# CERTIFICATE

This is to certify that the project work entitled “**DIABETES PREDICTION”** is a bonafide work carried out by **Ms. NAYELA QUDSIYA (1603-20-737-008)** in partial fulfilment of the requirements for the Summer Internship laboratory, VI sem **BACHELOR OF ENGINEERING IN INFORMATION & TECHNOLOGY** affiliated by the **OSMANIA UNIVERSITY**, Hyderabad, under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

| Summer Internship Coordinator | Internal Guide | HOD | External Examiner |
|---|---|---|---|
| Sameena Siddiqua<br>Asst. Professor | Tanveer Sultana<br>Asst. Professor | Dr. Ayesha Ameen<br>Professor & Head | |
| Department Of IT<br>DCET, Hyderabad | Department Of IT<br>DCET, Hyderabad | Department Of IT<br>DCET, Hyderabad | |

# DECLARATION

This is to certify that the work reported in the present project entitled “**DIABETES PREDICTION”** is a record of work done by us in the Department of Information Technology, Deccan College of Engineering and Technology, Osmania University. The reports are based on the project work done entirely by us and not copied from any other source.

**NAYELA QUDSIYA - 160320737008**

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude and indebtedness to my project supervisor **MS. P.SRUTHI REDDY**

Finally, I would like to take this opportunity to thanks my family for their support through the work. I sincerely acknowledge and thanks who gave directly or indirectly their support in this internship work.

**NAYELA QUDSIYA -160320737008**

# ABSTRACT

Nowadays, diabetes has become a common disease to the mankind from young to the old persons. The growth of the diabetic patients is increasing day-by-day due to various causes such as bacterial or viral infection, toxic or chemical contents mix with the food, auto immune reaction, obesity, bad diet, change in lifestyles, eating habit, environment pollution, etc. Hence, diagnosing the diabetes is very essential to save the human life from diabetes. The data analytics is a process of examining and identifying the hidden patterns from large amount of data to draw conclusions. In health care, this analytical process is carried out using machine learning algorithms for analyzing medical data to build the machine learning models to carry out medical diagnoses. This paper presents a diabetes prediction system to diagnosis diabetes. Moreover, this paper explores the approaches to improve the accuracy in diabetes prediction using medical data with various machine learning algorithms and methods. Using python language.

# Introduction

Diabetes is a chronic disease that directly affects the pancreas, and the body is incapable of producing insulin. Insulin is mainly responsible for maintaining the blood glucose level. Many factors, such as excessive body weight, physical inactivity high blood pressure, and abnormal cholesterol level, can cause a person get affected by diabetes. It can cause many complications, but an increase in urination is one of the most common ones [24]. It can damage the skin, nerves, and eyes, and if not treated early, diabetes can cause kidney failure and diabetic retinopathy ocular disease. According to IDF (International Diabetes Federation) statistics, 537 million people had diabetes around the world in 2021. In Bangladesh, approximately 7.10 million people had suffered from this disease, according to 2019 statistics.

Diabetes is the fast growing disease among the people even among the youngsters. Diabetes is caused by the increase level of the sugar (glucose) in the blood. The diabetes can be classified into two categories such as type 1 diabetes and type 2 diabetes. Type 1 diabetes is an autoimmune disease. In this case, the body destroys the cells that are essential to produce insulin to absorb the sugar to produce energy. This type can be caused regardless of obesity. The obesity is the increase of body mass index (BMI) than the normal level of BMI of an individual. Type 1 diabetes can occur in childhood or adolescence age. Type 2 diabetes usually affects the adults who are obese. In this type, the body resists observing insulin or fails to produce insulin. Type 2 generally occurs in the middle or aged groups. Moreover, there are other causes for diabetes such as bacterial or viral infection, toxic or chemical contents in food, auto immune reaction, obesity, bad diet, change of lifestyles, eating habit, environment pollution, etc. Diabetes leads various diseases such as cardiovascular complications, renal issues, retinopathy, foot ulcers, etc. Data analytic is a process of examining and identifying the hidden patterns from large amount of data for drawing conclusions. In health care, this analytical process is carried out using machine learning algorithms for analyzing the medical data to build machine learning models to carry out the medical diagnoses. Machine learning is a type of artificial intelligence (AI) that enables a system to learn by itself and develop the knowledge models to make decision by predicting the unknown data or label of the a given data. The machine learning algorithms can be roughly categorized into three types namely supervised learning, unsupervised learning and semi-supervised learning. The supervised learning algorithms are used when human expertise does not exist (navigating on Mars), humans are unable to explain their expertise (speech recognition). Solution changes in time series (routing on a computer function) and to solution needs to be adapted to particular cases (user biometrics). The supervised learning algorithms are classified into different types such as probability-based, function-based, rule-based, tree-based, instance-based, etc. The unsurprised learning is the descriptive type learning. This learning is used to describe or summarize the data. The examples of the unsupervised learning algorithms are clustering, association rule mining, etc. The semi-supervised learning is the combination of supervised and unsupervised. This paper presents a diabetes prediction system to diagnosis the diabetics. Moreover, the supervised learning algorithm is used to learn the diabetes data and to develop diabetes predication system for diagnosing diabetes. The accuracy of this prediction system is improved using pre-processing technique.

PROBLEM STATEMENT: In this project we want to identify a patient has diabetes or not. Our objective is to build a Diabetes prediction system using Machine learning techniques. In the past, such systems were rule -based. Machine learning offers powerful new ways. In this project, we will analyze dataset taken from Kaggle Website. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. It contains 768 instances and 8 attributes. The "target" field refers to patient is diabetic or not. It is integer valued 0 = no disease and 1 = disease.

## Software Requirement:

1. Operating System: WINDOWS 11
2. Coding Language: Python, Html
3. Tool: VS Code
4. Framework: Flask

## Hardware Requirement:

1. Monitor: 15.6” HD Display
2. System: 13 Processor
3. Ram: 8 GB

# LIST OF FIGURES

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

Diabetes is a chronic disease that directly affects the pancreas, and the body is incapable of producing insulin. Insulin is mainly responsible for maintaining the blood glucose level. Many factors, such as excessive body weight, physical inactivity high blood pressure, and abnormal cholesterol level, can cause a person get affected by diabetes. It can cause many complications, but an increase in urination is one of the most common ones. It can damage the skin, nerves, and eyes, and if not treated early, diabetes can cause kidney failure and diabetic retinopathy ocular disease. According to IDF (International Diabetes Federation) statistics, 537 million people had diabetes around the world in 2021. In Bangladesh, approximately 7.10 million people had suffered from this disease, according to 2019 statistics.

Diabetes is the fast growing disease among the people even among the youngsters. Diabetes is caused by the increase level of the sugar (glucose) in the blood. The diabetes can be classified into two categories such as type 1 diabetes and type 2 diabetes. Type 1 diabetes is an autoimmune disease. In this case, the body destroys the cells that are essential to produce insulin to absorb the sugar to produce energy. This type can be caused regardless of obesity. The obesity is the increase of body mass index (BMI) than the normal level of BMI of an individual. Type 1 diabetes can occur in childhood or adolescence age. Type 2 diabetes usually affects the adults who are obese. In this type, the body resists observing insulin or fails to produce insulin. Type 2 generally occurs in the middle or aged groups. Moreover, there are other causes for diabetes such as bacterial or viral infection, toxic or chemical contents in food, auto immune reaction, obesity, bad diet, change of lifestyles, eating habit, environment pollution, etc. Diabetes leads various diseases such as cardiovascular complications, renal issues, retinopathy, foot ulcers, etc. Data analytic is a process of examining and identifying the hidden patterns from large amount of data for drawing conclusions. In health care, this analytical process is carried out using machine learning algorithms for analyzing the medical data to build machine learning models to carry out the medical diagnoses. Machine learning is a type of artificial intelligence (AI) that enables a system to learn by itself and develop the knowledge models to make decision by predicting the unknown data or label of the a given data. The machine learning algorithms can be roughly categorized into three types namely supervised learning, unsupervised learning and semi-supervised learning. The supervised learning algorithms are used when human expertise does not exist (navigating on Mars), humans are unable to explain their expertise (speech recognition). Solution changes in time series (routing on a computer function) and to solution needs to be adapted to particular cases (user biometrics). The supervised learning algorithms are classified into different types such as probability-based, function-based, rule-based, tree-based, instance-based, etc. The unsurprised learning is the descriptive type learning. This learning is used to describe or summarize the data. The examples of the unsupervised learning algorithms are clustering, association rule mining, etc. The semi-supervised learning is the combination of supervised and unsupervised. This paper presents a diabetes prediction system to diagnosis the diabetics. Moreover, the supervised learning algorithm is used to learn the diabetes data and to develop diabetes predication system for diagnosing diabetes. The accuracy of this prediction system is improved using pre-processing technique.

# CHAPTER 2

# SYSTEM ANALYSIS

## 2.1 OBJECTIVES:

The objectives of diabetes prediction refer to the goals and aims of using predictive modeling and data analysis techniques to identify individuals at risk of developing diabetes or to predict future outcomes related to diabetes. These objectives are typically focused on improving healthcare, prevention, and patient management. Some of the key objectives of diabetes prediction include:

Early Detection, Risk Stratification, Personalized Medicine, Improved Patient Management, Public Health Initiatives, and Cost Reduction.

The objectives of diabetes prediction encompass a range of critical goals in healthcare. The primary aim is early detection, enabling timely intervention and preventive measures to reduce the onset and complications of diabetes. These objectives also include risk stratification, allowing healthcare providers to prioritize resources for high-risk individuals, thus ensuring they receive appropriate care and support. Personalized medicine is another key objective, tailoring treatment plans and recommendations to individual patients. By predicting diabetes risk, these models facilitate prevention and intervention strategies, guiding healthcare professionals in recommending lifestyle modifications or medications. Improved patient management, efficient resource allocation, and support for research and development are additional objectives. Predictive models also inform public health initiatives, contribute to cost reduction, promote patient education, and, ultimately, enhance the quality of life for individuals at risk of or living with diabetes. These objectives underscore the value of data-driven approaches in diabetes care, aiming to improve health outcomes and reduce the societal and economic burden of the disease.

- Early Detection: Early identification of individuals at risk of developing diabetes allows for timely intervention and preventive measures, which can help reduce the onset of the disease or its complications.
- Risk Stratification: Stratifying individuals based on their risk of developing diabetes helps healthcare providers prioritize resources and interventions for those who are at higher risk, ensuring that they receive appropriate care and support.
- Personalized Medicine: Diabetes prediction models can help tailor treatment plans and lifestyle recommendations to individual patients, improving the effectiveness of

interventions and outcomes.

- Public Health Initiatives: Diabetes prediction can inform public health campaigns and policies aimed at reducing the prevalence of diabetes and its associated healthcare burden.
- Cost Reduction: Early prediction and intervention can lead to cost savings in healthcare systems by reducing the long-term costs associated with diabetes complications.

## 2.2 PROBLEM SPECIFICATION:

PROBLEM STATEMENT: In this project we want to identify a patient has diabetes or not. Our objective is to build a Diabetes prediction system using Machine learning techniques. In the past, such systems were rule -based. Machine learning offers powerful new ways. In this project, we will analyze dataset taken from Kaggle Website. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. It contains 768 instances and 8 attributes. The "target" field refers to patient is diabetic or not. It is integer valued 0 = no disease and 1 = disease.

Diabetes is a chronic and increasingly prevalent health condition that poses a significant global public health challenge. Early detection and effective management are essential in mitigating its impact on individuals and healthcare systems. The objective of this project is to develop a predictive model that can accurately identify individuals at risk of developing diabetes, thereby enabling early intervention and personalized care. Current healthcare resources are often underutilized due to the lack of a systematic and data-driven approach to diabetes risk assessment. This project seeks to fill this gap by leveraging advanced data analytics and machine learning techniques to create a robust predictive model that can assist healthcare providers in identifying at-risk individuals, optimizing resource allocation, and improving the overall quality of diabetes care. The model will utilize various patient-related features, such as demographic data, medical history, and lifestyle factors, to predict the likelihood of diabetes development and to guide preventive measures. The successful implementation of this predictive model will lead to more efficient healthcare delivery, reduced healthcare costs, and improved patient outcomes in the context of diabetes management.

## 2.3 PROPOSED SYSTEM:

- EXISTING SYSTEM:

The existing system of diabetes prediction using machine learning typically involves various components and methodologies that are in use to forecast an individual's risk of developing diabetes. This system starts with data collection, where historical data, including demographic information, medical records, and lifestyle factors, is gathered from multiple sources, such as healthcare institutions, surveys, and wearable devices. This dataset is then preprocessed to handle data quality issues, missing values, and outliers, and feature engineering techniques are employed to extract relevant predictors.

- Machine learning models, such as logistic regression, decision trees, or support vector machines, are trained on this data, using historical outcomes to create predictive algorithms. Validation is crucial, and techniques like cross-validation and performance metrics are applied to assess the model's accuracy. In some cases, deep learning techniques, such as neural networks, may be used to capture complex patterns in the data.
- PROPOSED SYSTEM:

The deployment of these models varies, but they are often integrated into electronic health record systems or healthcare software tools. Users, such as healthcare providers, can input patient data into the system, which then generates risk predictions and may provide recommendations for preventive measures. The system may also incorporate alerting and notification features to ensure timely intervention for individuals identified as high risk.

The proposed system overcomes the above mentioned issue in an efficient way. It aims at analyzing the number of diabetic patients that are present in the dataset. The dataset is classified into trained and test dataset where the data can be trained individually, these algorithms are very easy to implement as well as very efficient in producing better results and can able to process large amount of data. Even for large dataset these algorithms are extremely fast and can able to give accuracy of about over 80%.

The proposed system for a diabetes prediction machine learning project is designed to leverage advanced data analytics and predictive modeling techniques to address the growing public health challenge of diabetes. The system will consist of several essential components and functionalities. It will start with data collection and integration, gathering relevant patient data from sources like electronic health records, surveys, and wearable devices, covering demographics, medical history, lifestyle factors, and biomarkers. This data will undergo preprocessing, including handling missing values and feature engineering, ensuring the dataset is ready for analysis.

Feature selection algorithms will then be applied to identify the most informative variables for diabetes prediction, optimizing the model's focus. The core of the system will involve the development of a machine learning model, trained on historical data with known diabetes outcomes. The model will be rigorously validated using techniques like cross-validation and performance metrics to ensure its predictive accuracy.

Upon successful validation, the model will be deployed within the healthcare system and integrated into existing software tools. A user-friendly interface will be provided for healthcare providers to input patient data and obtain risk predictions, which will be accompanied by recommendations for preventive measures. The system will also generate alerts and notifications to ensure timely intervention for high-risk individuals.

Data security and privacy measures will be paramount, safeguarding sensitive patient information and adhering to regulatory requirements. Continuous monitoring and updates will be part of the system to adapt to changing patient populations and evolving medical knowledge. The project will also include outcome monitoring to assess the impact of the predictive model on diabetes prevention and management.

Through comprehensive documentation and reporting, the system aims to provide transparency and compliance with healthcare administrators, researchers, and regulatory agencies. Ultimately, this proposed system seeks to improve healthcare outcomes, enhance diabetes prevention and management, and reduce the societal and economic burden of diabetes, contributing to a more data-driven and patient-centered approach to healthcare.

## 2.4 APPLICATIONS:

The applications of diabetes prediction ML projects are diverse and hold the potential to significantly impact healthcare, public health, and research. These projects are instrumental in early diagnosis and intervention, enabling healthcare providers to identify individuals at high risk of developing diabetes and initiate timely medical care. Furthermore, they support the development of personalized treatment plans by tailoring healthcare strategies based on individual risk profiles and health data. In the realm of preventive healthcare, diabetes prediction models allow for the targeted implementation of preventive measures, such as lifestyle modifications and regular check-ups, to prevent or delay the onset of diabetes. Resource allocation within healthcare systems can be optimized by focusing on individuals with the highest risk, resulting in more efficient healthcare resource utilization and cost savings.

Health insurance companies can use predictive models to assess the risk of diabetes in policyholders, allowing for customized insurance plans and incentives for healthy behaviors. By integrating with mobile health apps and wearable devices, these projects provide real-time risk assessment and recommendations to individuals, motivating healthier lifestyles. Telemedicine platforms can incorporate diabetes prediction to offer remote monitoring and consultation for patients at risk of or living with diabetes, improving healthcare accessibility and delivery. Moreover, predictive models support clinical trials and drug development, aiding in patient selection and the development of novel diabetes treatments and medications. Ultimately, the primary application of diabetes prediction ML projects is to enhance the quality of life for individuals by helping them make informed decisions, receive timely care, and alleviate the burden of diabetes, a condition with a substantial impact on global health and well-being.

## 2.5 HARDWARE AND SOFTWARE SPECIFICATION:

- SOFTWARE REQUIREMENTS:

1. Operating System: WINDOWS 11.
2. Coding Language: Python, Html.
3. Tool: VS Code.
4. Framework: Flask.

- HARDWARE REQUIREMENTS:

1. Monitor: 15.6" HD Display.
2. System: 13 Processor.
3. Ram: 8 GB.

# CHAPTER 3
# PROJECT CATEGORY

## 3.1 ABOUT PYTHON:

PYTHON (programming language)

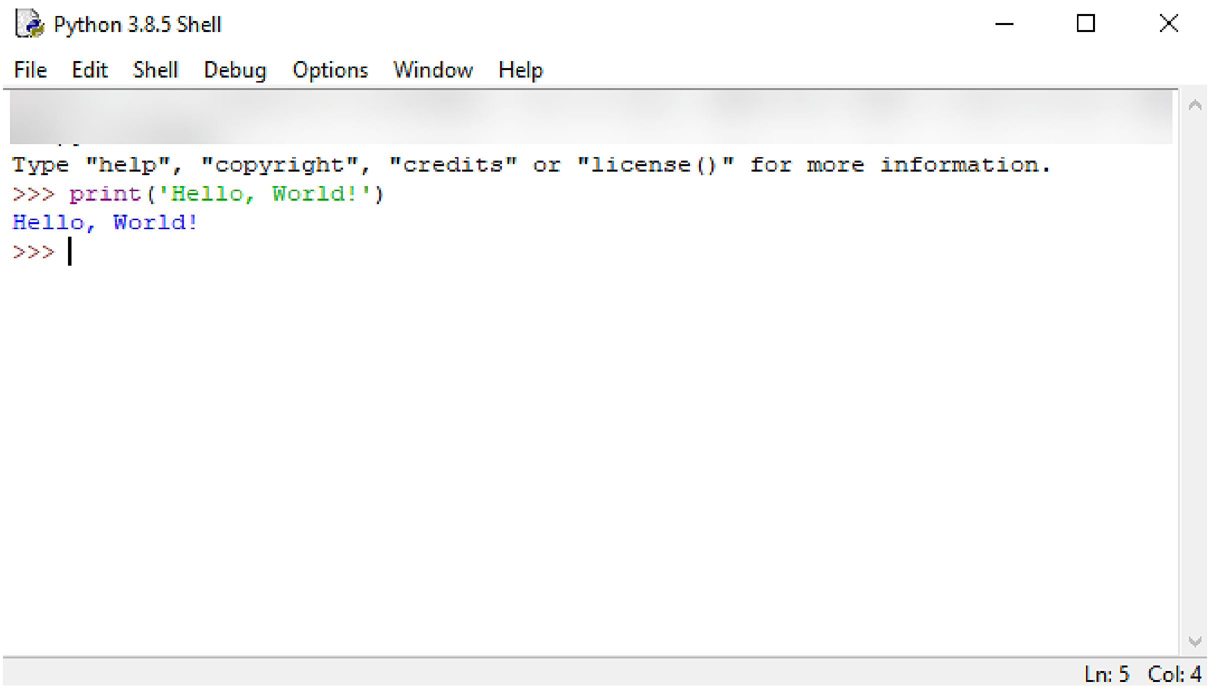
Python's history is an interesting tale! It all began in the late 1980s when Guido van Rossum, a Dutch programmer, started working on a new programming language during his Christmas holidays. He named it Python, not after the snake, but after the British comedy group Monty Python, whose work he enjoyed.

Python's first public release, Python 0.9.0, came in 1991. The language was designed with readability and simplicity in mind, emphasizing code readability and allowing developers to express concepts in fewer lines of code than might be possible in languages like C++ or Java.

Python 2 was the main version for many years, but in 2008, the development team announced that they would no longer improve Python 2 after 2020. This marked the beginning of the transition to Python which brought significant improvements and changes but also meant some code written in Python 2 needed to be modified to run on Python 3.

Python's popularity skyrocketed over the years due to its versatility and ease of use. It's used in various fields, from web development to scientific computing and artificial intelligence. The Python Software Foundation, a non-profit organization, manages and directs the development of Python.

Python, a programming language created by Guido van Rossum in the late 1980s, has emerged as a powerhouse in the world of software development. Recognized for its readability and simplicity, Python's syntax is designed to be clear and concise, making it an ideal choice for both beginners and seasoned developers. Its versatility is a standout feature, as Python finds applications across a spectrum of domains, including web development, data science, machine learning, and artificial intelligence. The language boasts an extensive standard library, embodying the "batteries included" philosophy, which provides modules for diverse tasks and facilitates efficient code reuse. Python's interpreted and interactive nature accelerates development, allowing users to experiment with code in real-time. The language's vibrant community, coupled with the Python Package Index (PyPI), contributes to a rich ecosystem of libraries and frameworks. Whether tackling small scripts or large-scale projects, Python's support for procedural and object-oriented programming, cross-platform compatibility, and open-source nature make it a preferred language, fostering collaboration and innovation in the ever-evolving landscape of software development.



```
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello, World!')
Hello, World!
>>>
```

FIG:3.1 PYTHON PROBLEM

This classic "Hello, World!" program is often the first program people write when learning a new programming language. In Python, it's just a single line, demonstrating the language's readability and simplicity. If you run this script, you should see the phrase "Hello, World!" displayed in your console or terminal.

Python is a cross-platform programming language, meaning it can run on various operating systems without modification. Whether you are using Windows, macOS, or Linux, you can write Python code and execute it on any of these platforms. This platform independence is one of the reasons for Python's widespread popularity.

To run a Python program, you generally need to have a Python interpreter installed on your system. Most operating systems come with Python pre-installed, but you can always download and install the latest version from the official Python website (https://www.python.org/).

Python's versatility across different platforms makes it a flexible and accessible choice for developers, allowing them to write code that can be seamlessly executed on different operating systems without major modifications.

The Python runtime environment encompasses the essential components that facilitate the execution of Python code. At its core is the Python interpreter, the engine responsible for translating and executing Python scripts. Complementing this, the standard library furnishes a wealth of modules and packages, offering a broad spectrum of functionalities without the need for additional installations. Virtual environments provide isolated spaces, enabling the management of project-specific dependencies and preventing conflicts within the global Python environment. External packages and modules, sourced from the Python Package Index (PyPI), further enrich the runtime environment by offering a vast array of ready-to-use tools. While not intrinsic to the runtime, integrated development environments (IDEs) enhance the development experience, providing features for coding, testing, and debugging. Throughout the execution, the runtime environment dynamically manages runtime data, including variables and objects, in the system memory. Whether deployed locally or on a server, comprehending and optimizing the Python runtime environment is pivotal for developers aiming to ensure compatibility and streamline the performance of their Python programs.

Uses of Python:

- Web Development: Python is widely used for web development. Frameworks like Django and Flask simplify the process, making it efficient to build robust and scalable web applications.
- Data Science and Machine Learning: Python is a go-to language for data scientists and machine learning engineers. Libraries such as NumPy, pandas, and scikit-learn make it easy to work with data, perform analysis, and implement machine learning algorithms.
- Artificial Intelligence: Python is extensively used in AI development. Frameworks like Tensor Flow and PyTorch enable developers to build and train deep learning models for various AI applications.
- Game Development: Python is employed in game development, often for scripting and automation within game engines. Libraries like Pygame provide a foundation for building 2D games.
- Desktop GUI Applications: With libraries like Tkinter, PyQt, and wxPython, Python is used to create graphical user interface (GUI) applications for desktop platforms.
- Cybersecurity: Python is used for cybersecurity tasks, including penetration testing, scripting for security tools, and developing security-related applications.
- Education: Python's readability and simplicity make it an excellent choice for teaching programming. Many educational institutions use Python as a first language for programming courses.
- Finance: In the financial industry, Python is employed for tasks like algorithmic trading, risk management, and quantitative analysis.

Information about Flask:

Flask, a lightweight and versatile web framework for Python, has gained immense popularity for its simplicity and flexibility in building web applications. Positioned as a micro framework, Flask provides a minimalistic core that empowers developers to tailor their applications by selecting the components they need. The framework follows the WSGI standard and excels in web development, offering intuitive routing through decorators, support for various HTTP methods, and a built-in development server for testing and debugging. Flask's use of the Jinja2 template engine facilitates the creation of dynamic HTML pages, promoting a clean separation of logic and presentation. Leveraging the Werkzeug WSGI toolkit, Flask ensures efficient handling of HTTP requests and responses. Its extensibility is a standout feature, allowing developers to integrate a plethora of Flask extensions for additional functionalities. The framework also shines in RESTful API development, making it a preferred choice for creating scalable and maintainable web services. With an active community, well-maintained documentation, and a focus on simplicity, Flask continues to be a favored option for developers seeking a streamlined yet powerful solution for web development in Python.

Flask, as a web framework for Python, stands out for its simplicity, flexibility, and ease of use. Born as a micro framework, Flask provides a minimalistic yet powerful foundation, allowing developers to structure their applications according to their specific needs. Its unobtrusive nature means that it doesn't impose rigid structures on developers, making it an excellent choice for projects of varying sizes and complexities.

One of Flask's defining features is its routing mechanism, which allows developers to map URLs to specific functions easily. This simplicity extends to the template engine, Jinja2, promoting clean and maintainable code by separating logic and presentation. Flask embraces the philosophy of "Don't reinvent the wheel," encouraging the use of extensions for additional features like authentication, database integration, and more.

Flask excels in RESTful API development, supporting various HTTP methods and providing tools to create scalable and efficient web services. It includes a built-in development server for quick testing, although for production, deploying with servers like Gunicorn or uWSGI is recommended.

The Flask community plays a vital role in its success, contributing to a wealth of resources, extensions, and tutorials. The framework's documentation is comprehensive and well-maintained, making it accessible for both beginners and experienced developers.

In summary, Flask's strength lies in its minimalist philosophy, allowing developers to build web applications and APIs with the components they need, promoting clean code and adaptability. Whether you're a newcomer or a seasoned developer, Flask provides a solid foundation for creating web applications in Python.

## 3.2 ABOUT HTML:

HTML, or Hypertext Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It has a rich history dating back to the early days of the internet.

HTML was first introduced by Sir Tim Berners-Lee in 1991 while working at CERN (European Organization for Nuclear Research). The initial version, HTML 1.0, was a basic language with a limited set of tags for structuring documents. It primarily focused on linking documents together using hyperlinks.

As the internet rapidly expanded in the mid-1990s, so did the need for more sophisticated web pages. This led to the development of HTML 2.0 in 1995, which introduced new features like forms and basic tables. However, it was HTML 3.2, released in 1997, that marked a significant milestone by adding support for scripting languages like JavaScript and enhancing support for style sheets.

The late 1990s and early 2000s saw a period of rapid innovation with the development of HTML 4.01 in 1999. This version introduced more complex formatting options, multimedia support, and improved accessibility features. Around the same time, web browsers like Netscape Navigator and Internet Explorer competed to support these new HTML features.

The evolution of HTML continued with XHTML, a reformulation of HTML as an XML application, maintaining its compatibility with XML standards. However, XHTML adoption faced challenges, and HTML5 emerged as the next major revision. HTML5, finalized in 2014, brought a host of new features, including native support for audio and video, canvas for graphics, and improvements for structuring content.

HTML5 has become the modern standard for web development, providing a foundation for creating dynamic, multimedia-rich web pages and applications. Its continuous development and widespread adoption ensure that HTML remains a crucial element in shaping the online experience.

HTML, or Hypertext Markup Language, serves as the backbone of the World Wide Web, providing a standardized way to structure and present content. Born in 1991 under the guidance of Sir Tim Berners-

Lee, HTML initially focused on creating hyperlinked documents. Over the years, it has undergone several transformations to keep pace with the dynamic nature of the internet.

From its humble beginnings with HTML 1.0, which laid the groundwork for document linking, to the more feature-rich HTML 2.0 and HTML 3.2, which introduced elements like forms and scripting support, the language evolved rapidly. The late 1990s witnessed the advent of HTML 4.01, a version that expanded formatting capabilities, incorporated multimedia elements, and enhanced accessibility.

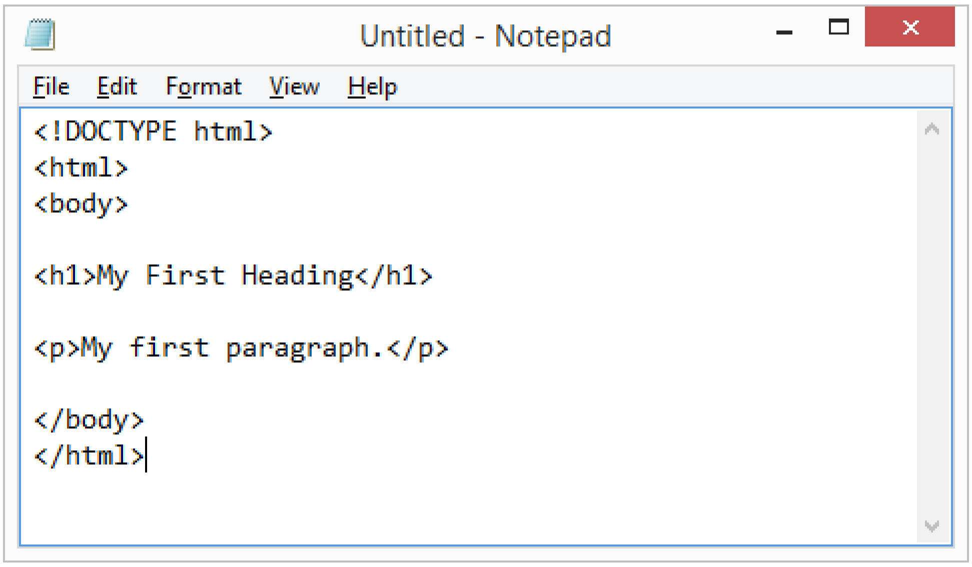
The subsequent shift to XHTML aimed to align HTML with XML standards, but the practical challenges of adoption paved the way for HTML5. Officially standardized in 2014, HTML5 revolutionized web development by introducing native support for audio and video, the canvas element for graphics, and a more versatile structure for content. This modern incarnation of HTML has become the cornerstone of contemporary web development, enabling the creation of dynamic, interactive, and visually appealing websites. As technology advances, HTML continues to evolve, ensuring its relevance in shaping the ever-changing landscape of the digital realm.

The HTML runtime environment is a dynamic space where HTML documents come to life, engaging users and providing interactive and responsive experiences. In this environment, the browser takes center stage as it interprets and renders HTML, along with its companion technologies like CSS (Cascading Style Sheets) and JavaScript.

When a user accesses a web page, the browser's rendering engine parses the HTML document, creating a Document Object Model (DOM) that represents the structure and content of the page. The CSS styles are applied to define the visual presentation, ensuring a cohesive and aesthetically pleasing layout.

The runtime environment truly comes to life with the integration of JavaScript. This scripting language adds a layer of interactivity, allowing developers to manipulate the DOM, respond to user actions, and dynamically update content. JavaScript also facilitates communication with servers, enabling the retrieval of data without requiring a full page reload through techniques like AJAX (Asynchronous JavaScript and XML).

As users interact with the web page, the runtime environment continuously processes events, triggers scripts, and updates the DOM, creating a seamless and engaging user experience. With the advent of modern web development practices and technologies, such as Single Page Applications (SPAs) and front-end frameworks like React and Vue.js, the HTML runtime environment has evolved to handle more complex and dynamic applications, pushing the boundaries of what can be achieved within the browser.



```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

FIG: 3.2 HTML PROGRAM

This simple HTML program consists of the basic structure with an HTML document, head, and body. It includes a title in the head section and a heading (<h1>) along with a paragraph (<p>) in the body section. You can copy and paste this code into an HTML file and open it in a web browser to see a basic webpage with a greeting.

HTML itself is not a platform but rather a markup language used for creating the structure of web pages. However, when we talk about the platform associated with HTML, we usually refer to the combination of HTML, CSS, and JavaScript within a web browser.

Web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and others serve as the platform for rendering and interpreting HTML documents. These browsers provide a runtime environment where HTML is parsed, the Document Object Model (DOM) is created, CSS styles are applied, and JavaScript is executed. This platform enables the presentation of content on the World Wide Web, offering users a visually appealing and interactive experience.

Additionally, various web development platforms and frameworks, such as React, Angular, and Vue.js, leverage HTML alongside other technologies to build dynamic and responsive web applications. These platforms extend the capabilities of HTML, allowing developers to create sophisticated user interfaces and enhance the overall user experience.

Uses of HTML:

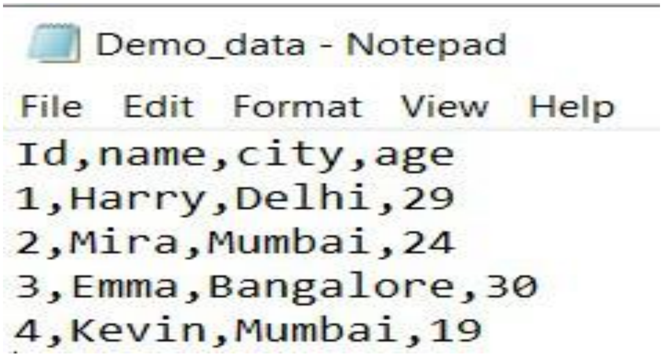
- Web Page Structure: HTML is the backbone of web pages, providing the structure and layout for content. It defines elements such as headings, paragraphs, lists, and more, organizing information in a readable and logical manner.
- Hyperlinking: HTML allows the creation of hyperlinks, enabling users to navigate between different web pages. The <a> (anchor) tag is a fundamental HTML element for linking to other documents or resources.

- Multimedia Integration: HTML supports the embedding of multimedia elements such as images, audio, and video. This is achieved through tags like <img>, <audio>, and <video>, enhancing the visual and auditory aspects of web content.
- Forms and User Input: HTML provides form elements (e.g., <form>, <input>, <select>, <text area>) for creating interactive forms. This allows users to submit data to a server, facilitating various online interactions like user registration, surveys, and more.
- Semantic Markup: HTML includes semantic tags that convey the meaning and structure of content to both browsers and developers. Semantic tags such as <header>, <footer>, <article>, and <section> enhance accessibility and search engine optimization.
- Responsive Web Design: HTML works in conjunction with Cascading Style Sheets (CSS) to create responsive web designs. Through responsive design techniques, developers can ensure that web pages adapt to different screen sizes and devices.
- Search Engine Optimization (SEO): Proper use of HTML, especially semantic markup, can positively impact a website's search engine ranking. Search engines rely on HTML structure to understand and index web content effectively.
- Documentation: HTML is often used for creating documentation, especially in combination with tools like Markdown. It provides a straightforward way to present information in a structured format.
- Integration with JavaScript: HTML works seamlessly with JavaScript, allowing developers to add dynamic behavior and interactivity to web pages. JavaScript can manipulate the HTML Document Object Model (DOM) for real-time updates and user interactions.

## 3.3 ABOUT CSV FILE:

A CSV (Comma-Separated Values) file is a simple and widely used format for storing tabular data, where each line of the file represents a row of data, and the values within a row are separated by commas. CSV files are plain text files, making them human-readable and easily understandable. This format is commonly employed for data exchange between different applications and systems, as it provides a lightweight and universally supported means of representing structured data.

In a CSV file, each line typically corresponds to a record, and the values within that line are the fields or attributes of the record. The first row of a CSV file often contains headers, describing the meaning of each column. For example:



```
Id,name,city,age
1,Harry,Delhi,29
2,Mira,Mumbai,24
3,Emma,Bangalore,30
4,Kevin,Mumbai,19
```

FIG: 3.3 CSC FILE EXAMPLE

In this example, the headers are "Id," "Name," "Age," and "City," and each subsequent line represents a person's information with corresponding values.

CSV files are versatile and widely used in various contexts, such as data import/export, spreadsheet applications, and database interactions. They are favored for their simplicity and ease of integration with different programming languages and tools. While commas are the most common delimiter, other characters like semicolons or tabs can be used based on the specific requirements of the application.

CSV files play a crucial role in data manipulation, analysis, and transfer due to their straightforward structure and compatibility across different platforms and software. They are especially useful for handling large datasets, making them an integral part of data-centric workflows in many industries.

Uses of CSV file:

- Data Backup and Storage: CSV files are a common choice for backing up structured data. They offer a simple and human-readable format, making it easy to review and restore data if needed.
- Statistical Analysis and Research: Researchers and data analysts often use CSV files to store and share datasets. The simplicity of the format allows for easy integration with statistical analysis tools and programming languages like R or Python.
- Machine Learning and Data Science: CSV files are a common data input format for machine learning models and data science projects. They can store labelled datasets for training models and serve as a standard way to share data among team members.
- Configuration Files: CSV files are sometimes used as configuration files for software applications. They provide a straightforward way to store settings or parameters in a structured format.

# CHAPTER 4

# DESIGN

## 4.1 ARCHITECTURE:

The architecture of a diabetes prediction system using machine learning encompasses a series of interconnected stages, each contributing to the development and deployment of an effective model. The process begins with the collection of diverse and representative health-related datasets, incorporating features such as age, body mass index (BMI), blood pressure, and insulin levels. Following data collection, thorough pre-processing is essential to handle missing values, outliers, and ensure data consistency. This may involve normalization and feature scaling to prepare the data for optimal model performance.

The next critical step involves feature selection, where techniques are employed to identify the most relevant variables for prediction, aiding in dimensionality reduction and model interpretability. With pre-processed data and selected features, the model selection phase begins. Various machine learning algorithms, including logistic regression, decision trees, random forests, support vector machines, and neural networks, are considered based on the specific characteristics of the dataset and the prediction task.

Once the model is selected, the training process commences, during which the model learns patterns and relationships within the training dataset, adjusting its parameters to minimize prediction errors. Following training, rigorous model evaluation is conducted using metrics such as accuracy, precision, recall, and F1-score to assess its performance on unseen data. Hyper parameter tuning is then applied to optimize the model's parameters, enhancing its generalization capabilities.

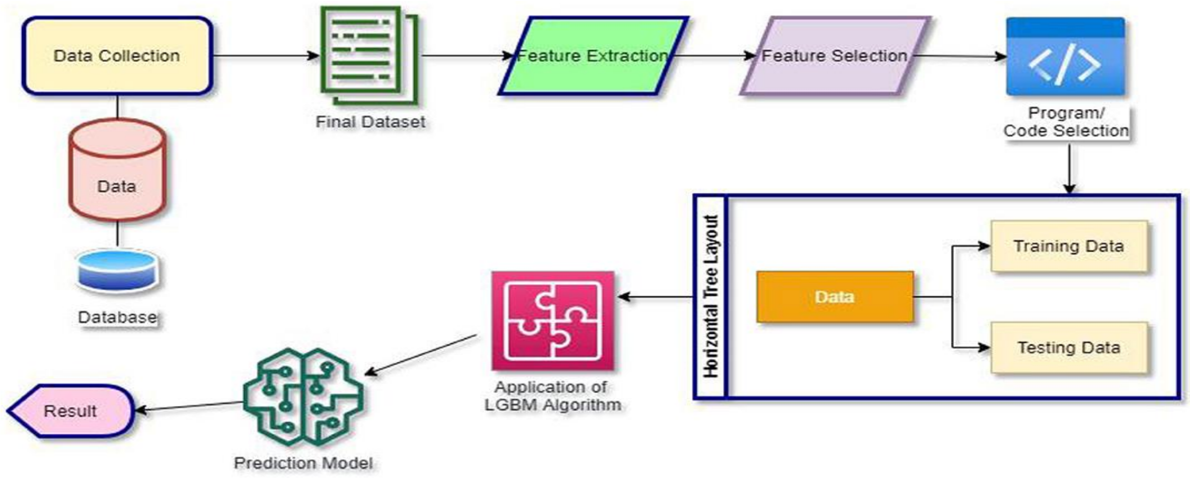


FIG: 4.1 ARCHITECTURE-1

Upon achieving satisfactory performance, the model is ready for deployment. Deployment involves integrating the trained model into a real-world environment where it can make predictions on new, unseen data. This integration could take the form of embedding the model in a web application, mobile app, or any other system relevant to the user's needs. Continuous monitoring of the model's performance is crucial post-deployment, and periodic updates and retraining may be necessary to ensure its relevance and accuracy over time.

Optionally, a user interface may be developed to facilitate user interaction with the model. This interface could be a web-based platform where users input their health parameters to receive predictions, enhancing the accessibility and usability of the diabetes prediction system.

Throughout this architectural process, considerations for privacy, ethics, and regulatory compliance are paramount, particularly in healthcare applications where sensitive patient data is involved. The comprehensive architecture outlined ensures a systematic and effective approach to developing, evaluating, and deploying a machine learning model for diabetes prediction.

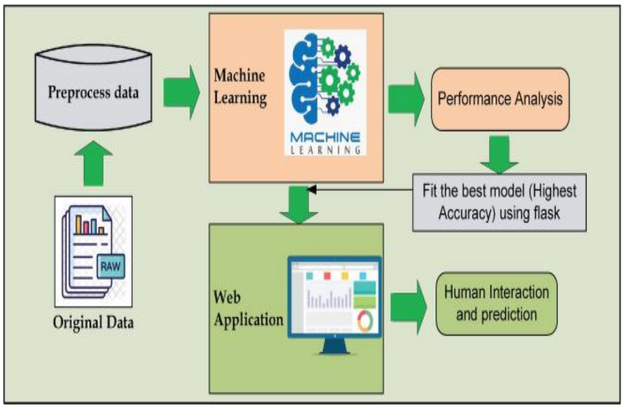


FIG: 4.1 ARCHITECTURE-2

## 4.2 USE CASE DIAGRAM:

A use case diagram for a diabetes prediction system using machine learning provides a visual representation of the interactions between different actors and the system itself. In this scenario, the primary actors are the "User" and the "Machine Learning Model," and the diagram illustrates how they collaborate to achieve the goal of predicting diabetes.

The "User" initiates the process by interacting with the system, typically through a user interface. This interface allows the user to input relevant health parameters, such as age, BMI, blood pressure, and insulin levels. The user is the driving force behind the prediction task, seeking insights into their potential risk of developing diabetes.

On the other side, the "Machine Learning Model" serves as the intelligent component of the system. It takes the input provided by the user and utilizes the trained algorithm to make predictions regarding the likelihood of diabetes based on the input features. The model processes the information and produces an output, which is the predicted result—indicating whether the user is likely to have diabetes or not.

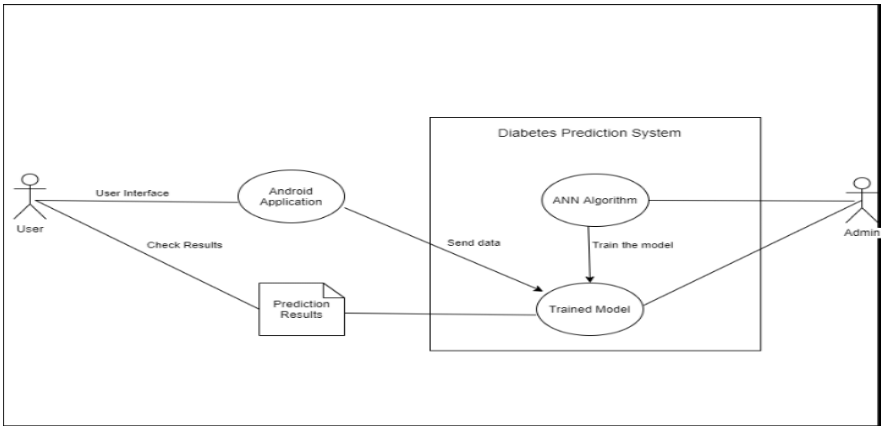


FIG: 4.2 USE CASE DIAGRAM

The arrows in the use case diagram represent the flow of information and actions. The user interacts with the system by providing input, triggering the prediction process. The machine learning model, as the core of the system, analyzes the input data and produces an output, which is then presented back to the user.

This use case diagram encapsulates the essential interactions and functionalities of a diabetes prediction system. It clarifies the roles of the user and the machine learning model, emphasizing their collaboration in the predictive analysis. The simplicity of the diagram makes it a valuable tool for stakeholders to understand the system's behavior at a high level, fostering effective communication and alignment of expectation.

# CHAPTER 5

# FUTURE ENHANCEMENT CONCLUSION

## 5.1 FUTURE ENHANCEMENT:

- Integration with Electronic Health Records (EHRs): Connect the prediction system with electronic health records to access patients' medical histories, laboratory results, and prescription data. This will provide a more comprehensive view of the patient's health.
- Mobile Apps and Telemedicine Integration: Integrate the prediction system with mobile apps and telemedicine platforms to provide patients with immediate access to their risk assessment and connect them with healthcare professionals for further guidance.
- Continuous Learning: Implement a continuous learning model that adapts and improves over time as more data becomes available. Regularly update the system's algorithms and models to stay current with the latest research and trends in diabetes prediction.
- Integration with Nutritional and Exercise Apps: Connect with apps that track nutrition and exercise routines to provide comprehensive lifestyle insights for better predictions and recommendations.
- Cross-Disease Predictions: Expand the system's capabilities to predict other related conditions or comorbidities, such as cardiovascular diseases, which often co-occur with diabetes.

## 5.2 CONCLUSION:

In conclusion, the application of machine learning in diabetes prediction marks a significant advancement in healthcare, offering a proactive and personalized approach to disease management. The journey from data collection and preprocessing to model training and deployment has showcased the potential of predictive analytics in identifying individuals at risk of diabetes. The current state of diabetes prediction models provides valuable insights into health outcomes, aiding both healthcare professionals and individuals in making informed decisions.

Looking ahead, the future holds promising opportunities for further refinement and enhancement of diabetes prediction in machine learning. As advancements in technology, data integration, and algorithmic sophistication continue, there is the potential for even more accurate, interpretable, and personalized models. The incorporation of multi-modal data, real-time monitoring, and the integration of diverse AI techniques are poised to elevate the precision and scope of these models.

Ethical considerations, privacy safeguards, and user-centric design will be pivotal in ensuring the responsible development and deployment of diabetes prediction models. Collaborative efforts on a global scale, data sharing initiatives, and interoperability with existing healthcare systems will contribute to the universality and effectiveness of these predictive tools.

Ultimately, the integration of machine learning in diabetes prediction aligns with the broader shift towards preventive and personalized healthcare. By empowering individuals with early risk assessments and actionable insights, these models have the potential to reduce the burden of diabetes and improve health outcomes on a global scale. As technology continues to evolve, so does the promise of more effective and accessible tools for diabetes prediction, reaffirming the role of machine learning in shaping the future of healthcare.

## 5.3 REFERENCES:

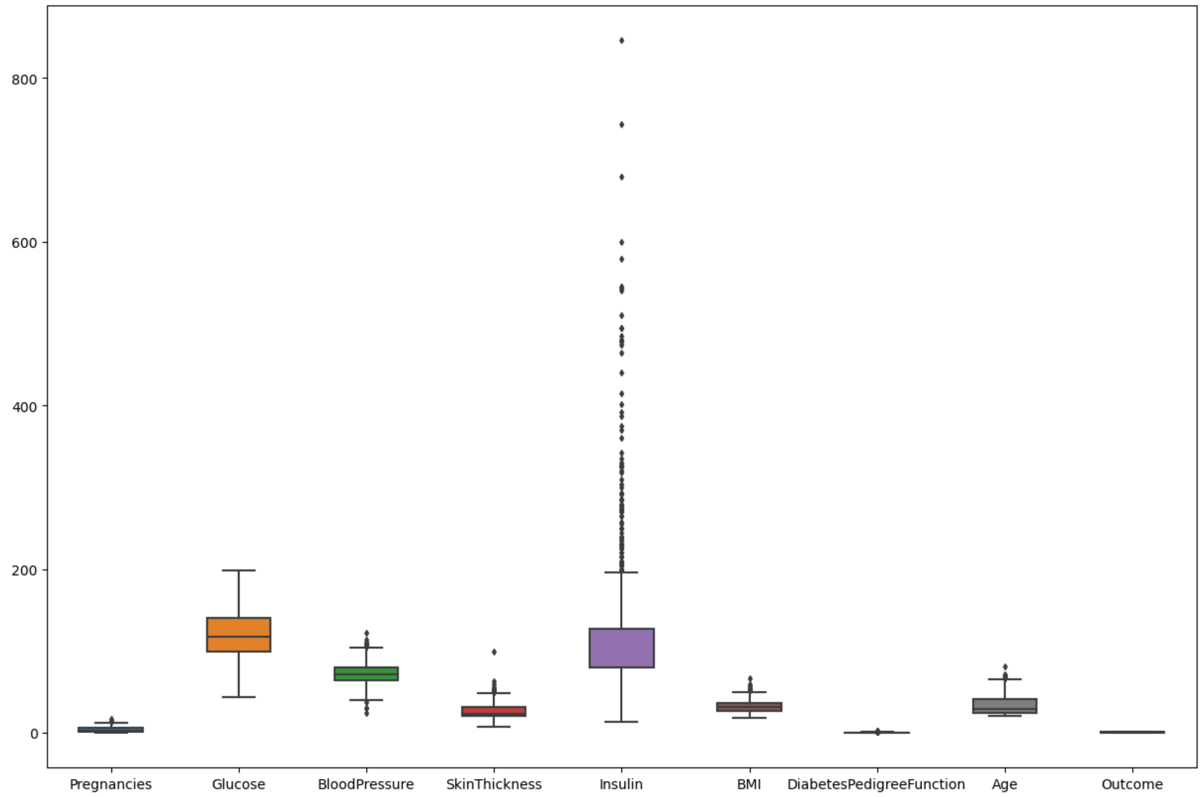
[1] Kaveeshwar, S.A., and Cornwall, J., 2014, “The current state of diabetes mellitus in India”. AMJ, 7(1), pp. 45-48. [2] Dean, L., McEntyre, J., 2004, “The Genetic Landscape of Diabetes [Internet]. Bethesda (MD): National Center for Biotechnology Information (US);. Chapter 1, Introduction to Diabetes. 2004 Jul 7. [3] Mohammed, A.K., Sateesh, K. P., Dash G. N., 2013, “A Survey of Data Mining Techniques on Medical Data for Finding Locally Frequent Diseases” International Journal of Advanced Research in Computer Science and Software Engineering, 3(8), pp. 149-153. [4] Chunhui, Z., Chengxia, Y., 2015, “Rapid Model Identification for Online Subcutaneous Glucose Concentration Prediction for New Subjects with Type I Diabetes”, IEEE Transactions on Biomedical Engineering, 62 (5), pp. 1333 – 1344 [5] Vaishali, A., Harsh, K., Anil, K.A, 2016, “Performance Analysis of the Competitive Learning Algorithms on Gaussian Data in Automatic Cluster Selection”, 2016 Second International Conference on Computational Intelligence & Communication Technology. [6] Srinivas, K., Kavihta, R.B., Govrdhan, A., 2010 “Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks” International Journal on Computer Science and Engineering”, 2(2), pp. 250-255 7] Durairaj, M., Ranjani, V., 2013, “Data Mining Applications In Healthcare Sector: A Study”, International Journal of Scientific & Technology Research, 2(10), pp. 31-35. [8] Salim, D., Suzan Mishol., Daniel, S.K., Dina M., Anael S., 2013, “Overview Applications of Data Mining in Health Care: The Case Study of Arusha Region” International Journal of Computational Engineering Research, 3(8), pp. 73 -77.

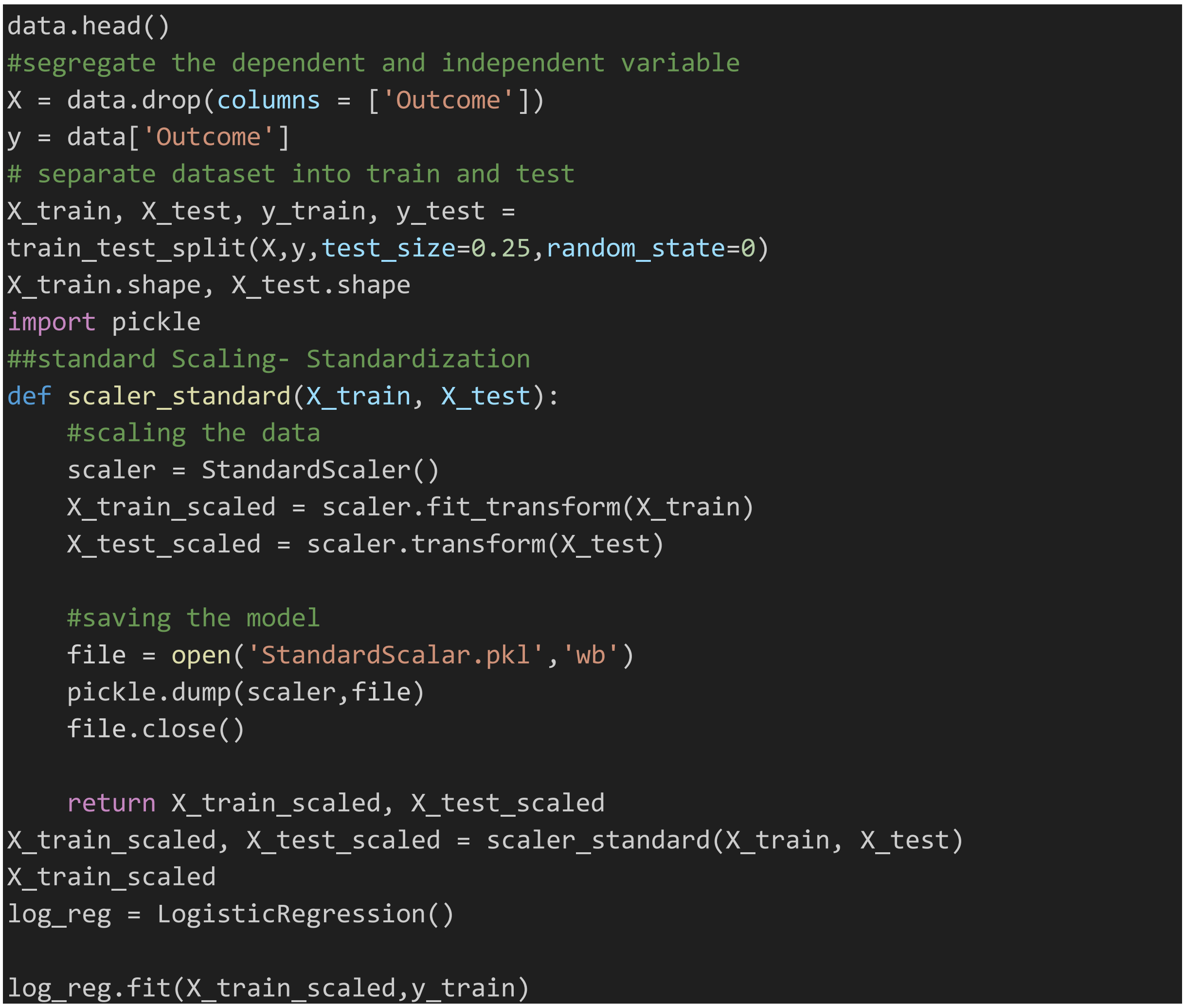
# CHAPTER 6

# IMPLEMENTATION

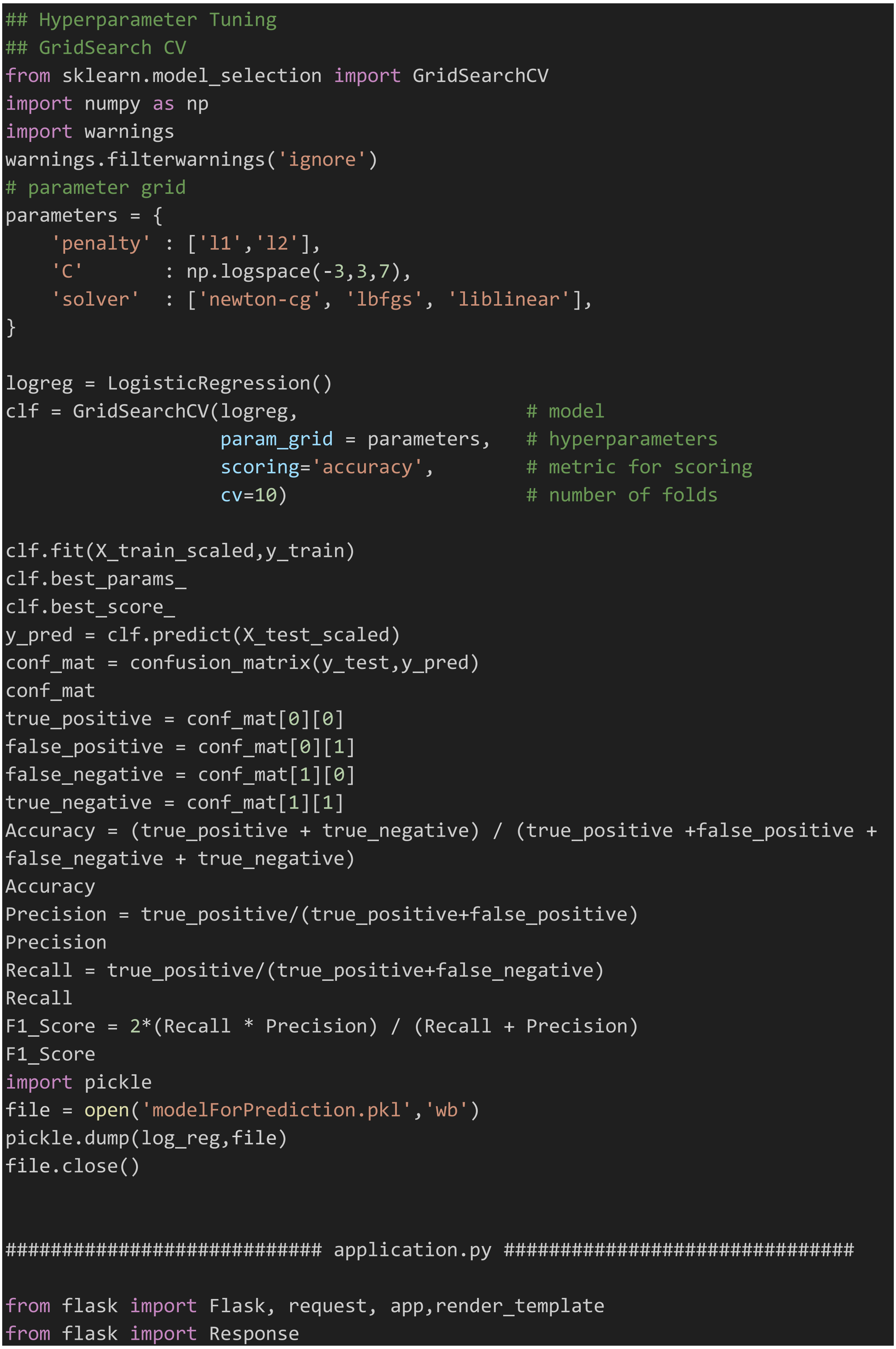
## 6.1 PARTIAL CODE:

```
#Let's start with importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model  import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as snS
#read the data file
data = pd.read_csv(r"C:\Users\admin\diabetes.csv")
data.head()
data.describe()
data.isnull().sum()
#here few misconception is there lke BMI can not be zero, BP can't be zero,
glucose, insuline can't be zero so lets try to fix it
# now replacing zero values with the mean of the column
data['BMI'] = data['BMI'].replace(0,data['BMI'].mean())
data['BloodPressure'] =
data['BloodPressure'].replace(0,data['BloodPressure'].mean())
data['Glucose'] = data['Glucose'].replace(0,data['Glucose'].mean())
data['Insulin'] = data['Insulin'].replace(0,data['Insulin'].mean())
data['SkinThickness'] =
data['SkinThickness'].replace(0,data['SkinThickness'].mean())
#now we have dealt with the 0 values and data looks better. But, there still
are outliers present in some columns.lets visualize it
fig, ax = plt.subplots(figsize=(15,10))
sns.boxplot(data=data, width= 0.5,ax=ax,  fliersize=3)
```

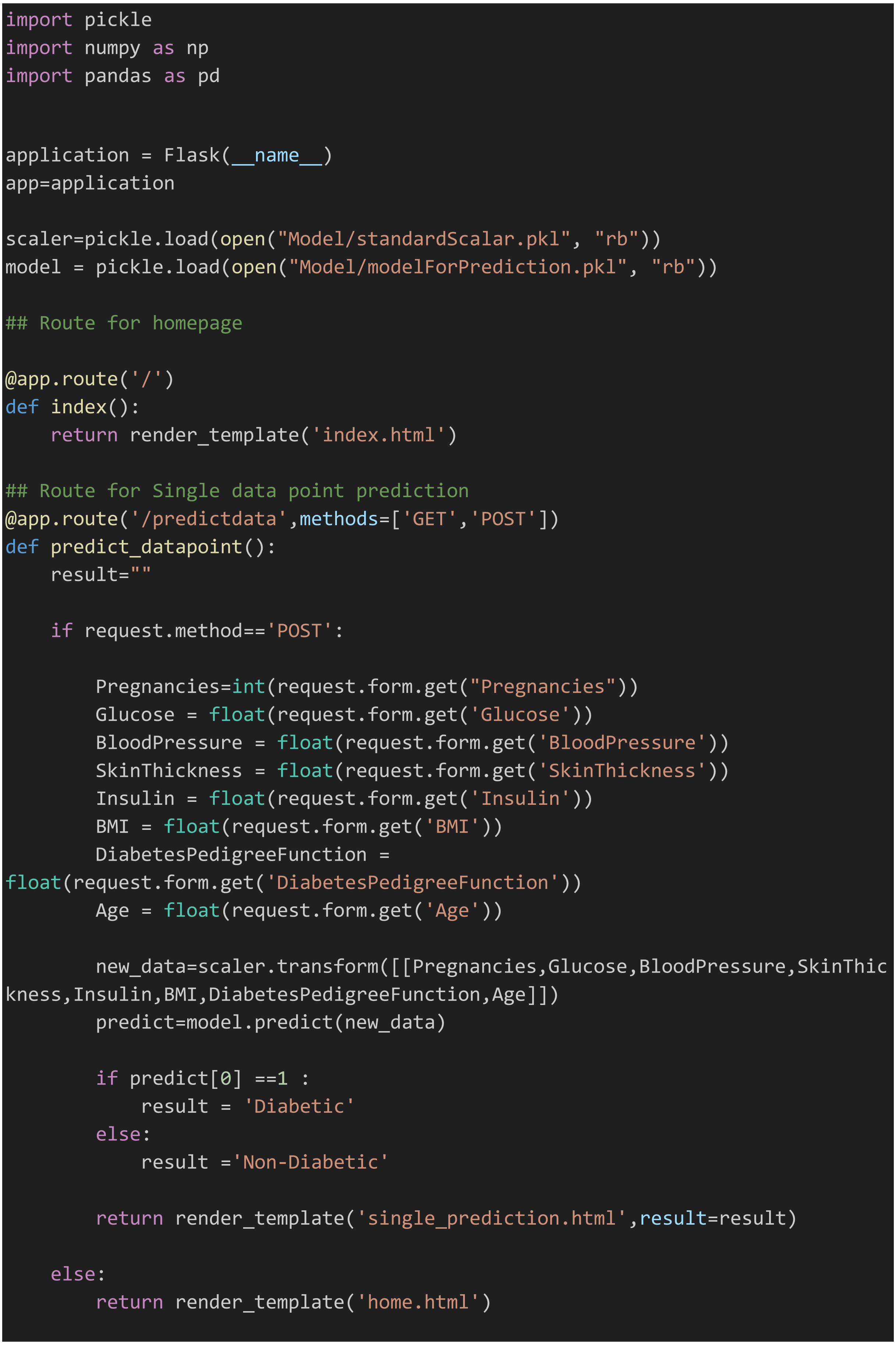




```
data.head()
#segregate the dependent and independent variable
X = data.drop(columns = ['Outcome'])
y = data['Outcome']
# separate dataset into train and test
X_train, X_test, y_train, y_test =
train_test_split(X,y,test_size=0.25,random_state=0)
X_train.shape, X_test.shape
import pickle
##standard Scaling- Standardization
def scaler_standard(X_train, X_test):
    #scaling the data
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    #saving the model
    file = open('StandardScalar.pkl','wb')
    pickle.dump(scaler,file)
    file.close()

    return X_train_scaled, X_test_scaled
X_train_scaled, X_test_scaled = scaler_standard(X_train, X_test)
X_train_scaled
log_reg = LogisticRegression()

log_reg.fit(X_train_scaled,y_train)
```

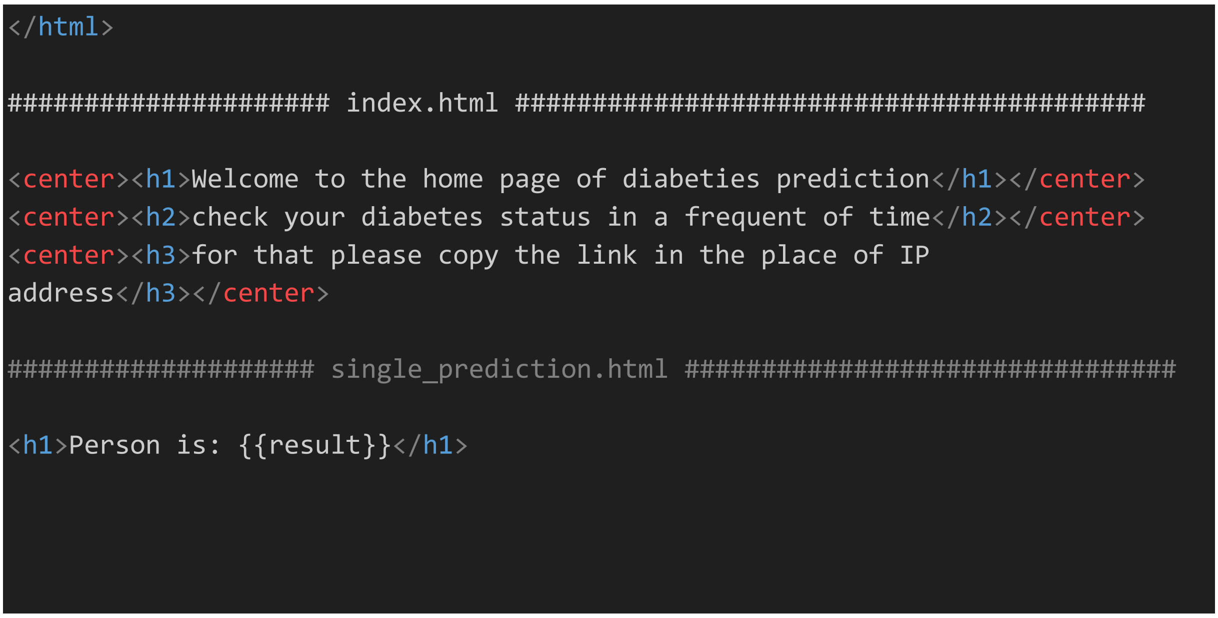


```
## Hyperparameter Tuning
## GridSearch CV
from sklearn.model_selection import GridSearchCV
import numpy as np
import warnings
warnings.filterwarnings('ignore')
# parameter grid
parameters = {
    'penalty' : ['l1','l2'],
    'C'       : np.logspace(-3,3,7),
    'solver'  : ['newton-cg', 'lbfgs', 'liblinear'],
}

logreg = LogisticRegression()
clf = GridSearchCV(logreg,                      # model
                   param_grid = parameters,     # hyperparameters
                   scoring='accuracy',          # metric for scoring
                   cv=10)                       # number of folds

clf.fit(X_train_scaled,y_train)
clf.best_params_
clf.best_score_
y_pred = clf.predict(X_test_scaled)
conf_mat = confusion_matrix(y_test,y_pred)
conf_mat
true_positive = conf_mat[0][0]
false_positive = conf_mat[0][1]
false_negative = conf_mat[1][0]
true_negative = conf_mat[1][1]
Accuracy = (true_positive + true_negative) / (true_positive +false_positive +
false_negative + true_negative)
Accuracy
Precision = true_positive/(true_positive+false_positive)
Precision
Recall = true_positive/(true_positive+false_negative)
Recall
F1_Score = 2*(Recall * Precision) / (Recall + Precision)
F1_Score
import pickle
file = open('modelForPrediction.pkl','wb')
pickle.dump(log_reg,file)
file.close()



############################ application.py ###############################

from flask import Flask, request, app,render_template
from flask import Response
```



```
import pickle
import numpy as np
import pandas as pd


application = Flask(__name__)
app=application

scaler=pickle.load(open("Model/standardScalar.pkl", "rb"))
model = pickle.load(open("Model/modelForPrediction.pkl", "rb"))

## Route for homepage

@app.route('/')
def index():
    return render_template('index.html')

## Route for Single data point prediction
@app.route('/predictdata',methods=['GET','POST'])
def predict_datapoint():
    result=""

    if request.method=='POST':

        Pregnancies=int(request.form.get("Pregnancies"))
        Glucose = float(request.form.get('Glucose'))
        BloodPressure = float(request.form.get('BloodPressure'))
        SkinThickness = float(request.form.get('SkinThickness'))
        Insulin = float(request.form.get('Insulin'))
        BMI = float(request.form.get('BMI'))
        DiabetesPedigreeFunction =
float(request.form.get('DiabetesPedigreeFunction'))
        Age = float(request.form.get('Age'))

        new_data=scaler.transform([[Pregnancies,Glucose,BloodPressure,SkinThic
kness,Insulin,BMI,DiabetesPedigreeFunction,Age]])
        predict=model.predict(new_data)

        if predict[0] ==1 :
            result = 'Diabetic'
        else:
            result ='Non-Diabetic'

        return render_template('single_prediction.html',result=result)

    else:
        return render_template('home.html')
```

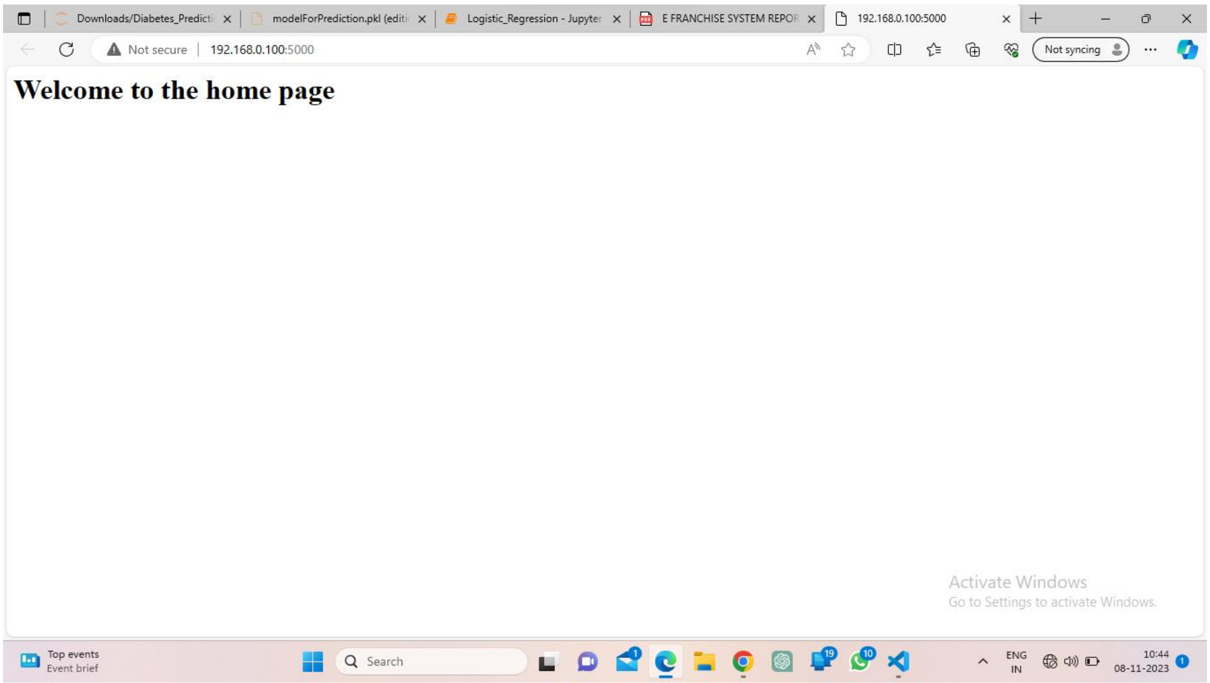
```
if __name__=="__main__":
    app.run(host="0.0.0.0")

############################# home.html #######################################

<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>

</head>

<body>
 <div class="login">
  <h1>Diabetes Prediction</h1>
    <form action="{{ url_for('predict_datapoint')}}" method="post">
      <input type="text" name="Pregnancies" placeholder="Pregnancies"
required="required" /><br>
      <input type="text" name="Glucose" placeholder="Glucose"
required="required" /><br>
        <input type="text" name="BloodPressure" placeholder="BloodPressure"
required="required" /><br>
        <input type="text" name="SkinThickness" placeholder="SkinThickness"
required="required" /><br>
        <input type="text" name="Insulin" placeholder="Insulin"
required="required" /><br>
        <input type="text" name="BMI" placeholder="BMI" required="required"
/><br>
        <input type="text" name="DiabetesPedigreeFunction"
placeholder="DiabetesPedigreeFunction" required="required" /><br>
        <input type="text" name="Age" placeholder="Age" required="required"
/><br>


        <button type="submit" class="btn btn-primary btn-block btn-
large">Predict</button>
    </form>
<body>

   <br>
   <br>


 </div>
 <!-- {{result}} -->


</body>
```



```
</html>

##################### index.html #########################################

<center><h1>Welcome to the home page of diabeties prediction</h1></center>
<center><h2>check your diabetes status in a frequent of time</h2></center>
<center><h3>for that please copy the link in the place of IP
address</h3></center>

#################### single_prediction.html ################################

<h1>Person is: {{result}}</h1>
```

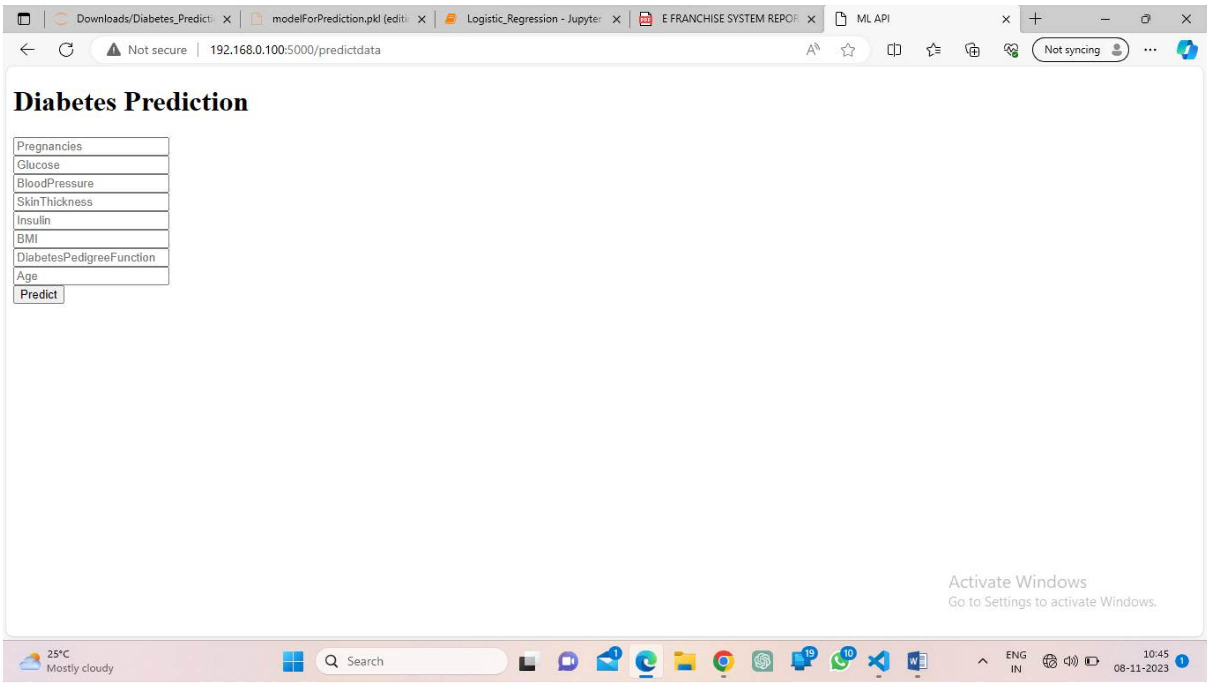
## 6.2 OUTPUTS:

INDEX.HTML



Welcome to the home page

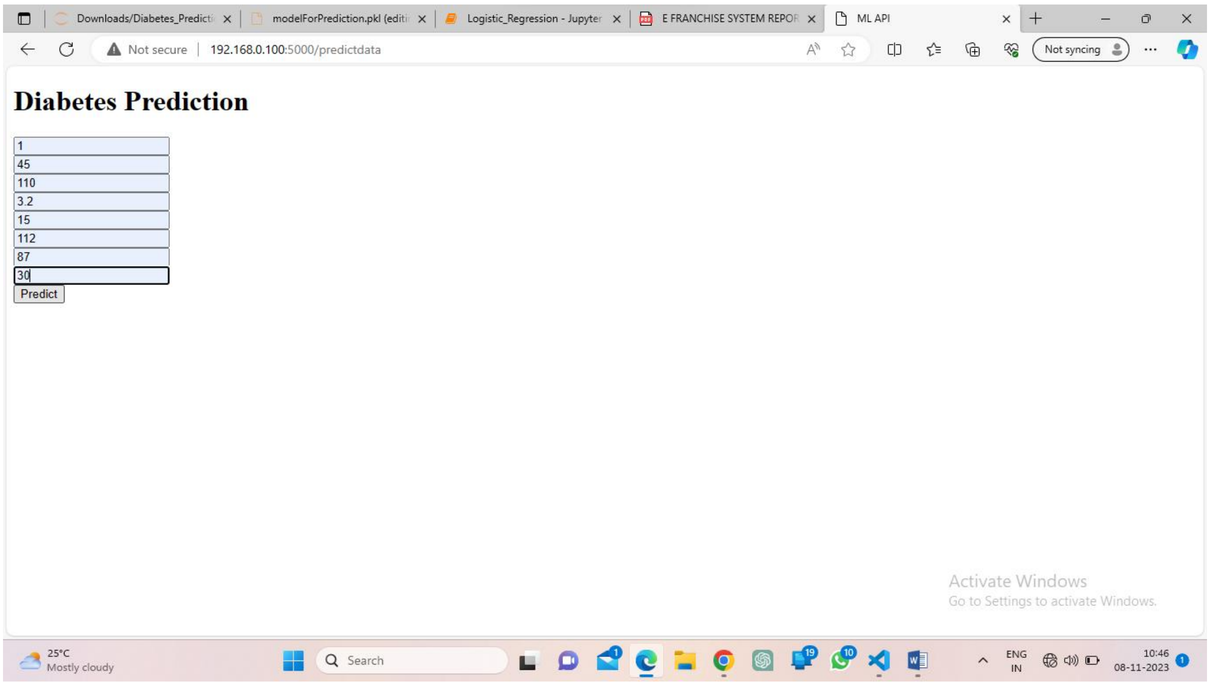
HOMEPAGE.HTML



Diabetes Prediction

Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Predict

ML API
192.168.0.100:5000/predictdata
Not secure
Activate Windows
Go to Settings to activate Windows.

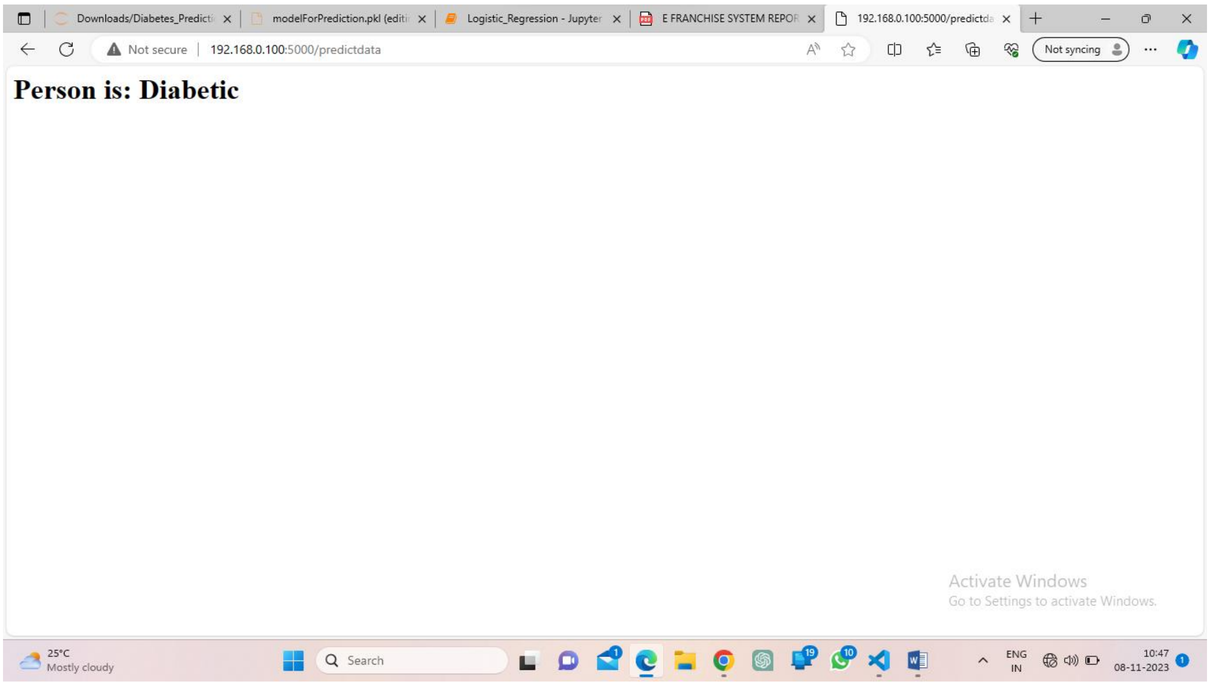

Downloads/Diabetes_Predicti
modelForPrediction.pkl (editi
Logistic_Regression - Jupyter
E FRANCHISE SYSTEM REPOR
ML API
Not secure | 192.168.0.100:5000/predictdata
Not syncing
Activate Windows
Go to Settings to activate Windows.
25°C
Mostly cloudy
Search
ENG
IN
10:46
08-11-2023

Diabetes Prediction

1
45
110
3.2
15
112
87
30

Predict

SINGLE_PREDICTION.HTML


Downloads/Diabetes_Predicti
modelForPrediction.pkl (editi
Logistic_Regression - Jupyter
E FRANCHISE SYSTEM REPOR
192.168.0.100:5000/predictda
Not secure | 192.168.0.100:5000/predictdata
Not syncing
Activate Windows
Go to Settings to activate Windows.
25°C
Mostly cloudy
Search
ENG
IN
10:47
08-11-2023

Person is: Diabetic