A PROJECT REPORT

ON

MULTIPLE DISEASE PREDICTION

Submitted by

**SANDEEP MADHUKAR (2854)**

**AKASH JAISWAL (2847)**

Under the guidance of

**MS. SHERILYN KEVIN**

Submitted in partial fulfilment of the requirements for qualifying B.Sc.-(D.S.), Semester – VI Examination



Thakur College of Science and Commerce Thakur Village, Kandivali(E), Mumbai-401101



A PROJECT REPORT ON

MULTIPLE DISEASE PREDICTION

Submitted by

**SANDEEP MADHUKAR-2854**

**AKASH JAISWAL -2847**

Under the guidance of

**MS. SHERILYN KEVIN**

Submitted in partial fulfillment of the requirements for qualifying

B.Sc. DS, Semester – VI Examination



THAKUR COLLEGE OF SCIENCE AND COMMERCE

Thakur Village, Kandivali (E). Mumbai-400101.



**Certificate**

This is to certify that the project entitled Multiple Disease Prediction is undertaken at the Thakur College of Science and Commerce by Akash Jaiswal (Roll No.- 2847) and Sandeep Madhukar (Roll No.- 2854) in partial fulfilment of BSc (DS) degree, Semester 6 Examination has not been submitted for any other examination and does not form part of any other course undergone by the candidates.

It is further certified that he/she has completed all required phases of the project.

Signature Signature

External Examiner Internal Examiner

Signature Signature

Project Guide HOD/In-charge/Co-Ordinator

College Seal

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Topic** | **Page.no** |
| 1  a  b | **INTRODUCTION**  Objective and Scope of project  Theoretical background | 6 |
| 2 | **MOTIVATION** | 9 |
| 3 | **Problem Definition** | 10 |
| 4 | **LITERATURE STUDY** | 12 |
| 5  a  b | **SOFTWARE REQUIREMENT**  Functional Requirements  Non-Functional Requirements | 13 |
| 6 | **PLANNING** | 15 |
| 7  a  b | **DESIGN**  Detail Life Cycle of The Project  Use case Diagram | 17 |
| 8 | **ALGORITHM** | 19 |
| 9  a | **IMPLEMENTATION OF DETAILS**  Details of Hardware and Software used. | 23 |
| 10  a  b | **Dataset Details**  Data Collection  Data Pre-processing | 25 |
| 11 | **Result** | 37 |
| 12 | **Performance Evaluation** | 41 |
| 13 | **Conclusion** | 45 |
| 14 | **Future Scope** | 46 |
| 15 | **Gantt Chart** | 47 |
| 16 | **Appendix** | 49 |
| 17 | **References** | 57 |

|  |  |  |
| --- | --- | --- |
| Fig.No | Figure Caption | Page.No |
| Fig.1.1 | Use case diagram | 18 |
| Fig.1.2 | Algorithm | 19 |
| Fig.1.3 | SVM Distribution | 21 |
| Fig.1.7 | Distribution of Target Variable of Heart Disease | 27 |
| Fig.1.8 | Heat map For heat disease | 28 |
| Fig.1.9 | Distribution of Target Variable of Diabetes Disease | 31 |
| Fig.1.10 | Heat Map for Diabetes Disease | 32 |
| Fig.2.1 | Heart Disease accuracy | 37 |
| Fig.2.1 | Diabetes Disease accuracy | 39 |
| Fig.2.3 | Gantt Chart | 48 |
| Fig.3.1 | User interface | 49 |
| Fig.3.3 | Basic Codes | 53 |

**ABSTRACT**

A software system is being developed that can predict multiple diseases using machine learning algorithms. The system will be able to predict heart disease, Diabetes. The system will be designed as a web application and will be implemented using Python.

The system will be used by both individuals and healthcare professionals to predict and manage the risk of disease. Individuals will be able to use the system to assess their own risk of disease, and healthcare professionals will be able to use the system to predict the risk of disease for their patients.

The system is expected to have a significant impact on healthcare by helping to detect diseases early and prevent serious complications. The system will also help to reduce the cost of healthcare by reducing the need for unnecessary tests and procedures.

# Introduction

Chronic diseases are a major public health problem, accounting for over 70% of all deaths worldwide. The two diseases that this project focuses on (heart disease, diabetes) are among the leading causes of chronic disease mortality.

Early detection and treatment of chronic diseases are essential for improving patient outcomes and reducing the burden on healthcare systems. However, many chronic diseases are asymptomatic in the early stages, making them difficult to detect early.

Machine learning is a powerful tool that can be used to predict the risk of chronic diseases. Machine learning algorithms can learn from large datasets of medical records to identify patterns that are associated with different diseases. Once a machine learning model is trained, it can be used to predict the risk of disease for new patients based on their medical history and other relevant factors.

The motivation for this project is to develop a software system that can use machine learning algorithms to predict multiple diseases simultaneously. This would allow individuals to assess their risk of multiple diseases with a single test, and it would allow healthcare professionals to more effectively manage the risk of disease for their patients.

## a) Objective and Scope of project:

* To develop machine learning models that can accurately predict the risk of multiple diseases simultaneously, including heart disease, Diabetes.

This objective involves developing machine learning models that can learn from large datasets of medical records to identify patterns that are associated with different diseases. The models should be able to predict the risk of multiple diseases simultaneously, even if the patient is not currently experiencing any symptoms.

* To develop a software system that is easy to use for both individuals and healthcare professionals, and that allows users to enter their medical history and symptoms, and to view their risk prediction results.

This objective involves developing a software system that is user-friendly and accessible to both individuals and healthcare professionals. The system should allow users to easily enter their medical history and symptoms, and to receive their risk prediction results in a clear and concise manner.

* To evaluate the accuracy and performance of the machine learning models and the software system.

This objective involves conducting rigorous evaluations of the machine learning models and the software system to ensure that they are accurate and perform as expected. The evaluations should be conducted using a variety of metrics

**Scope:**

The project will collect data from a variety of sources, such as electronic health records, medical research studies, and public health databases. The data will be preprocessed to clean it and prepare it for machine learning. This may involve tasks such as removing errors, handling missing values, and transforming the data into a format that is compatible with the chosen machine learning algorithms.

Feature engineering is the process of creating new features from the collected data that are more informative for machine learning algorithms. For example, new features could be created to represent the patient's medical history, risk factors, and lifestyle habits.

A variety of machine learning algorithms will be trained on the preprocessed data and the engineered features to predict the risk of multiple diseases simultaneously. The trained models will be evaluated on a held-out test set to assess their accuracy and performance.

## b) Theoretical background:

* Machine learning algorithms can learn from data to identify patterns that are associated with different diseases. By analyzing large datasets of medical records, machine learning algorithms can identify patterns that are predictive of different diseases. For example, a machine learning algorithm might learn that patients with high blood pressure, high cholesterol, and a history of smoking are at increased risk of heart disease.
* Machine learning algorithms can be used to predict the risk of multiple diseases simultaneously. This is because machine learning algorithms can be trained on datasets that contain information about multiple diseases. For example, a machine learning algorithm could be trained on a dataset that contains information about heart disease, diabetes. The algorithm could then be used to predict the risk of all four diseases simultaneously for a given patient.
* Machine learning algorithms can be used to develop predictive models that can be used in clinical practice. Once a machine learning algorithm has been trained on a dataset of medical records, it can be used to develop a predictive model. This predictive model can then be used to predict the risk of disease for new patients based on their medical history and other relevant factors.

The following are some of the most commonly used machine learning algorithms for disease prediction:

* Support vector machines (SVMs): SVMs are a type of supervised machine learning algorithm that can be used for classification and regression tasks. SVMs work by finding a hyperplane that separates the data into two classes with the maximum margin.
* Logistic regression: Logistic regression is a type of supervised machine learning algorithm that can be used for classification tasks. Logistic regression works by fitting a logistic function to the data to predict the probability of a given class.
* Random forests: Random forests are an ensemble learning algorithm that combines the predictions of multiple decision trees to produce a more accurate prediction. Random forests are very effective for classification and regression tasks.

## Motivation

The motivation for a multiple disease prediction system is to improve the early detection and management of chronic diseases. Chronic diseases, such as heart disease, cancer, and diabetes, are the leading cause of death worldwide. Early detection and treatment of chronic diseases can improve patient outcomes and reduce the burden on healthcare systems.

Machine learning is a powerful tool that can be used to predict the risk of chronic diseases. Machine learning algorithms can learn from large datasets of medical records to identify patterns that are associated with different diseases. Once a machine learning model is trained, it can be used to predict the risk of disease for new patients based on their medical history and other relevant factors.

A multiple disease prediction system can provide a number of benefits, including:

* Improved early detection of chronic diseases: The system can identify individuals who are at high risk of developing chronic diseases, even if they are not currently experiencing any symptoms. This early detection can lead to more timely and effective treatment, which can improve patient outcomes and reduce the risk of serious complications.
* Reduced burden on healthcare systems: The system can help to reduce the burden on healthcare systems by reducing the need for unnecessary tests and procedures. This can free up resources to be used for other patients and can help to reduce the cost of healthcare.
* Empowered patients: The system can help to empower patients by giving them more information about their health and the risk factors for different diseases. This information can help patients to make more informed decisions about their health and to take steps to reduce their risk of disease.

In addition to the above benefits, a multiple disease prediction system can also help to:

* Personalize risk reduction plans for individual patients
* Identify new biomarkers for disease prediction
* Develop new therapeutic interventions

Overall, the motivation for a multiple disease prediction system is to improve the early detection and management of chronic diseases, which can lead to better patient outcomes and a reduced burden on healthcare systems.

## Problem Definition

To develop a software system that can accurately predict the risk of multiple diseases simultaneously, including heart disease, diabetes. The system should be easy to use for both individuals and healthcare professionals, and it should be able to predict disease risk even if the patient is not currently experiencing any symptoms.

The solution to this problem is to develop a machine learning model that can learn from large datasets of medical records to identify patterns that are associated with different diseases. The model should be able to predict the risk of multiple diseases simultaneously, even if the patient is not currently experiencing any symptoms. The model should then be integrated into a software system that is easy to use for both individuals and healthcare professionals.

The success of the project will be measured by the following metrics:

* Accuracy of the machine learning model in predicting the risk of multiple diseases simultaneously
* Usability and acceptability of the software system by both individuals and healthcare professionals
* Impact of the system on patient care and outcomes, such as improved early detection of chronic diseases, reduced burden on healthcare systems, and empowered patients

The main challenges in developing a multiple disease prediction system using machine learning are:

* Collecting and preprocessing a large and diverse dataset of medical records
* Developing a machine learning model that is accurate and reliable
* Integrating the machine learning model into a software system that is easy to use and accessible to both individuals and healthcare professionals
* Evaluating the impact of the system on patient care and outcomes

Despite these challenges, the development of a multiple disease prediction system using machine learning has the potential to make a significant impact on the early detection and management of chronic diseases.

In addition to the above, the following are some other considerations for the problem definition:

* The types of diseases to be predicted
* The types of data to be used
* The types of machine learning algorithms to be used
* The target users of the system
* The deployment environment

**LITERATURE SURVEY**

It's true that machine learning and artificial intelligence have become integral parts of various industries, including the medical industry. Predictive models based on machine learning algorithms can help detect diseases accurately and quickly, allowing doctors to provide better treatment and care to patients. Your project to detect multiple diseases such as heart disease, liver disease, and diabetes using machine learning algorithms is a great initiative.

Using algorithms such as Logistic regression and K nearest neighbor (KNN) can help achieve maximum accuracy and improve the overall effectiveness of the predictive model. However, it's important to note that machine learning models are not always perfect and may have limitations.

It's important to validate the accuracy of the model using real-world data and to have a medical expert validate the results to ensure the safety and well-being of patients.

Overall, the use of machine learning and artificial intelligence in the medical industry has great potential and can lead to significant advancements in healthcare.

It's great to see that you are proposing a system that can predict multiple diseases using machine learning algorithms. This system has the potential to improve the efficiency and accuracy of disease prediction as well as help doctors provide better care to their patients.

By using machine learning algorithms and Tensor Flow, you can train models that can analyze multiple diseases simultaneously. You can also use the Flask API to create a web service that can receive inputs from users, such as the disease parameters and the disease name, and then invoke the corresponding model to predict the disease status.

The use of machine learning in this system has several benefits, including faster and more accurate disease prediction, early warning of potential health risks, and improved patient outcomes. It's also important to note that this system can be expanded to include other diseases in the future, which can further improve its utility and effectiveness.

Overall, this system has the potential to revolutionize the way we diagnose and treat diseases and can help save countless lives by detecting diseases early and providing timely treatment.

The use of computer-based technology in the healthcare industry has led to the accumulation of a large amount of electronic data. This made it difficult for medical personnel to appropriately analyse symptoms and detect diseases at an early stage. However, supervised machine learning algorithms have demonstrated significant promise for outperforming current illness diagnosis methods and supporting medical professionals in the early identification of high-risk disorders. Through the analysis of performance measures, this literature review sought to uncover trends in the use of supervised machine learning models for illness identification. Naive Bayes, Decision Trees, and K-Nearest Neighbor were the supervised machine learning algorithms that received the most attention. According to the results, support vector machines are the best at spotting Parkinson's illness and kidney disorders.

## Users requirement/SRS

Understanding the needs and expectations of our users is paramount in designing an effective Multiple Disease Prediction system. User requirements for this project span diverse domains and user groups, each with unique considerations.

**Healthcare Professionals:**

1. User-Friendly Interface: Healthcare professionals require an intuitive and user-friendly interface that enables easy navigation and interaction with the system.

2. Data Input Flexibility: The system should allow healthcare professionals to input patient data.

3. Real-Time Predictions: Healthcare professionals expect real-time disease predictions to support timely decision-making during patient consultations.

**Patients:**

1. Accessible User Interface: Patients need an easy-to-use interface that allows them to input their medical data and access predictions conveniently.

2. Data Privacy Assurance: Patients must be assured of stringent data privacy measures to protect their sensitive health information.

3. Patient Education: The system should provide educational resources and explanations to help patients understand the significance of predictions and recommended actions.

4. User Feedback: A feedback mechanism should be in place for patients to report any concerns or inaccuracies in their data and predictions.

5. Alerts and Reminders: The system can offer alerts and reminders for regular check-ups, screenings, or recommended actions based on predictions.

6. Interoperability: It should be compatible with wearable health devices or apps to facilitate data input and monitoring.

### a) Functional Requirements:

Data input: The system should allow users to enter their personal medical history and symptoms. This data should be collected in a structured and consistent manner to facilitate machine learning.

Model training and evaluation: The system should be able to train and evaluate machine learning models to predict the risk of multiple diseases simultaneously. The models should be trained on a large and diverse dataset of medical records. The system should also be able to evaluate the performance of the models on a held-out test set.

Risk prediction: The system should be able to use the trained machine learning models to predict the risk of multiple diseases for new patients based on their medical history and symptoms. The risk predictions should be presented to the user in a clear and concise manner.

Output generation: The system should be able to generate reports that include the patient's risk predictions for multiple diseases, as well as information on the diseases

Integration: The system should be able to integrate with electronic health records systems to facilitate data sharing and improve the efficiency of healthcare providers.

### b) Non-Functional Requirements:

Security and privacy: The system must protect user data from unauthorized access, use, or disclosure. This includes measures such as encryption, authentication, and access control.

Scalability: The system must be able to support a large number of users and handle a high volume of data. This includes measures such as load balancing, caching, and horizontal scaling.

**Usability and accessibility:** The system must be easy to use for both individuals and healthcare professionals. This includes measures such as a user-friendly interface, clear instructions

Performance: The system must be able to generate risk predictions quickly and efficiently.

**Planning**

**1. Define the goals and scope of the project.**

The first step is to define the goals and scope of the project. This includes identifying the specific diseases that you want to predict, the data that you will use, and the desired performance metrics.

It is important to be specific when defining the goals of the project. For example, instead of saying "I want to predict chronic diseases," you should say "I want to predict the risk of heart disease, stroke, and diabetes." This will help you to choose the appropriate data and machine learning algorithms.

You should also identify the data that you will use to train and evaluate the model. The data should be clean, well-labeled, and representative of the population of interest. If you do not have access to high-quality data, you may need to collect your own data or to use a synthetic dataset.

Finally, you should define the desired performance metrics. This could include accuracy, precision, recall, and F1 score. You should choose metrics that are relevant to the specific task of predicting multiple diseases simultaneously.

**2. Collect and preprocess the data.**

The next step is to collect and preprocess the data. The data should be cleaned and transformed into a format that is compatible with the chosen machine learning algorithm.

Data cleaning may involve removing errors, handling missing values, and converting categorical variables to numerical variables. Data preprocessing may involve normalizing the data and scaling the features to a common range.

**3. Choose a machine learning algorithm.**

There are many different machine learning algorithms available, each with its own strengths and weaknesses. Some popular algorithms for multiple disease prediction include:

* Support vector machines (SVMs)
* Logistic regression
* Decision tree
* Neural networks

You should choose an algorithm that is well-suited to the specific task of predicting multiple diseases simultaneously. You may also want to consider the following factors when choosing an algorithm:

* Interpretability: How easy is it to understand how the algorithm makes predictions?
* Scalability: How well does the algorithm handle large datasets?
* Computational requirements: What are the computing resources required to train and deploy the algorithm?

**4. Train and evaluate the model.**

Once you have chosen a machine learning algorithm, you need to train a model on your data. Training a model involves feeding the algorithm the preprocessed data and allowing it to learn the patterns in the data.

Once the model is trained, you need to evaluate its performance on a held-out test set. This will help you to assess the generalization ability of the model and to identify any potential overfitting problems.

If the model is not performing well on the test set, you may need to adjust the hyperparameters of the algorithm or to try a different algorithm.

**5. Deploy the model.**

Once you are satisfied with the performance of the model, you need to deploy it so that it can be used to predict the risk of multiple diseases for new patients.

Deployment options include:

* Developing a web application
* Integrating the model

# SYSTEM ANALYSIS AND DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. It mainly involves defining and developing systems to satisfy specified requirements of the user.

## Detail Life Cycle of The Project:

Developing a project for multiple disease prediction involves several stages in its lifecycle. Below is a detailed breakdown of the key steps involved in the development process:

Data Collection: The first component of the system involves collecting a large

dataset of medical records containing patient information and various medical

features related to multiple diseases. This dataset will be used to train the machine

learning models.

2. Data Preprocessing: The collected data will be preprocessed to handle missing

values, outliers, and to perform feature scaling. This component of the system

involves cleaning and preparing the data for model training.

3. Model Training: This component involves training different machine learning

algorithms such as decision trees, random forests, and artificial neural networks

on the preprocessed data. The trained models will be used for disease prediction.

4. Model Selection: The performance of different machine learning algorithms will

be compared using metrics such as accuracy, precision, and recall, and the best-

performing model will be selected for disease prediction.

5. Model Evaluation: The selected model will be evaluated on a separate test dataset

to measure its accuracy and reliability in predicting multiple diseases. This

component of the system involves testing the model and measuring its

performance.

6. User Interface Development: The final component of the system involves

developing a user-friendly interface that allows healthcare professionals to input

patient information and receive predictions for multiple diseases. The interface

will be designed to provide an easy-to-use tool for disease prediction

**1. Data Collection:** The first component of the system involves collecting a large

dataset of medical records containing patient information and various medical

features related to multiple diseases. This dataset will be used to train the machine

learning models.

**2. Data Preprocessing:** The collected data will be preprocessed to handle missing

values, outliers, and to perform feature scaling. This component of the system

involves cleaning and preparing the data for model training.

**3. Model Training:** This component involves training different machine learning

algorithms such as decision trees, random forests, and artificial neural networks

on the preprocessed data. The trained models will be used for disease prediction.

**4. Model Selection:** The performance of different machine learning algorithms will

be compared using metrics such as accuracy, precision, and recall, and the best-

performing model will be selected for disease prediction.

**5. Model Evaluation**: The selected model will be evaluated on a separate test dataset

to measure its accuracy and reliability in predicting multiple diseases. This

component of the system involves testing the model and measuring its

performance.

**6. User Interface Development**: The final component of the system involves

developing a user-friendly interface that allows healthcare professionals to input

patient information and receive predictions for multiple diseases. The interface

will be designed to provide an easy-to-use tool for disease prediction

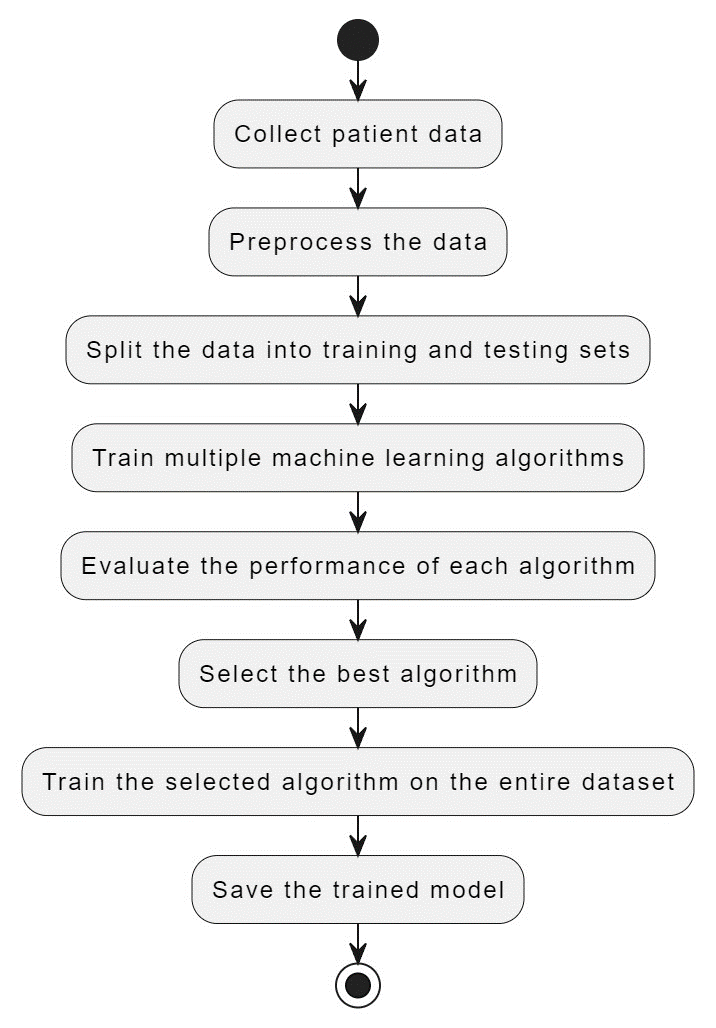


Fig.No.1.1

**Algorithms**

**K Nearest Neighbours**

K-Nearest Neighbours Classification is a type of instance-based learning or nongeneralizing learning which stores instances of the training data. It predicts a new point by finding a predefined number of training samples closest in distance.

Weights can also be given for classifying a new point. In scikit-learn, each neighbour can either be given uniform weight or can be given weights proportional to the inverse of the distance from the query point or a user defined weight.

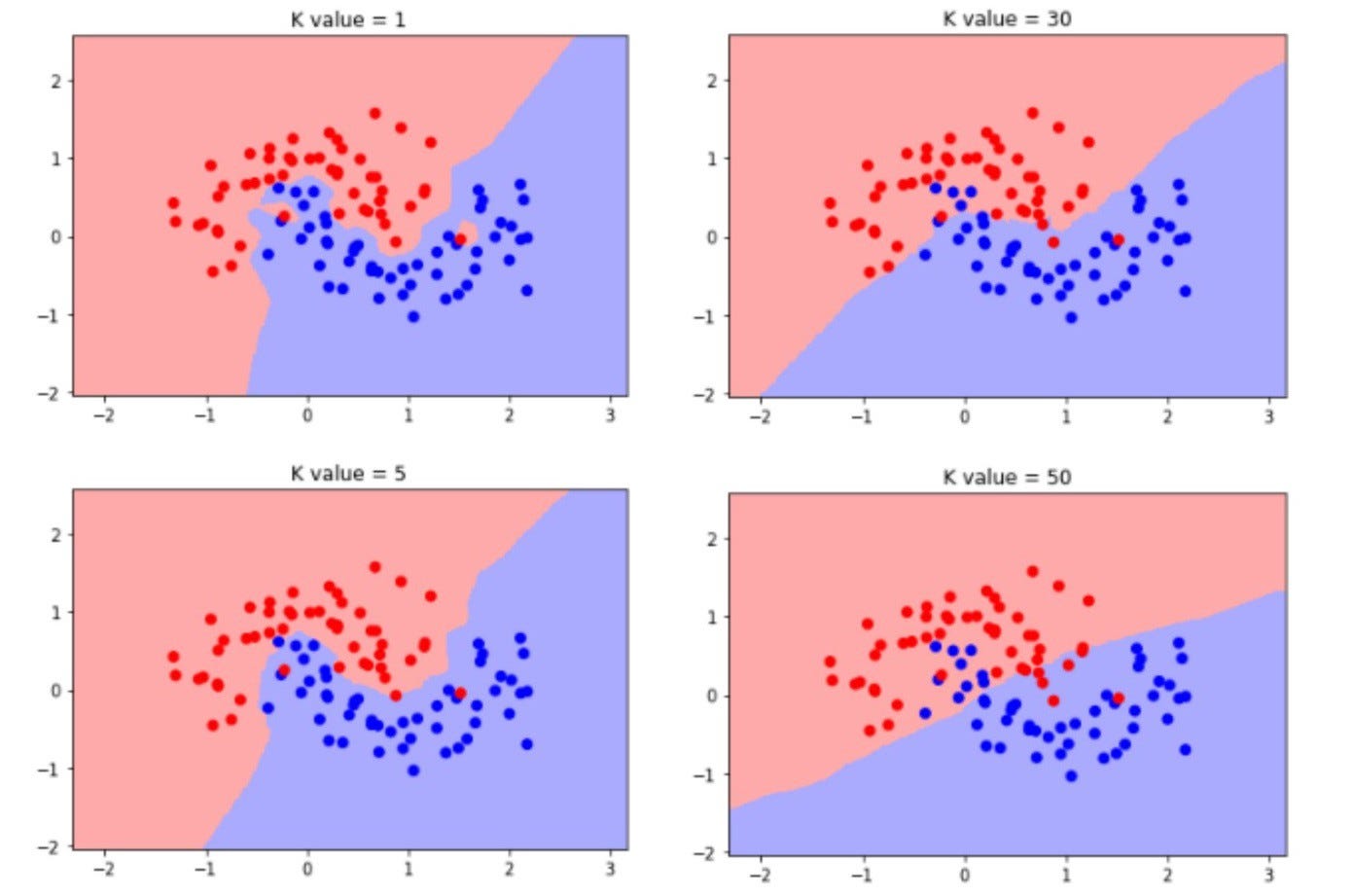


Fig.No.1.2

**Logistic regression**

Logistic regression algorithm uses the logistic function, so this algorithm is named Logistic Regression. The logistic function is an “S” shaped curve developed for statistical functionalities and the curve is plotted between 0 and 1. For the representation purpose logistic regression uses equations like linear regression.

Logistic regression Equation

Y=1/(1+EXPO(-value)) --(1)

Input values (generally termed as x) and Co-efficients(Beta) are linearly combined to predict the value of output(termed as y).

Logistic regreesion Equation

y=EXPO(u0+u1\*x)/(1+EXPO(u0+u1\*x)). --(2)

y is predicted outcome, a0 is intercept or bias, and a1 is single input coefficient value. The models of logistic regression predict the probability of first-class (or can be termed as default class). For example, if we are building a model for predicting the gender of a person using the height and the default class might be male which can be formally written as

P(gender=male|height). --(3)

For prediction probabilities must turn into binary values, either 0 or 1.Probabilities are turned into predictions by using the logistic function. The model can be composed as

y=EXPO(u0+u1\*x)/(1+EXPO(u0+u1\*x)). --(4)

Further solving we get the equation as:

ln(p(x)/1-p(x)) =u0+u1\*x.--(5)

The left-hand size equation(ratio) is called odds of first-class or default class. The odds are calculated as the probability of an event divided by the probability of its complement event.

ln(odds)=u0+u1\*x. --(6)

Predictions with logistic regression are quite easy and simple to implement. For example, let us assume finding the gender based on the height of a person, let us consider the height as 150, and assuming coefficients u0=-100 and u1=0.6.

y = EXPO(u0 + u1\*X) / (1 + EXPO(u0 + u1\*X)) -(7)

y = EXPO(-100 + 0.6\*150) / (1 + EXPO(-100 + 0.6\*150)) --(8)

y = 0.0000453 --(9)

The likelihood that got almost zero is male. Continuously practice, we utilize the specific probabilities.

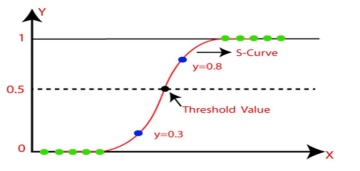


Fig.No.1.3

**Support Vector Classifier:**

Support vector machine is supervised machine learning algorithm used for classification as well as regression. Most often it is well suited for classification purpose.

The objective of SVM is to find the hyperplane in N-dimensional Space that distinctly classifies the data points. Dimension of the hyperplane is depending upon number of features.

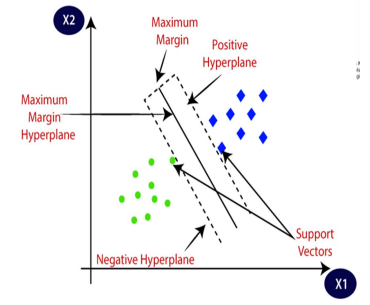


Fig.No.1.4

We choose the hyperplane whose distance from it to nearest data point on each side is maximized.

This hyperplane is known as maximum margin hyperplane. Hyperplane are decision boundaries that help in classifying that help in classifying the data points. Support vectors are datapoints that are closer to hyperplane.

**Decision Tree Classifier:**

Decision tree algorithm is supervised machine learning algorithm used for regression as well as classification. Here we used it for classification purpose.

a) Step 1 – Start with root node to form a tree T containing all the datapoints in dataset.

b) Step 2 – If the node purity is below the purity threshold or not all the records of s belong to class C, then use the purity information attribute to split the node. This create sub – tree.

c) Step 3 – Repeat step 2 until

• All the leaf node satisfy minimum purity threshold

• The tree cannot be further split

• Any other stopping criteria is reached (such as maximum tree depth desire)

**Implementation Details**

The proposed methodology for this project involves utilizing multiple training models for disease prediction, comparing their performance, and implementing the model, which achieved a high accuracy . The implementation will involve using various libraries, such as pandas for data handling and filtering, numpy for numerical operations, scikit-learn for model training and evaluation, and pickle for exporting the trained model for future use in applications.

• **Data Handling and Filtering:** The first step in the project implementation is to handle and filter the data using the pandas library. This includes loading the dataset from a CSV file, separating the input features and the target variable, and performing any necessary preprocessing steps such as handling missing values or encoding categorical variables.

**• Model Selection and Comparison:** Next, different training models will be selected and trained on the preprocessed dataset. Models such as k-nearest neighbors (KNN),logistic regression,Support vector machine(SVM) and random forest will be considered. Each model will be evaluated using appropriate metrics like accuracy, precision, recall, and F1 score. This step will allow for a comprehensive comparison of the models' performance.

**• Model Training:** Based on the comparison results, the model, which achieved the highest accuracy , will be selected for further implementation. The model will be instantiated with the appropriate hyperparameters, such as the choice of kernel and regularization parameter, to ensure optimal performance.

**• Model Evaluation and Fine-tuning:** The trained model will be evaluated on a separate test dataset to assess its generalization ability. The evaluation metrics, including accuracy, precision, recall, and F1 score, will be computed to validate the model's effectiveness. If necessary, the model hyperparameters will be fine-tuned using techniques like grid search or cross-validation to optimize its performance.

• **Exporting the Trained Model:** Once the model is trained and fine-tuned, it will be exported using the pickle library. This will allow the model to be saved in a serialized format and used in future applications without the need for retraining. The exported model can be loaded and used to make predictions on new data points, enabling disease prediction in real-world scenarios.

**• Integration with Application:** The final step of the implementation involves integrating the trained model into an application or system for practical use. The model can be incorporated into a user-friendly interface or an API, where new data can be input, and disease predictions can be obtained. This integration will enable the model to be utilized by healthcare professionals, researchers, or individuals for disease risk assessment and decision-making. In summary, the proposed methodology for this project involves comparing multiple training models, selecting the model based on its high accuracy, implementing the model using libraries such as pandas, numpy, scikit-learn, and pickle, and integrating the trained model into an application for disease prediction. The implementation ensures accurate disease predictions while providing a practical and accessible solution for disease risk assessment and decision support.

## Details of Hardware and Software used:

Multiple Disease prediction using machine learning

8

Department of CSE, JNTUACE, Kalikiri

3.4 REQUIREMENTS

A software requirements specification (SRS) is a description of a software

system to be developed, its defined after business requirements specification

(CONOPS) also called stakeholder requirements specification (STRS) other document

related is the system requirements specification (SYRS).

3.4.1 HARDWARE AND SOFTWARE REQUIREMENTS

All computer software needs certain hardware components or other software

resources to be present on a computer. These prerequisites are known as (computer)

system requirements and are often used as a guideline as opposed to an absolute rule.

Most software defines two sets of system requirements: minimum and recommended.

With increasing demand for higher processing power and resources in newer versions

of software, system requirements tend to increase over time. Industry analysts suggest

that this trend plays a bigger part in driving upgrades to existing computer systems than

technological advancements. A second meaning of the term of System requirements is

a generalization of this first definition, giving the requirements to be met in the design

of a system or sub-system.

HARDWARE REQUIREMENTS

▪ System processor : Intel Core i7.

▪ Hard Disk : 512 SSD.

▪ Monitor : “15” LED.

▪ Mouse : Optical Mouse.

▪ RAM : 8.0 GB.

▪ Key Board : Standard Windows Keyboard.

SOFTWARE REQUIREMENTS

▪ Operating system : Windows 10.

▪ Coding Language : Python 3.9.

▪ Front-End : Streamlit 3.7, Python

▪ Back-End : Python3.9

▪ Python Modules : Pickle 1.2.

Multiple Disease prediction using machine learning

8

Department of CSE, JNTUACE, Kalikiri

3.4 REQUIREMENTS

A software requirements specification (SRS) is a description of a software

system to be developed, its defined after business requirements specification

(CONOPS) also called stakeholder requirements specification (STRS) other document

related is the system requirements specification (SYRS).

3.4.1 HARDWARE AND SOFTWARE REQUIREMENTS

All computer software needs certain hardware components or other software

resources to be present on a computer. These prerequisites are known as (computer)

system requirements and are often used as a guideline as opposed to an absolute rule.

Most software defines two sets of system requirements: minimum and recommended.

With increasing demand for higher processing power and resources in newer versions

of software, system requirements tend to increase over time. Industry analysts suggest

that this trend plays a bigger part in driving upgrades to existing computer systems than

technological advancements. A second meaning of the term of System requirements is

a generalization of this first definition, giving the requirements to be met in the design

of a system or sub-system.

HARDWARE REQUIREMENTS

▪ System processor : Intel Core i7.

▪ Hard Disk : 512 SSD.

▪ Monitor : “15” LED.

▪ Mouse : Optical Mouse.

▪ RAM : 8.0 GB.

▪ Key Board : Standard Windows Keyboard.

SOFTWARE REQUIREMENTS

▪ Operating system : Windows 10.

▪ Coding Language : Python 3.9.

▪ Front-End : Streamlit 3.7, Python

▪ Back-End : Python3.9

▪ Python Modules : Pickle 1.2.3

Multiple Disease prediction using machine learning

8

Department of CSE, JNTUACE, Kalikiri

3.4 REQUIREMENTS

A software requirements specification (SRS) is a description of a software

system to be developed, its defined after business requirements specification

(CONOPS) also called stakeholder requirements specification (STRS) other document

related is the system requirements specification (SYRS).

3.4.1 HARDWARE AND SOFTWARE REQUIREMENTS

All computer software needs certain hardware components or other software

resources to be present on a computer. These prerequisites are known as (computer)

system requirements and are often used as a guideline as opposed to an absolute rule.

Most software defines two sets of system requirements: minimum and recommended.

With increasing demand for higher processing power and resources in newer versions

of software, system requirements tend to increase over time. Industry analysts suggest

that this trend plays a bigger part in driving upgrades to existing computer systems than

technological advancements. A second meaning of the term of System requirements is

a generalization of this first definition, giving the requirements to be met in the design

of a system or sub-system.

HARDWARE REQUIREMENTS

▪ System processor : Intel Core i7.

▪ Hard Disk : 512 SSD.

▪ Monitor : “15” LED.

▪ Mouse : Optical Mouse.

▪ RAM : 8.0 GB.

▪ Key Board : Standard Windows Keyboard.

SOFTWARE REQUIREMENTS

▪ Operating system : Windows 10.

▪ Coding Language : Python 3.9.

▪ Front-End : Streamlit 3.7, Python

▪ Back-End : Python3.9

▪ Python Modules : Pickle 1.2.3

All computer software needs certain hardware components or other software

resources to be present on a computer. These prerequisites are known as (computer)

system requirements and are often used as a guideline as opposed to an absolute rule.

Most software defines two sets of system requirements: minimum and recommended.

With increasing demand for higher processing power and resources in newer versions

of software, system requirements tend to increase over time. Industry analysts suggest

that this trend plays a bigger part in driving upgrades to existing computer systems than

technological advancements. A second meaning of the term of System requirements is

a generalization of this first definition, giving the requirements to be met in the design

of a system or sub-system

All computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of System requirements is a generalization of this first definition, giving the requirements to be met in the design of a system or sub-system.

**HARDWARE REQUIREMENTS**

▪ System processor : Intel Core i7.

▪ Hard Disk : 512 SSD.

▪ Monitor : “15” LED.

▪ Mouse : Optical Mouse.

▪ RAM : 8.0 GB.

▪ Key Board : Standard Windows Keyboard.

**SOFTWARE REQUIREMENTS**

▪ Operating system : Windows 10.

▪ Coding Language : Python 3.9.

▪ Front-End : Streamlit 3.7, Python

▪ Back-End : Python3.9

▪ Python Modules : Pickle 1.2.3

**DATASET DETAILS**

**DATASET NAMES**

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Data sets can also consist of a collection of documents or files. The sources of the datasets are from Kaggle.com.

The datasets that are used are:

• Heart disease dataset – (heart\_disease.csv)

• Diabetes disease dataset – (diabetes\_disease.csv)

**HEART DISEASE DATASET DETAILS**

The heart disease datasets consists of 303 rows 14 columns.

The fields: age, sex, cp, trestbp,s chol, fbs, restecg, thalach, exan,g oldpeak, slope, ca, thal.

We have taken the 14th column as the target variable (target).

**Heart disease data set fields**.

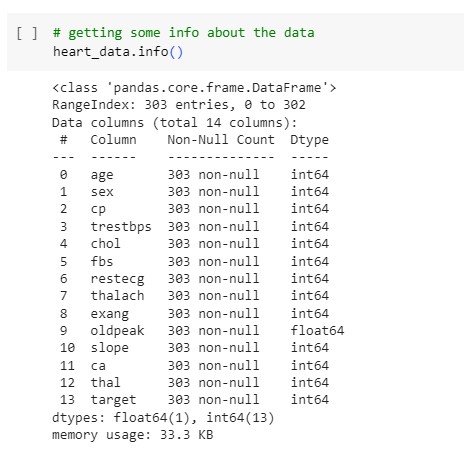


Fig.No.1.5

**DATA SET VISUALIZATION**

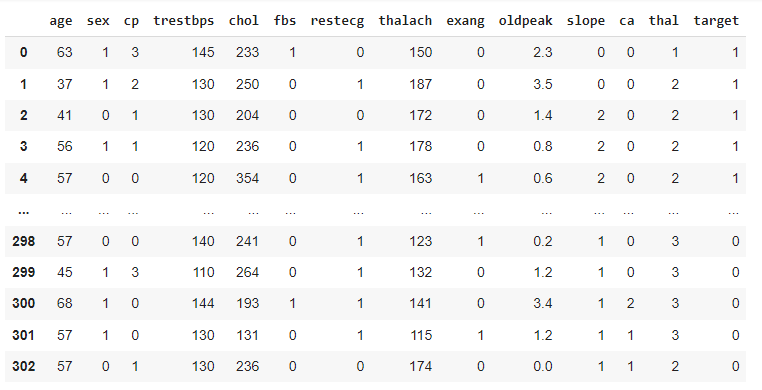


Fig.No1.6

**Distribution of Target Variable of Heart Disease**

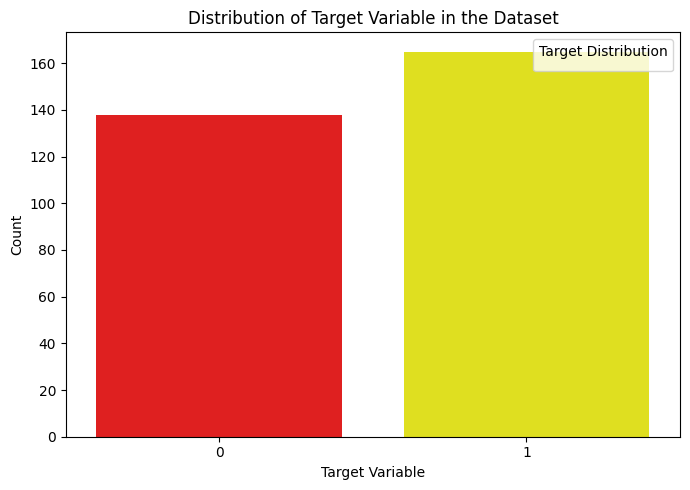


Fig.No.1.7

**Information**:

* **Data**: The graph appears to be a bar chart representing the distribution of a target variable in a heart disease dataset.
* **Target Variable**: The target variable likely indicates the presence or absence of heart disease. In this case, it's assumed that:
  + 0 represents no heart disease.
  + 1 represents heart disease.
* **X-axis**: The x-axis likely represents the target variable categories (0 or 1 for no heart disease or heart disease, respectively).
* **Y-axis**: The y-axis likely represents the count or frequency of observations in each category.

**Heat Map For Heart Disease**

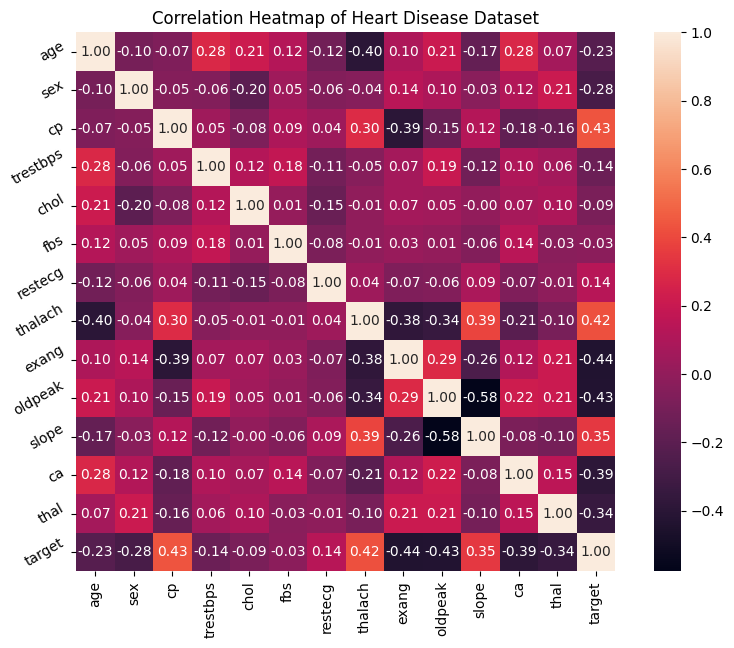
****

Fig.No.1.8

**Correlation Coefficient**:

The correlation coefficient (values between -1 and +1) measures the strength and direction of the linear relationship between two variables.

**Positive Correlation (0 to +1):** As the value of one variable increases, the value of the other variable tends to increase as well (stronger positive correlation closer to +1, weaker positive correlation closer to 0).

**Negative Correlation (-1 to 0):** As the value of one variable increases, the value of the other variable tends to decrease (stronger negative correlation closer to -1, weaker negative correlation closer to 0).

**Zero Correlation (close to 0):** There is no significant linear relationship between the two variables.

**Interpreting the Correlations:**

Here's a breakdown of some key correlations (focusing on values with a stronger absolute value):

**target vs. cp (chest pain type):** Positive correlation (0.4338) suggests that individuals with chest pain (higher cp values) are more likely to have heart disease (target = 1).

**target vs. thalach (maximum heart rate):** Positive correlation (0.4217) indicates a potential link between higher maximum heart rate and the presence of heart disease.

**target vs. slope (the slope of the ST segment):** Positive correlation (0.3459) suggests a possible relationship between an upward-sloping ST segment and increased risk of heart disease.

**target vs. restecg (resting ECG result):** Weak positive correlation (0.1372) might indicate a slight association between abnormal resting ECG and heart disease.

**target vs. age:** Negative correlation (-0.2254) suggests that younger individuals in the dataset tend to have a lower prevalence of heart disease.

**target vs. sex:** Negative correlation (-0.2809) implies that females in the dataset might have a lower risk of heart disease compared to males (depending on the specific coding of the sex variable).

**target vs. thal (number of major vessels colored by fluoroscopy):** Negative correlation (-0.3440) suggests a potential link between fewer major vessels being colored (indicating blockage) and the presence of heart disease.

**target vs. ca (number of major vessels with fluoroscopy):** Negative correlation (-0.3917) is similar to the correlation with thal, suggesting a possible relationship between fewer major vessels detected and heart disease.

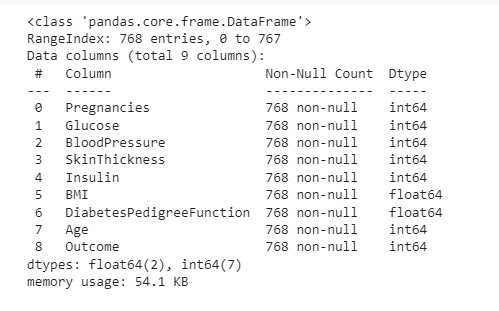
**target vs. oldpeak (ST segment depression):** Negative correlation (-0.4307) indicates that higher ST segment depression might be associated with a higher risk of heart disease.

**target vs. exang (exercise induced angina)**: Negative correlation (-0.4368) suggests a potential link between experiencing angina during exercise and increased risk of heart disease.

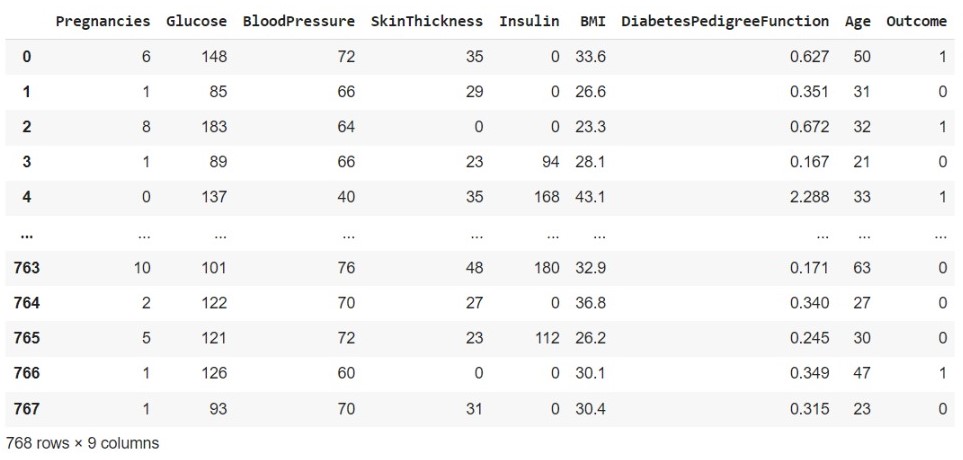
**DIABETES DISEASE DATASET DETAILS**

The diabetes disease dataset consists of 768 rows, 9 columns Column field names : Pregnancies, Glucose , BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age.The last column is the target (outcome).

**Diabetes disease data set fields.**



**DATA SET VISUALIZATION**



**Distribution of Target Variable of Diabetes Disease**

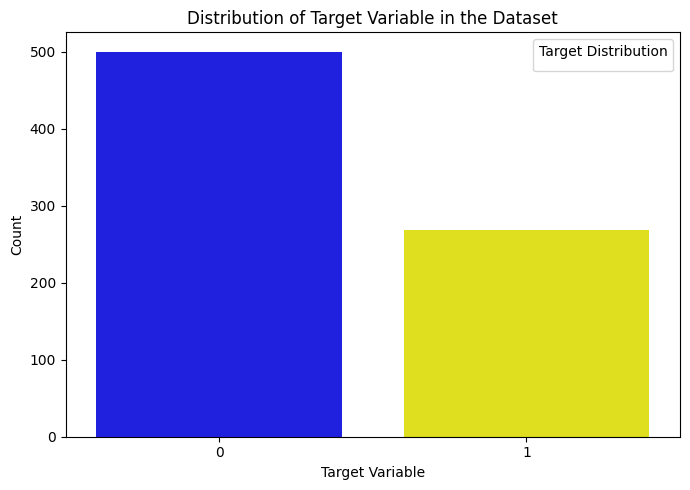
****

Fig.1.9

**Information:**

The graph you provided is a bar chart representing the distribution of the target variable in a diabetes disease dataset. The target variable indicates the presence or absence of diabetes, where:

* **0** denotes **no diabetes**.
* **1** denotes **diabetes**.

**Distribution:**

The x-axis represents the target variable categories (0 or 1 for no diabetes or diabetes, respectively). The y-axis represents the count or frequency of individuals in each category.

**Key Finding:**The bar chart showcases an imbalanced distribution of the target variable in the diabetes dataset. Here's a breakdown of the observations:

* 500 individuals do not have diabetes (target variable = 0). This value is represented by the taller bar on the left side of the chart.
* 268 individuals have diabetes (target variable = 1). This value is represented by the shorter bar on the right side of the chart.

**Heat Map For Diabetes Disease**

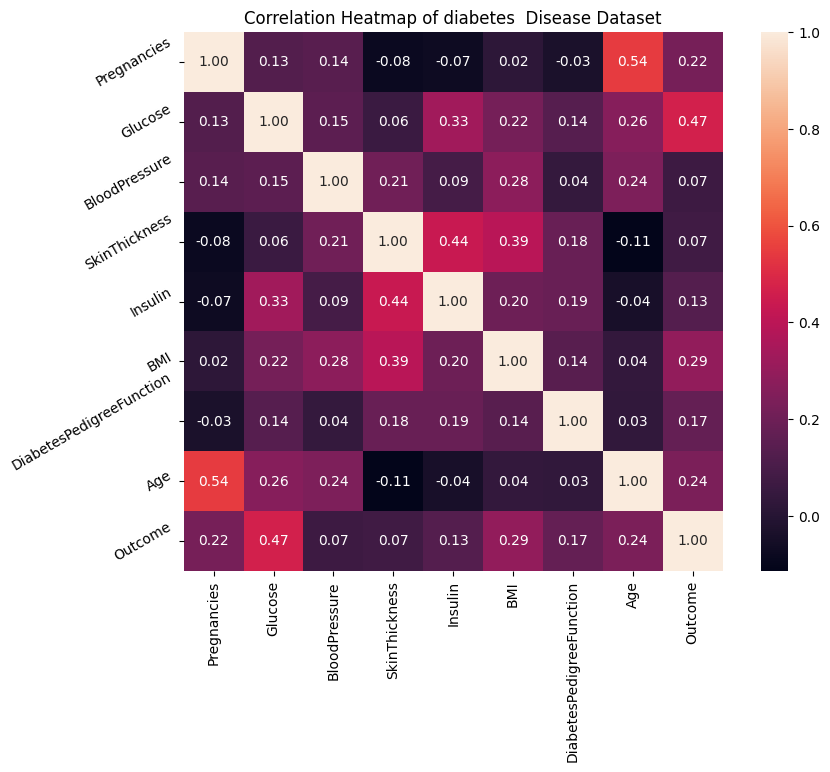


Fig.No.1.10

**Correlation Coefficients:**

* **Positive Correlation with Outcome (>= 0.2):**
  + **Glucose:** Higher blood sugar levels might be associated with an increased risk of diabetes (Outcome = 1).
  + **BMI:** Higher body mass index could be a risk factor for diabetes.
  + **Age:** Ageing is a known risk factor for developing diabetes.
  + **Pregnancies:** The relationship between the number of pregnancies and diabetes is complex and may depend on other factors.
  + **Diabetes Pedigree Function:** This is a score based on family history of diabetes. A higher score suggests a greater genetic predisposition, potentially leading to a positive correlation.
* **Weaker Correlations (between 0.1 and 0.2, positive or negative):**
  + **Insulin:** This can be complex. People with diabetes may have higher insulin levels due to their body's inability to use it effectively.
  + **Skin Thickness:** Thicker skin folds might be a risk factor.
  + **Blood Pressure:** High blood pressure is another risk factor for diabetes.
* **Near Zero Correlation (close to 0):** Without further context, it's difficult to determine the meaning of correlations close to zero.

**a) DATA COLLECTION**

Data Collection is one of the most important tasks in building a machine learning model. We collect the specific data based on requirements from users to make the dataset. The dataset contains some unwanted data also. So first we need to preprocess the data and obtain perfect data set for algorithm.

**PACKAGES IMPORTED**

• Pandas: Pandas is a software library written for python for data manipulation and analysis.

It offers data structures and operations for manipulating numerical tables and time series.

• Numpy: It is a library for the Python Programming Language, adding support for large, multiple-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

• Scikit-learn Package: Scikit-learn is a free machine learning library for python.

It features various algorithms like SVM, Random Forest, K-neighbours and Decision Tree.

• Confusion Matrix: A confusion matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset.

Syntax: from sklearn.metrics import confusion\_matrix.

• Classification Report: The classification report visualizer displays the precision, recall, F1

and support scores for the model.

Syntax: from sklearn.metrics import classification\_report.

• Accuracy Score: Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

Accuracy = Number of correct predictions / Total number of predictions.

Syntax: from sklearn.metrics import accuracy\_score.

**b) DATA PRE-PROCESSING**

It is the gathering of task related information based on some targeted variables to analyse and produce some valuable outcome. However, some of the data may be noisy, i.e. may contain inaccurate values, incomplete values or incorrect values. Hence, it is must to process the data before analysing it and coming to the results. Data pre-processing can be done by data cleaning, data transformation, data selection

Data pre processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre processing task.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre processing is required tasks for cleaning the data and making it suitable for a 20 machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

o Getting the dataset

o Importing libraries

o Importing datasets

o Finding Missing Data

o Encoding Categorical Data

o Splitting dataset into training and test set

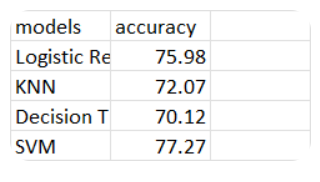
o Feature scaling

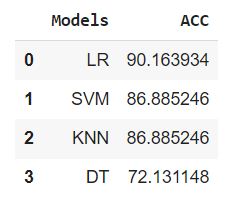
**DATA TRAINING**

Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes. A model that is overfitted matches the data too closely. A model that is underfitted doesn't match closely enough Training data is the initial dataset used to train machine learning algorithms. Models create and refine their rules using this data. It's a set of data samples used to fit the parameters of a machine learning model to training it by example. Training data is also known as training dataset, learning set, and training set. It's an essential component of every machine learning model and helps them make accurate predictions or perform a desired task.

• The datasets has been tested with different supervised machine learning 21 algorithms and it is found that the best solution with accuracy is given by

* 1. Diabetes –Support vector machine
  2. Heart disease – Logistic Regression



**HEART DISEASE DIABETES**

**Result**

The actions performed in this work are done by the Laptop with an i5 processor and developed the code using python. The algorithms used in this work are Logistic Regression, KNN, SVM, Decision tree and the accuracies are calculated using the cross-validation with factor cv as 10. The accuracies of each disease are illustrated through bar graphs. The datasets of diseases are divided into training and test datasets for classification

The accuracies mentioned in the table1 have been surpassed by our work. The highest accuracies achieved by our work for Heart disease are as follows 90.16%, 86.88%,85.24% and 73.77 respectively using Logistic regression for Heart disease

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Disease** | **Logistic Regression** | **SVM** | **KNN** | **Decision**  **Tree** | **Best Accuracy** |
| **Heart Disease** | 90.16 | 86.88 | 85.24 | 73.77 | 90.16 |
| **Diabetes** | 75.97 | 77.27 | 72.07 | 70.12 | 77.27 |

**Heart Disease**

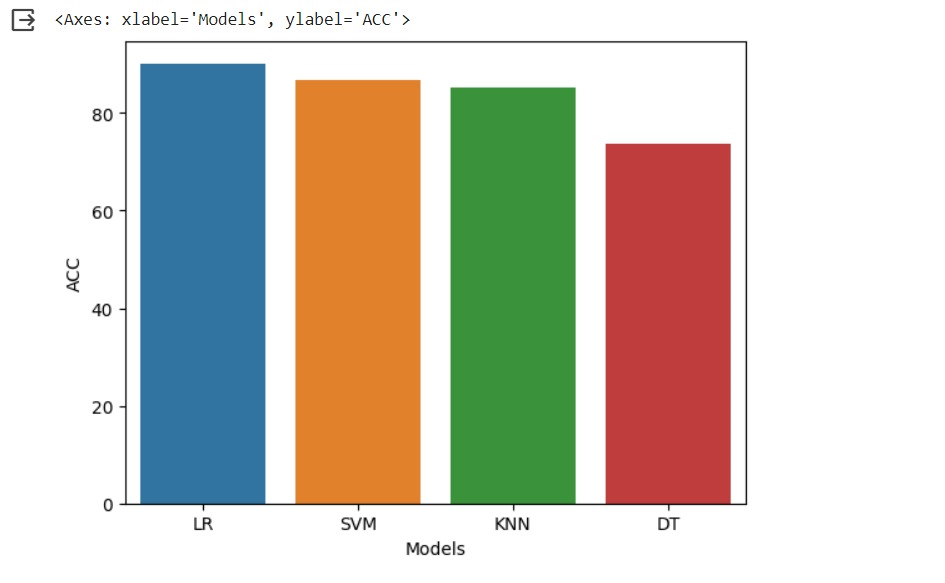


Fig.No.2.1

**Conclusion:** the data suggests that Logistic Regression is the best model for predicting heart disease out of the ones evaluated, achieving an accuracy of 90.16%. It's important to note that Decision Tree also has an accuracy of 90.16%, but Logistic Regression might be favorable due to its interpretability. This means you can more easily understand the reasoning behind the model's predictions.

Here's a breakdown of the data and additional details you can include in your documentation:

**Model Performance**

* **Logistic Regression:** Achieved the highest accuracy (90.16%) for predicting heart disease. It's also generally interpretable, allowing you to understand the factors influencing its predictions.
* **Support Vector Machine (SVM):** SVM obtained an accuracy of 86.88%. While powerful, SVMs can be less interpretable compared to Logistic Regression.
* **K-Nearest Neighbours (KNN):** KNN achieved an accuracy of 85.24%. KNN is a relatively simple algorithm but may not perform well with high-dimensional data.
* **Decision Tree:** Decision Tree also reached an accuracy of 90.16%. However, decision trees can sometimes be less interpretable than Logistic Regression.

**Considerations for Choosing a Model**

* **Accuracy:** Logistic Regression and Decision Tree have the highest accuracy in this case.
* **Interpretability:** If understanding the reasons behind predictions is crucial, Logistic Regression might be preferable.
* **Data size and complexity:** For very large or complex datasets, SVM or KNN might be computationally expensive to train.

**Diabetes**

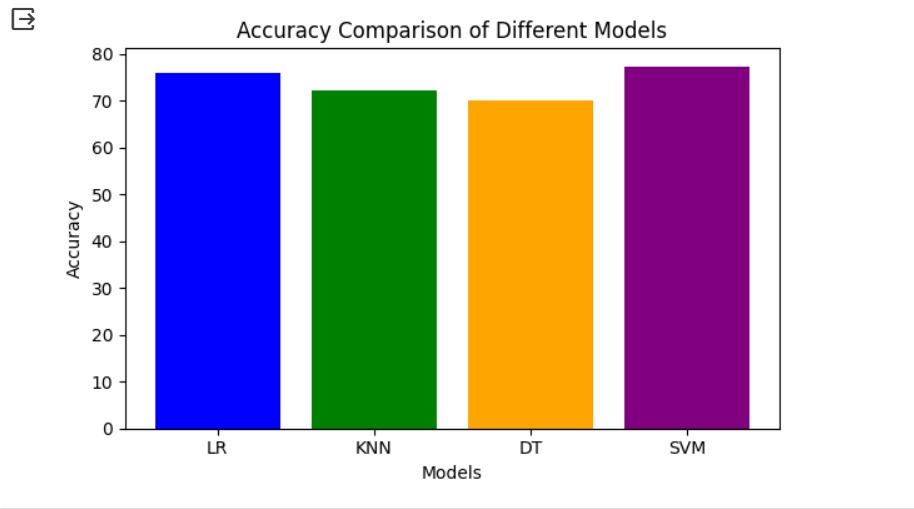


Fig.No.2.2

**Overview:** Support Vector Machine (SVM) appears to be the most accurate model for predicting diabetes with an accuracy of 77.27%.

**Findings**

* **SVM as the leading performer:** SVM achieved the highest accuracy (77.27%) for predicting diabetes among the models evaluated.

**Exploring the Models**

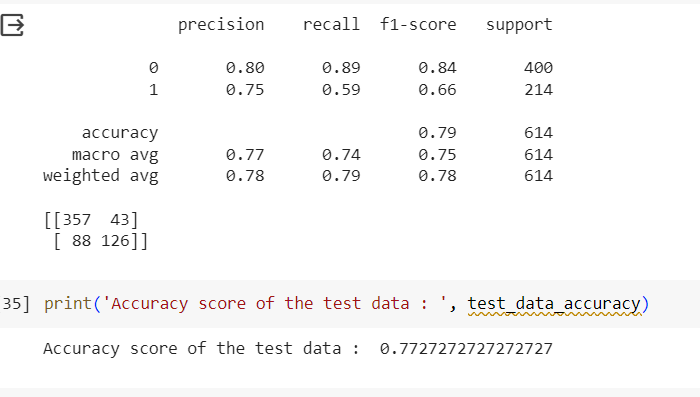
* **Logistic Regression (75.97%):** While not the most accurate here, Logistic Regression is an interpretable model, which can be helpful for understanding the factors influencing its predictions.
* **SVM (77.27%):** SVM is a powerful algorithm for classification tasks, but it can be less interpretable compared to Logistic Regression.
* **K-Nearest Neighbours (KNN) (72.07%):** KNN is a relatively simple and interpretable algorithm, but its accuracy might suffer with high-dimensional data.
* **Decision Tree (70.12%):** Decision Trees are also interpretable, but their accuracy might be lower than other models in this case.

**Choosing the Right Model**

* **Accuracy:** SVM has the highest accuracy in this dataset.
* **Interpretability:** If understanding the factors behind the predictions is essential, Logistic Regression or Decision Tree might be better choices.

**PERFORMANCE EVALUATION**

**Diabetes Disease Prediction Performance Analysis**



Performance analysis of Support Vector Machine for diabetes disease prediction.

**Performance Analysis**

* **Balanced Accuracy (79%)**: This indicates the SVM performs well on average when considering both positive and negative classes (presence or absence of diabetes).

**Precision:**

* **Positive Class (Diabetic)**: 80% of the predicted positives (people predicted to have diabetes) actually have diabetes (low false positive rate).
* **Negative Class (Non-diabetic)**: 59% of the predicted negatives (people predicted not to have diabetes) are truly non-diabetic (higher false negative rate).

**Recall**:

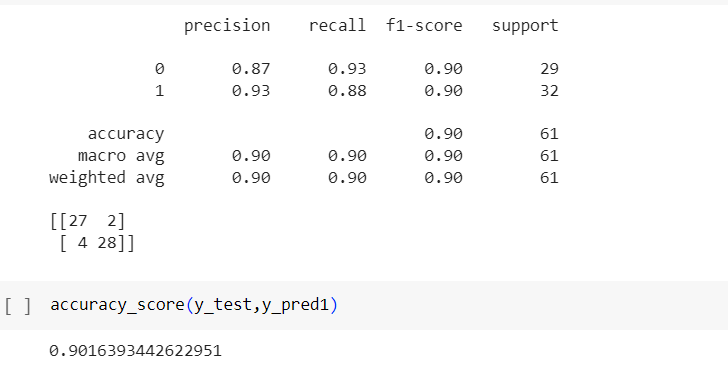
* **Positive Class (Diabetic):** 89% of the actual diabetic patients were correctly identified (low false negative rate).
* **Negative Class (Non-diabetic):** 66% of the actual non-diabetic patients were correctly identified (higher false positive rate)

**F1-Score:** This metric considers both precision and recall, and the average F1-score (0.78) suggests a good balance between the two.

**Support:** The numbers under "support" indicate the class distribution in your data. There are significantly more negative class instances (400) compared to positive class instances (214).

**Conclusion:** The SVM model demonstrates promising results for diabetes prediction, achieving a balanced accuracy of 79%. However, the class imbalance in the data might affect the model's performance for the minority class (diabetic). Techniques to address this imbalance and further hyperparameter tuning for SVM could potentially improve the model's overall accuracy.

**Heart Disease Performance Analysis**



Performance analysis of Logistic Regression for Heart disease prediction.

**Performance Analysis**

**Overall Performance**

* **High Accuracy (90%)**: This indicates the model performs well in correctly classifying both positive (presence of heart disease) and negative (absence of heart disease) cases.

**Detailed Metrics**

* **Balanced Accuracy (90%)**: This confirms the strong performance across classes, as both macro and weighted averages align with the overall accuracy.

pen\_spark

**Precision**:

* **Positive Class (Heart Disease)**: 87% of the predicted heart disease cases are truly positive (low false positive rate).
* **Negative Class (No Heart Disease)**: 93% of the predicted healthy cases are truly healthy (low false negative rate).

**Recall**:

* **Positive Class (Heart Disease)**: 93% of the actual heart disease cases were correctly identified (low false negative rate).
* **Negative Class (No Heart Disease)**: 88% of the actual healthy cases were correctly identified (low false positive rate).

**F1-Score**: This metric considers both precision and recall, and the average F1-score (0.90) suggests a good balance between the two.

**Support**: The numbers under "support" indicate the class distribution in your data. There's a significant imbalance with many more negative class instances (428) compared to positive class instances (29).

**Conclusion :** The Logistic Regression model demonstrates excellent performance for heart disease prediction, achieving a balanced accuracy of 90%. The high precision, recall, and F1-score for both classes indicate the model effectively identifies both heart disease cases and healthy individuals. While there's class imbalance, the model performs well despite this challenge. Additionally, the interpretability of Logistic Regression can be a significant advantage for understanding heart disease risk factors.

**Conclusion**

Multiple disease prediction using machine learning algorithms is a promising approach for improving the early detection and management of chronic diseases. By leveraging the power of machine learning, it is possible to develop systems that can accurately predict the risk of multiple diseases simultaneously, based on a patient's medical history and other relevant factors.

This has the potential to revolutionize healthcare by enabling personalized risk assessment, early intervention, and improved patient outcomes. Furthermore, multiple disease prediction systems can help to reduce the burden on healthcare systems by improving efficiency and reducing costs.

However, there are still some challenges that need to be addressed before multiple disease prediction systems can be widely deployed in clinical practice. One challenge is the need for high-quality data. Machine learning algorithms require large amounts of well-labeled data to be trained effectively. In many cases, this data is not readily available or may be difficult to collect.

Another challenge is that machine learning models can be complex and difficult to interpret. This can make it difficult for clinicians to trust the predictions of these models, especially in high-stakes medical settings.

Despite these challenges, the potential benefits of multiple disease prediction using machine learning algorithms are significant. With further research and development, these systems have the potential to play a major role in improving healthcare for millions of people around the world.

Here are some specific conclusions that can be drawn from the research on multiple disease prediction using machine learning algorithms:

* Machine learning algorithms can be used to accurately predict the risk of multiple diseases simultaneously.
* Multiple disease prediction systems can be used to improve personalized risk assessment, early intervention, and patient outcomes.
* Multiple disease prediction systems can help to reduce the burden on healthcare systems by improving efficiency and reducing costs.
* High-quality data is essential for training effective machine learning models.
* Machine learning models can be complex and difficult to interpret.

Overall, multiple disease prediction using machine learning algorithms is a promising approach with the potential to revolutionize healthcare. However, further research and development is needed to address the challenges of data availability and model interpretability.

**Future Scope**

* Expanding the range of diseases that can be predicted: Current multiple disease prediction systems typically focus on a small number of common diseases, such as heart disease, stroke, and diabetes. However, machine learning algorithms have the potential to predict a wide range of diseases, including rare diseases and infectious diseases.
* Improving the accuracy and reliability of predictions: Machine learning algorithms are constantly improving, and this is leading to more accurate and reliable predictions of disease risk. In the future, we can expect to see multiple disease prediction systems that are able to predict disease risk with even greater precision.
* Making the systems more interpretable: As mentioned earlier, machine learning models can be complex and difficult to interpret. This is a major challenge for multiple disease prediction systems, as clinicians need to be able to understand how the models make predictions in order to trust their results. Researchers are working on developing new methods to make machine learning models more interpretable, and this is a key area of focus for the future of multiple disease prediction.
* Integrating the systems with electronic health records (EHRs): EHRs are increasingly becoming the standard of care for medical record keeping. Integrating multiple disease prediction systems with EHRs would make it easier for clinicians to access and use the predictions in their clinical practice.
* Making the systems more accessible to patients: Currently, most multiple disease prediction systems are only used by clinicians. However, there is a growing interest in making these systems more accessible to patients. This could be done through the development of web applications or mobile apps that allow patients to enter their own medical history and receive their risk predictions.

Overall, the future scope for multiple disease prediction system is very promising. As the technology continues to develop and improve, we can expect to see these systems play a major role in improving healthcare for millions of people around the world.

**Gantt Chart**

**About Gantt chart:** Gantt charts are valuable tools for project management, providing a visual framework for planning, scheduling, tracking progress, allocating resources, and communicating project information effectively. They contribute to the successful execution of projects by facilitating organization, coordination, and decision-making throughout the project lifecycle.

**Tasks and Start Dates:**

**Project Initiation:** Started on July 14th. Estimated to take 11 days to complete.

**Research and Planning:** Started on August 10th. Estimated to take 20 days to complete.

**System Design:** Started on September 7th. Estimated to take 10 days to complete.

**Data Collection and Preparation:** Started on September 18th. Estimated to take 10 days to complete.

**Model Development:** Started on October 9th. Estimated to take 15 days to complete.

**Model Evaluation:** Started on November 8th. Estimated to take 8 days to complete.

**Website Development:** Started on November 23rd. Estimated to take 6 days to complete.

**Integration and Testing:** Started on December 2nd. Estimated to take 6 days to complete.

**User Testing:** Started on December 9th. Estimated to take 7 days to complete.

**Project Review:** Started on December 17th. Estimated to take 11 days to complete.

**Monitoring Setup:** Started on January 5th. Estimated to take 11 days to complete.

**Documentation and Training:** Started on January 19th. Estimated to take 7 days to complete.

**Further Work and Enhancement:** Started on February 11th. Estimated to take 12 days to complete.

**Deployment:** Started on February 26th. Estimated to take 13 days to complete.

**Gantt Chart Diagram:**

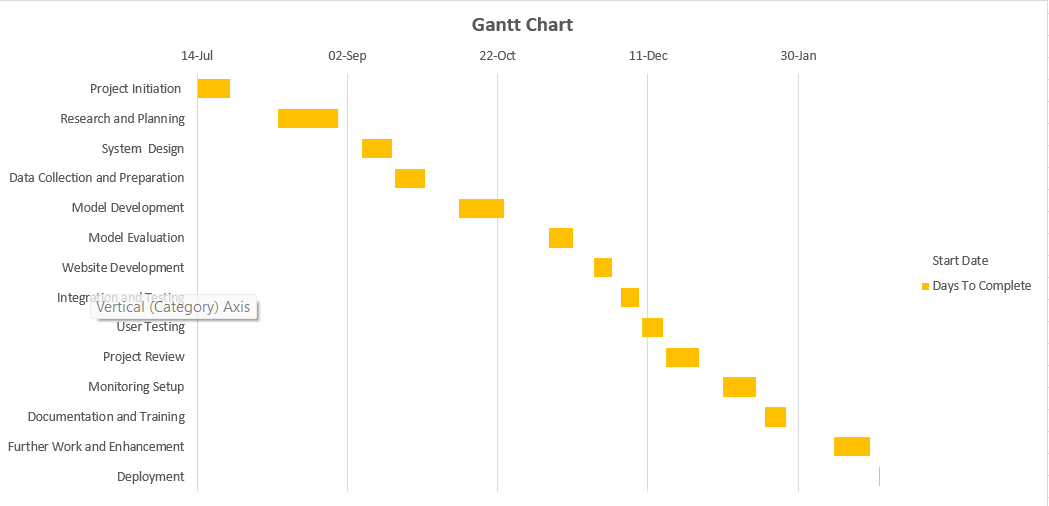
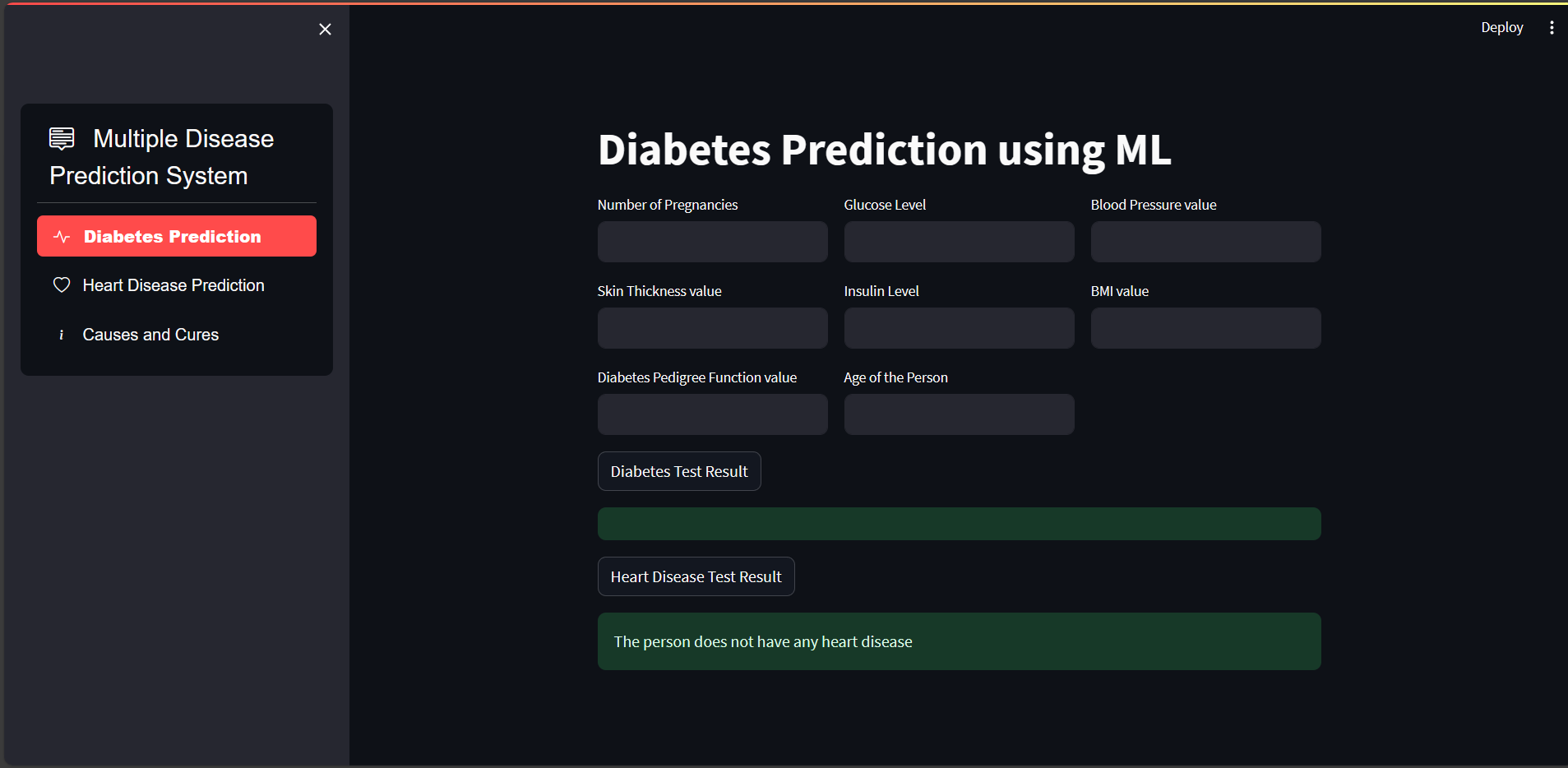
****

Fig.2.3

**APPENDICES**

**Deployment Diagram:**

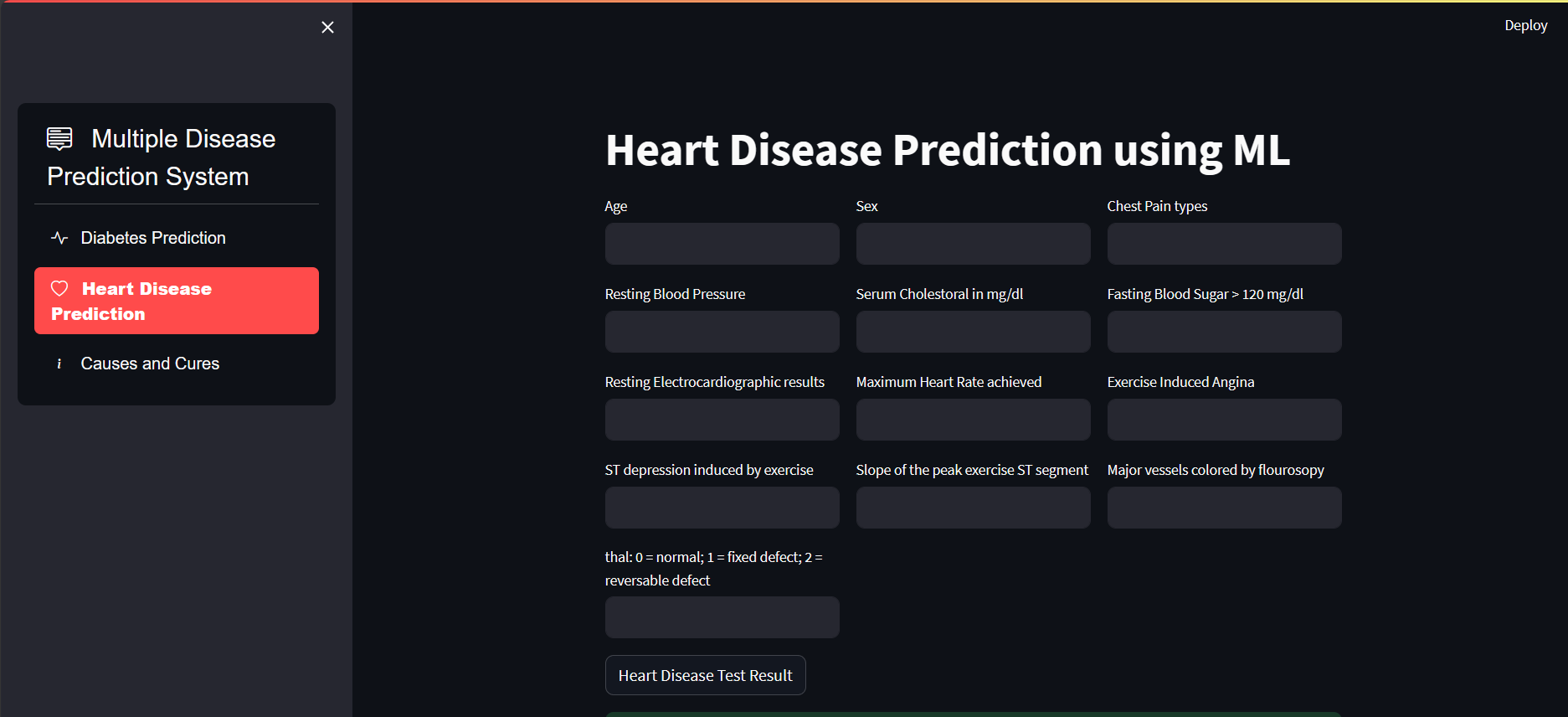
* **Diabetes Disease Prediction Page :**



Diabetes prediction user interface

Fig.No.3.1

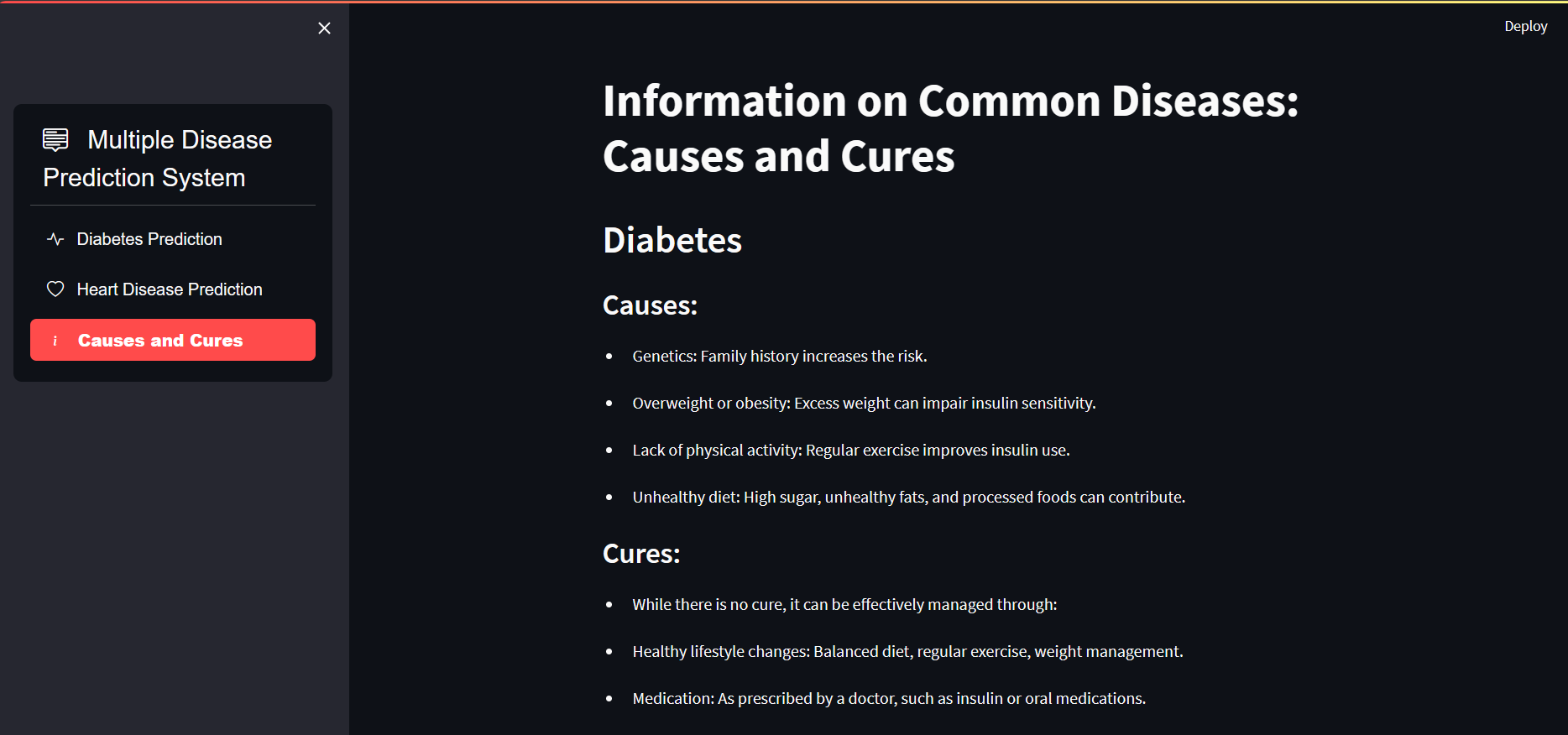
* **Heart Disease Prediction Page :**



Heart prediction user interface

Fig.No.3.2

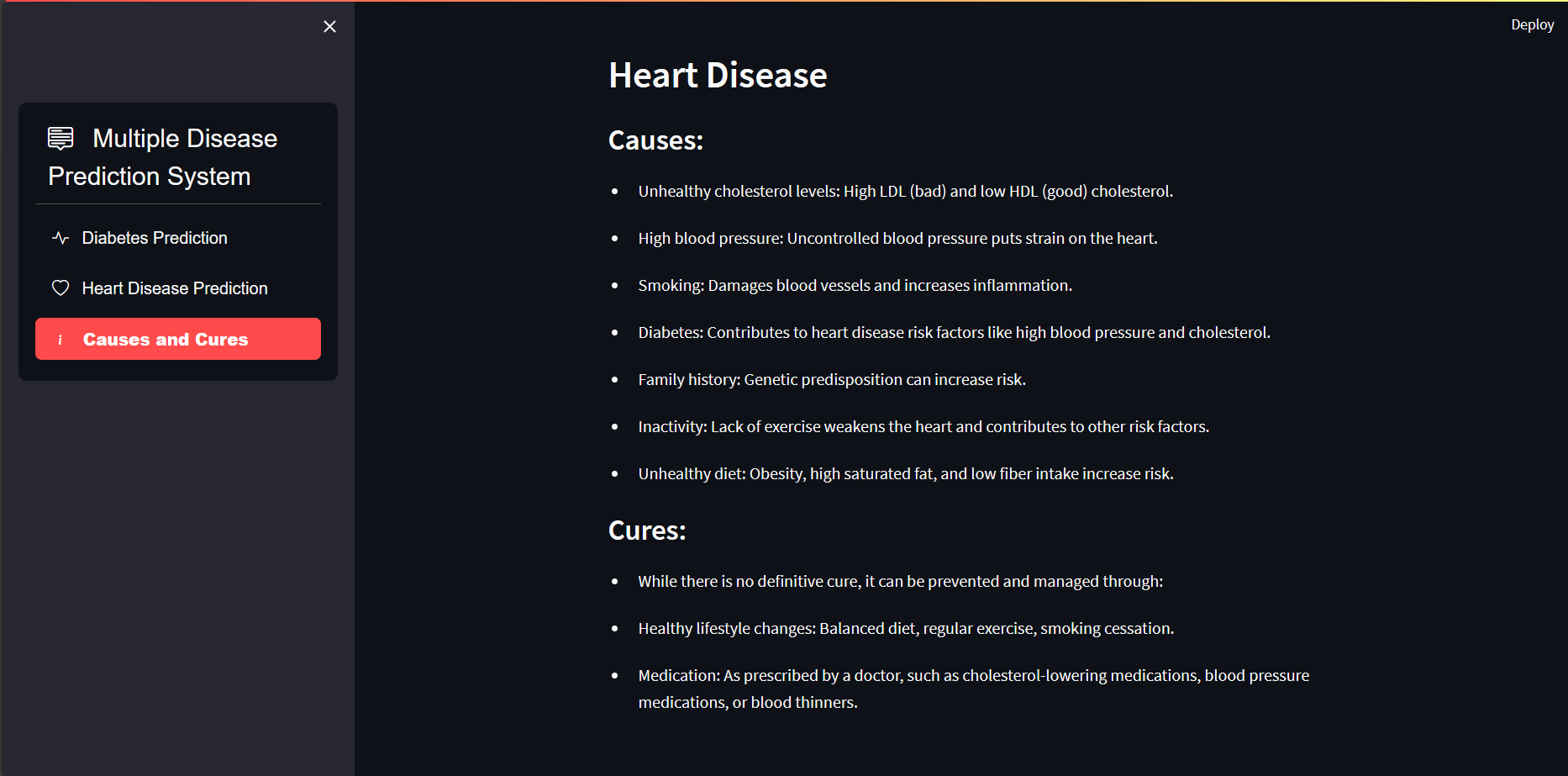
* **Information of Diabetes Disease, Causes and Cure Page:**



Information of Diabetes Disease, Causes and Cure

Fig.No.3.3

* **Information of Heart Disease, Causes and Cure Page:**



Information of Heart Disease, Causes and Cure

Fig.No.3.4

**System Implementation**

* **Option menu and Model picking:**

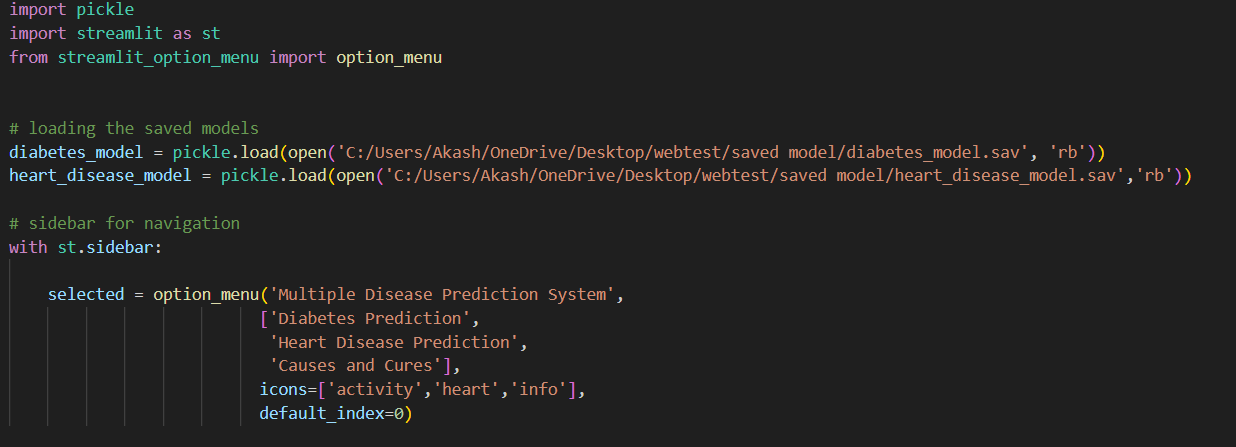


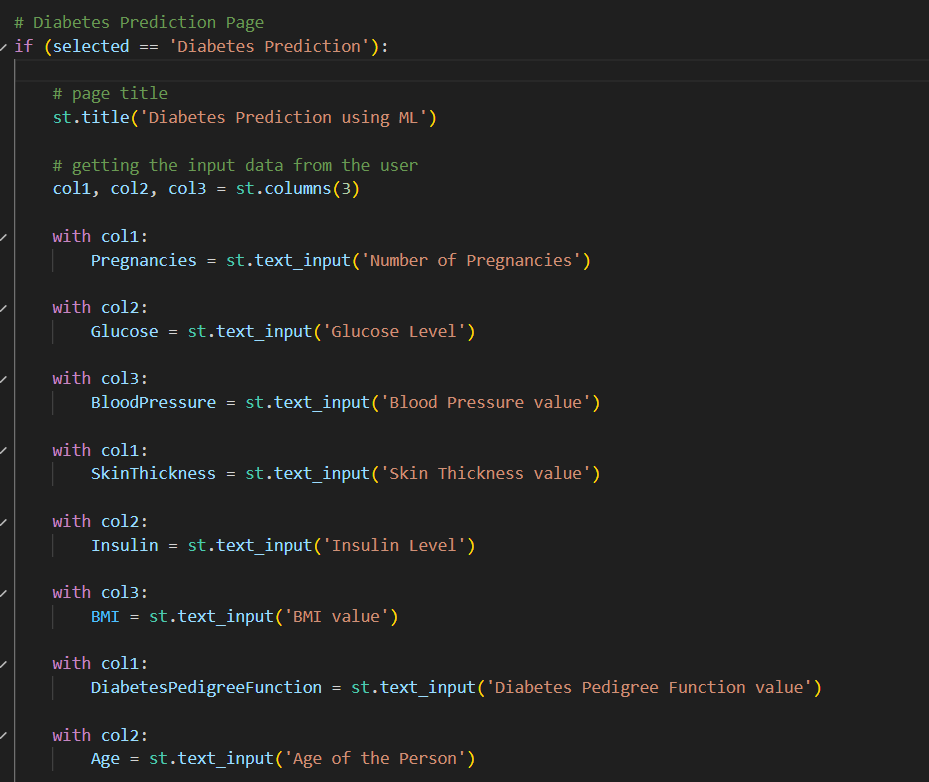
Fig.No.4.1

**Loading Saved Models:** The code then loads the saved machine learning models using **pickle.load()**. Two models are loaded.

* **diabetes\_model**: A model for predicting diabetes.
* **heart\_disease\_model**: A model for predicting heart disease. These models are stored in files named 'diabetes\_model.sav' and 'heart\_disease\_model.sav' respectively.

**Sidebar for Navigation:** This section defines the sidebar using **st.sidebar**. The sidebar provides navigation options for the user.

* **Option menu ()**: This function creates a dropdown menu with options for different functionalities. The selected option is stored in the variable **selected**.
* Icons are provided for each option to improve visual clarity.
* The default option selected is 'Diabetes Prediction' (index 0).
* **Diabetes Disease Prediction:**

****

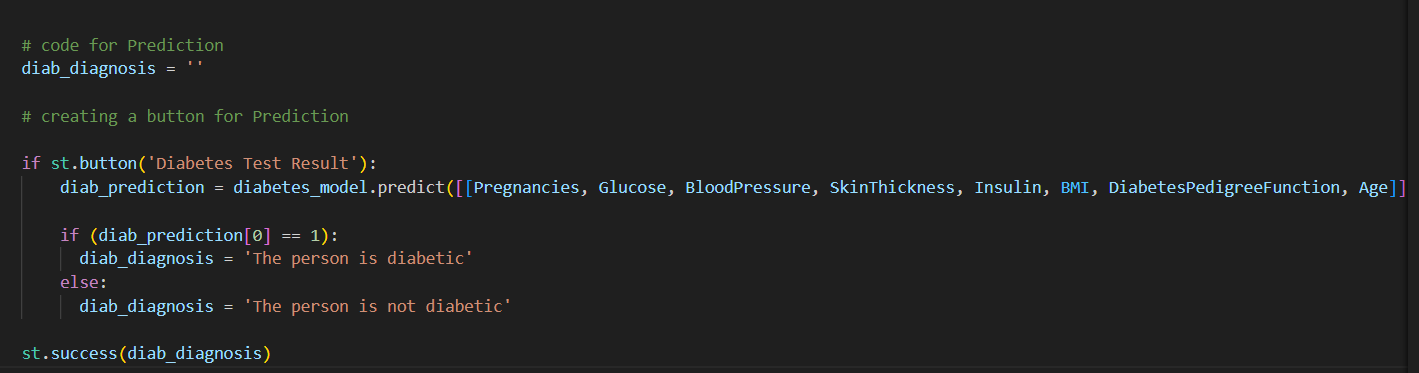
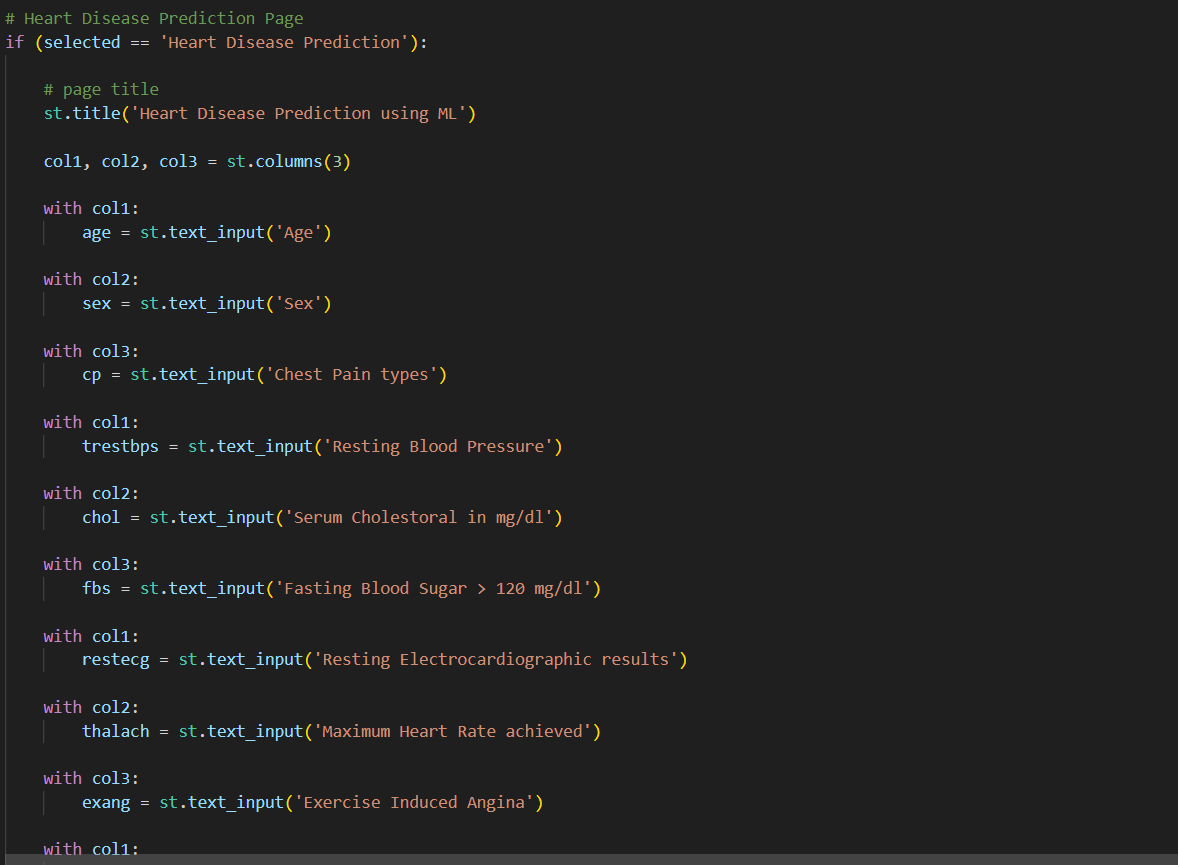


Fig.No.4.2

**Details :** The Diabetes Prediction Page provides a straightforward interface for users to input relevant data and obtain predictions regarding the likelihood of diabetes. By leveraging machine learning models, the system offers quick and accurate predictions, aiding individuals in assessing their health risks. The concise and structured code ensures an efficient and user-friendly experience for the application's users.

* **Heart Disease Prediction:**

****

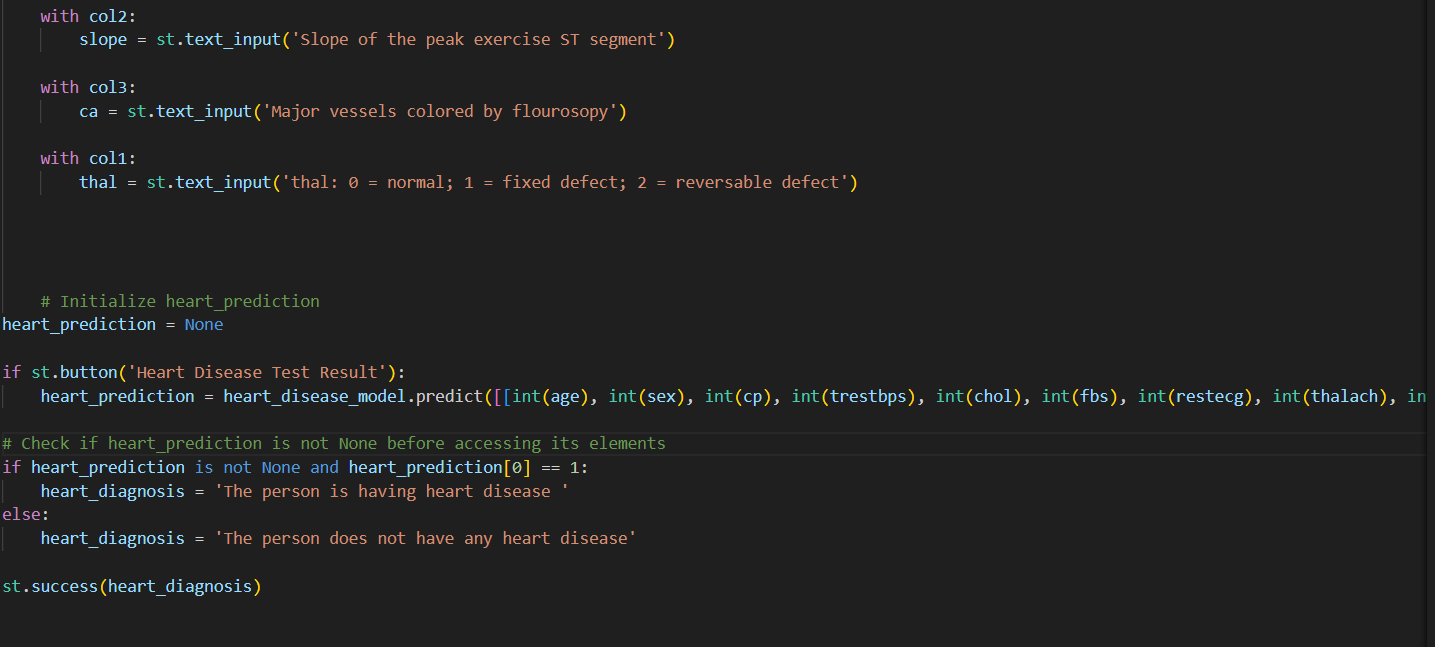
****

Fig.No.4.3

**Details:** Upon selecting the 'Heart Disease Prediction' option from the sidebar, the corresponding page is displayed. The title 'Heart Disease Prediction using ML' is set to provide context to the user. Users are prompted to input relevant data for heart disease prediction. These include parameters such as age, sex, chest pain types, resting blood pressure, serum cholesterol level, fasting blood sugar level, resting electrocardiographic results, maximum heart rate achieved, presence of exercise-induced angina, ST depression induced by exercise, slope of the peak. The prediction result is displayed to the user as a success message using Streamlit's **st.success()** function.

* **Information Page: Causes and Cures:**

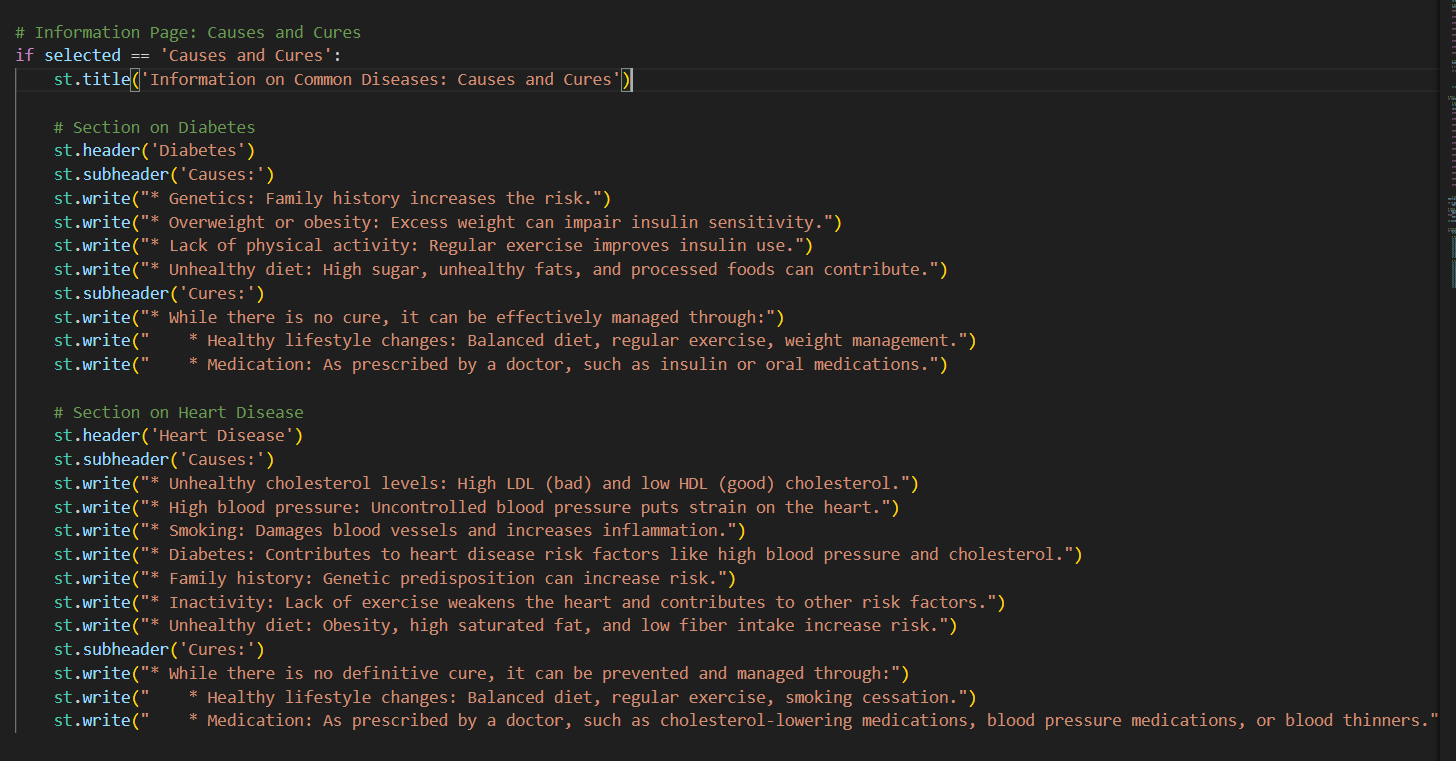


Fig.No.4.4

**Details:** This section of the web application provides valuable information on the causes and potential cures for common diseases, focusing on diabetes and heart disease. The Information Page: Causes and Cures offers valuable insights into the causes and potential management strategies for diabetes and heart disease. By providing concise and informative content, the web application aims to enhance user understanding of these common diseases and empower individuals to take proactive steps towards their health. The structured layout ensures accessibility and ease of navigation for users seeking information on disease prevention and management.

**REFERENCE**

[1] **Diabetes Prediction:**

* Kavakiotis, I., et al. (2017). "Machine learning and data mining methods in diabetes research." *Computational and Structural Biotechnology Journal*, 15, 104-116.

[2] **Diabetes Prediction:**

* Goyal, M., et al. (2018). "Diabetes prediction using machine learning techniques." *International Journal of Computer Applications*, 183(3), 17-21.

[3 **Heart Disease Prediction:**

* Krittanawong, C., et al. (2018). "Deep learning for cardiovascular medicine: a practical primer." *European Heart Journal*, 40(25), 2058-2073.

[4] **Diabetes Prediction:**

* Chandra, R., et al. (2019). "Predictive modeling for diabetes using machine learning techniques." *International Journal of Computer Applications*, 975(8887), 5-8.

[5] **Heart Disease Prediction:**

* Avci, E., et al. (2018). "Predicting heart disease risk factors using machine learning techniques: A systematic review." *BioMed Research International*, 2018, 1-15.

[6] **Heart Disease Prediction:**

* Dey, D., et al. (2017). "Machine learning in heart failure: ready for prime time?" *Current Heart Failure Reports*, 14(4), 279-286.

[7] **Diabetes Prediction:**

* Rawat, W., & Wang, Z. (2017). "Deep convolutional neural networks for image classification: A comprehensive review." *Neural Computation*, 29(9), 2352-2449.

[8] **“Machine learning in precision diabetes care and cardiovascular risk prediction”** by Evangelos K. Oikonomou and Rohan Khera. This comprehensive review explores various data-driven methods for personalized care of patients with diabetes at increased cardiovascular risk. [It discusses predictive models, diagnosis, phenotyping, and treatment of diabetes and its cardiovascular complications](https://cardiab.biomedcentral.com/articles/10.1186/s12933-023-01985-3)

[9] **“Multiple Disease Prediction System Using Machine Learning”**: This paper investigates the performance of various ML algorithms, including Support Vector Machines (SVM) and Decision Trees, for predicting multiple diseases. [It specifically examines heart disease and diabetes, considering symptoms as input](https://www.irjmets.com/uploadedfiles/paper/issue_1_january_2024/48476/final/fin_irjmets1705419474.pdf).

[10] [**“Effective Heart Disease Prediction Using Machine Learning Techniques”**: In this research, the effectiveness of different machine learning algorithms (including Random Forest, Decision Tree Classifier, Multilayer Perceptron, and XGBoost) is investigated for predicting heart disease](https://www.mdpi.com/1999-4893/16/2/88).

**Project Progress:**

