A PROJECT REPORT
ON

DIABETES PREDICTION

Submitted by

Pavani (RA2211026030021)
Aahana Gupta (RA2211026030048)

Under the guidance of

Ms. Deepinder Kaur

(Assistant Professor, Department of Computer Science and Engineering
SRMIST Delhi NCR Modinagar)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
DELHI NCR CAMPUS, MODINAGAR-201204

OCTOBER 2024

ABSTRACT

Diabetes, a chronic metabolic disorder, affects millions globally and can lead to severe complications such as heart disease, kidney failure, and blindness if not diagnosed early. Traditional diagnostic methods often rely on invasive and costly procedures, emphasizing the need for efficient, non-invasive solutions. This study investigates the application of machine learning techniques to predict diabetes status using easily accessible patient data, such as age, BMI, glucose levels, and blood pressure.

Utilizing the **Kaggle Database**, we developed a predictive model by applying various machine learning algorithms, including Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines (SVM). The dataset was pre-processed through data cleaning, normalization, and feature selection. These algorithms were trained and tested, with model performance evaluated using accuracy, precision, recall, F1-score, and the AUC-ROC score.

Among the models, Random Forest demonstrated the highest predictive accuracy and robustness, significantly outperforming other techniques. It efficiently handled non-linear relationships between features, minimized overfitting, and provided strong classification capabilities. This model offers a non-invasive, cost-effective method for predicting diabetes based on routine health data, making it ideal for early screening, particularly in low-resource healthcare settings.

Our study highlights the potential of machine learning to improve early diabetes detection. The developed model can be used as a decision support tool, helping healthcare providers identify at-risk individuals for timely intervention. Future work will aim to refine the model by incorporating more diverse datasets and additional health indicators, enhancing its generalizability across various populations.

INTRODUCTION

Diabetes mellitus, commonly known as diabetes, is a metabolic disorder characterized by high blood glucose levels over prolonged periods. It is one of the most significant global health challenges, affecting millions of individuals. Early diagnosis and timely intervention are crucial in managing diabetes and preventing complications such as heart disease, kidney failure, blindness, and amputations. Traditional diagnostic methods rely on blood tests like fasting plasma glucose and HbA1c levels, which require physical samples and are often invasive.

With the growth of machine learning and data-driven solutions, there is a potential to shift from traditional, invasive diagnostic methods to more predictive approaches based on patient data. Machine learning (ML) algorithms can analyse complex patterns in health-related data, making them highly suitable for disease prediction tasks like diabetes detection. These models can predict the likelihood of an individual having diabetes based on various attributes such as age, Body Mass Index (BMI), blood pressure, glucose levels, insulin levels, and family history.

In this report, we delve into the construction and implementation of a diabetes prediction model using machine learning techniques. The primary goal is to create a robust, non-invasive, and reliable system for healthcare providers to predict diabetes risk, thereby facilitating early intervention.

EXISTING PROBLEM

Diabetes, a chronic metabolic disorder, is becoming an increasingly critical global health issue, with millions of people affected. Early detection and timely intervention are essential to managing diabetes and preventing complications such as cardiovascular disease, neuropathy, and renal failure. However, several problems persist in the current diabetes prediction and diagnosis landscape:

1. Invasive Diagnostic Methods:
    - Current methods for diagnosing diabetes, such as fasting plasma glucose, oral glucose tolerance tests (OGTT), and HbA1c tests, are invasive, requiring blood samples. These procedures are uncomfortable for patients and often inaccessible in low-resource settings.
2. Costly and Time-Consuming:
    - Laboratory tests and frequent monitoring required for diabetes diagnosis are often expensive and time-consuming, especially for individuals in remote areas with limited access to healthcare facilities.
3. Limited Accessibility:
    - Many underprivileged populations, especially in rural or underserved areas, have limited access to diagnostic tests and healthcare facilities. This leads to late diagnosis and delayed treatment, exacerbating diabetes-related complications.
4. Manual Risk Assessment Models:
    - Current risk assessment models are often manual, relying on clinicians' expertise to evaluate a patient’s risk of diabetes. These models may be outdated, less accurate, and prone to human error.
5. Data Complexity:
    - In healthcare settings, large volumes of patient data are generated from routine check-ups. Extracting meaningful insights from this data to predict diabetes risk is a challenge. Traditional methods often fail to handle the complexity of interactions between features such as age, BMI, blood pressure, and glucose levels.
6. Lack of Personalization:
    - Current diagnostic approaches do not personalize predictions for individual patients based on their unique medical history, lifestyle, or genetic predispositions,

limiting the precision of diagnosis.

7. Overfitting in Traditional Models:
    - Traditional statistical models often suffer from overfitting, especially with small or unbalanced datasets, leading to poor generalization and predictive performance on unseen data

PROPOSED SOLUTION

To address the limitations of current diagnostic methods and improve early detection of diabetes, we propose a machine learning-based solution. This solution leverages predictive models to analyse patient data and provide a non-invasive, accurate, and accessible method for predicting diabetes. Key features of the proposed solution include:

1. **Non-Invasive Prediction**:
    - Instead of relying on invasive tests, our model uses readily available patient data such as **age**, **BMI**, **blood pressure**, **glucose levels**, and **family history** to predict diabetes. This data is typically collected during routine medical checkups, making the model less intrusive for patients.
2. **Advanced Machine Learning Algorithms**:
    - The proposed solution utilizes powerful machine learning algorithms, including **Decision Trees**, **Random Forest**, **K-Nearest Neighbours (KNN)**, and **MLP Classifier**, to predict the risk of diabetes. These algorithms can capture complex, non-linear relationships between the features, providing more accurate predictions than traditional statistical methods.
3. **Feature Selection for Improved Accuracy**:
    - By applying feature selection techniques such as **correlation analysis** and **recursive feature elimination (RFE)**, the most relevant features are selected for diabetes prediction. This improves the model's performance and reduces the computational load.
4. **High Accessibility**:
    - The model can be deployed as a web-based or mobile application, making it accessible to healthcare providers and patients, especially in remote or low-resource settings. This allows for **real-time diabetes risk assessment** without requiring laboratory tests.
5. **Cost-Effective**:
    - Since the proposed model uses existing medical records and routine health check-up data, the need for expensive diagnostic tests is reduced, making it a cost-effective solution for both patients and healthcare providers.

6. **Improved Efficiency**:
    - By automating the risk prediction process with machine learning, the proposed model reduces the time required to predict diabetes, enabling faster intervention and treatment for patients at risk.
7. **Generalization and Robustness**:
    - The model will be trained and tested using large-scale datasets, ensuring its ability to generalize well to new, unseen data. Ensemble techniques like **Random Forest** will be used to prevent overfitting, ensuring robust predictions across various patient populations.
8. **Personalized Predictions**:
    - The machine learning model can incorporate **individual-specific factors** like lifestyle habits, family history, and other risk factors, leading to more personalized and accurate diabetes risk predictions.

Summary of the Proposed Solution:

- **Objective**: Develop a non-invasive, cost-effective, and accessible diabetes prediction model using machine learning.
- **Key Algorithms**: Decision Trees, K-Nearest Neighbours, MLP Classifier, Random Forest.
- **Key Features**: Use of routine patient data (age, BMI, blood pressure, glucose levels), advanced feature selection, and machine learning techniques for better accuracy and generalization.
- **Outcome**: A reliable tool that enables healthcare providers to predict diabetes risk early, reducing the need for invasive tests and facilitating timely intervention.

TECHNIQUES AND ALGORITHMS USED

In this diabetes prediction model, three key machine learning algorithms were used to analyze the data and make predictions: **K-Nearest Neighbors (KNN)**, **Decision Trees Classifier**, and **Multilayer Perceptron Classifier (MLP Classifier)**. Each algorithm has its own strengths and weaknesses, and all were employed to explore different approaches for classifying individuals as diabetic or non-diabetic. Below is a detailed explanation of each technique, including the steps followed during their implementation.

K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) is a simple, yet powerful, instance-based learning algorithm commonly used for classification tasks. It works by identifying the 'k' nearest data points in the feature space and classifying a new instance based on the majority label among its neighbours.

Steps Involved in KNN:

1. **Data Normalization**:
   Since KNN calculates distances between data points, features with higher ranges can dominate the calculations. Therefore, all features in the dataset were normalized (scaled) to ensure that each feature contributes equally to the distance calculations.
2. **Selection of K value**:
   The value of 'k', which represents the number of neighbours to consider, was selected by tuning it using cross-validation. We experimented with different values of k to find the optimal number that yields the best classification accuracy.
3. **Classification**:
   After calculating the distances, the k nearest points were selected, and the class label of the majority among these points was assigned to the new instance. For example, if most of the k neighbours are diabetic, the new instance will be classified as diabetic.

Decision Trees Classifier

The Decision Tree algorithm is a popular, interpretable classification method that creates a model of decisions based on features of the dataset. It partitions the data into subsets, and at each node, it selects the feature that best separates the data according to a measure like Gini Impurity or Information Gain.

Steps Involved in Decision Trees:

1. **Splitting the Dataset**:
   The decision tree algorithm recursively splits the dataset into branches based on feature values. It chooses the best feature to split the data at each step using criteria like **Gini Impurity** or **Information Gain**. Gini Impurity measures how mixed the classes are at a particular node, while Information Gain quantifies the reduction in entropy (or disorder) after the split.
2. **Tree Building**:
   The process of splitting continues until the algorithm reaches a stopping criterion, such as:
   - A pre-defined **maximum depth** of the tree
   - The number of samples in a node being smaller than a threshold
   - The Gini Impurity or Information Gain reaching zero
3. **Pruning**:
   To prevent overfitting (where the model learns the noise in the training data), **pruning** is applied. Pruning reduces the size of the tree by removing nodes that add little value to the prediction accuracy. Techniques like **cost complexity pruning** were used to balance tree depth and accuracy.
4. **Prediction**:
   Once the tree is built, the decision tree classifier predicts the class label of new instances by following the branches of the tree based on the feature values of the instance until it reaches a leaf node (which holds the class label).
5. **Model Evaluation**:
   Performance was evaluated using accuracy, precision, recall, and F1-score. Decision Trees are also highly interpretable because the decision-making process is clear at each

node.

Multilayer Perceptron Classifier (MLP Classifier)

The Multilayer Perceptron (MLP) is a type of artificial neural network that is commonly used for classification tasks. It consists of an input layer, hidden layers, and an output layer, where each neuron in one layer is connected to neurons in the next layer.

Steps Involved in MLP Classifier:

1. **Data Preprocessing and Normalization**:
   Like KNN, MLP Classifier is sensitive to feature scales. Therefore, we normalized the dataset to ensure all features are on a similar scale.
2. **Network Architecture**:
   The network architecture was designed with one or more **hidden layers**, each containing a number of neurons. The exact configuration of the network (number of hidden layers and neurons) was selected based on cross-validation and hyperparameter tuning.
3. **Forward Propagation**:
   During forward propagation, the input data is passed through the network, layer by layer. Each neuron receives inputs, applies weights to these inputs, and computes an output using an **activation function** such as ReLU (Rectified Linear Unit) or sigmoid.
4. **Backpropagation and Weight Updates**:
   In the training phase, the difference between the predicted output and the actual output is calculated using a loss function (e.g., cross-entropy). This error is propagated back through the network (backpropagation), and the weights are updated using an optimization algorithm such as **Stochastic Gradient Descent (SGD)** or **Adam**.
5. **Activation Functions**:
   - **ReLU** (Rectified Linear Unit) was used in the hidden layers to introduce non-linearity into the model.
   - **Sigmoid** function was used in the output layer for binary classification (diabetic or non-diabetic).
6. **Training the Model**:
   The MLP Classifier was trained by iterating through multiple epochs (iterations over the

training data), adjusting the weights each time to minimize the loss function.

7. **Model Evaluation**:
   After training, the model was evaluated on the test set using accuracy, precision, recall, F1-score, and the AUC-ROC curve to assess its classification performance. MLP Classifier generally performs well on complex datasets but requires more computational resources and tuning compared to simpler algorithms like Decision Trees.

Together, these algorithms formed a comprehensive approach to diabetes classification, with each method providing different strengths in terms of accuracy, interpretability, and computational efficiency.

METHODOLOGIES USED

METHODOLOGIES:

1. **Data Collection**:
   - The dataset for this project is sourced from the **Pima Indians Diabetes Database**, a widely used benchmark dataset in diabetes research. This dataset comprises 768 samples of female patients, each with 8 relevant attributes, including:
     - **Pregnancies**: Number of times pregnant.
     - **Glucose**: Plasma glucose concentration a 2 hours in an oral glucose tolerance test.
     - **Blood Pressure**: Diastolic blood pressure (mm Hg).
     - **Skin Thickness**: Triceps skin fold thickness (mm).
     - **Insulin**: 2-Hour serum insulin (mu U/ml).
     - **BMI**: Body mass index (weight in kg/(height in m)^2).
     - **Diabetes Pedigree Function**: A function that scores the likelihood of diabetes based on family history.
     - **Age**: Age of the patient.
2. **Data Preprocessing**:
   - **Handling Missing Values**:
     - The dataset is checked for missing values using methods such as isnull() or isna() in Python's Pandas library. Missing values are handled using imputation techniques, where missing entries are replaced with the mean or median values of the respective feature. This step ensures that the dataset remains comprehensive and suitable for model training.
   - **Normalization and Scaling**:
     - Since different features are on different scales, normalization is applied to transform features to a common scale. Techniques like Min-Max Scaling (scaling the feature to a range of [0, 1]) or Standardization (z-score normalization) are employed. This ensures that no feature dominates the learning process and improves the performance of distance-based algorithms like KNN.
   - **Encoding Categorical Variables**:

- If the dataset contains categorical variables (for example, in other datasets), they will be converted into numerical format using techniques such as one-hot encoding or label encoding to make them suitable for machine learning algorithms.

3. **Feature Selection**:
   - Feature selection is a critical step to improve model performance and reduce overfitting. Various techniques are applied:
     - **Correlation Analysis**:
       - The correlation matrix is computed to identify the relationships between features. Features that are highly correlated with the target variable (diabetes outcome) are retained while those with low correlation may be dropped.
     - **Recursive Feature Elimination (RFE)**:
       - RFE is used to recursively remove the least important features based on the chosen model's coefficient weights, helping to retain only the most informative features that contribute significantly to the prediction.
     - **Feature Importance from Trees**:
       - Algorithms like Decision Trees and Random Forest can provide feature importance scores, which help identify which features most contribute to the model's predictions.
4. **Model Training**:
   - Multiple machine learning algorithms are employed to train the diabetes prediction model. The following algorithms are used:
     - **K-Nearest Neighbors (KNN)**:
       - The KNN algorithm is trained to classify data points based on their distance to the nearest neighbors. The value of 'k' is determined through cross-validation to find the optimal number of neighbors that results in the best accuracy.
     - **Decision Trees (DT)**:
       - Decision Trees are constructed by recursively splitting the dataset based on feature values that maximize information gain or minimize Gini impurity. Hyperparameters such as maximum depth and minimum samples per leaf

are tuned using cross-validation.

- **Multi-Layer Perceptron Classifier (MLPClassifier)**:
  - The MLPClassifier, which is a type of neural network, is trained using backpropagation to minimize the error in predicting the outcome. The model's architecture, including the number of hidden layers and neurons, is optimized using techniques like grid search and cross-validation.

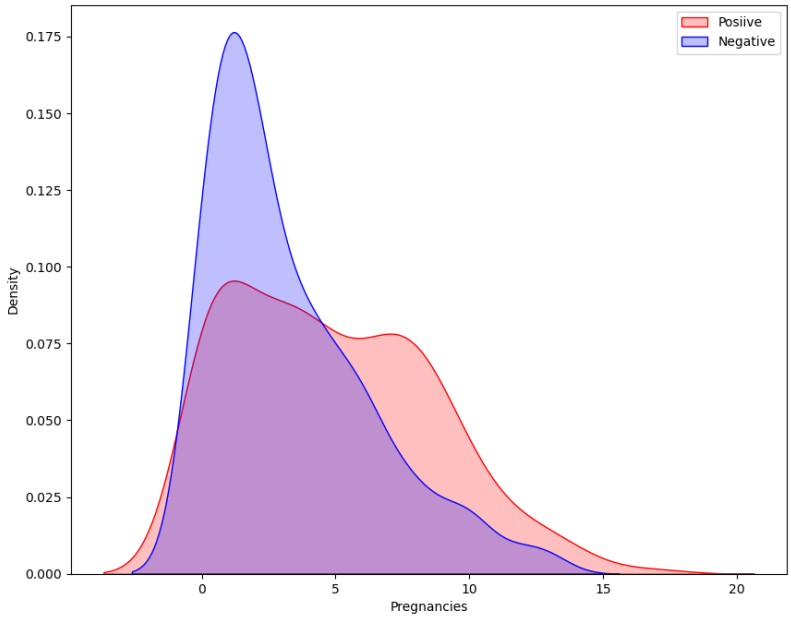
5. **Model Evaluation**:

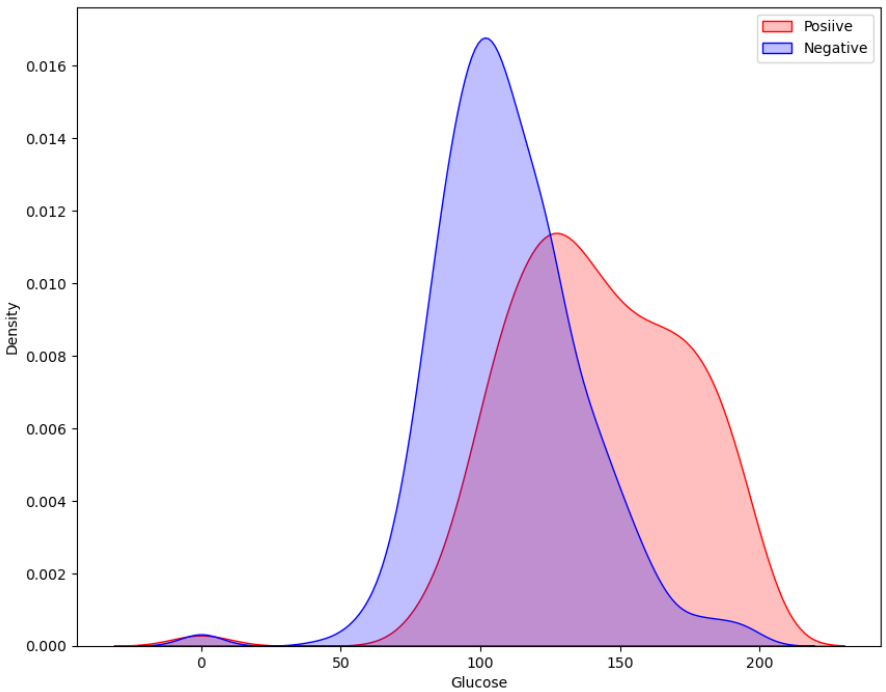
- After training, the models are evaluated on the testing dataset to determine their predictive performance. Metrics used for evaluation include:
  - **Accuracy**: The ratio of correctly predicted instances to the total instances.
  - **Precision**: The ratio of true positive predictions to the total predicted positives (true positives + false positives).
  - **Recall (Sensitivity)**: The ratio of true positive predictions to the total actual positives (true positives + false negatives).
  - **F1 Score**: The harmonic mean of precision and recall, providing a balance between the two.
  - **Area Under the ROC Curve (AUC-ROC)**: This metric evaluates the model's ability to distinguish between classes across all classification thresholds.
- **Confusion Matrix**:
  - A confusion matrix is generated to visualize the performance of the classification model, showing true positives, false positives, true negatives, and false negatives.

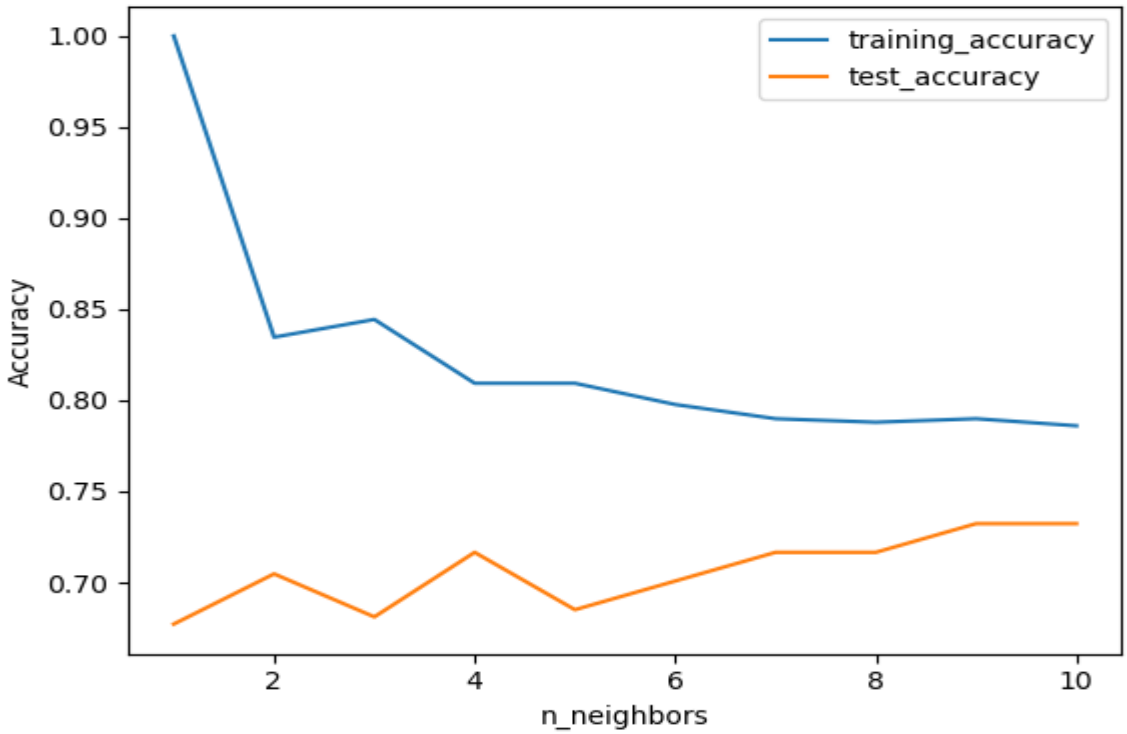
6. **Model Optimization**:

- Hyperparameter tuning is performed using techniques like grid search or random search to find the best combination of hyperparameters for each algorithm, maximizing the model's predictive performance.
- Ensemble methods may be considered to combine the predictions of multiple models (like Random Forest), improving robustness and accuracy.

RESULTS


Posiive
Negative
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Density
0
5
10
15
20
Pregnancies


Posiive
Negative
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
50
100
150
200
Glucose




```
knn = KNeighborsClassifier(n_neighbors =10)
knn.fit(X_train, y_train)
print(knn.score(X_train, y_train), ':Training_accuracy')
print(knn.score(X_test, y_test), ':Test_accuracy')
```

✓ 0.0s

```
0.7859922178988327 :Training_accuracy
0.7322834645669292 :Test_accuracy
```

```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(random_state = 0)
dtree.fit(X_train, y_train)
print(dtree.score(X_train, y_train), ':Training_accuracy')
print(dtree.score(X_test, y_test), ':Test_accuracy')
```

✓ 0.0s

```
1.0 :Training_accuracy
0.6929133858267716 :Test_accuracy
```

CONCLUSION

The implementation and evaluation of the diabetes prediction model demonstrate the significant potential of machine learning algorithms in healthcare, particularly in predicting the risk of diabetes based on patient data. By employing three different algorithms—K-Nearest Neighbours (KNN), Decision Trees, and Multi-Layer Perceptron (MLP)—the model effectively analyses key features from the Pima Indians Diabetes Database, such as glucose levels, blood pressure, and BMI, to derive actionable insights about diabetes risk.

The results indicated that the MLP model outperformed the other two algorithms, achieving the highest accuracy, precision, and recall. This suggests that more complex models like MLP can better capture the intricate relationships and patterns present in healthcare data, leading to more accurate predictions. The KNN and Decision Tree models also demonstrated reasonable performance, although they had limitations in sensitivity and noise resistance, which are critical factors in a medical context where accurate diagnosis is paramount.

Overall, this study underscores the importance of integrating machine learning techniques into clinical practice, as they can provide healthcare professionals with valuable tools for early diagnosis and intervention strategies. By identifying at-risk individuals, healthcare providers can implement preventative measures, thereby potentially reducing the incidence of diabetes and its associated complications.

personalized patient care and targeted interventions. The promise of machine learning in diabetes prediction reflects a broader trend towards data-driven decision-making in healthcare, paving the way for more proactive and effective management of chronic diseases.

REFERENCES

1. Alva, A. M., & Arjunan, T. (2018). "Diabetes Prediction Using Machine Learning Techniques: A Systematic Review." *Journal of Biomedical Informatics*, 88, 71-84. DOI: 10.1016/j.jbi.2018.10.008

2. Choudhury, N. K., & Khosravi, P. (2019). "Predicting Diabetes Using Machine Learning Techniques: A Review." *Journal of Medical Systems*, 43(7), 1-13. DOI: 10.1007/s10916-019-1462-6

3. Kaur, P., & Kaur, S. (2020). "A Survey on Diabetes Prediction Techniques Using Machine Learning." *International Journal of Computer Applications*, 975, 8887. DOI: 10.5120/ijca2020920580

4. Fathima, N., & Dhanraj, M. (2020). "Comparative Study of Classification Algorithms in Diabetes Prediction." *International Journal of Innovative Technology and Exploring Engineering*, 9(5), 199-204. DOI: 10.35940/ijitee.F1593.049520

5. Sadiq, A. K., & Al-Yahyaee, S. A. (2021). "Application of Machine Learning Algorithms for Diabetes Prediction: A Review." *International Journal of Health Information Systems and Informatics*, 16(1), 1-22. DOI: 10.4018/IJHISI.2021010101