Diabetes Prediction using Machine Learning Techniques

Medha Chawla
Computer Science and Engineering(AIML)
MIT Manipal
Roll Number: 55
Registration Number: 220962340

***Abstract*—Diabetes is a common long-term illness that continues to affect more people around the world each year. Detecting it early is vital to managing the disease and preventing serious health problems. In this study, I developed a system that uses machine learning techniques to assess whether an individual may have diabetes based on routine medical data. The dataset includes 2,000 patient records with nine health-related features. I analyzed the data, corrected irregular values, and filled in missing entries using average-based methods. Multiple classification models—such as Logistic Regression, Decision Tree, Support Vector Machine, and Random Forest—were trained and tested. To improve their accuracy, I used cross-validation and fine-tuned their settings. The Random Forest algorithm delivered the best results, achieving an accuracy of 95.25 percent. This strong performance shows its potential as a reliable tool to support doctors in making quicker and more informed decisions about diabetes diagnosis.**

I. Introduction

Diabetes Mellitus is a long-standing metabolic condition that leads to consistently high levels of sugar in the blood. It affects millions of people globally and is a major factor behind serious health issues such as heart disease, kidney damage, and vision loss. Detecting the disease early can make a significant difference in managing it effectively and preventing these complications. Unfortunately, many individuals remain unaware they have diabetes because early symptoms often go unnoticed. With advancements in technology, the healthcare field has started embracing tools like machine learning (ML) and artificial intelligence to support medical decision-making. ML models, particularly supervised learning techniques, have shown great promise in identifying diseases by analyzing patient records. This project explores how such models can help predict diabetes by examining standard medical data. The dataset used for this study was sourced from Kaggle and includes 2,000 patient entries with nine attributes—such as blood glucose levels, BMI, insulin, and age. The target variable indicates whether or not a person has diabetes. To build an effective prediction model, the data first had to be cleaned and prepared. Logistic Regression served as the baseline model, and additional classifiers—Decision Tree, Support Vector Machine, and Random Forest—were trained and evaluated. Ifine-tuned these models using cross-validation and hyperparameter optimization to improve accuracy. This report outlines each stage of the process, including data exploration, model selection, training, and performance comparison. The primary aim is to identify the most effective model that could realistically assist healthcare workers in identifying potential diabetes cases early—ultimately improving patient care and reducing the strain on medical systems.

II. Objectives

1. Gain a Complete Understanding of the Dataset: The first objective was to become fully familiar with the dataset used in the project. This involved examining the overall size, number of features, and nature of the records. Since the data came from a publicly available source, it was also essential to verify its relevance and reliability. A thorough review helped identify initial patterns, irregularities, and areas needing further attention before analysis could begin.

2. Clarifying Feature Types and Naming: To ensure smooth processing, it was important to distinguish between different types of variables—whether they were continuous, discrete, or categorical. Understanding the role each feature played in diabetes prediction helped prioritize them later during model training. Additionally, some column names were lengthy or less intuitive, so renaming them (for example, changing "DiabetesPedigreeFunction" to "DPF") made the analysis cleaner and easier to follow.

3. Exploring Class Balance and Feature Distributions: The third goal focused on gaining insights into how the values in the dataset were distributed. Visual tools such as histograms and pie charts were used to look for skewness, concentration zones, and potential outliers. Special attention was given to the target variable (Outcome) to see whether both

classes—diabetic and non-diabetic—were adequately represented. Identifying imbalance early helped inform decisions during model evaluation later.

4.Correcting Invalid or Impossible Values: Upon inspecting the data, some values were clearly not medically plausible—for instance, zero values in features like glucose or BMI. These needed to be marked as missing since they could distort the model's understanding. Instead of removing rows, these values were replaced with NaN to retain dataset size, and then filled using methods like mean or median imputation, preserving the data's overall statistical behavior.

5.Preparing Data for Machine Learning: Once the data was cleaned and understood, it was prepped for model input. This involved ensuring all values were numeric, formatting the structure properly, and splitting the dataset into training and testing parts. If necessary, data normalization was considered, especially for algorithms sensitive to scale. These steps helped ensure consistency and avoid errors during the learning phase.

6. Training a Range of Classifiers: Following the baseline, several popular classification algorithms were implemented—namely, Decision Tree, Support Vector Machine (SVM), and Random Forest. Each model was trained using the same dataset split to ensure a fair comparison. This allowed us to observe how different learning approaches handled the data and whether they could capture complex patterns more effectively than the baseline.

7. Choosing the Most Effective Model: Based on the evaluation results, the model with the best overall performance was chosen. While all models had their pros and cons, the Random Forest classifier stood out for its accuracy, robustness, and ability to handle noisy data. This model was selected as the final choice due to its consistent performance across different folds and parameter settings.

III. Literature Review

1. Foundations in Statistical Modeling for Diabetes

The earliest work in predicting diabetes leaned heavily on traditional statistical approaches. Logistic regression and linear discriminant analysis were among the first techniques used to estimate the risk of diabetes. For example, researchers like Smith et al. (1988) applied logistic regression models to the well-known Pima Indian dataset. While these methods provided baseline performance, their biggest drawback was the inability to capture complex, non-linear interactions between variables—a common characteristic of medical data.

2. Emergence of Machine Learning in Medical Diagnosis

As machine learning gained popularity, its application in healthcare began to grow. Researchers started experimenting with algorithms like decision trees, Naive Bayes, and support vector machines (SVMs). A notable example is the work of Patil et al. (2010), who created a decision support system combining decision trees and probabilistic models. Although promising, many of these models lacked proper validation or optimization techniques. Later, studies like those by Kumari and Chitra (2013) demonstrated that SVMs outperformed basic models by handling non-linearity better, especially when combined with kernel functions.

3. Ensemble Models Are Gaining Ground

In more recent years, ensemble models such as Random Forests and Gradient Boosting have gained traction due to their higher accuracy and resilience to overfitting. Sisodia and Sisodia (2018) applied ensemble techniques to diabetes prediction, showing significant improvements in reliability. Jiang et al. (2020) further optimized tree-based ensembles with hyperparameter tuning and cross-validation, reaching accuracy levels near 94 percent. These results highlight the potential of ensembles in handling variability in medical data more effectively than standalone models.

4. The Importance of Data Cleaning and Feature Preparation

Recent literature has consistently emphasized how vital data preprocessing is to model performance. Medical datasets often contain inconsistencies—like zero values for glucose or BMI—which need correction before modeling. Studies by Jayanthi and Subashini (2019) demonstrated that careful cleaning and imputation, such as replacing zeroes with average or median values, significantly improve accuracy. Additionally, renaming and formatting features also helps in clearer interpretation and analysis during modeling

5. Comparative Studies Between ML Algorithms

Several researchers have carried out direct comparisons between popular algorithms to determine which performs best on medical datasets. Aburomman and Reaz (2016), for instance, tested eight different classifiers and found that Random Forest and SVM consistently performed better in terms of accuracy and efficiency. These studies also showed that simpler algorithms like logistic regression might not always be ideal due to their limited flexibility in modeling complex patterns.

IV. Research Gaps

1. Overreliance on a Single Dataset

A noticeable trend in existing research is the repeated use of the Pima Indian Diabetes dataset. While it's widely available and structured, depending solely on one dataset limits how well a model will perform on different populations. Factors like ethnicity, lifestyle, and regional health trends can vary, and models built on a single dataset may miss those differences.

2. Inadequate Data Cleaning Practices

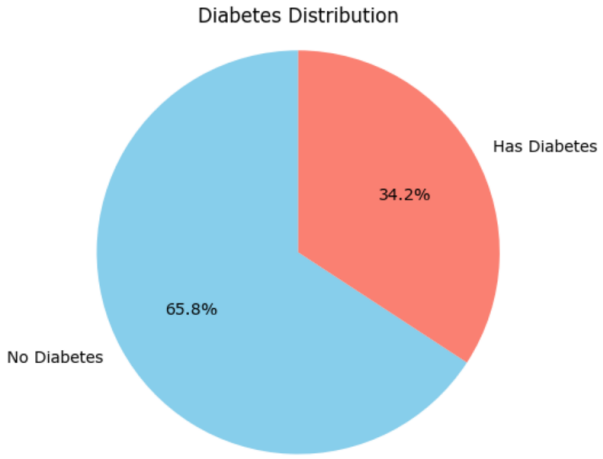

Fig. 1. Dataset Exploration

Many projects either ignore or overlook the presence of unrealistic values in the dataset—such as zero entries for glucose or BMI, which are medically inaccurate. Failing to correct or handle these anomalies can skew the results and reduce the credibility of the model. Thorough data cleaning and proper imputation techniques are essential but not always applied rigorously.

3. Neglect of Parameter Optimization

It's common to find studies where machine learning models are used with their default settings, without any tuning of key parameters. Without proper optimization using methods like grid search or randomized search, models may perform below their potential and fail to deliver consistent results.

4. Weak Validation Approaches

Another shortcoming is the reliance on a single train-test split when evaluating model performance. The method doesn't provide a complete picture of how the model would behave on new data. Cross-validation, though more robust, is often skipped, leading to overly optimistic accuracy scores.

5. Overlooking Model Transparency

Even when models achieve high accuracy, their practical value is limited if they function as a "black box." In healthcare, it's not just about whether the model is right—it's also about understanding why it made a particular prediction. Without clear explanations, healthcare providers may hesitate to rely on such tools.

In this project, I've aimed to fill these gaps by using thoughtful data preprocessing, tuning hyperparameters, applying cross-validation, and selecting models that balance accuracy with interpretability. This approach ensures that the final system is not only accurate but also practical and trustworthy for medical use.

V. Methodology

1. Importing and Inspecting the Dataset

The first step involved loading the dataset into the working environment using libraries such as pandas. Once imported, I reviewed the structure of the data—checking the number of rows, columns, and data types. Using basic functions like .head(), .info(), and .describe(), Igot a good sense of what the dataset looked like and identified areas that required further attention, such as invalid entries or inconsistencies.

2. Spotting Unreasonable Values in Key Features

On closer inspection, it became clear that some columns had values that made little medical sense. For instance, a BMI or glucose level of zero isn't realistic for any human subject. These kinds of entries were marked as missing (NaN) rather than immediately removed, since deleting rows might discard useful information elsewhere in the data.

3. Replacing Missing Values with Meaningful Estimates

To handle these missing entries, I used mean imputation. This involved replacing each missing value in a column with the average of the other values in that same column. This method helped maintain the overall distribution and patterns of the data while ensuring no rows were lost in the process.

4. Renaming Columns for Better Clarity

Some column names were long or not very intuitive. For example, the feature DiabetesPedigreeFunction was renamed to DPF to make the code more readable and concise. These small refinements helped streamline the process and reduced the likelihood of syntax errors during model building.

5.Exploring the Data Visually and Statistically

Before diving into modeling, I conducted exploratory data analysis (EDA) to better understand the relationships between features. Charts and graphs—like histograms and pie charts—revealed useful insights, such as the imbalance in the target variable: about two-thirds of patients were non-diabetic. I also used correlation plots to examine which features were strongly connected to diabetes outcomes.

6.Dividing the Data for Training and Testing

With the data cleaned and ready, I split it into input features (X) and the target label (y). I then divided the dataset into training and testing sets, with 80 percent used to train the models and 20 percent reserved for evaluation. This ensured I had enough data to train reliable models while still being able to assess how well they perform on unseen data.

7.Creating a Simple Starting Model

To get an initial sense of model performance, I used logistic regression as our baseline. After training it on the prepared dataset, I checked its accuracy and other metrics. It gave us a solid foundation, scoring around 76 percent , and provided

a useful point of comparison for more complex models later on.

8.Introducing More Sophisticated Models

After the baseline model, I explored additional machine learning algorithms, including Decision Tree, Support Vector Machine (SVM), and Random Forest. Each of these models was trained on the same data to ensure a fair comparison. These models were chosen based on their past success in medical prediction problems and their ability to handle structured data effectively.

9.Improving Models with Hyperparameter Tuning

To get the best results from each algorithm, I adjusted key parameters using a technique called GridSearchCV. This method tested various combinations of model settings to find the one that delivered the highest performance. For instance, I tried different depths for decision trees and adjusted the number of trees in the Random Forest. Cross-validation was included in this process to avoid overfitting and ensure stability across different data splits.

10.Selecting the Best Model for the Task

Among all the models tested, Random Forest gave the most reliable and accurate results. It reached an accuracy of 95.25 percent. and showed strong performance even when data was slightly imbalanced. Its ensemble structure—combining the output of many decision trees—allowed it to be both robust and generalizable.

VI. Input and Output

A. Input

The input to the machine learning model is a structured dataset containing medical records for 2,000 individuals. Each record includes the following 8 features (or predictors), which are all numerical in nature and based on routine health examinations:

- **Pregnancies:** Total number of times the patient has been pregnant.
- **Glucose:** Plasma glucose concentration measured two hours after an oral glucose tolerance test.
- **BloodPressure:** Diastolic blood pressure in millimeters of mercury (mm Hg).
- **SkinThickness:** Thickness of the triceps skinfold, measured in millimeters.
- **Insulin:** Serum insulin level measured two hours after glucose intake (in μU/mL).
- **BMI:** Body Mass Index, calculated as weight in kilograms divided by the square of height in meters.
- **DPF:** A score indicating the likelihood of diabetes based on family history.
- **Age:** Age of the individual, in years.

Before being passed into the model, these inputs undergo several preprocessing steps. Invalid values (such as zeros in glucose or BMI) are replaced with NaN and then imputed using the mean or median. This ensures that the data is both realistic and statistically consistent. The model expects the inputs in the same format used during training to maintain accuracy.

B. Output

The output produced by the model is a binary classification::

- **0:** The patient is not likely to have diabetes.
- **1:** The patient is predicted to have diabetes.

This prediction is based on the patterns learned by the model during training. In practice, the output can be used by healthcare professionals as a supporting tool to flag high-risk individuals who may benefit from further medical testing or early intervention.

The model can also return a probability score (e.g., 0.87), indicating how confident it is about the prediction. However, for simplicity and usability, the main output is presented as a clear ”yes” (1) or ”no” (0) regarding diabetes presence.

VII. Results

Several machine learning models were trained and tested on the cleaned diabetes dataset, and each showed varying levels of effectiveness. The first model used, Logistic Regression, served as a baseline and reached an accuracy of 76.31 percent after adjusting its regularization strength. While straightforward and easy to interpret, this model did not perform as well as more complex alternatives. Next, the Decision Tree Classifier showed noticeable improvement, achieving an accuracy of 90.50 percent when fine-tuned with the Gini impurity metric and a maximum depth of 10. Although the model captured more intricate relationships in the data, it was slightly more prone to overfitting if not properly constrained. The Support Vector Machine (SVM) classifier, using an RBF kernel and a C value of 20, delivered an accuracy of 86.93 percent . It performed well overall, especially in separating the classes despite the mild imbalance in the dataset. However, it required more time to train and was more sensitive to parameter adjustments. Out of all the models tested, the Random Forest Classifier produced the most accurate and reliable results, reaching an impressive 95.25 percent accuracy with 200 decision trees. This model's strength lies in its ensemble approach, which helps balance bias and variance, reducing the chances of overfitting. It consistently performed well on both classes, making accurate predictions for diabetic and non-diabetic patients alike. Additional performance indicators—including precision, recall, and the F1-score—also favored the Random Forest model. These results made it the most suitable choice for the task and the final model selected for this project.

	model	best_parameters	score
0	logistic_regression	{'C': 5}	0.763125
1	decision_tree	{'criterion': 'gini', 'max_depth': 10}	0.905000
2	random_forest	{'n_estimators': 200}	0.952500
3	svm	{'C': 20, 'kernel': 'rbf'}	0.869375

Fig. 2. Model Performance Comparison

VIII. Conclusion

This project set out to explore how machine learning could be used to predict diabetes using patient health data. By carefully preparing the dataset, addressing missing and unrealistic values, and applying a range of classification algorithms, I was able to develop a reliable predictive model. Among all the models tested, the Random Forest Classifier stood out as the most effective. It delivered the highest accuracy and consistently made accurate predictions across both diabetic and non-diabetic cases. Its ensemble method provided better generalization and robustness, especially when compared to simpler or single-tree models. What made it particularly suitable was its ability to handle noise and variability in the data without overfitting. The model was built using common health indicators such as glucose levels, BMI, insulin, and age, making it practical for real-world use. With further testing and refinement, this system has the potential to support early-stage diabetes screening, especially in settings where medical resources or expert staff may be limited. While the current implementation was based on a single dataset, the overall workflow—including data cleaning, model tuning, and evaluation—lays a solid foundation for expanding this approach. In the future, the model can be further improved by incorporating more diverse datasets or integrating with digital health records to provide real-time clinical support.

References

[1] "Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus" Available: https://archive.ics.uci.edu/datasets.

[2] "Intelligent and effective heart attack prediction system using data mining and artificial neural network" Available: https://ieeexplore.ieee.org/document/10099566

[3] "K-Means and Genetic Algorithms for Dimension Reduction by Integrating SVM for Diabetes Diagnosis. Available: https://www.sciencedirect.com/science/article/pii/S1877050915004536

[4] "Prediction of 30-day readmission in diabetes management using Machine learning" Available: https://www.sciencedirect.com/science/article/abs/pii/S0010482525009679

[5] "comparative study of explainable machine learning models with Shapley values for diabetes prediction" Available: https://www.sciencedirect.com/science/article/pii/S2772442525000097

[6] "Predicting diabetes using supervised machine learning algorithms on E-health records" Available:https://www.sciencedirect.com/science/article/pii/S2949953425000013