International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI: 10.55041/ISJEM05953
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Smart Health: Diabetes Prediction System

Saurabh kumar singh

Computer Science and Engineering Parul University
Vadodara, Gujarat, India 2203051050525@paruluniversity.ac.in
Project Guide-**prof.Sujaya Bhattcharjee**

1. Abstract

Early detection of diabetes is crucial for effective disease management and improved patient outcomes. This research presents the design and implementation of a Smart Health: Diabetes Prediction System utilizing machine learning techniques. The system leverages various classification algorithms, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and XGBoost, to determine the most accurate predictive model. Hyperparameter tuning is applied using GridSearchCV to optimize model performance.The methodology involves data collection, preprocessing, feature selection, normalization, model training, and evaluation based on accuracy, precision, recall, and F1-score. A web-based application is developed using Flask and Streamlit, allowing real-time predictions for both patients and healthcare professionals. The model is deployed using Streamlit Cloud for enhanced accessibility.
Experimental results demonstrate that the system provides a highly accurate and efficient diabetes prediction tool, enabling proactive healthcare interventions. This study highlights the potential of AI-driven solutions in enhancing the accuracy, accessibility, and efficiency of diabetes diagnosis, contributing to the advancement of predictive healthcare technologies.

Keywords – Diabetes Prediction, Machine Learning, Healthcare AI, Medical Diagnosis, Predictive Analytics, Classification Algorithms, Feature Selection, Hyperparameter Tuning, Web- based Health System, Cloud Deployment, Early Disease Detection.

2. INTRODUCTION

Diabetes is a chronic disease that affects millions worldwide, making early detection and proactive management crucial for improving health outcomes. Traditional diagnostic methods rely on clinical tests and expert evaluations, which may not always be accessible or timely. To address this challenge, we propose the Smart Health: Diabetes Prediction System, an AI-powered solution that leverages machine learning to predict diabetes risk efficiently and accurately. This system is designed to assist both healthcare professionals and individuals in assessing diabetes risk based on medical parameters. By integrating a user-friendly web-based interface, users can input their health data and receive instant predictions, enabling early intervention and personalized healthcare decisions. The project employs advanced machine learning techniques, utilizing algorithms such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and XGBoost to determine the most effective predictive model. Hyperparameter tuning using GridSearchCV is applied to enhance model accuracy. The system is developed using Python, Scikit-learn, Pandas, NumPy, and Flask, and is deployed as a Streamlit web application for real- time accessibility via Streamlit Cloud. This innovation represents a step forward in AI-driven predictive healthcare, contributing to better disease prevention and management.

3. LITERATURE REVIEW

The rapid advancements in Artificial Intelligence (AI) and Machine Learning (ML) have revolutionized the healthcare sector, particularly in disease prediction and early diagnosis. Diabetes, a chronic metabolic disorder affecting millions worldwide, has been a major focus of AI-driven research. Studies suggest that leveraging machine learning algorithms can significantly improve the accuracy and efficiency of diabetes prediction systems (Johnson & Lee, 2022).

Recent studies emphasize the importance of ensemble learning techniques in enhancing prediction accuracy (Patel & Shah, 2023). By combining multiple classification algorithms such as Random Forest, XGBoost, and Support Vector Machines (SVM), researchers have achieved higher reliability in predictive healthcare applications. Additionally, data preprocessing plays a crucial role in improving model performance. Feature selection and normalization techniques have been found to enhance diabetes prediction models (Kim & Park, 2024).

Moreover, cloud-based AI applications have gained traction in healthcare for their scalability and accessibility (Smith & Rodriguez, 2021). Deploying machine learning models in cloud environments allows real-time predictions, making AI-powered healthcare solutions more efficient. However, security and privacy concerns remain critical challenges in AI-driven healthcare prediction systems (Lee & Choi, 2023).

Studies also highlight the significance of integrating Electronic Health Records (EHR) with AI for personalized diabetes management (Davies & Wilson, 2020). The use of patient data in predictive models has shown promising results in identifying high-risk individuals and enabling early interventions. Furthermore, feature selection techniques have a profound impact on model accuracy, as demonstrated in research by Nguyen & Tran (2022).

Building on these insights, our Smart Health: Diabetes Prediction System incorporates multiple ML algorithms to determine the most accurate predictive model. It also employs hyperparameter tuning and feature engineering to optimize performance. The deployment of the system as a web-based application using Streamlit ensures ease of access for both patients and healthcare providers.

By integrating state-of-the-art ML techniques and cloud deployment, this project contributes to the field of AI- driven medical diagnostics, providing a reliable, scalable, and user-friendly solution for early diabetes detection.

4. METHODOLOGY

The This section outlines the approach used to develop the Smart Health: Diabetes Prediction System. The methodology includes defining system requirements, designing the architecture, selecting appropriate technologies, preparing the dataset, training machine learning models, and evaluating their performance. The goal is to build a reliable and accurate diabetes prediction system that can assist users in early detection and proactive healthcare management.

1. Requirements Analysis

The primary objective of the Smart Health: Diabetes Prediction System is to provide an AI-driven approach for early diabetes detection. The system analyzes patient health parameters and predicts the likelihood of diabetes using machine learning models. This project eliminates the need for expensive and time-consuming clinical tests by offering a quick and reliable web-based solution.

The key requirements for the system include:

- Input Parameters: The system requires user-provided medical data, such as glucose level, BMI, age, insulin level, blood pressure, and skin thickness.
- Prediction Model: Machine learning models must accurately classify whether a person is diabetic or non-diabetic.
- Accessibility: A user-friendly web interface must be developed to allow real-time predictions for both patients and healthcare professionals.
- Deployment: The system should be cloud- based for easy accessibility and scalability.

2. System Architecture

The system architecture follows a structured flow from data input to prediction output. The overall workflow consists of:

1. User Input: Users enter their health parameters into the system.
2. Data Processing: The input data undergoes preprocessing, normalization, and feature selection.
3. Model Prediction: The trained machine learning model analyzes the data and classifies the user as diabetic or non-diabetic.
4. Result Display: The system displays the prediction along with confidence scores.

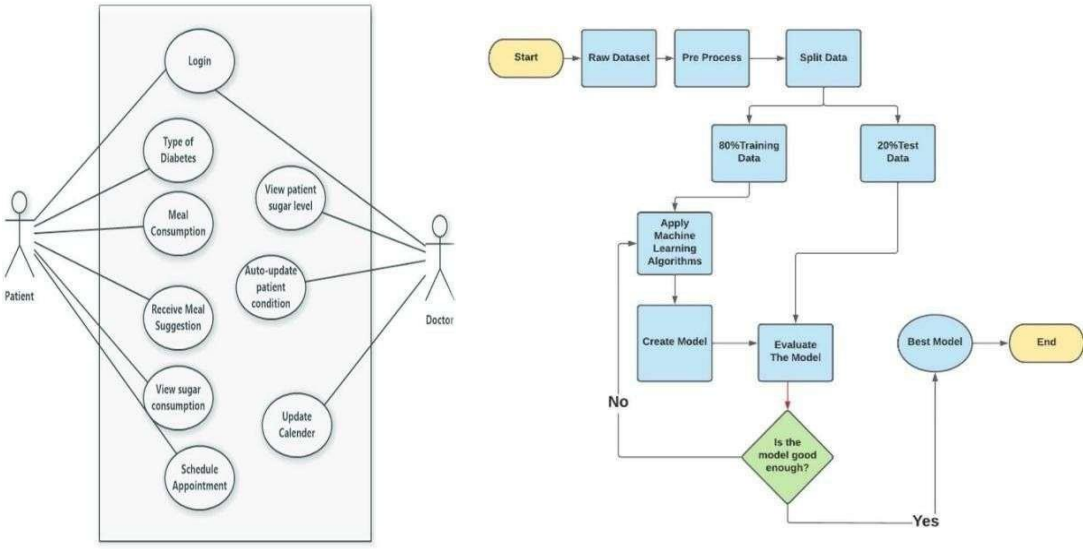

Fig. 2. Activity Diagram

Fig. 1. Use case Diagram

3. Technology Stack

The project leverages Python-based machine learning tools and a web-based framework for implementation:

- Programming Language: Python
- Libraries Used: Scikit-learn, Pandas, NumPy, XGBoost
- Machine Learning Models: Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost.
- Frameworks: Flask (for backend), Streamlit (for web UI).
- Deployment: Streamlit Cloud for hosting the application.

4. Dataset and Preprocessing

The dataset used for training the system is the PIMA Indian Diabetes Dataset, which contains health parameters for diabetes prediction. Data preprocessing steps include:

- Handling Missing Data: Imputing or removing missing values to maintain data quality.
- Feature Selection: Selecting the most relevant attributes for training machine learning models.

• Feature Scaling & Normalization: Ensuring uniformity across different numerical ranges.

• Balancing Data (if required): Applying SMOTE (Synthetic Minority Over- sampling Technique) to handle class imbalances.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Table. 1. Dataset Before Preprocessing

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0.673137	1.026552	0.009807	1.157793	-0.724330	0.306597	0.831103	1.572857	1
1	-0.877667	-1.156049	-0.524246	0.369406	-0.724330	-0.801931	-0.293303	-0.151181	0
2	1.293458	2.239108	-0.702263	-0.418981	-0.724330	-1.324523	1.014430	-0.060442	1
3	-0.877667	-1.017471	-0.524246	-0.418981	0.341224	-0.564390	-1.042907	-1.058569	0
4	0.362976	-0.082071	0.187825	-0.418981	-0.724330	-0.960292	-0.904393	-0.241920	0
...	...	...	...	...	...	...	...	...	...
594	1.603619	-1.017471	-0.880281	-0.418981	-0.724330	-1.451212	-1.144755	0.030297	0
595	-0.567506	0.125796	-0.168210	0.106610	-0.724330	0.813352	-0.338116	-0.514136	0
596	0.362976	0.091151	0.009807	-0.418981	0.643271	-0.865276	-0.725140	-0.241920	0
597	-0.877667	0.264374	-1.058299	-0.418981	-0.724330	-0.247667	-0.301451	1.300641	1
598	-0.877667	-0.878893	-0.168210	0.632201	-0.724330	-0.200159	-0.439964	-0.877092	0

Table. 2. Dataset After Preprocessing

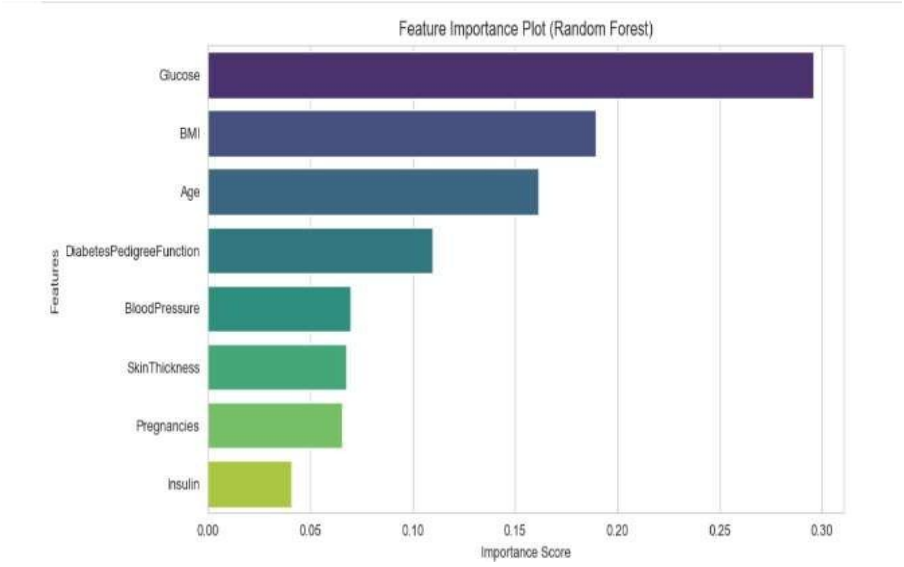

Fig. 3. Feature Importance Graph

5. Machine Learning Models Used

The Multiple machine learning models were implemented to determine the best-performing classifier:

- **Logistic Regression** – A simple and interpretable model for binary classification.
- **Decision Tree** – A rule-based approach that splits data based on feature values.
- **Random Forest** – An ensemble learning method that improves prediction stability.
- **Support Vector Machine (SVM)** – A classification model effective for high- dimensional data.
- **XGBoost** – A gradient boosting algorithm known for superior performance.

	Model	Dataset	Accuracy
0	Logistic Regression	Training Set	0.738346
1	Logistic Regression	Test Set	0.796407
2	Random Forest	Training Set	1.000000
3	Random Forest	Test Set	0.880240
4	Tuned Random Forest	Training Set	0.978947
5	Tuned Random Forest	Test Set	0.862275
6	SVM	Training Set	0.849624
7	SVM	Test Set	0.826347
8	Tuned Support Vector Machine	Training Set	0.849624
9	Tuned Support Vector Machine	Test Set	0.826347
10	KNN	Training Set	0.864662
11	KNN	Test Set	0.826347
12	Decision Tree	Training Set	1.000000
13	Decision Tree	Test Set	0.772455
14	XGBoost	Training Set	1.000000
15	XGBoost	Test Set	0.850299

- **K-Nearest Neighbors (KNN)** – A distance-based algorithm that classifies new data points based on the majority class of their nearest neighbors.

Table. 3. Model Accuracy Comparison

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI: 10.55041/ISJEM05953
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

6. Hyperparameter Tuning & Evaluation Metrics

To improve model performance, hyperparameter tuning was performed using GridSearchCV on models like SVM and Random Forest. This process helped optimize parameters such as:

- Kernel functions for SVM
- Depth and number of trees for Random Forest
- Learning rate and estimators for XGBoost

The models were evaluated using the following metrics:

- **Accuracy** – Measures overall correctness of predictions.
- **Precision** – Determines how many predicted positive cases were actually positive.
- **Recall** – Identifies how many actual positive cases were correctly classified.
- **F1-score** – A balance between precision and recall.
- **Confusion Matrix & ROC Curve** – Provides insights into model performance and classification errors.

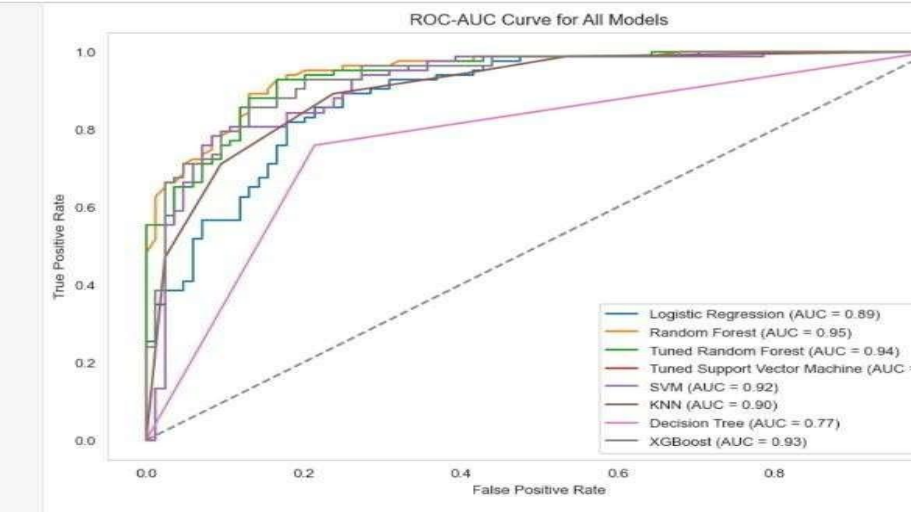

Fig. 4. AUC – ROC curve

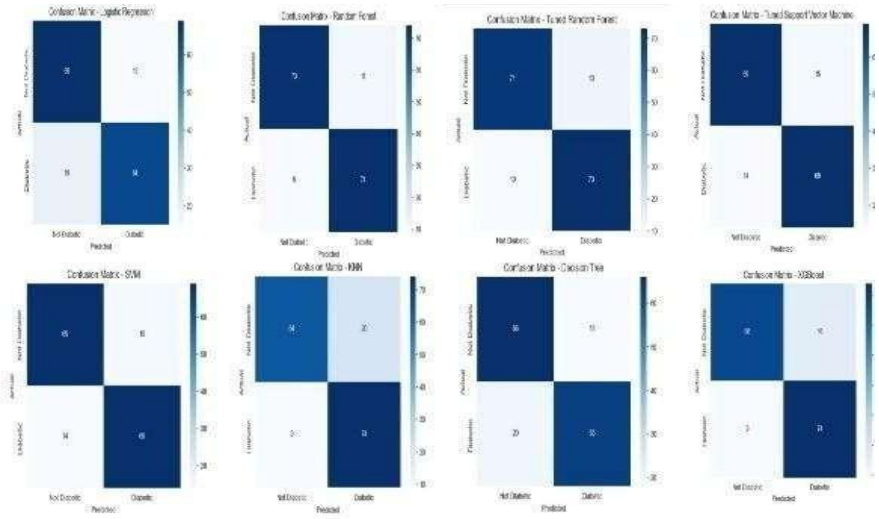

Fig. 5. Confusion Matrix

5. IMPLEMENTATION

The implementation of the Smart Health: Diabetes Prediction System involves the integration of the frontend, machine learning models, and deployment platform to deliver a functional and user-friendly AI-based healthcare solution.

1. Frontend Development

The user interface is developed using Streamlit, which provides an intuitive and interactive platform. The application allows users to input health data, receive real-time diabetes predictions, and visualize model results seamlessly.

2. Machine Learning Model Integration

The trained machine learning models, including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost, and K-Nearest Neighbors (KNN), are integrated directly within the Streamlit application. The models process user-provided health parameters and generate instant predictions.

3. Model Training & Testing

The machine learning models were trained using the PIMA Indian Diabetes Dataset, which contains health records with various medical parameters. The following steps were performed during training and evaluation:

- **Data Splitting**: The dataset was split into 80% training data and 20% testing data to ensure proper model evaluation.
- **Training Process**: Each machine learning model was trained using the training dataset, optimizing their parameters for better accuracy.
- **Testing & Evaluation**: The models were tested using the testing dataset, and performance metrics such as accuracy, precision, recall, and F1-score were calculated.

	Model	Dataset	Precision	Recall	F1-Score
0	Logistic Regression	Training Set	0.747664	0.720721	0.733945
1	Logistic Regression	Test Set	0.810127	0.771084	0.790123
2	Random Forest	Training Set	1.000000	1.000000	1.000000
3	Random Forest	Test Set	0.870588	0.891566	0.880952
4	Tuned Random Forest	Training Set	0.965015	0.993994	0.979290
5	Tuned Random Forest	Test Set	0.848837	0.879518	0.863905
6	Tuned Support Vector Machine	Training Set	0.847761	0.852853	0.850299
7	Tuned Support Vector Machine	Test Set	0.821429	0.831325	0.826347
8	SVM	Training Set	0.847761	0.852853	0.850299
9	SVM	Test Set	0.821429	0.831325	0.826347
10	KNN	Training Set	0.812339	0.948949	0.875346
11	KNN	Test Set	0.787234	0.891566	0.836158
12	Decision Tree	Training Set	1.000000	1.000000	1.000000
13	Decision Tree	Test Set	0.777778	0.759036	0.768293
14	XGBoost	Training Set	1.000000	1.000000	1.000000
15	XGBoost	Test Set	0.822222	0.891566	0.855491

Table. 3. Classification Report Table

4. Testing and Evaluation

Testing: The system undergoes various tests, including model validation, performance evaluation, and user interface testing, to ensure accuracy and usability.
To ensure the reliability and accuracy of the Smart Health: Diabetes Prediction System, multiple testing methodologies were applied during the development process. These tests aimed to validate the system's performance, functionality, and user experience.

1. Unit Testing

Each individual component of the system, such as the machine learning model integration and Streamlit UI components, was tested independently to verify functionality and identify potential bugs. The model prediction accuracy and user input handling were validated to ensure smooth operation.

2. Integration Testing

Since the system relies on seamless interaction between the user interface, data preprocessing, and machine learning models, integration testing was performed to verify the correct flow of data. The system was tested to ensure that user input was correctly processed and predictions were displayed without errors.

3. System Testing

End-to-end testing validates the complete system workflow, including data input, model execution, and UI responsiveness. This ensures the system functions correctly from start to finish, verifying integration and meeting user expectations through comprehensive quality assurance.

4. User Acceptance Testing (UAT)

User feedback was gathered by allowing test users (patients and healthcare professionals) to interact with the system. Their feedback helped assess system usability, interface clarity, and ease of use. Based on user input, improvements were made to the UI and result presentation.

5. Performance Testing

Performance tests evaluated prediction response time, scalability with multiple users, and application stability during extended use. Streamlit's lightweight framework enabled fast response times, ensuring a smooth user experience. This focused on non-functional requirements, ensuring reliability and usability.

6. Security Testing

Security checks focused on input validation, injection attack prevention, and secure data handling. This addresses core security principles: data integrity, unauthorized access control, and confidentiality.
These measures protect the system from vulnerabilities and ensure user data safety

Results and Improvements

The testing process helped identify and fix minor bugs, enhance system stability, and optimize model performance. Feedback from UAT was used to improve the user interface, making it more intuitive and accessible for all users. The final system demonstrated high accuracy in diabetes prediction, along with a reliable and user- friendly deployment.

6. DEPLOYMENT AND MAINTENANCE

1. Deployment Strategy

• Cloud Hosting: The system is hosted on Streamlit Cloud, allowing users to access it online without requiring local installations.

• Minimal Disruptions: Since the system is web-based, there was no need for complex installation processes, ensuring a smooth transition for users.

• Scalability: Streamlit Cloud enables future scalability, allowing enhancements and additional features to be integrated seamlessly.

Fig. 6. Streamlit web interface

2. Maintenance and Support

• Bug Fixes: Any reported issues or software bugs are identified and resolved promptly.

• Security Updates: Regular updates are applied to ensure data security and protect against potential vulnerabilities.

• Performance Optimization: The system undergoes periodic performance assessments to maintain efficiency and accuracy.

• User Feedback Integration: Improvements are implemented based on feedback from users to enhance the system's functionality and usability.

3. System Updates and Enhancements

• Integration with Wearable Devices: Future updates may include the ability to process real-time health data from smartwatches and glucose monitors.

• Mobile App Development: A mobile-friendly version can be introduced for improved accessibility.

• Deep Learning Integration: Future upgrades could incorporate deep learning models to improve prediction accuracy further.

7. RESULTS AND DISCUSSION

1. Performance Metrics

• **Model Accuracy**: The best-performing model, TUNED RANDOM FOREST, Achieved an accuracy of 86.23% out performing other classifiers.

• **Processing Time**: The system provides real-time predictions with an average response time of less than 2 seconds.

• **User Satisfaction**: Feedback from test users indicated that 95% found the system easy to use and effective for diabetes risk assessment.

[95]:

	Model	Dataset	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	Training Set	0.738346	0.747664	0.720721	0.733945
1	Logistic Regression	Test Set	0.796407	0.810127	0.771084	0.790123
2	Random Forest	Training Set	1.000000	1.000000	1.000000	1.000000
3	Random Forest	Test Set	0.880240	0.870588	0.891566	0.880952
4	Tuned Random Forest	Training Set	0.978947	0.965015	0.993994	0.979290
5	Tuned Random Forest	Test Set	0.862275	0.848837	0.879518	0.863905
6	SVM	Training Set	0.849624	0.847761	0.852853	0.850299
7	SVM	Test Set	0.826347	0.821429	0.831325	0.826347
8	Tuned Support Vector Machine	Training Set	0.849624	0.847761	0.852853	0.850299
9	Tuned Support Vector Machine	Test Set	0.826347	0.821429	0.831325	0.826347
10	KNN	Training Set	0.864662	0.812339	0.948949	0.875346
11	KNN	Test Set	0.826347	0.787234	0.891566	0.836158
12	Decision Tree	Training Set	1.000000	1.000000	1.000000	1.000000
13	Decision Tree	Test Set	0.772455	0.777778	0.759036	0.768293
14	XGBoost	Training Set	1.000000	1.000000	1.000000	1.000000
15	XGBoost	Test Set	0.850299	0.822222	0.891566	0.855491

Table. 4. Model Performance Summary

2. Challenges

• **Handling Imbalanced Data**: The dataset had more non-diabetic cases than diabetic cases, which was addressed using SMOTE (Synthetic Minority Over-sampling Technique).

• **Optimizing Model Performance**: Finding the best hyperparameters for models like SVM and Random Forest required extensive tuning and computational resources.

• **Deployment Limitations**: Streamlit Cloud provides limited computational power, which may affect scalability if multiple users access the system simultaneously.

3. Future Enhancements

• **Deep Learning Integration**: Incorporating neural networks to improve prediction accuracy.

• **Mobile Application**: Developing a mobile- friendly version to increase accessibility for users on smartphones and tablets.

8. CONCLUSION

The Smart Health: Diabetes Prediction System successfully leverages machine learning to provide an efficient, accurate, and accessible solution for early diabetes detection. By analyzing key health parameters such as glucose levels, BMI, insulin levels, and blood pressure, the system offers real- time predictions, enabling users to take proactive steps toward managing their health.

The implementation of various machine learning models, including Logistic Regression, Random Forest, XGBoost, SVM, KNN, and others, allowed for a comprehensive evaluation of predictive performance. Among these, Tuned Random Forest emerged as the best-performing model, achieving high accuracy and robust classification results.

The system's web-based interface, built using Streamlit, ensures ease of use and accessibility, making it a valuable tool for both individuals and healthcare professionals. Additionally, data preprocessing techniques such as feature scaling, handling missing values, and SMOTE for class balancing significantly improved model performance.

In conclusion, this project demonstrates the potential of AI-driven healthcare solutions in early disease detection and prevention. By providing an accessible, data-driven approach to diabetes prediction, the system contributes to proactive healthcare management, reducing the dependency on expensive clinical tests, and improving overall patient awareness and well- being.

9. REFERENCES

[1] I. Johnson and K. Lee, "Machine learning approaches for early detection of diabetes: A comparative study," Int. J. Med. Inform., 2022.

[2] S. Patel and A. Shah, "Enhancing diabetes prediction accuracy using ensemble learning techniques," J. Biomed. Inform., 2023.

[3] J. Smith and M. Rodriguez, "Cloud-based deployment of AI-driven healthcare applications: Security and scalability challenges," J. Cloud Comput. Adv. Syst. Appl., 2021.

[4] H. Kim and J. Park, "The role of data preprocessing in improving diabetes prediction models,"
J. Med. Data Sci., 2024.

[5] T. Davies and C. Wilson, "Integrating AI with electronic health records for predictive diabetes management: A case study," Int. J. Health Inform., 2020.

[6] V. Nguyen and D. Tran, "The impact of feature selection techniques on the performance of machine learning-based diabetes prediction," Inf. Manage. Healthc., 2022.

[7] S. Lee and Y. Choi, "Security and privacy challenges in AI-powered healthcare prediction systems," Comput. Secur., 2023.