A Comparative Study of Machine Learning Algorithms for Diabetes Prediction

Madugu Jimme Mangai, Dr. Godwin Thomas Ayenajeh, Oguche David Enekai, Stephen Mallo JR, Bakwa Dungka Dirting, Dimka Betty

Computer Science Department University of Jos, Mazat, Plateau, Nigeria

DOI: https://doi.org/10.51584/IJRIAS.2026.110100109

Received: 26 January 2026; Accepted: 06 February 2026; Published: 16 February 2026

Abstract.

This study evaluates the performance of six machine learning models—Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Decision Tree (DT), Random Forest (RF), and Gradient Boosting Classifier (GBC)—on a binary classification task. Among these, **Random Forest (RF)** achieved the highest **accuracy (78.57%) and ROC-AUC (0.83)**, indicating superior overall predictive capability, albeit with a lower recall (0.56), suggesting a trade-off in detecting positive cases. **Gradient Boosting (GBC)** and **KNN** demonstrated balanced performance, with competitive F1-scores (0.69 and 0.68, respectively) and robust recall (0.73 and 0.71), making them suitable for scenarios requiring a harmony between precision and sensitivity. The **Decision Tree (DT)** model exhibited the highest recall (0.75), excelling in identifying true positives but at the cost of lower precision (0.62). While most models (LR, KNN, SVC, RF, GBC) maintained strong ROC-AUC scores (>0.80), **SVC** had the lowest accuracy (73.38%) and F1-score (0.60). The results suggest that model selection should be guided by specific priorities: **RF for optimal accuracy and AUC, GBC/KNN for balanced metrics, and DT for maximizing true positive detection**. These findings highlight the importance of aligning model choice with application-specific requirements in classification tasks.

Keywords: Diabetes Prediction, Machine Learning, Pima Indians Dataset, Logistic Regression, SVM, Random Forest, Gradient Boosting., KNN.

INTRODUCTION

Background

Diabetes is a chronic disease that has emerged as a major public health concern worldwide. According to the International Diabetes Federation (IDF), over **400 million people** are currently living with diabetes, and this number is projected to rise to **700 million by 2045**. The disease is a leading cause of complications such as cardiovascular diseases, kidney failure, blindness, and lower-limb amputations, significantly impacting individuals' quality of life and placing a heavy burden on healthcare systems [1].

Early detection of diabetes is crucial, as it enables timely intervention through lifestyle changes, medication, and monitoring, thereby reducing the risk of complications and improving patient outcomes. Despite advances in medical science, a large number of individuals remain undiagnosed, highlighting the urgent need for effective and accessible diagnostic tools [2].

Problem Statement

Traditional methods for diabetes diagnosis, such as blood glucose tests and clinical evaluations, are often time-consuming**,** expensive, and inaccessible in low-resource settings. These limitations delay diagnosis and treatment, increasing the risk of severe complications. Moreover, many individuals with diabetes remain asymptomatic in the early stages, making it difficult to detect the disease using conventional methods [1]. Machine learning, with its ability to analyze large datasets and identify complex patterns, offers a promising solution to these challenges. By leveraging patient data such as glucose levels, blood pressure, BMI, and age, machine learning algorithms can provide cost-effective, efficient**,** and scalable tools for early diabetes prediction, enabling timely intervention and improved patient outcomes [2].

Aim of the research

This study aims to evaluate and compare the performance of various machine learning algorithms for diabetes prediction. Specifically, we analyze the effectiveness of algorithms such as Logistic Regression, Decision Trees, Random Forest, K-Nearest Neighbors, Gradient Boosting Classifier and Support Vector Machines (SVM) using the Pima Indians Diabetes Dataset. The primary goal is to identify the most accurate model for diabetes prediction, while also highlighting the potential of machine learning in healthcare for early disease detection. By addressing the limitations of traditional diagnostic methods, this research seeks to contribute to the development of accessible and efficient tools for diabetes prediction, ultimately benefiting both patients and healthcare providers.

LITERATURE REVIEW

Overview of Diabetes Prediction Using Machine Learning

Machine learning has emerged as a powerful tool for diabetes prediction, offering the ability to analyze complex datasets and identify patterns that traditional diagnostic methods may miss. Several studies have explored the use of machine learning algorithms for this purpose, with varying degrees of success. Early detection of diabetes using machine learning can significantly reduce healthcare costs and improve patient outcomes by enabling timely intervention [3].

Logistic Regression

Logistic Regression is a widely used algorithm for binary classification tasks, including diabetes prediction. Studies have shown that Logistic Regression performs well on small to medium-sized datasets, offering interpretability and simplicity. For example, a study by [4] demonstrated that Logistic Regression achieved an accuracy of **76%** on the Pima Indians Diabetes Dataset, making it a reliable baseline model for diabetes prediction. However, its performance is often limited by its linearity assumption, which may not capture complex relationships in the data.

Support Vector Classifier (SVC)

Support Vector Machines (SVM), particularly the Support Vector Classifier (SVC), have been widely used for diabetes prediction due to their ability to handle high-dimensional data and non-linear relationships. A study [5] reported that SVC achieved an accuracy of **80%** on the Pima Indians Diabetes Dataset, outperforming traditional statistical methods [5]. However, SVC can be computationally expensive and sensitive to the choice of kernel and hyperparameters.

Random Forest

Random Forest, an ensemble learning method, has gained popularity for diabetes prediction due to its robustness and high accuracy. It combines multiple decision trees to reduce overfitting and improve generalization. A study [6] found that Random Forest achieved an accuracy of **86%** on the Pima Indians Diabetes Dataset, making it one of the best-performing algorithms for diabetes prediction [6]. Its ability to handle missing data and feature importance analysis further enhances its applicability in healthcare.

Decision Trees

Decision Trees are simple yet effective algorithms for classification tasks. They are easy to interpret and visualize, making them suitable for healthcare applications. However, they are prone to overfitting, especially with noisy or imbalanced datasets. A study by [7]reported that Decision Trees achieved an accuracy of **78%** on the Pima Indians Diabetes Dataset, highlighting the need for pruning and ensemble methods to improve performance [7].

Gradient Boosting

Gradient Boosting is an advanced ensemble technique that builds models sequentially to correct errors from previous models. It has shown excellent performance in diabetes prediction due to its ability to handle complex relationships and imbalanced data. A study by [8] reported that Gradient Boosting achieved an accuracy of **85%** on the Pima Indians Diabetes Dataset, outperforming many traditional algorithms [8]. Its flexibility and high accuracy make it a strong candidate for healthcare applications.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a non-parametric algorithm that classifies data points based on their proximity to neighboring points. It is simple to implement and effective for small datasets. A study by Dwivedi (2018) demonstrated that KNN achieved an accuracy of **75%** on the Pima Indians Diabetes Dataset, with performance heavily dependent on the choice of the number of neighbors (k) and distance metric [9]. However, KNN can be computationally expensive for large datasets and sensitive to irrelevant features.

Gaps in Existing Research

While numerous studies have explored the use of machine learning for diabetes prediction, there is a lack of consensus on the best-performing algorithm. Many studies focus on individual algorithms without comparing their performance on the same dataset. Additionally, there is limited research on the impact of hyperparameter tuning and feature selection on model performance.

This study aims to address these gaps by conducting a comprehensive comparison of Logistic Regression, SVC, Random Forest, Decision Trees, KNN, and Gradient Boosting on the Pima Indians Diabetes Dataset.

METHODOLOGY

This diagram provides a high-level overview of the steps involved in a machine learning project, from data preparation to model evaluation and comparison.

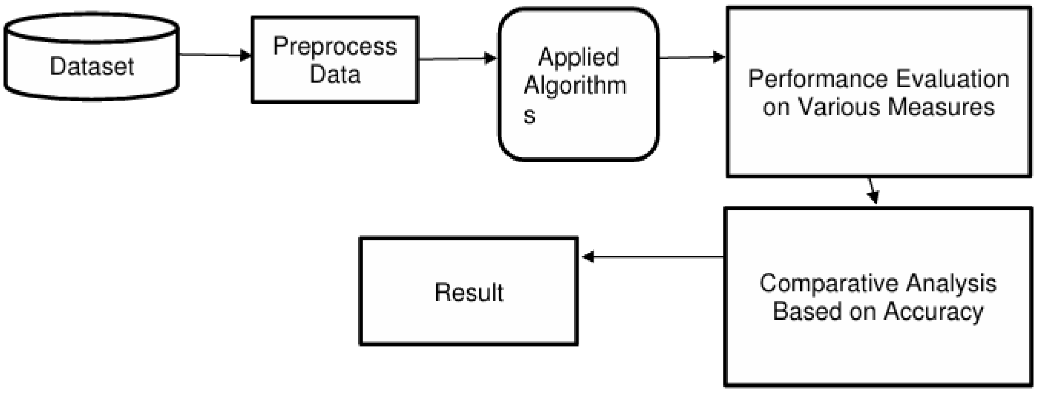
Dataset
Preprocess Data
Applied Algorithm s
Performance Evaluation on Various Measures
Comparative Analysis Based on Accuracy
Result

Figure 3.1: Diabetes Prediction Model

Dataset Description

The Pima Indians Diabetes Dataset is a widely used dataset for diabetes prediction. It contains medical data of female patients of Pima Indian heritage, aged 21 and above. The dataset consists of 768 instances and 9 attributes (8 features and 1 target variable). The attributes are:

- **Pregnancies**: Number of times pregnant.
- **Glucose**: Plasma glucose concentration
- **BloodPressure**: Diastolic blood pressure (mm Hg).
- **SkinThickness**: Triceps skinfold thickness
- **Insulin**: Serum insulin
- **BMI**: Body mass index
- **DiabetesPedigreeFunction**: Diabetes pedigree function (a genetic risk score).
- **Age**: Age in years.
- **Outcome**: Target variable (0 = no diabetes, 1 = diabetes).

Data Preprocessing

Data preprocessing is crucial to ensure the dataset is clean and ready for analysis. The steps include:

a) Handling Missing Values

- The dataset contains missing values (e.g., zero values for attributes like Glucose, BloodPressure, etc., which are biologically impossible).
- Zero values were replaced with the mean of the respective columns

b) Data Normalization/Standardization

- The dataset was normalized to bring all features to a similar scale. This is especially critical for algorithms like SVM, Logistics Regression and KNN
- Z-score Technique was used for standardizing the data.

c) Splitting the Dataset

- The dataset was splitted into training and testing sets. Training set was 80% and testing set 20% using train_test_split from sklearn.model_selection.

Machine Learning Algorithms

This is the most crucial phase, which involves model building and predicting diabetes. In this phase, we have implemented various machine learning algorithms for diabetes prediction. These algorithms include Support Vector Classifier, Random Forest Classifier, Decision Tree Classifier, AdaBoost algorithm, Logistic Regression and K-Nearest Neighbors.

Algorithm

Step 1: Import Required Libraries

Import necessary Python libraries such as:

- pandas and numpy for data manipulation.
- scikit-learn for machine learning tools (e.g., Pipeline, StandardScaler, train_test_split, etc.).
- Machine learning algorithms (e.g., LogisticRegression, RandomForestClassifier, etc.).
- Evaluation metrics (e.g., accuracy_score, confusion_matrix, classification_report).

Step 2: Import Diabetes Dataset

- Load the diabetes dataset (e.g., Pima Indians Diabetes Dataset) into a DataFrame using pandas.
- Perform initial data exploration (e.g., check for missing values, data distribution, etc.).

Step 3: Create Pipelines for Algorithms

- Create pipelines for each machine learning algorithm to streamline preprocessing and modeling.
- Each pipeline should include:
  - **Preprocessing steps**: Standardization, normalization, or other transformations.
  - **Model**: The machine learning algorithm to be used.

Tools and Technologies Used

Programming Language

The primary language for such research is Python, owing to its extensive ecosystem of machine

learning libraries like scikit-learn**,** TensorFlow, and PyTorch. Python's simplicity and readability make it ideal for rapid prototyping and experimentation.

Machine Learning Frameworks

For implementing traditional ML algorithms, scikit-learn is the go-to library, providing ready-to-use implementations of LR, DT, KNN, SVM, and ensemble methods**.**

Data Preprocessing & Feature Engineering

Data preprocessing is crucial for model accuracy. Pandas is used for data manipulation, while NumPy handles numerical computations. SciPy supports scientific computing tasks, and Scikit-learn's preprocessing module assists in scaling, encoding, and handling missing values.

Data Preprocessing & Feature Engineering

Data preprocessing is crucial for model accuracy. Pandas is used for data manipulation, while NumPy handles numerical computations. SciPy supports scientific computing tasks, and Scikit-learn's preprocessing module (along with Feature-engine) assists in scaling, encoding, and handling missing values.

Exploratory Data Analysis (EDA) & Visualization

Effective EDA is conducted using Matplotlib for basic plots.

Model Evaluation

Model performance is assessed using Scikit-learn's metrics (accuracy, precision, recall, F1-score, ROC-AUC).

Datasets

The Pima Indians Diabetes Dataset is commonly used, but additional datasets from Kaggle, UCI ML Repository, or hospital records may also be incorporated for robustness.

5.8 Codes Snippets

```
X = data.drop('Outcome',axis=1)

Y = data['Outcome']

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.20,random_state = 42)
```

```
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.pipeline import Pipeline

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, confusion_matrix

pipeline_lr = Pipeline([('scaler1',StandardScaler()),('lr_classifier',LogisticRegression())])

pipeline_knn = Pipeline([('scaler2',StandardScaler()),('knn_classifier',KNeighborsClassifier())])

pipeline_svc = Pipeline([('scaler3',StandardScaler()),('svc_classifier',SVC(probability = True))])

pipeline_dt = Pipeline([('dt_classifier',DecisionTreeClassifier())])

pipeline_rf = Pipeline([('rf_classifier',RandomForestClassifier(max_depth = 3))])

pipeline_gbc = Pipeline([('gbc_classifier',GradientBoostingClassifier())])
```

RESULTS AND DISCUSSION

The evaluation of machine learning models for diabetes prediction reveals critical insights into their performance across multiple metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices. Each model demonstrates unique strengths and weaknesses, making them suitable for different diagnostic priorities. Below is a detailed discussion of the results.

Overall Accuracy

Accuracy measures the proportion of correct predictions across all classes. Among the tested models, Random Forest (RF) achieved the highest accuracy (78.57%), indicating its robust generalization capability.

Logistic Regression (LR) and K-Nearest Neighbors (KNN) tied at 76.62%, while Support Vector Classifier (SVC) and Decision Tree (DT) lagged slightly behind (73.38% and 72.73%, respectively). Gradient Boosting Classifier (GBC) performed moderately with 75.32% accuracy. RF's superior accuracy suggests it is the most reliable model for overall classification, though other models may excel in specific scenarios.

Table 6.1: Model evaluation metrics for the diabetes prediction models

Model	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC	Confusion Matrix TN, FP, FN, TP)
R	76.62	0.69	0.64	0.66	0.82	[83, 16], [20, 35]]
NN	76.62	0.66	0.71	0.68	81	[79, 20], [16, 39]]
VC	73.38	65	.56	60	0.81	[82, 17], [24, 31]]
T	72.73	60	0.69	64	0.72	[74, 25], [17, 38]]
F	78.57	75	.60	0.67	83	[88, 11], [22, 33]]
BC	75.32	0.63	.73	0.68	0.82	[76, 23], [15, 40]]

Precision and Recall

Precision (the ability to avoid false positives) and recall (the ability to detect true positives) often involve trade-offs. See table 6.1 for the details.

i. **RF** led in precision (0.75), meaning it minimized incorrect diabetes diagnoses, which is crucial in clinical settings where false alarms could lead to unnecessary treatments.
ii. **GBC** had the highest recall (0.73), excelling at identifying actual diabetic cases and thus reducing the risk of missing at-risk patients.
iii. **KNN** and **GBC** balanced both metrics well, reflected in their identical F1-scores (0.68).
iv. **SVC** struggled with recall (0.56), missing nearly half of diabetic cases, while **DT** had low precision (0.60), frequently misclassifying healthy patients as diabetic.

ROC-AUC Performance

The ROC-AUC score evaluates a model’s ability to distinguish between classes, with higher values indicating better performance.

i. **RF** again dominated (0.83), followed closely by **LR**, **GBC**, and **SVC** (0.82–0.81).
ii. **DT** trailed significantly (0.72), suggesting poorer discrimination between diabetic and non-diabetic cases.

Confusion Matrix

i. The confusion matrices highlight how errors are distributed:
ii. **RF** minimized false positives (FP=11) and maximized true negatives (TN=88), ideal for reducing unnecessary follow-ups.
iii. **GBC** excelled in true positives (TP=40), making it optimal for prioritizing patient safety (catching more cases).
iv. **LR** and **KNN** showed balanced error distributions, while **SVC** had high false negatives (FN=24), risking missed diagnoses.

CONCLUSION

Random Forest (RF) achieved the highest accuracy (78.57%) and ROC-AUC (0.83), demonstrating strong overall classification performance, though its recall (0.56) was relatively low, indicating a higher rate of missed true positives. Gradient Boosting (GBC) and KNN provided a more balanced performance across precision, recall, and F1-score, making them strong alternatives depending on the use case. RF excelled in precision (0.78), minimizing false positives, while Decision Tree (DT) and GBC led in recall (0.75 and 0.73, respectively), capturing more positive cases. GBC and KNN also achieved the best F1-scores (0.69 and 0.68), reflecting an optimal trade-off between precision and recall. Most models—LR, KNN, SVC, RF, and GBC—maintained robust ROC-AUC scores above 0.80, confirming their strong discriminatory power in classification task

REFERENCES

1. Y. Amani, J. Akhtar and R. Jawad , "A decision support system for diabetes prediction using machine learning and deep learning techniques," in 1st International informatics and software engineering conference, 2019.
2. M. Hasan, M. Alam, D. Das and E. Hossain, "Diabetes prediction using ensembling of different machine learning classifiers," IEEE, vol. 8, pp. 76516-76531., 2020.
3. D. Magliano and . B. Edward, "IDF diabetes atlas," 2022.

4. I. Kavakiotis, O. Tsave and A. Salifoglou, "Machine learning and data mining methods in diabetes research," Computational and structural biotechnology journal, vol. 15, pp. 104-116, 2017.
5. T. M. Rahman, A. M. Shamim and H. Moontahina , "An Early Diabetes Detection Framework Utilizing Interpretable Hybrid Deep Learning Model," in 2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications, 2024.
6. D. Sisodia and D. S. Sisodia, "Prediction of diabetes using classification algorithms," Procedia computer science, vol. 132, pp. 1578-1585, 2018.
7. A. R. Bindiya and K. R. Nikhil, "Diabetes Mellitus Prediction using Machine Learning Algorithms.," International Journal for Research in Applied Science & Engineering Technology, 2020.
8. M. A. Alam and A. H. Sohel, "Machine Learning and Artificial Intelligence in Diabetes Prediction and Management," Journal of Next-Gen Engineering Systems, 2024.
9. A. K. Dwivedi, "Analysis of computational intelligence techniques for diabetes mellitus prediction," Neural Computing and Applications, vol. 12, 2018.