

# **Streamlit Web App for Multiple Disease Prediction using Machine Learning**

A Project Report

*submitted in partial fulfillment of the requirements*

of

Industrial Artificial Intelligent with Cloud Computing.

by

**Idrisi Hazira H. ,220841102501**

**Ramani Tushar J. ,210841102046**

**Patel Pushpak K. ,210841102036**

Under the Esteemed Guidance of

**Abdul Aziz Md, Master Trainer, Edunet Foundation.**

## ACKNOWLEDGEMENT

---

An endeavor over a long period can be successful only with the advice and support of many well-wishers. The task would be incomplete without mentioning the people who have made it possible, because is the epitome of hard work. So, with gratitude, we acknowledge all those whose guidance and encouragement owned our efforts with success.

We are highly thankful for introducing such an existing way of developing my knowledge regarding the project development. We take pleasure in presenting the report of my On-Job Training (OJT) on **Industrial Artificial Intelligent with Cloud Computing.**

First and foremost, we would like to express my deepest sense of gratitude and sincere thanks to our **Mr. Abdul Aziz Md, On- Job Trainer, Master Trainer, Edunet Foundation**, for their support and timely co-operation. His advice, encouragement and critics are source of innovative ideas, inspiration and causes behind the successful completion of this dissertation. The confidence shown on me by him was the biggest source of inspiration for me.

I express my gratitude to **Dr. Latesh B. Chaudhari, Principal, RNGPIT, Prof. Vivek C. Joshi, Head of Department, Bachelor of Vocation (Software Development) and Ms. Foram Shukla**, Assistant Professor, B.Voc (Software Development) for their constant encouragement, co-operation and support.

I am also thankful for those people who directly or indirectly helped us in project whose names not mentioned.

Finally, we would like to extend our deep sense of gratitude to all the staff members, and friends, and last but not least we are greatly indebted to our parents who inspired us under at all circumstances.

	Sincerely,
Idrisi Hazira H.	220841102501
Ramani Tushar J.	210841102046
Patel Pushpak K.	210841102036

## ABSTRACT

---

There are many existing machine learning models related to health care which mainly focuses on detecting only one disease. Therefore, this study has developed a system to forecast several diseases by using a single user interface. Multiple Disease Prediction using Machine Learning and Streamlit is a comprehensive project aimed at predicting various diseases including diabetes, heart disease, liver disease, Parkinson's disease, and breast cancer. This project leverages machine learning algorithms such as Decision Tree, Random Forest, XGBoost, Support Vector Machine (SVM), and Logistic Regression. The accuracy of each algorithm is validated and compared with each other to find the best one for prediction. Furthermore, multiple datasets (for each disease each dataset) are used to achieve utmost accuracy in the predicted results. The models are deployed using the Streamlit library, providing a user-friendly interface for disease prediction. The application interface comprises five disease options: heart disease, liver disease, diabetes, Parkinson's disease, and breast cancer. Upon selecting a particular disease, the user is prompted to input the relevant parameters required for the prediction model. Once the parameters are entered, the application promptly generates the disease prediction result, indicating whether the individual is affected by the disease or not. This project addresses the need for accurate disease prediction using machine learning techniques, allowing for early detection and intervention. The user-friendly interface provided by the Streamlit library enhances accessibility and usability, enabling individuals to easily assess their risk for various diseases. The high accuracies achieved by the different models demonstrate the effectiveness of the employed machine learning algorithms in disease prediction. The main goal is to create a web application capable of forecasting several diseases by using machine learning, including diabetes, heart disease, liver disease, Parkinson's and breast cancer.

## TABLE OF CONTENTS

Acknowledgement.....	2
Abstract .....	3
List of Figures .....	5
<b>Chapter 1. Introduction .....</b>	<b>6</b>
1.1 Problem Statement.....	7
1.2 Problem Definition .....	8
1.3 Expected Outcomes .....	8
1.4. Organization of the Report .....	10
<b>Chapter 2. Proposed Methodology.....</b>	<b>11</b>
2.1 System Design .....	14
2.2 Modules Used .....	14
2.3 Data Flow Diagram.....	16
2.4 Diagram .....	17
2.5 Streamlit.....	18
2.6 Input and Output Design.....	18
2.7 Advantages .....	19
2.8 Requirement Specification.....	19
<b>Chapter 3. Implementation and Results .....</b>	<b>20</b>
3.1. Implementation .....	21
3.2. Streamlit Interface.....	23
3.3. Result .....	25
3.4. Tools and Technologies .....	32
<b>Chapter 4. Conclusion .....</b>	<b>33</b>
<b>Chapter 5. Future Scope .....</b>	<b>35</b>
<b>GitHub Link.....</b>	<b>37</b>
<b>References .....</b>	<b>38</b>

**LIST OF FIGURES**

		<b>Page No.</b>
<b>Figure 1</b>	User Interface	<b>9</b>
<b>Figure 2</b>	Multiple disease prediction – Methodology	<b>13</b>
<b>Figure 3</b>	Multiple disease prediction - Architecture diagram	<b>14</b>
<b>Figure 4</b>	Flow of Diabetes model	<b>14</b>
<b>Figure 5</b>	Flow of Heart Disease model	<b>15</b>
<b>Figure 6</b>	Flow of Parkinson's Disease model	<b>15</b>
<b>Figure 7</b>	Flow of Liver Disease model	<b>15</b>
<b>Figure 8</b>	Flow of Breast Cancer model	<b>16</b>
<b>Figure 9</b>	Data Flow Diagram	<b>16</b>
<b>Figure 10</b>	Use Diagram	<b>17</b>
<b>Figure 11</b>	Sequence Diagram	<b>17</b>
<b>Figure 12</b>	SVM Algorithm	<b>22</b>
<b>Figure 13</b>	Logistic Regression Algorithm	<b>23</b>
<b>Figure 14</b>	Running Streamlit in Terminal	<b>23</b>
<b>Figure 15</b>	Streamlit Interface	<b>24</b>
<b>Figure 16</b>	Home Page of Web App	<b>25</b>
<b>Figure 17</b>	Single User Interface (UI) of Diabetes Disease Prediction	<b>26</b>
<b>Figure 18</b>	Output for diabetes disease for negative case	<b>26</b>
<b>Figure 19</b>	Single User Interface (UI) of Heart Disease Prediction	<b>27</b>
<b>Figure 20</b>	Output for heart disease for positive case	<b>27</b>
<b>Figure 21</b>	Single User Interface (UI) of Parkinson's Disease Prediction	<b>28</b>
<b>Figure 22</b>	Output for Parkinson's disease for positive case	<b>28</b>
<b>Figure 23</b>	Single User Interface (UI) of Liver Disease Prediction	<b>29</b>
<b>Figure 24</b>	Output for Liver disease for negative case	<b>29</b>
<b>Figure 25</b>	Single User Interface (UI) of Breast Cancer Prediction	<b>30</b>
<b>Figure 26</b>	Output for Breast Cancer for negative case	<b>31</b>

# **CHAPTER 1**

## **INTRODUCTION**

## CHAPTER 1

### INTRODUCTION

The project " Streamlit Web App for Multiple Disease Prediction using Machine Learning " focuses on predicting five different diseases: diabetes, heart disease, liver disease, Parkinson's disease, and breast cancer. The prediction models are built using machine learning algorithms, including Support Vector Machine (SVM) for diabetes and Parkinson's disease, Logistic Regression for heart disease and breast cancer, and XGBoost for liver disease. The application is deployed using Streamlit library. The project begins by collecting relevant data from Kaggle.com, which is then preprocessed to prepare it for training and testing the prediction models. Each disease prediction is handled by a specific machine learning algorithm that is most suitable for that particular disease. SVM is employed for diabetes and Parkinson's disease, Logistic Regression for heart disease and breast cancer, and XGBoost for liver disease. The application interface offers five options, each corresponding to a specific disease. When a user selects a particular disease, the application prompts for the necessary parameters required by the corresponding model to predict the disease result. The user provides the required parameters, and the application displays the prediction result based on the input. To deploy the prediction models, the Streamlit library are utilized. The Streamlit library simplifies the process of developing interactive and user-friendly web applications. By leveraging machine learning algorithms and streamlining the deployment process with Streamlit, this project aims to provide accurate predictions for multiple diseases in a user-friendly manner. The application's intuitive interface allows users to input disease-specific parameters and obtain prediction results, facilitating early detection and proactive healthcare management.

#### **1.1. Problem Statement:**

Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. For example, first is for Diabetes analysis, one for Heart analysis, one for Liver diseases like that. If a user wants to predict more than one disease, he/she has to go through different sites. There is no common system where one analysis can perform more than one disease prediction. Some of the models have lower accuracy which can seriously affect patients' health. When an organization wants to analyze their patient's health reports, they have to deploy many models which in turn increases the cost as well as time Some of the existing systems consider very few parameters which can yield false results.

## **1.2. Problem Definition:**

Develop a machine learning-based application using Support Vector Machine (SVM), and Logistic Regression to predict multiple diseases including diabetes, heart disease, liver disease, Parkinson's disease, and breast cancer. In multiple disease prediction, it is possible to predict more than one disease at a time. So, the user doesn't need to traverse different sites in order to predict the diseases. The application should allow users to input relevant parameters for a specific disease and provide an accurate prediction of whether an individual is affected by the disease based on the trained models. The project aims to improve healthcare outcomes by enabling early detection and prediction of diseases using machine learning algorithms and streamlining the prediction process through an intuitive user interface.

## **1.3. Expected Outcomes:**

### **1.3.1. Accurate Disease Prediction:**

- The primary objective is to develop machine learning models with high accuracy for predicting multiple diseases including diabetes, heart disease, liver disease, Parkinson's disease, and breast cancer.
- Each disease prediction model should achieve a satisfactory level of accuracy based on the selected algorithms and training data.

### **1.3.2. User-Friendly Interface:**

- The Streamlit web application should offer an intuitive and user-friendly interface.
- Users should be able to easily navigate through the application, select the disease they want to predict, and input relevant parameters.

### **1.3.3. Streamlined Deployment:**

- The deployment process should be seamless using the Streamlit library.
- The web application should be accessible to users without technical expertise, facilitating widespread use and adoption.

### **1.3.4. Early Detection and Prevention:**

- By enabling early detection of diseases, the application should contribute to proactive healthcare management.
- Early detection can lead to timely interventions, potentially improving treatment outcomes and reducing healthcare costs.

### **1.3.5. Cost and Time Efficiency:**

- Consolidating multiple disease prediction models into a single platform saves users time and effort.



## Streamlit Web App for Multiple Disease Prediction using Machine Learning

- Organizations can benefit from cost savings by utilizing a unified system for analyzing patient health reports instead of deploying multiple models.

### 1.3.6. Scalability and Extensibility:

- The application should be scalable to accommodate additional diseases in the future.
- Future enhancements can focus on improving prediction accuracy, adding new features, and enhancing user experience.

### 1.3.7. Contribution to Healthcare:

- The project contributes to the advancement of healthcare technology by leveraging machine learning for disease prediction.
- The outcomes of the project have the potential to positively impact healthcare outcomes and patient well-being.



Figure 1: User Interface

Welcome to “ Streamlit Web App For Multiple Disease Prediction using Machine Learning ”

### Explore Diseases:

- Diabetes
- Heart Disease
- Liver Disease
- Parkinson's Disease
- Breast Cancer

This is the home page from where we can navigate to all pages like diabetes, heart, liver, Parkinson's and breast cancer disease.

## **1.4. Organization of the Report**

### **Chapter 1: Introduction**

- Introduce the project, its objectives, and the need for predicting multiple diseases.
- Define the problem statement and problem definition.
- Outline the expected outcomes.

### **Chapter 2: Proposed Methodology**

- Describe the methodology followed in the project, including data collection, preprocessing, model selection, training, testing, and deployment.
- Detail the system design and modules used for each disease prediction.

### **Chapter 3: Implementation and Results**

- Discuss the implementation details of machine learning algorithms such as XGBoost, SVM, and Logistic Regression.
- Present the results obtained from the deployed models, including user interface screenshots.

### **Chapter 4: Conclusion**

- Summarize the project's achievements and contributions.
- Reflect on the implications of the findings and their potential impact on healthcare.

### **Chapter 5: Future Scope**

- Suggestions for future improvements or extensions to the project.
- Future enhancements or potential improvements.

### **References**

- Provide a list of all cited sources for further reading and validation.
- Citations for all sources referenced in the report.

## **CHAPTER 2**

### **PROPOSED METHODOLOGY**

## **CHAPTER 2**

### **PROPOSED METHODOLOGY**

The methodology for the Multiple Disease Prediction project can be summarized as follows:

1. **Data Collection:** Data is collected from Kaggle.com, a popular platform for accessing datasets. The data is obtained specifically for diabetes, heart disease, liver disease, Parkinson's disease, and breast cancer.
2. **Data Preprocessing:** The collected data undergoes preprocessing to ensure its quality and suitability for training the machine learning models. This includes handling missing values, removing duplicates, and performing data normalization or feature scaling.
3. **Model Selection:** Different machine learning algorithms are chosen for each disease prediction task. Support Vector Machine (SVM), Logistic Regression, and XGBoost are selected as the algorithms for various diseases based on their performance and suitability for the specific prediction tasks.
4. **Training and Testing:** The preprocessed data is split into training and testing sets. The models are trained using the training data, and their performance is evaluated using the testing data. Accuracy is used as the evaluation metric to measure the performance of each model.
5. **Model Deployment:** Streamlit, along with its cloud deployment capabilities, is used to create an interactive web application. The application offers a user-friendly interface with five options for disease prediction: heart disease, liver disease, diabetes, Parkinson's disease, and breast cancer. When a specific disease is selected, the application prompts the user to enter the required parameters for the prediction.

## Streamlit Web App for Multiple Disease Prediction using Machine Learning

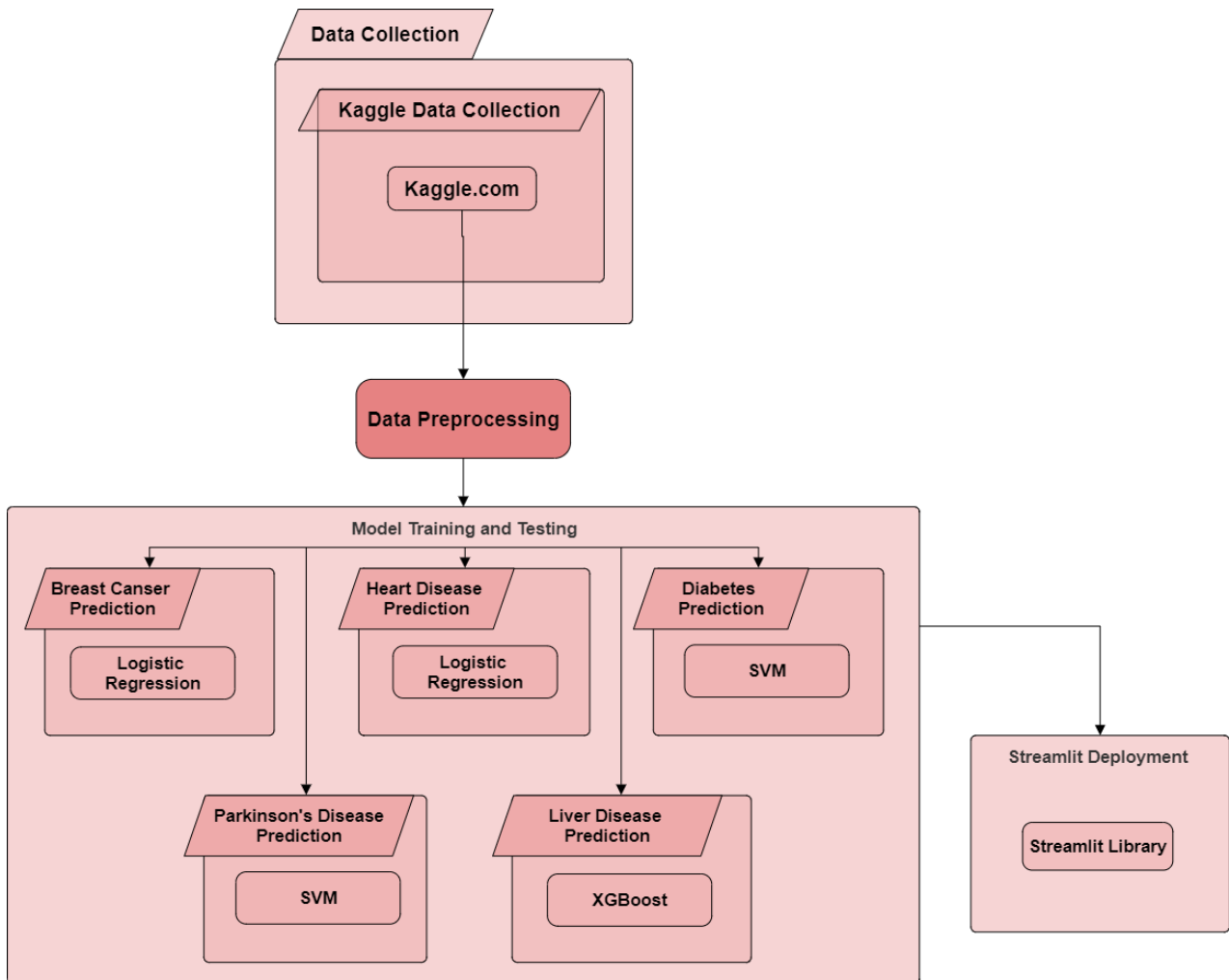


Figure 2 : Multiple disease prediction - Methodology

## 2.1 System Design

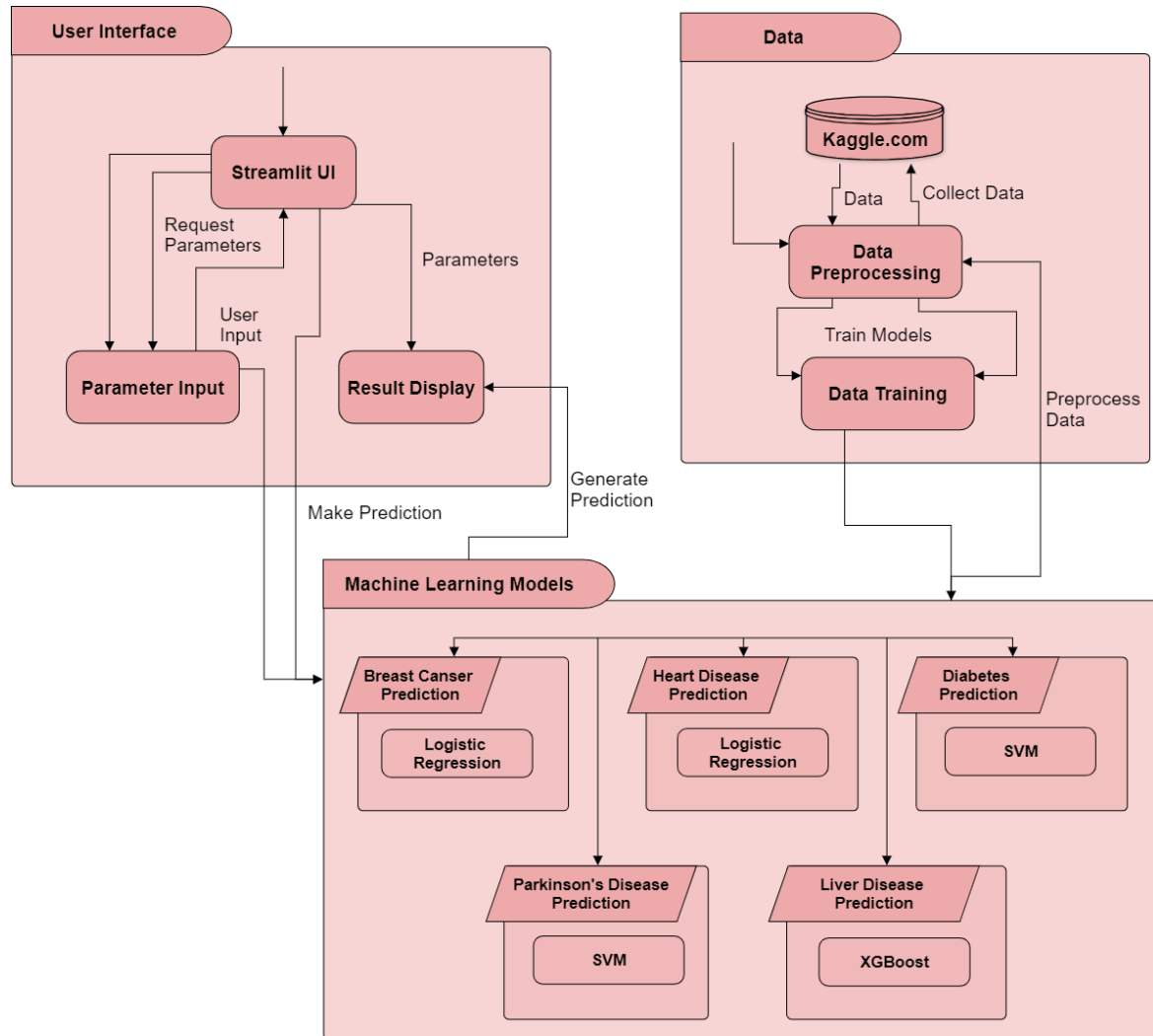


Figure 3: Multiple disease prediction - Architecture diagram

## 2.2 Modules Used

### 1. DIABETES DISEASE PREDICTION

- The aim of the prediction is which can perform early prediction of diabetes of a patient.
- This aims to predict via different supervised machine learning methods.
- It uses data about the Effected and normal people and preferences to generate Whether person is effected or not from a particular Disease.
- It performs the Different machine algorithms like Logistics Regression, Random Forest, SVM, Decision Tree, XGBoost, etc.
- Highest Accuracy achieved by SVM 78.34%



Figure 4 : Flow of Diabetes model

## 2. HEART DISEASE PREDICTION

- It uses data about the Effected and normal people and preferences to generate Whether person is effected or not from a particular Disease.
- This aims to predict via different supervised machine learning methods.
- It performs the Different machine algorithms like Logistics Regression, Random Forest, SVM, Decision Tree, XGBoost, etc.
- Highest Accuracy achieved by Logistic regression 81.96%

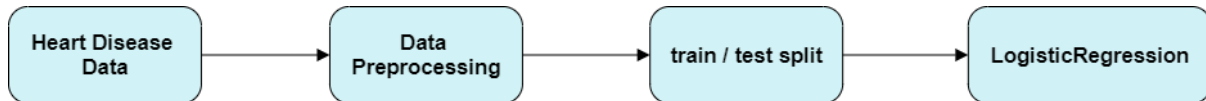


Figure 5 : Flow of Heart Disease model

## 3. PARKINSONS DISEASE PREDICTION

- The Parkinson Disease Prediction module is one of the core of multiple Disease Prediction.
- It uses data about the Effected and normal people and preferences to generate Whether person is effected or not from a particular Disease.
- It performs the Different machine algorithms like Logistics Regression, Random Forest, SVM, Decision Tree, XGBoost, etc.
- Highest Accuracy achieved by SVM 87.18%



Figure 6 : Flow of Parkinsons Disease model

## 4. LIVER DISEASE PREDICTION

- The Liver Disease Prediction module is one of the core of multiple Disease Prediction.
- It uses data about the Effected and normal people and preferences to generate Whether person is effected or not from a particular Disease.
- It performs the Different machine algorithms like Logistics Regression, Random Forest, SVM, Decision Tree, XGBoost, etc.
- Highest Accuracy achieved by XGBoost 80.89%.



Figure 7 : Flow of Liver Disease model

## 5. BREAST CANCER PREDICTION

- The Breast cancer Prediction module is one of the core of multiple Disease Prediction.
- It uses data about the Effected and normal people and preferences to generate Whether person is effected or not from a particular Disease.
- It performs the Different machine algorithms like Logistics Regression, Random Forest, SVM, Decision Tree, XGBoost, etc.
- Highest Accuracy achieved by Logistic regression 82.89%

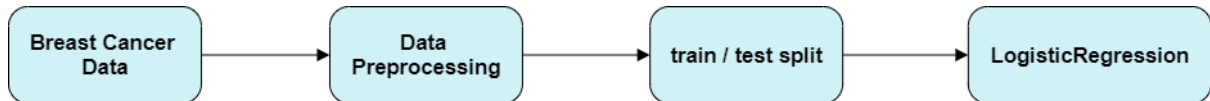


Figure 8 : Flow of Breast Cancer model

## 2.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

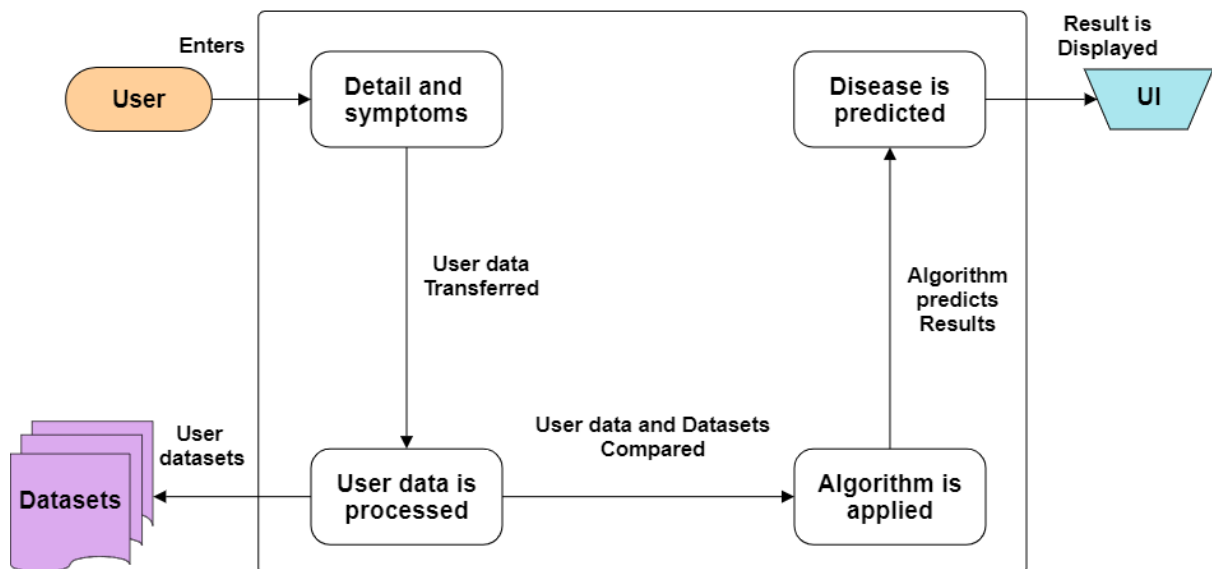


Figure 9 : Data Flow Diagram



## 2.4 Diagram

### 2.4.1 Use case diagram

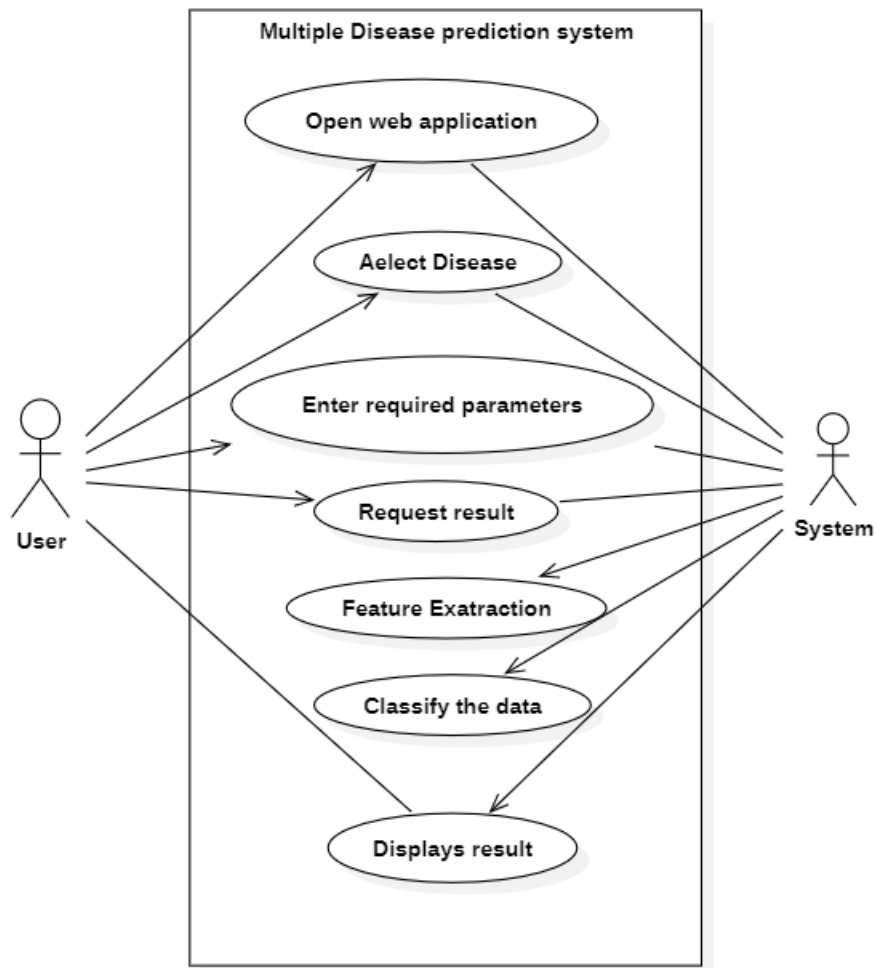


Figure 10 : Use Diagram

### 2.4.2 Sequence diagram

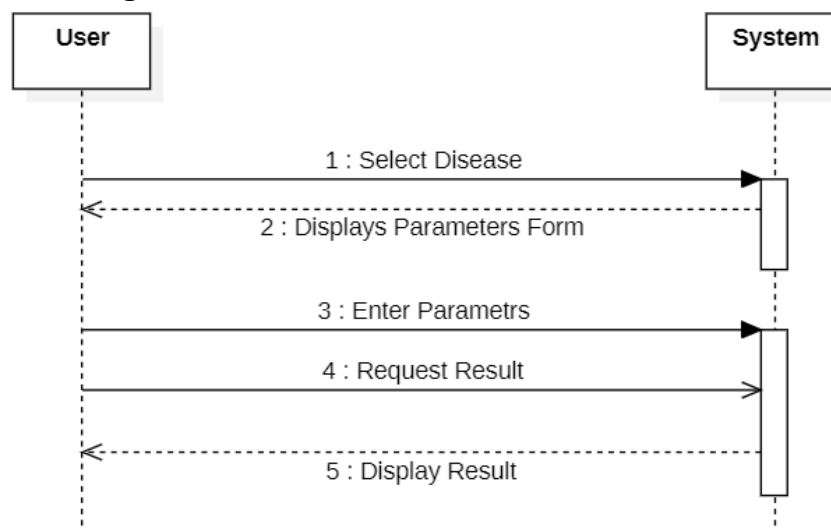


Figure 11 : Sequence Diagram

## 2.5 Streamlit

Streamlit is a Python library that allows you to create interactive web applications for machine learning, data science, and other computational tasks. With Streamlit, you can build web apps quickly and easily using familiar Python syntax, without needing to write HTML, CSS, or JavaScript code.

It's popular because of its simplicity and ease of use. You can create interactive elements like sliders, buttons, and text inputs with just a few lines of code, and Streamlit takes care of rendering them in the browser.

Many data scientists and machine learning engineers use Streamlit to create prototypes, share their work with others, or even deploy their models as web apps for broader accessibility.

### **Advantages of Streamlit :**

- 1 Easy to Use: Streamlit's simple Python syntax makes it accessible for developers of all levels, allowing them to quickly create interactive machine learning apps.
- 2 Fast Prototyping: With Streamlit, you can prototype machine learning models and apps rapidly, getting real-time feedback as you make changes.
- 3 Interactive Features: Streamlit offers intuitive components like sliders and buttons, enabling users to interact with machine learning models and data dynamically.
- 4 Python Integration: Streamlit seamlessly integrates with popular Python libraries like Pandas and Scikit-learn, leveraging your existing Python skills and libraries.
- 5 Easy Deployment: Streamlit simplifies deployment with built-in tools, allowing you to share your machine learning apps easily, whether locally or on the web.
6. Community Support: Streamlit has a strong community providing resources and extensions, making it easier to find help and accelerate development.

## 2.6 Input and Output Design

**Input Design:** The Multiple Disease Prediction system requires user input in the form of parameters specific to each disease. When the user selects a particular disease from the options menu, the system prompts for the relevant parameters. The input design should ensure that the user can easily provide the required information. The application provides a user interface with a menu containing five disease options: heart disease, liver disease, diabetes, Parkinson's disease, and breast cancer. When the user clicks on a specific disease, the application prompts for the required parameters for that particular disease prediction. The input design should ensure that the parameters requested are relevant and necessary for accurate disease

prediction. The user should be able to enter the parameters in a user-friendly and intuitive manner.

**Output Design:** The Multiple Disease Prediction system provides the predicted result of whether the person is affected by the selected disease or not. The output design should present the result in a clear and understandable format. The system should display the output after the user has entered the parameters. The output could be presented as:

- "Prediction: The person is affected by [Disease Name]." (If the prediction is positive)
- "Prediction: The person is not affected by [Disease Name]." (If the prediction is negative)

The output should be displayed on the user interface, allowing the user to easily interpret the prediction result. Overall, the input design ensures that the user can enter the necessary parameters for disease prediction, while the output design presents the prediction result clearly on the user interface

## 2.7 Advantages

- Easily analyze the disease
- High Accuracy
- Early detection: Finds sicknesses early.
- Saves money: Finding sicknesses early saves cash.
- Handles lots of data: Deals with loads of info from different sources.

## 2.8 Requirement Specification

Hardware Requirements	Software Requirements
• <b>System</b> : i5 Processor.	• <b>Operating system</b> : Windows 10.
• <b>Hard Disk</b> : 500 GB.	• <b>Coding Language</b> : Python.
• <b>Monitor</b> : “15” LED.	• <b>Front-End</b> : Streamlit 1.3.3.0, Python
• <b>Input Devices</b> : Keyboard, Mouse.	• <b>Back-End</b> : Python 3.11.5
• <b>Ram</b> : 4- 8 GB	• <b>Python Modules</b> : Pickle 4.0

## **CHAPTER 3**

### **IMPLEMENTATION AND RESULT**

## CHAPTER 3

### IMPLEMENTATION AND RESULT

#### 3.1 Implementation

##### 3.1.1 XG Boost Algorithm

The working of XG Boost algorithm are as follows:

Step 1: Firstly, create a single leaf tree.

Step 2: Then for the first tree, we must compute the average of the target variable as prediction and then calculate the residuals using the desired loss function and then for subsequent trees the residuals come from prediction that was there in the previous tree.

Step 3: Calculating the similarity score using formula:

$$\text{Similarity Score} = \frac{\text{Gradient}^2}{\text{Hessian} + \lambda}$$

where, Hessian is equal to the number of residuals; Gradient<sup>2</sup> = squared sum of residuals;  $\lambda$  is a regularization hyperparameter..

Step 4: Applying a similarity score we select the appropriate node. The higher the similarity score the more homogeneity.

Step 5: Applying similarity scores we calculate Information gain. Information helps to find the difference between old similarity and new similarity and tells how much homogeneity is achieved by splitting the node at a given point. It is calculated by the formula:

$$\text{Information Gain} = \text{Left Similarity} + \text{Right Similarity} - \text{Similarity for Root}$$

Step 6: Creating the tree of desired length using the above method pruning and regularization can be done by playing with the regularization hyperparameter.

Step 7: Then we can predict the residual values using the Decision Tree you constructed.

Step 8: The new set of residuals is calculated as:

$$\text{New Residuals} = \text{Old Residuals} + \rho \sum \text{Predicted Residuals}$$

where  $\rho$  is the learning rate.

Step 9: Then go back to step 1 and repeat the process for all the trees

### 3.1.2 SVM Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

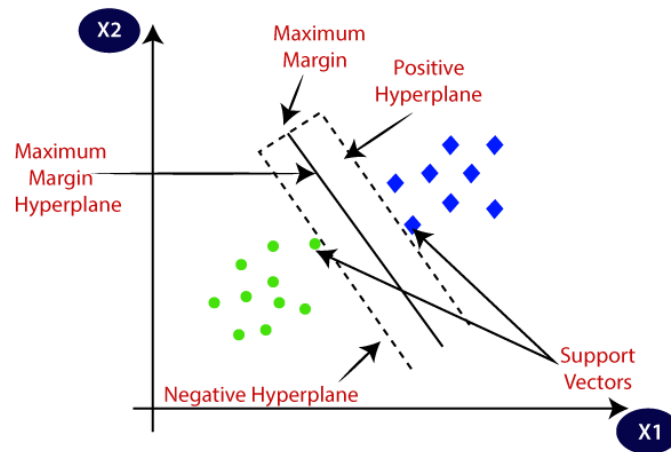


Figure 12 : SVM Algorithm

### 3.1.3 Logistic Regression Algorithm

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

A statistical model typically used to model a binary dependent variable with the help of logistic function. Another name for the logistic function is a sigmoid function and is given by:

$$F(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

This function assists the logistic regression model to squeeze the values from  $(-\infty, \infty)$  to  $(0, 1)$ . Logistic regression is majorly used for binary classification tasks; however, it can be used for multiclass classification. Logistic regression starts from a linear equation. However, this

equation consists of log-odds which is further passed through a sigmoid function which squeezes the output of the linear equation to a probability between 0 and 1. And, we can decide a decision boundary and use this probability to conduct classification task.

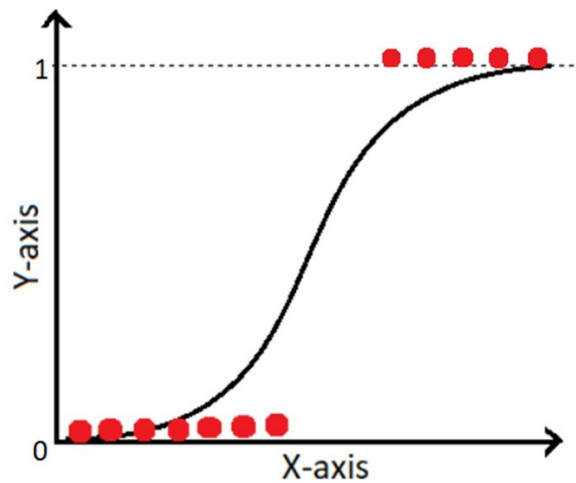


Figure 13 : Logistic Regression Algorithm

### 3.2 Streamlit Interface

The implementation of the interface happens through streamlit-option-menu library where the five options are diabetes, heart disease, Parkinson's disease, liver disease, breast cancer and the models are loaded on their respected options. The diabetes takes the input of 8 attributes with specified values mentioned and the button get result shows the result after giving all inputs. Same as diabetes the heart disease takes 13 attributes as specified names and gives the result. And the Parkinson's disease takes 22 attributes as specified names and gives the result. And the liver disease takes 10 attributes as specified names and gives the result. And last the breast cancer disease takes 30 attributes as specified names and gives the result. Fig shows the interface of five diseases as options at the left of the interface.

```
Administrator: C:\WINDOW × + v
(MachineLearning) C:\Users\tusha>streamlit run "C:\Users\tusha\Desktop\OJT\Heart Disease\app.py"
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.73.148:8501
```

Figure 14 : Running Streamlit in Terminal

## Streamlit Web App for Multiple Disease Prediction using Machine Learning



Figure 15 : Streamlit Interface



## 3.3 Result

### 3.3.1 Home Page

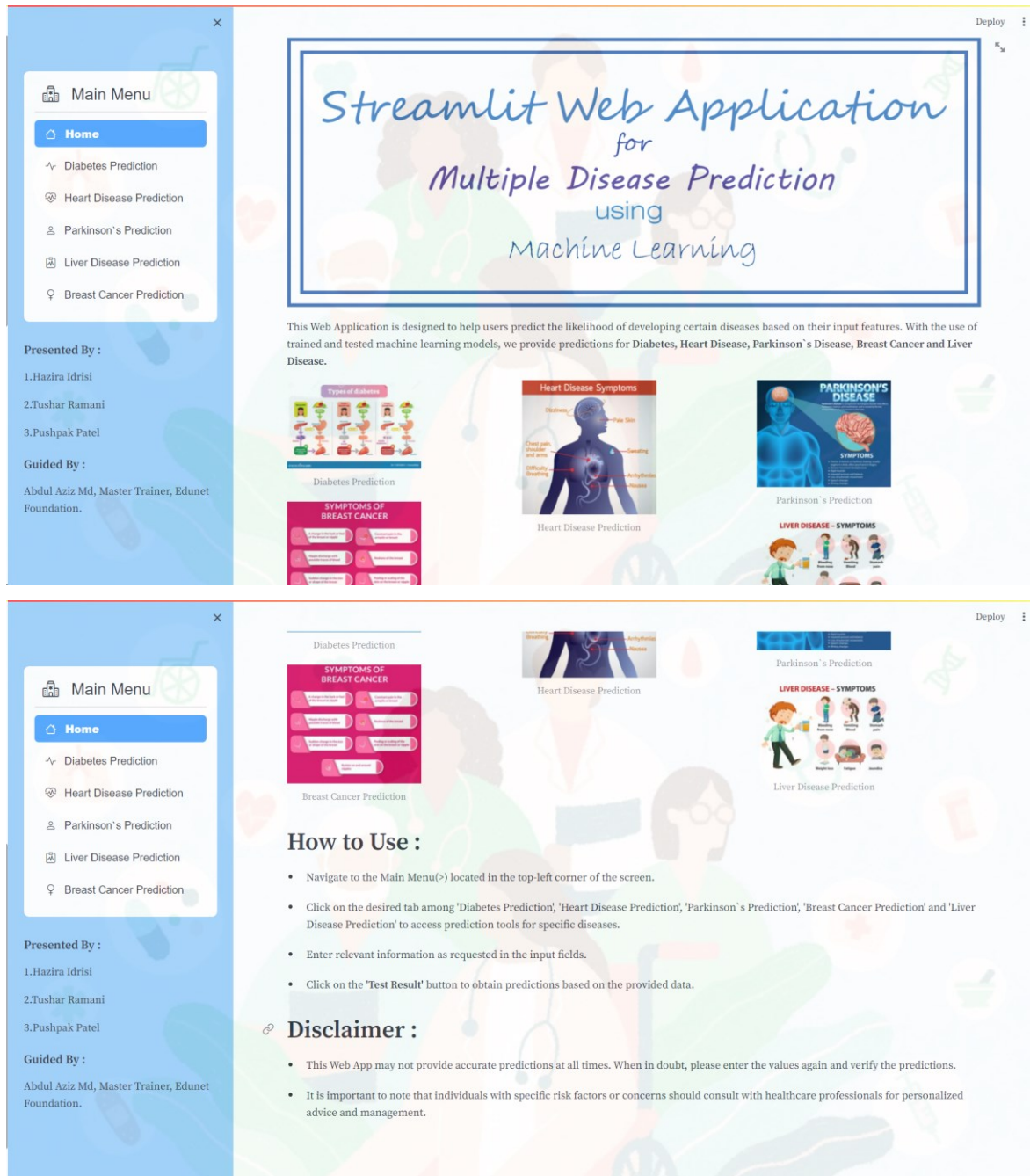


Figure 16 : Home Page of Web App

### 3.3.2 Diabetes Disease Prediction

Diabetes Prediction using ML

Enter Your Name:

Number of Pregnancies

Glucose Level

Blood Pressure value

Skin Thickness value

Insulin Level

BMI value

Diabetes Pedigree Function value

Age of the Person

Diabetes Test Result

Figure 17 : Single User Interface (UI) of Diabetes Disease Prediction

Diabetes Prediction using ML

Enter Your Name:

Tushar Ramani

Number of Pregnancies

0

Glucose Level

100

Blood Pressure value

50

Skin Thickness value

12

Insulin Level

20

BMI value

12

Diabetes Pedigree Function value

0.22

Age of the Person

20

Diabetes Test Result

**NEGATIVE**

Congratulation Tushar Ramani You are not diabetic

Figure 18 : Output for diabetes disease for negative case

### 3.3.3 Heart Disease Prediction

Figure 19 : Single User Interface (UI) of Heart Disease Prediction

Figure 20 : Output for heart disease for positive case

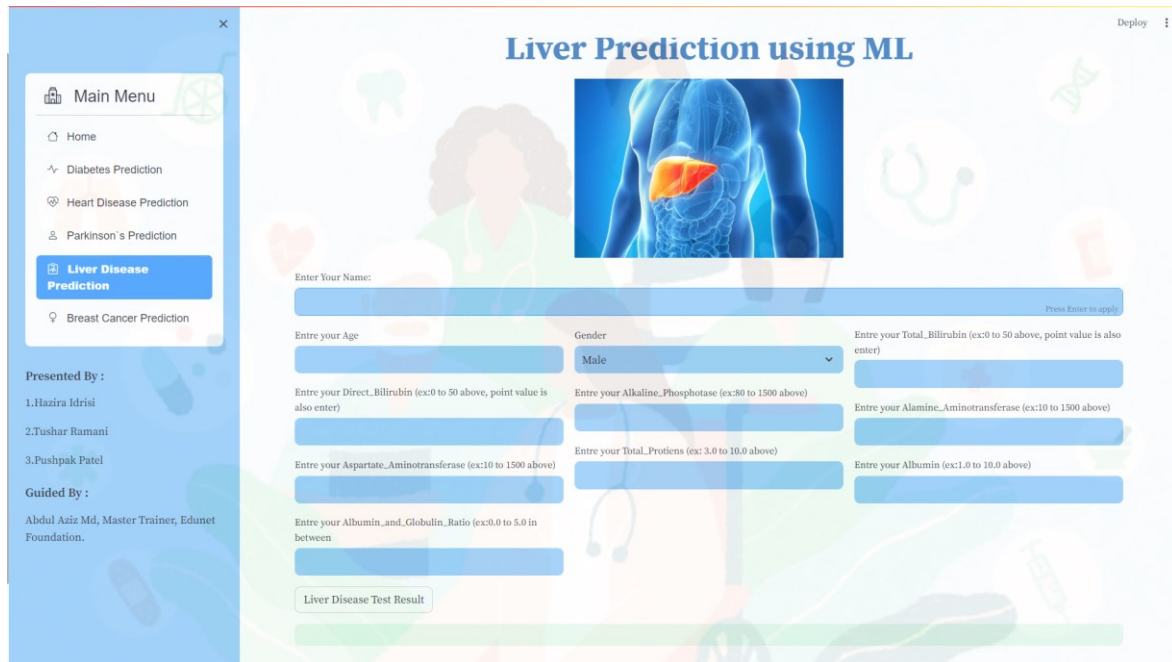


### 3.3.4 Parkinson's Disease Prediction

Figure 21 : Single User Interface (UI) of Parkinson's Disease Prediction

Figure 22 : Output for Parkinson's disease for positive case

### 3.3.5 Liver Disease Prediction



**Liver Prediction using ML**

Enter Your Name:  Press Enter to apply

Enter your Age:  Gender:

Enter your Direct\_Bilirubin (ex:0 to 50 above, point value is also enter):  Enter your Alkaline\_Phosphatase (ex:80 to 1500 above):  Enter your Total\_Bilirubin (ex:0 to 50 above, point value is also enter):

Enter your Aspartate\_Aminotransferase (ex:10 to 1500 above):  Enter your Total\_Proteins (ex: 3.0 to 10.0 above):  Enter your Alamine\_Aminotransferase (ex:10 to 1500 above):

Enter your Albumin\_and\_Globulin\_Ratio (ex:0.0 to 5.0 in between):  Enter your Albumin (ex:1.0 to 10.0 above):

Figure 23 : Single User Interface (UI) of Liver Disease Prediction



**Liver Prediction using ML**

Enter Your Name:

Enter your Age:  Gender:

Enter your Direct\_Bilirubin (ex:0 to 50 above, point value is also enter):  Enter your Alkaline\_Phosphatase (ex:80 to 1500 above):  Enter your Total\_Bilirubin (ex:0 to 50 above, point value is also enter):

Enter your Aspartate\_Aminotransferase (ex:10 to 1500 above):  Enter your Total\_Proteins (ex: 3.0 to 10.0 above):  Enter your Alamine\_Aminotransferase (ex:10 to 1500 above):

Enter your Albumin\_and\_Globulin\_Ratio (ex:0.0 to 5.0 in between):  Enter your Albumin (ex:1.0 to 10.0 above):

**NEGATIVE**

Congratulation Kush Halpati You don't have liver disease.

Figure 24 : Output for Liver disease for negative case

### 3.3.6 Breast Cancer Prediction

**Breast Cancer Prediction using ML**

Enter Your Name:

Enter your Radius Mean: 15.00

Enter your Radius SE: 1.00

Enter your Radius Worst: 20.00

Enter your Texture Mean: 20.00

Enter your Texture SE: 1.00

Enter your Texture Worst: 25.00

Enter your Perimeter Mean: 90.00

Enter your Perimeter SE: 10.00

Enter your Perimeter Worst: 120.00

Enter your Area Mean: 750.00

Enter your Area SE: 150.00

Enter your Area Worst: 1500.00

Enter your Smoothness Mean: 0.10

Enter your Smoothness SE: 0.01

Enter your Smoothness Worst: 0.15

Enter your Compactness Mean: 0.15

Enter your Compactness SE: 0.10

Enter your Compactness Worst: 0.30

Enter your Concavity Mean: 0.20

Enter your Concavity SE: 0.02

Enter your Concavity Worst: 0.40

Enter your Concave Points Mean: 0.10

Enter your Concave Points SE: 0.01

Enter your Concave Points Worst: 0.10

Enter your Symmetry Mean: 0.50

Enter your Symmetry SE: 0.50

Enter your Symmetry Worst: 0.50

Enter your Fractal Dimension Mean: 0.05

Enter your Fractal Dimension SE: 0.05

Enter your Fractal Dimension Worst: 0.10

Breast Cancer Test Result

Figure 25 : Single User Interface (UI) of Breast Cancer Prediction

## Streamlit Web App for Multiple Disease Prediction using Machine Learning

**Main Menu**

- Home
- Diabetes Prediction
- Heart Disease Prediction
- Parkinson's Prediction
- Liver Disease Prediction
- Breast Cancer Prediction**

**Presented By :**

- 1.Hazira Idrisi
- 2.Tushar Ramani
- 3.Pushpak Patel

**Guided By :**

Abdul Aziz Md, Master Trainer, Edunet Foundation.

**Enter Your Name:**  
Isha Prajapati

**Input Features:**

Feature	Value
Enter your Radius Mean	13.54
Enter your Radius SE	0.26
Enter your Radius Worst	15.11
Enter your Texture Mean	14.36
Enter your Texture SE	0.70
Enter your Texture Worst	15.20
Enter your Perimeter Mean	87.52
Enter your Perimeter SE	2.05
Enter your Perimeter Worst	99.09
Enter your Area Mean	546.21
Enter your Area SE	23.15
Enter your Area Worst	711.62
Enter your Smoothness Mean	0.09
Enter your Smoothness SE	0.01
Enter your Smoothness Worst	0.14
Enter your Compactness Mean	0.03
Enter your Compactness SE	0.02
Enter your Compactness Worst	0.30
Enter your Concavity Mean	0.05
Enter your Concavity SE	0.02
Enter your Concavity Worst	0.23
Enter your Concave Points Mean	0.09
Enter your Concave Points SE	0.01
Enter your Concave Points Worst	0.12
Enter your Symmetry Mean	0.28
Enter your Symmetry SE	0.12
Enter your Symmetry Worst	0.29
Enter your Fractal Dimension Mean	0.05
Enter your Fractal Dimension SE	0.01
Enter your Fractal Dimension Worst	0.07

**Breast Cancer Test Result**

**NEGATIVE**

Congratulation Isha Prajapati You don't have Breast Cancer.

Figure 26 : Output for Breast Cancer for negative case



### **3.4 Tools and Technologies:**

#### **3.4.1 Tools Used:**

- 1 Kaggle - Kaggle is a platform that provides access to diverse datasets
- 2 Google Colaboratory - Colaboratory is a data analysis and machine learning tool
- 3 Anaconda – Anaconda aims to simplify package management and deployment.
- 4 Spyder IDE - An open-source cross-platform integrated development environment

#### **3.4.2 Technologies Used:**

- 1 Python - Python is dynamically typed, high-level, general-purpose programming language.
- 2 NumPy - A library for the Python. adding support for large, multi-dimensional arrays & matrices
- 3 Pandas – A software library written for the Python programming language for data manipulation and analysis.
- 4 Sklearn - A free software machine learning library for the Python programming language.
- 5 Machine Learning Algorithms - Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output.
- 6 Pickle - Python pickle module is used for serializing and de-serializing a Python object structure.
- 7 Stream Lit - A free, Open-source framework to rapidly build and share beautiful machine learning web apps



## **CHAPTER 4**

## **CONCLUSION**

## **CHAPTER 4**

### **CONCLUSION**

The main objective of this project was to create a system that would predict more than one disease and do so with high accuracy. Because of this project the user doesn't need to traverse different websites which saves time as well. Diseases if predicted early can increase your life expectancy as well as save you from financial troubles. For this purpose, we have used various machine learning algorithms like Random Forest, XGBoost, and Logistic Regression to achieve maximum accuracy.

## **CHAPTER 5**

### **FUTURE SCOPE**

## **CHAPTER 5**

### **FUTURE SCOPE**

- In the future we can add more diseases in the existing API.
- We can try to improve the accuracy of prediction in order to decrease the mortality rate.
- Try to make the system user-friendly and provide a chatbot for normal queries.

## **GITHUB LINK**

<https://github.com/TusharRamani/Multiple-disease-app>

## REFERENCES

- [1]. Kaggle: Kaggle website. <https://www.kaggle.com/>
- [2]. Streamlit: Streamlit Documentation. <https://docs.streamlit.io/>
- [3]. Google Colab : <https://colab.google/>
- [4]. <https://github.com/>
- [5]. <https://chat.openai.com/>
- [6]. <https://youtube.com/>
- [7]. <https://www.javatpoint.com/>
- [8]. <https://www.csias.in/explain-the-step-by-step-implementation-of-xgboost-algorithm/>
- [9]. <https://www.smartdraw.com/>
- [10]. <https://staruml.io/>