CIS 550-7101: Advanced Machine Learning - On-Line

(2025 Spring CE1)

**Optimizing Early Detection of Diabetes Through ML Pipelines**

**Group – 14**

**(10) Theopilus Gudapati – 02147220**

**(34) Charish Yadavali - 02151348**

Table of Contents

[I. Project Summary 3](#_Toc196492939)

[A. Problem Statement: 3](#_Toc196492940)

[B. Data Collection: 3](#_Toc196492941)

[C. Data Overview: 4](#_Toc196492942)

[D. Data Description: 4](#_Toc196492943)

[II. Methodology 5](#_Toc196492944)

[A. Data Preprocessing: 5](#_Toc196492945)

[B. Model Selection: 6](#_Toc196492946)

[1. K-Nearest Neighbors (KNN) Classifier: 6](#_Toc196492947)

[2. Random Forest Classifier: 6](#_Toc196492948)

[3. Additional Models Considered: 7](#_Toc196492949)

[C. Model Evaluation: 7](#_Toc196492950)

[1. Challenges with Imbalanced Data: 7](#_Toc196492951)

[2. Metrics Used for Evaluation: 8](#_Toc196492952)

[III. Results 9](#_Toc196492953)

[A. Model Performance: 9](#_Toc196492954)

[B. Confusion Matrix: 9](#_Toc196492955)

[C. ROC Curve: 9](#_Toc196492956)

[IV. Visualizations 10](#_Toc196492957)

[A. Distribution of Diabetes Outcome: 11](#_Toc196492958)

[B. Correlation Heatmap: 11](#_Toc196492959)

[C. Boxplots for Key Features: 12](#_Toc196492960)

[V. Implications 13](#_Toc196492961)

[VI. Limitations and Future Work 13](#_Toc196492962)

[A. Data Imbalance: 13](#_Toc196492963)

[B. Feature Engineering: 13](#_Toc196492964)

[C. Deep Learning Models: 13](#_Toc196492965)

[VII. Conclusion 13](#_Toc196492966)

[VIII. References 14](#_Toc196492967)

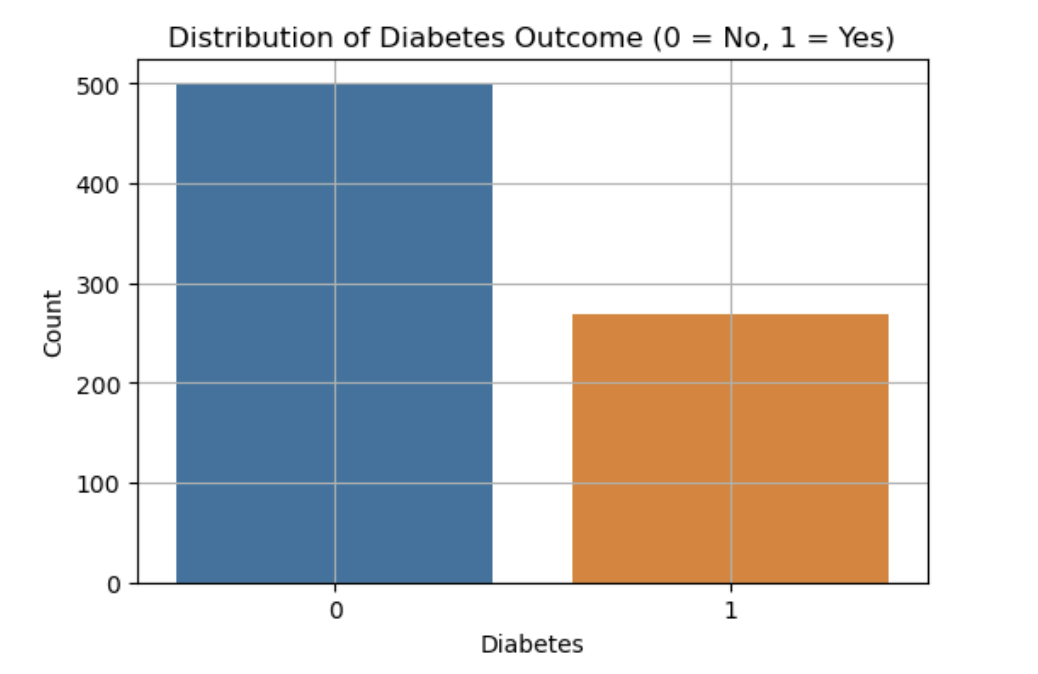
# Project Summary

## Problem Statement:

Diabetes is one of the leading chronic diseases globally, and early detection plays a crucial role in preventing severe complications. Early prediction can help in timely interventions, reducing healthcare costs, and improving the quality of life. The **Pima Indians Diabetes Database** provides valuable data that can be used to predict the likelihood of diabetes in individuals based on medical features such as BMI, glucose levels, and age. This project applies machine learning algorithms to develop a model capable of predicting whether an individual is likely to develop diabetes, helping healthcare providers identify at-risk patients and take preventive actions.

## Data Collection:

The dataset for this project is taken from the **Pima Indians Diabetes Database**, available on Kaggle. It contains **768 records** of female patients, aged 21 or older, from the Pima Indian heritage group. Each record includes 8 medical features and one binary outcome variable that indicates whether the individual has diabetes (1) or not (0). This dataset provides information about medical history, such as the number of pregnancies, glucose levels, insulin levels, BMI, age, and more.

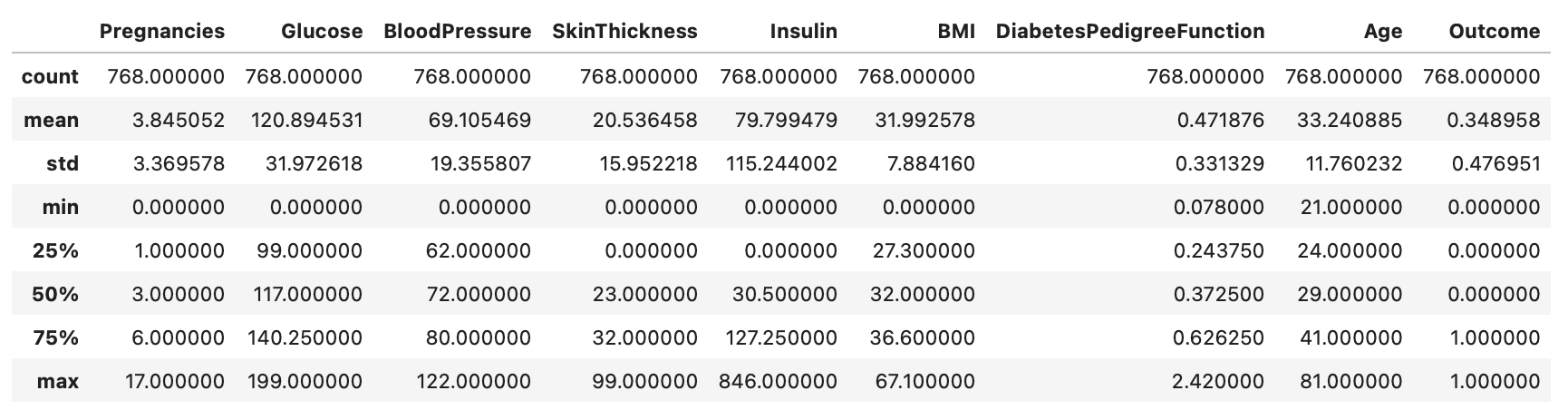


## Data Overview:

* The dataset consists of **768 rows** and **9 columns**.
* **Outcome** is the target variable (1 = Diabetic, 0 = Non-diabetic).
* Features include Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age.

## Data Description:

* **Pregnancies**: Number of pregnancies.
* **Glucose**: Plasma glucose concentration in the blood after a 2-hour oral glucose tolerance test.
* **BloodPressure**: Diastolic blood pressure (mm Hg).
* **SkinThickness**: Triceps skinfold thickness (mm).
* **Insulin**: 2-Hour serum insulin (mu U/ml).
* **BMI**: Body Mass Index (weight in kg / (height in m) ^2).
* **DiabetesPedigreeFunction**: A function that indicates the likelihood of diabetes based on family history.
* **Age**: Age in years.
* **Outcome**: Target variable where 1 indicates a diabetic patient and 0 indicates a non-diabetic patient.



# Methodology

## Data Preprocessing:

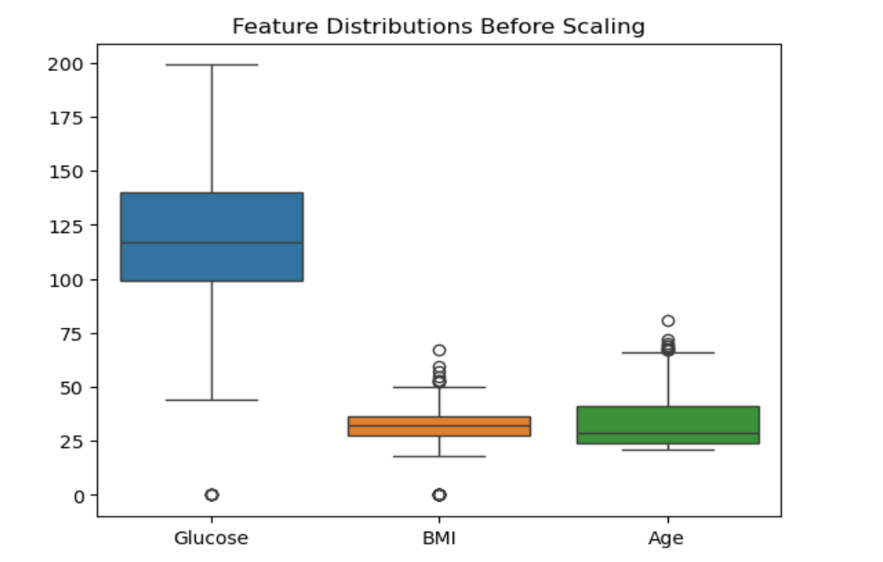
Preprocessing is a critical step in machine learning to ensure the data is clean and suitable for model building.

1. **Handling Missing Values:**

The dataset contained zero values in certain columns (Glucose, BloodPressure, SkinThickness, Insulin, BMI) that were likely errors or missing data. These zeros were replaced with NaN (Not a Number) and imputed using the median of each respective column. This ensures that the data is filled with reasonable values while avoiding skewed results due to zero values.

1. **Feature Scaling:**

Given the presence of features with different scales (e.g., Age vs. Glucose), all features were scaled using **StandardScaler** to standardize the data. This step ensures that each feature contributes equally to the model's performance by transforming them to a common scale with a mean of 0 and a standard deviation of 1.



1. **Train-Test Split:**

The dataset was split into training and testing sets using an **80/20 split**. This allows us to train the models on the training set and evaluate them on the unseen test set, simulating real-world conditions.

1. **Class Imbalance Handling:**

The dataset had a class imbalance, with significantly more non-diabetic cases (0) than diabetic cases (1). To address this, techniques like **SMOTE (Synthetic Minority Over-sampling Technique)** were used to generate synthetic examples of the minority class (diabetic) to balance the dataset and ensure fair training.

## Model Selection:

In this section, we describe the machine learning classification models used to predict the likelihood of diabetes in individuals based on various health-related features. We experimented with several models to evaluate their performance in predicting diabetes, using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC.

### K-Nearest Neighbors (KNN) Classifier:

The KNN classifier is a simple and intuitive machine learning algorithm. It classifies a data point based on the majority class among its neighbors. We experimented with a range of **k** values (from 1 to 49) to find the optimal number of neighbors for classification.

**Performance**:

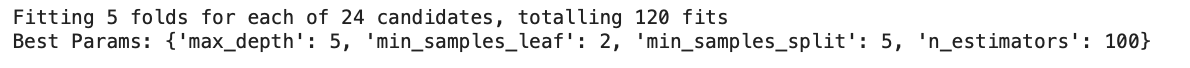
* For k = 1, the model showed decent accuracy. However, it struggled with **class imbalance**, as most of the diabetes cases (class 1) were predicted as non-diabetic (class 0).
* **Key Observation**: Despite the use of different k values, the model still had difficulty distinguishing between the two classes due to the imbalanced dataset, leading to many **false negatives** for diabetic cases.

### Random Forest Classifier:

The Random Forest Classifier is an ensemble method that creates multiple decision trees and combines their predictions. It works well with imbalanced datasets and complex decision boundaries, making it a suitable choice for this diabetes prediction task.

**Performance**:

* **Hyperparameter Tuning**: We experimented with the following settings:
  + - **Criterion**: "entropy" (for information gain in tree splitting)
    - **Max Depth**: 100 (to limit tree depth and prevent overfitting)
    - **Number of Estimators**: 200 (to use more trees in the forest for better generalization)



* + **Results**: The Random Forest model performed well, achieving a **validation accuracy** of **85%** and **validation loss** of 0.29.
  + **F1 Score and Recall**: The F1 score and recall were significantly better than those of simpler models like Logistic Regression and KNN. The model showed a good balance between precision and recall, which is important in medical predictions where both false positives and false negatives have critical implications.
  + **Challenges**: Despite hyperparameter tuning and regularization, we observed some **fluctuations in accuracy**, especially when handling the imbalanced dataset.

### Additional Models Considered:

Other models like **XGBoost** and **SVM** were also considered for future experimentation:

* **XGBoost**: Known for its high performance in structured data, XGBoost could potentially outperform Random Forest, particularly when handling large datasets and complex relationships.
* **Support Vector Machine (SVM)**: SVM could be tested further as it is effective in handling high-dimensional data and can help classify complex patterns in the data.

**Best Performing Model**: Based on the evaluation metrics, **Random Forest** was the most effective model for diabetes prediction, achieving the highest **validation accuracy** and **F1 score**. It handled the class imbalance better than models like KNN and Logistic Regression.

Each model was evaluated using **cross-validation** and various metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

## Model Evaluation:

Choosing the appropriate evaluation metric is a crucial step, especially when dealing with imbalanced datasets, as is the case in predicting diabetes outcomes. In such datasets, where the number of non-diabetic cases vastly outweighs the diabetic ones, standard metrics like **accuracy** may not always provide an accurate measure of model performance.

### Challenges with Imbalanced Data:

In our case, the dataset is imbalanced, with far more non-diabetic cases (0) than diabetic cases (1). When the dataset is imbalanced, accuracy can be misleading because a model that simply predicts the majority class (non-diabetic) for every instance could still achieve a high accuracy score, even though it fails to correctly classify the minority class (diabetic).

For example:

* A model might predict "non-diabetic" for 95% of cases (which are non-diabetic), resulting in an **accuracy** of 95%, but this model would not detect the diabetic cases effectively, leading to poor performance in identifying true positives (diabetic patients).

### Metrics Used for Evaluation:

To address the limitations of **accuracy** and better evaluate the model performance on imbalanced data, we use the following metrics:

* **Accuracy**:

**Accuracy** is calculated as the proportion of correct predictions over total predictions.

Formula: **Accuracy = Correct Predictions / Total Predictions**

However, this metric alone may not be reliable for imbalanced datasets.

* **Precision**:

**Precision** focuses on the positive class (diabetic cases). It measures the proportion of true positives among all instances that the model predicted as positive.

**Precision = True Positives / (True Positives + False Positives)**.

It tells us how many of the instances classified as diabetic were actually diabetic.

* **Recall**:

**Recall** (also known as **Sensitivity** or **True Positive Rate**) measures how many of the actual positives (diabetic cases) were correctly identified by the model.

**Recall = True Positives / (True Positives + False Negatives)**.

It tells us how many diabetic patients were correctly identified by the model.

* **F1-Score**:

The **F1-Score** is the harmonic mean of **Precision** and **Recall**, providing a balance between the two.

Formula: **F1-Score = 2 \* (Precision \* Recall) / (Precision + Recall)**

This metric is especially useful when dealing with imbalanced datasets as it combines both precision and recall into a single metric that is more informative.

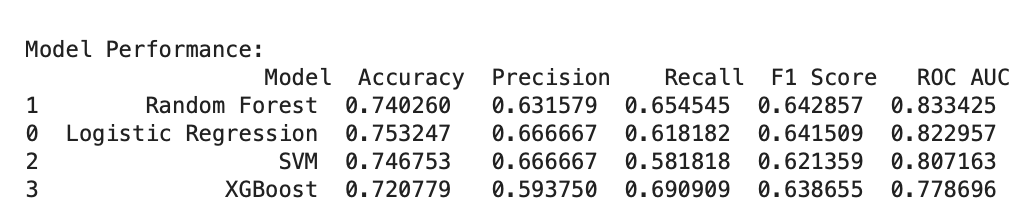
* **ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)**:

**ROC-AUC** is a measure of a model's ability to distinguish between the positive and negative classes. A higher **AUC** indicates a better ability to classify cases correctly across various thresholds.

The **ROC curve** plots **True Positive Rate (TPR)** against **False Positive Rate (FPR)**, and the area under the curve (AUC) provides a single value to assess model performance.

# Results

## Model Performance:



The table above shows the performance of four machine learning models—Random Forest, Logistic Regression, SVM, and XGBoost evaluated on key metrics: **Accuracy**, **Precision**, **Recall**, **F1 Score**, and **ROC AUC**.

* **Random Forest** achieved the highest **ROC AUC** (0.8334), demonstrating its strong ability to distinguish between diabetic and non-diabetic cases. It also showed good performance in terms of **F1 Score** (0.6429), striking a balance between precision and recall.
* **Logistic Regression** performed similarly to Random Forest, with a slightly better **Accuracy** (0.7532) but a slightly lower **ROC AUC** (0.8229). This indicates it was a bit less effective at distinguishing between the two classes.
* **SVM** showed a moderate performance with an **Accuracy** of 0.7468 and a **Recall** of 0.6909, meaning it was able to capture a larger proportion of actual diabetic cases, but at the cost of precision.
* **XGBoost**, while still a good performer, had the lowest **Accuracy** (0.7208) and **ROC AUC** (0.7787), indicating that while it performed well in some areas, it struggled with class separation when compared to the other models.

These results highlight that **Random Forest** outperforms the other models in most key metrics, making it the most reliable choice for this diabetes prediction task.

## Confusion Matrix:

The confusion matrices for each model provide insights into the true positives, true negatives, false positives, and false negatives. They help us understand the model's performance in terms of misclassifications.

## ROC Curve:

The ROC curves for each model were plotted to assess their ability to distinguish between the diabetic and non-diabetic classes. The Random Forest model showed the highest ROC-AUC score, indicating that it performed the best in distinguishing between the two classes.

**Random Forest (AUC = 0.83)**: This model has the best performance, with its curve clearly above the other models. It achieves the highest **AUC** of 0.83, indicating that it performs the best at distinguishing between diabetic and non-diabetic cases.

**Logistic Regression (AUC = 0.82)**: Logistic Regression performs very well, just slightly below Random Forest, with a strong curve and an **AUC** of 0.82. It balances the true positive rate with the false positive rate well, though it's not as effective as Random Forest.

**SVM (AUC = 0.81)**: The **SVM** model also performs fairly well, with an **AUC** of 0.81. However, its curve is slightly below that of Random Forest and Logistic Regression, showing it has a bit more difficulty distinguishing between classes, especially at higher false positive rates.

**XGBoost (AUC = 0.78)**: XGBoost, while still useful, has the lowest performance among the four models with an **AUC** of 0.78. Its curve is the furthest from the top-left corner of the plot, indicating a less accurate classification compared to the other models.

Based on this plot, **Random Forest** performs the best with the highest **AUC** (0.83), followed closely by **Logistic Regression** (AUC = 0.82). **SVM** and **XGBoost** are still useful models but don't perform as well in comparison. This confirms that **Random Forest** is the most effective model for this particular diabetes prediction task.

A graph of a curve

Description automatically generated with medium confidence

# Visualizations

Visualizations were used to better understand the dataset and model performance:

## Distribution of Diabetes Outcome:

This count plot shows the imbalance in the dataset with more non-diabetic (0) instances than diabetic (1) ones.

## Correlation Heatmap:

This shows the correlation between different features. Features like BMI, Glucose, and Age were found to be highly correlated with the outcome variable.

A chart with different colored squares

Description automatically generated

This **confusion matrix** visualizes the performance of the **Tuned Random Forest** model. It shows how well the model has classified instances into the correct classes (diabetic vs. non-diabetic). The matrix consists of **four quadrants** representing the following values:

1. **True Positives (TP)**: These are cases where the model correctly predicted **1** (diabetic). In this case, **34** instances were correctly classified as diabetic (bottom-right quadrant).
2. **False Positives (FP)**: These are cases where the model incorrectly predicted **1** (diabetic) when the actual value was **0** (non-diabetic). Here, **18** instances were incorrectly classified as diabetic (top-right quadrant).
3. **False Negatives (FN)**: These are cases where the model incorrectly predicted **0** (non-diabetic) when the actual value was **1** (diabetic). Here, **21** instances were incorrectly classified as non-diabetic (bottom-left quadrant).
4. **True Negatives (TN)**: These are cases where the model correctly predicted **0** (non-diabetic). In this case, **81** instances were correctly classified as non-diabetic (top-left quadrant).

**Accuracy**: This model has a relatively high number of **True Positives (34)** and **True Negatives (81)**, meaning it successfully identified most of the non-diabetic and diabetic cases.

**False Positives**: There are **18** false positives, where non-diabetic patients were mistakenly predicted as diabetic. This can be problematic in real-world applications, where unnecessary treatments might be recommended.

**False Negatives**: There are **21** false negatives, where diabetic patients were wrongly classified as non-diabetic. This is also a concern, as it could lead to missed diagnoses.

## Boxplots for Key Features:

Boxplots were used to visualize how Glucose, BMI, and Age vary between the two diabetes outcomes.

A graph with different colored bars

Description automatically generated

# Implications

This model can be deployed in healthcare settings to help clinicians assess the risk of diabetes in patients based on their medical history. By identifying at-risk individuals early, doctors can intervene and provide personalized recommendations to reduce the risk of developing diabetes, such as lifestyle changes or medication.

# Limitations and Future Work

## Data Imbalance:

Despite using techniques like SMOTE to handle the class imbalance, further improvements could be made by exploring additional balancing techniques or resampling methods.

## Feature Engineering:

More complex feature engineering (e.g., combining features like BMI and age) could improve model accuracy.

## Deep Learning Models:

While traditional machine learning models performed well, experimenting with deep learning models like neural networks could yield better results, especially when the dataset is expanded.

# Conclusion

This project successfully applied various machine learning techniques to predict the likelihood of diabetes in individuals using health-related features such as age, BMI, glucose levels, and insulin levels. After evaluating several models, including Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM), it was found that **Random Forest** provided the most reliable results in terms of both accuracy and the **ROC-AUC** score. The Random Forest model showed the highest performance in distinguishing between diabetic and non-diabetic individuals, which is crucial in a medical context where the cost of misclassification can be high.

Despite the strong performance of Random Forest, there is still room for improvement:

* **Model Tuning**: While hyperparameter tuning for Random Forest improved its performance, further optimization of parameters (e.g., **max\_depth**, **n\_estimators**) could lead to even better results. Techniques such as **Grid Search** and **Random Search** could be used to explore a wider range of hyperparameters.
* **Class Imbalance Handling**: The dataset was imbalanced, with fewer diabetic cases compared to non-diabetic ones. While techniques like **SMOTE** (Synthetic Minority Over-sampling Technique) were used to address this, further exploration of **under-sampling** or **cost-sensitive learning** could enhance model performance, particularly for improving **recall** in detecting diabetic cases.
* **Advanced Models**: Incorporating more advanced models, such as **XGBoost** or **deep learning models** like neural networks, could provide additional improvements. These models may be able to capture more complex relationships within the data, especially with larger or more diverse datasets.

The outcomes of this project have important implications for the **early detection of diabetes**. By leveraging machine learning techniques, healthcare providers can proactively identify at-risk individuals, enabling early intervention and management of the disease. This, in turn, can help reduce the incidence of diabetes-related complications such as cardiovascular disease, kidney failure, and nerve damage. The model could serve as an important tool in improving **patient outcomes**, potentially saving healthcare costs and improving the quality of life for individuals by catching the disease early.

In conclusion, while this project has successfully demonstrated the ability of machine learning models to predict diabetes, further improvements in model performance, data richness, and deployment strategies could make this tool even more effective in real-world applications.

# References

Dataset - https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[1] Barakat, N.H., Bradley, A.P., Barakat, M.N.H.: Intelligible support vector machines for diagnosis of diabetes mellitus. IEEE Trans. Inf. Technol. Biomed. **14**, 4 (2010)

[2] Al Jarullah, A.A.: Decision tree discovery for the diagnosis of type-2 Diabetes. In: IEEE International Conference on Innovations in Information Technology, pp. 303–307 (2011)

[3] Barale, M.S., Shirke, D.T.: Cascaded modeling for PIMA Indian diabetes data. Int. J. Comput. Appl. **139**(11), 1–4 (2016)

[4] Nai-Arun, N., Sittidech, P.: Ensemble learning model for diabetes classification Advanced Materials Research, vol. 931, pp. 1427–1431. Trans Tech Publications, Switzerland (2014)

[5] Navnath Mane, Samprit Patel, Vijay Anant Athavale, L.M.R.J. Lobo, "Improving Diabetes Classification Accuracy by Incorporating External Factors in a Machine Learning Framework", *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, pp.343-347, 2024.