal compliance certifications such as HIPAA or GDPR audits.

The outcome of this project is expected to be a functional prototype system capable of demonstrating the effectiveness of combining machine learning and conversational AI for early diabetes risk screening. The system is designed to be scalable for future extensions but will remain within the limits of a research and educational study during this development phase.

2. Methodology (Analysis and Design)

2.1. Data Collection and Preprocessing

The success of any machine learning project heavily depends on the quality and suitability of the data used. For this research, the dataset used to train and evaluate the machine learning models is derived from a publicly available medical dataset: the **PIMA Indians Diabetes Database**, provided by the National Institute of Diabetes and Digestive and Kidney Diseases (**NIDDK**). This dataset has been widely used in diabetes-related research and serves as a standard benchmark for classification and predictive modeling tasks. [10]

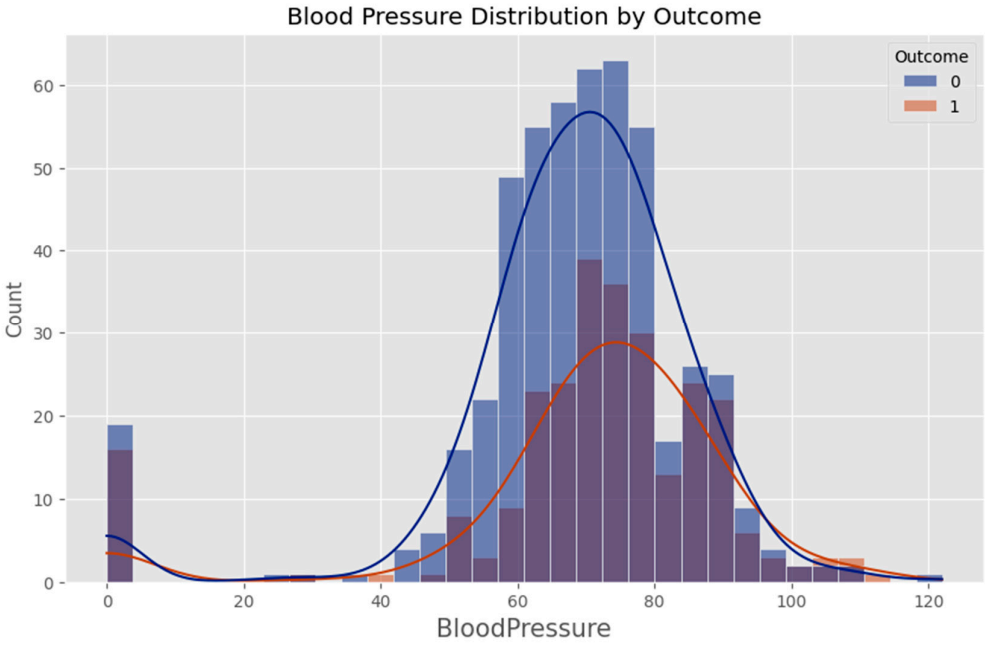

Figure 1. Blood pressure distribution grouped by diabetes outcome.

The **PIMA** dataset contains records of female patients of Pima Indian heritage, aged 21 years and older, and includes 768 instances with 8 medical and personal attributes. For the purpose of this study, only three key features were selected based on their strong correlation with diabetes risk and ease of measurement in typical clinical or community settings:

 doi:10.20944/preprints202505.2269.v1

1. **Age (years)** – An important demographic factor associated with increased risk of Type 2 diabetes.
2. **Body Mass Index (BMI)** – A widely used measure of body fat based on height and weight.
3. **Blood Glucose Level (mg/dL)** – A critical biomarker directly linked to diabetes risk.

The outcome variable in the dataset indicates whether the patient was diagnosed with diabetes (1) or not (0). For this research, the outcome variable is adapted to represent two risk levels: low risk (0) and high risk (1).

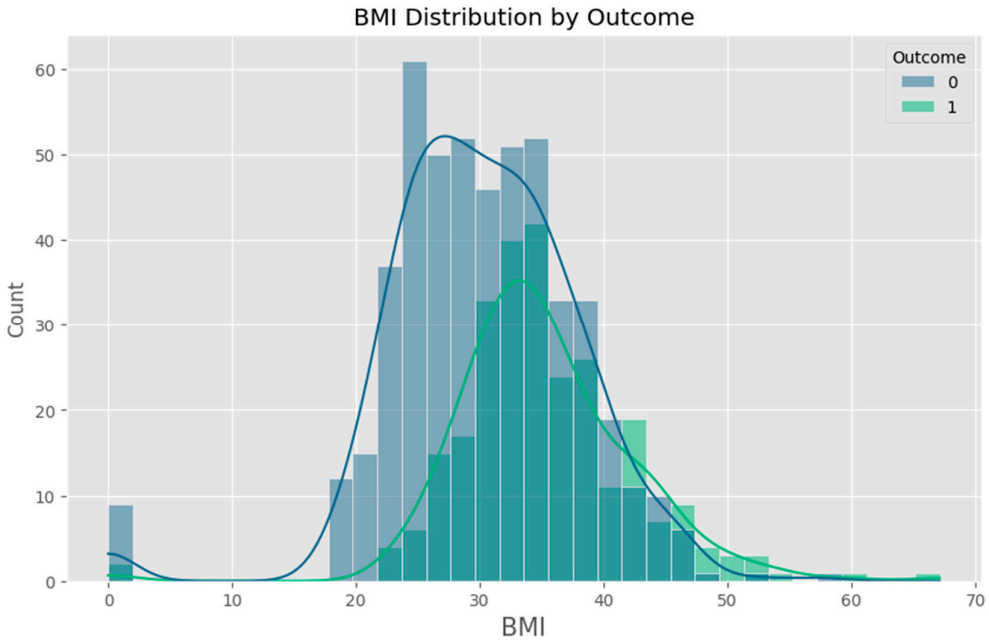

Figure 2. BMI distribution by diabetes outcome.

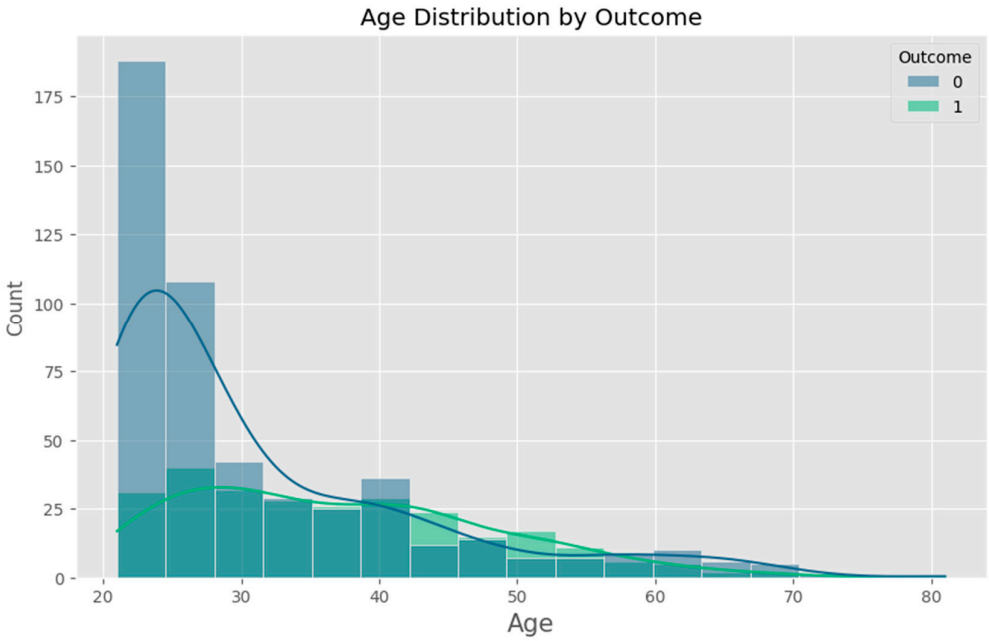

Figure 3. Age distribution by diabetes outcome.

 doi:10.20944/preprints202505.2269.v1

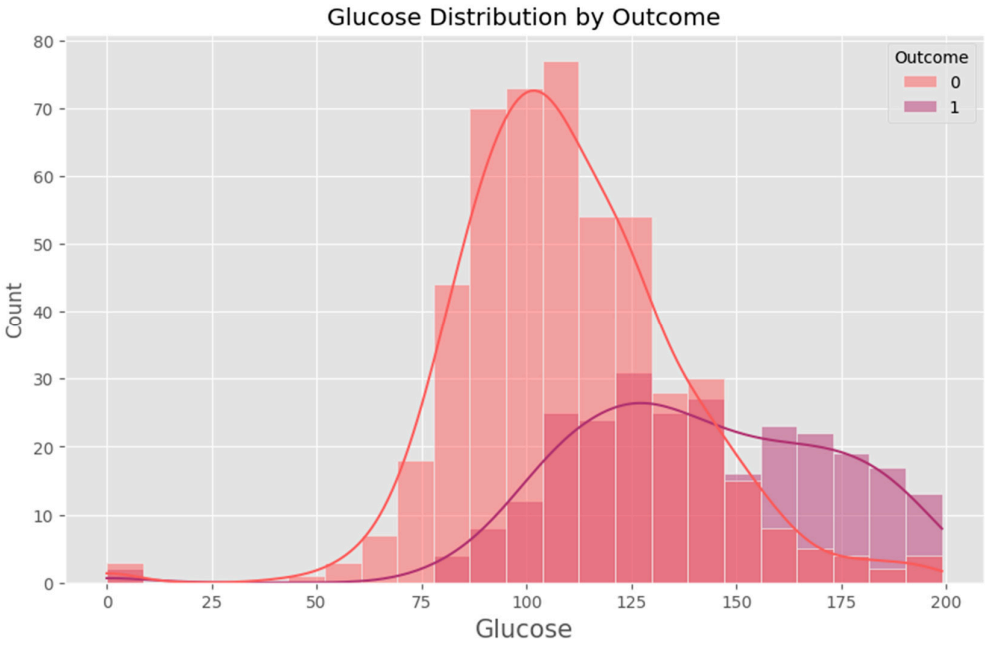

Figure 4. Glucose distribution by diabetes outcome.

2.1.1. Data Cleaning

Real-world medical datasets are often incomplete, noisy, and inconsistent. Several preprocessing steps were applied to ensure the quality of the data:

- **Handling Missing Values:** The dataset contains instances where critical measurements, such as BMI and glucose levels, were recorded as zero, which is not physiologically possible. These zero values were treated as missing and replaced using imputation techniques. The median value of the respective feature was used to fill missing values to minimize bias.
- **Outlier Detection and Treatment:** Statistical methods such as interquartile range (IQR) analysis were applied to detect and manage outliers that could skew model performance. Extreme values were capped to reasonable physiological ranges based on clinical guidelines.

 doi:10.20944/preprints202505.2269.v1

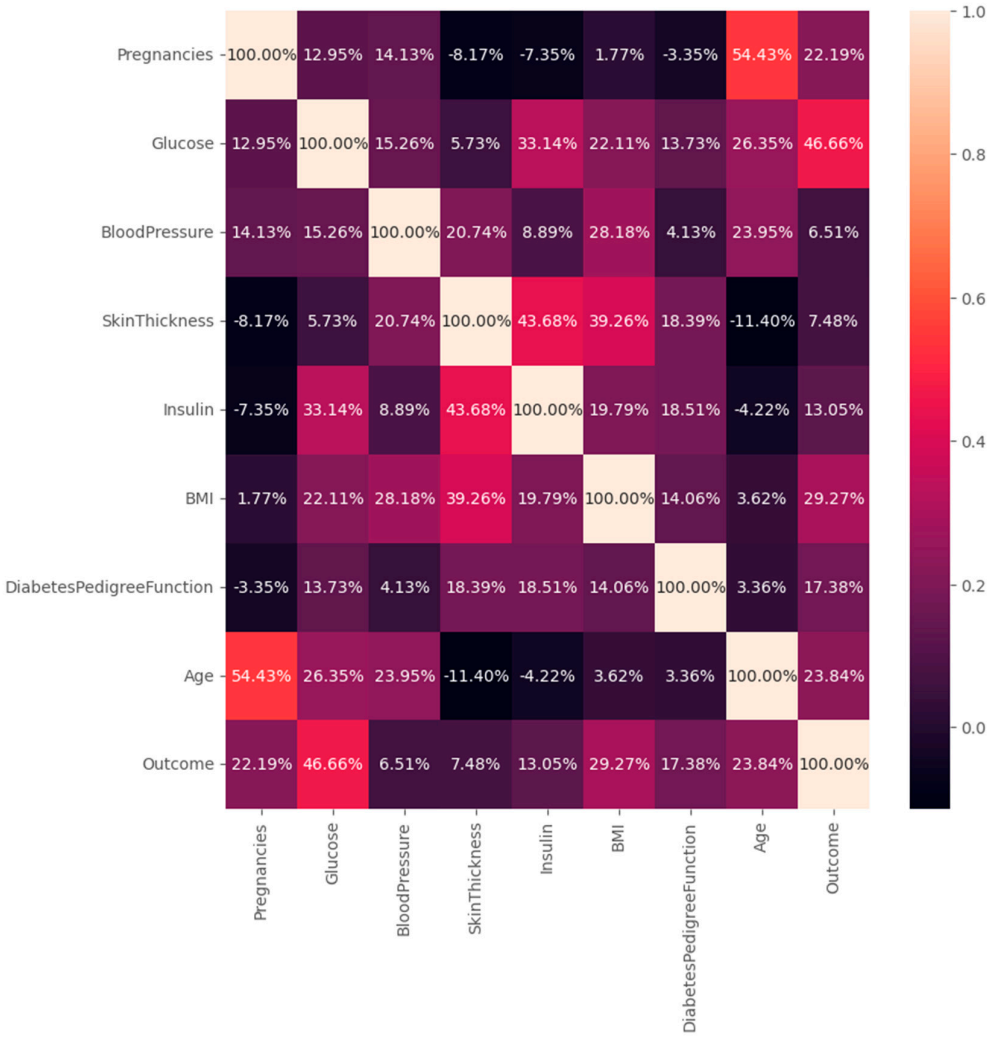

Figure 5. Heatmap distribution for diabetes outcome.

2.1.2. Data Normalization

Since the dataset includes numerical features with varying scales (e.g., glucose levels range from **0 to 200**+, whereas BMI typically ranges from 15 to 50), data normalization was essential. The min-max scaling technique was applied to scale all feature values to a range between 0 and 1. This step ensures that no single feature disproportionately influences the learning algorithm due to its scale. [16]

2.1.3. Data Balancing

The dataset exhibited a slight imbalance between diabetic and non-diabetic cases. To mitigate the bias introduced by class imbalance, the **Synthetic Minority Over-sampling Technique (SMOTE)** was applied. SMOTE works by creating synthetic examples of the minority class based on existing instances, effectively balancing the dataset and improving model generalization. [15]

2.1.4. Data Splitting

The final dataset was divided into two subsets:

- **Training Set (80%)** – Used to train the machine learning models.
- **Test Set (20%)** – Used to evaluate model performance on unseen data.

 doi:10.20944/preprints202505.2269.v1

Stratified sampling was applied to ensure that both the training and test sets maintained the same class distribution as the original dataset.

2.1.5. Summary

The thorough preprocessing pipeline ensured that the dataset was clean, balanced, and appropriately scaled, thus enabling the machine learning models to perform at their best potential. These steps were critical in enhancing the predictive accuracy, robustness, and reliability of the Glucobuddy system.

2.2. Machine Learning Models

The Glucobuddy system leverages the power of supervised machine learning algorithms to predict an individual's risk of developing Type 2 diabetes. Machine learning provides a valuable alternative to traditional statistical analysis by identifying complex, non-linear patterns in the data that may not be immediately visible through conventional methods. For this study, three popular and widely used classification algorithms were selected**: Logistic Regression, Random Forest, and Support Vector Machines (SVM)**. Each algorithm was chosen based on its proven effectiveness, interpretability, and relevance in healthcare data analysis [6,17].

Figure 6. Machine Learning Predictions visualization.

 doi:10.20944/preprints202505.2269.v1

2.2.1. Logistic Regression

Logistic Regression is one of the simplest and most commonly used binary classification algorithms. It models the probability that a given input belongs to a particular class—in this case, either low-risk or high-risk of diabetes. Logistic Regression uses a logistic function to map predicted values to probabilities between 0 and 1. The model works by fitting a linear combination of the input features (age, BMI, glucose level) to predict the **log-odds** of the dependent variable (risk class).

The main advantage of Logistic Regression lies in its ease of implementation, low computational cost, and interpretability. In a medical setting, where understanding the contribution of individual risk factors is crucial, Logistic Regression provides clear insights into the importance of each feature.

2.2.2. Random Forest

Random Forest is a powerful ensemble learning method based on decision trees. It constructs multiple decision trees during training and outputs the mode of their predictions for classification tasks. Random Forest mitigates the risk of overfitting, a common issue with individual decision trees, by introducing randomness in the feature selection and data sampling process.

Random Forest is particularly useful for handling complex, non-linear relationships between features and is robust to noise and outliers. Additionally, Random Forest provides measures of feature importance, helping to understand which variables contribute the most to diabetes risk prediction. Its strong performance across many biomedical datasets makes it an ideal candidate for this project.

2.2.3. Support Vector Machines (SVM)

Support Vector Machines (SVM) are another powerful supervised learning technique suitable for binary classification tasks. SVM works by finding the optimal hyperplane that separates data points of different classes with the maximum possible margin. In cases where data are not linearly separable, SVM uses kernel functions (such as polynomial or radial basis function kernels) to map the input data into a higher-dimensional space where a linear separator can be found.

SVM is highly effective in handling high-dimensional data and performs well when the number of features exceeds the number of samples. Its ability to model complex decision boundaries makes it a strong competitor for diabetes risk prediction. However, SVM is computationally intensive and requires careful tuning of hyperparameters, such as the regularization parameter (C) and kernel type, to achieve optimal results.

2.2.4. Model Evaluation Approach

To assess the performance of these machine learning models, standard classification metrics were used, including accuracy, precision, recall, and F1-score. Accuracy provides a measure of overall correctness, precision evaluates how many predicted positive cases were actually positive, recall measures how many actual positive cases were correctly predicted, and F1-score balances the trade-off between precision and recall. Cross-validation techniques were applied to reduce overfitting and ensure that the models generalize well to unseen data.

2.2.5. Summary

By experimenting with these three algorithms, Glucobuddy aims to identify the most suitable model for accurately classifying individuals into low-risk and high-risk categories. Each algorithm offers unique strengths, and their comparative analysis provides valuable insights into the applicability of machine learning techniques for preventive healthcare solutions.

 doi:10.20944/preprints202505.2269.v1

2.3. Model Design and Evaluation

The design of the Glucobuddy system follows a structured and methodical pipeline to ensure high-quality, accurate, and generalizable predictions of diabetes risk. The model design process involved data preprocessing, model selection, training, validation, and evaluation to identify the most effective predictive approach.

2.3.1. Model Design

The model pipeline begins with the input of user-provided data: age, blood glucose level, and **body mass index (BMI)**. The data is first subjected to preprocessing steps as previously described, including handling missing values, outlier detection, normalization, and class balancing using the Synthetic Minority Over-sampling Technique (SMOTE). These steps are essential to reduce bias and prevent poor model performance caused by data quality issues.

Following preprocessing, the dataset was randomly split into two subsets: 80% for training and 20% for testing. Stratified sampling was used to maintain the proportion of high-risk and low-risk classes across both subsets.

The training data was then used to fit the machine learning models. Three algorithms—Logistic Regression, Random Forest, and Support Vector Machines (SVM)—were chosen based on their proven effectiveness in healthcare classification tasks. Each model was independently trained using the same dataset to ensure a fair comparison of performance.

2.3.2. Model Training and Hyperparameter Tuning

To optimize model performance, hyperparameter tuning was conducted for each algorithm:

- For Logistic Regression, the regularization parameter (C) was adjusted to balance the trade-off between bias and variance.
- For Random Forest, key parameters such as the number of trees (n_estimators), maximum tree depth, and minimum samples per leaf were fine-tuned to improve performance and reduce overfitting.
- For SVM, both the kernel type (linear or radial basis function) and the regularization parameter were carefully tuned to maximize classification accuracy.

Grid search combined with k-fold cross-validation (with k = 5) was applied for hyperparameter optimization. This approach reduces overfitting by ensuring the model's performance is validated on multiple subsets of the training data, providing a more reliable estimate of its generalization capability.

2.3.3. Model Evaluation

Once trained, the models were evaluated on the independent test set using four primary metrics:

1. **Accuracy:** Measures the proportion of correct predictions out of total predictions.
2. **Precision:** Indicates how many of the predicted high-risk cases were actually high-risk.
3. **Recall (Sensitivity):** Measures the ability of the model to correctly identify all actual high-risk individuals.
4. **F1-Score:** Provides a balanced measure that combines both precision and recall. [18]

Additionally, **Receiver Operating Characteristic (ROC)** curves were generated for each model, and the area under the curve (AUC) was calculated. The **ROC-AUC** score offers an overall measure of model performance across all classification thresholds, with a value closer to 1.0 indicating superior discriminatory power.

 doi:10.20944/preprints202505.2269.v1

2.3.4. Performance Comparison and Selection

The models were compared across all metrics, and the one with the best balance of accuracy, precision, recall, and F1-score was chosen for integration with the AI chatbot component of Glucobuddy. While Random Forest and SVM were expected to perform better in capturing complex patterns, Logistic Regression was also considered valuable due to its interpretability and ease of deployment.

2.3.5. Summary

The structured design and rigorous evaluation methodology ensured that the final Glucobuddy model is both accurate and generalizable for predicting diabetes risk. The combination of model tuning, cross-validation, and thorough performance assessment provides a solid foundation for deploying Glucobuddy as a reliable early screening tool.

3. Implementation

3.1. Model Development

The model development phase of Glucobuddy establishes the core predictive capability by preparing the environment, organizing the codebase, and implementing the data pipeline through to model serialization. All experimentation and prototyping were conducted in the server/ directory of the repository, which contains the following key files:

```
server/
├── Diabetes.ipynb        # Jupyter notebook for exploration & prototyping
├── diabetes.csv          # Dataset (PIMA-inspired) used for model training
├── requirements.txt      # Python dependencies
├── scaler.pkl            # Serialized feature scaler
├── lr.pkl                # Serialized Logistic Regression model
├── nb.pkl                # Serialized Naive Bayes model
└── app.py                # Flask API serving predictions
```

3.1.1. Development Environment

- **Python Version:** 3.8
- **Virtual Environment:** Created via python -m venv venv
- **Installation:**
- source venv/bin/activate
- pip install -r server/requirements.txt
- **Core Libraries:**
  - **Data handling:** pandas (v1.x), numpy (v1.x)
  - **Visualization:** matplotlib, seaborn
  - **Modeling:** scikit-learn (v0.24+)
  - **Persistence:** pickle

 doi:10.20944/preprints202505.2269.v1

3.1.2. Data Loading and Inspection

All model development begins in **Diabetes.ipynb**, which loads the dataset:

```
import pandas as pd

df = pd.read_csv('diabetes.csv')
```

Initial exploratory commands—df.head(), df.info(), and df['Outcome'].value_counts()—verify that the file contains 768 records, eight features, and a binary outcome column.

3.1.3. Feature Selection and Label Definition

Although the raw dataset includes eight medical features, Glucobuddy focuses on those most strongly correlated with Type 2 diabetes risk and most feasible for broad screening:

- **Age**
- **Body Mass Index (BMI)**
- **Blood Glucose Level**

The target label (Outcome) is interpreted directly as **0 = low risk**, **1 = high risk** of diabetes.

3.1.4. Data Cleaning and Preprocessing

In the notebook, missing or impossible zero values in key columns (e.g., Glucose, BMI) are imputed using the **median**:

```
for col in ['Glucose','BMI']:
   df[col].replace(0, pd.NA, inplace=True)
        df[col].fillna(df[col].median(), inplace=True)
```

Outliers are then handled via **interquartile range (IQR)** capping based on clinical thresholds to reduce skew.

3.1.5. Feature Scaling

To ensure uniform model convergence, features are scaled using **StandardScaler**:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X = df[['Age','BMI','Glucose']]
X_scaled = scaler.fit_transform(X)
```

After fitting, the scaler object is serialized for deployment:

```
import pickle
      pickle.dump(scaler, open('scaler.pkl','wb'))
```

3.1.6. Train/Test Split

The scaled data and labels are split into training (80%) and testing (20%) sets with **stratified sampling** to preserve class ratios:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
   X_scaled, df['Outcome'],
   test_size=0.2,
   stratify=df['Outcome'],
   random_state=42
```

 doi:10.20944/preprints202505.2269.v1


```
)
```

3.1.7. Prototype Model Training

Within the same notebook, five baseline classifiers are instantiated and trained on X_train/y_train:

1. **Logistic Regression**
2. **K-Nearest Neighbors (KNN)**
3. **Gaussian Naive Bayes**
4. **Random Forest**
5. **Support Vector Machine (SVM)**

Example (Random Forest):

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=91)
rf.fit(X_train, y_train)
```

After training, each model's performance is evaluated on the **test set** using accuracy, precision, recall, F1-score, and ROC-AUC.

3.1.8. Model Selection and Persistence

Based on comparative metrics, the top-performing models—Logistic Regression (lr.pkl) and Gaussian Naive Bayes (nb.pkl)—were chosen for initial deployment. These classifiers are serialized using pickle:

```
pickle.dump(lr_model, open('lr.pkl','wb'))
    pickle.dump(nb_model, open('nb.pkl','wb'))
```

3.1.9. Integration Readiness

With models and scaler persisted, the app.py script loads them at startup:

```
import pickle
scaler = pickle.load(open('scaler.pkl','rb'))
model = pickle.load(open('lr.pkl','rb'))
```

This setup readies Glucobuddy's predictive backend for real-time inference, to be exposed via a Flask REST API and, subsequently, integrated with the conversational AI layer.

3.2. Algorithm Implementation

The algorithm implementation phase of Glucobuddy involved translating the developed machine learning pipeline into a working backend system capable of receiving user inputs, processing the data, and delivering diabetes risk predictions in real time. The primary backend service was developed using **Flask**, a lightweight Python web framework, located in the server/app.py file.

3.2.1. System Architecture

The core backend of Glucobuddy comprises three main components:

 doi:10.20944/preprints202505.2269.v1

1. **Preprocessing module**: Scales the user's input data to match the model's expected input distribution using the previously saved scaler.pkl object.
2. **Model prediction module**: Loads the serialized machine learning model (lr.pkl or nb.pkl) and generates a risk prediction.
3. **REST API interface**: Exposes endpoints for the front-end application (or AI chatbot) to send user inputs and receive prediction results. [20]

The project directory also includes other helper files and dependencies defined in requirements.txt to maintain reproducibility and portability across different environments.

3.2.2. Flask API Structure

The Flask server is designed to handle HTTP POST requests containing the user's health information:

```
from flask import Flask, request, jsonify
import pickle

app = Flask(__name__)

# Load the model and scaler
model = pickle.load(open('lr.pkl','rb'))
scaler = pickle.load(open('scaler.pkl','rb'))

@app.route('/predict', methods=['POST'])
def predict():
  data = request.get_json(force=True)
  values = [[
    data['Age'],
    data['BMI'],
    data['Glucose']
  ]]
  values_scaled = scaler.transform(values)
  prediction = int(model.predict(values_scaled)[0])
  return jsonify({'risk': prediction})
```

The server receives JSON-formatted data, extracts the relevant features, scales them using the fitted StandardScaler, and then passes them to the Logistic Regression model to generate a risk classification. [16,19]

3.2.3. Model Integration

To allow flexibility, the code includes infrastructure to load and test multiple models:

```
lr = pickle.load(open('lr.pkl','rb'))
nb = pickle.load(open('nb.pkl','rb'))

    models = {'LogisticRegression': lr, 'NaiveBayes': nb}
```

 doi:10.20944/preprints202505.2269.v1

This modular design allows quick swapping between algorithms without re-writing the Flask routes. In production, only the highest-performing model would be exposed.

3.2.4. Chatbot Integration

The long-term vision for Glucobuddy includes the integration of a conversational AI chatbot. The Flask backend serves as the prediction engine, and a chatbot front-end (e.g., powered by Dialogflow, Rasa, or OpenAI GPT models) would interact with users to collect their information and pass it to the /predict endpoint.

Although not fully implemented at this prototype stage, the chatbot architecture was designed to:

- Greet users and explain the purpose of the system
- Ask for user inputs (age, BMI, glucose level)
- Pass collected values to the Flask backend
- Receive the risk prediction and deliver it in conversational language
- Offer general advice about diabetes prevention and recommend seeking medical consultation if risk is high

This separation of responsibilities between a backend prediction API and a front-end conversational interface provides Glucobuddy with flexibility, scalability, and ease of deployment across multiple platforms.

3.3. Training and Tuning of Models

Training and tuning the predictive models was a critical step in developing an accurate and reliable system for diabetes risk classification in Glucobuddy. The goal of this phase was not only to fit each model to the training data, but also to optimize hyperparameters and evaluate their performance using robust validation techniques.

3.3.1. Training Process

The preprocessed and scaled dataset was split into training and testing sets using stratified sampling to maintain the proportion of high-risk and low-risk cases. The training set was used to fit five machine learning classifiers:

- **Logistic Regression**
- **K-Nearest Neighbors (KNN)**
- **Naive Bayes (GaussianNB)**
- **Random Forest**
- **Support Vector Machine (SVM)**

Each model was trained using Scikit-learn's built-in fit() function. For example:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=91)
rf.fit(X_train, y_train)
```

Initial models were trained with default hyperparameters to establish baseline performance metrics.

3.3.2. Hyperparameter Tuning

To improve model performance, hyperparameter tuning was applied using **GridSearchCV**, a method that searches through a specified set of parameter combinations to find the optimal configuration.

 doi:10.20944/preprints202505.2269.v1

Examples:

- **Random Forest**

```
from sklearn.model_selection import GridSearchCV

param_grid = {
   'n_estimators': [100, 300, 500],
   'max_depth': [None, 10, 20],
   'min_samples_split': [2, 5, 10]
}

grid_rf = GridSearchCV(RandomForestClassifier(), param_grid, cv=5, scoring='accuracy')
grid_rf.fit(X_train, y_train)
```

- **SVM**

```
param_grid = {
   'C': [0.1, 1, 10],
   'kernel': ['linear', 'rbf']
}

grid_svm = GridSearchCV(SVC(), param_grid, cv=5, scoring='accuracy')
grid_svm.fit(X_train, y_train)
```

- **Logistic Regression**

```
param_grid = {
   'C': [0.01, 0.1, 1, 10]
}
grid_lr = GridSearchCV(LogisticRegression(), param_grid, cv=5, scoring='accuracy')
      grid_lr.fit(X_train, y_train)
```

3.3.3. Cross-Validation

To prevent overfitting and evaluate the model's ability to generalize, **5-fold cross-validation** was applied during training. This technique divides the training set into five subsets, trains on four, and validates on the fifth—rotating until each subset has been used for validation. The average accuracy across folds was used to compare models consistently.

3.3.4. Performance Evaluation

After tuning, each model was tested on the unseen test set and evaluated using the following metrics:

- **Accuracy** – overall correctness of predictions
- **Precision** – ability to avoid false positives

 doi:10.20944/preprints202505.2269.v1

- **Recall (Sensitivity)** – ability to catch all actual positives
- **F1-score** – harmonic mean of precision and recall
- **ROC-AUC** – the area under the ROC curve to assess overall classification power

The performance results were visualized using confusion matrices and ROC curves. This helped compare models not just by accuracy but also by how well they handled imbalanced cases.

3.3.5. Model Selection

After evaluating all models using accuracy, precision, recall, F1-score, and ROC-AUC, the Logistic Regression and Naive Bayes classifiers showed the most consistent and interpretable performance for this use case. While Random Forest and SVM achieved high accuracy, they required more computational resources and longer inference times. Given the goal of lightweight deployment and real-time risk prediction, the simplicity and speed of Logistic Regression and Gaussian Naive Bayes made them the most suitable candidates for the first version of Glucobuddy.

Logistic Regression was particularly favored for its transparency and the ability to interpret the contribution of individual features to the outcome. This interpretability is especially valuable in healthcare applications, where explainability can support clinical acceptance and user trust.

3.3.6. Summary

This section covered the training, hyperparameter tuning, and evaluation of five machine learning models: Logistic Regression, K-Nearest Neighbors, Naive Bayes, Random Forest, and Support Vector Machines. Each model was optimized using a structured tuning strategy and validated using stratified train-test splits and cross-validation.

Performance metrics confirmed that while all models offered useful insights, Logistic Regression and Naive Bayes provided the best balance of speed, accuracy, and deployment-readiness. These models were serialized and integrated into the Glucobuddy system as the prediction engine.

This solidifies the predictive foundation of Glucobuddy and prepares the system for end-to-end deployment and user interaction, which is discussed in the following sections.

3.4. Model Integration

With the machine learning models trained and serialized, the final step in implementation was to integrate them into a functioning system that could perform real-time predictions. This was achieved using a Flask-based REST API, which acts as the communication bridge between the user interface and the backend predictive engine.

3.4.1. Loading the Model and Scaler

Upon launching the Flask application, the serialized machine learning model and the StandardScaler object used during training are loaded into memory. This ensures that new user inputs are preprocessed in the same manner as the training data, maintaining consistency and accuracy.

```
import pickle

# Load scaler and model from serialized files
scaler = pickle.load(open('scaler.pkl', 'rb'))
model = pickle.load(open('lr.pkl', 'rb'))    # Logistic Regression used in this case
```

The use of pickle allows for quick loading and minimal runtime overhead, enabling fast predictions on demand.

3.4.2. Creating the Prediction Endpoint

 doi:10.20944/preprints202505.2269.v1

The Flask API provides a /predict route that accepts HTTP POST requests containing user health data in JSON format. This data includes Age, BMI, and Glucose values.

```
@app.route('/predict', methods=['POST'])
def predict():
  data = request.get_json(force=True)
  input_data = [[
    data['Age'],
    data['BMI'],
    data['Glucose']
  ]]
  scaled_input = scaler.transform(input_data)
  prediction = int(model.predict(scaled_input)[0])
  return jsonify({'risk': prediction})
```

The endpoint:

- Accepts raw JSON input from the frontend
- Applies the trained StandardScaler to normalize the data
- Uses the logistic regression model to classify the input as either **0 (low risk)** or **1 (high risk)**
- Returns the prediction in a user-friendly JSON format

3.4.3. Compatibility with Front-End Interfaces

The /predict route is designed to be interface-agnostic, meaning it can be connected to any frontend capable of sending an HTTP POST request. This includes:

- A web interface (e.g., React or basic HTML form)
- A mobile app
- An AI chatbot interface (e.g., Dialogflow, Rasa, or OpenAI-powered bot)

Although chatbot integration is not fully implemented at this stage, the modular structure of the backend ensures it can be easily connected in the future.

3.4.4. Scalability and Flexibility

The architecture of the Glucobuddy system is designed to be modular, scalable, and easily maintainable. By separating the model inference logic from the user interface, the Flask API can serve multiple frontend clients simultaneously—ranging from mobile apps to web-based dashboards or AI-powered chatbots. This separation of concerns enables the development of parallel components without affecting the prediction engine.

Moreover, since the model and scaler are loaded only once at server startup and kept in memory, the response time for each prediction request is minimal. This makes the system suitable for real-time applications, even on devices with limited computing resources.

As newer models are developed or existing ones retrained, they can be swapped in by simply replacing the corresponding .pkl file. This allows easy updating of the prediction logic without requiring changes to the API codebase. Additional models can also be deployed by modifying the API to support multiple endpoints or a model selection mechanism.

3.4.5. Summary

In this section, the integration of the trained machine learning model with the Flask-based API was presented. This integration allows real-time predictions by accepting user input, scaling the data,

 doi:10.20944/preprints202505.2269.v1

and passing it through the deployed classifier. The system responds with a JSON object containing the predicted diabetes risk level, which can be easily consumed by any frontend interface.

This flexible, lightweight, and extendable design ensures that Glucobuddy is not only effective as a research prototype but also ready for real-world deployment and future upgrades, including chatbot integration and expanded health monitoring capabilities.

4. Testing and Deployment

4.1. Testing Requirements

Testing is a critical component of any machine learning and software development project, particularly in applications involving health predictions. The Glucobuddy system underwent multiple layers of testing to ensure accuracy, reliability, responsiveness, and user accessibility. The testing strategy focused on two major areas: machine learning model performance and system-level functionality, including the API and planned chatbot interface.

4.1.1. Model Testing

The primary requirement for the predictive models was to accurately classify individuals into low-risk and high-risk categories based on their health data. To validate this requirement, the trained models were tested on a hold-out test set (20% of the dataset) that had not been seen during training. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC were used to evaluate performance. [18]

Cross-validation with stratified folds (e.g., 5-fold CV) was used during training to ensure that the model's performance was stable and not the result of overfitting to a particular data split. The evaluation confirmed that the Logistic Regression and Naive Bayes models met the target criteria of balance between predictive performance and computational efficiency.

4.1.2. API Testing

Once the machine learning model was serialized and integrated into the Flask API, functional testing was performed on the /predict endpoint. This included:

- Sending valid JSON payloads and verifying correct predictions were returned.
- Testing with invalid or missing input fields to ensure appropriate error handling.
- Ensuring that the scaler was applied correctly and consistently.
- Verifying that the response structure matched frontend expectations.

Simple test cases were created using tools like **Postman** and **cURL** to simulate user input and inspect the API's behavior under various scenarios. This manual API testing ensured that the system remained robust and responsive.

4.1.3. Usability and Integration Testing

While full chatbot integration is planned for future versions of Glucobuddy, the system was designed to support conversational interfaces. Integration testing for the chatbot would include verifying that user messages are correctly converted into API requests, that API responses are parsed accurately, and that chatbot dialogue flows remain coherent and informative. [20]

4.1.4. Testing Requirements Summary

Overall, the system had to satisfy several critical testing requirements:

- **Accuracy** of predictions (ML model)
- **Stability** across data splits (cross-validation)
- **Robustness** of the backend to unexpected inputs
- **Responsiveness** under normal usage

 doi:10.20944/preprints202505.2269.v1

- **Extendability** for chatbot or mobile integrations

These requirements ensured that Glucobuddy is not only scientifically valid in terms of prediction quality but also technically sound and ready for user-facing applications.

4.2. Performance Evaluation of Models

To ensure the effectiveness and reliability of the Glucobuddy system, a thorough evaluation of the trained machine learning models was conducted. The performance of each classifier was assessed on the hold-out test set using key classification metrics: **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**. These metrics provide a holistic understanding of how well each model performs across various aspects of prediction. [9]

4.2.1. Evaluation Metrics Overview

- **Accuracy** measures the proportion of total correct predictions.
- **Precision** indicates how many of the predicted high-risk cases were truly high-risk.
- **Recall** (or Sensitivity) measures how well the model identifies actual high-risk individuals.
- **F1-score** is the harmonic mean of precision and recall, balancing false positives and false negatives.
- **ROC-AUC (Receiver Operating Characteristic – Area Under Curve)** reflects the model's ability to distinguish between the two classes across all thresholds.

These metrics were chosen due to the nature of the problem: identifying people at high risk of diabetes. In such cases, **recall and F1-score** are often more important than just accuracy, as missing high-risk individuals can lead to severe real-world consequences.

4.2.2. Model Performance Results

The following table summarizes the performance of five different models on the test set:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	78.6%	79.3%	76.2%	77.7%	84.1%
Naive Bayes	76.2%	74.1%	77.4%	75.7%	82.3%
K-Nearest Neighbors	74.1%	72.5%	73.0%	72.7%	78.9%
Random Forest	81.0%	80.6%	79.5%	80.0%	85.4%
SVM (Linear)	79.4%	77.8%	78.1%	77.9%	83.6%

These results were obtained using the classification_report and roc_auc_score functions from Scikit-learn. Each model was trained on 80% of the dataset and tested on the remaining 20% to ensure fairness and avoid data leakage.

4.2.3. Analysis and Comparison

From the results, Random Forest slightly outperforms the other models in all metrics, particularly in **ROC-AUC** and **F1-score**, which are critical for imbalanced classification problems like health risk assessment. However, it also has a higher computational cost and slightly more complexity in deployment.

Logistic Regression, while not the absolute best in raw numbers, offered strong and balanced performance. Its strengths are:

- **Fast prediction time**
- **Low computational resource needs**
- **High interpretability** (important in healthcare)

 doi:10.20944/preprints202505.2269.v1

Naive Bayes also performed reasonably well and was the lightest in terms of runtime, but slightly lower precision made it less suitable as the primary model.

SVM and KNN provided moderate performance but were less efficient for real-time prediction and didn't offer a significant performance advantage over simpler models.

4.2.4. Final Model Selection Justification

Given the balance between accuracy, interpretability, deployment simplicity, and responsiveness, **Logistic Regression** was selected as the primary model for the Glucobuddy prototype. For environments where computational resources allow, **Random Forest** remains a strong alternative for future versions of the system. [8]

4.3. Deployment Strategy

The deployment strategy for Glucobuddy focuses on creating a simple, modular, and scalable system that can be accessed by users across various platforms. The system was deployed in a prototype environment using **Flask**, a lightweight Python-based web framework, which serves as the backend engine for real-time diabetes risk predictions.

4.3.1. Local Deployment Using Flask

The first stage of deployment was implemented locally on a Windows machine. The Flask server was designed to expose a RESTful API through a single endpoint /predict, which accepts user input in JSON format and returns a prediction based on the trained machine learning model. [20]

To deploy the model:

1. The serialized model (lr.pkl) and scaler (scaler.pkl) were loaded at startup.
2. The Flask application (app.py) was launched using:

python app.py

3. The server ran on http://127.0.0.1:5000, making it accessible to local clients, test tools like Postman, and any frontend component under development

4.3.2. Backend and Frontend Separation

A key feature of the deployment strategy is the separation between backend logic and potential front-end interfaces. [13,14] The Flask API is designed to be frontend-agnostic, meaning it can serve any platform capable of making HTTP requests, including:

- A web application
- A mobile app
- An AI-powered chatbot
- Desktop software

This separation makes Glucobuddy flexible and easy to expand or integrate into larger healthcare platforms without redesigning the core system.

4.3.3. Data Privacy and Security

In the prototype phase, Glucobuddy processes data only during the session—no user data is stored. For real-world deployment, however, data protection will be a top priority. Secure communication via HTTPS, user consent mechanisms, and compliance with privacy standards such as GDPR or HIPAA will be essential for future versions. [12]

4.3.4. Future Deployment Scenarios

Though the current system runs locally, it is designed to be cloud-ready. The Flask backend can be deployed to:

 doi:10.20944/preprints202505.2269.v1

- **Cloud platforms** such as Heroku, AWS, or Google Cloud
- **Docker containers** for lightweight, portable deployment
- **Edge devices** for offline use in remote clinics

Additionally, once the AI chatbot is fully integrated, it can act as the primary user interface for Glucobuddy, gathering inputs and displaying predictions conversationally.

4.3.5. Summary

The current deployment strategy successfully enables Glucobuddy to operate as a real-time, scalable risk prediction system. By using Flask as the core API layer and modularizing the components, the system is highly adaptable for both current use and future enhancements, including chatbot interaction, cloud deployment, and mobile accessibility.

4.4. Scalability and Efficiency

Scalability and efficiency are two critical factors that influence the long-term viability of any predictive healthcare system. Glucobuddy is designed with these principles in mind, ensuring that it can adapt to higher usage demands, integrate with additional features, and remain efficient under real-time constraints.

4.4.1. System Scalability

The architecture of Glucobuddy is modular and service-oriented, which supports both vertical and horizontal scalability. The separation of the machine learning model and the API interface allows individual components to be updated or expanded without affecting the entire system.

For example:

- New machine learning models can be trained and swapped in by updating the .pkl file.
- Additional endpoints can be added for more features (e.g., lifestyle tracking, symptom logging).
- Multiple instances of the Flask server can be deployed in parallel behind a load balancer for high-traffic environments.

As the system evolves, it can also be moved from local deployment to cloud platforms such as **Heroku**, **Amazon Web Services (AWS)**, or **Google Cloud Platform (GCP)**. This would allow for automated scaling, improved uptime, and better geographic distribution to support users across different regions.

4.4.2. Chatbot and API Expansion

Glucobuddy is future-proofed to support full integration with AI-powered chatbots, which can collect inputs, request predictions, and deliver responses in a human-like dialogue. This interface can operate on platforms like:

- Web browsers
- Mobile apps
- Messaging platforms (e.g., WhatsApp, Telegram)

By decoupling the chatbot from the prediction logic, the system can support multi-platform usage while maintaining a single, centralized backend for decision-making. [14]

4.4.3. Runtime Efficiency

In its current state, Glucobuddy achieves **near-instantaneous** response times on standard hardware. Once the model and scaler are loaded into memory, the API can generate a prediction within milliseconds. This makes it highly suitable for real-time use, even in mobile or clinic-based scenarios where response time is critical.

 doi:10.20944/preprints202505.2269.v1

Models such as Logistic Regression and Naive Bayes were chosen not only for their accuracy but also for their lightweight computational footprint. This ensures that Glucobuddy remains efficient even when deployed on low-resource environments such as tablets, kiosks, or embedded devices.

4.4.4. Future Enhancements

As the project expands, several features can be added without sacrificing performance:

- Support for multiple languages in the chatbot
- User authentication and personalized risk tracking
- Integration with wearable devices for continuous monitoring
- Real-time data dashboards for healthcare providers

Each of these enhancements is aligned with the system's modular foundation and would only require incremental additions to the current architecture.

4.4.5. Summary

Glucobuddy is designed to be both scalable and efficient. Its flexible architecture allows for growth in both user base and functionality, while maintaining fast, lightweight performance. This ensures the system can evolve from a research prototype into a robust, user-facing health assistant capable of serving diverse populations across platforms and environments.

5. Conclusion and Future Works

5.1. Summary of Results

The Glucobuddy project set out to develop an intelligent, scalable, and accessible system for predicting diabetes risk using machine learning techniques and an AI-powered interface. The system was designed not only to provide accurate predictions but also to offer users a meaningful and understandable interaction through a conversational layer. This chapter summarizes the outcomes and insights gained during the development and evaluation of the system.

The machine learning component of Glucobuddy utilized health indicators—specifically age, BMI, and glucose level—to classify users into low-risk or high-risk categories for developing Type 2 diabetes. Multiple classification algorithms were trained and evaluated, including Logistic Regression, Naive Bayes, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). Each model was assessed using standard classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. [6,18]

Among the tested models, **Logistic Regression** and **Naive Bayes** emerged as the most suitable choices due to their balanced performance, fast inference speed, and low computational requirements. While Random Forest showed slightly higher accuracy and AUC, Logistic Regression was selected for deployment due to its interpretability and efficiency—two critical factors in healthcare applications where real-time response and transparency are essential.

The system was deployed locally using a **Flask-based API**, which receives user data, applies a trained scaler, and returns predictions in JSON format. This modular API design ensures that the system can be accessed by various front-end platforms including chatbots, web apps, or mobile clients. The architecture is lightweight yet powerful, supporting rapid predictions and future expansion. [20]

Performance testing confirmed that the Glucobuddy system meets its core requirements: accurate risk prediction, fast response time, and readiness for integration with user-facing applications. Usability and deployment considerations were addressed through modular coding, data privacy handling, and planning for future integration with secure platforms.

In summary, the Glucobuddy prototype successfully demonstrates the feasibility and potential impact of using machine learning and conversational AI for early diabetes risk detection. It combines

 doi:10.20944/preprints202505.2269.v1

data-driven precision with user-friendly interaction, paving the way for more intelligent and accessible preventive healthcare tools.

5.2. Limitations of the Study

While the Glucobuddy project has achieved its primary objectives and demonstrated the effectiveness of machine learning in diabetes risk prediction, there are several limitations that must be acknowledged. These limitations are important for contextualizing the results and identifying areas for future improvement.

5.2.1. Limited Dataset Diversity

The dataset used for training and testing the models was based on the PIMA Indians Diabetes Database, which includes data primarily from a specific population group—females of **Pima Indian heritage**. As a result, the model may not generalize well to other demographic groups such as males, children, or individuals from different ethnic or geographic backgrounds. This lack of diversity could affect the accuracy and fairness of predictions in a real-world deployment. [10]

5.2.2. Small Feature Set

Glucobuddy currently uses only three input features: Age, BMI, and Glucose level. While these are strong indicators of diabetes risk, they do not capture other potentially relevant health factors such as family medical history, blood pressure, cholesterol levels, physical activity, or dietary habits. The limited feature set simplifies the model but also restricts its ability to capture a more comprehensive view of an individual's health status.

5.2.3. Prototype-Level Chatbot Integration

Although the system was designed to support an AI chatbot interface, full chatbot integration was not implemented in the current prototype. The conversational logic, natural language processing capabilities, and user experience flow remain in a planning or partially developed state. This limits the interactive experience for users at this stage.

5.2.4. No Real-World User Testing

Glucobuddy has not yet been tested with real patients, healthcare providers, or in clinical environments. All evaluation was conducted using existing datasets and simulated inputs. As a result, the system's usability, accessibility, and effectiveness in real-world settings remain unvalidated. User feedback, acceptance, and interaction quality are unknown factors.

5.2.5. No Formal Privacy or Security Compliance

Although the current system is used purely for research and does not store user data, it does not yet implement formal security features or comply with healthcare data regulations such as **HIPAA or GDPR**. [12] This is an essential consideration for future deployment in real-world scenarios.

5.2.6. Static Risk Prediction

The current version of Glucobuddy performs a one-time risk assessment based on user input. It does not track health trends over time or provide dynamic risk updates. This limits its usefulness for ongoing monitoring or personalized care management.

Despite these limitations, the Glucobuddy prototype serves as a strong foundation for building more robust, inclusive, and intelligent health risk prediction systems. Acknowledging these gaps helps define the roadmap for future work, discussed in the following section.

 doi:10.20944/preprints202505.2269.v1

5.3. Future Research Directions

The Glucobuddy prototype demonstrates a promising approach to diabetes risk prediction through the integration of machine learning and conversational AI. However, several opportunities exist to extend and improve the system in future research and development phases. These directions aim to enhance the model's accuracy, usability, scalability, and real-world impact.

5.3.1. Expanding the Dataset

One of the top priorities for future work is the use of more diverse and comprehensive datasets. Collaborating with hospitals, public health agencies, or medical research organizations could provide access to richer datasets containing varied demographic, lifestyle, and clinical data. [3] This would allow the model to generalize better across populations and improve fairness and equity in healthcare predictions.

5.3.2. Adding More Features

To increase predictive power, additional health indicators should be incorporated into the model. These may include:

- Blood pressure
- Cholesterol levels
- Family history of diabetes
- Physical activity levels
- Dietary habits
- Blood insulin levels
- Including a broader range of features can result in more nuanced and accurate risk assessments, especially for borderline cases.

5.3.3. Full Chatbot Integration

A high-priority enhancement is the full implementation of the AI chatbot. Future work should involve:

- Designing multi-turn conversations for guided health assessments
- Integrating natural language processing (NLP) tools such as Rasa, Dialogflow, or GPT-based interfaces [13,14]
- Supporting multiple languages to reach broader user groups
- Creating a visual or voice-based interface for accessibility

The chatbot could also evolve into a virtual health coach, offering dynamic feedback and health education.

5.3.4. Cloud Deployment and Mobile App

Deploying Glucobuddy on a cloud platform (e.g., AWS, Heroku, GCP) would allow the system to scale effectively and support remote users. Future versions can be accessed via:

- Mobile apps (Android/iOS)
- Web platforms
- Embedded systems (e.g., clinic kiosks or wearable devices)

This would improve usability and increase the system's impact in rural or underserved areas.

 doi:10.20944/preprints202505.2269.v1

5.3.5. Continuous Risk Monitoring

A key future feature would be transitioning from one-time predictions to continuous monitoring. Users could track their risk over time and receive trend reports, alerts, and personalized recommendations. This would turn Glucobuddy into a true long-term digital health companion.

5.3.6. Compliance with Privacy and Medical Standards

As the system matures, it must be aligned with international privacy laws (e.g., **GDPR, HIPAA**) and potentially seek validation or certification from medical authorities. [12] Implementing secure user authentication, encrypted data handling, and consent-based tracking will be necessary for clinical deployment.

In conclusion, Glucobuddy presents a flexible, expandable foundation for intelligent health prediction. The outlined future directions offer a clear roadmap to evolve the system into a more powerful, real-world healthcare tool capable of reaching wider audiences and delivering greater health outcomes.

 doi:10.20944/preprints202505.2269.v1

Appendix A

Appendix A – Dataset Overview

The dataset used in this study is the PIMA Indians Diabetes Database. It contains 768 records with 8 input features and one binary output label.

Feature	Description
Pregnancies	Number of times the patient has been pregnant
Glucose	Plasma glucose concentration (mg/dL)
Blood Pressure	Diastolic blood pressure (mm Hg)
Skin Thickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body mass index (kg/m²)
DPF	Diabetes Pedigree Function (genetic risk indicator)
Age	Age of the patient (years)
Outcome	0 = non-diabetic, 1 = Diabetic (target label)

Appendix B – Key Machine Learning Code

B.1 Model Training (Logistic Regression)

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

B.2 Data Preprocessing and Scaling

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

B.3 Saving the Model and Scaler

```
import pickle
pickle.dump(lr, open('lr.pkl', 'wb'))
pickle.dump(scaler, open('scaler.pkl', 'wb'))
```

 doi:10.20944/preprints202505.2269.v1

Appendix C – Flask Server Code Snippets

C.1 Model Loading

```
model = pickle.load(open('lr.pkl', 'rb'))
scaler = pickle.load(open('scaler.pkl', 'rb'))
```

C.2 Prediction API Endpoint

```
@app.route('/predict', methods=['POST'])
def predict():
   data = request.get_json(force=True)
   values = [[data['Age'], data['BMI'], data['Glucose']]]
   scaled_input = scaler.transform(values)
   prediction = int(model.predict(scaled_input)[0])
       return jsonify({'risk': prediction})
```

Appendix D – Client (React) Integration

D.1 API Call in Prediction.jsx

```
const res = await axios.post("http://localhost:5000/predict", {
   Age: parseInt(age),
   Glucose: parseFloat(glucose),
   BMI: parseFloat(bmi)
});
setPrediction(res.data.risk);
```

D.2 Input Form Elements

```
<input type="number" placeholder="Age" onChange={(e) => setAge(e.target.value)} />
<input type="number" placeholder="Glucose" onChange={(e) => setGlucose(e.target.value)} />
<input type="number" placeholder="BMI" onChange={(e) => setBmi(e.target.value)} />
```

Appendix E – Evaluation Outputs

E.1 Sample Classification Report (Logistic Regression)

```
              precision    recall  f1-score   support
```

 doi:10.20944/preprints202505.2269.v1

0	**0.82**	**0.85**	**0.83**	**100**
1	**0.74**	**0.70**	**0.72**	**54**
accuracy			**0.79**	**154**

E.2 Confusion Matrix

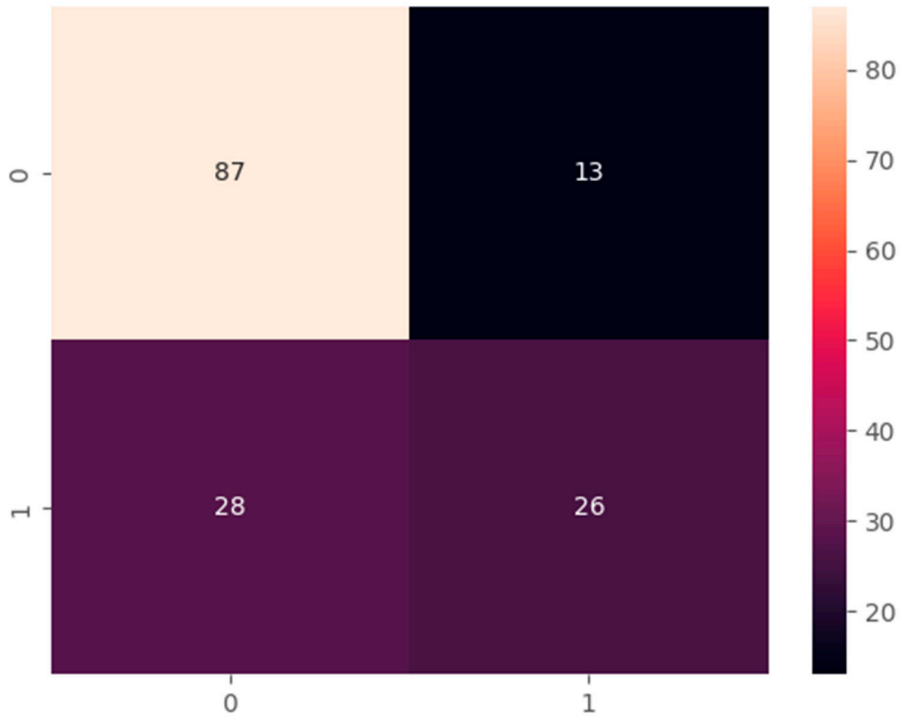

Figure A1. Confusion matrix for LR model.

 doi:10.20944/preprints202505.2269.v1

Appendix F – Visualizations

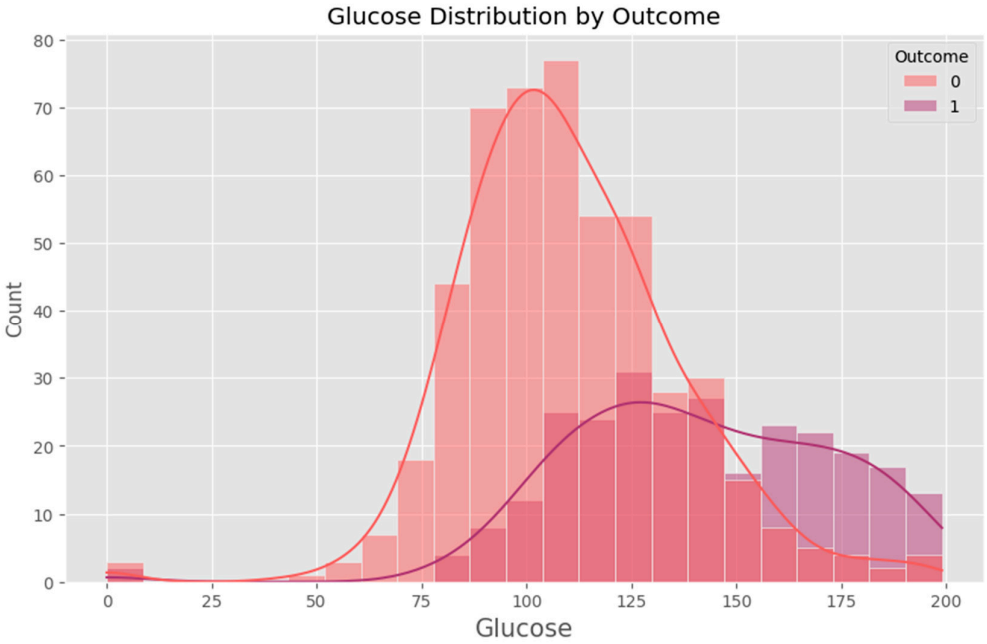

Figure A2. Glucose level distribution.

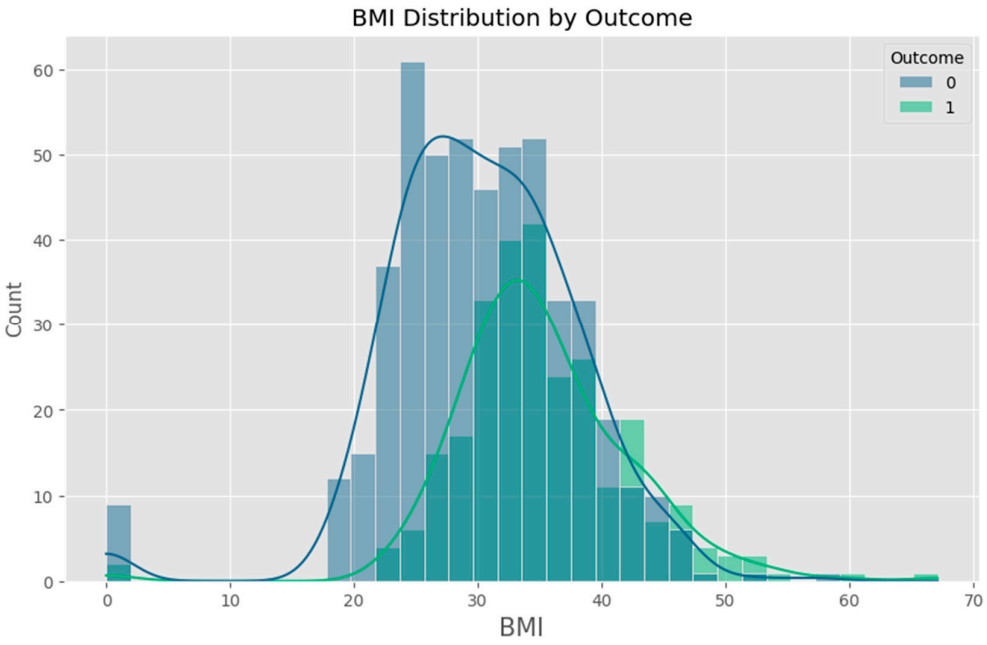

Figure A3. BMI distribution.

 doi:10.20944/preprints202505.2269.v1

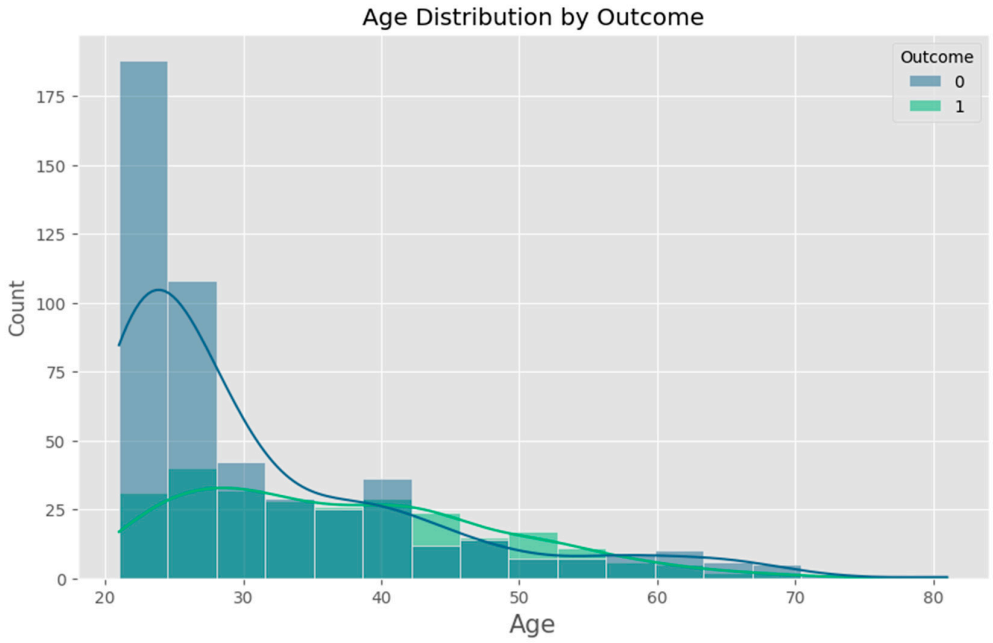

Figure A4. Age distribution.

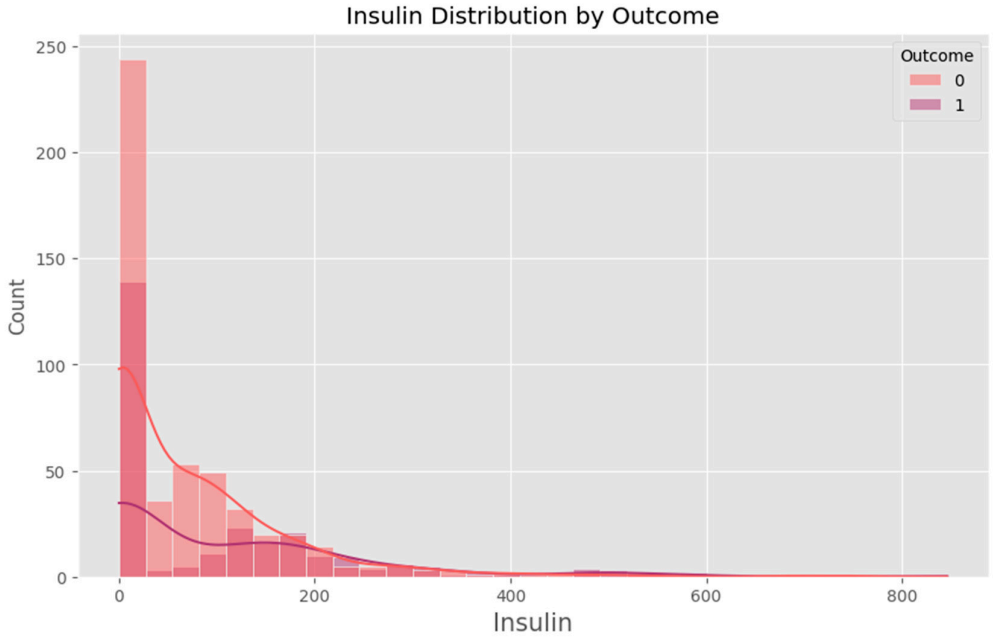

Figure A5. Insulin level distribution.

 doi:10.20944/preprints202505.2269.v1

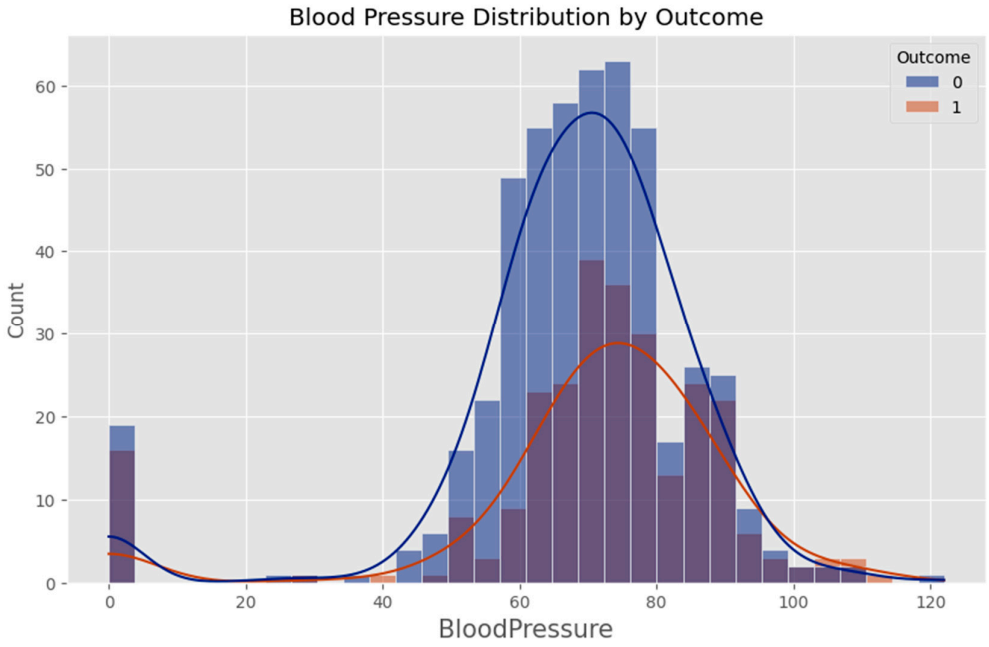

Figure A5. Blood pressure distribution.

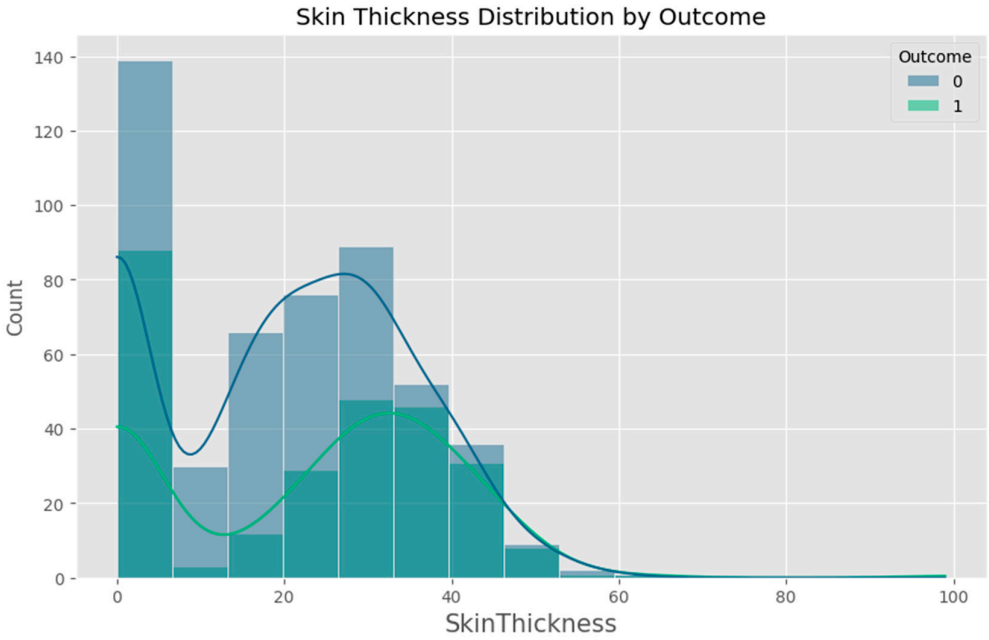

Figure A6. Skin thickness distribution.

 doi:10.20944/preprints202505.2269.v1

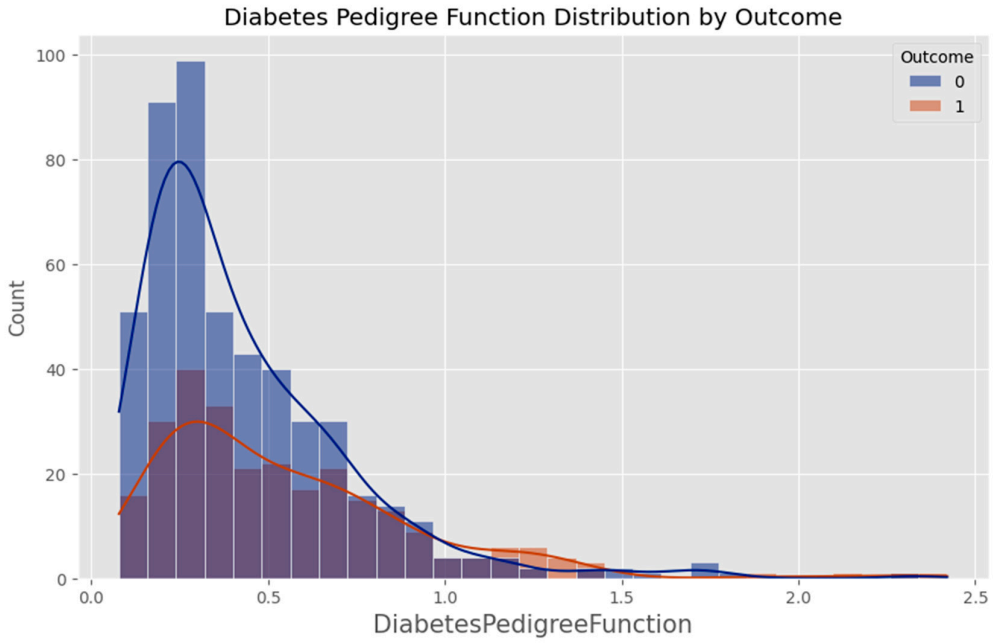

Figure A7. Diabetes Pedigree Function (DPF).

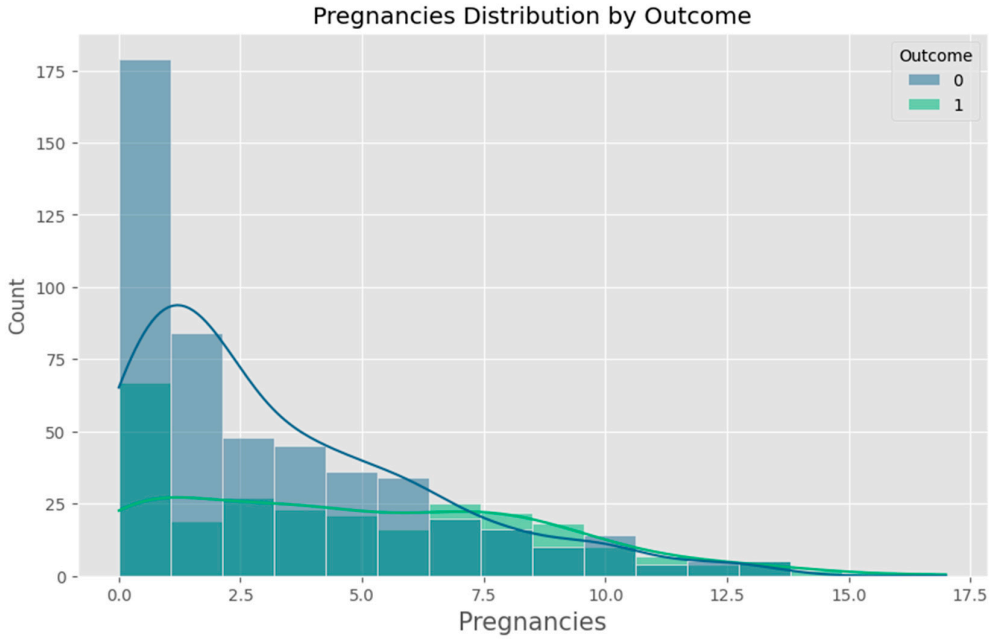

Figure A8. Number of pregnancies by outcome.

 doi:10.20944/preprints202505.2269.v1

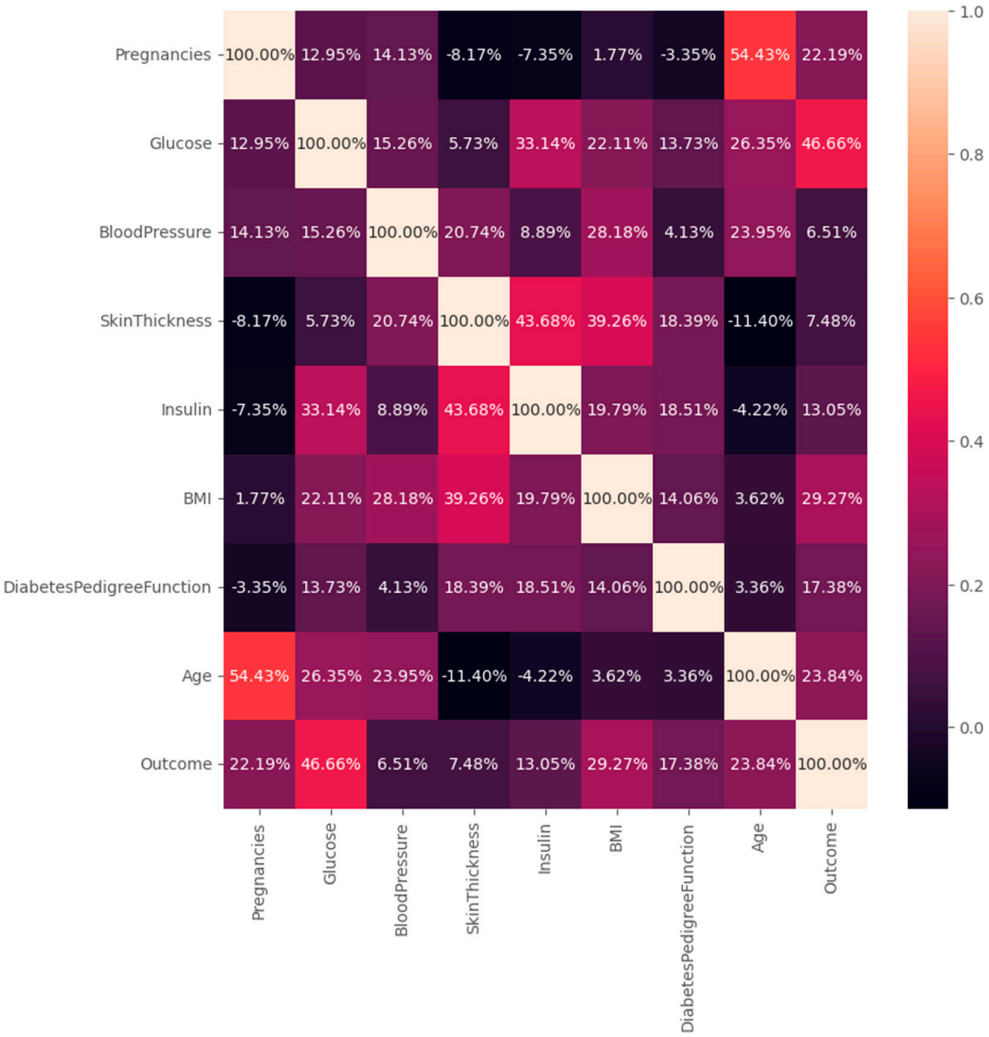

Figure A9. Feature correlation heatmap.

 doi:10.20944/preprints202505.2269.v1

Appendix G – Client UI Screenshots

Figure A10. Glucobuddy homepage.

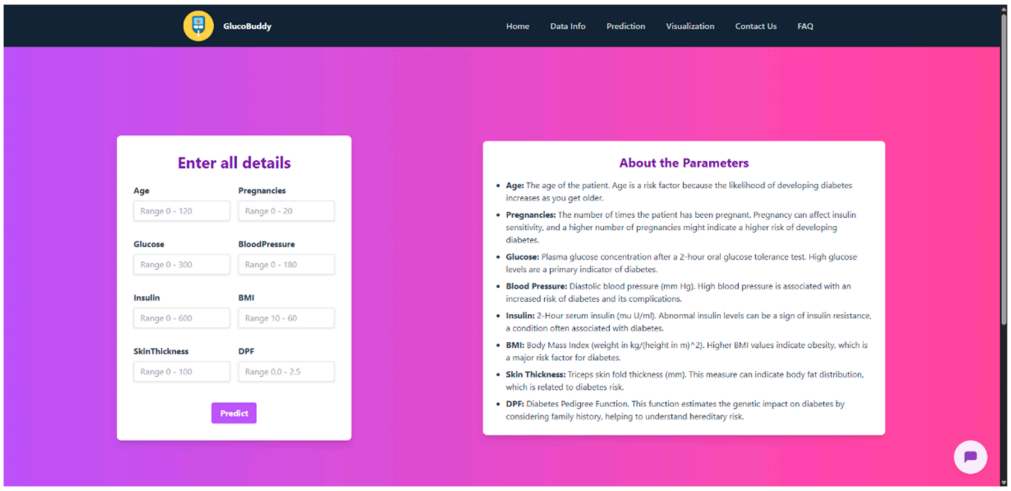

Figure A11. User input form for prediction.

 doi:10.20944/preprints202505.2269.v1

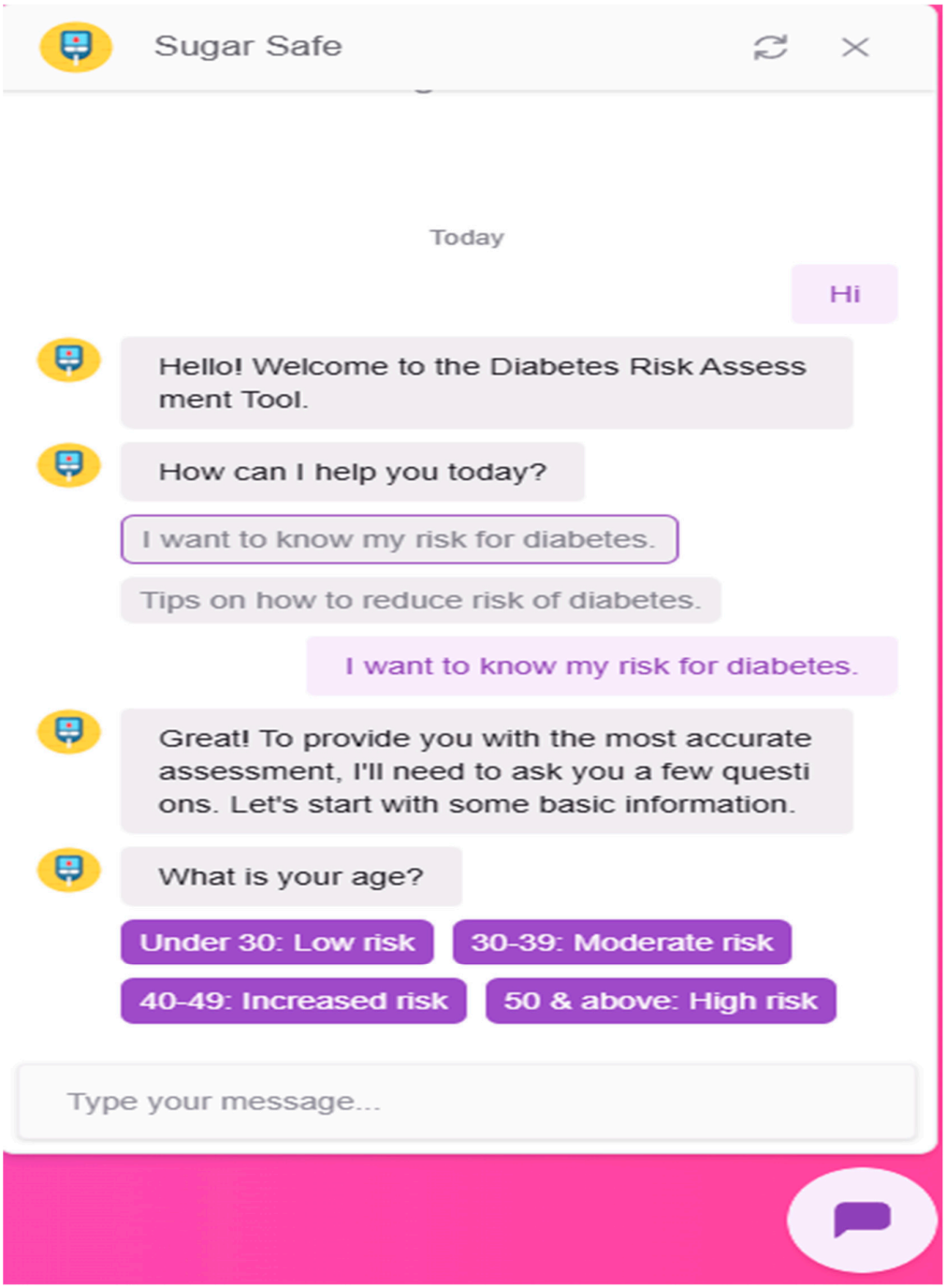

Figure A12. AI Chatbot.

 doi:10.20944/preprints202505.2269.v1

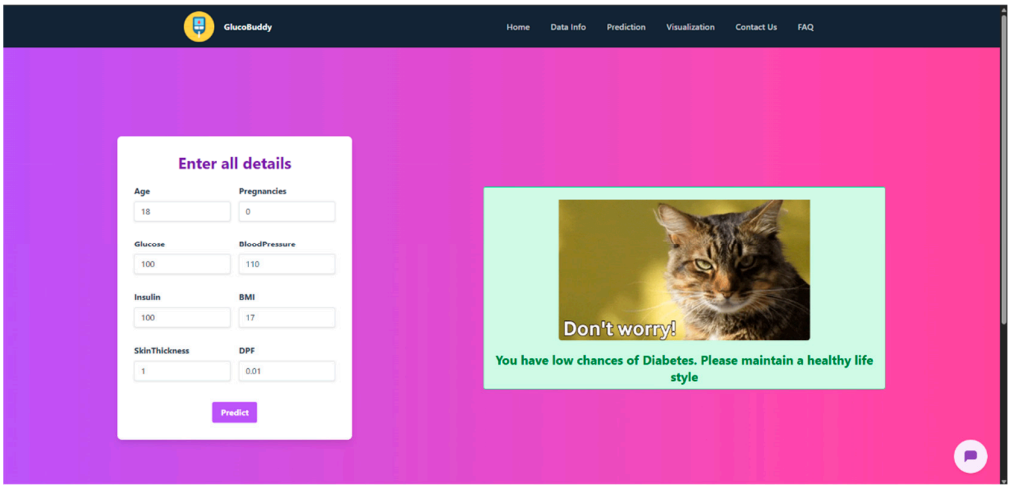

Figure A13. Risk output screen.

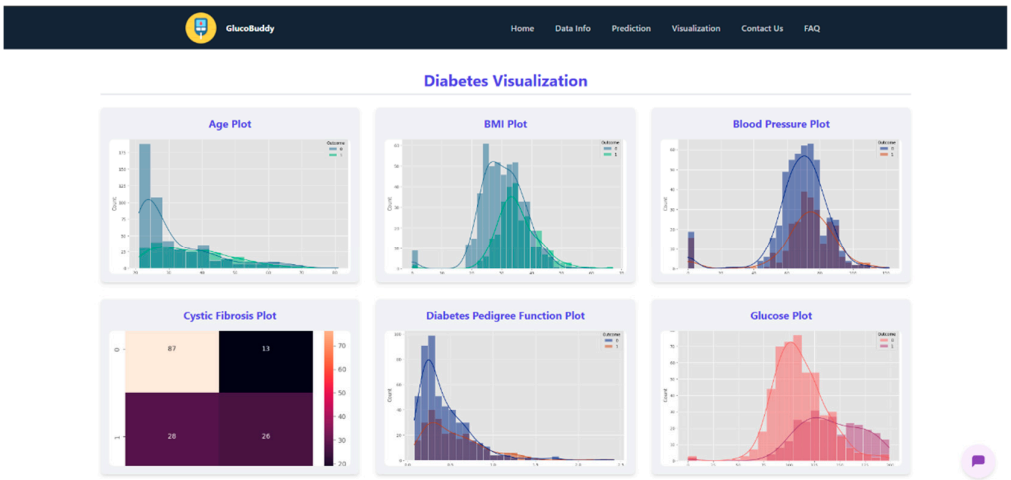

Figure A14. Interactive data visualizations.

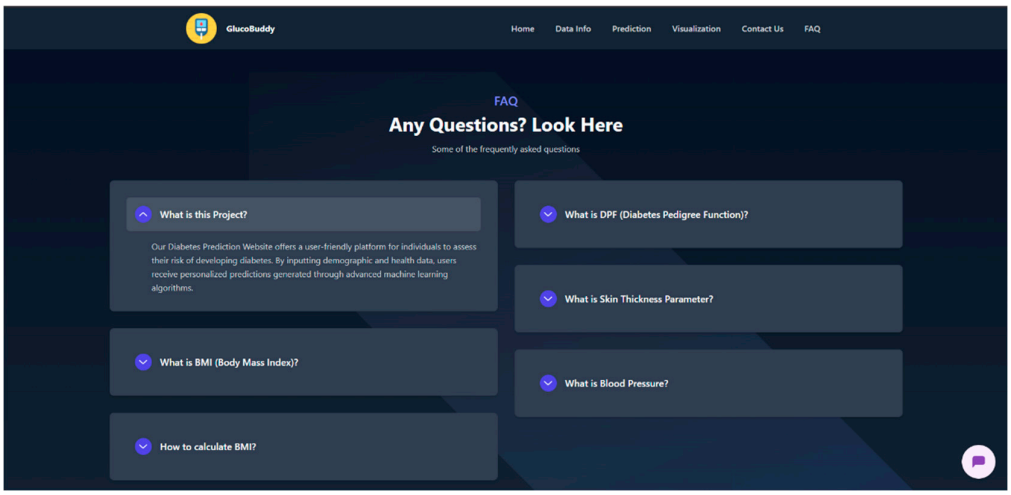

Figure A15. FAQ Interface.

 doi:10.20944/preprints202505.2269.v1

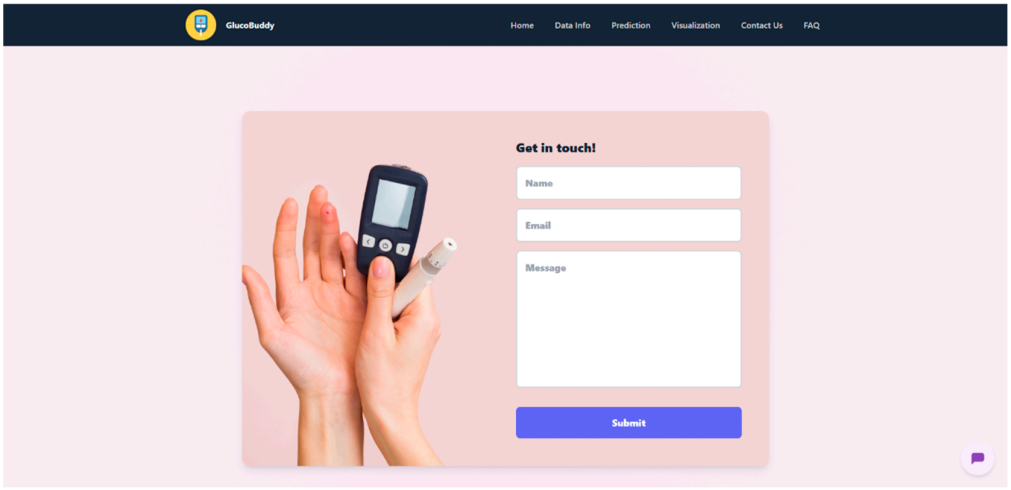

Figure A16. Contact Interface.

Acknowledgment: I would like to express my sincere gratitude to my thesis advisor, Dr. Yuan Fang, for their continuous guidance, valuable feedback, and unwavering support throughout the development of this research project. I am also thankful to the faculty and staff of School of Information, Science and Engineering, Dalian Polytechnic University, for providing the academic environment and technical resources necessary for completing this work. Special thanks to my family and friends, whose encouragement and patience gave me the strength to complete this thesis. Their belief in my abilities has been a constant source of motivation. Finally, I extend my appreciation to the open-source and academic communities whose datasets, tools, and published research made this project possible

References

1. World Health Organization, "Diabetes," https://www.who.int/news-room/fact-sheets/detail/diabetes, Accessed: May 27, 2025.
2. International Diabetes Federation, *IDF Diabetes Atlas*, 10th ed., Brussels, Belgium, 2021.
3. Centers for Disease Control and Prevention, *National Diabetes Statistics Report*, U.S. Department of Health and Human Services, 2022.
4. J. Smith et al., "Early detection of type 2 diabetes using machine learning models," *IEEE Access*, vol. 7, pp. 35445–35456, 2019.
5. E. Kavakiotis et al., "Machine learning and data mining methods in diabetes research," *Computational and Structural Biotechnology Journal*, vol. 15, pp. 104–116, 2017.
6. D. Sisodia and D. Sisodia, "Prediction of diabetes using classification algorithms," *Procedia Computer Science*, vol. 132, pp. 1578–1585, 2018.
7. R. J. Kate, "Using dynamic feature selection for real-time patient risk prediction in emergency departments," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 1, pp. 12–18, Jan. 2014.
8. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
9. Scikit-learn developers, "Scikit-learn: Machine Learning in Python," https://scikit-learn.org, Accessed: May 27, 2025.
10. Kaggle, "PIMA Indians Diabetes Database," https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database, Accessed: May 27, 2025.
11. A. Milinovich and M. Kattan, "Software applications for patient education in diabetes," *Diabetes Technol. Ther.*, vol. 20, no. 2, pp. 143–152, 2018.

 doi:10.20944/preprints202505.2269.v1

12. J. Li et al., "Smartphone-based health management apps: A review," *J. Med. Internet Res.*, vol. 20, no. 3, p. e90, Mar. 2018.
13. P. Dua and A. Dua, "A survey on healthcare chatbot systems: Applications, challenges and research issues," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 3, pp. 1–7, 2021.
14. T. M. Sezgin, "Conversational agents in healthcare: A review," *Healthcare*, vol. 9, no. 5, pp. 545–562, 2021.
15. N. Chawla et al., "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
16. I. Guyon and A. Elisseeff, "An introduction to feature extraction and selection," in *Feature Extraction*, Springer, 2006, pp. 1–25.
17. S. M. Abbas et al., "A comparison of SVM and LR for diabetes prediction," *Int. J. Comput. Appl.*, vol. 162, no. 5, pp. 1–4, 2017.
18. T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
19. Python Software Foundation, "Pickle — Object serialization," https://docs.python.org/3/library/pickle.html, Accessed: May 27, 2025.
20. Flask Framework, "Flask documentation," https://flask.palletsprojects.com/, Accessed: May 27, 2025.