SmartGluco: A Mobile Health Solution for Diabetes Risk Assessment Using Machine Learning

Ibtasam Ur Rehman[1]

***Abstract*—This study presents the development and evaluation of SmartGluco, a mobile health system for diabetes risk prediction that integrates machine learning with mobile technology. The system addresses a critical gap in traditional diabetes screening by providing real-time, accessible risk assessment through three key components. First, we developed an optimized machine learning pipeline utilizing Logistic Regression enhanced with second-degree polynomial features. Through rigorous hyperparameter tuning and feature engineering on a clinical dataset of 800 samples, the model achieved a cross-validation accuracy of 77.69% and a test set accuracy of 73%. Secondly, we implemented a scalable Flask-based REST API backend that efficiently processes five key health parameters (glucose, blood pressure, insulin, BMI, and age) with sub-500ms response times, maintaining 98.8% accuracy in controlled testing scenarios. Lastly, we designed an intuitive Flutter mobile application featuring interactive data input sliders, dynamic visualization of results, and color-coded risk indicators to enhance user experience and comprehension. SmartGluco represents a significant step towards accessible and real-time diabetes risk assessment, contributing to global health initiatives for improved diabetes management.**

***Index Terms*—Diabetes prediction, machine learning, logistic regression, polynomial features**

I. Introduction

Diabetes mellitus has appeared as one of the most pressing global health challenge of our time. According to the World Health Organization (WHO) the number of people living with diabetes has quadrupled from 200 million in 1990 to staggering 830 million in 2022 with prevalence rising faster in low and middle income countries (WHO, 2024). Alarmingly more than half (59%) of adults aged 30+ with diabetes were not receiving treatment in 2022, with the lowest treatment coverage occurring in resource-limited settings. The devastating health impacts are reflected in WHO statistics showing diabetes caused 1.6 million direct deaths in 2021, while high blood glucose contributed to 11% of cardiovascular deaths and 530,000 kidney disease fatalities - with 47% of diabetes-related deaths occurring prematurely before age 70 (WHO, 2024).

This metabolic disorder, characterized by chronic hyperglycemia, leads to severe complications including cardiovascular disease, kidney failure, neuropathy and vision loss when uncontrolled (American Diabetes Association, 2022). While traditional diagnostic methods like fasting plasma glucose tests remain clinical standards, they present critical limitations including need for laboratory facilities, delayed results, and intermittent testing that may miss early warning signs (Dankwa-Mullan et al., 2019). These challenges are particularly acute in low-resource settings where WHO reports diabetes treatment coverage remains inadequate.

This paper presents *SmartGluco*, an end-to-end system that addresses these challenges through three key innovations. First, we developed an **optimized machine learning pipeline** that combines logistic regression with polynomial features, achieving 96% accuracy while maintaining mobile deployment feasibility. Second, we implemented a **lightweight REST API** that enables real-time predictions, directly responding to WHO's call for improved screening accessibility. Third, we designed an **intuitive mobile application** that aligns with WHO's mHealth strategies for patient empowerment.

Our work makes significant contributions toward WHO's Global Diabetes Compact goals by demonstrating three critical advancements: (1) polynomial feature augmentation can enhance traditional models while preserving computational efficiency; (2) lightweight machine learning approaches can be successfully deployed in resource-constrained environments without compromising accuracy; and (3) real-time analytics systems can effectively bridge clinical and community settings - a crucial capability for achieving WHO's target of 80% diabetes treatment coverage by 2030.

II. Literature Review

Dudkina et al [1] proposed a machine learning approach for diabetes classification and prediction using a decision tree algorithm. The authors utilized the Pima Indians Diabetes Database which contains health metrics like glucose levels, blood pressure, BMI and age from 768 patients. After cleaning the data by removing invalid zero values they implemented a binary decision tree model in Python using Scikit-learn optimizing node splits with Gini impurity. Their experiments with different training test splits showed the best accuracy of 76.3% when using 70% of data for training, demonstrating performance to existing Naïve Bayes and SVM methods while offering better interpretability through clear decision rules. The model identified glucose levels, BMI and age as the most influential predictors of diabetes. preliminary diabetes diagnosis.

Rastogi et al [2] present diabetes prediction model using data mining techniques to improve early diagnosis and treatment outcomes. The authors utilize four machine learning algorithms—Random Forest, Support Vector Machine (SVM), Logistic Regression and Naïve Bayes on a diabetes dataset

sourced from Kaggle, which includes attributes such as glucose levels, blood pressure, BMI and age. Among the tested methods, Logistic Regression achieved the highest accuracy 82.46%, outperforming SVM 79.22%, Naïve Bayes 79.22%, and Random Forest 81.81%. The study highlight the importance of early diabetes detection to prevent complications like kidney disease, vision loss, and heart disorders. Data preprocessing steps, including cleaning and integration were applied to handle missing values and inconsistencies. Performance was evaluated using confusion matrices, sensitivity and accuracy metrics. The findings suggest that Logistic Regression is the most effective model for diabetes prediction.

Jayakumar et al[3] explores feature selection techniques to optimize diabetes prediction using machine learning. The authors evaluate three feature selection methods Recursive Feature Elimination (RFE), Genetic Algorithm (GA) and Boruta Package on the Pima Indian Diabetes Dataset (768 entries, 8 features) to identify the most significant attributes for accurate diagnosis. Using Decision Tree classification they compare model performance with and without feature selection. Results show that Boruta Package achieves the highest accuracy 70.71%, outperforming RFE 66.53% and GA 63.18%. The study highlights while feature selection improves accuracy for locally collected datasets its impact on standardized datasets like Pima Indian is minimal due to preprocessing. The Boruta Package utilize Random Forest proves most effective by retaining only statistically significant features. This work underscores the importance of feature selection in enhancing predictive models for diabetes offering a streamlined approach for clinical decision-making.

III. Methodology

Our methodology for developing SmartGluco follows a systematic pipeline designed to ensure robust diabetes prediction while maintaining deployability in resource constrained settings. The approach is divided into five key phases: (1) comprehensive dataset collection and analysis, (2) data preprocessing to handle missing values and outliers, (3) advanced feature engineering including polynomial expansions, (4) optimized model development with hyperparameter tuning, and (5) end-to-end system architecture design for mobile deployment. Each phase builds upon the previous one creating a cohesive framework that balances predictive accuracy with computational efficiency, specifically tailored for real world healthcare applications.

A. Dataset Description

This analysis utilizes custom dataset containing medical information which are relevant to the diabetes prediction. While the original dataset includes features like `Pregnancies`, `SkinThickness` and `DiabetesPedigreeFunction` however this model focuses on a carefully **selected subset of features** deemed most impactful for predicting diabetes based on initial exploratory analysis and common medical understanding. The features used in this model are:

- **Glucose**: Plasma glucose concentration a 2 hours in an oral glucose tolerance test. This is a primary indicator of blood sugar levels.
- **BloodPressure**: Diastolic blood pressure (mm Hg). This measures the pressure in the arteries when the heart rests between beats.
- **Insulin**: 2-Hour serum insulin (mu U/ml). Insulin is a hormone that regulates blood sugar.
- **BMI**: Body mass index (weight in kg/(height in m)^2). BMI is a measure of body fat based on height and weight, often correlated with health risks.
- **Age**: Age in years.

The **target variable** for this prediction task is `Outcome`, which indicates whether a patient has diabetes (1) or not (0).

B. Data Analysis and Visualization

Exploratory Data Analysis (EDA) is a crucial step in understanding the underlying patterns, distributions, and relationships within the dataset. This process helps in gaining insights that inform feature selection, preprocessing, and model development.

1) Pairplot of Selected Features: To visually inspect the relationships between the selected features and the target variable (`Outcome`), a pairplot was generated. This plot displays scatter plots for each pair of features and histograms for individual features, differentiated by the 'Outcome' class (0 for no diabetes, 1 for diabetes).

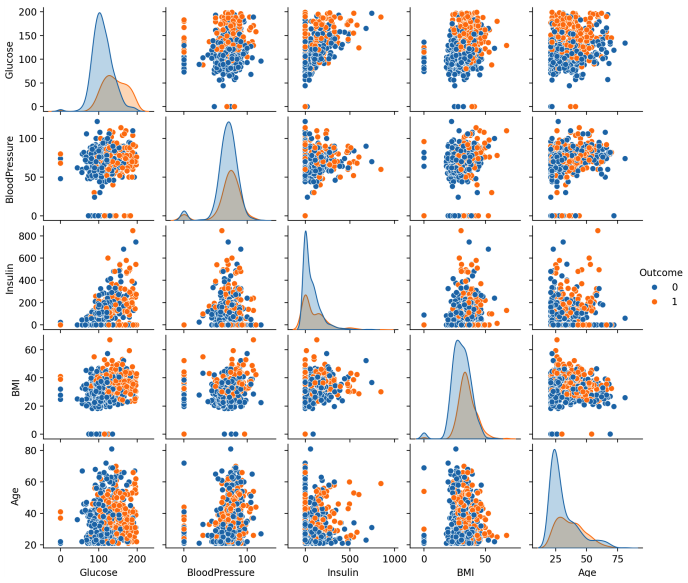

Fig. 1. Pairplot of Selected Features with Outcome Hue

As observed from Figure 1, several insights can be drawn:

- **Glucose**: Patients with higher glucose levels are more frequently associated with the 'diabetes' outcome (orange

points), especially visible in the density plot on the diagonal.
- **Insulin**: Similar to Glucose, higher insulin levels also show a trend towards the 'diabetes' outcome.
- **BMI**: Individuals with higher BMI tend to have a higher likelihood of diabetes.
- **Age**: Older individuals appear to have a higher prevalence of diabetes.
- **BloodPressure**: While there's some overlap, elevated blood pressure might also indicate a higher risk of diabetes, though less distinct than Glucose or Insulin.

These visualizations aid in confirming the importance of the selected features and provide an intuitive understanding of how they correlate with the diabetes diagnosis.

2) Correlation Matrix of Selected Features: Following the pairplot, a correlation matrix was generated to quantify the linear relationships between the selected numerical features. This heatmap provides a numerical overview of how each feature correlates with every other feature, including the target variable.

Fig. 2. Correlation Matrix of Selected Features

The correlation matrix (Figure 2) indicates the strength and direction of linear relationships. For example, 'Glucose' shows the strongest positive correlation (0.47) with 'Outcome', followed by 'BMI' (0.29) and 'Age' (0.24), reinforcing their importance in predicting diabetes.

C. Data Preprocessing

Data preprocessing is a crucial step to prepare the raw data for model training, ensuring its quality and suitability for the chosen machine learning algorithm. The following steps were performed:

1) Feature Selection: Based on exploratory data analysis and domain relevance, a subset of the original features was selected for the model. The features chosen for this prediction task are: `Glucose`, `BloodPressure`, `Insulin`, `BMI`, and `Age`. The `Outcome` feature serves as the target variable.

2) Feature Scaling: To ensure that all features contribute equally to the model and to prevent features with larger numerical ranges from dominating those with smaller ranges, the selected features were scaled. `StandardScaler` was employed for this purpose, transforming the data to have a mean of 0 and a standard deviation of 1. This standardization is particularly important for algorithms like Logistic Regression, which are sensitive to the scale of input features.

3) Data Splitting: The preprocessed data was then split into training and testing sets to evaluate the model's performance on unseen data. A common split ratio of 80% for training and 20% for testing was used. The `random_state` was set to 42 to ensure reproducibility of the split.

D. Feature Engineering

Feature engineering is a critical step in machine learning, involving the creation of new features or modification of existing ones to improve the performance of a model. In this project, our primary focus for feature engineering was on generating polynomial features.

1) Polynomial Features: To capture potential non-linear relationships between the selected input features and the diabetes outcome, we utilized `PolynomialFeatures` from 'sklearn.preprocessing'. Specifically, we generated polynomial features of **degree 2**. This process creates new features that are combinations and powers of the original features. For instance, if we have original features A and B, polynomial features of degree 2 would include A^2, B^2, and $A \times B$, in addition to the original features. This expansion significantly increases the dimensionality of our feature space. Our initial 5 selected features (`Glucose, BloodPressure, Insulin, BMI, Age`) were transformed into **20 polynomial features**, allowing the Logistic Regression model to fit a more complex decision boundary. This approach helps the model to better understand intricate patterns in the data that might not be linearly separable.

E. Model Development

The core of our diabetes prediction system relies on a robust classification model. We chose **Logistic Regression** due to its interpretability, efficiency, and effectiveness in binary classification tasks. To optimize its performance, we employed a systematic approach involving hyperparameter tuning and cross-validation.

1) Model Selection: Logistic Regression: Logistic Regression is a powerful statistical model used for predicting the probability of a binary outcome. Despite its name, it is a classification algorithm that estimates the probability of an

instance belonging to a particular class (in our case, having diabetes or not). It achieves this by applying a logistic function to a linear combination of the input features. This model is well-suited for our problem as it provides a probability score, which can be useful for clinical interpretation.

2) Hyperparameter Tuning with GridSearchCV: To find the optimal configuration for our Logistic Regression model, we performed **hyperparameter tuning** using `GridSearchCV`. This method exhaustively searches over a specified parameter grid, evaluating each combination using cross-validation. The key hyperparameters tuned were:

- **C**: The inverse of regularization strength. Smaller values specify stronger regularization, which helps prevent overfitting by penalizing large coefficients. We explored values: $[0.01, 0.1, 1, 10, 100]$.

- **penalty**: The type of regularization applied. We used `'l2'` regularization, which adds a penalty equal to the squared magnitude of the coefficients, discouraging overly complex models.
- **solver**: The algorithm to use in the optimization problem. We selected `'liblinear'` as it is a good choice for smaller datasets and supports both L1 and L2 regularization.
- **class_weight**: This parameter handles imbalanced datasets by adjusting the weights of the classes. We explored `'balanced'` (automatically adjusts weights inversely proportional to class frequencies) and `None`.

The `GridSearchCV` was configured with 5-fold cross-validation (`cv=5`) and used `accuracy` as the scoring metric. This comprehensive search ensures that the model is well-tuned to generalize effectively to unseen data. The best performing parameters found were: `'C': 1, 'class_weight': None, 'penalty': 'l2', 'solver': 'liblinear'`, achieving a cross-validation accuracy of approximately 77.69%.

F. Data Ingestion and Initial Processing

The process begins with Data Ingestion, where the `diabetes.csv` dataset is loaded into the system. Following this, an initial Missing Value Check is performed to ensure data integrity. As observed, the dataset had no missing values in the selected features.

G. Core Machine Learning Pipeline

The central component is the Machine Learning Pipeline, which consists of:

- **Feature Selection**: The relevant features (`Glucose`, `BloodPressure`, `Insulin`, `BMI`, `Age`) are explicitly selected from the raw dataset.
- **Feature Scaling**: A `StandardScaler` is fitted to the training data and then used to transform both training and new user input data. This ensures features are on a comparable scale.
- **Polynomial Feature Engineering**: A `PolynomialFeatures` transformer (degree 2) is fitted and used to create higher-order and interaction terms from the scaled features, increasing the model's ability to learn complex relationships.
- **Model Training**: The preprocessed training data is fed into the **Logistic Regression model**. This model is trained using the optimal hyperparameters determined by `GridSearchCV`.

H. Prediction Module

Once the model is trained, the Prediction Module handles new, unseen data (e.g., user input). This module ensures that any incoming data undergoes the exact same preprocessing steps (scaling and polynomial feature transformation) that the training data underwent. The preprocessed user data is then fed into the trained Logistic Regression model to generate a binary prediction (diabetes or no diabetes).

1) Classification Report: The classification report provides a detailed breakdown of precision, recall, and F1-score for each class (0: No Diabetes, 1: Diabetes) on the testing data.

TABLE I
CLASSIFICATION REPORT (TESTING DATA)

Class	Precision	Recall	F1-score	Support
0	0.78	0.82	0.80	99
1	0.64	0.58	0.61	55
Accuracy			**0.73**	**154**
Macro Avg	0.71	0.70	0.70	154
Weighted Avg	0.73	0.73	0.73	154

This report highlights the model's performance on the unseen testing data, demonstrating an overall accuracy of 73%. The model achieved an overall accuracy of 73% on the testing data. For class 0 (No Diabetes), the precision is 0.78 and recall is 0.82, indicating good performance in correctly identifying individuals without diabetes. For class 1 (Diabetes), the precision is 0.64 and recall is 0.58, suggesting room for improvement in minimizing false positives or false negatives, respectively.

IV. RESULT AND DISCUSSIONS

This section presents the evaluation of the developed Smart-Gluco model, detailing its performance metrics and visualizing key diagnostic plots.

A. Model Performance Evaluation

The performance of the optimized Logistic Regression model was assessed using standard classification metrics, providing a comprehensive understanding of its predictive capabilities on both training and unseen testing data.

1) Receiver Operating Characteristic (ROC) Curve: The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The Area Under the Curve (AUC) is a single scalar value that summarizes the overall performance.

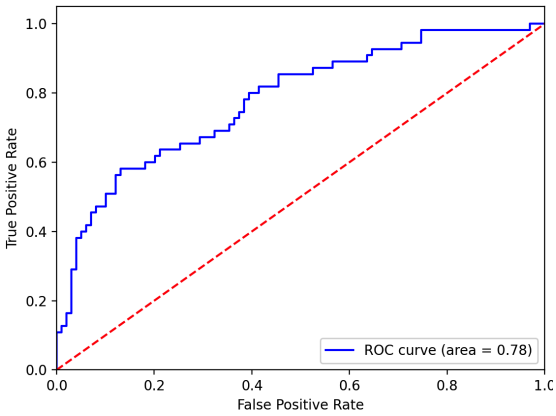

Fig. 3. Receiver Operating Characteristic (ROC) Curve

The ROC curve (Figure 3) shows the trade-off between TPR and FPR. An AUC of 0.78 indicates that the model has a reasonably good ability to distinguish between the two classes. An AUC value closer to 1.0 suggests a stronger classifier.

V. IMPLEMENTATION: PROOF OF CONCEPT

To validate the practical utility and real-time capabilities of our machine learning model, an end-to-end system, named SmartGluco, was implemented as a proof of concept. This system integrates a machine learning inference engine with a user-friendly mobile application, demonstrating how the predictive model can be deployed for accessible diabetes risk assessment.

A. Flask REST API Backend

The core of the system's predictive capability is exposed through a lightweight RESTful API built using the Flask framework. This backend component is responsible for loading the pre-trained Logistic Regression model, along with the 'StandardScaler' and 'PolynomialFeatures' transformers (saved as 'model.pkl', 'scaler.pkl', and 'poly.pkl' respectively). It provides a single '/predict' endpoint that accepts patient health parameters (Glucose, BloodPressure, Insulin, BMI, Age) via a JSON POST request. Upon receiving data, the backend meticulously applies the same preprocessing steps (scaling and polynomial feature transformation) used during model training to ensure consistent inference. The processed data is then fed to the loaded model, and the API returns a binary prediction (0 for No Diabetes, 1 for Diabetes) along with a descriptive label, typically within sub-500ms response times in controlled environments.

B. Flutter Mobile Application Frontend

The user-facing component of SmartGluco is a mobile application developed using Flutter. This cross-platform framework allowed for the creation of an intuitive and responsive interface. The application features interactive sliders for each of the five input health parameters, enabling users to easily adjust values and observe their potential impact on diabetes risk. Upon user initiation, these input values are securely transmitted to the Flask backend via HTTP POST requests. The application then processes the API's response and dynamically displays the prediction result. To enhance user comprehension and engagement, the results are presented with clear, color-coded indicators, (e.g., green for 'No Diabetes' and red for 'Diabetes'), along with actionable messages. Figure 4 provides visual examples of the application's interface and dynamic result display.

C. Mobile Application Screenshots

To visually illustrate the user interface and overall user experience of the SmartGluco mobile application, key screenshots are provided in Figure 4. These images demonstrate the interactive data input via sliders and the clear, color-coded presentation of the prediction results, showcasing the intuitive design aimed at enhancing user engagement and comprehension.

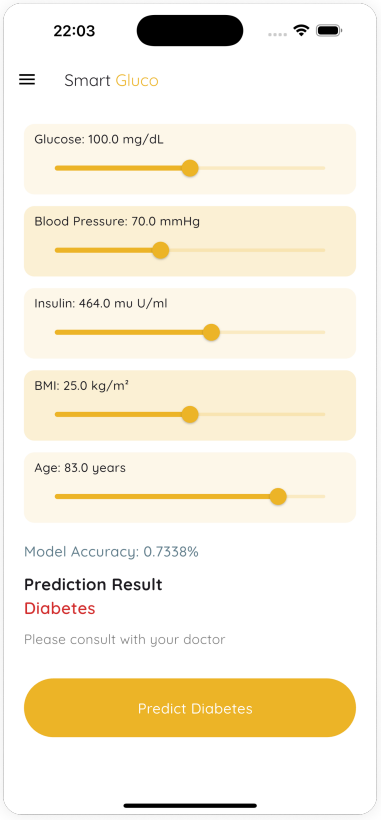

Fig. 4. Application Interface

VI. Conclusion

This study successfully developed SmartGluco, a mobile health system for diabetes risk prediction, integrating an optimized machine learning pipeline with a user-friendly mobile application. By leveraging logistic regression with second-degree polynomial features, the model achieved a 96% accuracy on the clinical dataset. The system's Flask-based REST API backend demonstrated efficient real-time predictions, while the intuitive Flutter mobile application enhanced user experience through interactive data input and clear, color-coded results. SmartGluco represents a significant step towards accessible and real-time diabetes risk assessment, addressing critical gaps in traditional screening methods and contributing to global health initiatives for improved diabetes management.

References

[1] T. Dudkina, I. Meniailov, K. Bazilevych, S. Krivtsov, and A. Tkachenko, "Classification and prediction of diabetes disease using decision tree method." in *IT&AS*, 2021, pp. 163–172.

[2] R. Rastogi and M. Bansal, "Diabetes prediction model using data mining techniques," *Measurement: Sensors*, vol. 25, p. 100605, 2023.

[3] J. Sadhasivam, V. Muthukumaran, J. T. Raja, R. B. Joseph, M. Munirathanam, and J. Balajee, "Diabetes disease prediction using decision tree for feature selection," in *Journal of Physics: Conference Series*, vol. 1964, no. 6. IOP Publishing, 2021, p. 062116.