Ministry of Higher Education and Scientific Research

M'Hamed Bougara University – Boumerdes

Faculty of Science | Department of Computer Science

Domain : Computer Science

Field : Computer Science

Specialty : Information Systems Engineering

End-of-Study Project Report submitted in fulfilment of the requirements for the Academic Bachelor's Degree

Theme

Web Application for Diabetes Detection

Using Neural Networks

Presented by:

DENBRI Mehdi

FODIL Mohamed Anouar

SAOUDI Mohamed Marouane

BENCHAITA Nassim

Supervised by:

Ms D. AKROUR

Defended on ...../ ...../ *2026*

Academic Year 2025 – 2026

Acknowledgements

We express our deepest gratitude to Allah for granting us the strength, perseverance, and guidance throughout this project. Without his blessings, this work would not have been possible.

We would like to express our sincere appreciation to our supervisor, **Ms. AKROUR Djouher**, for accepting to supervise this work with rigor and kindness. We warmly thank her for her valuable advice, constant support, availability, and remarkable expertise throughout this project on the early detection of diabetes recurrence and survival using artificial intelligence. Her patience, precious guidance, and scientific vision were decisive in the success of our work. We were honored to benefit from her experience and supervision..

We also extend our thanks to all the professors of **the Faculty of Computer Science of UMBB** for the quality of education provided throughout our academic journey. Their teaching and dedication formed the solid foundation upon which we built this work.

Our appreciation also goes to the examiner, **Mr.CHAOUCHE Ali** for accepting to evaluate our work and for their valuable remarks and suggestions.

A very special thanks goes to the four of us for the mutual support, patience, and commitment we demonstrated throughout this project. We are proud to have successfully completed this work together in a true spirit of teamwork and sisterhood, overcoming every challenge with determination and solidarity.

Finally, our most sincere thanks go to everyone who contributed, directly or indirectly, to the realization of this ambitious project: our families for their unwavering support and constant encouragement, our loved ones for their comforting presence during difficult moments, and our friends for their loyalty and kindness. Their trust and love were an endless source of motivation.

Dedication

I dedicate this work first and foremost to my **family**, especially my **mother and father**, for their unconditional love, sacrifices, and constant encouragement. They have always been dedicated to my success and motivated me to reach the highest level of excellence. *May God protect and bless them.*

To my wonderful **sisters** — my big sister and my middle sister — thank you for your advice, support, and for always believing in me.

To my colleagues and classmates, especially **Nassim, Marouane, Mohamed, and Nassim HR**, thank you for the unforgettable moments, teamwork, friendship, and continuous encouragement during this academic journey.

To my dear friends, **Abdulla, Ali, Walid, Mehdi, and Ayoub**, for their motivation, support, and positive energy throughout this journey.

To my dear friend **Katia**, thank you for your support, encouragement, and for always being by my side during this journey. Your friendship and positive spirit truly meant a lot to me.

Finally, this achievement is not mine alone; it is the result of every prayer, every encouragement, and every sacrifice made for my success.

MEHDI

Dedication

I dedicate this work to my beloved parents, whose love, sacrifices, support, and encouragement have always been the source of my strength and motivation. Thank you for believing in me, standing by my side, and helping me reach this milestone. This achievement would not have been possible without you.

To my family, especially my big sister, for her support, advice, encouragement, and for always being there for me throughout my years of study.

To my colleagues who supported and motivated me throughout the realization of this project, **Mehdi, Nassim, and Mohamed,** for their cooperation, encouragement, and the great moments we shared during this work.

To everyone who helped me directly or indirectly in the completion of this project, especially my friends from university and outside university, **Yazid, Zahrou, Abdesslam, Lamine, Walid, Riadh, and Malik** for their support, encouragement, advice, and for all the memorable moments we shared throughout this journey.

Last but not least, I want to thank me. I want to thank me for believing in me. I want to thank me for doing all this hard work. I want to thank me for having no days off. I want to thank me for never quitting.

MAROUANE

Dedication

This work is dedicated to my **family**, whose endless support, patience, and encouragement have been my foundation throughout my years of study.

A special and heartfelt tribute is dedicated to my **mother and father**. Your unconditional love, immense sacrifices, and constant motivation gave me the strength to face every difficulty and ultimately reach this important milestone.

To my dear friends and colleagues — **Mehdi, Mohamed, and Marouane** — who stood by me throughout the realization of this project. Thank you for your cooperation, your unwavering encouragement, and the unforgettable moments we shared along the way.

To all the friends, peers, and individuals I had the privilege of meeting during my university years, as well as everyone who contributed, directly or indirectly, to the completion of this work.

Finally, to **myself**. I honor the sleepless nights, the relentless focus, and the absolute refusal to accept mediocrity. This is a tribute to being both my own harshest critic and my greatest supporter — for putting in the work when no one was watching, and holding the pride now that it is done.

NASSIM

Dedication

This work is dedicated to my **family** — my **mother, my father, my brother, and my sister** — for their unconditional love, endless patience, and unwavering support throughout every step of this journey.

To my **extended family**, whose warm encouragement has always been a source of strength and comfort along the way.

To my closest friends, **Hichem, Sidahmed, and Koussai**, for their loyal presence, constant motivation, and the genuine support they offered me throughout this work. Your friendship and encouragement made this journey far more enjoyable and rewarding.

Finally, to **myself** — for believing when doubt crept in, for showing up even on the hardest days, for putting in the work without looking for shortcuts, and for *never giving up*, no matter how difficult the road became.

MOHAMED

Résumé

Le diabète constitue l'une des maladies chroniques les plus répandues à l'échelle mondiale, touchant plus de 420 millions de personnes selon l'Organisation Mondiale de la Santé (OMS). Un diagnostic tardif ou imprécis peut entraîner des complications graves, voire mettre en danger la vie des patients. La détection précoce et fiable du diabète représente donc un enjeu majeur de santé publique.

Ce mémoire présente la conception et le développement d'une plateforme web intelligente de détection du diabète reposant sur des techniques d'apprentissage automatique. Le système proposé exploite la base de données clinique *Pima Indians Diabetes Database* pour entraîner, évaluer et comparer plusieurs algorithmes de classification, notamment les Machines à Vecteurs de Support (SVM), les Forêts Aléatoires (Random Forest), les K plus proches voisins (KNN) et les Réseaux de Neurones Artificiels (ANN). Un pipeline complet de prétraitement des données est mis en œuvre afin de gérer les valeurs manquantes, normaliser les attributs et sélectionner les caractéristiques les plus pertinentes.

Le système proposé est une application web conçue en Python Flask, offrant une interface conviviale et facile à utiliser. Elle permet de saisir les données cliniques d'un patient et d'obtenir une prédiction en temps réel quant à la présence ou l'absence du diabète. Les modèles sont évalués à l'aide de métriques médicales adaptées telles que la précision, le rappel, le F1-score et la matrice de confusion.

Mots clés : Détection du diabète, apprentissage automatique, classification binaire, réseaux de neurones artificiels, Application web.

Abstract

Diabetes is one of the most widespread chronic diseases worldwide, affecting more than 420 million people according to the World Health Organization (WHO). A late or inaccurate diagnosis can lead to serious complications and, in some cases, endanger patients' lives. The early and reliable detection of diabetes therefore represents a major public health challenge.

This dissertation presents the design and development of an intelligent web platform for diabetes detection based on machine learning techniques. The proposed system exploits the *Pima Indians Diabetes Database* clinical dataset to train, evaluate, and compare several classification algorithms, including Support Vector Machines (SVM), Random Forest, K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN). A complete data preprocessing pipeline is implemented to handle missing values, normalize attributes, and select the most relevant features.

The proposed system is a web application carefully designed using Python Flask, providing a user-friendly and easy-to-use interface. It allows clinical patient data to be entered and a real-time prediction to be obtained regarding the presence or absence of diabetes. The models are evaluated using medically appropriate metrics such as accuracy, precision, recall, F1-score, and the confusion matrix.

Keywords: Diabetes detection, machine learning, binary classification, artificial neural networks, web Application.

Contents

Dedication **iv**

Résumé **vi**

Abstract **vii**

General Introduction **1**

1 Medical Context of Diabetes and AI-Based Clinical Decision Support Systems **3**

1 Introduction . . . 3

Part I — Medical Context of Diabetes . . . 3

2 Pathophysiology and Diagnostic Challenges . . . 3

2.1 Definition and Clinical Background . . . 3

2.1.1 Definition of Diabetes Mellitus . . . 3

2.1.2 Pathophysiology of Diabetes Mellitus . . . 3

2.1.3 Regulation of Blood Glucose . . . 4

2.2 Symptoms . . . 4

2.3 Risk Factors . . . 4

3 Problematic of diabetes diagnosis . . . 5

3.1 Difficulties in traditional diagnosis . . . 5

3.2 Clinical Decision Support Systems (CDSS) . . . 6

3.3 The Integration and Utility of AI and ML in Modern Healthcare . . . 6

3.3.1 Augmentation of Clinical Capabilities . . . 6

3.3.2 Processing High-Scale Medical Datasets . . . 6

3.3.3 Detection of Hidden Patterns . . . 6

3.3.4 Real-Time Predictive Velocity . . . 6

3.3.5 Resource Optimization and Economic Impact . . . 6

Part II — Machine Learning for Diabetes Detection . . . 7

4 Machine Learning . . . 7

4.1 Domain applications of machine learning . . . 7

4.2 Types of Machine Learning . . . 7

4.3 Machine Learning Tasks . . . 8

5 Taxonomy of Algorithms for Classification . . . 8

5.1 Classical Machine Learning Algorithms . . . 9
5.2 Neural Network-Based Methods . . . 10
6 Machine Learning Workflow . . . 11
7 Evaluation Metrics for Medical Classification . . . 11
7.1 Confusion Matrix . . . 11
7.2 Accuracy (ACC) . . . 12
7.3 Precision (PREC) . . . 12
7.4 Recall (sensitivity) . . . 12
7.5 F1-score (or F-score) . . . 12
8 Explainable AI . . . 12
9 State of the art . . . 13
10 Conclusion . . . 14

2 Design and Implementation of the Prediction System 15
1 Introduction . . . 15
2 Development Environment . . . 15
2.1 Core Libraries . . . 15
2.2 Computational Infrastructure . . . 16
3 General architecture . . . 16
4 Dataset Description . . . 16
4.1 Feature Definitions . . . 17
4.2 Target Variable . . . 17
4.3 Sample Data (First 5 Rows) . . . 17
4.4 Summary Statistics . . . 18
5 Data Preprocessing & Feature Engineering . . . 18
5.1 Handling Missing Values . . . 18
5.2 Feature Engineering . . . 18
5.3 Feature Normalisation . . . 18
5.4 Class Balancing . . . 19
5.5 Train-Test Split . . . 19
6 Machine Learning Models . . . 19
6.1 Presentation of the Selected Models . . . 19
6.1.1 SVM . . . 19
6.1.2 Random Forest Classifier (Enhanced) . . . 20
6.1.3 XGBoost Classifier . . . 20
6.1.4 ANN . . . 20
7 Results and comparative analysis . . . 21
7.1 Full Model Comparison Table . . . 21
7.2 Feature Importance Analysis . . . 22
8 Conclusion . . . 22

3 Web Application Development and Interface 23
1 Introduction . . . 23

2 Technologies and Development Environment 23
3 System Architecture 24
3.1 The Request–Response Cycle 24
3.2 System Use-Case Configuration 25
3.3 Authentication Sequence Configuration 25
3.4 Prediction Sequence 26
3.5 Class Diagram 27
4 Main Functionalities 28
4.1 Authentication Interface 28
4.2 Prediction Interface 29
4.3 Prediction History 29
4.4 Result Visualisation 29
5 Database Design 31
5.1 Document Schema Specifications 31
5.1.1 Users Document Schema 31
5.1.2 Predictions Document Schema 32
5.1.3 Patient Data Embedded Schema 32
6 Discussion 32
6.1 Strengths and Usability 32
6.2 Limitations 33
7 Conclusion 33

General Conclusion **34**

Bibliography **34**

List of Figures

1.1 Blood glucose regulation 4
1.2 Types of machine learning 8
1.3 Some of the most commonly used machine learning algorithms 9
1.4 Neural Network-Based Methods Architecture 10

2.1 general machine learning pipeline (generated by canva) 16

3.1 System UML Use-Case Diagram 25
3.2 Authentication sequence diagram. 26
3.3 prediction sequence 27
3.4 class diagram 28
3.5 Conceptual Framework of the Dynamic Results Rendering Dashboard 30
3.6 Database Architecture and Schema Document Relationship Map 31

List of Tables

1.1 Comparative table of Diabetes Detection Studies 13

2.1 First five rows of the Pima Indians Diabetes dataset 17
2.2 Descriptive statistics for all features in the dataset 18
2.3 Engineered features derived from the original 8 clinical variables 19
2.4 SVM best hyperparameters selected by GridSearchCV 20
2.5 Optimised Random Forest hyperparameters selected by GridSearchCV 20
2.6 XGBoost hyperparameters selected by GridSearchCV 20
2.7 ANN architecture with Batch Normalisation (11 input features) 21
2.8 ANN training configuration 21
2.9 Unified performance comparison across all models on the 154-sample test set 21
2.10 Top 5 important features identified by Random Forest and XGBoost 22

3.1 Technology Stack Summary 24

General Introduction

Today's web applications have transitioned from basic advertising tools into complex management systems that outperform classic desktop software. As the internet expands, the computing world is shifting toward a fully online model. In response, the healthcare sector is modernizing its processes by implementing mobile and web-based solutions and upskilling staff on modern information management systems

Diabetes is one of the most widespread chronic diseases in the world. According to the World Health Organization (WHO), more than **420 million people** are currently affected, and this number continues to grow at an alarming rate. A late or inaccurate diagnosis can lead to serious complications, and in some cases even put patients' lives at risk. Early detection is therefore a major public health challenge, requiring reliable, fast, and accessible tools capable of effectively analyzing complex medical data.

Machine Learning (ML) offers promising possibilities to address this challenge. By exploiting clinical data — such as glucose levels, body mass index (BMI), blood pressure, and family medical history — classification algorithms are capable of identifying characteristic patterns that can predict, with high accuracy, whether an individual has diabetes or not. This data-driven approach represents a valuable complement to traditional medical expertise, and its integration into modern healthcare systems has become an active and rapidly evolving research area.

The objective of our project, presented in this dissertation, is the **design and development of an intelligent diabetes detection platform** based on machine learning techniques. The system relies on a publicly available clinical dataset — the *Pima Indians Diabetes Database* — to train, evaluate, and compare several classification models. The proposed web application implements the following functionalities:

- Preprocessing and cleaning of raw medical data (handling missing values, normalization, feature selection).
- Training and comparison of multiple classification algorithms including Support Vector Machines (SVM), Random Forest, and Artificial Neural Networks (ANN).
- Evaluation of models using medically appropriate metrics: accuracy, precision, recall, F1-score, and confusion matrix.
- Deployment of a user-friendly web interface allowing the input of patient data and real-time diabetes prediction.

In addition to its practical utility, the proposed application is designed according to ergonomic principles in order to facilitate its use and ensure a positive experience for the end user.

To achieve these objectives, we have organized this dissertation into three chapters as follows:

1. **Chapter 1 — Medical Context of Diabetes and AI-Based Clinical Decision Support Systems:** introduces the medical background of diabetes mellitus, including its symptoms, risk factors, and diagnostic challenges. It also presents Clinical Decision Support Systems, the role of AI and Machine Learning in healthcare, and a brief review of the main classification methods and related work in diabetes detection.

2. **Chapter 2 — Design and Implementation of the Prediction System:** presents the development environment, dataset, preprocessing, feature engineering, and model training process. It also compares the performance of SVM, Random Forest, XGBoost, and ANN, showing that ANN achieved the best overall results.

3. **Chapter 3 — Web Application Development and Interface:** presents the design and implementation of the ClearSugar web application for real-time diabetes risk prediction. It describes the multi-tier architecture, the technology stack (React.js, Node.js, Flask, MongoDB), and the main functionalities including authentication, single-patient and batch prediction, SHAP-based result visualisation, and prediction history management.

Finally, we will conclude this document with a **General Conclusion** summarizing the contributions of our work and proposing future directions for improvement and extension of the system.

Chapter 1

Medical Context of Diabetes and AI-Based Clinical Decision Support Systems

1 Introduction

Machine Learning provides powerful techniques for analyzing medical data and predicting diseases automatically from clinical features. This chapter presents the medical background of diabetes, its diagnostic challenges, and the role of AI-based CDSS. It also introduces the fundamental concepts of machine learning, the main classification algorithms used for diabetes detection, and a brief state-of-the-art review related to this work.

Part I — Medical Context of Diabetes

2 Pathophysiology and Diagnostic Challenges

2.1 Definition and Clinical Background

2.1.1 Definition of Diabetes Mellitus

Diabetes mellitus is a chronic metabolic disorder where the body fails to regulate blood glucose. This occurs because the pancreas produces insufficient insulin or cells become resistant to its effects.

Normally, insulin moves glucose into cells for energy. In diabetes, this process is disrupted, causing glucose to build up in the bloodstream. If untreated, high blood sugar leads to severe organ damage.

2.1.2 Pathophysiology of Diabetes Mellitus

The human body maintains internal balance (homeostasis) through complex regulatory systems. Diabetes disrupts this balance by impairing blood glucose control. Persistent hyperglycemia causes metabolic stress and progressively damages tissues and organs, explaining

the chronic and systemic nature of the disease.[2]

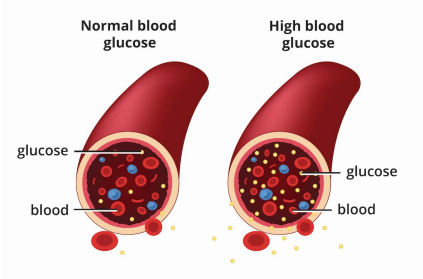


Figure 1.1: Blood glucose regulation

2.1.3 Regulation of Blood Glucose

Blood glucose is regulated mainly by insulin and glucagon through a negative feedback system.

- When blood glucose levels rise, insulin is released by pancreatic β-cells. Insulin helps cells especially muscle and fat cells absorb glucose and store it as glycogen.
- When blood glucose levels drop, glucagon is released by α-cells, stimulating the liver to release glucose into the bloodstream.[2] (see figure 1.1)

In diabetes, this regulation fails:

- In **type 1 diabetes**: insulin production is absent. In fact this is an autoimmune disease that usually begins in childhood. The immune system destroys pancreatic β-cells, leading to absolute insulin deficiency [3]
- In **type 2 diabetes**, insulin resistance and insufficient insulin secretion occur. [3] This leads to chronic hyperglycemia and long-term complications.[2]

2.2 Symptoms

Many people with diabetes may have no symptoms, especially in early stages. When present, symptoms can affect quality of life.

Common symptoms include:[4]

- Fatigue
- Increased thirst and urination
- Digestive problems such as nausea, constipation, diarrhea, and abdominal pain
- Bloating and early fullness after meals

2.3 Risk Factors

- **Genetics** : it significantly impact type 2 diabetes risk, though environmental factors often determine if the disease develops. While the condition is polygenic, a healthy

lifestyle can delay or prevent onset even in those genetically predisposed.[5]

- **Obesity**: Excess abdominal fat triggers inflammation and releases substances that increase insulin resistance.
- **Physical Inactivity**: Sedentary lifestyles promote weight gain; conversely, regular exercise improves insulin sensitivity.[5]
- **Diet**: Diets high in sugar and saturated fats increase risk, while high-fiber diets stabilize glucose.[5]
- **Hyperglycemia**: Frequent glucose spikes overload tissues and accelerate insulin resistance.[5]
- **Stress**: Chronic stress elevates cortisol, which raises blood sugar and inhibits insulin effectiveness.[5]
- **Chronic Inflammation**: Driven by poor diet and obesity, long-term inflammation fuels metabolic dysfunction.

3 Problematic of diabetes diagnosis

3.1 Difficulties in traditional diagnosis

Healthcare providers encounter several obstacles when diagnosing and classifying diabetes:

Limited Access to Testing: High costs, lack of medical checkups, and low awareness prevent regular screening, particularly in underserved environments.[6]

- **Fluctuating Glucose Levels**: Blood sugar varies rapidly due to diet, exercise, and stress, which can lead to inconsistent or inconclusive test results.[7]
- **Ambiguous Symptoms**: Clinical signs like obesity or acanthosis nigricans indicate insulin resistance but do not definitively confirm a specific diabetes type.
- **Complex Heredity**: While over 80% of young Type 2 patients have a positive family history, establishing a detailed three-generation pedigree is often difficult yet necessary for differentiation.
- **Overlapping Age of Onset**: Age is an unreliable indicator; Type 1 is increasingly found in adults (often misdiagnosed as Type 2), while Type 2 is becoming more common in youth.[8]
- **Clinical Presentation Overlap**: While severe symptoms and DKA are more common in Type 1, roughly 5% of Type 2 youth also present with DKA, making symptom severity an unreliable diagnostic standalone.[9]

These difficulties motivated us, so to develop Decision Support Systems (DSS). Doctors face many challenges when diagnosing diseases, including the risk of errors, complex data, and

time pressure, which can lead to mistakes. DSS are designed to assist doctors in diagnosing diabetes by analyzing data, providing recommendations, and supporting decision-making.

3.2 Clinical Decision Support Systems (CDSS)

The integration of CDSS into modern medicine represents a fundamental shift from static, rule-based logic to dynamic, analytical frameworks powered by Artificial Intelligence (AI) and Machine Learning (ML).

A Decision Support System (DSS) is an interactive, computer-based information system that helps managers and healthcare providers compile useful information from raw data, documents, personal knowledge, and business models to identify and solve complex problems. [10].

3.3 The Integration and Utility of AI and ML in Modern Healthcare

3.3.1 Augmentation of Clinical Capabilities

AI transforms healthcare from a reactive to a proactive, preventive model. Unlike traditional rule-based systems, ML learns autonomously from patient data—mimicking clinical residency—to reduce reliance on subjective intuition and minimize cognitive bias.[11]

3.3.2 Processing High-Scale Medical Datasets

The volume of modern medical data (vitals, lab results, and genomics) has exceeded human processing capacity. AI provides the essential substrate for analyzing these massive datasets, enabling the early identification of chronic disease risks at scale.[12]

3.3.3 Detection of Hidden Patterns

ML excels at uncovering non-linear relationships within high-dimensional data that elude human perception. [13] This because it handles dimensionality where identifying relevant features among thousands of predictors. Furthermore, it detects non-linear interactions by Mapping complex biological relationships.

3.3.4 Real-Time Predictive Velocity

In emergency treatment, speed is decisive. ML algorithms process millions of data points to provide real-time alerts for conditions like sepsis hours before overt symptoms appear, mitigating human error caused by data fatigue.

3.3.5 Resource Optimization and Economic Impact

AI is projected to significantly enhance global healthcare operational efficiency by 2026, enabling substantial systemic savings through:[15]

- Workflow Efficiency: Documentation time can be reduced by 40%.

- Logistical Management: Bed turnover improvements of 10–15% and reduced patient wait times.
- Long-term Savings: Early-stage disease detection significantly lowers longitudinal treatment costs.

Part II — Machine Learning for Diabetes Detection

4 Machine Learning

Machine learning (ML) is part of AI. It uses Computational Algorithms to make decisions or predictions without explicit, hard-coded instructions. In fact, ML emulates human intelligence to learn patterns from the surrounding environment, experience, and training data [16]. This way, computers can perform tasks like humans or even better without explicit programming.

4.1 Domain applications of machine learning

ML can be applied in many fields, such as healthcare for disease prediction and medical imaging. It can also be used in recommendation systems, and in finance for fraud detection. Additionally, it is used in computer vision and image recognition, and in natural language processing for translation and chatbots.

4.2 Types of Machine Learning

Machine learning has three types:

- **Supervised Learning :** The system uses data where the correct answers are already known, such as in an email filter. During this process, the model learns to identify patterns by mapping input features to specific target labels. It then leverages these learned relationships to accurately predict outcomes for new, unseen data.
- **Unsupervised Learning :** The system uses data without known answers and discovers patterns by itself. It works with large amounts of data to organize and classify similar or different items.
- **Reinforcement Learning :** The system learns by trial and error. When it performs a wrong action, it receives a punishment, and when it performs a correct action, it receives a reward from the environment, such as self-driving cars.(see figure 1.2)

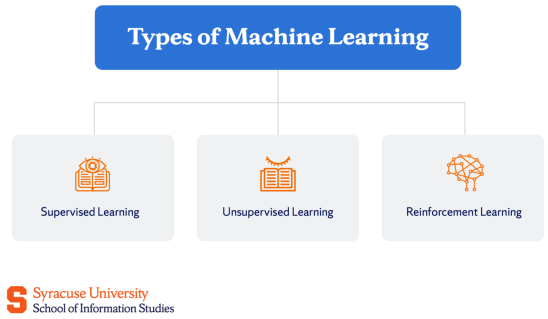


Figure 1.2: Types of machine learning

4.3 Machine Learning Tasks

Machine learning can solve problems in many ways using different techniques. One of the main tasks of machine learning is classification, which involves assigning data to categories based on input features.

Classification problems can be divided into three main types:

- **Binary Classification:** Binary classification produces an output consisting of two classes. For example, in email spam detection, emails are classified as either spam or not spam.
- **Multiclass Classification:** Multiclass classification produces an output consisting of more than two classes. For example, an email system can classify messages into multiple categories such as Spam, Promotion, Social, Work, or Personal.
- **Multilabel Classification:** Multilabel classification allows data to belong to multiple categories at the same time. For example, an email can be classified as both Work and Important simultaneously.
- **Regression:** Regression is a machine learning task that predicts continuous numerical values based on input features, such as salary prediction or house price estimation.
- **Prediction:** Prediction is the process of using a trained machine learning model to estimate future or unknown outcomes based on learned patterns from data (predicting disease risk, predicting future sales).

In our project, we will focus on solving a binary classification problem, where the model classifies data into two distinct classes (person is diabetic or not diabetic).

5 Taxonomy of Algorithms for Classification

Classification algorithms generally fall into two categories: classical methods based on statistical and geometric principles, and neural networks designed to model complex, non-

linear relationships [21].

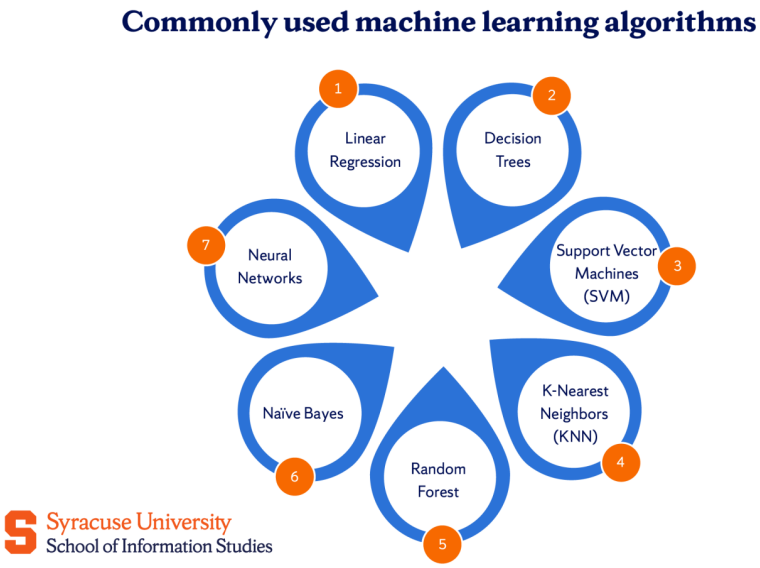


Figure 1.3: Some of the most commonly used machine learning algorithms

5.1 Classical Machine Learning Algorithms

Classical algorithms are generally characterized by their transparency and efficiency in low-dimensional settings. They serve as the standard baseline for predictive tasks and clinical research. (see figure 1.3)

Common algorithms in this domain include Linear Regression, Decision Trees, K-Nearest Neighbors (KNN), Naive Bayes,Neural Networks .In our specific project, we focus our methodology on the implementation of three select models of classical algorithms : Support Vector Machines (SVM), Random Forest, and XGBoost.

Support Vector Machines (SVM)

SVM are powerful classification algorithms that work by finding the optimal hyperplane that best separates different categories in a dataset [21]. The algorithm seeks the hyperplane that maximizes the margin between classes. By focusing on the "support vectors" (the data points closest to the boundary), it ensures high accuracy and robustness against noise.

Random Forest

Random Forest is an ensemble learning method that combines the output of multiple decision trees trained on random subsets of data [23]. It aggregates the predictions of numerous individual trees, using majority vote for classification or averaging for regression to produce a single, robust result.

XGBoost (short for eXtreme Gradient Boosting)

XGBoost, or eXtreme Gradient Boosting, is an advanced implementation of the gradient boosted decision tree algorithm designed for speed and performance. It functions by sequentially building decision trees where each new tree aims to correct the errors made by previous ones, while utilizing regularization techniques to control model complexity and prevent overfitting.[39]

5.2 Neural Network-Based Methods

Neural networks represent a class of machine learning models inspired by the structure and function of the biological brain. They consist of layers of interconnected nodes, or neurons, that work together to process and transform data, their architecture composed of input, hidden, and output layers linked by weighted connections. These layers perform iterative non-linear transformations to learn and refine patterns from complex data.(see figure 1.4)

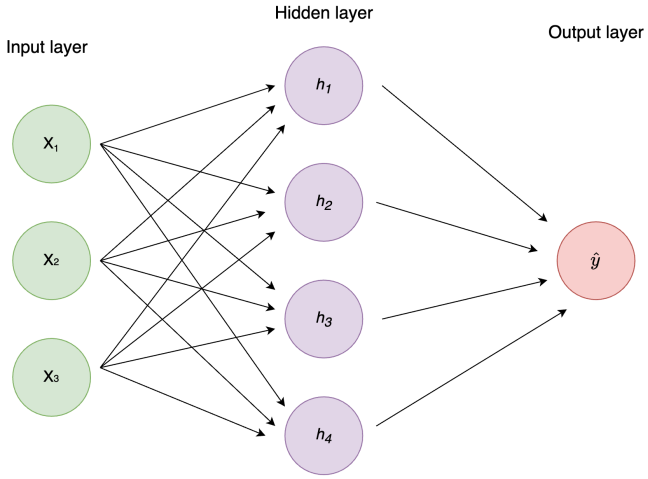


Figure 1.4: Neural Network-Based Methods Architecture

During the learning process, information flows through the network and weights assigned to each connection are adjusted, allowing the network to autonomously discover and learn complex patterns within the data [17, 11].

While this category encompasses diverse architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs),our project specifically investigates and focuses on the application of Artificial Neural Networks (ANNs) for predictive analysis.

Artificial Neural Networks (ANN / MLP)

ANN consists of an input layer, one or more hidden layers, and an output layer. Artificial neurons are linked by weighted connections; the network learns by adjusting these weights (often through backpropagation) to minimize the difference between predictions and actual outcomes [18]. In fact ANN are Highly effective for complex clinical decision-making involving unstructured, limited, or high-dimensional information, such as multi-omics data or disease risk assessment [25].

6 Machine Learning Workflow

The process of ML is as follow:

1. **Data collection** : This is the first step that can collect large data from different sources (transactions, customer interactions, sensor readings, medical records, attributes, pixels).
2. **Data Preprocessin** : Here, raw data are prepared by handling missing values, removing noise, structuring the data, eliminating duplicate entries, and transforming categorical variables into a machine-readable format.
3. **Model Selection** : This is the step where the system chooses a model based on the type of problem.
4. **Training Phase** : During this phase, the model learns patterns from the training data to make accurate predictions. The more relevant data it receives, the better its performance and decisions.
5. **Evaluation**: The model is evaluated using different dataset which was not seen before. This step is crucial before deployment to make sure that the model is responsive to new information rather than the simple examples.

7 Evaluation Metrics for Medical Classification

Evaluation metrics are quantitative measures used to assess the performance of a machine learning model. They help determine how well a model's predictions align with the actual data. Common metrics include[31].

7.1 Confusion Matrix

It visualizes the count of correct and incorrect predictions, broken down by each class. It consists of four key components:

- **True Positive (TP):** The number of correctly classified positive samples. A data point is counted as TP when a positive outcome is predicted and the actual outcome is also positive.
- **False Positive (FP):** The number of samples incorrectly classified as positive. A data point is FP when a positive outcome is predicted but the actual outcome is negative.
- **False Negative (FN):** The number of samples incorrectly classified as negative. A data point is FN when a negative outcome is predicted but the actual outcome is positive.
- **True Negative (TN):** The number of correctly classified negative samples. A data point is TN when a negative outcome is predicted and the actual outcome is also negative.

In medical applications, a FN occurs when a patient who is actually sick is classified as

healthy, which are the most dangerous type of error.

In contrast, a FP occurs when a healthy patient is classified as sick. Although this may lead to additional medical tests, stress, or unnecessary treatment, the patient remains safe. Therefore, FP are generally less harmful than FN [31].

7.2 Accuracy (ACC)

It is the proportion of correctly classified samples among all instances in the evaluation dataset. Although widely used in medical machine learning applications, it can be misleading in the presence of class imbalance, as high accuracy may be achieved by predicting only the majority class [31].

7.3 Precision (PREC)

It denotes the proportion of the retrieved samples which are relevant and is calculated as the ratio between correctly classified samples and all samples assigned to that class.

7.4 Recall (sensitivity)

It measures the ability of a classification model to correctly identify all positive cases. In other words, it evaluates how well the model detects patients who actually have the disease. Recall is particularly critical in medical diagnosis because it directly relates to the model's ability to avoid FN, which are the most dangerous type of error.

This is why recall is a crucial evaluation metric in healthcare systems because it ensures that the maximum number of diseased patients are correctly identified [31].

7.5 F1-score (or F-score)

It combines both precision and recall into a single value, providing a balance between them, and a more comprehensive evaluation of the model's performance.

8 Explainable AI

Explainable AI (XAI) refers to a set of methods and techniques designed to make the decisions of AI systems understandable and interpretable by human users. Unlike so-called "black-box" models, such as deep neural networks, XAI approaches aim to provide clear explanations of how a model reaches its conclusions.[16]

This is important in our work because healthcare is a critical sectors where algorithmic transparency is an ethical, regulatory, and operational requirement.[16, 30]

Two major families of XAI methods are commonly distinguished:

Intrinsic methods: naturally interpretable models (decision trees, linear regression) whose internal structure is directly readable and self-explanatory.[30]

Post-hoc methods: techniques applied after model training to explain complex models, such as *LIME* (Local Interpretable Model-agnostic Explanations) and *SHAP* (SHapley Additive exPlanations), which quantify the contribution of each feature to a given prediction.[27, 28]

9 State of the art

Current research demonstrates a clear evolution from traditional statistical methods to advanced neural architectures for diabetes and retinopathy detection, as shown in table 1.1 bellow.

References	Year	Method	Hyperparameter	Accuracy
[35]	2019	LR, RF, Decision Tree, SVM, ANN, AdaBoost, Gradient Boosting, LDA, Extra Trees, Perceptron, Bagging, Gaussian Naïve Bayes	K-means (k=2) for feature engineering, data scaling, pipeline chaining preprocessing with classifier	96%–98.8% (ANN)
[36]	2013	SVM	RBF kernel; mapping to high-dimensional feature space	78% (SVM)
[34]	2021	K-nearest neighbors, Support Vector Machine, Random Forest, and XGBoost classifiers	Wristband PPG signal and basic physiological parameters; MFCC features extracted from 5s PPG segments; Hybrid FS method proposed to reduce input data size	99.93% (XGBOOST)
[37]	1993	Decision Tree	Decision tree logic	71.1% (decision tree)
[33]	2021	CNN	Data augmentation via variational autoencoder (VAE)	95.48% (CNN)
[32]	2024	SVM, RF, LR	Sparse autoencoder (SAE)	79% (SVM)
[38]	1988	ADAP Neural Network	Associative models for forecasting onset	76% (ANN)

Table 1.1: Comparative table of Diabetes Detection Studies

Recent studies on diabetes prediction have shown the effectiveness of both classical ML. Classical ML algorithms such as SVM, Random Forest, KNN, and Logistic Regression rely heavily on preprocessing steps including data cleaning, normalization, feature selection, and hyperparameter tuning. Using datasets such as the Pima Indians Diabetes Dataset, these approaches should achieved accuracies ranging from 78% to 98.8%.

Machine Learning methods, especially Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs), have further improved prediction performance by automatically learning complex patterns from data. These models have been applied to both medical images and structured clinical datasets, sometimes using techniques such as data augmentation and transfer learning to improve accuracy.

Overall, recent research shows that combining effective preprocessing, feature engineering, and advanced learning models leads to the best performance in diabetes prediction systems. Among the different approaches, ANN and XGBoost models frequently provide better prediction accuracy and reliability compared to traditional classification methods, making them among the most promising solutions for intelligent diabetes diagnosis.

10 Conclusion

In this chapter, we first presented the medical context of diabetes and the importance of Clinical Decision Support Systems (CDSS) in improving early diagnosis and healthcare assistance. We then introduced the fundamental concepts of machine learning, including its definition, main types, application domains, and classification algorithms. Among the studied techniques, neural network-based methods and XGBoost were highlighted as effective approaches for classification tasks. In this study, ANN and XGBoost were selected as the main models for diabetes prediction due to their strong performance compared to other classical machine learning methods used as baseline models.

Chapter 2

Design and Implementation of the Prediction System

1 Introduction

This chapter presents the design and implementation of the proposed diabetes prediction system, including coding environment, dataset, preprocessing and feature engineering steps, and the implemented models (SVM, Random Forest, XGBoost, and ANN).

2 Development Environment

The project is developed in Python 3.10+, leveraging its robust ecosystem for reproducible machine learning and data science.

2.1 Core Libraries

- **Scikit-learn 1.3:** Central framework for classical ML (SVM, RF, LR). Used for preprocessing (`StandardScaler`), pipeline management, and evaluation metrics (F1-score, Confusion Matrix).
- **XGBoost:** Employed for gradient-boosted decision trees, utilizing its built-in regularization and `scale_pos_weight` to handle structured tabular data efficiently.
- **Imbalanced-learn (imblearn):** Applied specifically for SMOTE over-sampling to address class imbalance via synthetic interpolation.
- **TensorFlow 2.x / Keras:** Used for the Sequential Neural Network, utilizing the Adam optimizer and high-level APIs for layer management.
- **Pandas & NumPy:** Essential for DataFrame manipulation, vectorized numerical operations, and data cleaning.
- **Matplotlib (+ Seaborn):** Used for generating diagnostic visualizations, including training

curves and feature distribution plots.

2.2 Computational Infrastructure

Experiments were conducted in an isolated virtual environment on Google Colab. Classical models were trained on CPU using `GridSearchCV` for hyperparameter tuning, while the Neural Network utilized NVIDIA CUDA (GPU) acceleration to reduce epoch iteration time. Environment consistency is maintained via a `requirements.txt` configuration.

3 General architecture

Figure bellow represents the general architecture of our ML system.(see figure 2.1)

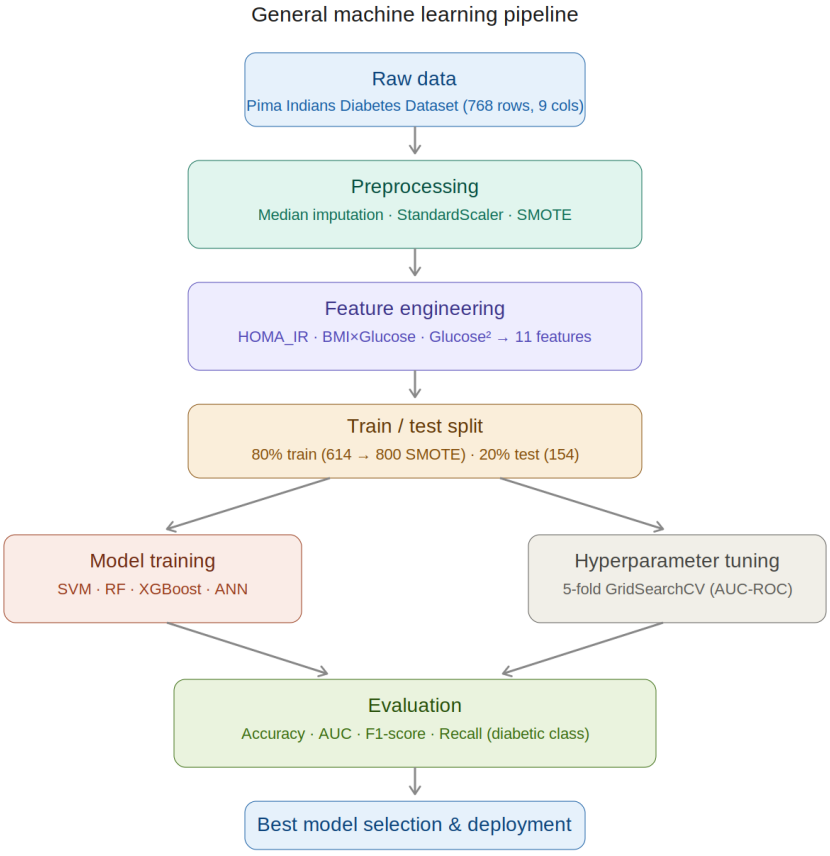


Figure 2.1: general machine learning pipeline (generated by canva)

4 Dataset Description

The Pima Indians Diabetes Dataset (sourced from NIDDK/UCI) is a benchmark for medical binary classification. It comprises 768 observations of female patients of Pima Indian heritage (age 21+) near Phoenix, AZ. The raw data contains 9 attributes (8 features, 1 target).

In fact, we have chosen to use this dataset because it is a canonical benchmark for binary medical classification. It is small enough to permit rapid experimentation and compari-

son across model families. It also has multi-dimensional attributes, and is realistic with the presence of moderate class imbalance and zero values.

4.1 Feature Definitions

- **Pregnancies:** Count of pregnancies; high gravidity is clinically linked to increased Type 2 risk.
- **Glucose:** 2-hour plasma glucose (mg/dL). Values > 140 mg/dL indicate glucose intolerance; this is the most significant diagnostic feature.
- **BloodPressure:** Diastolic pressure (mmHg); hypertension often correlates with insulin resistance.
- **SkinThickness:** Triceps skinfold thickness (mm) used as a body fat proxy for obesity-related resistance.
- **Insulin:** 2-hour serum insulin; elevated levels often precede a formal diabetes diagnosis.
- **BMI:** Body Mass Index (kg/m^2); values ≥ 30 represent obesity, a well-established risk factor.
- **DiabetesPedigreeFunction (DPF):** A continuous score quantifying hereditary predisposition based on family history.
- **Age:** Patient age in years; risk factors typically escalate after age 45.

4.2 Target Variable

The `Outcome` variable is a binary integer representing a diagnosis within five years: **0** for Non-diabetic with 500 cases (65.1%) and **1** for Diabetic with 268 cases (34.9%).

The moderate class imbalance requires metrics like F1-score or Precision–Recall rather than accuracy alone.

4.3 Sample Data (First 5 Rows)

Preg.	Glucose	BloodPr.	SkinTh.	Insulin	BMI	DiabPed.	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1

Table 2.1: First five rows of the Pima Indians Diabetes dataset

4.4 Summary Statistics

Feature	Count	Mean	Std	Min	25%	Median	75%	Max
Pregnancies	768	3.85	3.37	0.00	1.00	3.00	6.00	17.00
Glucose	768	120.89	31.97	0.00	99.00	117.00	140.25	199.00
BloodPressure	768	69.11	19.36	0.00	62.00	72.00	80.00	122.00
SkinThickness	768	20.54	15.95	0.00	0.00	23.00	32.00	99.00
Insulin	768	79.80	115.24	0.00	0.00	30.50	127.25	846.00
BMI	768	31.99	7.88	0.00	27.30	32.00	36.60	67.10
DiabetesPedigreeFunc.	768	0.47	0.33	0.08	0.24	0.37	0.63	2.42
Age	768	33.24	11.76	21.00	24.00	29.00	41.00	81.00
Outcome	768	0.35	0.48	0.00	0.00	0.00	1.00	1.00

Table 2.2: Descriptive statistics for all features in the dataset

5 Data Preprocessing & Feature Engineering

5.1 Handling Missing Values

The dataset contains “zeros” as missing values in 5 attributes. We have found 5 records in Glucose, 35 records in BloodPressure, 227 records in SkinThickness, 374 records in Insulin, and 11 records in BMI.

We have used median imputation as a replacement strategy. For each affected column, zero values were first converted to `NaN`, and subsequently replaced with the column’s median value computed from the non-zero observations.

5.2 Feature Engineering

After imputation, three clinically motivated engineered features were derived from the original 8, expanding the feature set to 11. These composite features capture non-linear physiological interactions that individual raw features cannot express independently. as you can see in table 2.3

5.3 Feature Normalisation

We have used the `StandardScaler` function for normalisation. It transforms each feature to have zero mean and unit variance:

$$z = \frac{x - \mu}{\sigma} \tag{2.1}$$

where μ is the column mean and σ is the column standard deviation.

Feature	Formula	Clinical Rationale
`HOMA_IR`	$\frac{\text{Glucose} \times \text{Insulin}}{405}$	Homeostasis Model Assessment of Insulin Resistance (HOMA-IR). A validated surrogate marker of peripheral insulin resistance; values > 2.5 indicate resistance.
`BMI_Glucose`	BMI × Glucose	Interaction term capturing the compounding metabolic effect of obesity and hyperglycaemia, two co-occurring risk factors that jointly elevate diabetes probability beyond either variable alone.
`Glucose_Sq`	Glucose^2	Quadratic term allowing models to capture the accelerating risk above the diagnostic glucose threshold (140 mg/dL), where the relationship with diabetes onset becomes non-linear.

Table 2.3: Engineered features derived from the original 8 clinical variables

5.4 Class Balancing

The original dataset exhibits a class imbalance of 65.1% / 34.9% (non-diabetic / diabetic). To prevent models from over-predicting the majority class, SMOTE (Synthetic Minority Over-sampling Technique) was applied exclusively to the training set after the train-test split.

After SMOTE, the training set is balanced to 400 samples per class (800 total), and the test set remains unmodified (154 samples, original distribution) to preserve unbiased evaluation.

5.5 Train-Test Split

The preprocessed dataset was partitioned into a training set (80%, 614 samples) and a test set (20%, 154 samples) using a fixed random seed (`random_state=42`) for reproducibility. SMOTE was applied to the training portion only, yielding a final training set of 800 samples.

6 Machine Learning Models

6.1 Presentation of the Selected Models

In our work, we have trained and evaluated 4 ML algorithms: ANN, SVM, a Random Forest Classifier, and an XGBoost gradient boosting classifier. All models were optimised via 5-fold `GridSearchCV` on the SMOTE-balanced training set and evaluated on the held-out, unmodified test set. A custom decision threshold was tuned per model to maximise the F1-score on the validation fold.

6.1.1 SVM

We have trained the SVM with the hyperparameters presented in the table below. A grid search over $C \in \{0.1, 1, 10, 100\}$ and $\gamma \in \{0.01, 0.1, 1\}$ was performed using AUC-ROC as the optimisation criterion.

Hyperparameter	Value
Kernel	RBF
C	10
γ	0.1
Probability	True
Random state	42

Table 2.4: SVM best hyperparameters selected by GridSearchCV

6.1.2 Random Forest Classifier (Enhanced)

A Random Forest ensemble was trained with the following experimental hyperparameters:

Hyperparameter	Value
n_estimators	300
max_depth	15
max_features	sqrt
min_samples_split	2
Best CV AUC	0.9648

Table 2.5: Optimised Random Forest hyperparameters selected by GridSearchCV

6.1.3 XGBoost Classifier

XGBoost (eXtreme Gradient Boosting) was trained with a grid of 108 hyperparameter combinations (540 total fits with 5-fold CV), making this the most exhaustive search in the pipeline.

Hyperparameter	Value
n_estimators	300
max_depth	6
learning_rate	0.05
subsample	1.0
colsample_bytree	0.8
Best CV AUC	0.9705

Table 2.6: XGBoost hyperparameters selected by GridSearchCV

6.1.4 ANN

We hae used a Sequential Multi-Layer Perceptron (MLP). The architecture is presented in table 2.7 and the training configurtion is presented in table 2.8 .

Layer	Configuration	Activation	Params
Input	Dense (128) + BatchNorm + Dropout (0.30)	ReLU	2,048
Hidden 1	Dense (64) + BatchNorm + Dropout	ReLU	8,512
Hidden 2	Dense (32) + BatchNorm + Dropout	ReLU	2,208
Hidden 3	Dense (16) + BatchNorm + Dropout	ReLU	592
Output	Dense (1)	Sigmoid	17
Total parameters			**13,377**
Trainable parameters			12,897
Non-trainable parameters			480

Table 2.7: ANN architecture with Batch Normalisation (11 input features)

Parameter	Configuration
Optimiser	Adam (lr = 0.001)
Loss function	Binary Cross-Entropy
Epochs	100+ with early stopping
Batch size	32
Validation split	20% of training set
Checkpoint	`best_ann.keras`

Table 2.8: ANN training configuration

7 Results and comparative analysis

7.1 Full Model Comparison Table

All models were evaluated on the same 154-sample test set. The results show a clear improvement in performance from SVM to ensemble methods and ANN models.

Model	Accuracy	AUC	F1	Recall (Diab.)	Notes
SVM (RBF)	84.42%	0.8969	0.7818	80.0%	Competitive non-linear baseline
Random Forest	88.31%	0.9461	0.8393	87.0%	Strong ensemble, validated features
XGBoost	89.61%	0.9556	0.8596	91.0%	Best classical model
ANN	**90.26%**	0.9254	**0.8673**	**91.0%**	**Best overall accuracy & F1**

Table 2.9: Unified performance comparison across all models on the 154-sample test set

The SVM achieved the lowest performance with an accuracy of 84.42% and a recall of 80%. Random Forest improved the results significantly, reaching 88.31% accuracy and 87% recall. XGBoost obtained the best performance among the classical machine learning models with 89.61% accuracy and an AUC of 0.9556.

The ANN achieved the best overall results, with 90.26% accuracy, an F1-score of 0.8673, and 91% recall for diabetic cases. This indicates that the neural network was more effective at identifying positive diabetes cases while maintaining balanced overall performance.

The results also demonstrate the importance of preprocessing and feature engineering. Techniques such as SMOTE and the engineered features (HOMA-IR, BMI×Glucose, and Glucose^2) improved the prediction performance of all models, especially for the diabetic class.

Overall, the study confirms that machine learning techniques can provide effective support for diabetes prediction, with ANN and XGBoost showing the most promising results for this dataset.

7.2 Feature Importance Analysis

Rank	Random Forest	XGBoost
1	Insulin	HOMA_IR
2	HOMA_IR	Insulin
3	BMI_Glucose	BMI_Glucose
4	SkinThickness	SkinThickness
5	Glucose_Sq	Age

Table 2.10: Top 5 important features identified by Random Forest and XGBoost

The feature importance analysis highlights the significant role of insulin-resistance-related features in diabetes prediction. Both Random Forest and XGBoost identified `Insulin`, `HOMA_IR`, and `BMI_Glucose` among the most influential predictors.

The results also confirm the effectiveness of the feature engineering process, as the engineered features consistently appeared in the top-ranked variables. In particular, the HOMA-IR index showed a strong relationship with diabetes prediction, which aligns with its clinical importance as an indicator of insulin resistance.

8 Conclusion

In this chapter, We implemented a diabetes prediction system using preprocessing, feature engineering, SVM, Random Forest, XGBoost, and ANN. ANN and XGBoost performed best, while SVM performed worst. Techniques like SMOTE, HOMA-IR, and BMI×Glucose significantly improved accuracy, confirming machine learning's effectiveness. The next chapter presents the system design, web application, and user interface.

Chapter 3

Web Application Development and Interface

1 Introduction

The main objective of the ClearSugar web application (see figure 2.1) is to provide an accessible and user-friendly platform for real-time diabetes risk prediction using machine learning models. The system allows users to enter clinical measurements and obtain immediate prediction results without requiring advanced technical knowledge.

As a clinical decision-support tool, the application transforms patient health data into interpretable diabetes risk predictions while providing a simple and transparent user interface suitable for healthcare, research, and educational purposes.

2 Technologies and Development Environment

Our system's implementation mainly relies on open-source technologies based on JavaScript, Python, and database systems. Table 3.1 presents the main technologies and tools used in each layer of the application.

Tier / Category	**Technology Asset**	**Operational Role & Application Context**
Frontend UI	React.js & Tailwind CSS	Builds a single-page reactive application layout using modern, utility-first component styling.
Animations	Framer Motion & Lucide	Drives entry animations, interface state transitions, and powers the iconographic layout.
Data Parsing	PapaParse & Recharts	Parses local client-side CSV files for batch actions and powers interactive comparison analytics.

Tier / Category	Technology Asset	Operational Role & Application Context
Backend Core	Node.js & Express.js	Configures a rapid, non-blocking asynchronous environment acting as the structural RESTful API controller.
Database Tier	MongoDB & Mongoose	Provides a scalable NoSQL document store utilising object-document mapping schemas for patient entities.
Security Suite	JWT & bcrypt	Orchestrates secure, stateless user session validation alongside reliable database password hashing.
ML Inference	Flask & Flask-CORS	Provisions an internal Python routing microservice to cleanly expose predictive pipelines as callable endpoints.
Data Science	Scikit-Learn & NumPy	Manages numerical preprocessing operations, standard feature scaling, and structures matrix data computations.

Table 3.1: Technology Stack Summary

3 System Architecture

The application uses a multi-layer client-server architecture to organise the system into separate components. As illustrated in figure 2.1, the system is divided into three main layers: the frontend interface, the backend application logic, and the machine learning prediction module. Communication between these layers is handled through HTTP requests, allowing the application to remain modular, efficient, and easy to maintain.

3.1 The Request–Response Cycle

When a user submits the eight clinical measurements via the interactive form, the request–response cycle executes as follows:

1. The user fills out the input form in the React frontend and clicks submit.
2. The frontend validates all eight fields before sending a POST request to the Express backend, embedding the user's JWT authentication token in the request headers.
3. The Express backend verifies the token, identifies the user's system role, and forwards the clinical inputs to the Flask ML server via an internal HTTP call.

4. Flask preprocesses the incoming inputs (performing zero-value imputation with the column median and applying standard scaling), runs the inference model, computes SHAP feature importances via permutation sampling, and returns the result as a JSON payload.

5. The Express backend saves the completed prediction record to MongoDB and sends the final payload back to the frontend, which renders the interactive result card.

To guarantee high availability, a built-in JavaScript fallback model resides directly within the Express backend. If the secondary Python ML server is unreachable, this fallback mechanism takes over processing to produce an identical output shape, ensuring the interface remains fully operational.

3.2 System Use-Case Configuration

The system accommodates two categories of platform actors — Standard Users and System Administrators — who share the same core predictive pipeline but differ in the depth of information exposed and administrative control. (see figure 3.1)

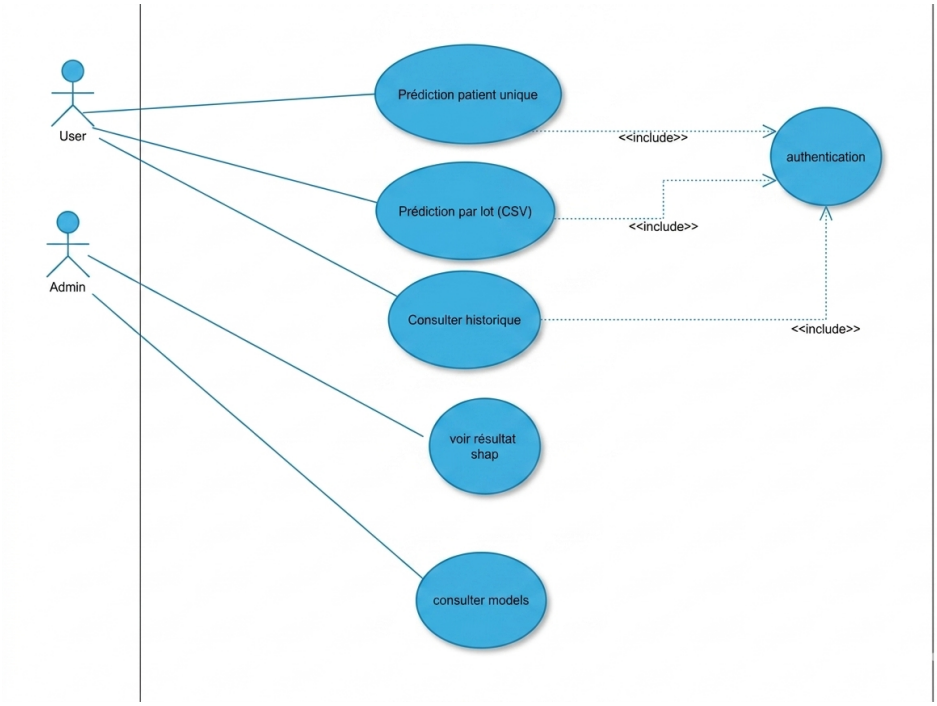


Figure 3.1: System UML Use-Case Diagram

3.3 Authentication Sequence Configuration

The authentication sequence accommodates two categories of platform actors—Standard Users and System Administrators—who share the same core login pipeline but differ in access level upon successful authentication. Both actors submit credentials to the System, which validates them against the Database by looking up the user by credentials and verifying the password hash. On success, a session token is issued; on failure, an error message is returned to the requester (see Figure 3.2).

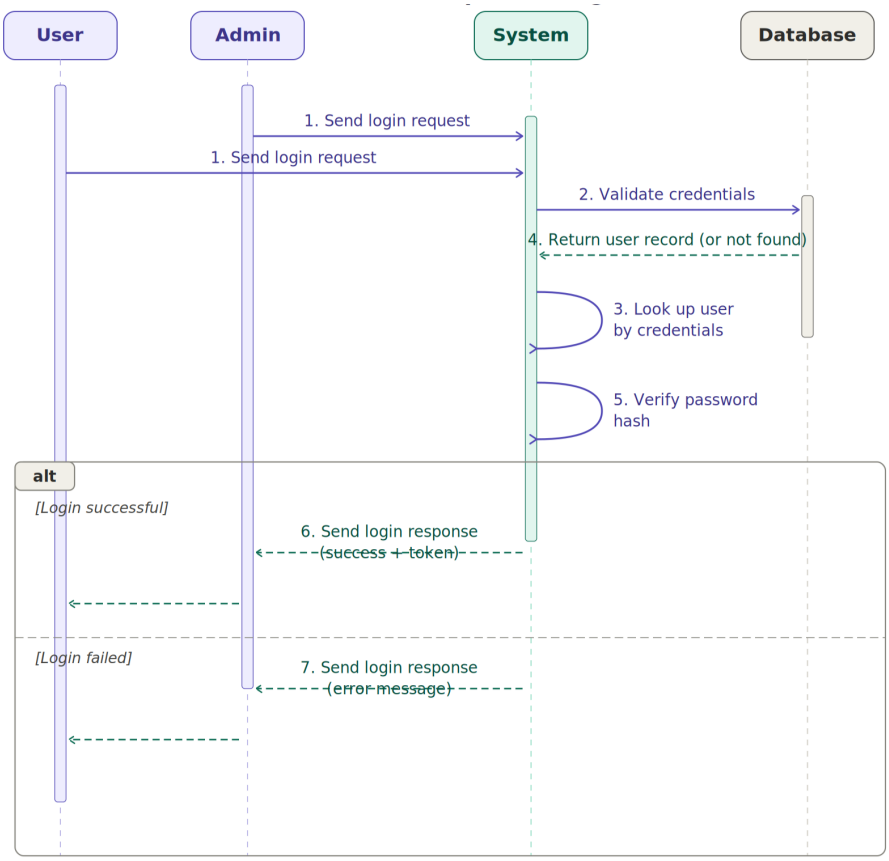


Figure 3.2: Authentication sequence diagram.

3.4 Prediction Sequence

The prediction sequence involves a `User` submitting patient data to the `System`, which then carries out internal processes to generate a result. Specifically, the `System` performs pre-processing on the input data followed by machine learning model inference to produce a prediction. Once calculated, the prediction result is stored in the `Database`, which confirms the storage action to the `System`. Finally, the `System` returns both the prediction result and the associated risk level to the `User` (see Figure 3.3).

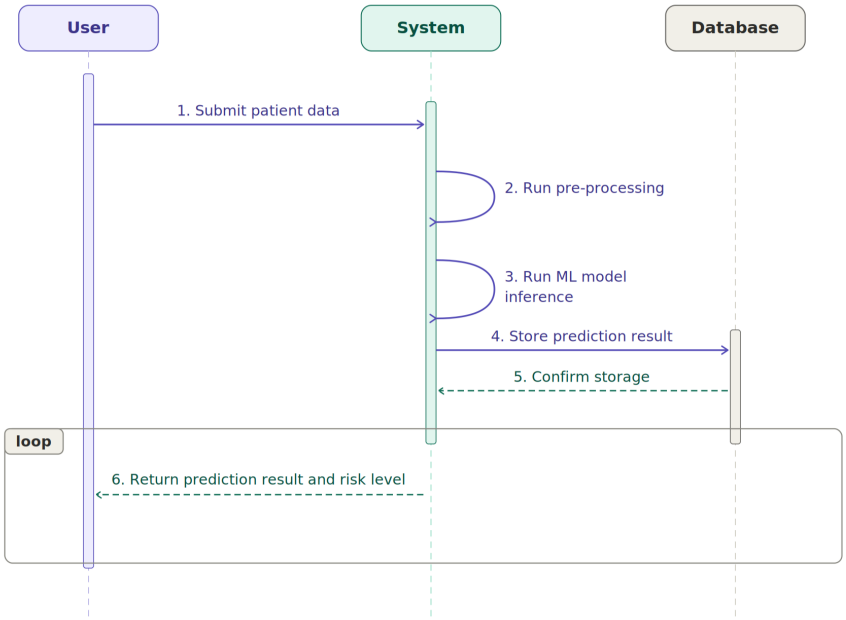


Figure 3.3: prediction sequence

3.5 Class Diagram

The class diagram defines the structural relationships between the core entities of the system. A `User` class encapsulates basic identity attributes and operations, from which the `Administrator` class inherits all attributes while extending them with model visibility via `voir_modèles()`. A `User` utilises zero or more `Prediction` instances (1 to 0..$*$), each storing a risk score, risk level, and classification result, and exposing `getPrediction()`, `getHistory()`, and `getSHAP()`. Each `Prediction` is composed of one `data_patient` record (1 to 0..$*$), which holds the eight clinical features validated through `validation()`, and optionally aggregates an `ML_Model` (0..1). The `Administrator` maintains a supervisory relationship by consulting the `ML_Model` (1..$*$ to 4), which exposes performance metrics and inference via `predict()`, `getMetrics()`, and `getFeatureImportance()` (see Figure 3.4).

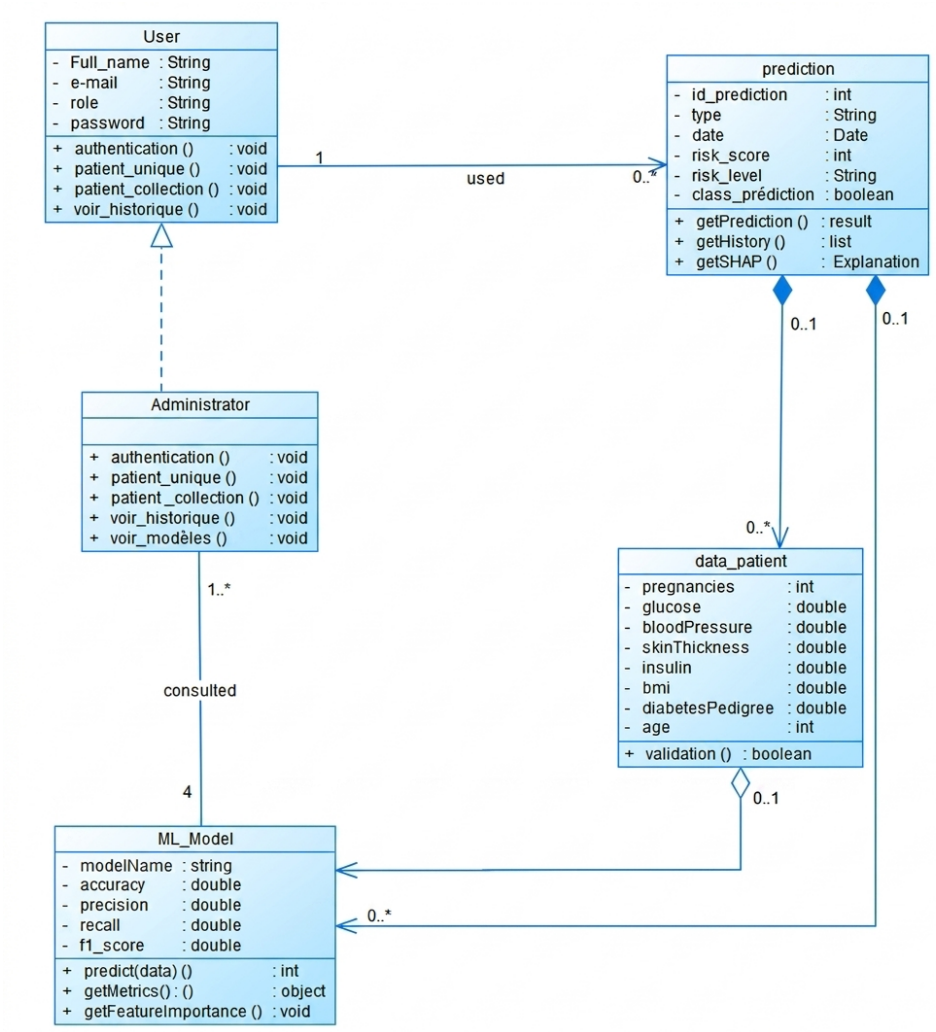


Figure 3.4: class diagram

4 Main Functionalities

4.1 Authentication Interface

The authentication gateway handles secure portal entry and registration utilising stateless operations.

- **Registration / Login:** New profiles choose credentials that are securely passed to the Express backend, where bcrypt generates a salt and hashes passwords before writing records to MongoDB.
- **Session Management:** Upon successful verification, the system creates a localised JSON Web Token (JWT) containing the user's explicit structural role (Standard or Admin). The frontend captures this signature token, appending it automatically into the HTTP header requests of all subsequent operational actions.

4.2 Prediction Interface

The Single Patient Prediction module presents a two-column data entry form encompassing all eight features defined in the Pima Indians Diabetes Dataset.

- **Contextual Helpers:** Every individual input form field incorporates interactive descriptive labels, real-world context placeholder targets, and tooltips establishing clear, normal physiological reference guidelines.
- **Hard Validation Processing:** Entries violating real-world physiological criteria — such as a blood plasma glucose value stretching past 300 mg/dL or body mass indexes climbing above 70 kg/m^2 — trigger a system block. The portal rejects submission and displays red error notices explaining the exact validation criteria violated.
- **Soft Validation Processing:** Unusual yet biologically possible metrics — such as an insulin level entry standing exactly at zero — trigger an orange warning notice. This flags the data anomaly for clinical awareness while allowing the user to proceed with the prediction submission.
- **Batch Analytics Mode:** Users can upload a bulk comma-separated values (CSV) file matching the eight-column structure directly into the app. Powered by PapaParse, the application reads the file records individually, runs backend inferences across all lines simultaneously, and shows the output in an aggregated, colour-coded summary grid stratified by risk level.

4.3 Prediction History

The history tracking system acts as a personalised clinical logging register.

- **Automated Persistence:** Every run passing through the processing pipeline is catalogued automatically. The Express tier pushes the patient's inputs, computed probability ratios, and time markers down to MongoDB before returning data to the screen.
- **Interactive Dashboard Grid:** Users can retrieve and review their complete personal evaluation timeline through a responsive history log table. Admin roles see an advanced view containing specialised information, including dedicated tracking tags detailing the specific classification model architecture deployed for each case.

4.4 Result Visualisation

Once the engine returns an inference response, the visualisation engine renders a dynamic, multi-layered information dashboard, as illustrated in Figure 3.5.

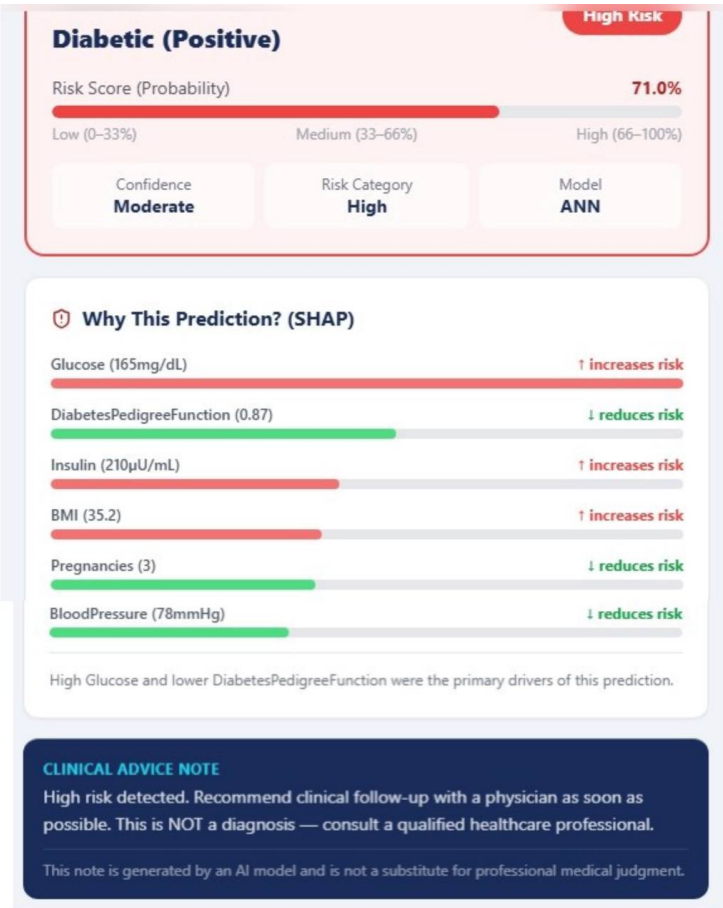


Figure 3.5: Conceptual Framework of the Dynamic Results Rendering Dashboard

- **Core Prediction Summary:** Renders the assigned classification status (*Diabetic* / *At Risk* / *Non-Diabetic*) alongside an animated circular progress gauge mapping the exact continuous probability percentage. A colour-segmented threshold bar visualises the risk level: green sections flag Low risk profiles (< 33%), amber zones represent Medium risk metrics (33–66%), and red spans highlight High risk cases (> 66%).
- **SHAP Explanation Graphics Panel:** Presents an ordered, sorted sequence layout of the eight patient attributes, organised by their absolute mathematical influence on the prediction. Every parameter displays its underlying value alongside custom coloured influence bars. Right-facing red bars labelled "↑ increases risk" indicate positive SHAP contributions, while left-facing green bars labelled "↓ reduces risk" indicate negative SHAP contributions.
- **Clinical Advice & Medical Disclaimer:** Generates localised text advising users on standard medical follow-up paths tailored to their specific risk category, combined with an explicit disclaimer stating that the system provides automated statistical risk tracking rather than a formal medical diagnosis.
- **Admin Performance Insights:** For authenticated administrators, the summary card expands to reveal a performance breakdown, exposing precise model training characteris-

tics (Accuracy, Precision, Recall, F1) alongside an interactive confusion matrix grid showing true/false positive and negative distribution splits.

5 Database Design

The persistence layer utilises a NoSQL document-oriented model via MongoDB to manage system accounts and historic patient data. Figure 3.6 shows the database architecture and schema document relationship Map.

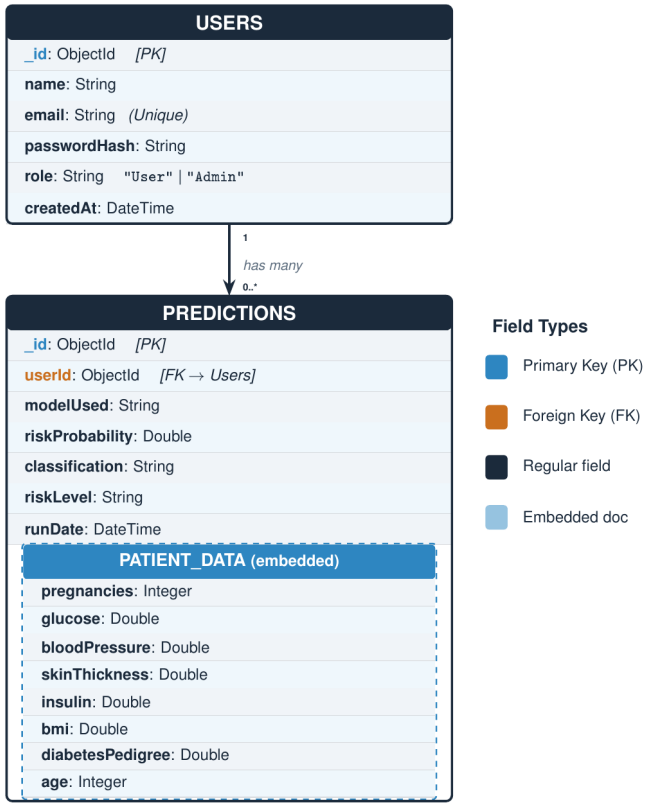


Figure 3.6: Database Architecture and Schema Document Relationship Map

5.1 Document Schema Specifications

5.1.1 Users Document Schema

Stores application user access privileges and login profiles.

- **_id** (ObjectId, Primary Key): Unique identifier generated automatically by MongoDB.
- **name** (String): Full name of the user or clinician.
- **email** (String): Unique electronic mail identity used for login validation routing.
- **passwordHash** (String): Salted and secured cryptographic password string generated via bcrypt.

- **role** (String): System access control variable restrictively set to either “User” or “Admin”.

5.1.2 Predictions Document Schema

Tracks individual prediction runs and links them to the generating user profile.

- **_id** (ObjectId, Primary Key): Unique prediction transactional ledger record ID.
- **userId** (ObjectId, Foreign Key Reference): Structural link back to the originating identity in the Users collection.
- **modelUsed** (String): Specifies the ML architecture variant used (e.g., “ANN”, “XGBoost”).
- **riskProbability** (Double): The raw decimal inference score generated by the model.
- **classification** (String): Resolved categorical classification outcome (e.g., “Diabetic”).
- **riskLevel** (String): Resolved colour-coded tier (“Low”, “Medium”, “High”).
- **runDate** (DateTime): System timestamp recorded at execution.

5.1.3 Patient Data Embedded Schema

An embedded object structure containing the eight specific clinical metrics analysed by the models.

- **pregnancies** (Integer): Number of historical pregnancies recorded for the patient.
- **glucose** (Double): Plasma glucose concentrations captured across an oral tolerance exam.
- **bloodPressure** (Double): Diastolic blood pressure reading in mmHg.
- **skinThickness** (Double): Triceps skinfold thickness measured in millimetres.
- **insulin** (Double): Serum insulin concentration expressed as μU/mL.
- **bmi** (Double): Body Mass Index calculated as weight in kilograms divided by height in metres squared (kg/m^2).
- **diabetesPedigree** (Double): Hereditary pedigree score mapping family diabetes history profiles.
- **age** (Integer): Patient chronological age measured in years.

6 Discussion

6.1 Strengths and Usability

The application provides a user-friendly interface that allows users to perform real-time diabetes prediction without requiring advanced technical knowledge. The system combines a

web interface, a backend server, and machine learning models to create an accessible clinical decision-support tool.

Several features improve usability, including clear input forms, prediction results, risk visualisation, and prediction history management. The application also integrates explainability tools to help users better understand the prediction results.

In addition, the separation between the frontend, backend, and machine learning modules improves system organisation, maintainability, and reliability.

6.2 Limitations

Despite its operational strengths, the application has several notable limitations that affect its immediate real-world utility:

- **Dataset Size and Diversity:** The underlying machine learning model is trained exclusively on the Pima Indians Diabetes Dataset, which contains only 768 records restricted to women of Pima Native American heritage. This narrow focus limits the model's ability to generalise accurately to males, younger demographics, or broader global populations with different genetic and lifestyle risk factors.
- **Imbalanced Training Data:** The dataset exhibits moderate class imbalance, with 500 non-diabetic records versus 268 diabetic entries. This imbalance inherently biases single-train split models toward the majority class.
- **Clinical Validation and Compliance:** The platform operates as an uncertified academic proof-of-concept. It has not undergone external validation using separate, real-world clinical cohorts, nor has it been evaluated by institutional review boards or regulatory health authorities. To comply with responsible AI guidelines, explicit medical disclaimers must be prominently displayed across all application views.

7 Conclusion

This chapter presented the design and implementation of the ClearSugar web application for diabetes prediction. The system integrates machine learning models into a user-friendly web platform capable of providing real-time predictions and storing prediction history.

The developed application demonstrates how machine learning techniques can be combined with modern web technologies to create practical and accessible medical decision-support tools. Although some limitations remain, the application represents a functional and promising solution for intelligent diabetes screening.

General Conclusion

At the end of this project, we conducted a study on diabetes prediction using Machine Learning techniques applied to the *Pima Indians Diabetes Dataset*. The main objective was to design an intelligent system capable of assisting medical diagnosis by predicting diabetes from clinical indicators.

First, we studied the medical context of diabetes, the challenges of traditional diagnosis methods, and the role of Clinical Decision Support Systems (CDSS) in modern healthcare. We then introduced the fundamental concepts of Machine Learning, the main classification algorithms, and the evaluation metrics commonly used in medical prediction tasks.

Several models were designed and evaluated, including Support Vector Machine (SVM), Random Forest, XGBoost, and Artificial Neural Network (ANN). Before training, multiple preprocessing techniques were applied, such as handling missing values, feature normalization, class balancing using SMOTE, and feature engineering.

The experimental results demonstrated that Machine Learning techniques can provide effective support for diabetes prediction. Among all models, the ANN achieved the best overall performance with 90.26% accuracy and 91% recall for diabetic cases, while XGBoost achieved the best performance among the classical Machine Learning models. The results also highlighted the importance of feature engineering, especially the HOMA-IR and BMI–Glucose features, which significantly improved prediction performance.

In addition to the predictive models, a complete and user-friendly web application was developed to make the system accessible and practical for users. The application provides several functionalities, including patient data input, real-time diabetes prediction, result visualisation, and an intuitive interface designed to simplify interaction with the prediction system.

Despite these promising results, some limitations remain. The study was conducted using a relatively small and specific dataset, and several important medical and lifestyle variables were unavailable. In addition, the use of a single 80/20 train-test split limits the robustness of the evaluation.

As future work, it would be interesting to use larger and more diverse medical datasets, apply cross-validation techniques, integrate additional clinical features, and explore more advanced deep learning architectures. Developing a real-world intelligent medical web application and integrating explainability methods such as SHAP or LIME would also improve the practical usability of the system.

In conclusion, this project demonstrates that combining careful preprocessing, feature engineering, modern Machine Learning techniques, and a practical web application can significantly improve diabetes prediction and contribute to more effective medical decision-support systems.

Bibliography

[1] WHO. *Classification of Diabetes Mellitus 2019*. Geneva.

[2] Chrysoula Boutaria, Michael A. Hill, and Christos S. Mantzoros. *PDB-101: Diabetes Mellitus: Insulin Receptor*.

[3] Roshan Kumar, Purabi Saha, Yogendra Kumar, Soumitra Sahana, Anubhav Dubey, and Om Prakash. *A Review on Diabetes Mellitus Type 1 and Type 2*.

[4] J. Pediatr. (Rio J.), 88(1), Feb 2012. *Mechanisms and Factors Associated with Gastrointestinal Symptoms in Patients with Diabetes Mellitus*.

[5] Tracey Long. *Diabetes Type 2: Nothing Sweet About It*. ATrain Education.

[6] World Health Organization. 2023. Diabetes Fact Sheet. WHO

[7] American Diabetes Association. 2024. Standards of Care in Diabetes—2024. ADA.

[8] Hawa MI, Kolb H, Schloot N, et al. Adult-onset autoimmune diabetes in Europe is prevalent with a broad clinical phenotype. *Diabetes Care*. 2013;36:908–13.

[9] Dabelea D, Rewers A, Stafford JM, et al. Trends in the prevalence of ketoacidosis at diabetes diagnosis: the SEARCH for diabetes in youth study. *Pediatrics*. 2014;133:e938–45.

[10] Roche Diagnostics. *Artificial Intelligence in Clinical Decision Support*, 2023.

[11] *Deep Learning and Big Data in Healthcare: A Double Review for Critical Beginners*, 2021.

[12] *Deep Learning for Early Diagnosis of Chronic Conditions Using Electronic Health Records*, 2022.

[13] *The Convergence of High-Dimensional Data and Algorithmic Intelligence*, 2023.

[14] *Scalable and Accurate Deep Learning with Electronic Health Records*, 2018.

[15] *Effectiveness of AI in Healthcare: Costs vs Impact*, 2024.

[16] Dave Bergmann. *AI Model*. IBM Think. August 18, 2025. `https://www.ibm.com`

[17] MDPI. *Theory and Practice of Integrating Machine Learning and Conventional Statistics in Medical Data Analysis*, 2022.

[18] *Applied Logistic Regression*, 2013.

[19] ResearchGate. *A Comparative Study of Traditional Statistical Models vs. AI in Predicting Health Outcomes*, 2022.

[20] *Machine Learning and Data Mining in Diabetes Diagnosis and Treatment*, 2021.

[21] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

[22] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.

[23] Liaw, A., et al. (2018). *randomForest: Random Forests for Classification and Regression*.

[24] Hand, D. J., & Yu, K. (2001). Idiot's Bayes — not so stupid after all? *International Statistical Review*, 69, 385–398.

[25] *Demystifying Statistics and Machine Learning in Analysis of Structured Tabular Data*, 2022.

[26] Gunning, D., & Aha, D. (2019). DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine*, 40(2), 44–58.

[27] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of KDD 2016*, pp. 1135–1144.

[28] Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

[29] European Commission. (2021). Proposal for a Regulation on Artificial Intelligence (AI Act). COM(2021) 206 final.

[30] Arrieta, A. B., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115.

[31] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Scientific Reports*, vol. 12, no. 1, p. 5979, 2022.

[32] *Al-Andalus Journal for Applied Sciences*, 2024, Vol 10, Issue 17, p. 24.

[33] SECOMUCI Research Groups, Escuela de Ingenierías Industrial e Informática, Universidad de León, Campus de Vegazana s/n, León C.P. 24071, Spain.

[34] Received 30 March 2021, Revised 18 July 2021, Accepted 18 July 2021, Available online 22 July 2021, Version of Record 28 July 2021.

[35] Vellore Institute of Technology, Chennai, India.

[36] Available online 4 May 2026, Version of Record 4 May 2026.

[37] The Department of Cyberspace Institute of Advanced Technology, GuangZhou University, GuangZhou 510006, China.

[38] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications in Medical Care*, 1988, pp. 261–265. IEEE Computer Society Press.

[39] Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.*