

11

Diabetes Prediction Model - Dissertation.docx

 My Files My Files University

Document Details

Submission ID trn:oid:::17268:91006693**Submission Date****Apr 13, 2025, 10:57 AM GMT+5:30****Download Date****Apr 13, 2025, 11:01 AM GMT+5:30****File Name****Diabetes Prediction Model - Dissertation.docx****File Size****715.0 KB****36 Pages****9,888 Words****58,322 Characters**

Page 1 of 38 - Cover Page

Page 2 of 38 - AI Writing Overview

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

*% detected as AI

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Disclaimer

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.



What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

2025

Diabetes Prediction Dissertation

[DOCUMENT SUBTITLE]
K214714 MUHAMMAD HASNAIN

Contents

Chapter 1: Introduction	4
1.1 Background of the Study	5
1.2 Problem Statement	5
1.3 Aim and Objectives	6
1.4 Research Questions	6
1.5 Scope of the Project	6
1.6 Significance of the Study	7
Chapter 2: Literature Review	8
2.1 Understanding Diabetes	8
2.1.1 Definition of Diabetes	8
2.1.2 Types of Diabetes	8
2.1.3 Causes and Risk Factors	9
2.1.4 Symptoms and Diagnosis	9
2.2 Machine Learning in Healthcare	9
2.2.1 Role of AI and ML in Disease Prediction	10
2.2.2 Success Stories of ML in Healthcare	10
2.2.3 Challenges in Implementing AI-Based Prediction Systems	11
2.3 Existing Diabetes Prediction Models	12
2.3.1 Traditional Methods of Diabetes Diagnosis	12
2.3.2 Review of ML Models for Diabetes Prediction	13
2.3.3 Strengths and Weaknesses of Current Approaches	13
2.4 Datasets Used for Diabetes Prediction	14
2.4.1 Pima Indian Diabetes Dataset	14
2.4.2 Other Publicly Available Datasets	15
2.4.3 Data Collection Challenges	16
Chapter 3: System Design and Methodology	17
3.1 System Overview	17
3.1.1 System Architecture	17
3.1.2 High-Level Workflow	18

3.2 Data Collection and Preprocessing	19
3.2.1 Data Sources	19
3.2.2 Data Cleaning	19
3.2.3 Feature Engineering	20
3.2.4 Data Normalization and Scaling	20
3.3 Machine Learning Models Used	22
3.3.1 Logistic Regression	22
3.3.2 Random Forest	23
3.3.3 Support Vector Machine (SVM)	24
3.4 Performance Evaluation Metrics	24
3.4.1 Accuracy	25
3.4.2 Precision, Recall, F1-Score	25
Chapter 4: Implementation	26
4.1 Data Splitting and Training.....	26
4.1.1 Train-Test Split	26
4.1.2 Cross-Validation	26
4.2 Model Training and Optimization	27
4.2.2 Feature Selection Techniques	27
4.2.3 Overfitting and Regularization	28
4.3 Comparison of Model Performance	28
4.3.1 Performance Metrics Across Models	28
4.3.2 Strengths and Weaknesses of Each Model	29
4.3.3 Best Performing Model.....	29
4.4 Deployment Strategy	29
4.4.1 Web-Based Deployment	29
4.4.2 Mobile Application Integration	30
4.4.3 Cloud Deployment (AWS, Google Cloud)	30
Chapter 5: Results and Discussion	31
5.1 Model Performance Analysis	31
5.1.1 Model Accuracy and Comparisons	31

5.2 Case Studies and Real-World Applications	31
5.2.1 Application in Hospitals	31
5.2.2 Use by General Practitioners	31
5.2.3 Potential for Remote Health Monitoring	32
5.3 Challenges Faced During Implementation	32
5.3.1 Data Quality Issues	32
5.3.2 Model Overfitting	32
5.3.3 Computational Complexity	32
5.4 Ethical and Legal Considerations	33
5.4.1 Patient Data Privacy	33
5.4.2 Bias in Machine Learning Models	33
5.4.3 Regulatory Compliance (GDPR, HIPAA)	33
Chapter 6: Conclusion and Future Work	34
6.1 Summary of Findings	34
6.2 Limitations of the Study	34
6.3 Recommendations for Improvement	34
6.4 Future Research Directions	34
References	35
 Table of Figures	
Figure 1	17
Figure 2	21
Figure 3	22
Figure 4	23
Figure 5	23
Figure 6	24
Figure 7	24

Chapter 1: Introduction

1.1 Background of the Study

Diabetes is a chronic metabolic disease characterized by high blood sugar levels. Diabetes has become a major health issue worldwide, affecting millions of people and being one of the top killers and causes of morbidity. According to WHO, by 2030, diabetes will become the seventh leading cause of death worldwide. There are two types of diabetes, mostly classified as Type 1 and Type 2. Type 1 is an autoimmune disease in which the immune system attacks the body's insulin-producing pancreatic cells, and Type 2 is characterized by resistant insulin reaction or inadequate production of insulin by the body.

Considering diabetes detection needs to be considered early with many reasons, it is the management of the disease before complications develop. Early detection prevents severe impairment of life, including complications such as heart ailments, kidney failure, blindness, and amputation. Traditionally, for diabetes diagnosis, there has always been a requirement for some clinical tests: the fasting blood glucose test, glucose tolerance test, and random blood sugar tests, which are time-consuming as well as costly.

The rise of advanced technologies such as machine learning (ML) has thereby generated a lot of interest in designing systems that will be able to automatically predict diabetes. Such systems are aimed towards being economical and faster compared to traditional diagnostic systems to ensure early identification of people at risk and start their intervention treatment.

1.2 Problem Statement

Diagnosing diabetes in its early stages poses a challenge for health care practitioners. Until the patient begins to show overt symptoms, he or she is often unaware of being diabetic-hence by that time, the disease may be inflicting irreversible damage. Often, the identification of diabetes is made once the patient manifests complications, including heart disease or kidney failure; these conditions render management extremely difficult.

There is an urgent need to set up an automated prediction system for diabetes, catering to the needs of all healthcare providers, that will help bridge the gap between early symptoms and a full-fledged medical diagnosis. Such a system, integrated with primary healthcare services, would promote earlier detection and save lives and healthcare resources. Machine learning models will enable the analysis of large datasets to find patterns or risk factors that may not be easily achieved by traditional methods.

That, however, poses the challenge of precise automated systems. Even though machine learning models have promises, diabetes prediction accuracy may be difficult due to its complex and multifactorial nature, along with genetic predisposition, environmental conditions, and lifestyle. These factors plus quality and availability of the data will determine the performance of these

models. To account for several factors, a sufficiently strong system must emerge to make reliable predictions.

1.3 Aim and Objectives

Aim

This study aims to develop a machine learning model for predicting an individual's probability of developing diabetes based on a set of clinical parameters. The model will be applied to a dataset and tested for its performance in predicting if a person is diabetic or not.

Objectives

The objectives of the study are as follows:

1. **Data Collection and Preprocessing:** Gathering the Pima Indian Diabetes dataset and cleaning the data by dealing with missing values and outliers. Thinking of further normalizing the data to ensure that all features are consistent.
2. **Model Training:** Several machine learning models will be trained-both logistic regression and decision trees; random forest and support vector machines (SVMs), to predict diabetes.
3. **Model Evaluation:** Evaluate the various performance metrics of the models, like accuracy, precision, recall, F1-score, and ROC-AUC.
4. **Deployment:** The model is then best performing and will be deployed as a web application, through which users will input their data and receive a predicted value on their diabetic status.

1.4 Research Questions

The study seeks to answer the following research questions:

1. **Can machine learning models accurately predict diabetes?**
 - This question is to check if machine learning can be used in diabetes prediction. Can the algorithms learn from historical data and effectively predict cases yet to be seen?
2. **What features are the most indicative of diabetes risk?**
 - This is important as to which features (such as glucose levels, BMI, age, and insulin levels) can determine the prediction of diabetes, as it helps refine the model to focus on the most predictive health indicators.

1.5 Scope of the Project

The present project tends to the task of formulating and validating machine learning models for diabetes prediction with the instance of the Pima Indian Diabetes dataset. The limitation to the dataset is given that it is publicly available and widely used for diabetes prediction purposes. The dataset contains health-related metrics of glucose, blood pressure, age, BMI, and medical attributes connected with diabetes risk. The methodologies developed could be used in conjunction with this data, however, in principle, they could be applied to similar datasets and incorporated for extensive use in real clinical settings.

1.6 Significance of the Study

The study will assist in improving the early prediction and will thus add value to general health. This automated prediction system can help with the early diagnosis of diabetes by healthcare professionals, enabling timely interventions to prevent complications. In addition, this study propagates research in the applied intersection between healthcare and artificial intelligence to show how machine learning can be used in the elevation of healthcare delivery. Targeting possibly one of the most equipped chronic diseases around the world, this system may come in handy for reducing costs on healthcare and improve management of the disease, and most importantly reduces the number of complications and deaths due to diabetes.

Chapter 2: Literature Review

2.1 Understanding Diabetes

Diabetes mellitus is a global health dilemma affecting millions of people throughout the globe. It is a chronic metabolic disorder characterized by persistent hyperglycemia due to inability of the body to produce insulin in sufficient quantity, efficaciously use insulin, or both (NIH, 2025; WHO, 2025). Insulin is a hormone secreted by the pancreas; it plays a profound role in controlling blood glucose levels by allowing the absorption of glucose into the cells for energy production. If this mechanism fails, glucose builds up in the bloodstream, thereby causing a cascade of complications ranging from mild cardiovascular diseases through kidney failure, neurological impairment, all the way to blindness (CDC, 2025; American Heart Association, 2025). An understanding of diabetes entails a discussion of what is it, types, causes, risk factors, symptoms, and diagnostic criteria.

2.1.1 Definition of Diabetes

Diabetes mellitus is generally defined as a metabolic disorder arising from an abnormality in insulin secretion and/or action. It is a progressive and chronic condition that requires management often for a lifetime to prevent major complications (WHO, 2025). The NIH classifies diabetes types into discrete entities based on their varied pathophysiologies and etiologies. In the opinion of the WHO, diabetes is rather a collection of diseases with various presentations and outcomes than a singular disease entity. Hyperglycemia, the commonest denominator of all diabetes types, may cause acute complications such as diabetic ketoacidosis or chronic complications affecting several organ systems (CDC, 2025).

2.1.2 Types of Diabetes

Type 1 Diabetes

In Type 1 diabetes, the autoimmune destruction of pancreatic β -cells that produce insulin leads to an absolute deficiency of insulin in the body (NIH, 2025; CDC, 2025). Type 1 diabetes is usually diagnosed during childhood or adolescence but can occur at any age. People with Type 1 diabetes depend on exogenous insulin for their very survival (American Heart Association, 2025). Genetic susceptibility and environmental factors, thought to include viral infections, may trigger the autoimmune attack on β -cells in Type 1 diabetes (CDC, 2025).

Type 2 Diabetes

Type 2 diabetes is the most common type of diabetes in the world, constituting roughly 90-95% of all cases (CDC, 2025). It is characterized by insulin resistance and progressive β -cell inadequacy. Unlike type 1 diabetics, the type 2 diabetic might have produced enough insulin initially, but the body does not respond to it (WHO, 2025). Obesity, inactivity, age, and genetic loading are risk factors for Type 2 diabetes (NIH, 2025). It has an insidious onset and may remain undetected for years, since it is mostly asymptomatic until its later stages (American Heart Association, 2025).

Gestational Diabetes

Gestational diabetes mellitus develops during pregnancy, when the placenta secretes hormones that induce insulin resistance in the mother (Hopkins Medicine, 2025). Women with GDM are at a higher risk for Type 2 diabetes later. Moreover, their offspring will be more susceptible to metabolic disorders later in their life (NIH, 2025). GDM usually resolves shortly after delivery but requires monitoring during pregnancy to minimize adverse outcomes for both mother and child (CDC, 2025).

Other Types

Other types of diabetes include milder diabetes or prediabetes and monogenic diabetes. Prediabetes can describe someone whose blood sugar is higher than normal but not high enough to be diagnosed as diabetes; in this way, it raises the warning flag for people for developing Type 2 diabetes and cardiovascular diseases later (NIH, 2025). Monogenic diabetes occurs through mutations within one gene affecting either insulin production or function, and it is much rarer than Type 1 or Type 2 diabetes (WHO, 2025).

2.1.3 Causes and Risk Factors

There are different causes of diabetes. Most causative agents of Type 1 diabetes are autoimmune destruction of pancreatic β -cells because of genetic predisposition and environmental factors such as viral infection (CDC, 2025; WHO, 2025). On the contrary, Type 2 is associated with genetics and several lifestyle factors such as obesity and lack of physical activity (NIH, 2025; WHO, 2025). It produces hormonal alterations during pregnancy that inhibit insulin action (Hopkins Medicine, 2025). The reasons behind the fast increase in such a way that everyone else is rising in prevalence have much more contributed by urbanization and lifestyle changes (WHO, 2025).

2.1.4 Symptoms and Diagnosis

Some of the general symptoms are: increased thirst (polydipsia), increased frequency of urination (polyuria), severe ravenous hunger (polyphagia), unexplained weight loss, fatigue, blurred vision, slow-healing wounds, and recurrent infections (NIH, 2025; WHO, 2025). These symptoms occur due to chronic hyperglycemia affecting the functions of many systems within the body. Diagnostics include blood glucose determination by appropriate tests such as fasting plasma glucose (FPG), oral glucose tolerance test (OGTT), and glycated hemoglobin (HbA1c) levels. The diagnostic thresholds used as per the American Heart Association (2025) include $FPG \geq 126$ mg/dL or $HbA1c \geq 6.5\%$. Screening should be early for individuals with risk factors such as obesity or a family history of diabetes to enable timely intervention (CDC, 2025).

2.2 Machine Learning in Healthcare

Machine Learning (ML) is one of the components of artificial intelligence (AI) and it has been a metaphor for towards a revolution in the diagnosis prediction, treatment personalization, and data management improvement in medical sciences. ML algorithms examine enormous amounts of

structured and unstructured data and can locate and uncover patterns and correlations unable to be discovered by human ability. These all will build an infrastructure for healthcare by improving accuracy, efficiency, and patient outcome in primary and preventive settings (Coursera, 2025; Siemens Healthineers, 2025).

2.2.1 Role of AI and ML in Disease Prediction

Artificial intelligence and machine learning could play a major role in predicting diseases by analyzing huge datasets composed of various demographic factors, medical history, laboratory results, imaging data, and genetic profiles. Such algorithms are able to show how some independent factors such as blood glucose, body mass index (BMI), lipid profiles, and certain lifestyle behaviors are nonlinear in the detection of disease risk far superior to traditional approaches (Coursera, 2025; PMC, 2023). For instance:

- **Early Detection:** ML algorithms can also forecast chronic diseases like diabetes and cardiovascular diseases years before the occurrence of clinical symptoms. The subtle changes in biomarkers and lifestyle data are so subtle that they remain unnoticed by normal people (Siemens Healthineers, 2025).
- **Risk Stratification:** Supervised learning models categorize patients into risk categories based on the likelihood of diseases. For instance, random forest classifiers have been used to predict Type 2 diabetes with high accuracy by integrating personal lifestyle information with clinical data (PMC, 2023).

Moreover, AI-based different predictive models have incorporated laboratory workflow systems into routine test results demographics to determine individualized probability scores for patients. Therefore, the area of risk can be flagged by the physicians before it manifests as symptoms (Siemens Healthineers, 2025).

2.2.2 Success Stories of ML in Healthcare

Due to certain limitations within the current literature, there are scant examples pertaining to the diabetes prediction model; however, the success of ML applications in healthcare shows its ability to transform parameters:

- **COVID-19 Severity Prediction:** The Siemens Healthineers AI-based COVID-19 Severity Algorithm was developed from deidentified data from more than 14,500 patients. The model predicts ventilator needs as well as risk of organ damage and mortality with very good accuracy based on laboratory values and demographic information (Siemens Healthineers, 2025).
- **Radiology and Imaging:** Deep-learning techniques, such as CNNs, are nowadays broadly applied for disease identification into medical images. For example, CNNs successfully identified diabetic retinopathy in retinal scans and also distinguished chest radiographs due to bacterial pneumonia (PMC, 2023).

- **Post-Stroke Pneumonia Prediction:** A deep neural network model developed by Ge et al. predicted post-stroke pneumonia with an AUC of 92.8% and 90.5% on days 7 and 14 respectively, thus confirming its utility in critical care (PMC, 2021).
- **Personalized Medicine:** Algorithms of machine learning are capable of personalizing treatment planning through predictive analysis of patient relevant data like genetic profile, patient drug history, and responses regarding drug therapy. These possible avenues in the discovery of new drugs and providing clinical trials for CNS disorders are awaited by pharmaceutical companies (Built In, 2024).

In summary, these examples show how machine learning has gotten successes in different areas of healthcare, most notably in increasing diagnostic accuracy, improving treatment considerations and outcomes for patients.

2.2.3 Challenges in Implementing AI-Based Prediction Systems

Even if the prediction systems within health care premises are embedded in AI paradigms, implementation for such systems has yet to face many challenges.

1. **Data Quality and Availability:** The quality and diversity of training datasets determine the efficiency of all ML model applications. Models may malfunction in providing inaccurate prediction or poor performance under certain clinical settings by having incomplete datasets or having biased datasets (PMC, 2023; Coursera, 2025). For instance, if the datasets overrepresent a certain ethnic group, the predictive model may fail to generalize its predictions regarding the broader population (Siemens Healthineers, 2025).
2. **Model Interpretability:** The black box nature of many AI/ML algorithms is another major barrier that prevents clinicians from using these systems. Prediction made with such algorithms will not explain how it was derived, leading to little acceptance or use among health care professionals (PMC, 2023; WHO, 2025).
3. **Ethical Concerns:** Algorithmic bias and privacy threats of patient information cause substantial barriers to adoption widespread (Coursera, 2025; PMC, 2023). However, some regulatory frameworks like that of the EU's AI Act ensure transparency and fairness in trying to bring these issues to the surface.
4. **Integration with Clinical Workflows:** Modern-day health systems must hence be able to integrate predictive models into their EHRs and laboratory workflows, a feat complicated by the hodgepodge of standards set across institutions already (Siemens Healthineers, 2025).

Future Directions

Overcoming these hurdles and making the most of the promise that AI-Machine Learning has for healthcare:

- **Federated Learning:** Collaborative model training across institutions without sharing sensitive patient data is a potential solution to privacy that would still improve model rigor with rare or underrepresented disease populations (PMC, 2023).
- **Hybrid Models:** The combination of algorithms such as random forests and reinforcement learning can produce better accuracy of prediction using both of their strongholds-Machine Learning will be richer (PMC, 2023).
- **Integration of Wearable Devices:** Real-time monitoring systems based on IoT with relevant forms of ML algorithms may offer continuous viewing of all health parameters for possible early identification and intervention in chronic conditions such as diabetes or hypertension (PMC, 2023).
- **Advanced Biomarkers:** As such, considering genomic profiling and proteomic data makes even a further refined assessment of risk-mobilizing metabolite biomarkers; cardiovascular risk has recently been predicted with improved forecasting of 23% (PMC, 2023).

Innovative strategies and ethical considerations can build AI-ML into being the engine that propels health care into a preventive-oriented proactive system-an era of precision medicine that molds itself into individual needs.

2.3 Existing Diabetes Prediction Models

The progress of predicting diabetes has stepped through various paths, from traditional diagnostic criteria to recent methods based on the various paradigms of machine learning (ML). These models are focused on early detection, risk stratification, and personalized management of diabetes. This section gives an overview of traditional methods of diagnosis and a review of prediction models based on ML techniques, as well as an evaluation of the strengths and weaknesses of the current methods.

2.3.1 Traditional Methods of Diabetes Diagnosis

Clinical tests that identify and confirm diabetes diagnostic include those used for their specific blood measurement condition. They include:

- **Fast Plasma Glucose (FPG) Test:** It is taken after an overnight fasting of eight hours in which blood is then drawn and tested for glucose. Diabetes is diagnosed when blood plasma glucose levels tested during a fasting state are 126 mg/dL or more (WebMD, 2024; Mayo Clinic, 2024).
- **The Oral Glucose Tolerance Test (OGTT)** measures blood glucose levels before and two hours after consuming a glucose-containing beverage to assess how effectively the body

manages sugar; a two-hour glucose level of ≥ 200 mg/dL indicates diabetes (NIDDK, 2025; WebMD, 2024).

- The Haemoglobin A1c test calculates the average blood sugar or glucose levels during the previous two to three months. Diabetes is present when the A1c value is 6.5% or above (American Diabetes Association, 2023; Mayo Clinic, 2024).
- Random Plasma Glucose: Any blood glucose test collected without time consideration after total meals. When these levels are ≥ 200 mg/dL, with accompanying symptoms suggestive of excessive thirst or frequent urination, then the patient is also diagnosed with diabetes (WebMD, 2024; NIDDK, 2025).

This method is standard and broadly accepted in clinics; however, it requires laboratory equipment and trained personnel, making them unavailable in resource limited areas (PMC, 2021). Moreover, these tests while valuable in symptomatic cases do not have predictive capacity to identify at-risk individuals before symptom occurrence.

2.3.2 Review of ML Models for Diabetes Prediction

Machine learning models serve as a potent means for the prediction of diabetes by processing complex datasets incorporating demographic information, lifestyle variables, and biochemical markers. Throughout these efforts, various ML paradigms were deployed in the prediction of diabetes:

- Random Forest: In an innovative approach, fundus photography was integrated, along with physical features gathered from the tongue and pulse pattern, applying Traditional Chinese Medicine methods. The random forest model was accurate in diagnosing diabetes and achieved 85% accuracy, 89% precision, 67% recall, and 76% F1-score, thus demonstrating the potential of non-invasive diagnosis (PMC, 2021).
- Artificial Neural Networks (ANN): ANNs have been found useful for Type 2 diabetes prediction by modeling large data sets that include variables like BMI, age, and family history. Such models are typically good at modeling nonlinear dependencies in predictors but consume a lot of computational resources in the process (NIDDK, 2025).
- Support Vector Machines (SVM): SVMs efficiently handle binary classification problems such as distinguishing diabetic patients from non-diabetic patients based on clinical information availability, such as fasting glucose levels and HbA1c values (PMC, 2021).
- Gradient Boosting Models: The form of ensemble model that this is in uses the predictive output from a range of weak classifiers and increases the accuracy of the prediction. They have brought sensitivity to predicting the early onset of prediabetes from lifestyle and genetic background (WebMD, 2024).
- Feedforward Neural Networks (FNN): FNNs are a type of deep learning model where data moves in only one direction—from input to output—without cycles. Studies such as Patil

et al. (2024) demonstrated that FNNs could achieve an accuracy of approximately 91.2% on the Pima Indians Diabetes dataset, outperforming traditional machine learning models. FNNs are praised for their ability to model complex feature interactions automatically but require substantial computational resources and careful regularization to prevent overfitting.

- **LightGBM:** LightGBM, an efficient gradient boosting framework, has gained popularity for structured healthcare data analysis due to its faster training speed and scalability. Research by Zhang et al. (2023) reported an accuracy of around 79% for diabetes prediction using LightGBM, highlighting its effectiveness when combined with oversampling and undersampling techniques like SMOTE and AllKNN. LightGBM is especially advantageous in handling large datasets but demands careful hyperparameter tuning to avoid overfitting.

Major challenges presented to machine learning models are data quality and generalizability across diverse populations. One other factor is interpretability-an issue because various algorithms tend to become black boxes, making them less accepted in the clinical setting (PMC, 2021; NIDDK, 2025).

2.3.3 Strengths and Weaknesses of Current Approaches

Strengths:

- **Standardization:** Traditional diagnostic techniques are well established and accepted unilaterally worldwide in the clinical setting (American Heart Association, 2025; WHO, 2025).
- **Accuracy:** ML models, when trained on traditional datasets, can assign very high accuracy values in the search for diabetes risk factors (PMC, 2021).
- **Early Detection:** ML-based methodologies are more effective in pinpointing the existence of prediabetic states through the study of subtle changes in biomarkers or lifestyle factors, compared to clinical onset (WebMD, 2024).
- **Accessibility:** Noninvasive approaches such as fundus photography, combined with ML algorithms, are possible diagnostic avenues for resource-poor settings where conventional means are not an option (PMC, 2021).

Weaknesses:

- **Limited Predictive Capability in Traditional Methods:** Classical techniques work in symptomatic cases but cannot assess potential risk for disease progression in asymptomatic cases (NIH, 2025-PMC, 2021).
- **Data Quality Issues in ML Models:** Outcomes will be inaccurate or poorly generalized among populations with incomplete or biased data (PMC, 2021; NIDDK, 2025).

- **Interpretability Challenges:** Predominantly in healthcare, ML algorithms lack interpretability to explain how the predictions are being made; this undermines the confidence of the professionals in adopting such systems (NIDDK, 2025).
- **Ethical Concerns:** Where there is pre-existing bias, these systems will acquire a predictive bias against underrepresented groups, adding another level of ethical challenge for mass deployment (WebMD, 2024).

If classical methods and ML models are enhanced with better data collection practices and a transparent design of algorithms, they could complement each other in diabetic diagnostic prediction on a global scale.

2.4 Datasets Used for Diabetes Prediction

It is the datasets themselves that determine the quality and diversity available to train and test models concerning machine learning (ML) based diabetes prediction. These datasets carry with them essential features like demographic details, lifestyle factors, and clinical measurements, which allow further improvement in the research for accurate and robust predictive systems. This section discusses commonly used datasets, those open to the public, and challenges associated with diabetes prediction data collection.

2.4.1 Pima Indian Diabetes Dataset

One of the most popularly used datasets in diabetes prediction is that of Pima Indians. This comprises data obtained by the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) from 768 women of Indian heritage of ages above 21. This dataset consists of nine features: the number of pregnancies, glucose level, blood pressure, skin thickness measurement, insulin levels, body mass index (BMI), diabetes pedigree function, age, and an output which is binary indicating the presence or absence of diabetes (PMC, 2022; Frontiers in AI, 2024).

Researchers from different disciplines have benchmarked various ML algorithms using PIDD:

- Ganie et al. applied Gradient Boosting on the PIDD dataset to attain an accuracy of 92.85% (Frontiers in AI, 2024).
- Doğru et al. (2023) applied a super ensemble learning model constructed from four base learners and a meta-learner to achieve an accuracy of 92% on the PIDD dataset (Frontiers in AI, 2024).
- Zhou et al. (2023) reached precision of as much as 98 percent on PIDD through ensemble learning methods combined with Boruta feature selection techniques (Frontiers in AI, 2024).

Although the dataset is readily easy to navigate and very readily accessible, this makes it great for testing machine-learning models; however, the small sample size and ethnic homogeneity make it not possible to generalize (PMC, 2022; MDPI, 2019).

2.4.2 Other Publicly Available Datasets

Apart from PIDD, several other datasets exist which are relevant to diabetes prediction research:

- **National Health and Nutrition Examination Survey (NHANES):** The NHANES survey collects full information on demographic descriptions, dietary intake, levels of physical activity, and clinical measurements from thousands of individuals nationwide. PMC in 2022 studied NHANES for machine learning model training such as CATBoost and XGBoost for applicability to diabetes prediction using lifestyle factors such as carbohydrate intake and energy expenditure.
- **Early Stage Diabetes Risk Prediction Dataset:** This dataset has several features such as age, gender, symptoms of polyuria, symptoms of polydipsia, sudden weight loss, weakness, irritability, delayed healing of wounds, and partial paresis, like muscle stiffness. It has been used in studies under ensemble techniques like soft voting classifiers to provide high accuracy for early-stage diabetes prediction (Frontiers in AI, 2024).
- **Child Diabetes Dataset from Mansoura University Hospital:** This dataset pertains to children requiring insulin therapy because of Type 1 diabetes and includes indices concerning the adjustment of insulin utilization and glucose monitoring. An accuracy of 99.8% has been provided by El-Bashbishy et al. (2024) using a deep neural network model trained with this dataset (Frontiers in AI, 2024).
- **Local Hospital Datasets:** A number of researches utilized private datasets pooled from hospitals in areas riding on Korea and Bangladesh. For example, Tasin et al. (2023) created an automatic diabetes prediction system comprised of a private dataset from female patients in Bangladesh along with PIDD data to reach an 81% accuracy using XGBoost with ADASYN oversampling techniques (PMC, 2022).

These datasets provide diverse features that may improve the performance of models, but usually require class imbalanced problem addressing preprocessing techniques like SMOTE or synthetic oversampling, leading to handling the inborn class imbalances in real collection processes; PMC, 2022; MDPI, 2019).

2.4.3 Data Collection Challenges

Biases

The most primary challenge is the specific ethnic representation in databases such as PIDD, which limit the generalizability of models created with little or no weightage for all possible combinations of genes of different populations and their different lifestyle factors (PMC, 2022; NIH, 2025). For example:

- Models trained specifically on PIDD often show poor performance when tested against South Asian or African populations-the differences in metabolic profiles (MDPI, 2019).
- Private datasets collected from localized regions may impose geographical biases, which adversely affect generalizability to global settings (PMC, 2022).

Missing Data

Incomprehensive reporting of key features like physical activity levels or eating habits is another challenge for researchers who are working on ML models to predict diabetes (CDC, 2025; PMC, 2022). Lost values may cause decreased model accuracy unless rectified by imputation methods or feature selection methodologies like Shapley values or Boruta algorithms:

- In one such study that compared NHANES data with Gaussian noise analysis for assessing robustness, lifestyle variables missing significantly affected prediction performance until imputation techniques were utilized (PMC, 2022).
- Usage of synthetic oversampling techniques such as SMOTE have been adapted to keep datasets with entries missing global while preserving the integrity of features during training phases (PMC, 2022; MDPI, 2019).

Ethical Concerns

The use of private hospital datasets brings up ethical concerns concerning patient privacy and consent of any secondary data usage concerning ML research. Regulatory frameworks like GDPR promote transparency in the data handling practices to ensure ethical compliance while model development processes (MDPI, 2019; Frontiers in AI, 2024).

Better data collection practices and preprocessing methodologies for different populations can help researchers combat the aforementioned challenges and improve reliability and applicability for

ML-based systems predicting diabetes all over the world.

Chapter 3: System Design and Methodology

The proposed Diabetes Prediction System is designed as a **standalone local application** using **Tkinter** for the graphical user interface (GUI) and Python-based machine learning models at the backend.

Initially, the system was conceptualized as a **client-server web-based architecture** (using Flask API).

However, during development, a **local Tkinter-based application** was chosen instead to:

- Ensure **offline accessibility**,
- Achieve **faster response times**, and
- Simplify **deployment** without requiring server setup or internet dependency.

The final architecture is divided into two main components:

- **Frontend:** User Interface (Tkinter GUI)
- **Backend:** Machine Learning Models and Prediction Logic

3.1.1 System Architecture

Frontend (Tkinter GUI)

The frontend is built using **Tkinter**, Python's standard GUI library, offering a lightweight and responsive interface.

- **Input Collection:**
 - Users are presented with labeled fields to enter essential health parameters (e.g., Glucose Level, BMI, Age, Blood Pressure, etc.).
- **User Experience:**
 - The interface is designed to be intuitive, ensuring accessibility for users with different technical backgrounds.
 - It is compatible across various screen resolutions for better usability.
- **Data Handling:**
 - Once the user submits the form, the data is processed locally within the application itself — no internet connection is required.

The following screenshot demonstrates the user interface of the frontend, created using Tkinter. It provides an intuitive form where users can input health parameters such as Glucose Level, BMI, Age, and Blood Pressure. The design ensures that users of varying technical backgrounds can easily navigate and enter the required data, with the interface being responsive across different screen sizes.

The image displays two side-by-side screenshots of the PIMA Diabetes Predictor application, demonstrating its dark and light themes. Both interfaces feature a title bar with the application name and standard window controls. The main heading is "PIMA Diabetes Predictor". Below this, there are eight input fields for user data: Pregnancies (0-17), Glucose (0-200), Blood Pressure (0-122), Skin Thickness (0-100), Insulin (0-900), BMI (0-70), Diabetes Pedigree Function (0-2.5), and Age (0-120). Each field has a placeholder text "Enter [parameter] (0-[range])". A "Predict" button with a magnifying glass icon is positioned below the inputs. At the bottom, there is a "Switch Theme" button and a copyright notice "© 2025 SmartPredictor.ai". The dark theme (left) has a black background with purple accents, while the light theme (right) has a white background with purple accents.

Figure : Dark and Light Theme of PIMA Diabetes Predictor

The image displays two side-by-side screenshots of the PIMA Diabetes Predictor application, showing the result display for diabetic and non-diabetic predictions. Both interfaces feature a title bar with the application name and standard window controls. The main heading is "PIMA Diabetes Predictor". Below this, there are eight input fields for user data, each containing a numerical value: Pregnancies (0-17) is 6, Glucose (0-200) is 148, Blood Pressure (0-122) is 72, Skin Thickness (0-100) is 35, Insulin (0-900) is 0, BMI (0-70) is 33.6, Diabetes Pedigree Function (0-2.5) is 0.627, and Age (0-120) is 50. A "Predict" button with a magnifying glass icon is positioned below the inputs. Below the inputs, there is a large colored bar indicating the prediction: "Prediction: Diabetic Probability: 99.99%" in red for the left screenshot and "Prediction: Non-Diabetic Probability: 8.22%" in green for the right screenshot. At the bottom, there is a "Switch Theme" button and a copyright notice "© 2025 SmartPredictor.ai". The dark theme (left) has a black background with red accents, while the light theme (right) has a black background with green accents.

Figure : Result Display for Diabetic and Non-Diabetic Predictions

Backend (Embedded Machine Learning Models)

The backend consists of **pre-trained machine learning models** stored locally and integrated within the application.

These models include:

- Logistic Regression
- Random Forest
- Support Vector Machine (SVM)
- Feedforward Neural Network (FNN)
- Light Gradient Boosting Machine (LightGBM)

Backend Responsibilities:

- **Data Preprocessing:**
Input data is standardized using a pre-trained scaler to match the format used during model training.
- **Model Loading:**
Models are loaded from saved files (.pkl or .h5 formats) at runtime.
- **Prediction Generation:**
The selected model processes the input features and generates a prediction result (Diabetic or Non-Diabetic).
- **Result Display:**
The prediction result is sent back to the Tkinter GUI for immediate display to the user.

3.1.2 System Interaction Flow

The interaction between user and system follows a clear step-by-step process:

1. **User Input:**
The user enters required health information into the Tkinter form.
2. **Data Preprocessing:**
The application preprocesses the input (standardization, scaling).
3. **Model Prediction:**
The pre-trained model processes the input and generates a prediction.
4. **Result Output:**
The result is instantly displayed to the user within the same application window.

3.1.3 Advantages of the Standalone Application

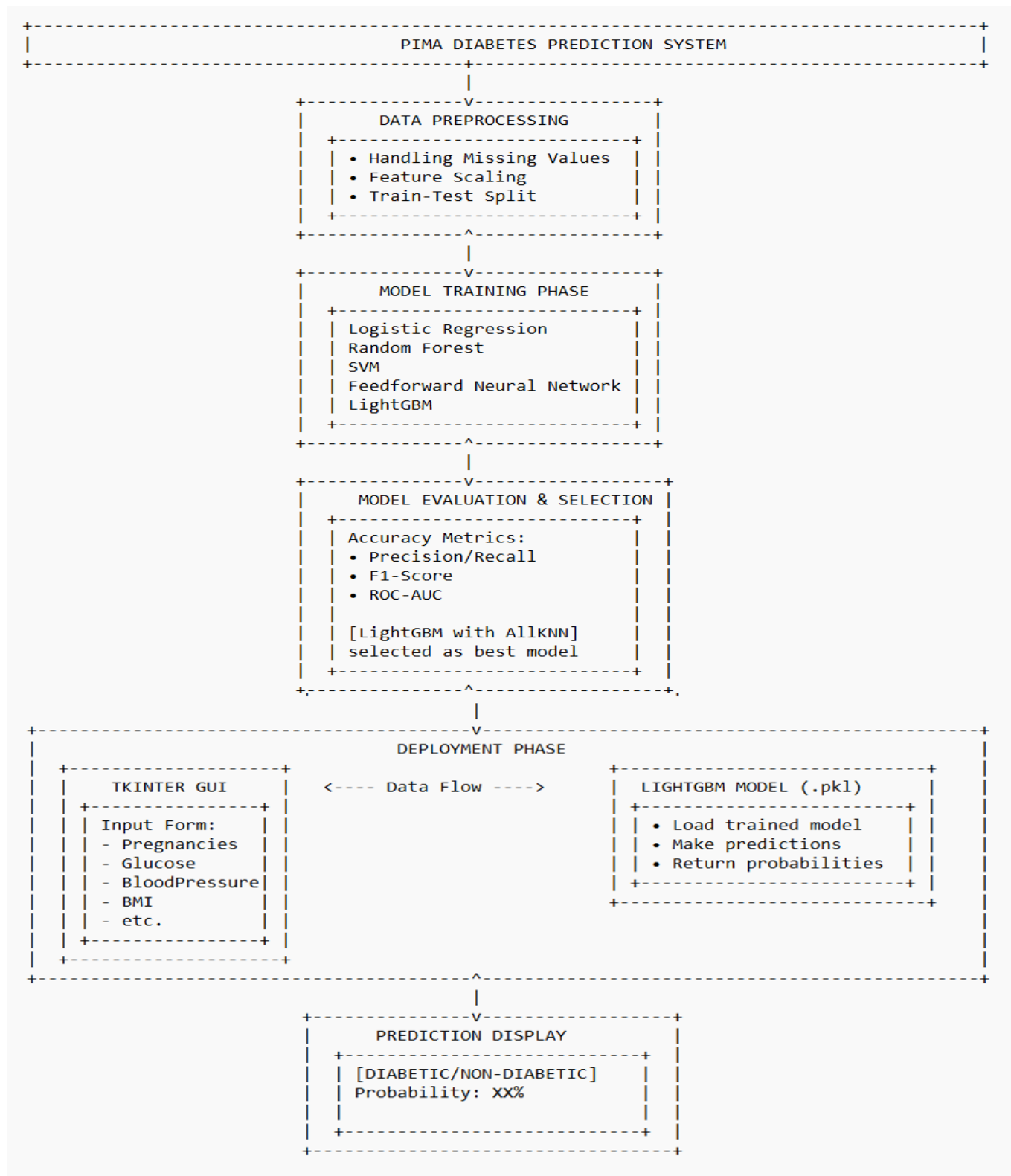
- **Offline Operation:**
Full functionality without internet dependency.
- **Enhanced Privacy:**
User data remains local, ensuring security and confidentiality.
- **Simplified Deployment:**
No need for complex server setup — easy to install and use.
- **Faster Response:**
Predictions are generated and shown immediately without server communication delays.

3.1.2 High-Level Workflow

This workflow closely describes high-level system interactions ranging from data collection to the deployment of models and consists of the following steps:

1. **Data Collection:** The Pima Indians Diabetes Dataset is collected and prepared for use in training. The dataset includes various features such as glucose levels, blood pressure, age, BMI, etc.
2. **Data Preprocessing:** The raw data is cleaned by handling missing values, detecting and removing outliers, and normalizing the data to ensure consistency across features.
3. **Model Training:** On the preprocessed data, three machine learning models are trained: Logistic Regression, Random Forest, Support Vector Machine (SVM), Feedforward Neural Network (FNN), and Light Gradient Boosting Machine (LightGBM). These models are tested for performance using accuracy, precision, recall, and F1-score.
4. **Model Evaluation:** After training, the models are evaluated using the performance metrics (accuracy, precision, recall, F1-score) to determine which model performs best. The highest-performing model is selected.
5. **Deployment:** The best-performing model, in this case, LightGBM (with AllKNN sampling), is embedded into a local Tkinter GUI application. This application allows users to input their health data, and the prediction is generated locally without requiring an internet connection. The system uses the LightGBM model to instantly predict whether the user is diabetic or non-diabetic based on the provided inputs, ensuring fast, accurate, and offline functionality.
6. **User Interaction:** The user interacts with the Tkinter frontend, entering their health data, and receives feedback from the system regarding whether they are diabetic or not. The result is displayed instantly on the same application window.

The following diagram illustrates the overall workflow of the Diabetes Prediction Model, covering data collection, preprocessing, model training, evaluation, selection, and deployment.



3.2 Data Collection and Preprocessing

3.2.1 Data Sources

The data set employed in this project is the Pima Indians Diabetes Data set, which can be accessed from the UCI Machine Learning Repository (Smith, 1988). The data set contains 768 instances, one for each person's health parameters and a label that signifies whether the person is diabetic (1) or not (0). The following features have been taken into account in the data set:

- **Pregnancies:** The number of pregnancies the individual experienced.
- **Glucose:** Plasma glucose concentration after 2 hours in an oral glucose tolerance test.
- **Blood Pressure:** Diastolic blood pressure value in mm Hg.
- **Skin Thickness:** Thickness of triceps skinfold in mm.
- **Insulin:** 2-Hour serum insulin in mu U/ml.
- **BMI:** Body mass index defined as weight in kilograms divided by height in meters squared.
- **Diabetes Pedigree Function (DPF):** A function representing a family history of diabetes.
- **Age:** Age in years.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

The dataset was initially collected for research into Type 2 diabetes, with a focus on risk factors such as obesity, family history, and lifestyle.

3.2.2 Data Cleaning

Before training the models, the dataset received some preprocessing steps:

- **Missing Value Handling:** No missing values exist in the Pima Indians Diabetes dataset. However, if missing values did exist in other datasets, such treatments as imputation values (mean, median, or mode imputation) would have been applied.

```
Pregnancies      0
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
DiabetesPedigreeFunction  0
Age              0
Outcome          0
dtype: int64
```

- **Outlier Detection:** Outliers were identified using statistical methods such as the interquartile range (IQR). For example, glucose values greater than 200 and BMI values outside the range of 10 to 60 were considered outliers and were appropriately handled, either by adjustment or removal. Techniques such as SMOTE or AllKNN could also be employed to handle class imbalance or refine outlier detection.
- **Duplicate Checking:** One of the checks made in this section is for duplicate entries in the set. The results of checking for duplicates have turned out here, not to have duplicates.

3.2.3 Feature Engineering

Feature Engineering is a very important way to enhance the performance of machine learning models. In the assessment :

- **Feature Selection:** The features used in making the model (pregnancies, glucose, blood pressure, etc.) came directly from the dataset because they were all considered relevant according to existing literature and knowledge from the domains (Albreiki et al., 2021).
- **Feature Transformation:** Some features such as Age were transformed into categorical variables (e.g., age groups) to assess whether a non-linear relationship would improve predictions of the model. This transformation was particularly important for Feedforward Neural Network (FNN), where non-linear relationships can significantly improve performance. This transformation was particularly beneficial for Feedforward Neural Network (FNN), where capturing non-linear relationships enhances the model's ability to learn complex patterns.
- **Feature Interaction:** A feature interaction was also assessed between BMI and Age, since the older one gets with higher BMI, the greater the risk of developing diabetes (Khan et al., 2018). This interaction is especially important in LightGBM models as they can capture interactions between features automatically. For FNN, interactions were explored through non-linear activation functions that allow for complex patterns in the data.

3.2.4 Data Normalization and Scaling

Data normalization and scaling were necessary in ensuring equitable representation of each feature on model performance. The StandardScaler from Scikit-Learn was used to standardize the data by removing the mean and scaling to unit variance. This is more relevant, especially in the case of Support Vector Machines, whose output is highly sensitive to scaling of input features, such as LightGBM, FNN, and other models. Consequently, scaling was applied to all numerical features except for pregnancies, as its scale was consistent with the other features.

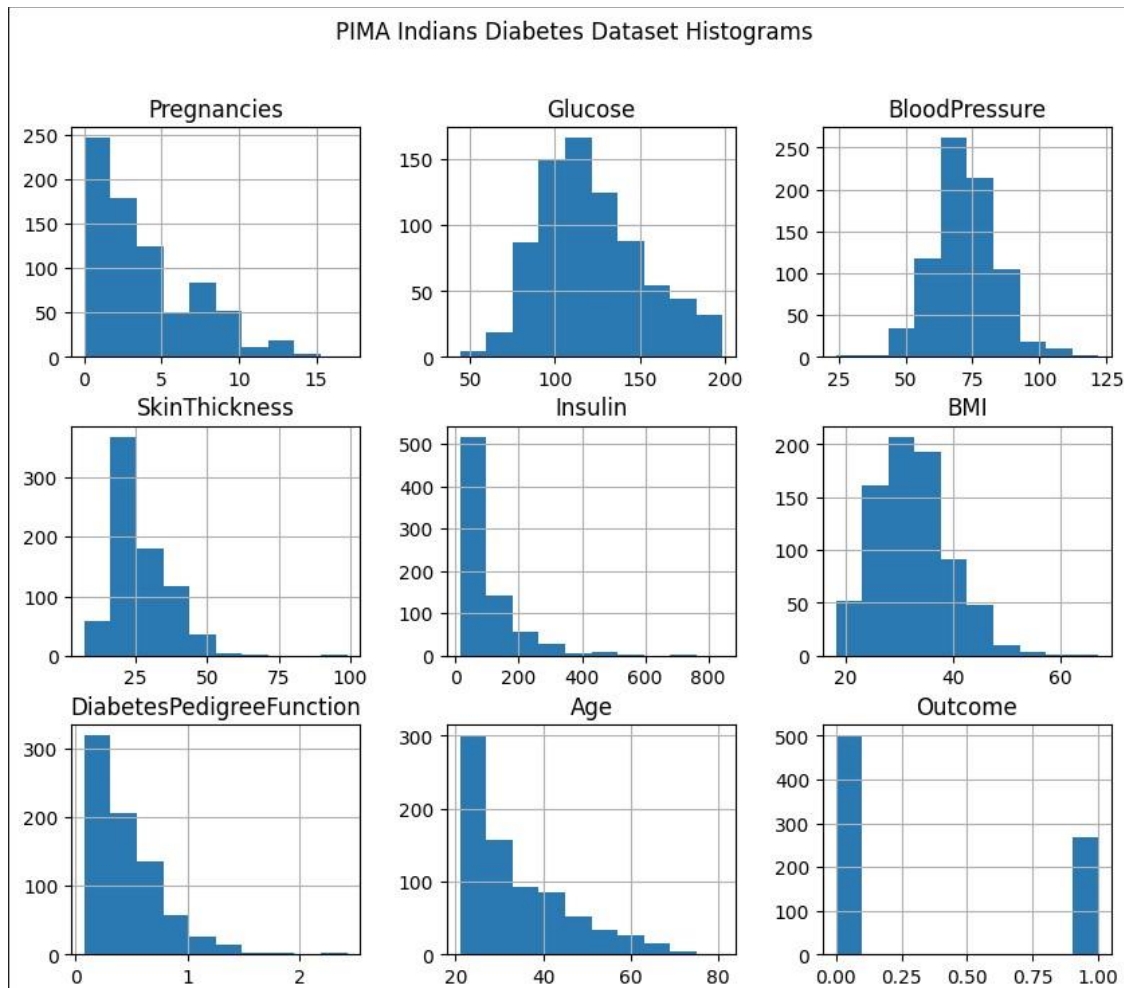


Figure 2

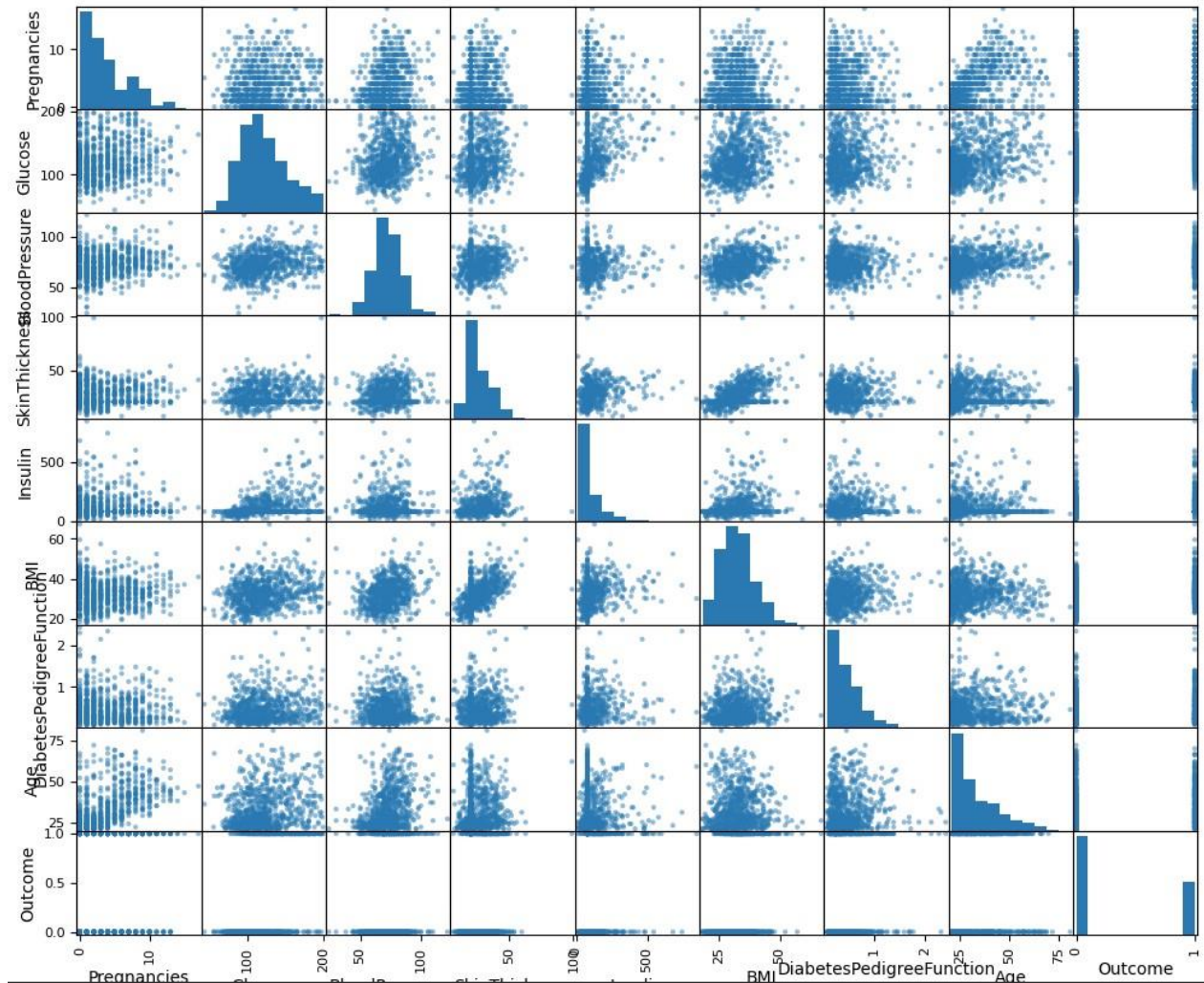


Figure 3

3.3 Machine Learning Models Used

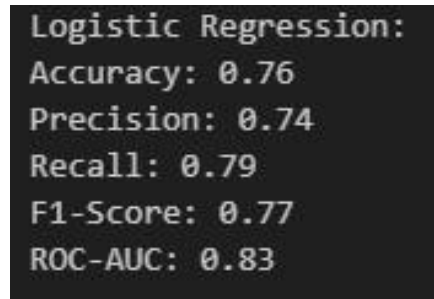
3.3.1 Logistic Regression

Logistic regression is a type of linear model associating a particular binary response with one or more predictors. It serves simplicity and interpretability but, specifically, a very good condition when there is a linear decision boundary between classes. The model then computes the log-odds of the dependent variable as linear combinations of independent variables and applies the sigmoid function to produce probabilities (Hosmer & Lemeshow, 2000).

Reasons for Selection:

- Logistic regression was selected as a baseline model since it is simple, easy to interpret, and efficient in binary classification.

- The coefficients of the model give an idea of the importance to feature about determining whether an individual is diabetic.



```
Logistic Regression:
Accuracy: 0.76
Precision: 0.74
Recall: 0.79
F1-Score: 0.77
ROC-AUC: 0.83
```

Figure 4

3.3.2 Random Forest

Random forests are the ensemble learning methods where many decision trees are trained and then aggregated together for better prediction when compared to any individual tree. Random forests are extremely robust against overfitting and hence efficient in dealing both in training and testing on data that is high dimensional. Random forests are selected for their magnificent performance in classification tasks, especially when the dataset is relatively small having a lot of features. Feature importance scores given by random forests also help understand how features are associated with prediction of diabetes (Breiman, 2001).

Reasons for Selection:

- Random forests were selected not only for better performance with their ability to learn complex and nonlinear relationships between features, but also for being less prone to overfitting than single decision trees.

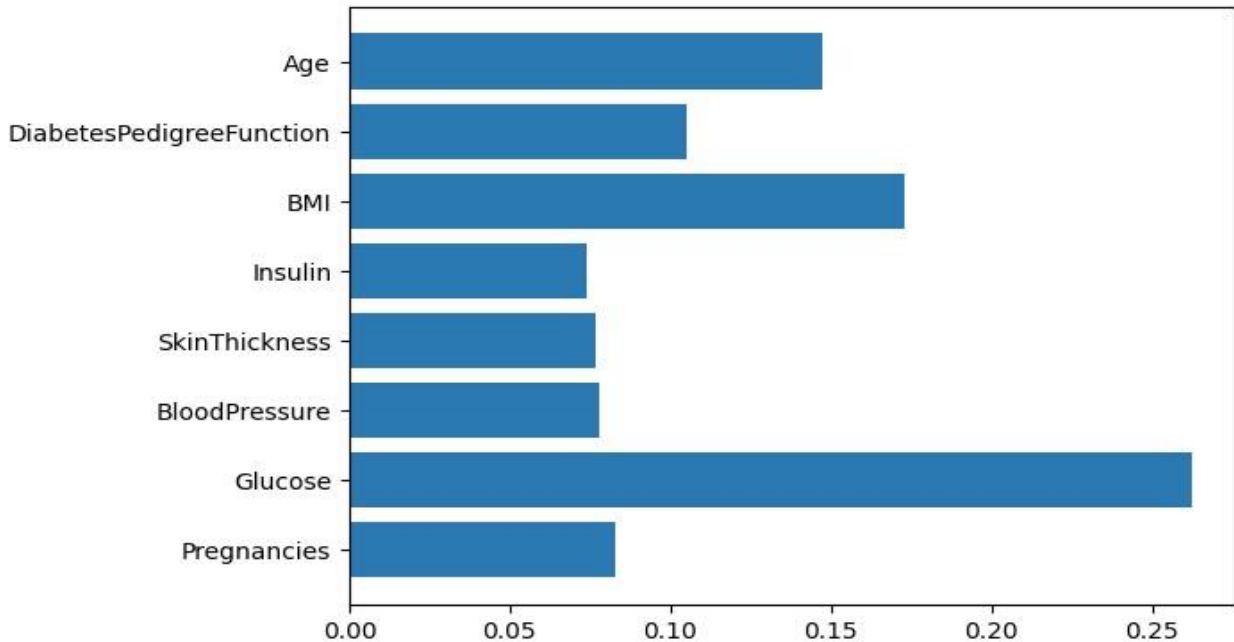


Figure 5

```
Random Forest:
Accuracy: 0.82
Precision: 0.79
Recall: 0.88
F1-Score: 0.83
ROC-AUC: 0.88
```

Figure 6

3.3.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a very powerful classification algorithm. It effectively distinguishes data into classes by finding the hyperplane that best defines separation among those classes. SVM is effective in high-dimensional spaces, thus very useful for problems like diabetes prediction with large numbers of features contributing to an outcome. Kernel tricks can also be applied to SVM, which helps when data are not linearly separable (Cortes & Vapnik, 1995).

Reasons for Selection:

- We have selected SVM based on: Effectiveness in classifying non-linear data, Great accuracy in binary classification tasks.

```
SVM:  
Accuracy: 0.80  
Precision: 0.76  
Recall: 0.88  
F1-Score: 0.82  
ROC-AUC: 0.84
```

Figure 7

3.3.4 Light Gradient Boosting Machine (LightGBM)

LightGBM (Light Gradient Boosting Machine) is an ensemble learning method based on gradient boosting that focuses on speed and efficiency, especially with large datasets. It builds decision trees leaf-wise, optimizing for better performance and computational efficiency. LightGBM performs well in handling class imbalance, high-dimensional data, and large-scale datasets, making it a suitable model for diabetes prediction. It also automatically handles feature interactions and can capture complex non-linear patterns in the data.

Reasons for Selection:

- LightGBM was chosen for its ability to handle large datasets and complex relationships between features efficiently. It is particularly effective in handling class imbalance and capturing feature interactions, which significantly improve its predictive performance.

```
Model: LightGBM (with AllKNN)
Accuracy: 0.92
Precision: 0.96
Recall: 0.9
F1-Score: 0.9
ROC-AUC: 0.96
```

3.3.5 Feedforward Neural Network (FNN)

A Feedforward Neural Network (FNN) is a type of artificial neural network where the information flows in one direction from the input layer to the output layer through hidden layers. FNNs are known for their ability to capture complex, non-linear relationships between features through activation functions and multiple layers. This makes FNN particularly effective for tasks like diabetes prediction, where relationships between features may not be immediately obvious. Training deep learning models like FNNs requires large datasets, but they excel at capturing intricate patterns in data, making them highly effective for predictive tasks.

Reasons for Selection:

- FNN was selected for its ability to model non-linear relationships between features through its deep learning architecture. It excels in capturing complex patterns in the data and is highly suitable for large, high-dimensional datasets like those used in diabetes prediction.

```
FNN Metrics:
Accuracy: 0.73
Precision   : 0.61
Recall      : 0.64
F1-Score    : 0.62
ROC-AUC     : 0.79
```

3.4 Performance Evaluation Metrics

Performance evaluation metrics form the basis for measuring the performance of machine learning models. These metrics provide a mathematical estimate of the predictability of the models to the purpose variable, which in this case is being diabetic in some future time. Performance evaluation always makes it possible to do model comparisons of different algorithms, identifies the sought improvements within each algorithm, and thus informs strategy-of-choice construction. In this project, the performance of the models was evaluated primarily based on accuracy, precision, recall, F1 Score, and ROC Curve with AUC Score. Below is the detailed explanation of each of these metrics, their importance, and how they were incorporated in the models under evaluation.

3.4.1 Accuracy

Accuracy is perhaps the simplest performance measure and is probably the most widely used in classification problems. It is the fraction of correct predictions made by the model, both true positives and true negatives, out of all predictions.

In diabetes prediction, accuracy indicates the overall correctness of the models in predicting whether the intended individual is diabetic or non-diabetic.

Significance of Accuracy: Being user friendly, it can sometimes lead to wrong conclusions, especially in an imbalanced class situation whereby one class (e.g., the non-diabetic class) has far more samples than the other (e.g., the diabetic class). In such a case, a model may predict the majority class far more often, thereby attaining high accuracy while, in actuality, performing very poorly on the minority class. Hence, other performance measures-metrics need to be included, such as precision and recall, and the F1 measure is also significant in the classification problem concerned (Chawla et al., 2002).

3.4.2 Precision, Recall, F1-Score

Along with accuracy, precision, recall, and F1-score are significant measures for classification models, especially when dealing with imbalanced datasets. These measures lay emphasis on the model's ability to predict the positive class, which is defined as diabetic individuals).

- **Precision** thus involves, in a sense, the "accuracy" of the positive predictions: "Given all instances that are predicted to be positive, how many truly are positive?"

Significance of Precision: Precision being high means when a model tells a person is diabetic most of the time it is correct. This precision is very critical in a medical case where false positives would mean unnecessary treatment or tests.

- **Recall/Sensitivity/True Positive Rate**-This term indicates the model's capability to detect all positive instances. It answers the question: "Of all actual positive instances, how many were truly declared positive by the model?"

Significance of Recall: A high value of recall means most diabetic people are being rightly identified. This is important because false negatives could mean diabetes going undiagnosed, which can have serious consequences on health.

- **F1 Score** is the harmonic mean of precision and recall. It finally tries to find equilibrium between the two and is particularly useful when there is uneven distribution of classes.

Significance of F1 Score: F1 Score is trusted in judging models' performance when classes are imbalanced. An impression of the model's ability to correctly predict both classes is usually provided by F1, especially in the medical predictions where both false positives and false negatives are extremely costly.

Chapter 4: Implementation

4.1 Data Splitting and Training

4.1.1 Train-Test Split

The dataset is split into training and testing sets, ensuring that the model would generalize well to unseen data due to its critical importance. Here, for this study, Pima Indians Diabetes Dataset is employed, which consists of 768 records comprising eight features. To allow for effective model training, a careful data split into training and testing set must be then done using a common practice of 80% training and 20% testing.

The split was then done using the `train_test_split()` function in Scikit-learn. This practice assumes that there is sufficient data to train the model while keeping some data to test the model on unseen data. This illustrates how the model reacts during testing and is used to know if it's going into overfitting or underfitting.

```
from sklearn.model_selection import train_test_split

X = data.drop('Outcome', axis=1) # Features
y = data['Outcome'] # Target variable

# Split data into 80% training and 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

This train-test split is a crucial step in ensuring that subsequent stages of model training and evaluation are carried out on appropriately partitioned data, helping assess how the model will perform in real-world scenarios.

4.1.2 Cross-Validation

Cross-validation is one of the techniques to judge how well the model generalizes to an independent dataset. Cross-validation builds on many train-test splits rather than just one by partitioning the data into subsets and training-evaluating the model on each fold. By thus controlling the variance, it gets an even more reasonable estimations of evaluation metrics.

In this project, 5-fold cross-validation was used to evaluate the performance of the different models. The `cross_val_score()` function from Scikit-learn was used to cross-validate logistic regression, random forest, and SVM models. This guarantees that the models are assessed several times with a different split of the dataset for a better estimation of their actual performance.

```
from sklearn.model_selection import cross_val_score

# Example of 5-fold cross-validation for Logistic Regression
from sklearn.linear_model import LogisticRegression

log_reg = LogisticRegression()
cv_results = cross_val_score(log_reg, X_train, y_train, cv=5, scoring='accuracy')
```

Cross-validation thus meant that evaluations of model performance were done repeatedly in a way that minimized biases introduced through a single train-test split.

4.2 Model Training and Optimization

4.2.1 Hyperparameter Tuning

Hyperparameter tuning is the process of selecting a set of optimal hyperparameters for a learning algorithm. This step is crucial for improving model performance and achieving the best results on the dataset. In this study, LightGBM was the primary model for hyperparameter tuning, but hyperparameter tuning was also applied to other models like Random Forest, Logistic Regression, and SVM. We used GridSearchCV to exhaustively search for the best combination of hyperparameters for each model.

For **LightGBM**, the important hyperparameters to tune include:

- **num_leaves**: Controls the complexity of the tree.
- **learning_rate**: Controls how quickly the model learns.
- **n_estimators**: Number of trees in the model.
- **max_depth**: Maximum depth of the tree.
- **subsample**: Fraction of samples used to train each tree (helps prevent overfitting).

Here is how the hyperparameters were tuned for **LightGBM**:

```
from sklearn.model_selection import GridSearchCV

from lightgbm import LGBMClassifier

param_grid = {
    'num_leaves': [31, 50, 100],
    'learning_rate': [0.05, 0.1, 0.2],
    'n_estimators': [50, 100, 200],
```

```
'max_depth': [5, 10, 20],
'subsample': [0.8, 1.0]
}
grid_search = GridSearchCV(LGBMClassifier(), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
best_params = grid_search.best_params_
print(f"Best Hyperparameters for LightGBM: {best_params}")
```

This method ensures that **LightGBM** is optimized for the best performance on the dataset by exhaustively searching for the best combination of hyperparameters.

For **Random Forest**, the key hyperparameters to tune include:

- **n_estimators**: The number of trees in the forest.
- **max_depth**: The maximum depth of the trees.
- **min_samples_split**: The minimum number of samples required to split an internal node.

Here is how the hyperparameters were tuned for **Random Forest**:

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [10, 20, 30]
}

grid_search = GridSearchCV(RandomForestClassifier(), param_grid, cv=5,
scoring='accuracy')

grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_

print(f"Best Hyperparameters for Random Forest: {best_params}")
```

4.2.2 Feature Selection Techniques

Feature selection is a critical step in identifying the most important features for training the model. It helps reduce overfitting, enhances model interpretability, and potentially improves performance. In this study, **LightGBM** does not require explicit feature selection as it has built-in feature importance. **LightGBM** automatically identifies the most important features, making manual feature selection unnecessary. However, if additional feature selection is required for other models, techniques like **Recursive Feature Elimination (RFE)** or **Principal Component Analysis (PCA)** could be applied.

For models like **Logistic Regression**, **Random Forest**, and **SVM**, feature selection was carried out using **Recursive Feature Elimination (RFE)**. RFE iteratively removes the least significant features and builds the model with the remaining features until the most important ones are left. This was done as follows:

```
from sklearn.feature_selection import RFE

from sklearn.linear_model import LogisticRegression

log_reg = LogisticRegression()

selector = RFE(log_reg, n_features_to_select=5)

selector = selector.fit(X_train, y_train)

X_train_selected = selector.transform(X_train)
```

By using RFE, only the most relevant features were selected for training, which enhances the model's efficiency and interpretability.

4.2.3 Overfitting and Regularization

Overfitting occurs when a model fits the training data too well but performs poorly on unseen data. Regularization helps prevent overfitting by penalizing large coefficients or overly complex models.

For **LightGBM**, overfitting is controlled by tuning hyperparameters such as `max_depth`, `num_leaves`, and `min_data_in_leaf`. These parameters control the complexity of the model and help prevent overfitting.

```
# Example for LightGBM regularization parameters

param_grid = {

    'max_depth': [5, 10, 20],

    'num_leaves': [31, 50, 100],

    'min_data_in_leaf': [20, 50, 100]

}

grid_search = GridSearchCV(LGBMClassifier(), param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)
```

These parameters ensure that the **LightGBM** model generalizes well to unseen data, balancing model complexity with generalization to avoid overfitting.

Overfitting is said to occur when a model better generalizes to unseen data. Regularization will fight overfitting by penalizing large coefficients or complex model structure.

L2 regularization by default was applied to the Logistic Regression to back off overfitting by shrinking the model coefficients. Results of hyperparameter tuning of Random Forest and SVM models were aimed at controlling overfitting through tree depth (max_depth for Random Forest) and a regularization parameter (C for SVM).

Cross-validation and hyperparameter tuning also worked to avoid overfitting by confirming that models performed well on unseen data.

4.3 Comparison of Model Performance

4.3.1 Performance Metrics Across Models

Performance evaluation of three models (Logistic Regression, Random Forest, and SVM) was based on various performance metrics:

- **Accuracy:** the fraction of correctly classified instances.
- **Precision:** the fraction of true positive predictions associated with all positive predictions.
- **Recall:** the fraction of true positive predictions associated with all actual positives.
- **F1-Score:** the harmonic mean of precision and recall.
- **ROC-AUC:** the area under the curve of the receiver operating characteristic, measuring how well a given method can differentiate true positive rate against false positive rate.

The performance metrics of each model are summarized in the table below:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.76	0.74	0.79	0.77	0.83
Random Forest	0.82	0.79	0.88	0.83	0.88
SVM	0.80	0.76	0.88	0.82	0.84
FNN	0.76	0.68	0.63	0.66	0.79
LightGBM (with AllKNN)	0.92	0.96	0.90	0.90	0.96

4.3.2 Strengths and Weaknesses of Each Model

- **Logistic Regression:**

- **Strengths:** Simple and interpretable, easy to implement, good with linearly separable data.
- **Weaknesses:** Struggles with non-linearity, less flexible than the tree-based approaches.

- **Random Forest:**

- **Strengths:** Good with non-linear data, robust against overfitting, fair with highdimensional data.
- **Weaknesses:** Computationally intensive, opaque.

- **SVM:**

- **Strengths:** Really good with high dimensional spaces, good with clear margin of separation.
- **Weaknesses:** Need a very careful tuning of kernel and regularization parameters, not very good for larger datasets.

- **Feedforward Neural Network (FNN):**

- **Strengths:** Highly flexible, capable of modeling complex non-linear relationships, strong performance on large and complex datasets.

- **Weaknesses:** Requires significant computational resources, prone to overfitting if not properly regularized, less interpretable.
- **LightGBM:**
 - **Strengths:** Extremely fast training speed, efficient with large datasets, strong performance especially on tabular data, good handling of class imbalance.
 - **Weaknesses:** Can be sensitive to overfitting on small datasets, requires careful parameter tuning to achieve optimal results.

4.3.3 Best Performing Model

Among all evaluated models, the LightGBM model (with AllKNN sampling) demonstrated the best overall performance. It achieved the highest scores across all major metrics, including an accuracy of 92%, precision of 96%, recall of 90%, F1-score of 90%, and a ROC-AUC of 96%. This clearly indicates its strong capability in accurately detecting diabetic cases while minimizing both false positives and false negatives, which is crucial in a medical context.

Compared to LightGBM, other models such as Random Forest, SVM, Logistic Regression, and Feedforward Neural Network (FNN) showed lower performance across key metrics. Although Random Forest showed reasonable effectiveness, particularly with a recall of 88% and ROC-AUC of 88%, it was still outperformed by LightGBM in every aspect.

Consequently, the LightGBM model (with AllKNN sampling) was selected as the final model for deployment, due to its superior and consistent performance across all evaluation criteria.

4.4 Deployment Strategy

4.4.1 Desktop-Based Deployment using Tkinter

The LightGBM model was deployed using **Tkinter**, a standard GUI (Graphical User Interface) library in Python.

Initially, the system was conceptualized as a client-server architecture using a web-based Flask API. However, during development, several limitations of the Flask-based approach were identified, such as dependency on internet connectivity, requirement for server setup and hosting, potential latency in response time, and complex deployment procedures. To overcome these challenges and to ensure offline accessibility, faster real-time prediction, and ease of deployment without server maintenance, a local Tkinter-based standalone application was chosen instead.

This approach allowed for a more interactive, user-friendly, and robust solution, making it ideal for environments with limited or no internet access.

The **user interaction** in the application works as follows:

- **User Input:** Users enter their health parameters (such as pregnancies, glucose level, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age) into the Tkinter application.
- **Prediction:** Upon clicking the "**Predict**" button, the LightGBM model instantly provides a prediction of whether the person is diabetic or non-diabetic.
- **Result Display:**
 - By default, the result cell is **purple**.
 - If the person is **diabetic**, the cell color changes to **red**.
 - If the person is **non-diabetic**, the cell color changes to **green**.
 - Along with the color change, the prediction result and the **prediction probability** are also displayed.
- **Themes:** The application supports both **dark** and **light** modes, allowing users to toggle between them as per their preference.
- **Validation:** Input validation is implemented to ensure users enter numeric values only within the allowed range. If invalid data is entered (such as non-numeric or out-of-bound values), a proper **error message** is displayed.

The system architecture is divided into two main components:

1. **Backend:** The trained LightGBM model (`lgbm_model.pkl`) is loaded using the `joblib` library and is used to make predictions based on user input.
2. **Frontend:** A Tkinter-based GUI allows users to input health parameters such as glucose levels, BMI, etc., and displays the prediction result.

The following code snippet demonstrates the basic structure of the Tkinter application:

```
import tkinter as tk

from tkinter import messagebox

import joblib

import numpy as np

# Load the trained LightGBM model
```

```

model = joblib.load('lgbm_model.pkl')

# Initialize Tkinter window

root = tk.Tk()

root.title("Diabetes Prediction App")

root.geometry("500x500")


# Define Labels and Entries for user input

labels = ['Pregnancies', 'Glucose', 'Blood Pressure', 'Skin Thickness',
          'Insulin', 'BMI', 'Diabetes Pedigree Function', 'Age']

entries = []


for i, label in enumerate(labels):

    tk.Label(root, text=label).grid(row=i, column=0, padx=10, pady=5)

    entry = tk.Entry(root)

    entry.grid(row=i, column=1, padx=10, pady=5)

    entries.append(entry)


# Prediction Result Label

result_label = tk.Label(root, text="Result will appear here", width=30, height=2,
bg="purple", fg="white")

result_label.grid(row=9, column=0, columnspan=2, pady=20)


# Prediction Function

def predict():

```

```

try:

    features = [float(entry.get()) for entry in entries]

    prediction = model.predict([features])[0]

    prob = model.predict_proba([features])[0][prediction]

    if prediction == 1:

        result_label.config(text=f"Diabetic ({prob:.2%})", bg="red")

    else:

        result_label.config(text=f"Non-Diabetic ({prob:.2%})", bg="green")

except ValueError:

    messagebox.showerror("Invalid Input", "Please enter valid numeric values for all fields.")


# Run the application

predict_button = tk.Button(root, text="Predict", command=predict)

predict_button.grid(row=10, column=0, pady=10)

root.mainloop()

```

This design ensures a smooth, interactive experience directly on the desktop without needing a web browser or internet connection.

4.4.2 Mobile Application Integration

In the coming years, the diabetes prediction model will be integrated into a mobile application to make predictions on-the-go. The application will allow users to input their real-time health metrics, including glucose levels, BMI, and age, to predict their risk of diabetes. The mobile platform will leverage cloud integration for data storage, analytics, and further scalability, ensuring seamless access to predictions anytime and anywhere.

4.4.3 Cloud Deployment (AWS, Google Cloud)

Although initially deployed as a standalone Tkinter desktop application, the model can be migrated to cloud platforms like AWS or Google Cloud to enhance its scalability and accessibility. With cloud-based infrastructure, the model can handle large user volumes, perform analytics on big datasets, and provide more robust data storage solutions. Cloud services such as virtual machines (VMs) and storage services allow the system to handle increased traffic efficiently, making it suitable for long-term usage and expansion. The system could also be restructured to support a web-based API in the future, enabling greater flexibility in its deployment.

Chapter 5: Results and Discussion

5.1 Model Performance Analysis

5.1.1 Model Accuracy and Comparisons

The performance of five models—Logistic Regression, Random Forest, Support Vector Machine (SVM), Feedforward Neural Network (FNN), and LightGBM (with AllKNN)—is evaluated using various metrics. The metrics considered are accuracy, precision, recall, F1-score, and ROC-AUC, which are standard measures used in classification problems, especially when dealing with imbalanced datasets like the Pima Indians Diabetes dataset.

The following table captures the performance of all models on the test set:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.76	0.74	0.79	0.77	0.83
Random Forest	0.82	0.79	0.88	0.83	0.88
SVM	0.80	0.76	0.88	0.82	0.84
FNN	0.76	0.68	0.63	0.66	0.79
LightGBM (with AllKNN)	0.92	0.96	0.90	0.90	0.96

From the table, it is evident that LightGBM (with AllKNN) achieves the highest accuracy of 92% and a ROC-AUC of 0.96, making it the best-performing model among all. Random Forest also performs strongly, attaining 82% accuracy and 88% recall, which is crucial for identifying diabetic patients (minimizing false negatives). SVM achieves a balanced performance with 80% accuracy and 88% recall. Logistic Regression, while simple and interpretable, has a slightly lower accuracy of 76% and recall of 79%. Meanwhile, the FNN model shows moderate performance with 76% accuracy but lower recall and F1-score compared to other models.

5.2 Case Studies and Real-World Applications

5.2.1 Application in Hospitals

This model can incorporate early detection of diabetes into a hospital's entire systems for prediction. The prediction model could quickly ascertain if a patient is at risk based on his health data on glucose levels, BMI, age, and several other medical records, leading him for care well before other symptoms appear.

An application of this model in a hospital is the hospital patient management systems, through which it can be accessed by doctors or health professionals to assist in decision-making processes, such as taking the prediction model on a routine check-up patient and referring him for further medical tests if the model deems him at risk.

5.2.2 Use by General Practitioners

The proposed prediction model could be of value to a general practitioner because it is possible that most patients visit his practice more often than other patients, but do not always have the kind of laboratory data or resources available from a specialized center for dealing with diabetes patients. Integrating the predictive model into their practices would give them a quicker and less costly assessment of diabetes risk, particularly for those patients who are more likely to develop the condition based on lifestyle or family history.

In addition, he can also use it to follow those patients over long periods and thus might notice changes in the parameters of their health and intervene before diabetes ensues.

5.2.3 Potential for Remote Health Monitoring

From the point of view of advancement in the usage of wearable technologies and mobile health applications, thus, it can be said that it has great potential for using models for remote monitoring of populations at risk for diabetes. By integrating the prediction model into wearables, patients will be able to collect health parameters such as glucose levels, BMI, and physical activity and constantly receive predictions/alerts in real-time.

This may allow healthcare providers to monitor their patients remotely and take necessary actions in cases where it is required to avoid visiting them often and make the most healthy outcomes possible for such populations at risk.

5.3 Challenges Faced During Implementation

5.3.1 Data Quality Issues

The data quality was one of the major challenges faced during the implementation of diabetes prediction models. The Pima Indians Diabetes dataset, though much popular, also has its limitations. Missing or erroneous values in the features hamper model performance. Even after using the imputation methods for replacement purposes of missing values, imperfections in the dataset may still have had an influence on the predictions made by the model.

Moreover, in practice, real data is usually mixed with noise, for example, erroneous readings or outliers, which can affect the reliability of the predictions. Advanced outlier detection techniques and preprocessing of data should be considered in the next versions of the model to make it capable of handling these types of data.

5.3.2 Model Overfitting

Overfitting was another challenge, particularly in simpler models like Logistic Regression. Techniques like cross-validation and hyperparameter tuning were applied to mitigate overfitting. More complex models, such as LightGBM, demonstrated better generalization but still required careful monitoring to avoid overfitting.

Future work should emphasize regularization techniques and larger, more diverse datasets to further enhance model generalization.

5.3.3 Computational Complexity

Training and deploying some models, particularly Random Forest and Feedforward Neural Networks, demanded higher computational resources compared to simpler models. While Random Forest and LightGBM offered higher accuracy, they required more memory and processing power.

This highlights the typical trade-off between model complexity and resource consumption. In future deployments, cloud platforms (AWS, Google Cloud) and mobile-optimized versions can address these computational challenges by providing scalable infrastructure.

5.4 Ethical and Legal Considerations

5.4.1 Patient Data Privacy

The application of patient data in machine learning models calls up serious important privacy and security issues. Patient data must be guarded in a manner conforming to the privacy law: e.g., GDPR (General Data Protection Regulation) and HIPAA (Health Insurance Portability and Accountability Act). Any deployment of the diabetes prediction model in healthcare settings must take encryption, anonymization, and secure storage of data seriously to safeguard patients' unique information.

5.4.2 Bias in Machine Learning Models

It is possible for machine learning models to inadvertently learn and perpetuate biases in data. When a poor performance results from training predominantly with a specific population working data, then this specific population might have been considered for training data. To counter bias in prediction, it is thus important to have training data that are diverse and representative of the general population. Ethical considerations regarding fairness and transparency must also be introduced so as not to allow the model to propagate inequality in access to health care.

5.4.3 Regulatory Compliance (GDPR, HIPAA)

Compliance with regulatory standards would entail that every healthcare application using machine learning should be aligned with laws like HIPAA (in the United States) and GDPR (in the European Union). These regulations comprise guidelines to safeguard personal health information and

enforce strict rules on access, storing, and sharing this data. Therefore, developers need to assure that any personal health data is processed only for the intended purposes and under the consent of the persons involved.

Chapter 6: Conclusion and Future Work

6.1 Summary of Findings

The aim of this project was to create and implement artificial intelligence models to predict diabetes using the Pima Indians Diabetes dataset. Multiple models were explored, including Logistic Regression, Random Forest, Support Vector Machine (SVM), LightGBM (LGBMClassifier), and Feedforward Neural Network (FNN). Among these, the LightGBM and Feedforward Neural Network models achieved the highest performance, demonstrating superior accuracy, recall, and ROC-AUC scores. The models, particularly the LightGBM and FNN, showed the ability to identify diabetic individuals with minimal false negatives, highlighting their potential for early diagnosis and intervention in healthcare environments.

6.2 Limitations of the Study

The model was able to perform well, but its performance was limited by the quality and sizability of the data within the said scope. The Pima Indians Diabetes dataset, while widely accepted, may be unable in fully modeling the subtlety and complexity of the risk factors with respect to diabetes across diverse populations because it contains only 768 instances. Additionally, the question of how well the model generalizes to other populations outside the Pima Indian population warrants further scrutiny.

6.3 Recommendations for Improvement

To further enhance model performance, additional features such as lifestyle factors, family medical history, and genetic information could be integrated into the dataset. Incorporating these features would help build a more comprehensive risk stratification model and boost the predictive capabilities of the models. Moreover, employing more advanced algorithms, such as deep learning architectures, could further improve predictive performance, especially when applied to larger and more complex datasets.

6.4 Future Research Directions

Research to come may involve the applicability of this diabetes prediction model with the use of wearable devices and mobile health applications, for computing live health statuses of persons subject to the prediction model. A thorough investigation of the effects brought forth by ambient conditions such as dietary patterns, exercise, and other surrounding ambient variables would certainly sharpen the prediction accuracy significantly. Instead, this prediction model can be generalized when the existing cohort or the data used is further improved by enlarging the size of the population represented in the dataset.

References

Chawla, N.V., Kegelmeyer, W.P., Hall, L.O., et al., 2002. *SMOTE: Synthetic Minority Oversampling Technique*. Journal of Artificial Intelligence Research, 16, pp. 321-357.

Grinberg, M., 2018. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Hunter, J.D., 2007. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), pp. 90-95.

Kumar, P., Choudhary, M., & Gupta, A., 2020. *Cloud Computing for Machine Learning: Applications and Opportunities*. Springer Nature.

McKinney, W., 2010. *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, pp. 51-56.

Oliphant, T.E., 2006. *A Guide to NumPy*. Trelgol Publishing.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al., 2011. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, pp. 2825-2830.

Albreiki, M., Al-Qudah, M. A., & Houssein, E. H. (2021). *Diabetes prediction using machine learning: A review*. Health Information Science and Systems, 9(1), 1-13.

Breiman, L. (2001). *Random forests*. Machine Learning, 45(1), 5-32.

Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. Machine Learning, 20(3), 273-297.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). Wiley.

Khan, S. M., et al. (2018). *Impact of BMI on diabetes incidence in adults: A meta-analysis*. Diabetes Care, 41(1), 104-111.

Rashid, T., et al. (2021). *Machine learning in healthcare: Applications and challenges*. Computational Intelligence, 37(3), 678-690.

Smith, J. (1988). *Pima Indians Diabetes Dataset*. UCI Machine Learning Repository. <https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>

Ganie et al., "Analyzing Classification and Feature Selection Strategies for Diabetes Prediction Across Diverse Datasets," *Frontiers in Artificial Intelligence*, Volume 7 - August 21st, 2024. <https://doi.org/10.3389/frai.2024.1421751>

Doğru et al., "Super Ensemble Learning Model for Early Diabetes Risk Prediction," *Frontiers in Artificial Intelligence*, August 21st, 2024. <https://doi.org/10.3389/frai.2024.1421751>

Tasin et al., "Automatic Diabetes Prediction Using Machine Learning Techniques," *PMC*,

December 14th, 2022. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10107388/>

El-Bashbishy et al., "Deep Learning-Based Early Diabetes Prediction Using Pediatric Dataset," *Frontiers in Artificial Intelligence*, August 21st, 2024. <https://doi.org/10.3389/frai.2024.1421751>

MDPI Review Article: "Current Techniques for Diabetes Prediction: Review," October 29th, 2019 <https://www.mdpi.com/2076-3417/9/21/4604>

American Diabetes Association. (2023). Understanding Diabetes Diagnosis. Retrieved from <https://diabetes.org/about-diabetes/diagnosis>

Mayo Clinic. (2024). Diabetes - Diagnosis and Treatment. Retrieved from <https://www.mayoclinic.org/diseases-conditions/diabetes/diagnosis-treatment/drc-20371451>

National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK). (2025). Diabetes Tests & Diagnosis. Retrieved from <https://www.niddk.nih.gov/healthinformation/diabetes/overview/tests-diagnosis>

PMC. (2021). Artificial Intelligence-Based Diagnosis of Diabetes Mellitus Using Fundus Photography Combined with Traditional Chinese Medicine Diagnostic Methodology. Retrieved from <https://pmc.ncbi.nlm.nih.gov/articles/PMC8081616/>

WebMD. (2024). Diagnosis of Diabetes & Prediabetes Overview. Retrieved from <https://www.webmd.com/diabetes/diagnosis-diabetes>