EARLY DETECTION OF DIABETES USING SUPERVISED LEARNING ALGORITHMS FOR ROBUST PREDICITVE MODELING

BY

AKASUKPE OGHENEKARO CLINTON
(19CJ025767)

JUNE 2024

EARLY DETECTION OF DIABETES USING SUPERVISED LEARNING ALGORITHMS FOR ROBUST PREDICITVE MODELING

AKASUKPE OGHENEKARO CLINTON
(19CJ025767)

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & INFORMATION ENGINEERING, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF ENGINEERING DEGREE (B. Eng.) IN COMPUTER ENGINEERING, COVENANT UNIVERSITY, OTA, OGUN STATE, NIGERIA.

SUPERVISOR
DR. OLOWOLENI JOSEPH

JUNE 2024

DECLARATION

I hereby declare that I carried out the work reported in this project in the Department of Electrical and Information Engineering, Covenant University, under the supervision of Dr. Olowoleni Joseph. I also solemnly declare that to the best of my knowledge, no part of this report has been submitted here or elsewhere in a previous application for the award of a degree. All sources of knowledge used have been duly acknowledged.

…….

AKASUKPE OGHENEKARO CLINTON

19CJ025767

CERTIFICATION

This is to certify that the project titled "EARLY DETECTION OF DIABETES USING SUPERVISED LEARNING ALGORITHMS FOR ROBUST PREDICITVE MODELING" by AKASUKPE OGHENEKARO CLINTON meets the requirements and regulations governing the award of the Bachelor of Engineering Computer Engineering degree of Covenant University and is approved for its contribution to knowledge and literary presentation.

Supervisor: Sign: _____

Name: Dr. Olowoleni Joseph Date: _____

Internal Examiner: Sign: _____

Name: _____ Date: _____

Head of Department: Sign: _____

Name: Dr. Isaac A. Samuel Date: _____

DEDICATION

I dedicate this project to God Almighty for the unrelenting grace and wisdom He gave to me to see this project to its completion. I also dedicate this project to my loving parents, Mr. and Mrs. Akasukpe for all their support throughout my years of study.

ACKNOWLEDGEMENTS

First, I would like to acknowledge and thank God Almighty, for the strength and grace he has bestowed upon me during the course of this project. From the first moment to the last, He has guided me, directed me and led me to the completion of this project.

Next, I would like to greatly appreciate the management of Covenant University, led by the Chancellor, Bishop David Oyedepo, who have continuously strived and worked hard to ensure that the university stands out, raising a new generation of leaders. The insistence on diligence, responsibility and capacity building as core values has been instrumental to the completion of this work.

I would also like to acknowledge the department of Electrical and Information Engineering for the impartation of knowledge needed for this project, especially Dr. Isaac, the Head of Department. He consistently made sure we adhered to deadlines, no matter what. Special thanks to Dr. Omoruyi, for all the knowledge gained from the programming courses lectured by him.

The heartiest of thanks goes to my supervisor, Dr. Olowoleni, for his unwavering support and guidance through the course of this project. He continuously pushed me to bring out the best in me, even when I could not see it myself. He ensured I stayed equipped with all the knowledge I needed to make this project a reality. For that, I am truly grateful.

Finally, to my parents, extended family, friends and well-wishers, who have offered moral and emotional support in their own way, thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN:	Artificial Neural Network
API:	Application Programming Interface
BMI:	Body Mass Index
CPU:	Central Processing Unit
CSS:	Cascading Style Sheets
CSV:	Comma-Separated Values
DT:	Decision Tree
DTC:	Decision Tree Classifier
EDA:	Exploratory Data Analysis
FBS:	Fasting Blood Sugar
GBM:	Gradient Boosting Machine
HTML:	Hypertext Markup Language
IDF:	International Diabetes Federation
JSON:	JavaScript Object Notation
KNN:	K-Nearest Neighbors
LGBM:	Light Gradient Boosting Machine
NB:	Naïve Bayes
NN:	Neural Network
OGTT:	Oral Glucose Tolerance Test
RAM:	Random Access Memory
RF:	Random Forest
SVC:	Support Vector Classifier
SVM:	Support Vector Machine
SVR:	Support Vector Regression
UML:	Unified Modeling Language
VGA:	Video Graphics Array
WSGI:	Web server gateway Interface

ABSTRACT

Diabetes remains a significant global health challenge, especially in low and middle-income nations. Early detection and management are crucial to mitigate long-term complications and alleviate healthcare burdens. Traditional diagnostic methods, while effective, are often time-consuming and require skilled personnel. This project aims to develop a predictive model using logistic regression and other algorithms for the early detection of diabetes. The goal is to harness supervised learning algorithms to accurately predict diabetes from patient data, providing a reliable and efficient tool for early diagnosis. The process involved collecting a comprehensive dataset, performing exploratory data analysis, selecting appropriate algorithms, and evaluating their performance. Logistic regression was chosen for its superior performance in discerning patterns indicative of diabetes. The model was integrated into a web application designed for use by medical professionals, offering a user-friendly interface to facilitate diagnosis. The test accuracies of the models trained were: 78.13% for Gaussian Naïve Bayes, 78.91% for Logistic Regression, 75.00% for KNN, and 65.63% for the Decision Tree classifier. The results show that logistic regression had the best overall performance among the models tested and the chosen model was integrated into a web application. Future work could focus on enhancing the model's accuracy and exploring other machine learning techniques to further improve early detection capabilities.

Keywords: Diabetes, Early Detection, Logistic Regression, Machine Learning Predictive Modeling.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Diabetes mellitus, also known as diabetes, stands as a significant global health concern characterized by elevated blood sugar levels, impacting millions worldwide. In 2019, the International Diabetes Federation (IDF) estimated that in Africa, encompassing low and middle-income nations, approximately 19 million adults aged 20–79 years were living with diabetes, indicating a prevalence of 5.2%. This region also records the highest proportion of undiagnosed diabetes cases, with more than two-thirds (67%) of individuals with diabetes unaware of their condition. Timely detection and effective management are critical to mitigate long-term complications and alleviate healthcare burdens. In recent years, supervised learning algorithms have emerged as promising tools for early diabetes detection. Research efforts, such as the comprehensive study conducted by [1], have delved into employing machine learning and deep learning approaches for diabetes detection. Their investigations unveiled the potential of advanced computational techniques in improving the accuracy and efficiency of diabetes detection systems. Through the utilization of supervised learning algorithms like support vector machines and random forests, these studies demonstrated the capability to develop robust predictive models for early diabetes identification.

Additionally, work by researchers highlighted in [2] focused on utilizing machine learning models specifically for early detection and precise classification of type 2 diabetes. Their emphasis on the significance of supervised learning algorithms in achieving accurate classification of diabetes subtypes underscores the potential for tailored intervention strategies and enhanced early detection model accuracy. Furthermore, the exploration into early detection of type 2 diabetes mellitus using machine learning-based prediction models, as detailed in [3], underscores the necessity for sophisticated predictive modeling techniques. Continuous updates and enhancements in early diabetes detection are crucial, emphasizing the importance of

harnessing the power of supervised learning algorithms to develop dynamic and accurate prediction models.

In the contemporary healthcare landscape, technology plays a pivotal role in driving a shift towards predictive disease management, particularly for prevalent conditions like diabetes. Traditional detection methods often suffer from delays and invasiveness. Leveraging advanced technologies and supervised learning algorithms presents an opportunity for early and efficient diabetes detection. These algorithms analyze diverse health data, enabling the development of robust predictive models for timely identification and personalized healthcare. The proposed project aligns with this broader trend of data-driven healthcare solutions, aiming to utilize cutting-edge technology to develop a model for early diabetes detection. By harnessing the potential of artificial intelligence and supervised learning algorithms, this endeavor seeks to positively impact public health outcomes and contribute to the advancement of diabetes management practices.

1.2 Significance and Motivation of the Study

The prediction of diabetes using supervised machine learning is a significant area of research in healthcare technology. Diabetes is a chronic disease that affects millions of people worldwide and can lead to severe complications such as heart disease, kidney failure, and blindness [4]. Early detection and management of diabetes can significantly reduce the risk of complications and improve the quality of life for patients [5]. The motivation for this study is to develop an accurate and reliable system for detection of diabetes using supervised machine learning algorithms. Previous studies have employed various machine learning approaches to predict diabetes, but most of them focused on a single accuracy measure and used the open-source Pima Indian dataset [4]. Therefore, this study aims to evaluate the proposed prediction system based on multiple accuracy measures [4]. Moreover, the study conducted in Afar regional state, Northeastern Ethiopia, highlights the importance of context-specific information for program planners and policymakers to prioritize the more affected groups [5]. The application of supervised machine learning algorithms for classification and prediction of type-2 diabetes disease status can facilitate diagnostic activity with suggestive and informative diagnostic results for physicians and lead to early intervention for clients [5].

1.3 Problem Statement

Approximately 5.8% of Nigeria's population has diabetes, with a significant portion of cases going undiagnosed, including children and adolescents. In 2019, diabetes caused around 2 million deaths, with 48% occurring in individuals under 70 years. Diabetes leads to amputations and visual defects, affecting communities. High sugar-sweetened beverage (SSB) consumption, noted in 49% of adults and 63% of children between 2011 and 2014, contributes to diabetes. Early detection is crucial to mitigate severe outcomes, prompting a proposal for a predictive model using individual data. Information from similar text [6], [7] complement data from the International Diabetes Federation (IDF), highlighting the global rise of diabetes in disability rankings from tenth to ninth place [8]. This emphasizes the urgency of proactive, data-driven approaches for improving public health outcomes.

1.4 Aim and Objectives

Aim

The aim of this project is to develop a robust predictive model for the early detection of diabetes using supervised learning algorithms.

Objectives of Study

In accomplishing the above-mentioned aim, the following are the objectives:

i. Collect and clean data on individual features influencing diabetes, ensuring the dataset is comprehensive, accurate, and free of irrelevant or skewing information.
ii. Use visualization tools to select the specific features which will be used to develop the models based on their correlations.
iii. Split the data into training and testing data and afterwards train and test the pre-built models.
iv. Select a model out of the groups of models based on its performance measure as compared to that of the other models and deploy the selected model on a web application.

1.5 Methodology

The flowchart in figure 1.1 represents the steps to be taken to achieve the project.

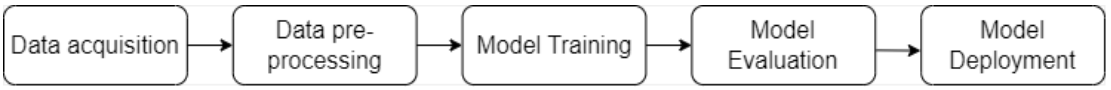

Figure 1.1: Block Representing Project Methodology of the Model

Each step is explained as follows;

a) **Data acquisition:** Data samples, including patient health records and relevant medical information, are collected from healthcare databases.

b) **Data Preparation/Preprocessing:**
   i. Clean the dataset: Handle missing values, outliers, and inconsistencies.
   ii. Feature engineering: Extract relevant features from the dataset and preprocess them for model training.
   iii. Split the dataset: Divide the dataset into training, validation, and test sets to evaluate model performance.

c) **Model Training:**
   i. Implement various supervised learning algorithms including Naive Bayes, Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree Classifier using Python libraries such as scikit-learn.
   ii. Train each algorithm on the training dataset.

d) **Model Evaluation:**
   i. Assess the performance of each model using the validation set, considering metrics like accuracy, precision, recall, and F1-score.
   ii. Select the algorithm with the best performance based on evaluation metrics.

e) **Model Deployment:**
   i. Develop a user-friendly web application using frameworks like Flask or Django.

ii. Deploy the selected algorithm on the web application, allowing users to input their health parameters and get results.

1.6 Scope and Limitations of the Study

The scope of the study is dedicated to developing an innovative system for the early detection of diabetes, harnessing the capabilities of supervised learning algorithms to construct a robust predictive model. The primary goals of the system extend beyond accurately identifying the presence of diabetes; they also include elevating predictive modeling capabilities for early intervention and management. The study's scope encompasses the following pivotal elements:

i) **Feature Importance and Interpretability:** The accuracy of the early detection model will be complemented by an analysis of feature importance, providing insights into the factors contributing to diabetes prediction.
ii) **Patient Data Analysis:** The system will process diverse data sources, including medical records and diagnostic reports, written in various languages.
iii) **Individualized Risk Assessment:** The system will be capable of detecting patterns and anomalies in patient data to enhance predictive modeling accuracy.

The study aims to develop a system that would allow for early detection of diabetes using supervised learning algorithms for robust predictive modeling; there are certain limitations to consider:

i) Data Imbalance and Representativeness: The efficacy of the predictive model may be impacted by potential imbalances in the training data, leading to underrepresentation or misrepresentation of certain demographic groups. This could affect the model's generalizability to diverse populations.
ii) Inherent Variability in Health Data: The inherent variability in health data, including lifestyle factors and biomarkers, poses a challenge. The model's accuracy may be affected by the dynamic nature of these variables and the evolving understanding of diabetes risk factors.
iii) Dependency on Available Features: The accuracy of the predictive model is contingent on the availability and quality of features in the input data.

Incomplete or inadequately detailed health records may limit the model's ability to make accurate predictions.

iv) **Interpretability of Predictive Model:** The inherent complexity of some supervised learning algorithms might result in a lack of interpretability of the predictive model. Understanding the decision-making process of the model may be challenging, especially in a healthcare context where interpretability is crucial.

1.7 Project Report Organization

The project report is organized in the following manner:

Chapter One is the introductory chapter. It will include the background study of the project, the motivation for the project, the aim and objectives of the project, the scope of the study, the methodology and limitations of the project.

Chapter Two of the project provides a comprehensive review of literature pertaining to the early detection of diabetes using supervised learning algorithms for predictive modeling. It begins with an overview of diabetes mellitus, including its historical context and clinical manifestations. The chapter explores various detection methods and challenges in adaptation, emphasizing the significance of early detection in managing diabetes. It thoroughly analyzes scholarly literature on the topic, aiming to inform the project's methodology and address gaps in research for advancing early diabetes detection strategies.

Chapter Three will focus on the approach utilized to execute the project to attain the project's aim.

Chapter Four will cover the results obtained from the training, validation, evaluation, and deployment of the system.

Chapter Five will contain the conclusion of the project. It outlines the challenges faced in accomplishing the project and the contributions that can be made to the project as well as recommendations for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter provides a concise overview of diabetes and defines relevant terms associated with the condition, as well as terminology pertinent to machine learning. It reviews diabetes and its current detection methods, examining the challenges inherent in these approaches. The chapter will encompass a thorough review of automated diabetes prediction and diagnosis methods utilizing machine learning, highlighting their advantages and addressing fundamental issues and solutions crucial to diabetes prediction.

2.2 Overview of Diabetes Mellitus

Diabetes mellitus also simply known as Diabetes stands as a pervasive chronic metabolic disorder characterized by elevated blood glucose levels, attributed to deficiencies in insulin secretion, insulin action, or a combination of both [9]. This multifaceted condition has emerged as a significant global public health concern, with an estimated 463 million adults grappling with diabetes in 2019, a number that is projected to surge to 700 million by 2045 [10]. The escalating prevalence of diabetes necessitates a comprehensive understanding of its types, diagnostic methodologies, and the intricate interplay between genetic and non-genetic factors. Diabetes manifests in various forms, encompassing type 1, type 2, gestational diabetes, and other classifications, with diagnostic criteria spanning clinical presentations, biochemical tests, and genetic markers [10]. The complexity of diabetes etiology involves a dynamic interplay between genetic factors and environmental influences [10]. Ongoing research in the field of molecular genetics has uncovered a multitude of mutations and single nucleotide polymorphisms within genes pivotal to glucose metabolism and the regulation of pancreatic cell development and function [10]. A crucial facet of addressing the diabetes epidemic is early detection, an imperative for effective management and the prevention of debilitating complications. Research in [10] underscore the significance of proactive

screening, particularly in underdeveloped nations, to curtail late-stage diagnoses and enhance overall public health outcomes. Diabetes represents a formidable public health challenge, necessitating not only early detection but also continuous research efforts to unravel the intricate classifications, diagnostic methodologies, etiological factors, and molecular genetics underpinning the condition. The quest for improved diagnostic tools, therapeutic strategies, and the mitigation of chronic complications remains an ongoing priority in the global battle against diabetes [10].

2.2.1 Symptoms of Diabetes

Diabetes presents a diverse array of symptoms that extend beyond the conventional understanding of its metabolic impact. While the hallmark of diabetes is elevated blood glucose levels, individuals with the condition may also experience a spectrum of physiological and psychological symptoms, contributing to its complex clinical picture and some of which are highlighted below.

A) Pain Locations: Diabetes-related discomfort may manifest in various areas, including between the fingers, the back, neck, chin, left arm, or upper belly. These pain locations can serve as potential indicators of the systemic impact of diabetes on different bodily regions.

B) Discomfort Examples: The discomfort associated with diabetes often takes on diverse forms, such as a sense of squeezed tension, mimicking the sudden onset of a heart attack, or subtle tremors. Understanding these discomfort nuances is vital for accurate symptom interpretation and timely intervention.

C) Circumstances of Pain: Contrary to common expectations, pain related to diabetes may occur even at rest. This distinctive characteristic underscores the pervasive nature of the condition and highlights the importance of vigilant monitoring and early detection.

D) Whole-Body Impact: Diabetes can exert a profound influence on the entire body, leading to symptoms like dizziness, fatigue, light-headedness, and episodes of cold perspiration or sweating. These systemic manifestations underscore the need for holistic healthcare approaches that consider the multi-faceted impact of diabetes.

E) Gastrointestinal Symptoms: Beyond the typical metabolic symptoms, diabetes can also manifest as gastrointestinal issues. Heartburn, diarrhea, and vomiting are among the gastrointestinal problems that individuals with

diabetes may encounter, adding a layer of complexity to the clinical presentation of the condition.

F) Poitrine: Diabetes can further manifest as chest-related symptoms, including pain or tightness in the chest. Recognizing these cardiovascular implications is crucial for a comprehensive understanding of diabetes and its potential impact on cardiovascular health.

2.2.2 Types of Diabetes Mellitus

There are many distinct types of diabetes, also known as diabetes mellitus, each uniquely affecting the pancreas and other organs. They include the following:

I. Type 1 Diabetes

Type 1 diabetes is an autoimmune condition where the immune system targets and destroys insulin-producing beta cells in the pancreas. This leads to an insulin deficiency, affecting blood sugar regulation. The causes involve a mix of genetic and environmental factors triggering an immune response. Symptoms include elevated blood sugar, increased thirst, urination, weight loss, and fatigue. Without insulin, the body can't effectively use glucose, resulting in the breakdown of muscle and fat for energy. Management revolves around insulin therapy, with individuals requiring regular injections or insulin pumps. Continuous monitoring and insulin dose adjustments are crucial for stable blood sugar control.

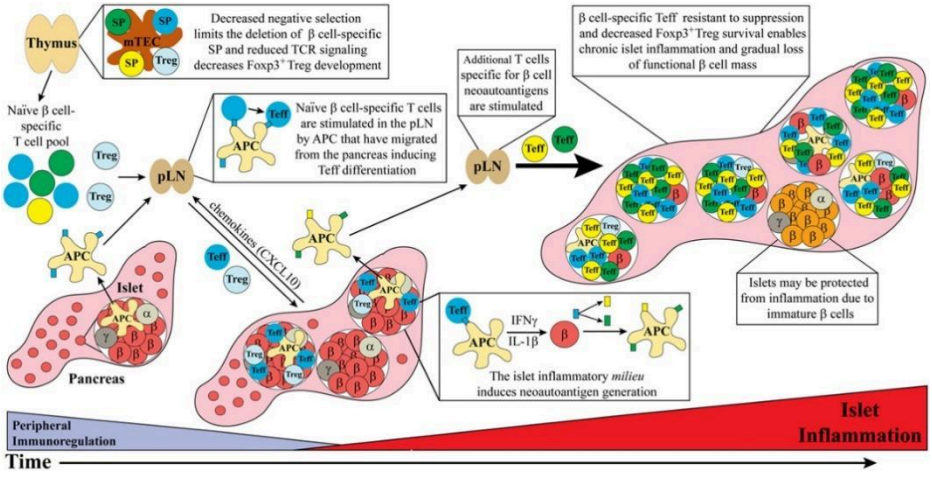

Figure 2.1: Overview of how Type 1 Diabetes affects body organs

II. Type 2 Diabetes

Type 2 diabetes is a metabolic disorder characterized by insulin resistance, where cells do not respond effectively to insulin. It typically develops later in life and is linked to factors like genetics, poor diet, sedentary lifestyle, and obesity. While the pancreas may produce insulin, the body's cells resist its effects, leading to elevated blood sugar levels. Symptoms include increased thirst, frequent urination, unexplained weight changes, and persistent fatigue due to inefficient glucose use. Management involves lifestyle changes such as a healthy diet, regular exercise, and weight control. Medications may be prescribed to enhance insulin sensitivity or stimulate production. Unlike Type 1 diabetes, insulin therapy is not always the first treatment for Type 2. Regular monitoring, adjustments in treatment, and a proactive lifestyle approach are vital for stable blood sugar levels and preventing complications.

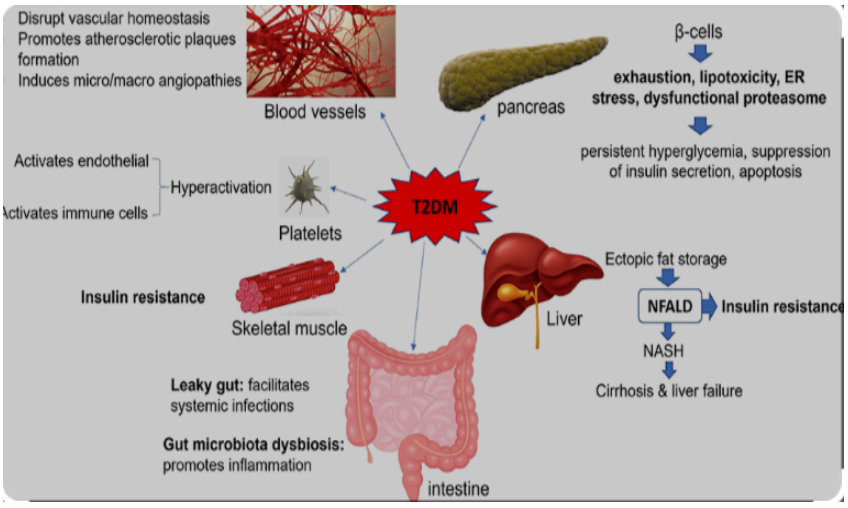

Figure 2.2: Overview of how Type 2 Diabetes affects body organs

III. Gestational Diabetes

Gestational diabetes, a temporary condition during pregnancy, affects blood sugar regulation due to hormonal changes, leading to increased insulin resistance. The pancreas produces more insulin, but cells may not respond effectively, causing elevated blood sugar levels. Symptoms include thirst, frequent urination, and fatigue. Unmanaged gestational diabetes poses risks like excessive birth weight and delivery complications. Management involves lifestyle changes, a balanced diet, and physical activity, with insulin therapy

recommended in some cases. Regular monitoring and adjustments in the treatment plan are crucial for both mother and baby's well-being. While typically temporary, proper management is essential to reduce complications, necessitating close collaboration with healthcare professionals.

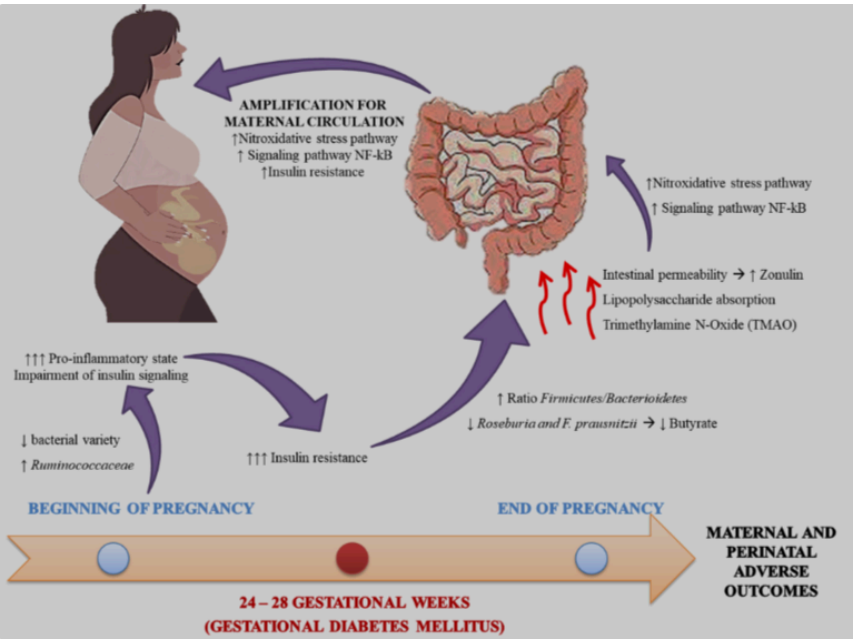

Figure 2.3: Overview of how Gestational Diabetes affects the body

There are other types of diabetes which are not so common and they include: Maturity Onset Diabetes of the Young, Neonatal Diabetes, Wolfram Syndrome, Alström Syndrome, Latent Autoimmune Diabetes in Adults, Type 3c Diabetes, Steroid-induced Diabetes and Cystic fibrosis-related Diabetes.

2.3 Current Diabetes Detection Methods

For diabetes, it is crucial to have alternative tests that are essential for comprehensive detection and cater to the necessity of widespread screening. These tests are designed to assess different aspects of diabetes risk and metabolic health, providing a thorough understanding of the condition. They are as follows:

a) Oral Glucose Tolerance Test (OGTT): Oral Glucose Tolerance Testing is a diagnostic procedure evaluating the body's glucose metabolism. After an overnight fast, a standardized glucose solution is consumed, and blood

samples are collected at intervals to measure glucose levels. This test is essential for identifying abnormalities in glucose regulation, playing a significant role in diabetes diagnosis and assessing metabolic health [11].

b) Hemoglobin A1c Test: The Hemoglobin A1c (HbA1c) test is a crucial tool in diabetes management, providing a comprehensive, non-fasting assessment of long-term blood sugar levels over 2-3 months. With low within-person variability and standardized results, it serves as a reliable diagnostic and monitoring tool. HbA1c is strongly associated with diabetes complications, making it a valuable prognostic marker. In cases where HbA1c reliability is in question, alternative measures like fructosamine and glycated albumin can assess hyperglycemia. Supported by epidemiologic studies, the HbA1c test is indispensable in diabetes care, playing a significant role in diagnosis, management, and monitoring [12].

c) Fasting Blood Sugar Test (FBS): The Fasting Blood Sugar Test is crucial for assessing diabetes risk and overall metabolic health by measuring glucose levels after an overnight fast. Elevated fasting blood sugar signals potential diabetes, leading to further investigation and management. Widely used for its simplicity and effectiveness, regular monitoring is essential for proactive health management and complication prevention in those at risk or managing diabetes.

d) Glycated Albumin Test: The Glycated Albumin Test rapidly evaluates recent blood sugar control by measuring glycated albumin percentage, offering dynamic insights into glucose level changes. Particularly useful for quick assessments and conditions like anemia, it ensures accuracy despite variations in red blood cell lifespan. This test is vital for healthcare professionals to make timely interventions and adjustments for comprehensive diabetes management. Regular monitoring is essential for effective diabetes care, aiding in complication prevention by capturing short-term changes in blood sugar control.

e) Random Blood Sugar Test: Conducted at any time, this test helps identify immediate blood sugar levels and potential indicators of diabetes.

2.3.1 Risk Factors Associated with Diabetes

A risk factor plays an important role in determining whether or not an individual is likely to develop diabetes. Risk factors are the specific behaviors, repeated activities or habits that enhance one's chance of contracting a disease. In this project, diabetes is our center of attention. Risk factors, in most cases, do not exist on their own. They may also be a collection of several distinct factors interacting with one another. For specific types of diabetes, such as Type 2 diabetes, complications stem from factors like insulin resistance and impaired insulin production, affecting blood sugar regulation and organ function. Lifestyle-related variables, including unhealthy diets and sedentary living, contribute to weight gain and obesity, increasing the risk of insulin resistance. Genetic factors, such as family history, age, and sex, influence susceptibility to diabetes. Insulin resistance develops over time due to a mix of poor lifestyle choices and genetic predisposition. Recognizing these risk factors is crucial for healthcare providers to categorize individuals, enabling targeted interventions and preventive measures. Similar to heart diseases, diabetes complications result from a combination of lifestyle choices, genetic factors, and other risks, emphasizing the importance of identifying and addressing these factors for effective management and prevention.

A systematic review meticulously identified diverse risk factors associated with the onset of type 2 diabetes, including serum uric acid, sleep patterns, smoking, cardiovascular disease, hypertension, depression, dyslipidemia, ethnicity, family history, and obesity [13]. Emphasizing the imperative for prospective studies to validate these associations, the review provides valuable insights into the multifaceted nature of type 2 diabetes risk factors [13]. Furthermore, a comprehensive analysis focusing on recent advances in treating and preventing type 2 diabetes highlighted the significance of understanding disease mechanisms. The study underscored the need for effective interventions, proactive prevention measures, early diagnosis, and the development of pharmaceuticals with enhanced efficacy and safety profiles [14]. These seminal studies, as denoted by [13] and [14], hold paramount importance in shaping public health policies and clinical interventions aimed at addressing type 2 diabetes. The comprehensive insights provided enable the development of targeted strategies to mitigate the impact of

identified risk factors, contributing significantly to improving overall public health outcomes.

2.4 Machine Learning

Machine learning is a technique that allows programs and computers to learn things without having to be programmed specifically. Its goal is to enable computers to learn on their own without the need for human intervention. Machine learning is the process of creating computer algorithms that understand and utilize data to learn and improve decision making. Correlations are used by machine-learning based systems to detect patterns in large amounts of data. In today's world, the usage of various machine learning algorithms is becoming more popular and frequent in a variety of fields. Because of the sheer volume and complexity of data involved in building and implementing these machines learning systems, obtaining meaningful knowledge from the machine learning systems is critical. In the Health Industry, such information may be utilized in the diagnosis and treatment of patients in order to enhance the quality of various treatments and services offered. Other bodies, such as corporate organizations, governments, academics, and others may add value to their services by collaborating with consumers and partners. Machine learning systems are extensively utilized in a variety of sectors, including electronics, nursing, criminal detection, expert modelling, and mobile computing, among others [15].

2.4.1 Approaches to Machine Learning

Machine learning techniques are often divided into three categories depending on the quality of the data supplied to the learning system. There are several approaches to machine learning, these are systematic techniques for teaching computers to make data-driven choices utilizing complex algorithms. For years, engineers and clinicians have used different machine learning methods. Some examples of such methods are highlighted below:

i. Supervised learning: In supervised learning, a learner algorithm supplies the machine with examples of inputs and the anticipated outputs they should produce, with the goal of teaching the machine the relationship or function that maps the inputs to the predicted outcomes.

ii. Unsupervised learning: Unsupervised learning occurs when the learning algorithm is not provided any identifiers, enabling it to discover meaning in the data on its own. In fact, unsupervised learning may be considered a goal in and of itself, since it can be used to discover hidden patterns in data.

iii. Reinforcement learning: A computer system uses reinforcement learning to cope with a complicated environment in which a particular goal must be met. The software receives input that is functionally similar to incentives as it navigates its issue space, which it tries to optimize and therefore gets wiser the more it interacts with such environment.

2.4.2 Supervised Machine Learning

Supervised Learning is a kind of Machine Learning technique essentially used for building machine learning models using labelled training data. It allows us to predict the results of future or unknown events. Supervised learning is a subset of artificial intelligence and machine learning, sometimes known as supervised machine learning. It relies on data sets labeled to train computers that can properly categorize data or forecast results. As the data are included in the model, it modifies its weight as parts of the cross-validation procedure. Supervised learning allows businesses to scale solutions for many real-world problems, such as spam categorization in a separate email folder and healthcare for quick infection diagnostics. Below is a picture of the supervised method to machine learning.

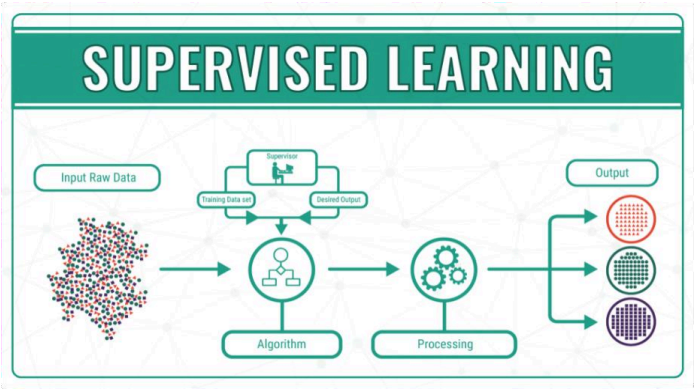

Figure 2.4: Pipeline diagram for Supervised Machine Learning

Machine learning models can be built using either supervised or unsupervised learning techniques. Below, the supervised learning techniques are highlighted with brief explanations:

Classification

Classification is a Supervised Machine Learning Technique, which includes stratification or grouping of items into their target class via list or database. The classification techniques may be employed to estimate the category attribute. The value of this property may be calculated using a different collection of attributes. Because they can handle large data sets, classification methods in the medical industry are extensively utilized and may also serve as a basis for defining a diagnostic procedure and prognosis based on the symptoms and circumstances of the patient. The classification methods consist primarily of two stages: the classification step and the learning phase. The learning phase considers the input as training data and develops a classification system that outputs classification rules. The accuracy of the classifier is controlled using test data during the classification step. At this stage, the prediction accuracy of the classifier is computed. The accuracy of the classifier is the number of test data whose class marks are categorized properly by the algorithm. When the consistency of the classifier is deemed suitable, other potential data tuple class IDs may be identified using the classifier model.

Decision tree

To build classification or regression models, a decision tree structure is utilized. It decomposes a data set into progressively smaller groups while also creating a decision tree to go with it. As a result, a tree is constructed with deciding nodes and branch nodes. A decision node may have two or more branches. A classification or judgment is represented by a leaf node. The root node in a tree refers to the strongest indicator and is the ultimate decision node. Decision tree algorithm can be used to analyze both numerical and categorical data.

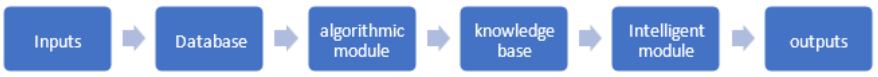

Figure 2.5: Flowchart for Decision tree

Random Forest Classifiers

Random forest classifiers are a subset of ensemble-based learning algorithms. They are easy to build, operate quickly, and have shown exceptional effectiveness in

a number of areas. The random forest method is based on the concept of training numerous "simple" decision trees and then classifying them using the majority vote (mode). Among other advantages, this voting method compensates for decision trees' bad tendency to overfit training data. Random forests use the method known as bagging on individual trees in the ensemble during the training stage.

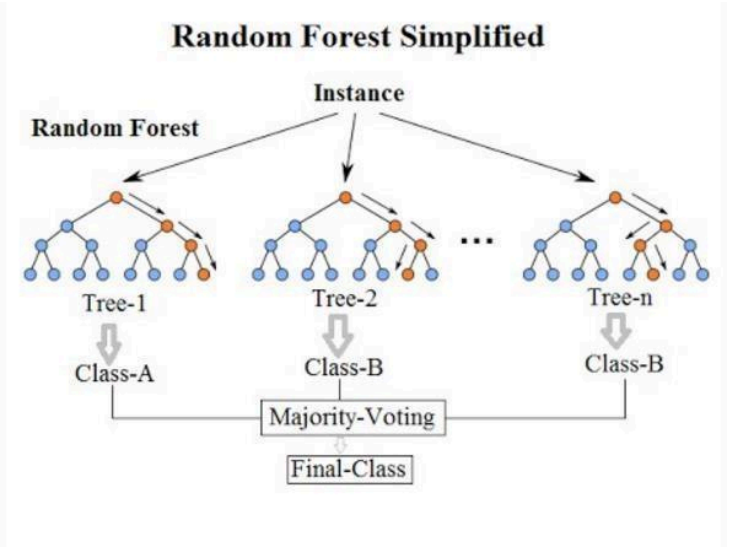

Figure 2.6: Random Forest classifier majority voting node

Regression

Regression is a technique for determining the probability of a single result in the presence of many factors. It is mainly used for predicting and simulation. For example, utilizing various factors such as usability, consumer order, and competition, logistic regression may be utilized to determine a certain cost price of an item. The main goal of regression, in this case, is to reveal a common connection between (at least two) variables inside a particular knowledge space.

Clustering

It is a technique for categorizing variable sets in such a manner that identical objects/variables are grouped together. Clustering, often known as cluster analysis, is a machine learning method that organizes unlabeled datasets. It is described as "a method of grouping data points into distinct clusters, consisting of similar data points, in which the objects with potential similarities stay in a group that has fewer or no similarities with another group." A group may be thought of as a grouping of linked components. Clustering may be thought of as an unsupervised learning

process, since it is acquired only from observation of unrelated characteristics in the dataset. Clustering neither comprehends nor takes into account the notion of class. Since a result, it is applicable to research involving large quantities of data, as only a subset of the data is known. K-means clustering is a partitioning technique that is often used to reduce sum-squared errors [16]. Although the need to specify the number of clusters in advance is a major drawback, k-means clustering is widely used in biology, medicine, anthropology, marketing, and economics, among other fields.

Association Rule Learning

Associating two or more items together is a method through which a pattern is discovered in a single transaction based on the connection between a particular object and a group of objects. The process of identifying recurring patterns, correlations, and connections among many items in a collection is utilized on a regular basis. This correlation technique can be used in the healthcare industry to discover the complex link between the effects of a sickness, the clinical status of various people with the same illness, and the relationship between numerous diseases that affect the human body [16]. When the Associative Rule is found, researchers may be able to create proof-based hypotheses about the symptoms that underpin a disease.

2.5 Application of Machine Learning in Healthcare

The healthcare industry is a critical field in which machine learning approaches can be implemented. As a result, an appropriate computational approach to discover elusive and useful knowledge about healthcare is required. Because of the exponential increase in the population, it is becoming more difficult to monitor and analyze the enormous amount of patient information. Machine learning offers us with a method of automatically identifying and processing this data, increasing the complexity and resilience of the healthcare system as a result. In recent years, there has been significant development in the application of different machine learning algorithms to medical data in order to identify various illnesses, including diabetes diagnosis, detection of various kinds of cancer, cardiovascular disorders, and others. By analyzing the relationships between organ malfunction and the underlying reasons

of failure, as well as the impact of symptoms that occur spontaneously in patients, it will be possible to establish appropriate measures to assist the patient in surviving this disease. There is a significant quantity of medical data generated by technology and processes in healthcare that is both too large and complex to be assessed using traditional techniques of evaluation.

Machine learning classification algorithms play a crucial role in healthcare due to their efficiency in handling large datasets. Notable methods include Gaussian Bayesian Networks, Decision Trees, Neural Networks, Genetic Algorithms, and Support Vector Machines. While these techniques offer advantages in illness detection, their performance varies based on data features and research context. Big data analysis in healthcare yields significant benefits, but challenges such as clinical implementation and ethical considerations must be addressed. Machine learning surpasses traditional biostatistical methods in flexibility and scalability, allowing applications in risk stratification, diagnosis, and survival forecasts. These algorithms can integrate diverse data sources for comprehensive predictions, but challenges like data preprocessing, model training, and system refinement persist. Ethical concerns, including medico-legal implications, physician knowledge of machine learning, and data privacy, pose additional challenges in healthcare delivery. In [17] we explore some of the advantages and difficulties of big data and machine learning in health care in this Review.

2.5.1 Machine Learning Algorithms Applied to Medical Datasets

Using electronic medical data, Data Analysts, scientific researchers, and Machine learning and computer science experts actively involved in the field of public health have undertaken many research initiatives in order to be able to predict the occurrence of diabetes. Several machine learning methods have been employed in the past to generate such predictions. In this project the Logistic Regression, Decision Tree Model, Naive Bayes and k nearest neighbor model are implemented for the prediction of diabetes. Below are the reviews of these learning algorithms:

Logistic regression

Logistic regression is a linear classifier method that fits the data and predicts the probability of a target variable using the sigmoid function. More so, due to the

fact that the data set's goal feature is dichotomous, it's suitable for binary classification tasks. This model was taken from the area of statistics. Logistic regression models, according to Witten and Frank, generate projections of likelihood rather than forecasts, so the likelihood that a given instance belongs to each class is calculated for each class [18]. With this in mind, the possibility of prediction must be converted into a binary value (0 or 1) to be used as a classifier. The logistic function is a sigmoid that generates a number between 0 and 1. The categorization is done on the basis of probability estimates. The logistic regression theorem seems to restrict the cost function to a range of 0 to 1. The equation for the logistic regression theorem is indicated in equation (2.1) below.

$$\mathbf{y} = \mathbf{ln}\left(\frac{\boldsymbol{P}}{-\boldsymbol{P}}\right) = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x}\ \mathbf{1} \tag{2.1}$$

The anticipated output is y, the bias is b0, and the single-input vector (x) is b1 according to the preceding equation. A coefficient b (constant real value) has been given to the input data column, which will be retrieved from the training results. The Sigmoid function, as illustrated in figure 2.4 below, converts anticipated values into probabilities by mapping every real value between 0 and 1 to another value. In machine learning, sigmoid is used to translate predictions to probabilities.

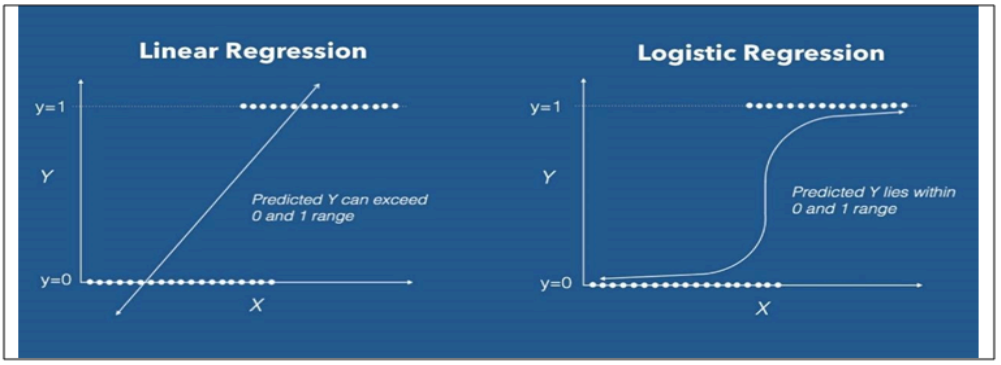

Figure 2.7: Graphical representation of the sigmoid function

Naive Bayes

For classification, the supervised learning technique naive Bayes classifier is employed, which calculates the probability that a given instance belongs to a certain

class. Since it is based on the Bayesian principle, which says that all data set features' values are independent of one another, it's termed naive. For each feature, it estimates the previous probability and the presence of each attribute value (predictor frequency in the training dataset). The probabilities are then used to locate an undefined instance [19]. The equation for the Naïve bayes algorithm is indicated in equation (2.2) below

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{2.2}$$

Full attribute independence is assumed by the statistical Naïve Bayes (NB) classifier. The Naive Bayes classifier works on the following principle:

a. Training Step: Because predictors for a class are intended to be independent and identically distributed, the method uses the training data to estimate the probability distribution parameters. This is considered a chance in advance.
b. Prediction Step: The technique calculates the likelihood function of the collection falling to a class in the case of ambiguous test results. Finally, on the basis of the most relevant postural probability, the method classifies the test data.

Decision tree (DT) classifiers

The Decision Trees algorithm is widely used in medicine. A decision tree functions as a tree-like structure, with each node representing an attribute, each leaf representing the result of a goal, and each branch representing a rule. Decision trees are simple to construct, helpful for target group grouping, and resistant to noisy data and irrelevant information. Although the model's accuracy may be lower than that of other techniques, the rules generated are comprehensible and fit well with large datasets. The equation indicated in equation (2.3) below is used to calculate the tree information gain.

$$H(X) = -\sum_{i=1}^{n} P_i \log_2 P_i \quad (2.3)$$

Decision trees classify datasets by working its way down from the root to a single terminal node, with the leaf node giving classification for data points. Others may use the tree node as a test case, and the growing node edge refers to potential test case solutions. While the addition of sub-nodes increases the uniqueness of the developing sub-nodes, the node's integrity deteriorates as the goal value increases. The decision tree partitions the nodes into all possible variables and then selects the partition that results in the most homogenous sub nodes. This is then performed for each subtree rooted in the newly created node. This method is shown in Figure 2.8

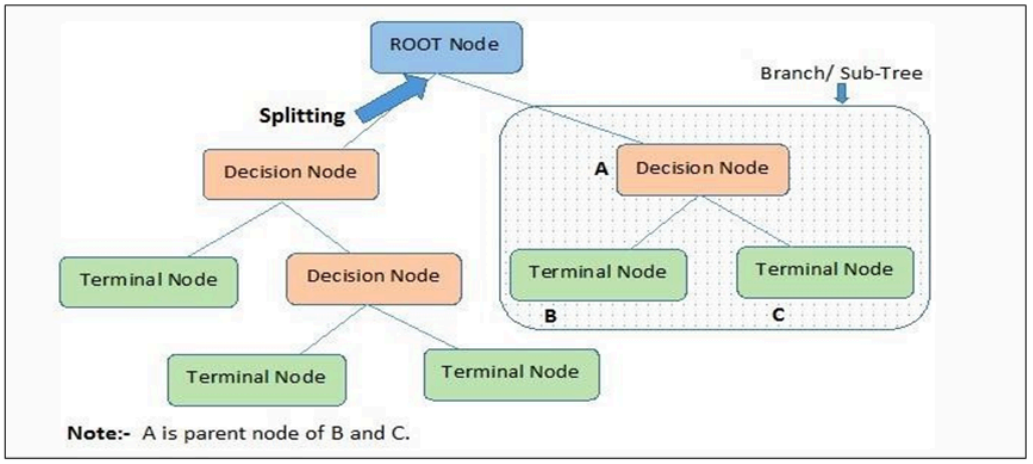

Figure 2.8: Graphical representation of the Decision Trees algorithm

K-nearest neighbor

The k-nearest neighbor (KNN) classification algorithm is a commonly used decision rule in pattern classification. The conventional implementation of this technique, on the other hand, is computationally costly. A K-Nearest-Neighbor algorithm is a data classification technique that examines the data points immediately surrounding it in order to determine the group to which the data point belongs. When determining whether a point on a grid belongs to group A or B, an algorithm examines the states of neighboring points. While the range is chosen randomly, the goal is to get a representative sample of the data. If the majority of points are classified as belonging

to group A, the data point in question is likely to be classified as belonging to group A rather than group B, and vice versa. Because the K-Nearest-Neighbor method does not build a model of the data set in advance, it is an example of a "lazy learner." It performs computations solely in response to a request to pull the data point's neighbors. This simplifies K- NN's applications to data mining.

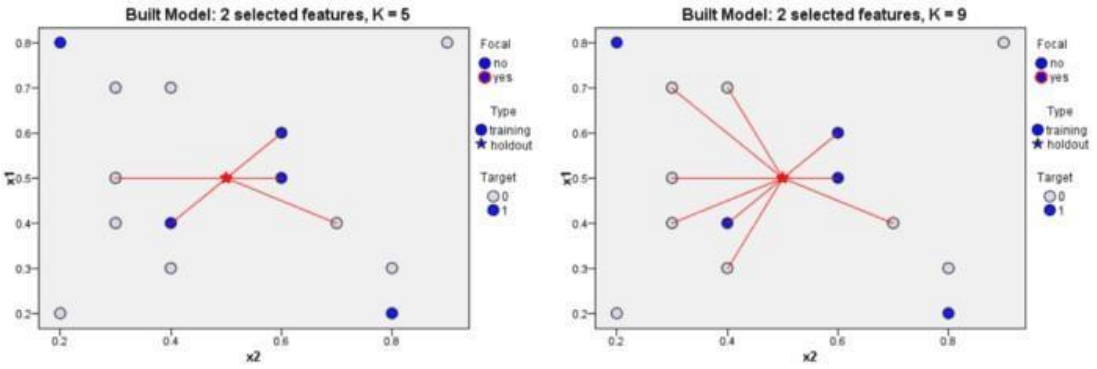

Figure 2.9: K-Nearest Neighbor Classification

2.6 Review of Related Work Done

Numerous research studies have been conducted on diabetes detection using supervised machine learning techniques, leveraging diverse datasets and feature engineering methodologies. Below is a summary of recent research endeavors in the field of diabetes detection and classification found in Table 2.1:

Table 2.1: Summary of Related Literature Based on Diabetes Detection

	Year of Publication	Methodology	Process	Result	GAPS
[20]	2022	KNN, RTF, SVM, BC, An R-based feature selection algorithm and SMOTE were employed.	Data was extracted from NGHA databases, then prepared, cleaned, and pre-processed for consistency and accuracy. Missing values were replaced with means, and age was discretized into four groups. Both	The Bayesian network algorithm performed best, achieving AUC scores of 0.75 and 0.71. The study analysed	The study identified missing and unavailable data as a major factor affecting the model's performance.

	Year of Publication	Methodology	Process	Result	GAPS
			under-sampling and oversampling techniques were employed to tackle data imbalance.	21,431 unique records with 41 attributes.	
[21]	2020	K-Nearest Neighbour, Support Vector Machine, Random Forest.	Data from the Pima Indian diabetic and Germany diabetes datasets underwent pre-processing steps for consistency and accuracy. This included cleaning, normalization, and feature selection. Missing values were handled, and features like age were discretized. To address data imbalance, both under-sampling and oversampling techniques were applied to enhance predictive capabilities.	KNN achieved 71.35% accuracy, SVM 73.43%, and Random Forest topped with 74.47%.	More machine learning algorithms could have been used comparatively.
[22]	2023	Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), AdaBoost, and Bagging models.	Kaggle datasets were used to test SVM, Decision Trees, Random Forest, Naive Bayes, and Logistic Regression for diabetes prediction. Data was balanced and split into test and train sets using 10-fold cross-validation. Performance metrics like accuracy, precision, recall,	The accuracy rates were 96% for GWO-MLP and 97% for APGWO-MLP. The proposed ML framework achieved 95.94% accuracy using	Lack of consideration for additional performance evaluation metrics beyond accuracy.

	Year of Publication	Methodology	Process	Result	GAPS
			and error rates were calculated.	SVM, NB, RF, AdaBoost, and Bagging models for the dataset.	
[23]	2019	SVM, NB, KNN and Decision tree algorithm.	Diabetes data from Medical Centre Chittagong, Bangladesh, including attributes like age, sex, weight, diet, and symptoms, underwent preprocessing by converting numeric values into nominal categories. For instance, age was discretized into three groups. Various machine learning classification techniques were then applied to conduct predictive analysis on the dataset.	Decision tree algorithm had the highest precision of 72%, recall of 74% and f-measure of 72%.	The lack of discussion on the limitations or challenges encountered during the performance analysis.
[24]	2021	Decision tree algorithm, NB, ANN and logistic regression.	Data on global diabetes statistics and cases was extracted from reputable sources like the World Health Organization and pre-processed. ML models (Logistic regression, NB, ANN, C4.5 Decision Tree)	Logistics regression had the best overall performance with a precision of 83.03%, recall of 89.86%, accuracy of	Rather than use ANN another machine learning algorithm such as KNN could have been used for

	Year of Publication	Methodology	Process	Result	GAPS
			were used to predict adult diabetes.	80.42% and F-score of 86.31%.	better comparative result.
[25]	2020	K-means clustering, SVM, KNN, ANN.	Data from 20,578 medical reports underwent preprocessing to remove invalid values, resulting in 10,860 accurate reports. These were used to train a machine learning model to detect diabetes status, ensuring reliability through data cleaning.	Linear SVM was the most accurate algorithm with an accuracy of 99%.	Only three columns were used in order to obtain performance metrics.
[26]	2019	K-means clustering, NB and decision tree algorithm.	Data from diabetic medical records in Indonesia underwent preprocessing via the ETL process, yielding 158 records with 15 attributes. Eight attributes, comprising modified and unmodified diabetes risk factors, were chosen for in-depth analysis. The data was subjected to clustering and classification techniques via the WEKA program to uncover latent patterns in risk factors.	The accuracy of the proposed model was 68% with the highest accuracy on the model for retinopathy prediction.	More algorithms could have been used for risk factor analysis.
[27]	2021	SVM, ANN (Artificial	Data was collected from various demographic and	For accurate detection of	The parameter

	Year of Publication	Methodology	Process	Result	GAPS
		Neural Network), SVR, association rules and hybrid schemes.	health surveys to predict diabetes. The process involved a comprehensive review of data mining techniques applied to diabetes detection and prediction, focusing on data preprocessing, feature extraction, classification models, and performance evaluation.	diabetes, the data needs to be preprocessed along with the application of hybrid techniques.	selection process used is not optimal which in turn affects the accuracy.
[28]	2023	KNN, logistics regression and random forest.	Data from the PIMA Indian Diabetes dataset was used to predict diabetes. The process involved data preprocessing, feature selection, and model training using K-Nearest Neighbor, Random Forest, and Logistic Regression.	Random forest was the most accurate algorithm with an accuracy of 93.4%.	The use of accuracy as the only performance metrics.
[29]	2023	Extreme gradient boosting, SVM, RF and DT.	Data from the Indian Demographics Diabetes dataset was cleaned, converted to CSV, and balanced using SMOTE. PCA was used for feature selection. Machine learning models, including SVM, Random Forest, and XGBoost, were then	The random forest algorithm gave the maximum accuracy and XGboost performed moderately in terms of	There are other risk factors which could have been added in order to obtain better results.

	Year of Publication	Methodology	Process	Result	GAPS
			trained on the balanced dataset.	precision and recall.	
[30]	2022	LR, SVM, DT, RF, NB, NN, KNN and GBM.	Data from a dataset on laboratory tests, lifestyle, and family history was cleaned, preprocessed, and missing values were replaced with mean values. The age feature was discretized into four groups. Under-sampling and oversampling techniques were used to address data imbalance between diabetes and no diabetes samples.	GBM had the highest performance metrics with an accuracy of 95.50%.	Additional variables could have been considered for better result.
[31]	2022	LR, XGBoost, gradient boosting, DT, random forest and LGBM.	The PIMA Indian dataset was cleaned and preprocessed, then split into training and testing sets. Various machine learning algorithms, including Logistic Regression (LR), XG3Boost (XGB), Gradient Boosting (GB) and others were used to predict Type-2 Diabetes Mellitus.	LGBM had the highest accuracy of 95.20%.	The use of accuracy as the only performance metrics.
[32]	2021	DT, LR, RF and KNN.	The study used the PIMA Indian Diabetes dataset to	Random forest gave the best	More inputs could have

	Year of Publication	Methodology	Process	Result	GAPS
			develop a risk score for diabetes mellitus using supervised learning algorithms. The process involved data preprocessing, feature engineering, and model training using algorithms like Logistic Regression, Decision Trees, and SVM.	accuracy of about 92%.	been using in order to obtain a better result.
[33]	2023	KNN and SVM.	Data from electronic health records and public datasets was used for early prediction of diabetes. The process involved data preprocessing, feature selection, and training machine learning models like SVM and KNN.	KNN algorithm had an accuracy of 80.12%.	More machine learning algorithms could have been used comparatively.
[34]	2019	DT, RF, KNN, logistics regression and SVM.	Data was extracted from the PIMA India diabetes dataset, then prepared, cleaned, and pre-processed for consistency and accuracy. Missing values were replaced with means, and age was discretized into four groups.	KNN algorithm had the best accuracy of 80%.	The only performance metric used was accuracy.
[35]	2023	SVC, NB, RF, XGBoost and Ensemble	Data was extracted from the Kaggle dataset, then prepared, cleaned, and	The Ensemble model for random forest	More inputs could have been using in

	Year of Publication	Methodology	Process	Result	GAPS
		model for random forest and SVC.	pre-processed for consistency and accuracy. Feature scaling was performed by normalizing the data, and attributes with a correlation coefficient above 0.1 with the 'Outcome' variable were retained. The data was split into training and testing sets in an 80:20 ratio.	and SVC algorithms outperformed other algorithms with an accuracy of 85.71%.	order to obtain a better result.
[36]	2022	SVM, KNN, RF, Gradient boosting classifier, NB, LR, voting classification.	Data was extracted from the PIMA dataset. The data was cleaned by removing duplicates and null values. Preprocessing involved transforming the raw data into a format suitable for analysis. Data imbalance was addressed using both under sampling and oversampling techniques.	Random forest outperformed other algorithms with an accuracy of 80.7%.	Additional performance metrics could have been used.
[37]	2019	KNN, RF, LR, DT and bagging.	Data from the diabetes datasets underwent preprocessing for consistency and accuracy, including cleaning, normalization, imputing missing values for key	Amongst all other algorithms LR gave the best accuracy of 96%.	Accuracy was the only performance metrics used.

	Year of Publication	Methodology	Process	Result	GAPS
			attributes, and scaling for normalization. Features were also analyzed for correlation.		
[38]	2023	KNN, SVM, RF, NB and stacked ensemble model.	Data was extracted from the Pima Indian Diabetes Database, then prepared, cleaned, and pre-processed for consistency and accuracy. Missing values were replaced with means, and linear and stratified sampling techniques were employed to tackle data imbalance.	The stacked ensemble model outperformed other algorithms and also had an accuracy of 94.17%.	The consideration of fewer characteristics affected overall performance.
[39]	2022	KNN, DT, RF, Adaboost and LR.	Data was extracted from a clinical dataset in Bandipora, India, containing 768 patient records with variables like Pregnancy, Glucose, Blood Pressure, Insulin, BMI, Age, and Outcome. The data was pre-processed through cleaning, normalization, and analysis to identify patterns and ensure accuracy.	Random forest algorithm had the best accuracy.	More algorithms could have been used comparatively.

	Year of Publication	Methodology	Process	Result	GAPS
[40]	2018	ANN, SVM, LR, DT, RF and NB.	Data was extracted from the Pima Indian Diabetes Database, consisting of 768 patient records with variables like pregnancy, BMI, insulin level, age, glucose concentration, blood pressure, skin thickness, and diabetes pedigree function. The data was cleaned, pre-processed for consistency and accuracy, and missing values were replaced with means. Eight parameters were used: pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age.	The machine learning classifiers had an accuracy of nearly 75%.	Parameter optimization was not carried out for better accuracy.

2.7 Chapter Summary

The literature review examines the application of supervised machine learning in diabetes detection, exploring diverse methodologies and approaches for identifying and interpreting diabetes-related patterns. It emphasizes the crucial role of diabetes detection in healthcare technology, aiding individuals at risk and managing the condition. The analysis evaluates machine learning algorithms, considering their advantages and limitations for diabetes detection, which is vital for early intervention and effective management. With the global prevalence of diabetes expected to rise significantly, there's a pressing need for accurate detection methods. This study aims to address the diabetes epidemic by developing and evaluating a comprehensive system using supervised machine learning techniques for early detection and management. The proposed system's design and methodology will be detailed in subsequent chapters, offering insights into leveraging machine learning for diabetes detection and management.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

The primary objective of this project is to develop an early prediction tool employing machine learning techniques to ascertain the presence or absence of diabetes in individuals. To achieve this, models are constructed using scikit-learn, leveraging their effectiveness in binary classification tasks. The training dataset utilized in this project comprises a combination of continuous and categorical data from relevant sources, such as the Pima Indian Diabetes Database sourced from Kaggle. Logistic regression emerges as the optimal choice among the algorithms evaluated for accuracy, surpassing Naive Bayes, k-Nearest Neighbors (KNN), and Decision Tree Classifier. This algorithm exhibits superior performance in discerning patterns within the data, particularly vital for early detection of diabetes and minimizing false positives or negatives. The project's findings and insights are encapsulated in Chapter 4, shedding light on the efficacy of logistic regression in diabetes prediction.

3.2 Design specification

Each stage is explained as follows:

I. Exploratory Data Analysis

The very first step that is carried out in this project is the process of importing all the necessary python modules. Python, in itself is just a programming language which gives just the basic functionalities of any average programming language. For machine learning, it is essential that the python modules such as, pandas, numpy, sklearn are utilized so as to aid in importing data, cleaning data, formatting data, and aid in building, evaluating and drawing up the models for Logistic regression, Naive Bayes, k-Nearest Neighbors (KNN), and Decision Tree Classifier. Exploratory data analysis is a method to examine data sets so that its main characteristics, usually using statistical graphics and other visualization approaches, may be summarized.

EDA may be used either with or without a statistical model, but most of the data can be utilized to examine the data without formal modeling or testing of hypotheses.

Preparation of data/ Loading data from the data file

This dataset is imported from the Kaggle data set library (Pima Indian Diabetes Dataset). During the data preparation phase, a number of processes are implemented. These processes also include the correction of missing information. The biggest part of any data analysis project is making sure that the data is correctly formatted and fixing it when it is not. The next step after importing the data is to deal with all missing data. Missing data is simply a blank space or surrogate value that indicates that a data feature was not obtained. For example, an empty "age" cell means we forgot to document an "age". Hence, we would have an empty cell. For this project there are no empty cells within the column function. The columns descriptions are as follows:

a) Age: The age is calculated every 12 months (in years).
b) BMI: BMI (Body Mass Index) is a fundamental measure used to assess body composition, calculated by dividing weight (in kilograms) by the square of height (in meters). It serves as a key indicator of metabolic health and is closely linked to the risk of developing conditions like type 2 diabetes.
c) Blood Pressure: Blood pressure, measured in mmHg, indicates the force of blood against vessel walls. Essential for cardiovascular health, it includes systolic and diastolic readings. High blood pressure (hypertension) is linked to diabetes risk, necessitating monitoring.
d) Pregnancies: Pregnancy history, particularly the number of pregnancies (parity), is a crucial factor in diabetes prediction. Higher parity is linked to increased risk of gestational diabetes and type 2 diabetes.
e) Diabetes Pedigree Function: The diabetes pedigree function (DPF) evaluates diabetes risk based on family history. A higher score indicates greater genetic predisposition.
f) Insulin: Body insulin levels are pivotal in predicting diabetes risk, reflecting metabolic function. Low levels or insulin resistance increase the likelihood of type 2 diabetes.
g) Skin Thickness: Skin thickness is a crucial factor indicating metabolic health as thicker skin may signal insulin resistance and higher body fat levels.

h) Glucose: Body glucose level is a key indicator having elevated levels suggest impaired glucose tolerance, increasing the likelihood of type 2 diabetes.
i) Outcome: This refers to whether or not you have diabetes where 1 means positive; 0 means negative.

From the above analysis of each column, we can see that they are almost all ‘int64’, however, two columns (BMI, Diabetes Pedigree Function) have “float64” data type. This is important for further analysis of the dataset.

II. Feature Engineering

The process of transforming unstructured data into a more understandable format is known as feature engineering. In this instance, it is important to make a machine learning model give accurate predictions. To make this possible, certain feature may be extracted from the dataset to make the system more accurate. Also, a number of features can be modified to enable the data used to be more understandable for people without expertise in data-analysis. It involves crafting and selecting features that capture relevant information and patterns in the data, ultimately enhancing the performance and interpretability of the models. In this diabetes project, it was opted not to encode the features using techniques such as one-hot encoding. This decision was driven by the inherent characteristics of the dataset. The features, including 'Pregnancies,' 'Glucose,' 'BloodPressure,' 'SkinThickness,' 'Insulin,' 'BMI,' 'DiabetesPedigreeFunction,' and 'Age,' are predominantly numerical and continuous in nature.

These features represent physiological measurements and demographic information that already exist on a continuous scale, making them readily interpretable by machine learning algorithms without the need for encoding. Unlike categorical variables, which consist of distinct categories or groups, the features in this project represent quantitative measurements that can take on any real value within a given range. Therefore, encoding techniques such as one-hot encoding, which are typically used to convert categorical data into a numerical format, were deemed unnecessary for the dataset. By leveraging the inherent numerical nature of the features, the data preprocessing pipeline was streamlined and focus was shifted towards other aspects of model development and evaluation.

III. Data Visualization

Data visualization reveals the need to present the numerical and nominal data in a visual context such as a map or a graph, to provide understanding. The primary aim of data visualization in big data sets is to facilitate the identification of patterns, trends and outliers. Below is a number of generated visuals of the data with the aid of pandas.

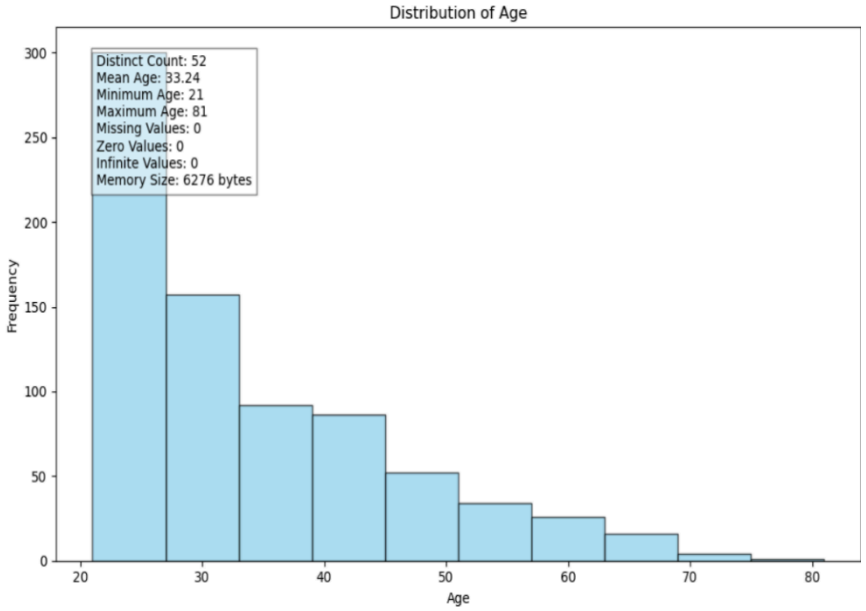

Figure 3.1: Pandas Data visualization for Age

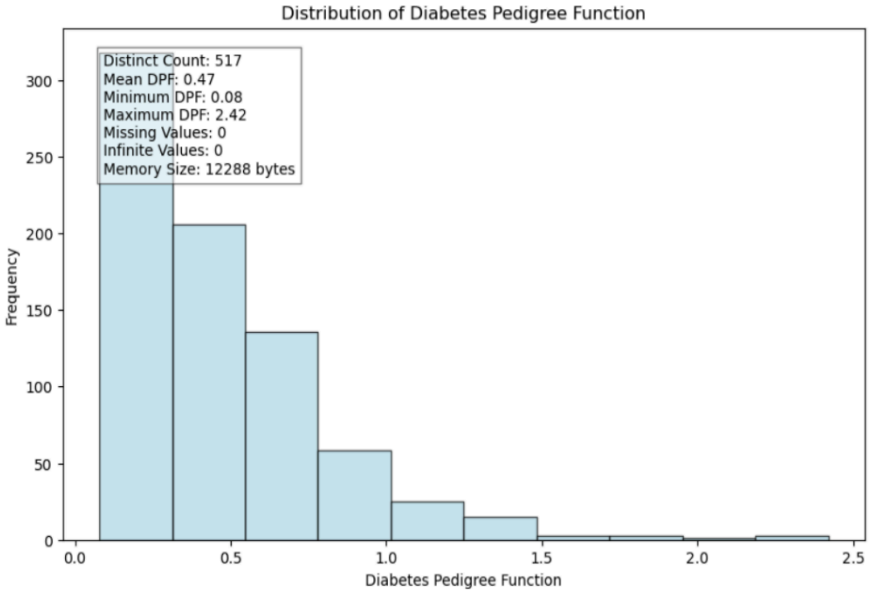

Figure 3.2: Pandas Data visualization for DPF

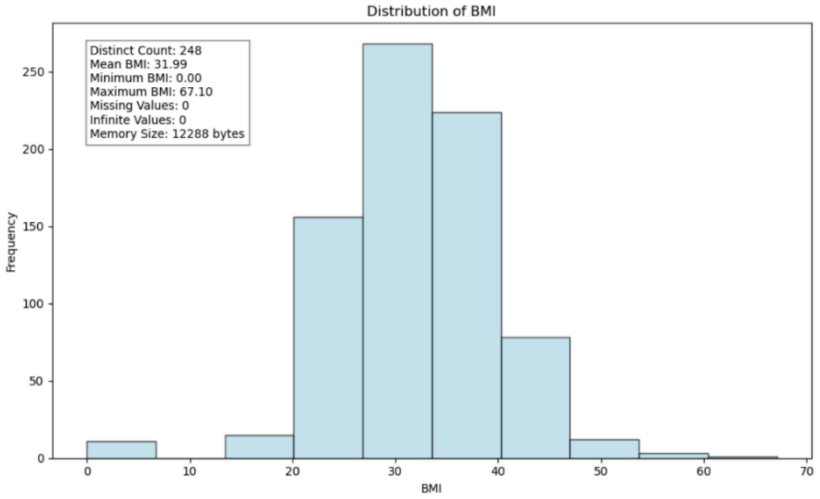

Figure 3.3: Pandas Data visualization for BMI

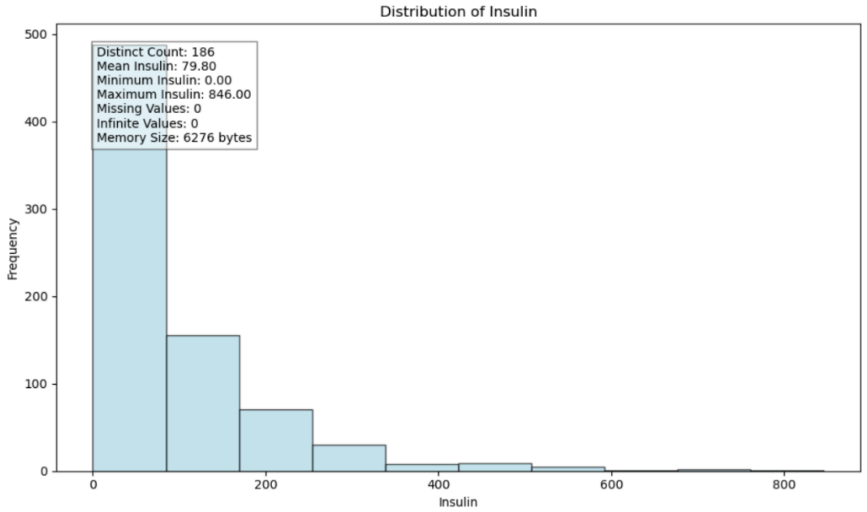

Figure 3.4: Pandas Data visualization for Insulin

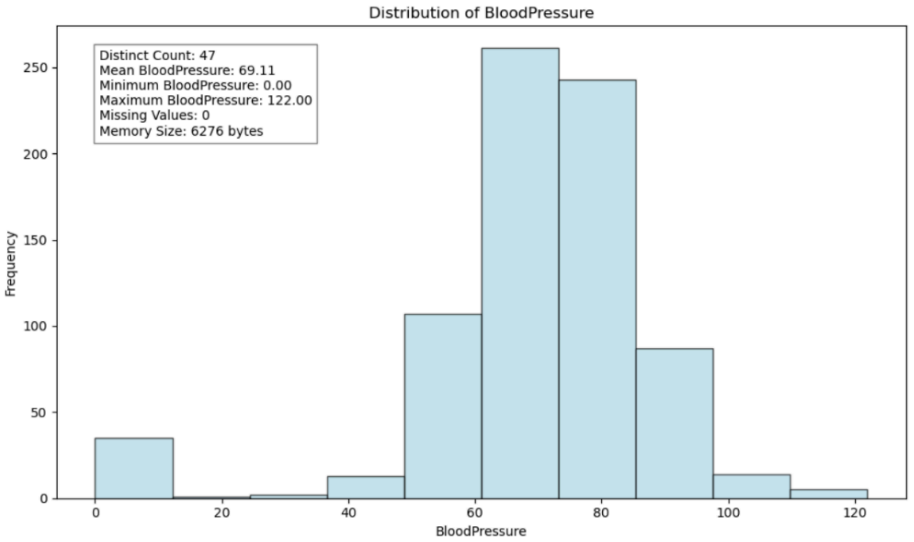

Figure 3.5: Pandas Data visualization for BloodPressure

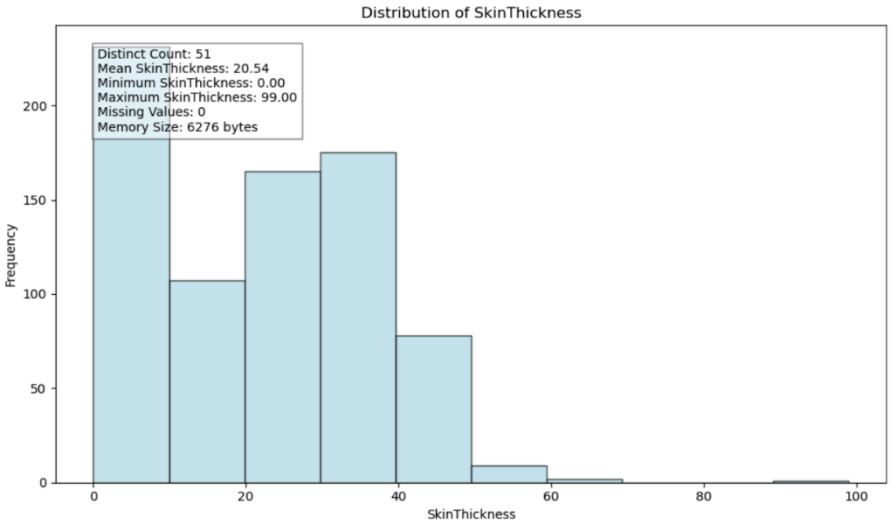

Figure 3.6: Pandas Data visualization for SkinThickness

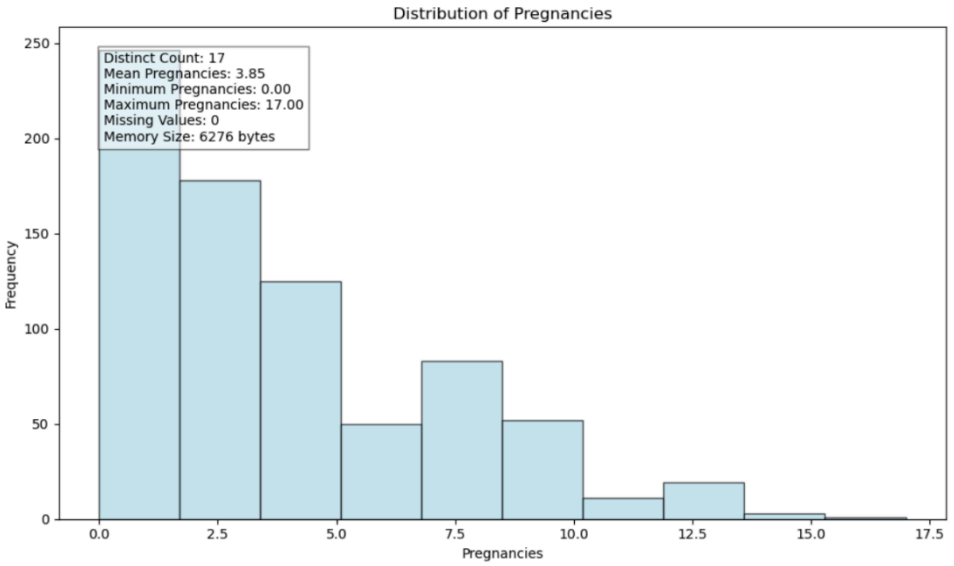

Figure 3.7: Pandas Data visualization for Pregnancies

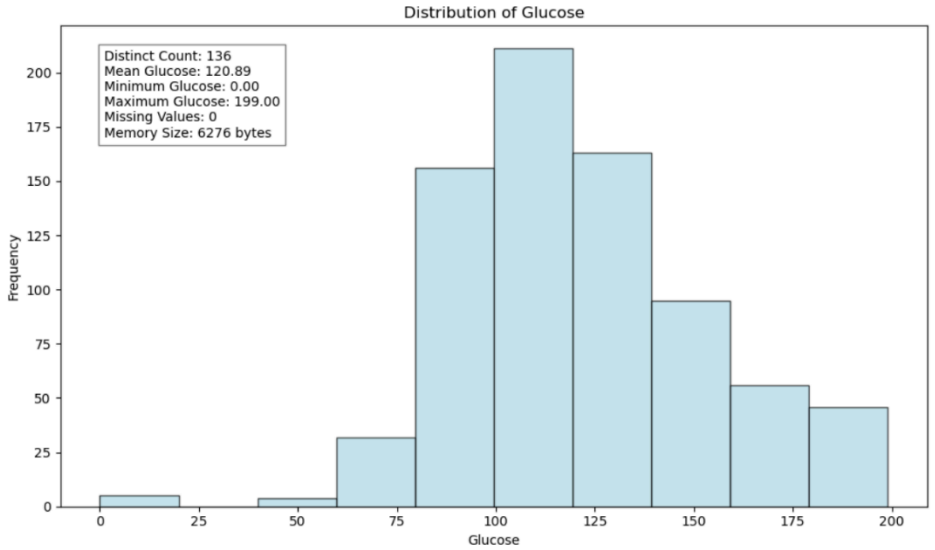

Figure 3.8: Pandas Data visualization for Glucose

IV. Classifier Parameter Optimization

A model parameter is an internal model configuration variable whose value may be inferred from the supplied data. They are the parts of the model that have been learnt from previous training data. The Classifier parameters are optimized in this phase using the dataset. The table (3.1) below displays the columns in the dataset used.

Table 3.1: Table showing dataset used

S/N	Attributes	Description	Values
1	Age	This is to display the age of the individual which is usually in years.	Continuous
2	BMI	This refers to the Body Mass Index of the patient.	Continuous value in kg/m^2
3	BloodPressure	This refers to the blood pressure of the patient.	Continuous value in mmHg
4	SkinThickness	This refers to the skin thickness of the patient.	Continuous value in µm
5	Glucose	This refers to the glucose level of the patient.	Continuous value in mg/dL
6	Pregnancies	This refers to the number of pregnancies the patient has	Continuous

		had which is 0 for males and can also be 0 for females.	
7	Insulin	This refers to the insulin level of the patient.	Continuous value in µIU/mL
8	DiabetesPedigreeFunction	This refers to the diabetes pedigree function score of the patient.	Continuous
9	Outcome	This refers to the likelihood of having diabetes.	1 means positive 0 means negative

After the data is imported, the format of the data is checked using pandas (pd). Data used in data analysis ought to be in a .csv format, otherwise known as a CSV file. A Comma Separated Values file (CSV) is a plain text file containing a list of data separated by commas. Often, these files are used for data sharing between various applications. Databases and contact managers, for example, often support the CSV files. A CSV is a text file and can be created and edited using any editor of the text. When pandas (pd) read imported data, it returns data frame, which is a lot like a spreadsheet. The data are organized in rows and columns and each row can contain a mixture of text and numbers. The standard variable name for a data frame is the initials df, and that is what is used here:

```
[2]: df= pandas.read_csv('C:\\Users\\USER\\Downloads\\diabetes dataset.csv')
     df.head()
```

[2]:

	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

Figure 3.9: Code snippet for Pima Indian Diabetes Data set

Ideally, the data is pre-processed before prediction tests, and incomplete data are replaced by values created using attribute mean. The column names are also changed to make it easier to know how to format the data. The data is filtered using a mean approximation of the function which substitutes all missing values with a value derived from the mean of the column values using the mean principle in statistics. This is a very crucial step, especially for the real data set that is vulnerable to missing values.

V. Model Selection

In this project, several machine learning algorithms have been implemented, including logistic regression, naive Bayes, decision tree classifier, and k-nearest neighbors (KNN), using the scikit-learn library in Python. These algorithms are utilized to predict the likelihood of an individual developing diabetes based on various features such as glucose levels, BMI, age, and others. Logistic regression excels in estimating the probability of binary outcomes, making it well-suited for predicting diabetes status based on patient attributes. The objective of logistic regression is to estimate the probability that a query sample belongs to a particular class based on the values of its input features. It achieves this by fitting a logistic curve to the input data, which models the relationship between the independent variables and the binary outcome variable. Naive Bayes, with its probabilistic framework and assumption of feature independence, efficiently handles high-dimensional datasets, providing fast and reliable predictions. Decision tree classifiers, on the other hand, offer intuitive insights into the relationships between input features and the target variable, facilitating interpretable models. Lastly, KNN leverages the similarity between data points in the feature space to classify individuals, making it particularly useful for capturing non-linear relationships in diabetes prediction. The data set for this model is externalized from Kaggle Machine Learning library to determine whether a patient has a diabetes or not.

For this model using logistic regression as the primary, the following steps were taken

i. Importing the dataset for exploratory data analysis: During this phase, we meticulously identify missing data and implement strategies to handle it effectively, ensuring it does not adversely impact our machine learning

models. Missing data is imputed using techniques such as mean, median, or mode imputation, where missing values in a given column are replaced with the corresponding summary statistic. However, in the dataset used there were no missing values and as such the techniques to mitigate missing data were not used.

ii. Formatting the data for Logistic Regression model: To prepare the data for logistic regression and other supervised learning algorithms, we split the dataset into dependent and independent variables. In our case, the target variable (dependent variable) is whether an individual has diabetes or not, while the independent variables consist of various features such as glucose levels, BMI, age, and others. These features are crucial for predicting the likelihood of diabetes and are used to train the logistic regression model. After this, the data is centered and scaled. The average of a variable is removed from the data when it is centered. When data is scaled, the standard deviation of a variable is subtracted from the total.

iii. The Logistics Regression model is trained.

iv. The next phase entails building, evaluating and interpreting the final Logistic Regression model and also comparing its accuracy with that of the other models trained as well.

VI. Web Interface Implementation

The Web application serves as a bridge for connecting the client to the predictive model. The web application serves as a user interface where clients provide Model data. The web interface implementation for this application was built using Django, and a simple form layout was created using HTML and CSS to enhance the structure of the webpage.

3.2.1 Performance Matrix

The following key words are commonly used in the equations for these evaluation metrics:

i. True Positives (TP) refers to the instances where a model accurately predicts a positive class, indicating that the model correctly classifies an instance as belonging to the positive class based on the ground truth or actual labels.

ii. True Negatives (TN) refers to instances where a model accurately predicts a negative class, indicating that the instance is indeed in the negative class according to the ground truth or actual labels.
iii. False Positives (FP) refers to instances where a model incorrectly identifies negative instances as positive.
iv. False Negatives (FN) refers to instances where the model predicts a negative outcome despite the outcome being positive.

Accuracy is a fundamental metric for measuring performance and it represents the ratio of correctly predicted results to the total results. The mathematical expression is given in equation (3.1):

$$Accuracy = \frac{number\ of\ correct\ precictions}{total\ number\ of\ predictions} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.1)$$

The degree of precision is determined by the ratio of true positive predictions to the total number of positive predictions. This is mathematically represented in the following equation (3.2):

$$Precision = \frac{TP}{TP+FP} \quad (3.2)$$

Recall is the ratio of true positive predictions to the total number of actual positive observations. This evaluates the model's capacity to accurately detect all instances of positive cases, including the ones that are missed by the model. The mathematical expression is given in equation (3.3):

$$Recall = \frac{TP}{TP+FN} \quad (3.3)$$

The F1 score considers both false positives and false negatives while evaluating a model's performance based on precision and recall. It is particularly useful when the costs associated with false positives and false negatives are similar. In such cases, the F1 score offers a fair assessment of the model's accuracy and recall. The mathematical expression is given in equation (3.4):

$$F1\ score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (3.4)$$

3.3 System Requirements

The hardware, software, functional and non-functional requirements needed by the system for maximum and efficient performance are listed below:

3.3.1 Hardware Requirements

This basically refers to the physical resources that a system requires to run the web application. It is the most common set of requirements that is defined by the operating system. The table below shows the minimum hardware requirements and recommendations for web applications:

Table 3.2: Hardware requirements of this system

Component	**Minimum**	**Recommended**
Processor	Instruction sets of 1.9 Gigahertz (GHz) x86 – or x68-bit twin core CPU	The twin core processor with SSA2 instruction set is 3.3 Gigahertz (GHz) or faster.
Memory	2-GB RAM	4-GB RAM or more
Display	Super VGA at 1024 x 768 resolution	Super VGA at 1024 x 768 resolution

3.3.2 Software Requirements

Software requirements refer to the specification of the programs that are used on a computer system and are required for the development process. For this project, the software requirements are shown in Table 3.3 below.

Table 3.3: Software requirements of this system

REQUIREMENTS	**SOFTWARE**
Operating System	Windows 10
Supported web browsers	Microsoft Edge, Mozilla Firefox, Google chrome, Apple Safari. All applications should be running on windows 7, Safari should be working on Mac OS or versions with the Apple iPad.
Programming Language Used	Python 3.11.5
Development Environment	Anaconda Python – Used to get ML libraries Django – for Front-End
Text Editor	Visual Studio Code
Libraries	Django==5.0.3 NumPy==1.26.4 scikit-learn==1.4.1 matplotlib==3.8.3 pandas==2.2.1

3.3.3 Functional Requirements

The functional requirements delineate the precise features and functionalities essential for the early diabetes detection system's operation. These requirements encapsulate the system's foundational abilities and delineate its expected capabilities.

a) Data Input:
   i. The system should allow users to input relevant health data necessary for diabetes risk assessment, such as age, BMI, pregnancies, blood pressure and other factors.
   ii. Input fields should be clearly labeled and organized to facilitate user input and understanding.

b) Diagnosis Output:
   i. The diagnosis output should be presented in a user-friendly format, such as a visual indicator or a straightforward textual message.

c) Supervised Machine Learning Algorithm:
    i. The system should integrate a supervised machine learning algorithm trained on a comprehensive dataset of diabetes-related parameters and outcomes.
    ii. The algorithm should analyze the input data to generate predictions regarding the likelihood of diabetes development.
d) User Experience:
    i. The system interface should be intuitive and user-friendly, guiding users through the data input process seamlessly.

These functional requirements ensure that the system delivers a comprehensive, personalized, and effective early diabetes detection experience, equipped with robust tools for accurate risk assessment and timely intervention.

3.3.4 Non-functional Requirements

The non-functional requirements delineate the quality and performance characteristics expected of the early diabetes detection system. The system must adhere to the following attributes:

a) Usability: The system interface should be intuitive and user-friendly, facilitating easy input of specific health data for diagnosis without the need for user authentication.
b) Scalability: The system should efficiently handle varying volumes of user inputs and data without compromising the diagnosis process's speed or accuracy.
c) Reliability/Availability: The system should maintain high availability, ensuring that users can access the diagnosis tool whenever needed without downtime or service interruptions.
d) Modularity: Designing the system with modular components facilitates scalability and future enhancements. Separating the diagnosis algorithm, user interface, and data storage components supports ease of maintenance and potential integration with other systems.

3.4 General System Architecture

The system will provide a web interface where doctors can input the patients' medical data in order to predict a patient's risk of having diabetes with the use of a back-end machine learning model trained via the Python's Sci-kit learn library. The Logistics Regression model is built and implemented to predict the risk of developing diabetes based on patients' medical records. This system consists of the machine learning model which is linked to the Web Application developed using Django. Django is a WSGI (Web Server Gateway Interface) web application framework. The Web Server Gateway Interface, or WSGI, is a connection between a web server and a Python-based web application and it is necessary since a web server can't communicate directly with a Python program. The machine learning model architecture consists of data sets that have been compiled and cleaned from Kaggle.

3.4.1 Machine Learning Model Architecture

Logistic Regression stands as a formidable tool in classification tasks and it will be used alongside other algorithms in this machine learning model architecture. In the context of predicting heart disease, classification algorithms like logistic regression facilitate the categorization of data into binary outcomes: 0 for the absence of heart disease and 1 for its presence. The architecture of the machine learning model is illustrated below through a flow chart, depicting the development process of the model.

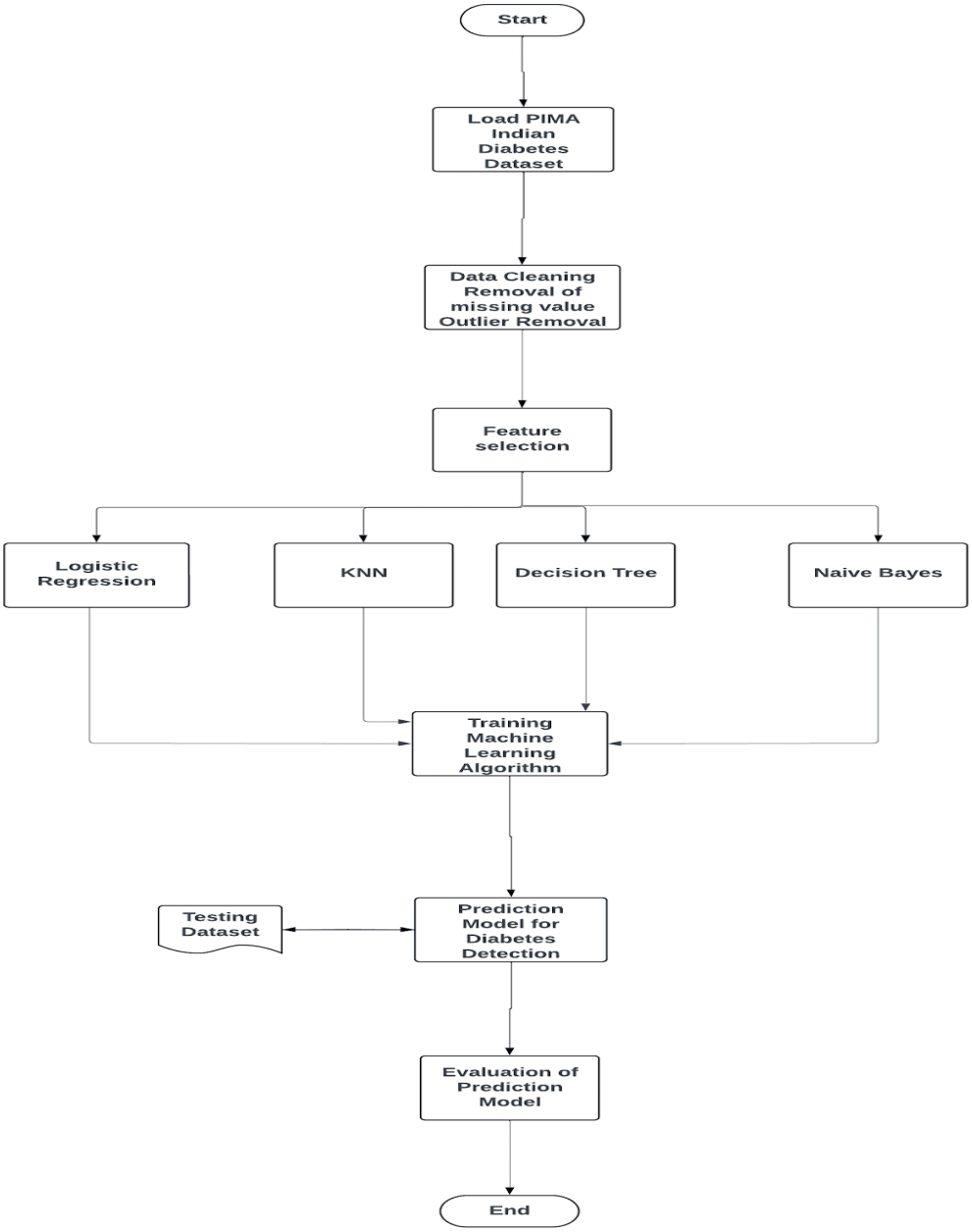

Figure 3.10: Flowchart showing the Model Architecture

This diagram illustrates the steps involved in building a predictive model for detecting diabetes. Initially, raw data, structured or unstructured, is gathered. Subsequently, data cleaning is crucial to eliminate missing values and outliers. Feature selection follows, picking the most relevant features from the dataset. Then, a machine learning model is trained using a supervised learning classification algorithm. Typically, an 80-20 split is employed for training and testing data, respectively. The trained model is then tested using the test data. The model predicts

outcomes based on past experiences stored in its knowledge base. Performance evaluation of different classifiers is conducted using various parameters, with accuracy being the primary measure for selecting the best classifier. However, the F-score is also significant, particularly considering the imbalanced nature of the diabetic dataset.

3.5 Training the Machine Learning Model

To train the machine learning model for diabetes prediction, Logistic Regression is employed as the primary classification algorithm although the naïve bayes, KNN and decision tree classifier algorithms are also trained. Logistic Regression is a binary classification algorithm that models the probability of a binary outcome based on one or more predictor variables. It finds the best-fitting logistic function to predict the probability of the occurrence of an event (e.g., diabetes) by minimizing the logistic loss function. In order to train the model, the data was correctly formatted and after which the model is built. The operational flowchart of the logistic regression model is shown below:

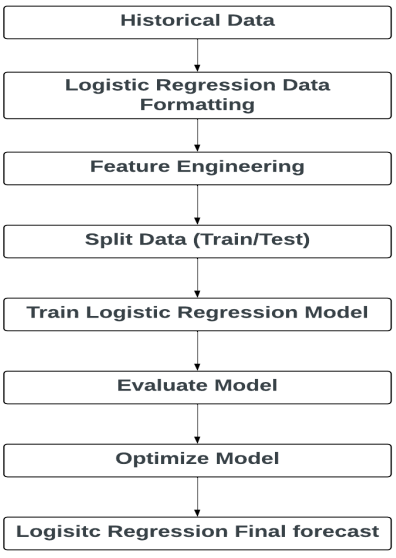

Figure 3.11: Logistic Regression Flow Chart

After building the logistic regression model, the classification report is generated, and the confusion matrix is also generated. In this confusion matrix, it shows that from 128 individuals that did not have diabetes, 85 (80%) were correctly classified as non-diabetic and of the individuals that have diabetes 43 were correctly classified as diabetic. So, the logistic regression model shows promising results, particularly in

identifying non-diabetic individuals. However, it demonstrates a lower performance in correctly identifying diabetic individuals. This suggests that further optimization may be beneficial to enhance the model's predictive accuracy. Below shows the code snippets for the confusion matrix and the classification report

```
[15]: class_report = classification_report(y_test, yp)

      # Print classification report
      print("Classification Report:")
      print(class_report)

      Classification Report:
                    precision    recall  f1-score   support

                 0       0.80      0.92      0.85        85
                 1       0.77      0.53      0.63        43

          accuracy                           0.79       128
         macro avg       0.78      0.73      0.74       128
      weighted avg       0.79      0.79      0.78       128
```

Figure 3.12: Code snippet for classification report for Logistic Regression

```
conf_matrix = confusion_matrix(y_test, yp)

#print(loss)
#print("Accuracy:",metrics.accuracy_score(y_test, yp))
print("Confusion Matrix:")
print(conf_matrix)

Confusion Matrix:
[[78  7]
 [20 23]]
```

Figure 3.13: Code snippet for the Confusion Matrix

3.6 Web Interface Implementation

The Web application serves as a user-friendly interface for connecting the client to the predictive model. The web application serves as a user interface where clients interact with the web application by filling a webform which serves as an input source to interact with the model in the backend. The users are required to input the required data, this data is then sent to be pre-processed in the backend and finally give the results are generated. So, in order to display this, the user interface is designed using HTML and CSS and using the POST request the input data is passed to the server which does the remaining work. The sequence of steps taken by the program in generating the predictions are highlighted below:

i. To begin, the user fills in the necessary information on the form (predict.html).

ii. These inputs are now delivered to the result route through a POST request when the user hits the result button.

iii. Now, as seen in the accompanying code, these inputs are obtained from val1 to val8.

iv. These inputs are now given into our previously trained model, which we have loaded, and it simply predicts if a patient has diabetes or not.

v. By rendering result.html with predicted class, the result is presented in the form of the predicted class using render template.

```
def predict(request):
    return render(request, 'predict.html')
def result(request):
    data = pd.read_csv('C:\\Users\\USER\\Downloads\\diabetes dataset.csv')

    x= data.drop('Outcome',axis=1)
    y= data['Outcome']
    #split your data into training and test data
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,shuffle=False)

    model = linear_model.LogisticRegression()
    model.fit(x_train,y_train)
    val1= float(request.GET['n1'])
    val12= float(request.GET['n2'])
    val3= float(request.GET['n3'])
    val4= float(request.GET['n4'])
    val5= float(request.GET['n5'])
    val6= float(request.GET['n6'])
    val7= float(request.GET['n7'])
    val8= float(request.GET['n8'])

    pred=model.predict([[val1, val12, val3, val4, val5,val6,val7,val8 ]])

    result1 =""
    if pred==[1]:
        result1= "positive"
    else:
        result1= "negative"
```

Figure 3.14: Code snippet for Django web interaction

So, a request is sent from the browser to the web server (Apache) to request a specific URL. The web server then routes this request to a WSGI server for processing.

The subsequent figure 3.15 below illustrates how a user request is carried out

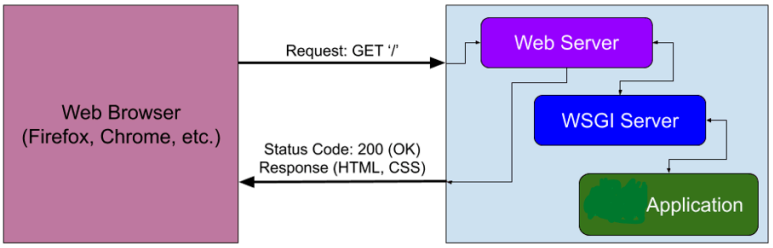

Figure 3.15: Model, View, Template Architecture

The system is designed for use by professional medical experts due to the nature of data inputs used by the Model. The system acts as a computer-aided diagnostic tool to help medical doctors and experts provide a better diagnosis to their patients.

3.7 System Modelling

Generating abstract representations of a system using techniques like UML to capture its essential elements, relationships, and behaviors. This process helps engineers understand the system's structure and functionality, focusing on important aspects while hiding unnecessary details. UML diagrams, such as use case and class diagrams, facilitate effective requirement analysis, design visualization, and system validation, providing a common language for communication and analysis.

3.7.1 Use Case for the system

A use case is a diagram used in system engineering to show interactions between actors (users or administrators) and a system. It illustrates how users accomplish tasks, such as adding items or searching for information, by interacting with the system to achieve goals. Although often associated with software systems, use cases can apply to any process. Each use case represents a unique task involving external system participation, with actors being either people or other systems.

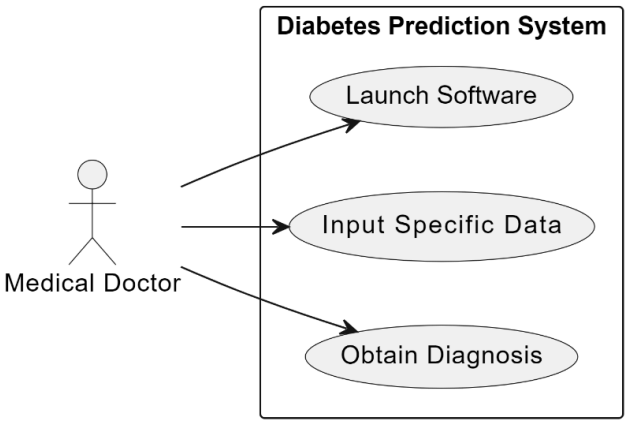

Figure 3.16: Use-case diagram for the system

3.7.2 Activity Diagram

An activity diagram, a type of UML diagram, visually represents the workflow or behavior of a system or process. It shows various activities, actions, decisions, and their interconnections, modeling the dynamic aspects of a system. Activity diagrams illustrate the sequence of activities, their conditions, and control flow, aiding in understanding business processes, system behavior, and interactions. They are valuable for software engineers in analyzing, designing, and documenting complex systems, ensuring functionality and efficiency. Common shapes include rounded rectangles for actions, diamonds for decisions, and black circles for workflow starts.

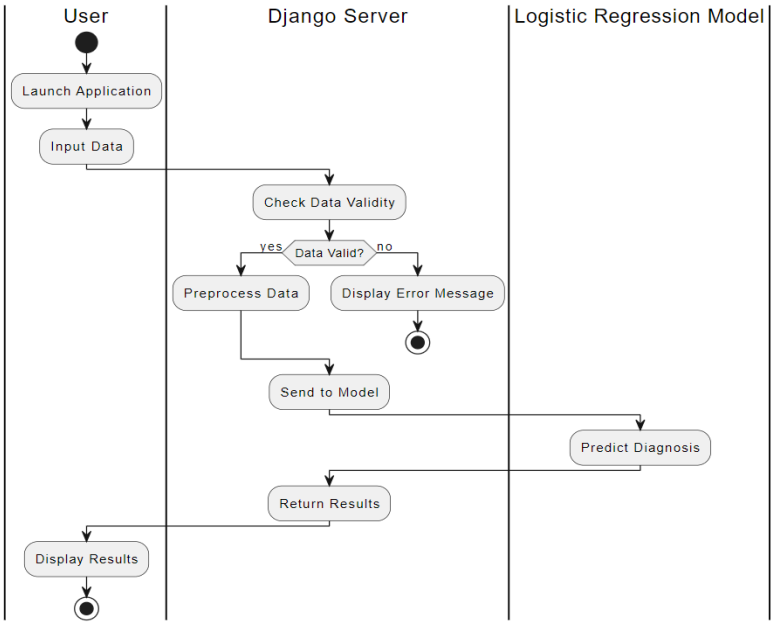

Figure 3.17: Activity diagram of the system

3.7.3 Sequence Diagram

A sequence diagram is a UML diagram that shows the sequence of interactions between objects in a system over time. It details the order of messages exchanged to perform a specific task, emphasizing the time order of interactions. Sequence diagrams help model dynamic system behavior and highlight control and data flow. They are useful for understanding, designing, and documenting complex interactions, ensuring clarity in communication. Key elements include lifelines for objects, arrows for messages, and activation bars for activity duration.

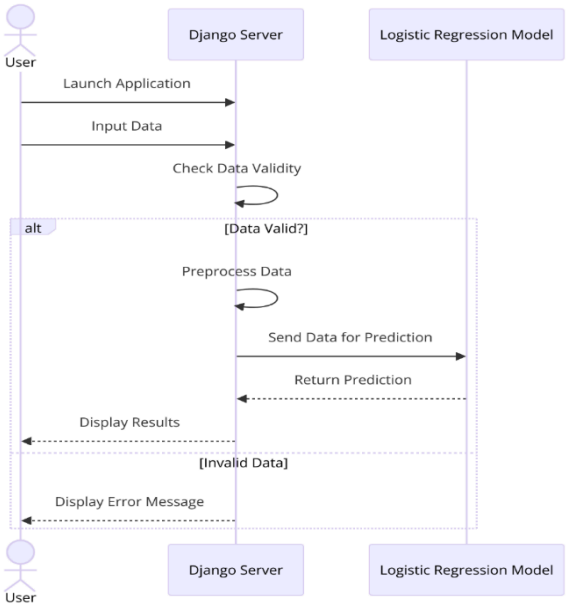

Figure 3.18: Sequence diagram of the proposed system

3.8 Chapter Summary

In summary, this chapter delineated the development of an early prediction tool for diabetes, centered on logistic regression. The chapter outlined four key phases: data collection, exploratory analysis, algorithm selection, and model evaluation as well as deployment. It elucidated the significance of logistic regression in discerning patterns for diabetes prediction, based on its superior performance over other algorithms. The design specifications of the system, performance evaluation metrics and architecture of the system were also discussed. Overall, this chapter provides a clear understanding of the system's design and implementation and serves as a foundation for the subsequent chapter, which will evaluate the system's performance and discuss the results.

CHAPTER FOUR: RESULTS AND DISCUSSION

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents and discusses the results of the project on early detection of diabetes using supervised learning algorithms, including logistic regression, decision trees, KNN, and Naive Bayes. The performance and results of these models are detailed, relating the findings to the project's aims, objectives, and existing literature.

4.2 System Implementation

The Diabetes Risk Prediction system utilizes logistic regression, which emerged as the most accurate model, and is built using scikit-learn. This model interacts with a web-based API to determine if an individual is at risk of developing diabetes. Although logistic regression is the deployed model, a comprehensive comparison was conducted with decision trees, Naive Bayes, and KNN to ensure robustness and reliability. The section below describes the model training process.

4.2.1 Model Training

Model training was carried out using Jupyter Notebooks within the Python Anaconda Suite. After importing and preprocessing the data, which included handling any missing values, the data was formatted for the machine learning algorithms. The dataset was split into dependent and independent variables, and since the features were numerical, One-Hot Encoding was not necessary. When this is done, the data is centered and scaled and the logistic regression, naïve bayes, decision tree classifier and KNN models are built.

4.2.2 Model Building

The first step entails importing all the necessary python libraries such as Pandas, NumPy (for reading the data), seaborn, matplotlib, scikit-learn, DTC, LR, GaussianNB, KMeans, Confusion Matrix, Classification report (for ML model building; pre-processing and evaluation) after which exploratory data analysis (EDA) is carried out.

EDA in this project includes the following:

i. **Finding Unwanted Columns**

The dataset is first examined to check for unwanted columns that would be irrelevant to the purpose of the data analysis. After examination, it is evident that there is no unwanted column present in the dataset to remove.

ii. **Find Missing Values**

A variety of methods are implemented to check missing values. Missing values refer to empty cells that can affect the accuracy of the data, and can also cause data leakages, this is cared for using the following methods;

1. df.isnull().sum(): It returns null values for each column.
2. isnull().any(): It returns True if column have NULL Values and false if it doesn't. However, from the two methods used above it is understood that there are no missing values.

iii. **Finding features with one value**

It is important to rid out columns with only one feature because they will be irrelevant to the dataset. There could be chance of only one category in a particular feature. In Categorical features, suppose gender column we have only one value i.e male. Then there is no use of that feature in the dataset. The output is as listed below; Pregnancies 14, Glucose 132, BloodPressure 39, SkinThickness 48, Insulin 148, BMI 227, DiabetesPedigreeFunction 442, Age 46, Outcome 2.

From the output above it is obvious that there is no column with only one feature.

iv. **Explore the Numerical Features**

All the data in the dataset are numerical features totaling 639 rows and 9 columns (769 X 9).

v. **Finding Discrete Numerical Features**

There is only one discrete numerical feature as observed from the dataset.

vi. **Finding Continuous Numerical Features**

The Continuous numerical features as observed from the dataset total a number of eight.

vii. Distribution of continuous Numerical Features

From the exploratory data analysis, it is evident that all continuous features are not normally distributed.

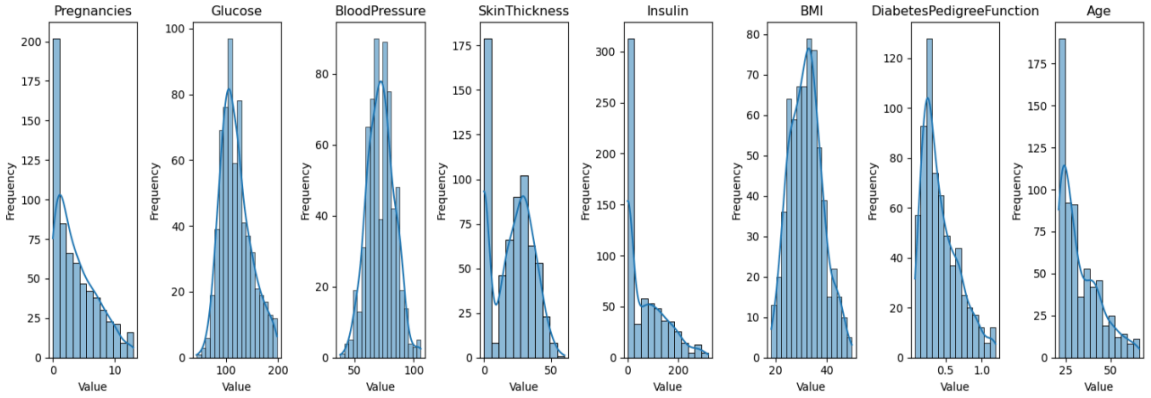

Figure 4.1: Univariate distribution of Continuous Observations

viii. Relation between Continuous numerical Features and Outcome

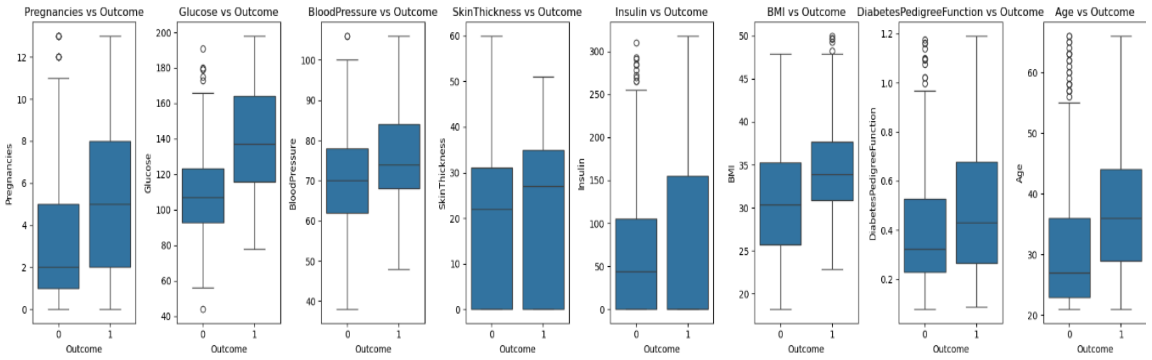

Figure 4.2 : Relation between Continuous numerical Features and Outcome

ix. Finding Outliers in numerical features

An outlier is a value in a data series that is either incredibly small or very big, and therefore may influence the overall conclusion drawn from the data series. Because they occur at the extremities of a data series, outliers are also known as extremes. Outliers are abnormal numbers that may have an impact on the entire observation owing to their very high or low extreme values, and therefore should

be removed from the data. Below the Box plot is used to detect the outliers in the data set.

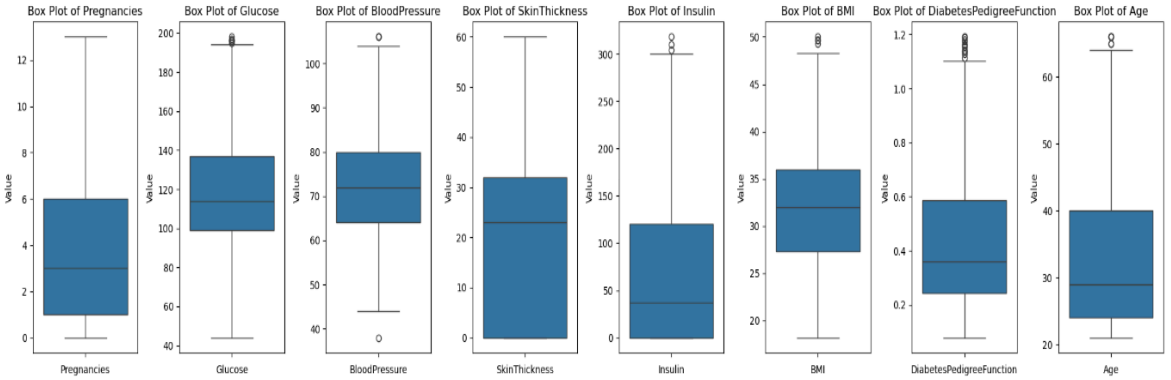

Figure 4.3: Graphical representation (Box Plots) showing Outliers

x. Explore the Correlation between numerical features

Correlation is a measure of how two variables vary over time. A correlation matrix may be shown to indicate whether variable has a high or low correlation with another one.

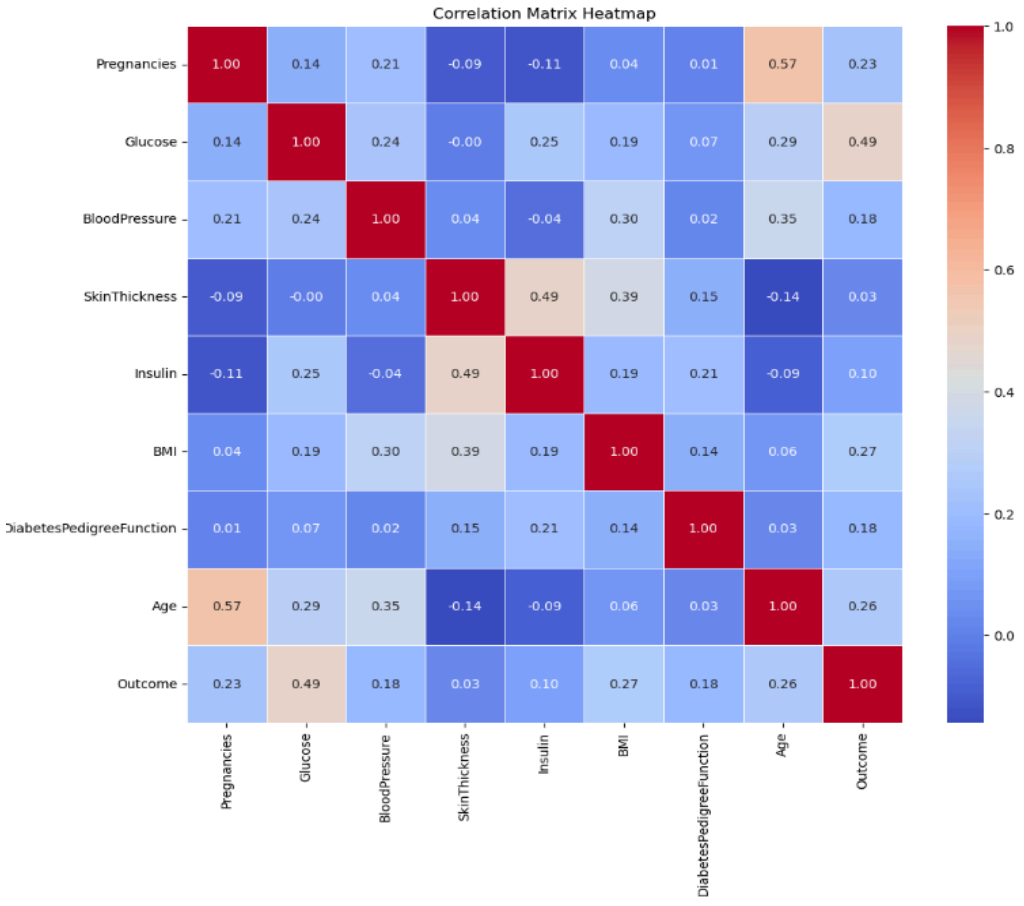

Figure 4.4: Graphical representation of the correlation matrix

After all these necessary steps are carried out, the data is formatted for making the model. The first step includes splitting the data into two parts:

a) The columns of data that will be used to make classifications.
b) The columns of data to be predicted.

The conventional notation of `X` (capital X) is used to represent the columns of data that will be used to make classifications and `y` (lower case y) to represent the intended predictions. In this case, to predict diabetes. Once the data frame is split into two pieces; `X`, which contains the data we will use to make, or predict, classifications, and `y`, which contains the known classifications in our training dataset, A closer look at the variables in `X` is required. The following list informs us what each variable is and what kind of data (float or category) it should contain:

i. Age, Float.
ii. BMI: This is used to denote the body mass index measured in value in kg/m^2, Float.
iii. BloodPressure: This is used to denote the blood pressure measured in mmHg, Float.
iv. SkinThickness: This is used to denote the skin thickness measured in μm, Float.
v. Insulin: This is used to denote the insulin level measured in μIU/mL, Float
vi. DiabetesPedigreeFunction, Float.
vii. Pregnancies: This is used to highlight the number of pregnancies if any, Float.
viii. Glucose: This is used to denote the glucose level measured in mg/dL, Float.

After the data is correctly formatted for training the logistic regression model which is used as the primary model, the next step involves centering and scaling the data. Since logistic regression assumes that the data is properly scaled, this preprocessing step is crucial. This scaling is applied to both the training and testing datasets. We split the data into training and testing datasets and then scale them separately to avoid data leakage. Data leakage occurs when information from the training dataset influences the testing dataset, potentially leading to overly optimistic performance estimates.

4.3 Model Results Analysis

This section outlines the detailed results for each model. Their accuracies, sensitivity/recall and precision, and their F1-scores are all detailed in their respective classification reports here.

4.3.1 Logistic Regression results

The analysis has been carried out on the test set. In the confusion matrix for the logistic regression model, we see that of the 78 + 7 = 85 people that did not have diabetes, 78 (92%) were correctly classified. And of the 20 + 23 = 43 people that have diabetes, 23 were correctly classified. So, the logistic regression model performed reasonably well, especially in identifying non-diabetic cases.

```
model = linear_model.LogisticRegression()
model.fit(x_train,y_train)

#use the test data to make predictions and test the model
yp=model.predict(x_test)
loss=log_loss(y_test,yp)
conf_matrix = confusion_matrix(y_test, yp)

#print(loss)
#print("Accuracy:",metrics.accuracy_score(y_test, yp))
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[78  7]
 [20 23]]
```

Figure 4.5: Code snippet for confusion matrix of logistic regression

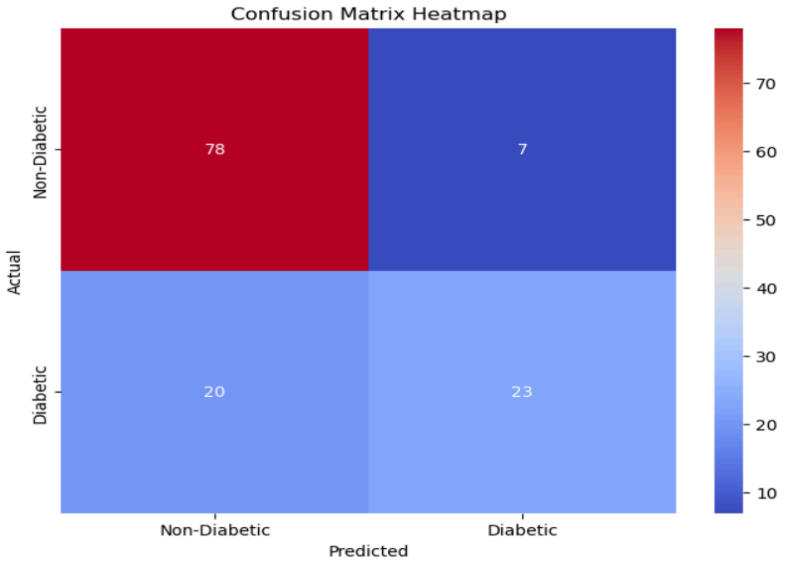

Figure 4.6: Confusion Matrix for LR

```
[15]: class_report = classification_report(y_test, yp)

      # Print classification report
      print("Classification Report:")
      print(class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.92      0.85        85
           1       0.77      0.53      0.63        43

    accuracy                           0.79       128
   macro avg       0.78      0.73      0.74       128
weighted avg       0.79      0.79      0.78       128
```

Figure 4.7: Classification report for LR (Showing performance scores).

4.3.2 Naïve Bayes results

The analysis has been carried out on the test set. In the confusion matrix, we see that of the 72 + 13 = 85 people who did not have diabetes, 72 (85%) were correctly classified. And of the 15 + 28 = 43 people who have diabetes, 28 (65%) were correctly classified. So, the Naive Bayes model performed reasonably well, especially in identifying non-diabetic cases.

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train, y_train)

#Predict the response for test dataset
y_pred2 = gnb.predict(x_test)
loss2=log_loss(y_test,y_pred2)

print(loss2)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred2))
```

```
conf_matrix2 = confusion_matrix(y_test, y_pred2)
print("Confusion Matrix:")
print(conf_matrix2)
```

```
Confusion Matrix:
[[72 13]
 [15 28]]
```

Figure 4.8: Code snippet for confusion matrix of naive bayes

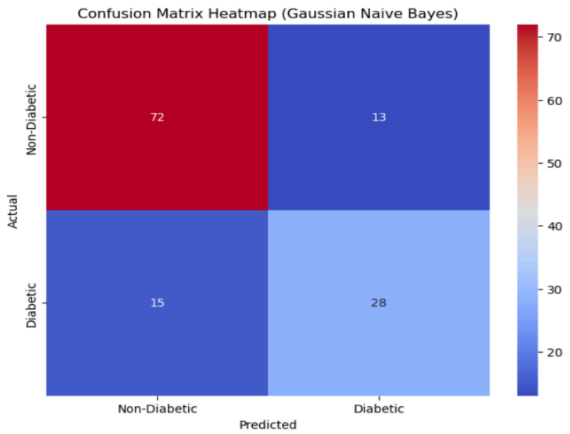

Figure 4.9: Confusion Matrix for NB

Classification Report:

	precision	recall	f1-score	support
0	0.83	0.85	0.84	85
1	0.68	0.65	0.67	43
accuracy			0.78	128
macro avg	0.76	0.75	0.75	128
weighted avg	0.78	0.78	0.78	128

Figure 4.10: Classification report for NB (Showing performance scores).

4.3.3 Decision tree classifier results

The analysis has been carried out on the test set. In the confusion matrix, we see that of the 61 + 24 = 85 people who did not have diabetes, 61 (72%) were correctly classified. And of the 22 + 21 = 43 people who have diabetes, 21 (49%) were correctly classified. So, the decision tree classifier performed reasonably well, especially in identifying non-diabetic cases.

```
# Make and fit your decision tree model
clf = DecisionTreeClassifier()
clf = clf.fit(x_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(x_test)
loss1=log_loss(y_test,y_pred)

#print(loss1)
#print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
conf_matrix2 = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix2)
```

```
Confusion Matrix:
[[61 24]
 [22 21]]
```

Figure 4.11: Code snippet for confusion matrix of decision tree classifier

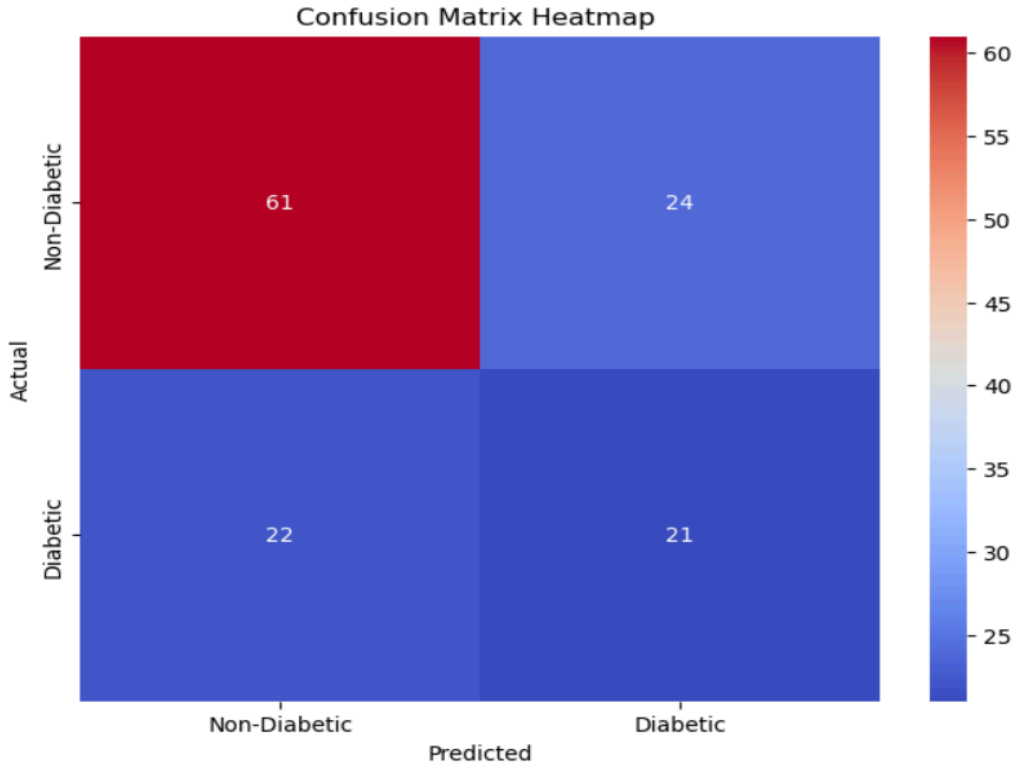

Figure 4.12: Confusion Matrix for decision tree classifier

```
Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.72      0.73        85
           1       0.47      0.49      0.48        43

    accuracy                           0.64       128
   macro avg       0.60      0.60      0.60       128
weighted avg       0.64      0.64      0.64       128
```

Figure 4.13: Classification report for DTC (Showing performance scores).

4.3.4 KNN results

The analysis has been carried out on the test set. In the confusion matrix, we see that of the 67 + 18 = 85 people who did not have diabetes, 67 (79%) were correctly classified. And of the 14 + 29 = 43 people who have diabetes, 29 (67%) were correctly classified. So, the KNN model performed reasonably well, especially in identifying non-diabetic cases.

```
#make your knn model using 3 as your k
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)

#Predict the response for test dataset
y_pred3 = knn.predict(x_test)
loss3=log_loss(y_test,y_pred3)

#print(loss3)
#print("Accuracy:",metrics.accuracy_score(y_test, y_pred3))
```

```
conf_matrix = confusion_matrix(y_test, y_pred3)

print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[67 18]
 [14 29]]
```

Figure 4.14: Code snippet for confusion matrix of KNN

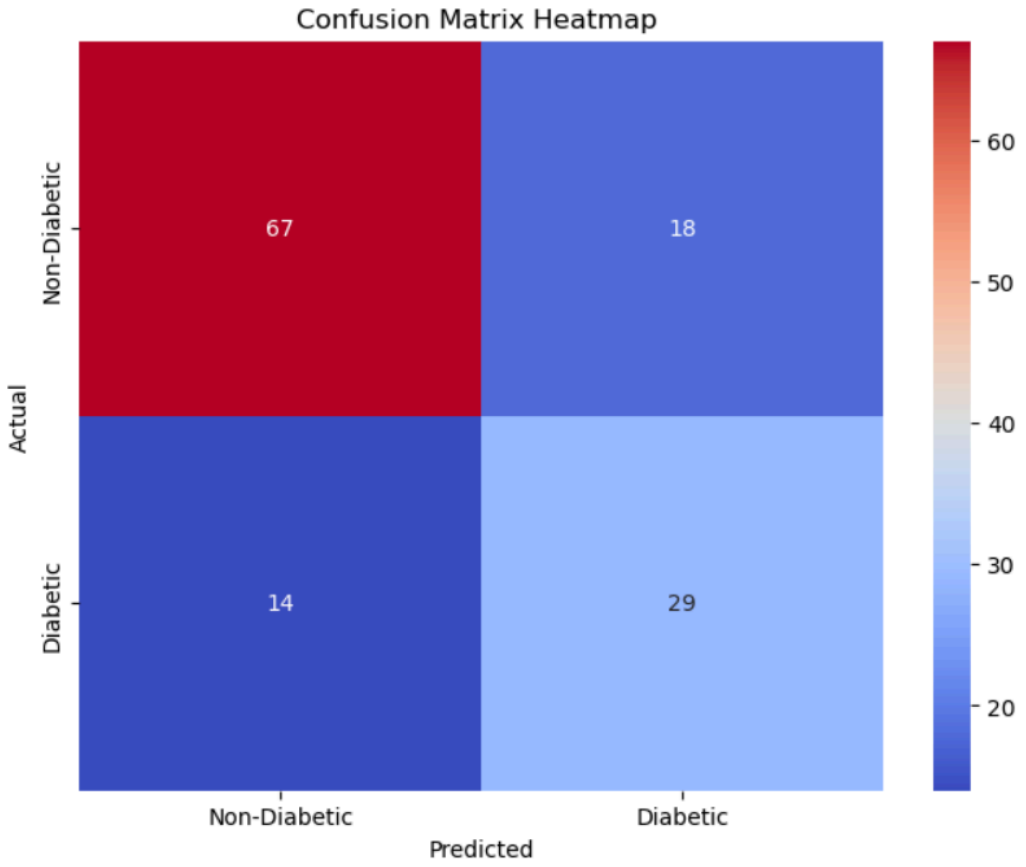

Figure 4.15: Confusion Matrix for KNN

```
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.79      0.81        85
           1       0.62      0.67      0.64        43

    accuracy                           0.75       128
   macro avg       0.72      0.73      0.73       128
weighted avg       0.76      0.75      0.75       128
```

Figure 4.16: Classification report for KNN (Showing performance scores).

4.3.5 Performance metrics for all algorithms

As discussed in earlier chapters, multiple algorithms were applied to the dataset. The algorithm that achieved the highest accuracy was the logistic regression Algorithm.

The table below summarizes the algorithms tested and their respective performance scores.

Table 4.1: Performance Metrics of Various Classification Algorithms

Classification Algorithm	Accuracy	Precision for class 0	Precision for class 1	Recall for class 0	Recall for class 1	F1-score for class 0	F1-score for class 1
Gaussian Naïve Bayes (NB)	78.13%	83.00%	68.00%	85.00%	65.00%	84.00%	67.00%
Logistic regression (LR)	78.91%	80.00%	77.00%	92.00%	53.00%	85.00%	63.00%
KNN	75.00%	83.00%	62.00%	79.00%	67.00%	81.00%	64.00%
Decision Tree classifier	65.63%	73.00%	47.00%	72.00%	49.00%	73.00%	48.00%

From the table above it can be seen that logistic Regression (LR) demonstrated superior performance with an accuracy of 78.91%. This performance metric indicates that Logistic Regression correctly classified approximately 78.91% of the instances in our dataset, which is the highest accuracy achieved compared to the other algorithms. Logistic Regression not only excelled in terms of accuracy but also outperformed some of the algorithms across additional performance metrics, including precision, recall, and the F1-score as seen in the table above.

4.4 Web Application Modules and Interface

The web application is developed using Django. Django is a robust and comprehensive web framework for Python that provides a wide range of tools and features to create complex and scalable web applications. Django's built-in components and adherence to best practices make it a powerful choice for both new and experienced developers, promoting rapid development and clean, pragmatic design. The figure 4.17 below shows the implementation of loading the serialized model into the Django web Application setup.

```
def predict(request):
    return render(request, 'predict.html')
def result(request):
    data = pd.read_csv('C:\\Users\\USER\\Downloads\\diabetes dataset.csv')

    x= data.drop('Outcome',axis=1)
    y= data['Outcome']
    #split your data into training and test data
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,shuffle=False)

    model = linear_model.LogisticRegression()
    model.fit(x_train,y_train)
    val1= float(request.GET['n1'])
    val12= float(request.GET['n2'])
    val3= float(request.GET['n3'])
    val4= float(request.GET['n4'])
    val5= float(request.GET['n5'])
    val6= float(request.GET['n6'])
    val7= float(request.GET['n7'])
    val8= float(request.GET['n8'])

    pred=model.predict([[val1, val12, val3, val4, val5,val6,val7,val8 ]])

    result1 =""
    if pred==[1]:
        result1= "positive"
    else:
        result1= "negative"
```

Figure 4.17: Code snippet showing web implementation in Django

The Web application serves as a user-friendly interface for connecting the client to the predictive model. The web application serves as a user interface where clients interact with the web application by filling a webform which serves as an input source to interact with the model in the backend. The users are required to input the required data, this data is then sent to be pre-processed in the backend and finally give the results. So, in order to display this, the user interface is designed using HTML and CSS and using the POST request the input data is passed to the server which does the remaining work. The sequence of steps taken by the program in generating the predictions are highlighted below:

i. To begin, the user fills in the necessary information on the form (predict.html).
ii. These inputs are now delivered to the result route through a POST request when the user hits the result button.
iii. Now, as seen in the accompanying code, these inputs are obtained from val1 to val8.
iv. These inputs are now given into our previously trained model, which we have loaded, and it simply predicts if a patient has diabetes or not.

v. By rendering result.html with predicted class, the result is presented in the form of the predicted class using render template.

The Web interface comprises of user interfaces which are the welcome page and the data input and Prediction page.

Welcome page

Figure 4.18 displays the welcome page. This contains the first page the user sees which welcomes them to the diabetes prediction system.

Figure 4.18: Welcome page to the system

Data input page

Figure 4.19 displays the prediction page. This contains the data input options for the user to input patient data required for prediction

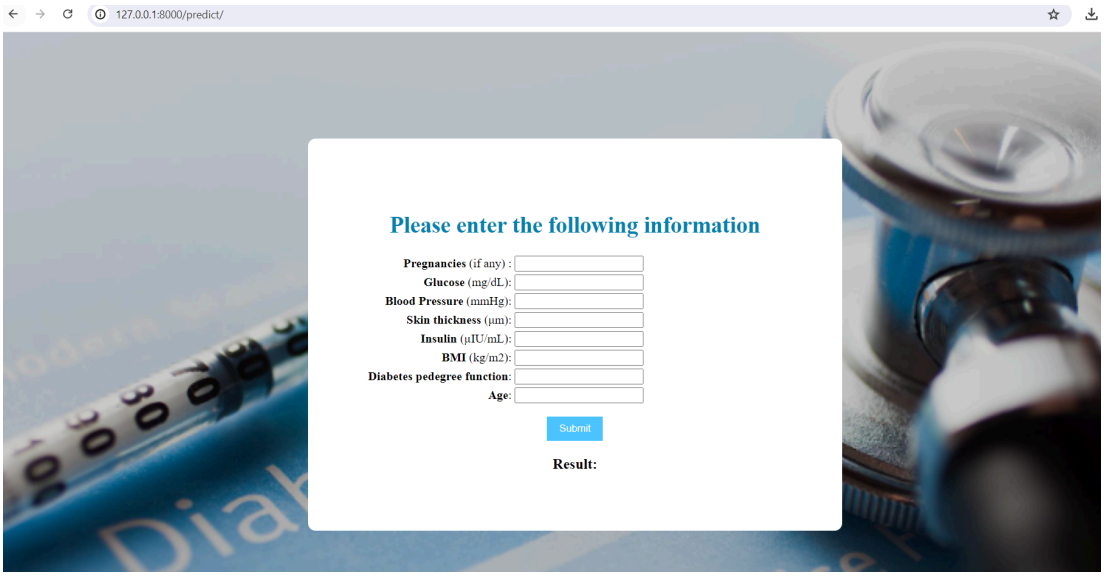

Figure 4.19: Data input prediction page

Prediction Result Page

Figure 4.20 and 4.21 displays the results of the prediction. This contains the prediction results by the logistic regression algorithm.

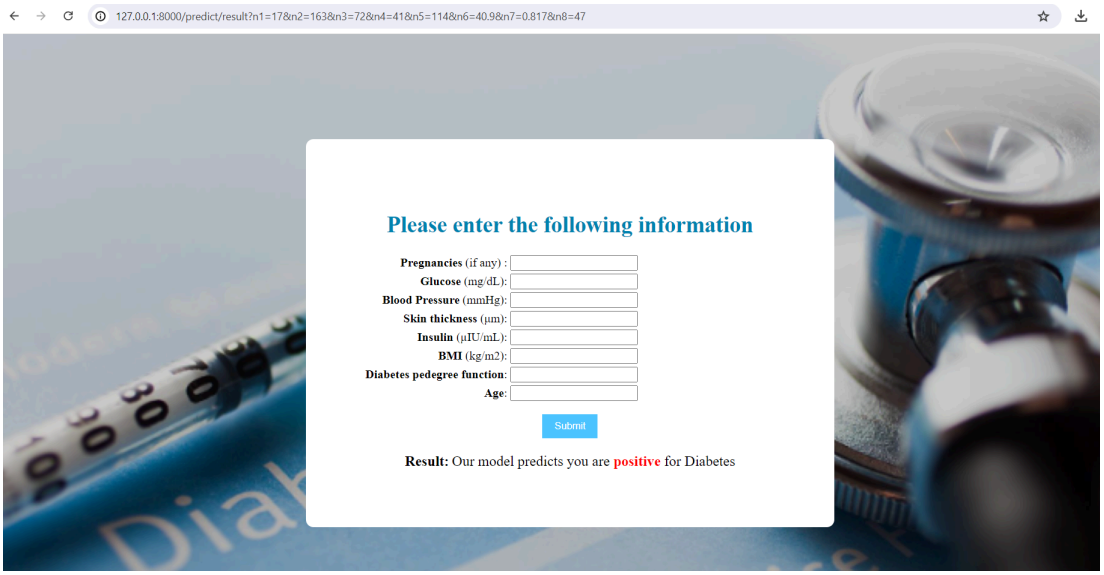

Figure 4.20: Result showing the patient is at risk of developing diabetes

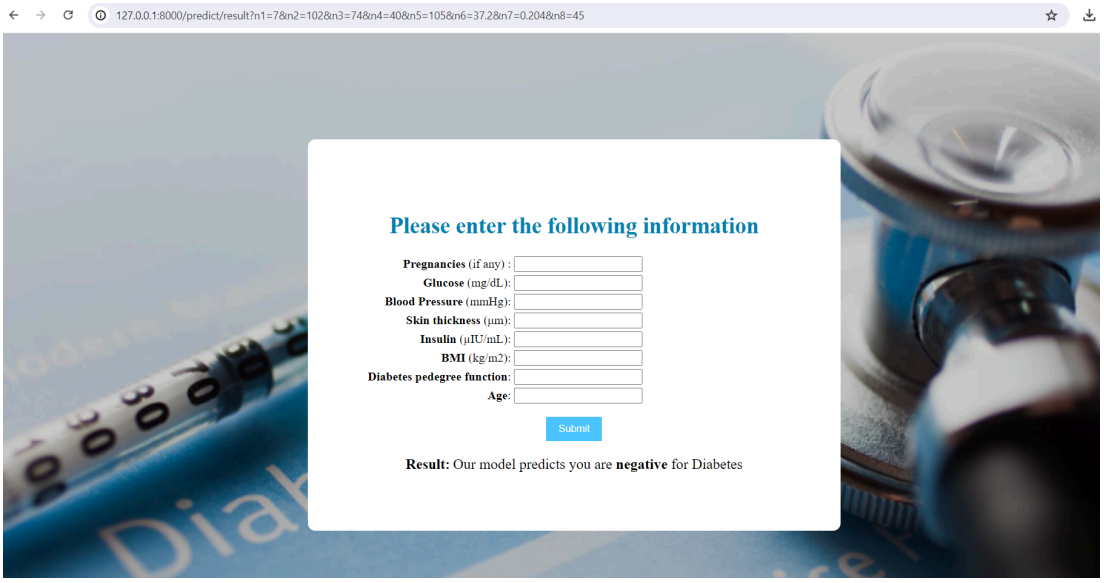

Figure 4.21: Result showing the patient is not at risk of developing diabetes

4.5 Chapter Summary

This chapter outlined the results of each of the models presented in this study with logistic regression identified as the optimal model for deployment due to its superior accuracy and robustness. Logistic regression is ideal for this application because it effectively handles the dataset and offers reliable performance, which is essential for real-time diabetes risk assessment. The web interface was also developed and integrated for practical user interaction, demonstrating the system's application in a real-world scenario.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Summary

The Early Prediction Tool for Diabetes was developed and implemented as outlined in this project. Through meticulous research and testing, particularly using the logistic regression model, a high level of accuracy in diabetes prediction has been achieved. The web application, built with Django, offers a user-friendly interface, ensuring flexibility and ease of use. Emphasizing feature importance and interpretability, the model provides valuable insights into the factors influencing diabetes prediction. Although developed and tested in a research context, the tool's scalability, reliability, and modularity make it suitable for real-world healthcare applications. It enables early intervention and personalized risk assessment, highlighting the importance of transparency in decision-making. This project not only demonstrates the effectiveness of logistic regression in predictive modeling but also sets a foundation for future research and practical implementations in healthcare, showcasing the transformative impact of technology in improving patient outcomes and healthcare management.

5.2 Achievements

Several objectives were successfully achieved using the stated methodology:

i) A robust predictive model was developed using logistic regression and other supervised learning algorithms to detect diabetes risk.

ii) Essential features for diabetes prediction were extracted from diverse data formats, with feature importance analysis conducted to enhance model interpretability.

iii) The system was integrated with a Django-based web application to provide a user-friendly platform for early diabetes risk assessment which is deployed to the live URL https://diabetesprediction-x0rs.onrender.com/ .

iv) The performance of the predictive model was thoroughly evaluated to ensure accuracy and reliability.

5.3 Recommendations

While the project was successfully implemented, there are some recommendations for further development:

i) Larger and Diverse Datasets: Expanding the dataset with a more diverse range of patient health records can enhance the model's ability to generalize across different demographics, improving robustness and predictive accuracy.

ii) Advanced Hyperparameter Tuning: Conducting thorough hyperparameter tuning experiments using automated techniques like grid search or random search can fine-tune the model's parameters for improved predictive performance and generalization.

iii) Enhanced Feature Engineering: Future iterations should focus on refining feature engineering techniques to capture nuanced relationships and patterns in the data, improving model interpretability and predictive power.

By implementing these recommendations and leveraging advancements in machine learning and data processing techniques, the Early Detection of Diabetes system can be further refined and optimized for more accurate and reliable predictions, ultimately improving healthcare outcomes and early intervention strategies in diabetes management.

REFERENCES

[1] B. F. Wee, S. Sivakumar, K. H. Lim, W. K. Wong, and F. H. Juwono, "Diabetes detection based on machine learning and deep learning approaches," *Multimed. Tools Appl.*, Aug. 2023, doi: 10.1007/s11042-023-16407-5.

[2] O. Iparraguirre-Villanueva, K. Espinola-Linares, R. O. Flores Castañeda, and M. Cabanillas-Carbonell, "Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes," *Diagn. Basel Switz.*, vol. 13, no. 14, p. 2383, Jul. 2023, doi: 10.3390/diagnostics13142383.

[3] L. Kopitar, P. Kocbek, L. Cilar, A. Sheikh, and G. Stiglic, "Early detection of type 2 diabetes mellitus using machine learning-based prediction models," *Sci. Rep.*, vol. 10, no. 1, p. 11981, Jul. 2020, doi: 10.1038/s41598-020-68771-z.

[4] I. Tasin, T. U. Nabil, S. Islam, and R. Khan, "Diabetes prediction using machine learning and explainable AI techniques," *Healthc. Technol. Lett.*, vol. 10, no. 1–2, pp. 1–10, 2023, doi: 10.1049/htl2.12039.

[5] O. A. Ebrahim and G. Derbew, "Application of supervised machine learning algorithms for classification and prediction of type-2 diabetes disease status in Afar regional state, Northeastern Ethiopia 2021," *Sci. Rep.*, vol. 13, no. 1, p. 7779, May 2023, doi: 10.1038/s41598-023-34906-1.

[6] R. Lozano *et al.*, "Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the Global Burden of Disease Study 2010," *Lancet Lond. Engl.*, vol. 380, no. 9859, pp. 2095–2128, Dec. 2012, doi: 10.1016/S0140-6736(12)61728-0.

[7] T. Vos *et al.*, "Years lived with disability (YLDs) for 1160 sequelae of 289 diseases and injuries 1990-2010: a systematic analysis for the Global Burden of Disease Study 2010," *Lancet Lond. Engl.*, vol. 380, no. 9859, pp. 2163–2196, Dec. 2012, doi: 10.1016/S0140-6736(12)61729-2.

[8] J. Bhutani and S. Bhutani, "Worldwide burden of diabetes," *Indian J. Endocrinol. Metab.*, vol. 18, no. 6, pp. 868–870, Nov. 2014, doi: 10.4103/2230-8210.141388.

[9] American Diabetes Association, "Diagnosis and classification of diabetes mellitus," *Diabetes Care*, vol. 37 Suppl 1, pp. S81-90, Jan. 2014, doi: 10.2337/dc14-S081.

[10] A. T. Kharroubi and H. M. Darwish, "Diabetes mellitus: The epidemic of the century," *World J. Diabetes*, vol. 6, no. 6, pp. 850–867, Jun. 2015, doi: 10.4239/wjd.v6.i6.850.

[11] M. A. Mollar-Puchades, V. Pallarés-Carratalá, M. S. Navas de Solís, and F. Piñón-Sellés, "Fasting Glucose Versus Oral Glucose ToleranceTesting in the Diagnosis of Diabetes Mellitus," *Rev. Esp. Cardiol. Engl. Ed.*, vol. 59, no. 12, pp. 1349–1350, 2006, doi: 10.1016/S1885-5857(07)60096-6.

[12] E. Selvin, "Hemoglobin A1c-Using Epidemiology to Guide Medical Practice: Kelly West Award Lecture 2020," *Diabetes Care*, vol. 44, no. 10, pp. 2197–2204, Sep. 2021, doi: 10.2337/dci21-0035.

[13] Y. Wu, Y. Ding, Y. Tanaka, and W. Zhang, “Risk factors contributing to type 2 diabetes and recent advances in the treatment and prevention,” *Int. J. Med. Sci.*, vol. 11, no. 11, pp. 1185–1200, 2014, doi: 10.7150/ijms.10001.

[14 L. Ismail, H. Materwala, and J. Al Kaabi, “Association of risk factors with type 2 diabetes: A systematic review,” *Comput. Struct. Biotechnol. J.*, vol. 19, pp. 1759–1785, 2021, doi: 10.1016/j.csbj.2021.03.003.

[15] C. N. Arighi *et al.*, “Overview of the BioCreative III Workshop,” *BMC Bioinformatics*, vol. 12 Suppl 8, no. Suppl 8, p. S1, Oct. 2011, doi: 10.1186/1471-2105-12-S8-S1.

[16] J. Sukanya, Dr. K. Rajiv Gandhi, and Dr. V. Palanisamy, “Heart Disease Prediction using Cluster Based MapReduce Paradigm,” *Int. J. Sci. Res. Publ. IJSRP*, vol. 11, no. 3, pp. 473–477, Mar. 2021, doi: 10.29322/IJSRP.11.03.2021.p11167.

[17] K. Y. Ngiam and I. W. Khor, “Big data and machine learning algorithms for health-care delivery,” *Lancet Oncol.*, vol. 20, no. 5, pp. e262–e273, May 2019, doi: 10.1016/S1470-2045(19)30149-4.

[18] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting,” *J. Educ. Res.*, vol. 96, no. 1, pp. 3–14, Sep. 2002, doi: 10.1080/00220670209598786.

[19] K. Vembandasamy, R. Sasipriya, and E. Deepa, “Heart Diseases Detection Using Naive Bayes Algorithm,” vol. 2, no. 9.

[20] M. Z. Al Yousef, A. F. Yasky, R. Al Shammari, and M. S. Ferwana, “Early prediction of diabetes by applying data mining techniques: A retrospective cohort study,” *Medicine (Baltimore)*, vol. 101, no. 29, p. e29588, Jul. 2022, doi: 10.1097/MD.0000000000029588.

[21] J. R. Raut, “PERFORMANCE EVALUATION OF VARIOUS SUPERVISED MACHINE LEARNING ALGORITHMS FOR DIABETES PREDICTION,” *Clin. Med.*, vol. 7, no. 8, 2020.

[22] K. Kangra and J. Singh, “Comparative analysis of predictive machine learning algorithms for diabetes mellitus,” *Bull. Electr. Eng. Inform.*, vol. 12, no. 3, pp. 1728–1737, Jun. 2023, doi: 10.11591/eei.v12i3.4412.

[23] Md. F. Faruque, Asaduzzaman, and I. H. Sarker, “Performance Analysis of Machine Learning Techniques to Predict Diabetes Mellitus,” in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox’sBazar, Bangladesh: IEEE, Feb. 2019, pp. 1–4. doi: 10.1109/ECACE.2019.8679365.

[24] A. Sharma, K. Guleria, and N. Goyal, “Prediction of Diabetes Disease Using Machine Learning Model,” in *International Conference on Communication, Computing and Electronics Systems*, vol. 733, V. Bindhu, J. M. R. S. Tavares, A.-A. A. Boulogeorgos, and C. Vuppalapati, Eds., in Lecture Notes in Electrical Engineering, vol. 733. , Singapore: Springer Singapore, 2021, pp. 683–692. doi: 10.1007/978-981-33-4909-4_53.

[25] S. Karkhanis, M. W. Q. Pathan, Y. Gandhi, and S. Deshpande, “Detection and Risk Analysis of Diabetes Using Machine Learning,” vol. 07, no. 05, 2020.

[26] C. Fiarni, E. M. Sipayung, and S. Maemunah, “Analysis and Prediction of Diabetes Complication Disease using Data Mining Algorithm,” *Procedia Comput. Sci.*, vol. 161, pp. 449–457, 2019, doi: 10.1016/j.procs.2019.11.144.

[27] F. A. Khan, K. Zeb, M. Al-Rakhami, A. Derhab, and S. A. C. Bukhari, “Detection and Prediction of Diabetes Using Data Mining: A Comprehensive Review,” *IEEE Access*, vol. 9, pp. 43711–43735, 2021, doi: 10.1109/ACCESS.2021.3059343.

[28] A. M. Adeshina, “Prediction of Diabetes Mellitus using Machine Learning Algorithms: Comparative Analysis of K-Nearest Neighbor, Random Forest and Logistic Regression,” *SLU J. Sci. Technol.*, pp. 205–213, Mar. 2023, doi: 10.56471/slujst.v6i.319.

[29] P. N. Thotad, “A Machine Learning-based Diagnosis and Prediction of Diabetes Mellitus Disease,” In Review, preprint, Apr. 2023. doi: 10.21203/rs.3.rs-2707299/v2.

[30] P. K. Darabi and M. J. Tarokh, “Type 2 Diabetes Prediction Using Machine Learning Algorithms”.

[31] B. S. Ahamed, M. S. Arya, and A. O. Nancy V, “Prediction of Type-2 Diabetes Mellitus Disease Using Machine Learning Classifiers and Techniques,” *Front. Comput. Sci.*, vol. 4, p. 835242, May 2022, doi: 10.3389/fcomp.2022.835242.

[32] T. S. Sharika, A. Al Farabe, G. Ashraf, N. Raonak, and A. Chakrabarty, “A Supervised Learning Approach by Machine Learning Algorithms to Predict Diabetes Mellitus (DM) Risk Score,” in *Soft Computing: Theories and Applications*, vol. 1381, T. K. Sharma, C. W. Ahn, O. P. Verma, and B. K. Panigrahi, Eds., in Advances in Intelligent Systems and Computing, vol. 1381. , Singapore: Springer Singapore, 2021, pp. 289–300. doi: 10.1007/978-981-16-1696-9_27.

[33] B. Rathi and F. Madeira, “Early Prediction of Diabetes Using Machine Learning Techniques,” in *2023 Global Conference on Wireless and Optical Technologies (GCWOT)*, Malaga, Spain: IEEE, Jan. 2023, pp. 1–7. doi: 10.1109/GCWOT57803.2023.10064682.

[34] S. I, S. Bhat, M. K. R, V. M, and C. M. B N, “Prediction of Diabetes Using Machine Learning Algorithm,” *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3430638.

[35] S. Sain, A. Singh, D. Bhatnagar, and A. S. Juneja, “Diabetes Prediction Using ML,” vol. 10, no. 03, 2023.

[36] Z. Mushtaq, M. F. Ramzan, S. Ali, S. Baseer, A. Samad, and M. Husnain, “Voting Classification-Based Diabetes Mellitus Prediction Using Hypertuned Machine-Learning Techniques,” *Mob. Inf. Syst.*, vol. 2022, pp. 1–16, Mar. 2022, doi: 10.1155/2022/6521532.

[37] A. Mujumdar and V. Vaidehi, “Diabetes Prediction using Machine Learning Algorithms,” *Procedia Comput. Sci.*, vol. 165, pp. 292–299, 2019, doi: 10.1016/j.procs.2020.01.047.

[38] M. A. Rahim, M. A. Hossain, M. N. Hossain, J. Shin, and K. S. Yun, “Stacked Ensemble-Based Type-2 Diabetes Prediction Using Machine Learning Techniques,” *Ann. Emerg. Technol. Comput.*, vol. 7, no. 1, pp. 30–39, Jan. 2023, doi: 10.33166/AETiC.2023.01.003.

[39] M. Banday, S. Zafar, P. Agarwal, and M. A. Alam, “Utilising Machine Learning Algorithms for Predictive Analysis of Diabetes”.

[40] S. M. H. Mahmud, M. A. Hossin, Md. R. Ahmed, S. R. H. Noori, and M. N. I. Sarkar, “Machine Learning Based Unified Framework for Diabetes Prediction,” in *Proceedings of the 2018 International Conference on Big Data Engineering and Technology*, Chengdu China: ACM, Aug. 2018, pp. 46–50. doi: 10.1145/3297730.3297737.

APPENDIX A

CODE FOR THE MODELS

```
import pandas
import numpy
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import log_loss
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import precision_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
from sklearn import linear_model

import pandas
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
pandas.set_option('display.max_rows', None)
df= pandas.read_csv('C:\\Users\\USER\\Downloads\\diabetes dataset.csv')
df.head()
def clean(col):
    global df
    for x in [col]:
        q75,q25 = numpy.percentile(df.loc[:,x],[75,25])
        intr_qr = q75-q25

        max = q75+(1.5*intr_qr)
        min = q25-(1.5*intr_qr)

        df.loc[df[x] < min,x] = numpy.nan
        df.loc[df[x] > max,x] = numpy.nan
```

```
clean('Age')
clean('DiabetesPedigreeFunction')
clean('BMI')
clean('Insulin')
clean('SkinThickness')
clean('BloodPressure')
clean('Glucose')
clean('Pregnancies')
print(df.isnull().sum())
df=df.dropna(axis=0)
print(df.isnull().sum())
df.boxplot()
sns.heatmap(df)
plt.show()
features_with_one_value = []
for column in df.columns:
    unique_values_count = df[column].nunique()
    print(f"{column}: {unique_values_count}")
    if unique_values_count == 1:
        features_with_one_value.append(column)

if not features_with_one_value:
     print("From the output above, it is obvious that there is no column with only one unique value.")
else:
    print("Features with just one unique value:", features_with_one_value)
discrete_features = []
for column in df.columns:
    unique_values_count = df[column].nunique()
    if unique_values_count <= 10:  # Adjust threshold as needed
        discrete_features.append(column)

print("Discrete numerical features:", discrete_features)
```

```
print("Total number of discrete numerical features:", len(discrete_features))
continuous_features = []
for column in df.columns:
    unique_values_count = df[column].nunique()
    if unique_values_count > 10:  # Adjust threshold as needed
        continuous_features.append(column)

print("Continuous numerical features:", continuous_features)
print("Total number of continuous numerical features:", len(continuous_features))
continuous_features = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

# Plot the distribution of continuous numerical features
fig, axes = plt.subplots(nrows=1, ncols=len(continuous_features), figsize=(15, 5))
for i, feature in enumerate(continuous_features):
    sns.histplot(df[feature], kde=True, ax=axes[i])
    axes[i].set_title(feature)
    axes[i].set_xlabel("Value")
    axes[i].set_ylabel("Frequency")

plt.tight_layout()
plt.show()
continuous_features = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']
label = 'Outcome'  # Replace with the name of your label column

# Plot the relation between continuous numerical features and labels
fig, axes = plt.subplots(nrows=1, ncols=len(continuous_features), figsize=(20, 5))
for i, feature in enumerate(continuous_features):
    sns.boxplot(x=label, y=feature, data=df, ax=axes[i])
    axes[i].set_title(f"{feature} vs {label}")
    axes[i].set_xlabel(label)
    axes[i].set_ylabel(feature)
```

```
plt.tight_layout()
plt.show()
continuous_features = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

# Plot the box plots for continuous numerical features to show outliers
fig, axes = plt.subplots(nrows=1, ncols=len(continuous_features), figsize=(20, 5))
for i, feature in enumerate(continuous_features):
    sns.boxplot(y=df[feature], ax=axes[i])
    axes[i].set_title(f"Box Plot of {feature}")
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel("Value")

plt.tight_layout()
plt.show()
corr_matrix = df.corr()

# Plot the heatmap of the correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title("Correlation Matrix Heatmap")
plt.show()
corr = df.corr()
f, ax = plt.subplots(figsize=(12, 10))
mask = numpy.triu(numpy.ones_like(corr, dtype=bool))
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(corr, annot=True, mask = mask, cmap=cmap)
x=numpy.array(df[['BloodPressure','DiabetesPedigreeFunction','Glucose','BMI','Age',
'Pregnancies']]).reshape(-1,6)
y=numpy.array(df['Outcome'])
#split your data into training and test data
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,shuffle=False)
```

```
model = linear_model.LogisticRegression()
model.fit(x_train,y_train)

#use the test data to make predictions and test the model
yp=model.predict(x_test)
loss=log_loss(y_test,yp)
conf_matrix = confusion_matrix(y_test, yp)

print(loss)
print("Accuracy:",metrics.accuracy_score(y_test, yp))

print("Confusion Matrix:")
print(conf_matrix)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='coolwarm',
xticklabels=['Non-Diabetic', 'Diabetic'], yticklabels=['Non-Diabetic', 'Diabetic'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
plt.show()
class_report = classification_report(y_test, yp)

# Print classification report
print("Classification Report:")
print(class_report)
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train, y_train)

#Predict the response for test dataset
y_pred2 = gnb.predict(x_test)
loss2=log_loss(y_test,y_pred2)
```

```
print(loss2)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred2))
conf_matrix2 = confusion_matrix(y_test, y_pred2)
print("Confusion Matrix:")
print(conf_matrix2)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix2, annot=True, fmt='d', cmap='coolwarm',
xticklabels=['Non-Diabetic', 'Diabetic'], yticklabels=['Non-Diabetic', 'Diabetic'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap (Gaussian Naive Bayes)')
plt.show()
class_report2 = classification_report(y_test, y_pred2)
print("Classification Report:")
print(class_report2)
# Make and fit your decision tree model
clf = DecisionTreeClassifier()
clf = clf.fit(x_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(x_test)
loss1=log_loss(y_test,y_pred)

print(loss1)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
conf_matrix2 = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix2)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix2, annot=True, fmt='d', cmap='coolwarm',
xticklabels=['Non-Diabetic', 'Diabetic'], yticklabels=['Non-Diabetic', 'Diabetic'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
```

```
plt.title('Confusion Matrix Heatmap ')
plt.show()
class_report2 = classification_report(y_test, y_pred)
print("Classification Report:")
print(class_report2)
from sklearn.cluster import KMeans
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(df)
    distortions.append(kmeanModel.inertia_)
plt.figure(figsize=(16,8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
#make your knn model using 3 as your k
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(x_train, y_train)

#Predict the response for test dataset
y_pred3 = knn.predict(x_test)
loss3=log_loss(y_test,y_pred3)

print(loss3)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred3))
conf_matrix = confusion_matrix(y_test, y_pred3)

print("Confusion Matrix:")
print(conf_matrix) plt.figure(figsize=(8, 6))
```

```
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='coolwarm',
xticklabels=['Non-Diabetic', 'Diabetic'], yticklabels=['Non-Diabetic', 'Diabetic'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap ')
plt.show()
class_report2 = classification_report(y_test, y_pred3)
print("Classification Report:")
print(class_report2)
```

APPENDIX B

CODE FOR WEB APP

```
#views.py
from django.shortcuts import render
import pandas
import numpy
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import log_loss
from sklearn.metrics import precision_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.model_selection import train_test_split
from sklearn import linear_model

import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from django.shortcuts import HttpResponse
from django.http import HttpRequest

def home(request):
    return render(request, 'home.html')
def predict(request):
    return render(request, 'predict.html')
def result(request):
    data = pd.read_csv('C:\\Users\\USER\\Downloads\\diabetes dataset.csv')

    x= data.drop('Outcome',axis=1)
    y= data['Outcome']
    #split your data into training and test data
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,shuffle=False)
```

```
    model = linear_model.LogisticRegression()
    model.fit(x_train,y_train)
    val1= float(request.GET['n1'])
    val12= float(request.GET['n2'])
    val3= float(request.GET['n3'])
    val4= float(request.GET['n4'])
    val5= float(request.GET['n5'])
    val6= float(request.GET['n6'])
    val7= float(request.GET['n7'])
    val8= float(request.GET['n8'])

    pred=model.predict([[val1, val12, val3, val4, val5,val6,val7,val8 ]])

    result1 =""
    if pred==[1]:
            result1= "<span style='font-size: larger;'>Our model predicts you are <b style='color: red;'>positive</b> for Diabetes</span>"
    else:
            result1= "<span style='font-size: larger;'>Our model predicts you are <b style='color: black;'>negative</b> for Diabetes</span>"


    return render(request, "predict.html", {"result2":result1})
#home.html
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <style type= text/css>
```

```
    div{
    color:white;
}
h1{
    color: 'white';
    font-family: arial, sans-serif;
    font-size: 60px;
    font-weight: bold;
    margin-top: 200px;
}
h2{
    color: 'white';
    font-family: arial, sans-serif;
    font-size: 15px;
    font-weight: bold;
    margin-top: 400px;
}

body{
    background-image: url("{% static '/Diabetes/images/diab.svg' %}");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: cover;
}
input[type=submit]{
    background-color: #4dc3ff;
    border: 2px;
    color: white;
    padding: 16px 32px;
    cursor: pointer;
    margin-top: 15px;
}
</style>
```

```
</head>
<body>
  <div style="text-align: center;">
   <h1>
       DIABETES PREDICTION SYSTEM
   </h1>
   <form action="predict">
      <input type="submit" value="Let's get started">
   </form>
  </div>
</body>
</html>
#predict.html
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>prediction page</title>
   <style>
     body{
         background-image: url("{% static '/Diabetes/images/diab.svg' %}");
         background-repeat: no-repeat;
         background-attachment: fixed;
         background-size: cover;
      }
      .main{
         position: fixed;
         top: 140px;
         left: 410px;
         width: 550px;
         background-color:#ffffff;
```

```
            border-radius: 10px;
            align-items: center;
            padding: 5%;
        }
        h1{
            color: #0086b3;
            font-size: 30px;
            font-weight: bold;
        }
        input[type=submit]{
            background-color: #4dc3ff;
            border: 2px;
            color: white;
            padding: 8px 16px;
            cursor: pointer;
            margin-top: 15px;}

            .rest{
            margin-top: 20px;
        }
    </style>
</head>
<body>

  <div style="text-align: center;"  class="main">
<h1>Please enter the following information</h1>
<form action="result">
    <table >
        <tr>
            <td align="right">Pregnancies:</td>
            <td align="left"> <input type="text" name="n1"></td>
        </tr>
        <tr>
```

```
            <td align="right">Glucose:</td>
            <td align="left"> <input type="text" name="n2"></td>
        </tr>
        <tr>
            <td align="right">Blood Pressure:</td>
            <td align="left"> <input type="text" name="n3"></td>
        </tr>
        <tr>
            <td align="right">Skin thickness:</td>
            <td align="left"> <input type="text" name="n4"></td>
        </tr>
        <tr>
            <td align="right">Insulin:</td>
            <td align="left"> <input type="text" name="n5"></td>
        </tr>
        <tr>
            <td align="right">BMI:</td>
            <td align="left"> <input type="text" name="n6"></td>
        </tr>
        <tr>
            <td align="right">Diabetes pedegree function:</td>
            <td align="left"> <input type="text" name="n7"></td>
        </tr>
        <tr>
            <td align="right">Age:</td>
            <td align="left"> <input type="text" name="n8"></td>
        </tr>

    </table>
    <input type="submit">
</form>
<div class="rest">
    <span style='font-size: larger;'><b>Result:</b></span> {{ result2|safe }}
```

```
</div>
</div>
</body>
</html>
```