



Department of Computer Engineering  
Rizvi College of Engineering

# Multiple Disease Prediction Webapp

Submitted in partial fulfillment of the requirements  
of the Mini-Project 1 for Third Year of  
Bachelors of Engineering

By

Danish Khan (UIN: 201P027)

Muzaffar Khan (UIN: 201P007)

Soham Manjrekar (UIN: 201P018)

Danish Jamadar (UIN: 201P005)

Guide:

PROF. MOHAMMED JUNED



University of Mumbai

2022-2023

# CERTIFICATE

This is to certify that the mini-project entitled “**Multiple Disease Prediction System**” is a bonafide work of “**Danish Khan, Muzaffar Khan, Soham Manjrekar, Danish Jamadar**” submitted to the University of Mumbai in partial fulfillment of the requirement for the Mini-Project 1 for Third Year of the Bachelor of Engineering in “**Computer Engineering**”.

---

Prof. Mohammed Juned  
**Guide**

---

Prof. Shiburaj Pappu  
**Head of Department**

---

Dr. Varsha Shah  
**Principal**

# Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

-----  
(Signature)

-----  
(Name of student and Roll No.)

Date:

# ABSTRACT

Healthcare falls under the essential conveniences to be given to the society. Many of the current AI models for medical services examination are focusing on one disease prediction for each analysis. Our point is to anticipate the various sorts of illness in single stage by utilizing inbuilt python module Streamlit. In this task we are utilizing Naïve Bayes algorithm, random forest, decision tree and svm classifier are utilized for prediction of a particular disease. The calculation which gives more accuracy is used to train the data set before implementation. To implement multiple disease analysis used machine learning algorithms, Streamlit and python pickling is utilized to save the model behavior. In this article we analyze Diabetes analysis, Heart disease, jaundice disease, hepatitis, liver disease and Parkinson's disease by using some of the basic parameters such as Pulse Rate, Cholesterol, Blood Pressure, Heart Rate, etc., and also the risk factors associated with the disease can be found using prediction model with good accuracy and Precision. Further we can include other kind of chronic diseases, skin diseases and many other. In this work, demonstrated that using only core health parameters many diseases can be predicted. The significance of this analysis to analyses the maximum diseases to screen the patient's condition and caution the patients ahead of time to diminish mortality proportion. To implement multiple disease analysis used machine learning algorithms, Streamlit. We have considered six diseases for now that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes and in the future, many more diseases can be added. The user has to enter various parameters of the disease and the system would display the output whether he/she has the disease or not. This project can help a lot of people as one can monitor the persons' condition and take the necessary precautions thus increasing the life expectancy.

**Keywords:** Diabetes, Heart, Parkinson, Machine Learning.

# Index

<b>Sr. No</b>	<b>Title</b>	<b>Page No</b>
<b>1.</b>	Introduction	6
<b>2.</b>	Review and Literature	8
2.1.	Introduction	8
2.2.	Survey of existing system	8
2.3.	Problems of existing system	9
<b>3.</b>	Theory, Methodology and Algorithm	10
3.1	Flowchart	10
3.2	Details of Hardware and Software	10
3.3	Methodology	11
3.4	ML System Architecture	12
3.5	Proposed Method	12
3.6	Approach	13
3.7	Correlation matrix	19
<b>4.</b>	Implementation Plan	22
4.1	Gantt Chart	22
4.2	Cost of project	22
4.3	ML Algorithms	22
<b>5.</b>	Conclusion	24
<b>6.</b>	References	24
<b>7.</b>	Acknowledgement	26
<b>8.</b>	Publication	27

# Chapter 1

## Introduction

In this digital world, data is an asset, and enormous data was generated in all the fields. Data in the healthcare industry consists of all the information related to patients. Here a general architecture has been proposed for predicting the disease in the healthcare industry. Many of the existing models are concentrating on one disease per analysis. Like one analysis for diabetes analysis, one for cancer analysis, one for skin diseases like that. There is no common system present that can analyze more than one disease at a time. Thus, we are concentrating on providing immediate and accurate disease predictions to the users about the symptoms they enter along with the disease predicted. So, we are proposing a system which used to predict multiple diseases by using Streamlit. In this system, we are going to analyze Diabetes, Heart, and Parkinson disease analysis. Later many more diseases can be included. To implement multiple disease prediction systems, we are going to use machine learning algorithms, and Streamlit. Python pickling and SVM is used to save the behavior of the model. The importance of this system analysis is that while analyzing the diseases all the parameters which cause the disease is included so it is possible to detect the disease efficiently and more accurately. The final model's behavior will be saved as a python pickle file or sav file.

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. We are taking six diseases that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes. As all the six diseases are correlated to each other. To implement multiple disease analyses we are going to use machine learning algorithms and Streamlit. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Our Model will invoke the corresponding model and returns the status of the patient. Our basic idea is to develop a system which will predict and give the details of the disease predicted along with its severity which as symptoms are given as input by the user. The system will compare the symptoms with the datasets provided in the database. If the symptom matches the datasets, then it should ask other relevant symptoms specifying the name of the symptom. If not, the symptom entered should be notified as wrong symptom. After this a prompt will come up asking whether you want to still save the symptom in the database. If you click on yes, it will be saved in the database, if not it will go to the recycle bin. The main feature will be the machine learning, in which we will be using algorithms such as Naïve Bayes Algorithm, K-Nearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will predict accurate disease and also, will find which algorithm gives a faster and efficient result by comparatively-comparing.

GitHub Project Link: <https://github.com/sohammanjrekar/rcoe22-sem5-group2>

## 1.1. AIMS AND OBJECTIVES

The main objective of the study is to develop a Multiple Disease Prediction Webapp. The system aims to achieve the following objectives:

- To design a Multiple Disease Prediction Webapp system.
- The website will provide range of the values during the prediction of the disease.
- To predict more than one disease at a time

Main objective behind to develop a system helps the doctors to cross verify their diagnosed results which gives promising solution over existing death rates. By using our proposed work try to invent unique platform and most promising solution for early diagnosis of multiple diseases. Existing work analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease wrong. So, we are giving more accurate solution by using machine learning and Convolutional neural network to detect diseases and make predictions.

## 1.2. LIST OF ABBREVIATIONS

Terminology	Meaning
<b>XG Boost</b>	Extreme Gradient Boost
<b>SVM</b>	Support Vector Machine

# **Chapter 2**

## **Review of Literature**

### **2.1. INTRODUCTION**

Literature review is an expressive study based on the detailed review of earlier pertinent studies related to the various concepts of Disease prediction. It highlights the status of online multiple disease prediction, importance and problems of disease prediction, factors affecting disease prediction.

### **2.2. SURVEY OF EXISTING SYSTEM**

A lot of analysis over existing systems in the health care industry considered only one disease at a time. For example, one system is used to analyze diabetes, another is used to analyses diabetes retinopathy, and another system is used to predict heart disease. Maximum systems focus on a particular disease. When an organization wants to analyses their patient's, health reports then they have to deploy many models. The approach in the existing system is useful to analyses only particular diseases. In multiple diseases prediction system, a user can analyze more than one disease on a single website. The user doesn't need to traverse different places in order to predict whether he/she has a particular disease or not. In multiple diseases prediction system, the user needs to select the name of the particular disease, enter its parameters and just click on submit. The corresponding machine learning model will be invoked and it would predict the output and display it on the screen [1].

In the existing system the data set is typically small, for patients and diseases with specific conditions. These systems are mostly designed for the more colossal diseases such as Heart Disease, Cancer etc. The pre-selected characteristics may sometimes not satisfy the changes in the disease and its influencing factors which could lead to inaccuracy in results. As we live in continuously evolving world, the symptoms of diseases also evolve over a course of time. Also, most of the current systems make the users wait for long periods by making them answer lengthy questionnaires [2].

### **2.3 PROBLEMS OF EXISTING SYSTEMS**

Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. For example, first is for liver analysis, one for cancer analysis, one for lung 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 analyse 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.

## **2.4. THE SOLUTION**

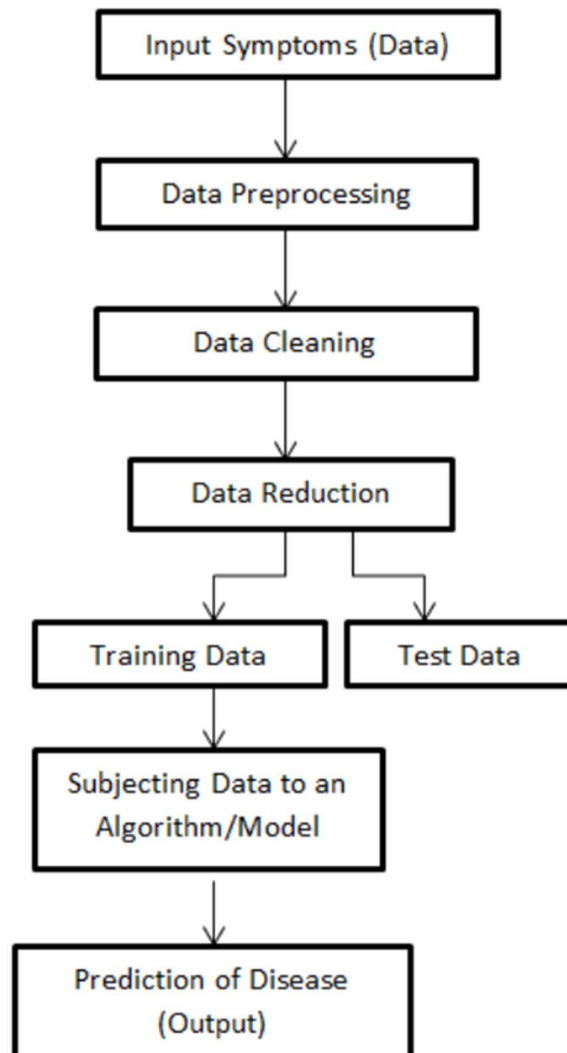
In multiple disease prediction, it is possible to predict more than one disease at a time. So, the user doesn't need to traverse different sites in order to predict the diseases. We are taking six diseases that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes. As all the six diseases are correlated to each other. To implement multiple disease analyses we are going to use machine learning algorithms. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name.

The exact examination of clinical data set advantages in early illness expectation, patient consideration and local area administrations. The methodology of Machine Learning (ML) has been effectively utilized in grouped technologies including Disease forecast. The objective of generating classifier framework utilizing Machine Learning (ML) models is to massively assist with addressing the well-being related issues by helping the doctors to foresee and analyze illnesses at a beginning phase. Sample information of 4920 patient's records determined to have 41 illnesses was chosen for examination. A reliant variable was made out of 41 sicknesses. 95 of 132 autonomous variables (symptoms) firmly identified with infections were chosen and advanced. This examination work completed shows the illness expectation framework created utilizing Machine learning calculations like XG Boost.

# Chapter 3

## Theory, Methodology

### 3.1. FLOWCHART



### 3.2. DETAILS OF HARDWARE & SOFTWARE

Project Prerequisites

We will use the following technologies:

- I. Used Technology: GitHub, VS Code, Xampp.
- II. Operating System: Windows 7 and above
- III. Browser: Any browser and IE 8 and above.

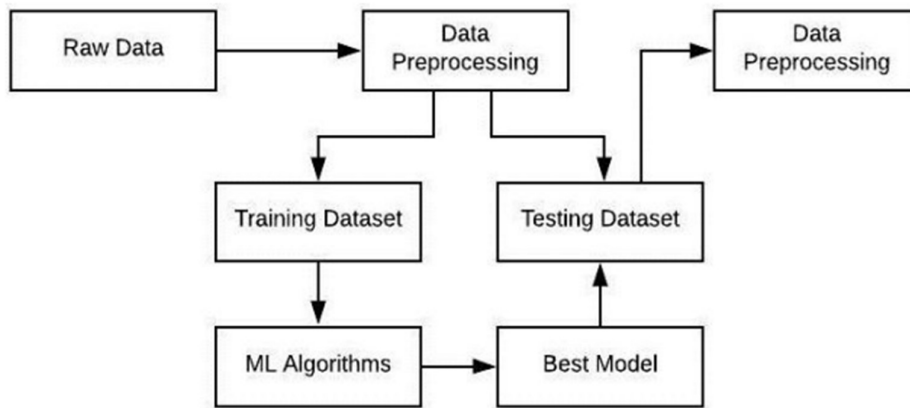
- IV. Processor: A single-core 2GHz processor
- V. RAM: 512 Mb and above

### 3.3 METHODOLOGY

The dataset we have considered comprises of 132 indications, the blend or stages of which leads to 41 illnesses. In light of the 4920 documents of various patient samples, mainly to point foster a forecast algorithm that considers in the side effects of various client and forecasts the sickness that the person is bound to be affected.

- A. Inputs (Patient Symptoms): When planning the algorithm, we have expected to be the client can have an unmistakable thought regarding the indications he is encountering. The Prediction created considers 95 manifestations in the midst of which the client can permit the indications his preparing as the input.
- B. Data pre-processing: The mining of the data's approaches that changes the crude information or then again encrypts the information to form a structure so that it can be effectively deciphered with the help of calculation is known as information pre-processing. The information pre-processing strategies utilized in the introduced work which listed as follows:
  - 1) Data Purification: Data is purified using certain measures like stuffing in lost worth, along these lines settling the irregularities in the information.
  - 2) Data Reduction: The examination turns out to be hard when managing gigantic information base. Thus, we kill those autonomous variables (symptoms) which may not affect the objective variables (diseases). So that in the progress task, which of around 95 of 132 side effects firmly identified with the illnesses are chosen.
- C. Models: The entire system is designed in such a way to predict the diseases by utilizing the three Algorithms i.e., Decision Tree model, LightGBM model and Random Forest classifier model, so that the predictive analysis study is proposed at the end of the study by exploring its speed, efficiency and performance of the various algorithms for the input dataset.
- D. Output(diseases): While a framework is made with the preparation set utilizing the validated calculations standard datasets are shaped and whenever the client indications are provided as a contribution as input of the algorithm, and the side effects are composed agreeing as the standard dataset created, accordingly creating arrangements and foreseeing the high probable infection.

### 3.4 ML SYSTEM ARCHITECTURE



As shown in the above figure, the raw data from the original dataset is passed onto the first phase i.e., Data pre-processing. In Data pre-processing this raw data is then cleaned of all redundancies, missing values etc. The new clean data is fit for training different algorithmic models on it. The process of training models is fundamental process in Machine learning Projects. There are two approaches to machine learning mainly Supervised Learning and Unsupervised Learning. Our model mostly applies the first approach initially. i.e., Supervised Learning. Now in Supervised Learning, the system is trained on some examples i.e., Training set and then the model is asked to predict new values based on the test set. The partitioning of dataset becomes crucial for getting good accuracy in models. The percentage mostly used while partitioning is 80/20 .i.e., 80% for training and 20% for testing purposes. In our system we aim at first applying different algorithms on the training dataset and based on the model's Confidence and testing dataset accuracy, we select the best model algorithm and apply it on testing dataset to generate accurate results.

### 3.5. PROPOSED METHOD

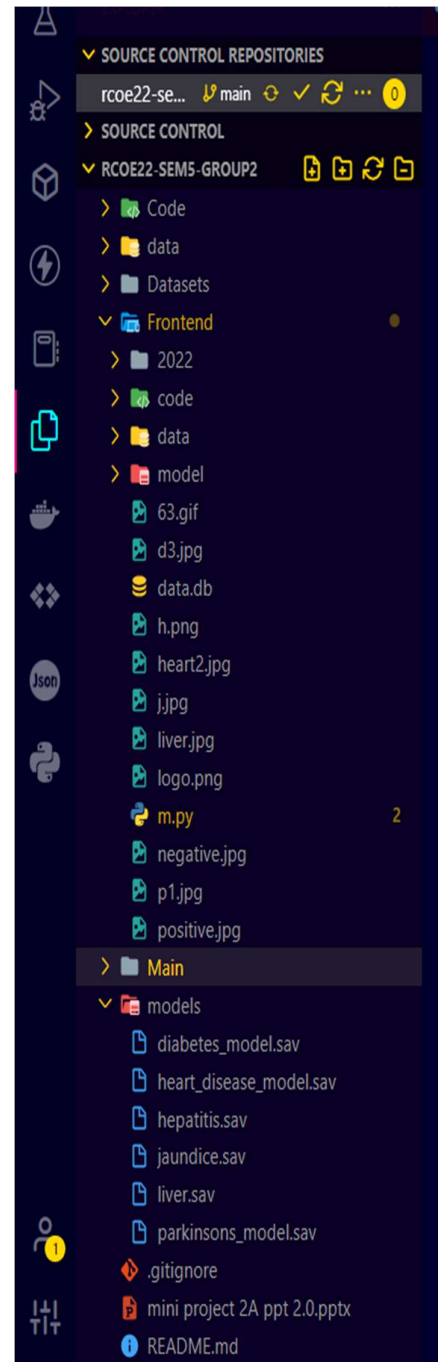
We are proposing such a system which will flaunt a simple and elegant User Interface and also be time efficient. In order to make it less time consuming we are aiming at a more specific questionnaire which will be followed by the system. Our aim with this system is to be the connecting bridge between doctors and patients. The main feature will be the machine learning, in which we will be using algorithms such as Naïve Bayes Algorithm, K-Nearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will help us in getting accurate predictions and also, will find which algorithm gives a faster and efficient result by comparatively-comparing. Another feature that our system will comprise of is Doctor's Consultation. After delivering the results, our system will also suggest the user to get a doctor's consultation on this

report. By using this feature, we will not only address the other class of users i.e., the Doctors but we will also gain their trust in this system as in that this system is not affecting their business.

### 3.6. APPROACH

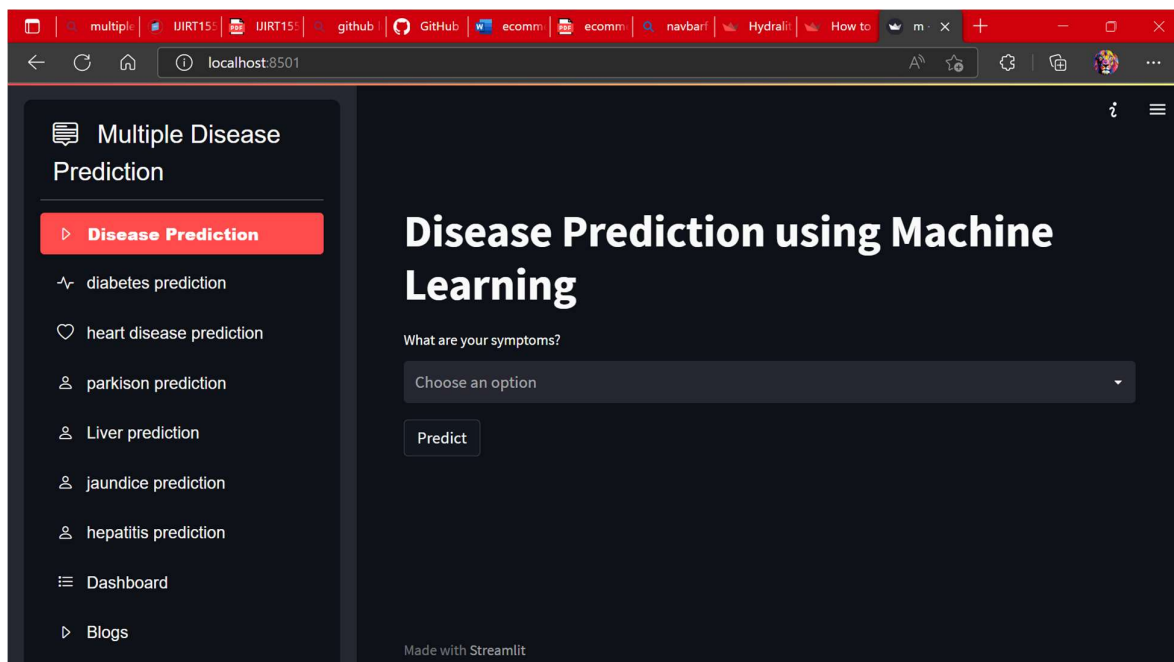
- Create Normal Project: Open the IDE and create a normal project by selecting File -> New Project.
- Reading the dataset: Firstly, we will be loading the dataset from the folders using the pandas library. While reading the dataset we will be dropping the null column. This dataset is a clean dataset with no null values and all the features consist of 0's and 1's. Whenever we are solving a classification task it is necessary to check whether our target column is balanced or not. We will be using a bar plot, to check whether the dataset is balanced or not.
- Splitting the data for training and testing the model. Now that we have cleaned our data by removing the Null values and converting the labels to numerical format, it's time to split the data to train and test the model. We will be splitting the data into 80:20 format i.e., 80% of the dataset will be used for training the model and 20% of the data will be used to evaluate the performance of the models. Check Python3 version: `python3 --version`
- Model Building: After splitting the data, we will be now working on the modeling part. We will be using K-Fold cross-validation to evaluate the machine learning models. We will be using Support Vector Classifier, Gaussian Naive Bayes Classifier, and Random Forest Classifier for cross-validation. Before moving into the implementation part let us get familiar with k-fold cross-validation and the machine learning models.

- 1) K-Fold Cross-Validation: K-Fold cross-validation is one of the cross-validation techniques in which the whole dataset is split into k number of subsets, also known as folds, then training of the model is performed on the k-1 subsets and the remaining one subset is used to evaluate the model performance.

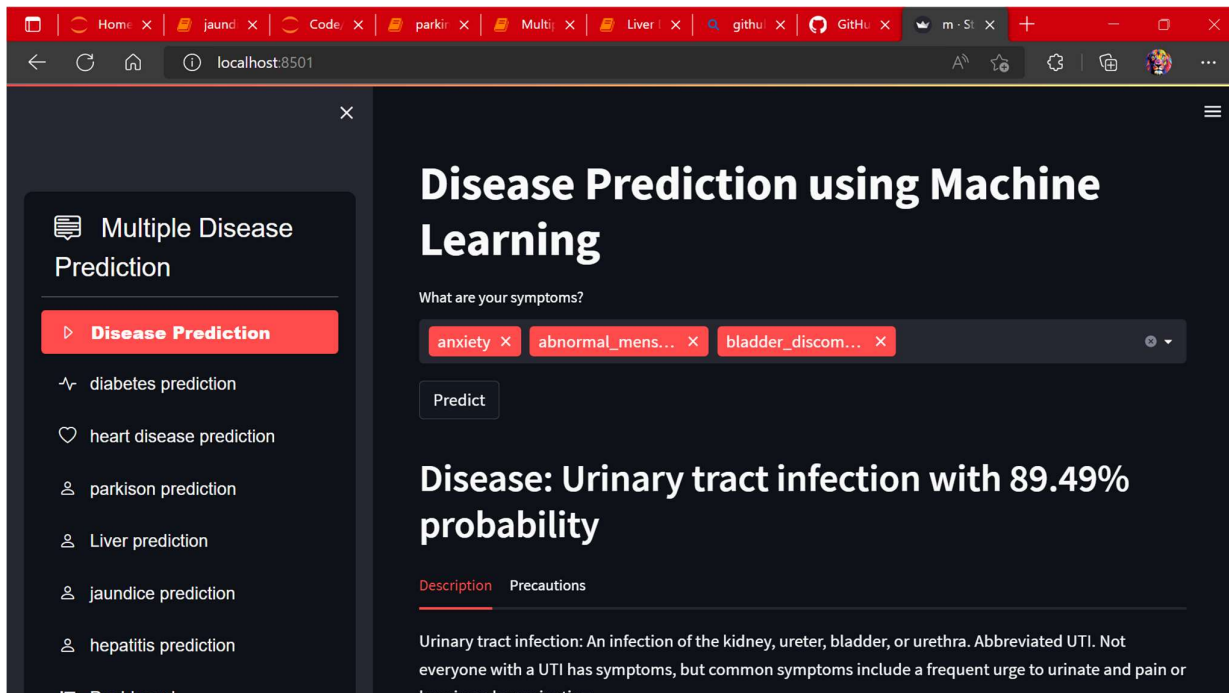


- 2) **Support Vector Classifier:** Support Vector Classifier is a discriminative classifier i.e., when given a labeled training data, the algorithm tries to find an optimal hyperplane that accurately separates the samples into different categories in hyperspace.
  - 3) **Gaussian Naive Bayes Classifier:** It is a probabilistic machine learning algorithm that internally uses Bayes Theorem to classify the data points.
  - 4) **Random Forest Classifier:** Random Forest is an ensemble learning-based supervised machine learning classification algorithm that internally uses multiple decision trees to make the classification. In a random forest classifier, all the internal decision trees are weak learners, the outputs of these weak decision trees are combined i.e., mode of all the predictions is as the final prediction.
- **Create streamlit app:**  
Let's install streamlit. Type the following command in the command prompt.  
`pip install streamlit`
  - **Run streamlit code:**  
Open command prompt or Anaconda shell and type  
`streamlit run filename.py`

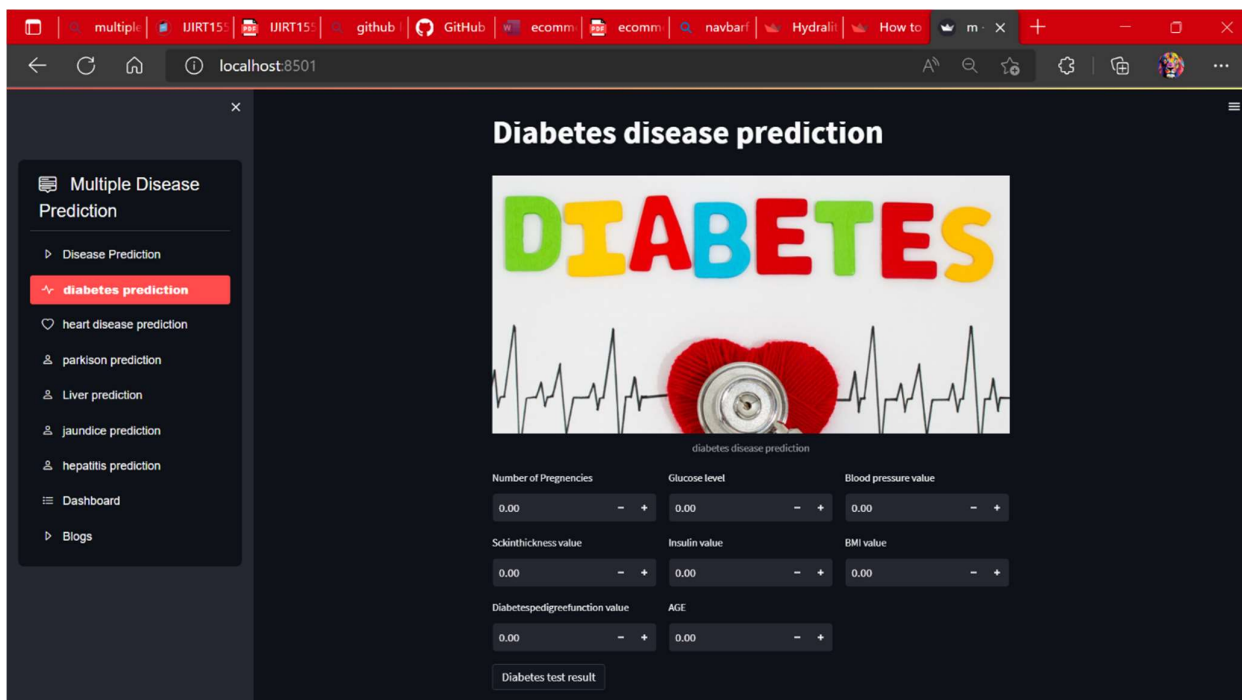
### 3.6.1. USER INTERFACE DESIGN



### 3.6.2 MULTIPLE DISEASE PREDICTION USING SYMPTOMS



### 3.6.3. DIABETES PREDICTION



### 3.6.4. HEART DISEASE PREDICTION


multiple | UIRT15 | UIRT15 | github | GitHub | ecomm | ecomm | navbar | Hydrall | How to | m · X | +

← ↻ 🏠 ⓘ localhost:8501 🔍 ⚙️ 📁 👤 ⋮

Multiple Disease Prediction

- ▶ Disease Prediction
- ✚ diabetes prediction
- ♥ heart disease prediction
- ⌘ parison prediction
- ⌘ Liver prediction
- ⌘ jaundice prediction
- ⌘ hepatitis prediction
- ☰ Dashboard
- ▶ Blogs

## Heart disease prediction



heart failure

AGE	sex	cp value
19.00 - +	0.00 - +	0.00 - +
trestlgs value	chol value	fbs value
0.00 - +	0.00 - +	0.00 - +
restng value	thalach value	exang value
0.00 - +	0.00 - +	0.00 - +
oldpeak value	slope value	ca value
0.00 - +	0.00 - +	0.00 - +
Thal value		
0.00 - +		

Heart test result

### 3.6.5. PARKINSON'S PREDICTION

multiple

UIRT15

github

GitHub

ecom

ecom

navbar

Hydrati

How to

m

+

localhost:8501

Multiple Disease Prediction

Disease Prediction

diabetes prediction

heart disease prediction

**parkinson prediction**

Liver prediction


jaundice prediction

hepatitis prediction

Dashboard

blogs

Parkinson prediction

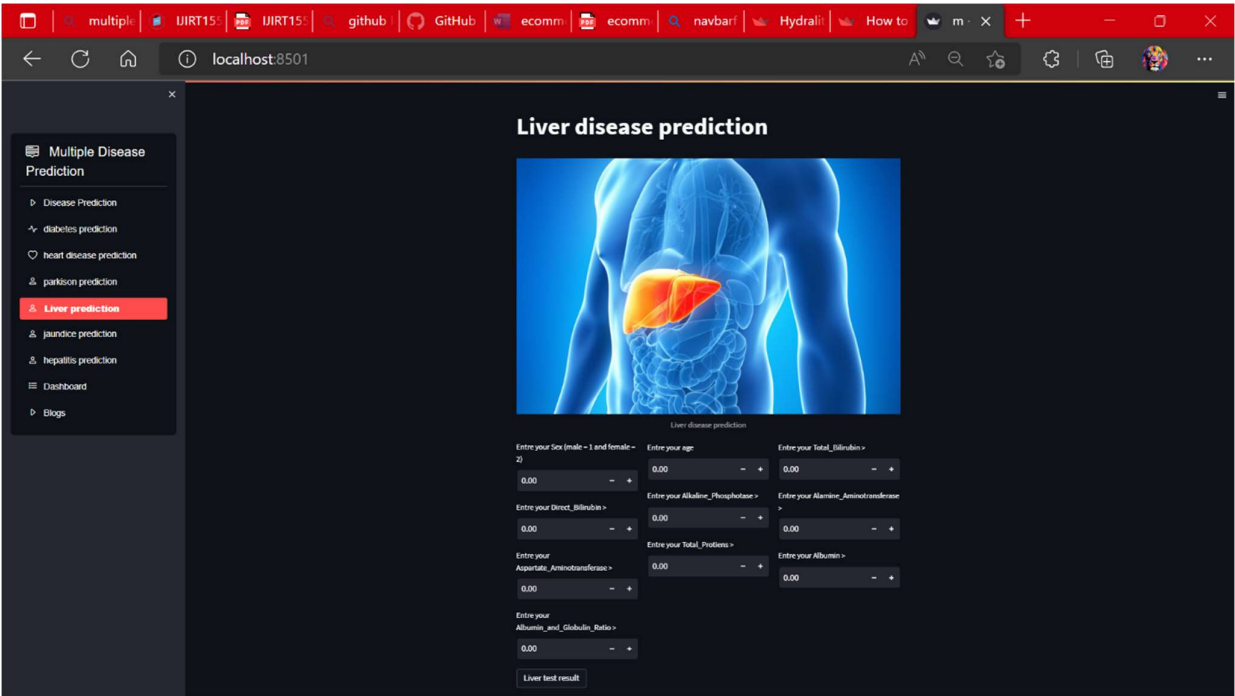


performance disease

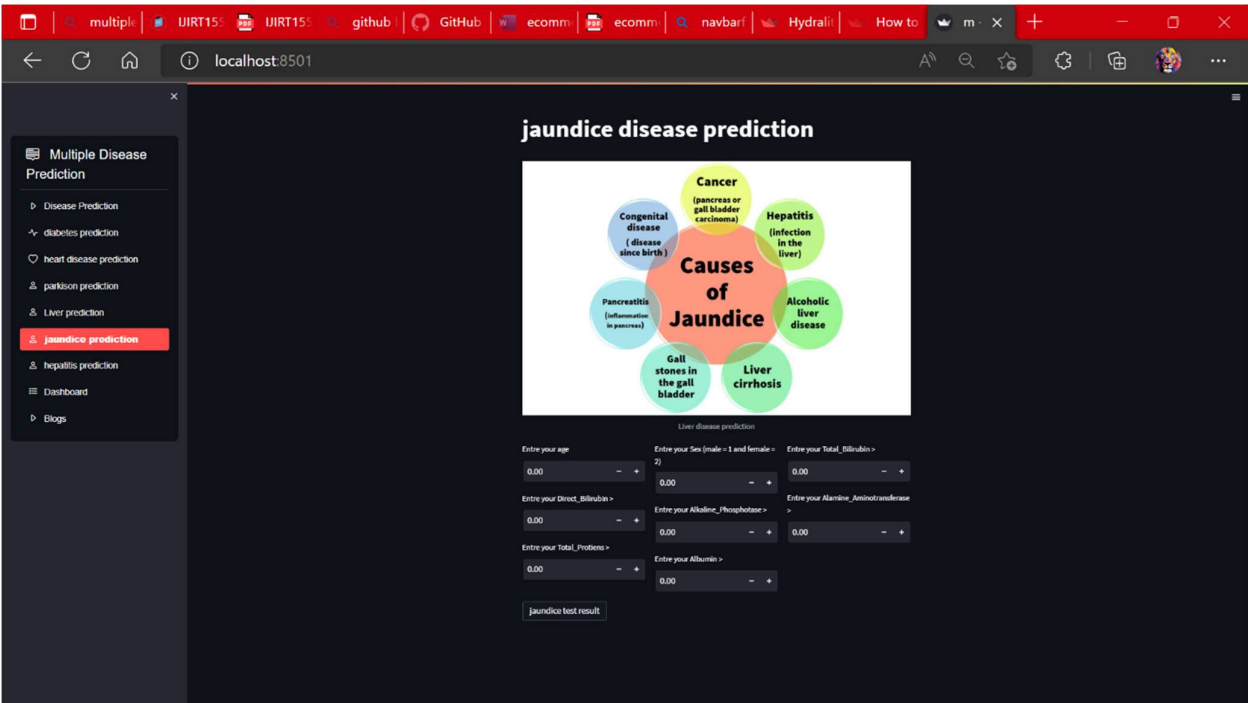
MNPF-Fu(N)	MNPF-Fu(N)	MNPF-Fu(N)
0.00 - +	0.00 - +	0.00 - +
MNPF-Jitter(N)	MNPF-Jitter(N)	MNPF-RAP
0.00 - +	0.00 - +	0.00 - +
MNPF-Skinner	MNPF-SPQ	Jitter-SDP
0.00 - +	0.00 - +	0.00 - +
Shimmer-APQ5	MNPF-Shimmer(B)	Shimmer-APQ3
0.00 - +	0.00 - +	0.00 - +
HR	MNPF-APQ	Shimmer-DXA
0.00 - +	0.00 - +	0.00 - +
spread1	HR	DFA
0.00 - +	0.00 - +	0.00 - +



3.6.6. LIVER DISEASE PREDICTION



3.6.7. JAUNDICE DISEASE PREDICTION

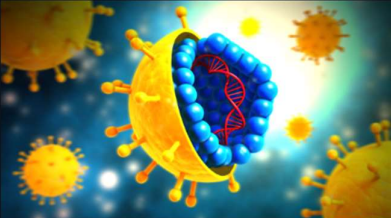


### 3.6.8. HEPATITIS PREDICTION

multiple | UJRT15 | UJRT15 | github | GitHub | ecomm | ecomm | navbar | Hydrall | How to | m · X

localhost:8501

## hepatitis disease prediction



hepatitis disease prediction

Enter your age: 0.00

Enter your gender: 0.00

Enter your Total\_Bilirubin: 0.00

Enter your Direct\_Bilirubin: 0.00

Enter your Alkaline\_Phosphatase: 0.00

Enter your Alanine\_Aminotransferase: 0.00

Enter your Aspartate\_Aminotransferase: 0.00

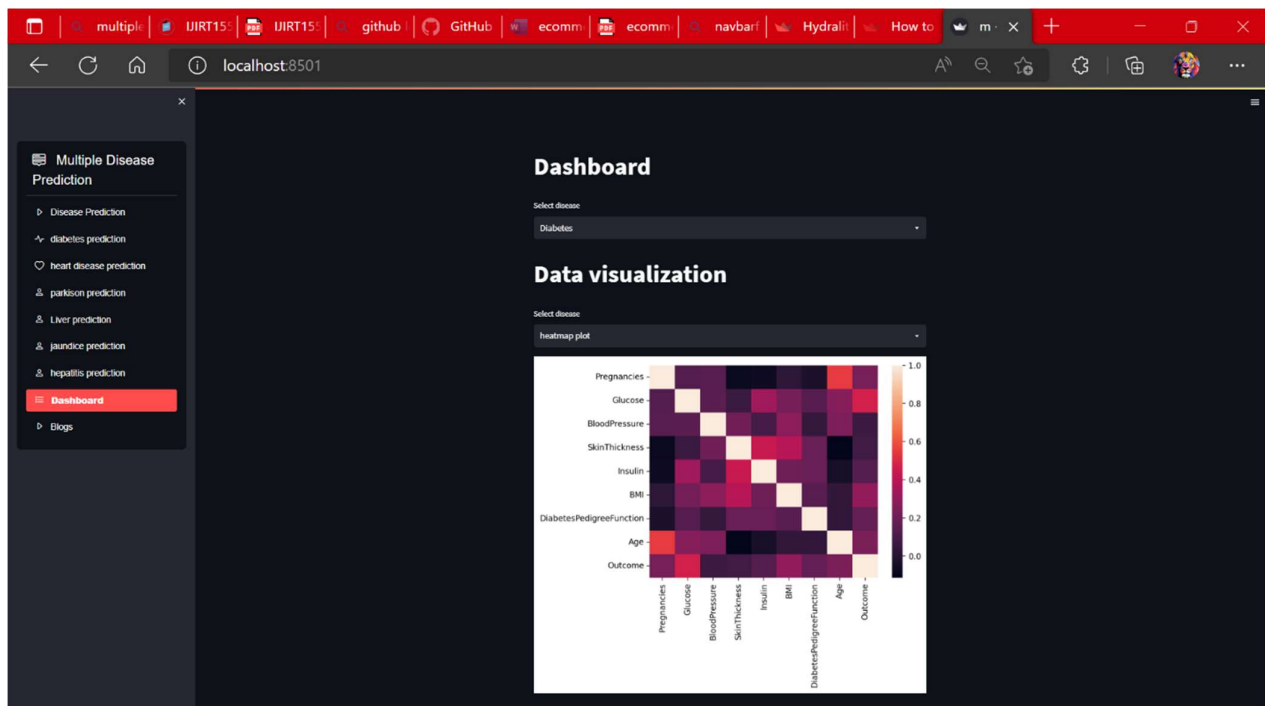
Enter your Total\_Proteins: 0.00

Enter your Albumin: 0.00

Enter your Albumin\_and\_Globulin\_Ratio: 0.00

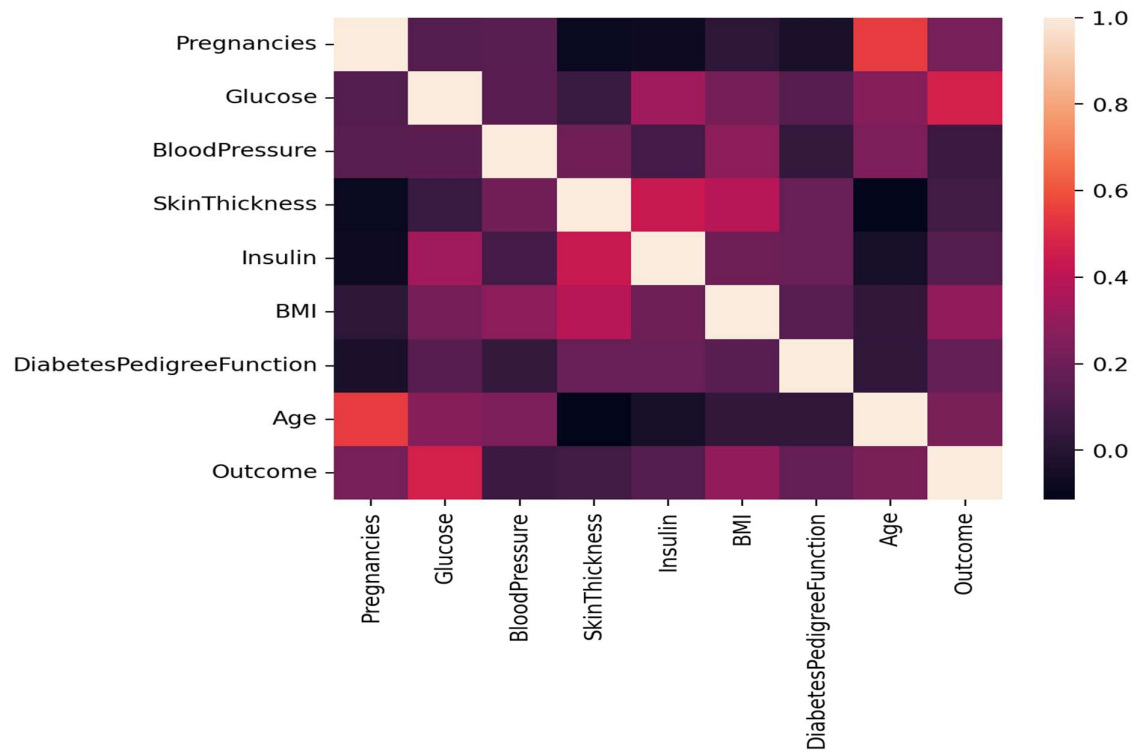
Liver test result

### 3.6.9. DASHBOARD

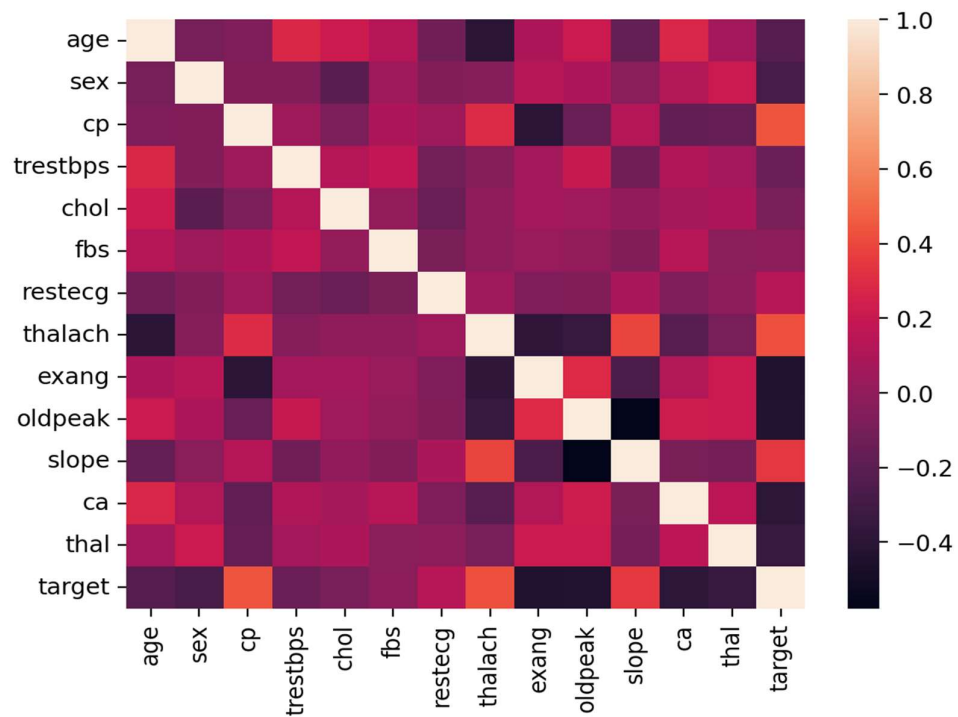


### 3.7. CORRELATION MATRIX

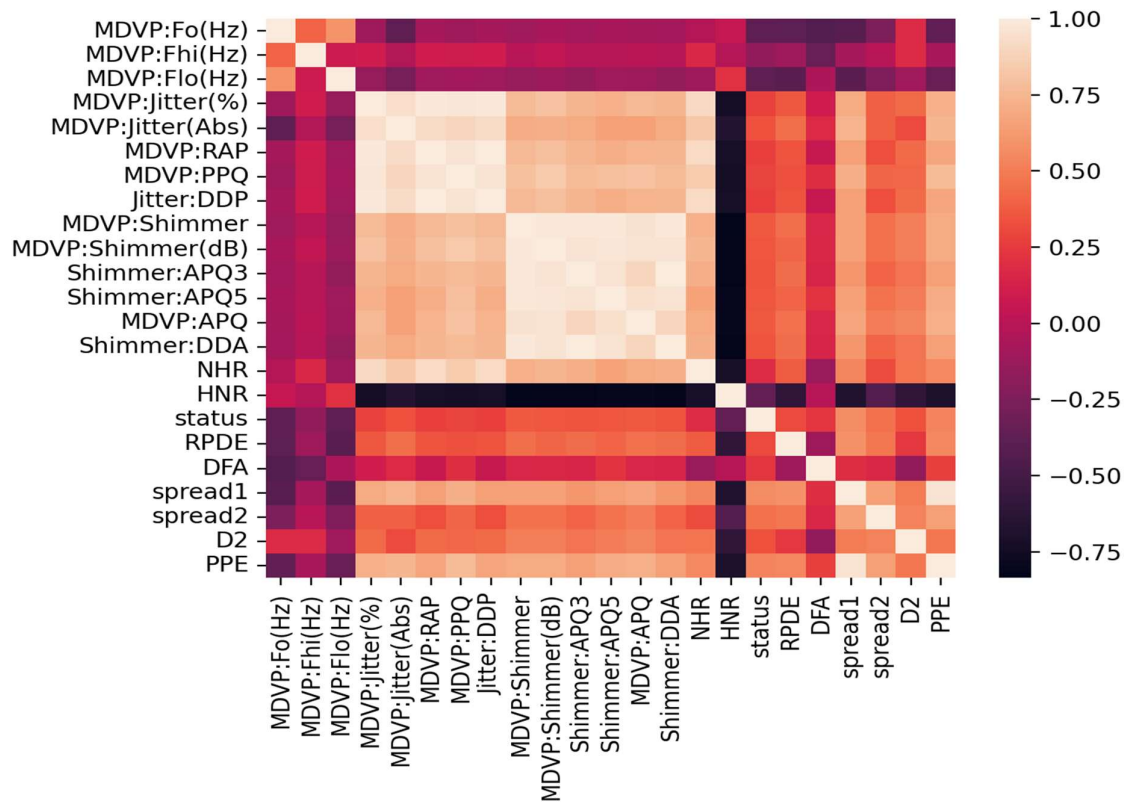
#### 3.7.1. DIABETES CORRELATION MATRIX



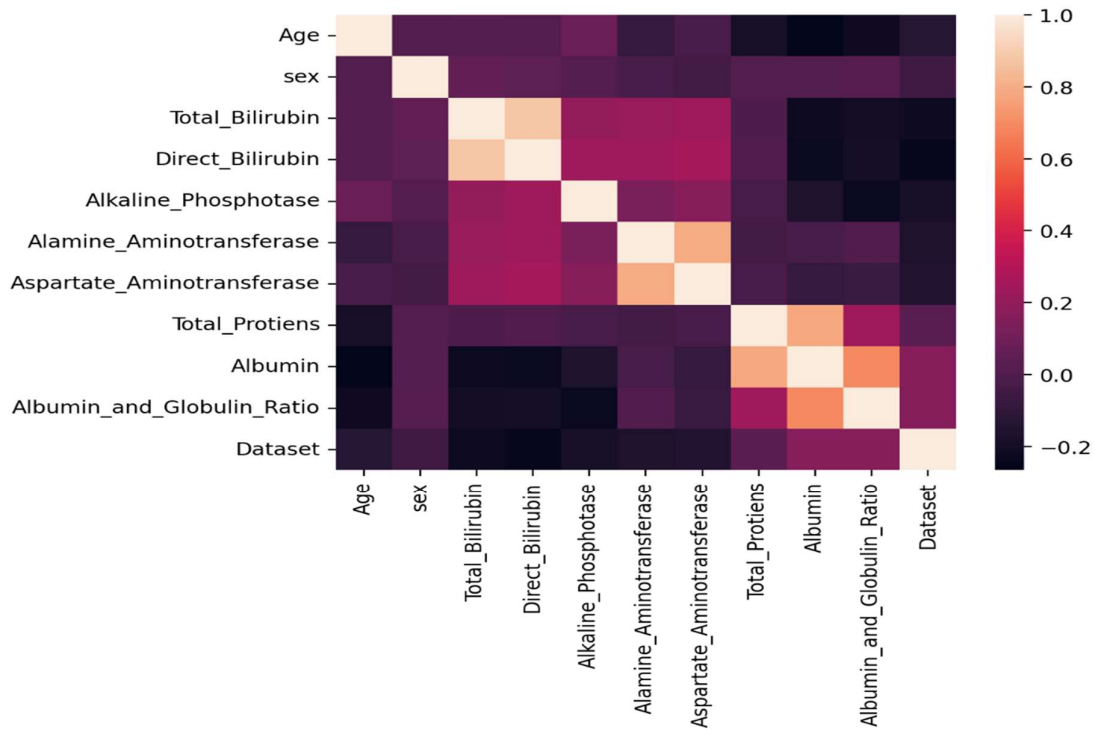
#### 3.7.2. HEART DISEASE CORRELATION MATRIX



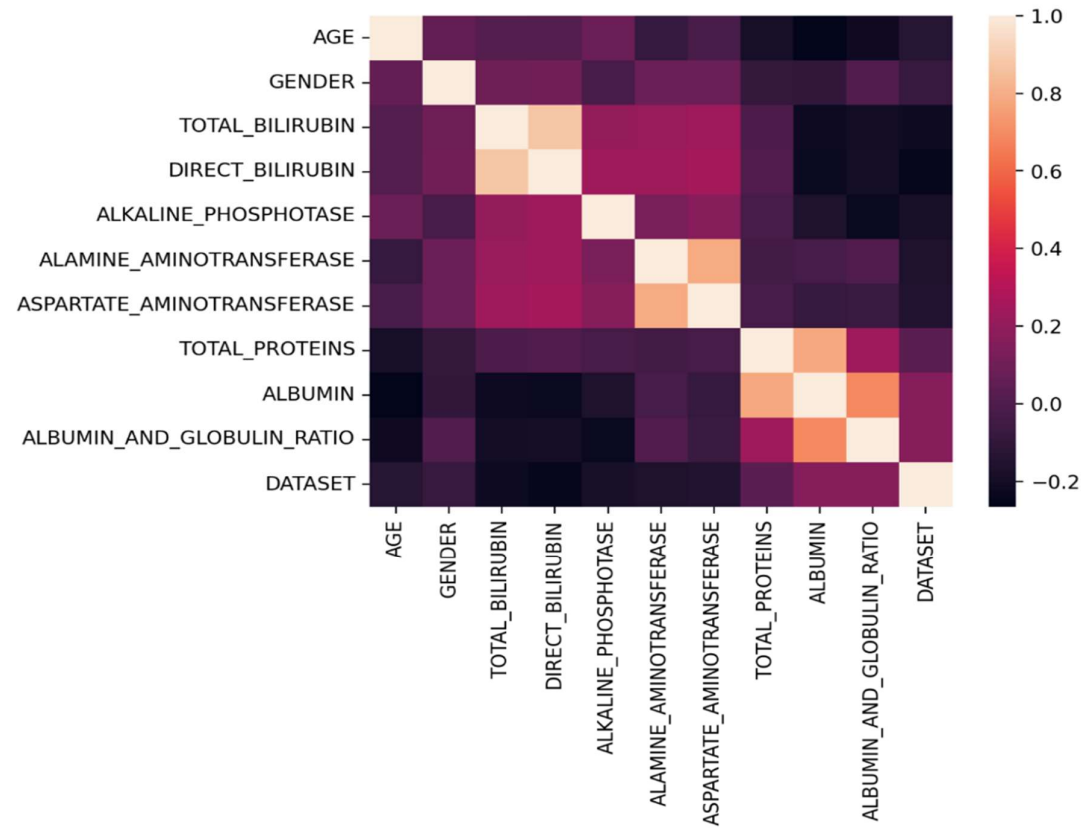
### 3.7.3. PARKINSON'S DISEASE CORRELATION MATRIX



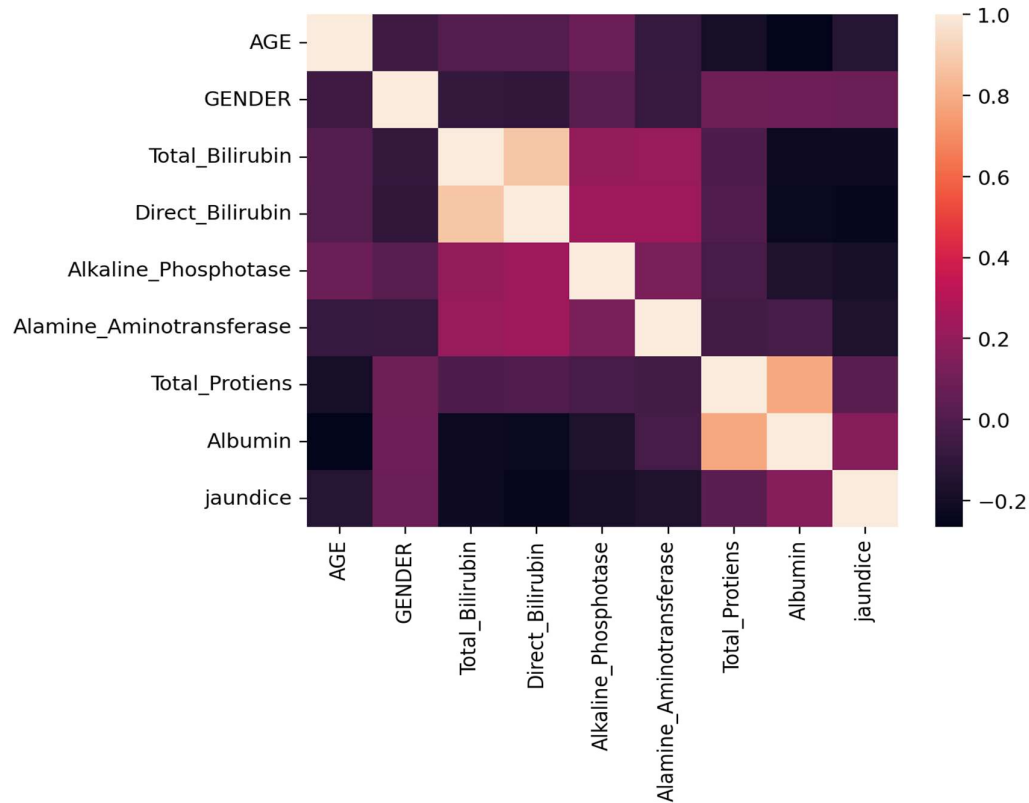
### 3.7.4. LIVER DISEASE CORRELATION MATRIX



3.7.5. HEPATITIS DISEASE CORRELATION MATRIX



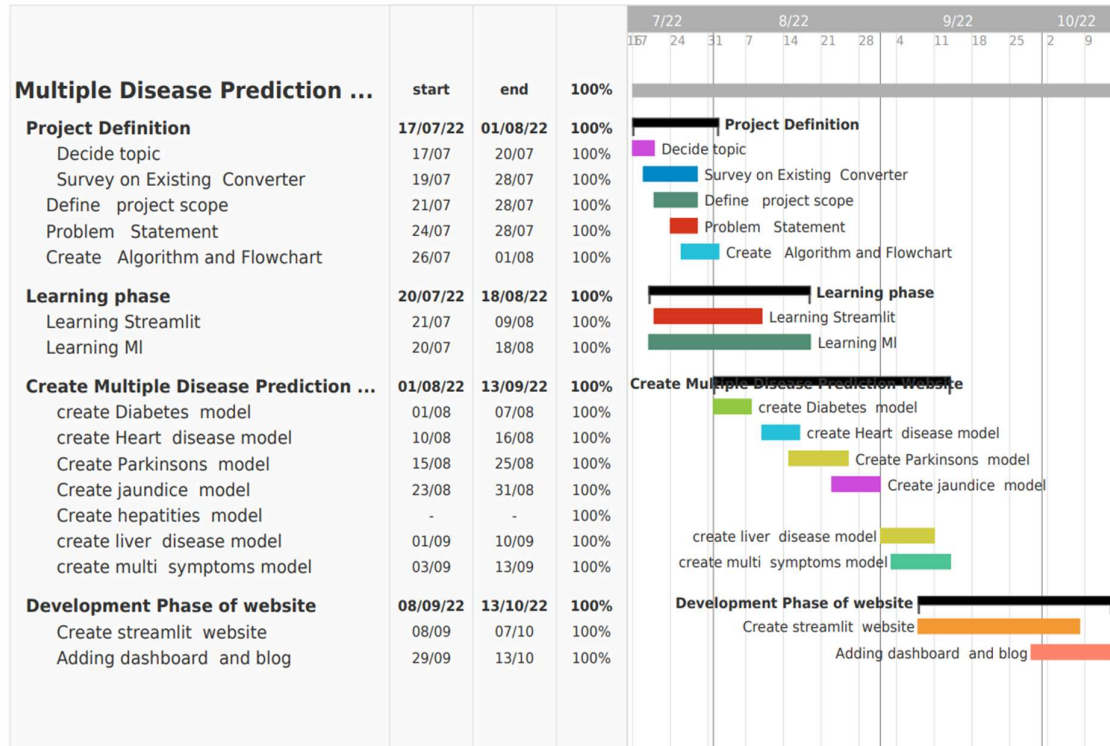
3.7.6. JAUNDICE DISEASE CORRELATION MATRIX



# Chapter 4

## Implementation Plan

### 4.1. GANTT CHART



### 4.2. COST OF PROJECT

Cost of project is approx. Rs. 0 because we use free hosting and domain no buy any template or logo. But in future upgrade cost of website is follows:

Domain Name	Rs. 199-1000
Website Hosting	Rs. 2000-10000
Logo	Rs. 500-6000
Images	Rs. 500-2000

### 4.3. ML ALGORITHMS

#### 4.3.1. Random Forest Algorithm

The working of the random forest is as follows:

Step-1: Firstly, it will select random K data points from the training set.

Step-2: After selecting k data points then building the decision trees associated with the selected data points (Subsets).

Step-3: Then choosing the number N for decision trees that you want to build.

Step-4: Repeating step 1 and 2.

Step-5: Finding the predictions of each decision tree, and assigning the new data points to the category that wins the majority votes.

#### **4.3.2. XGBoost Algorithm**

The working of XGBoost algorithm are as follows:

Step 1: Firstly, creating a single leaf tree.

Step 2: Then for the first tree, we have to compute the average of target variable as prediction and then calculating the residuals using the desired loss function

Step 3: Calculating the similarity score using formula: where, Hessian is equal to number of residuals;  $\text{Gradient}^2 = \text{squared sum of residuals}$ ;  $\lambda$  is a regularization hyperparameter.

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

Step 5: Applying similarity score we calculate Information gain.

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: where  $\rho$  is the learning rate.

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

### **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

## **Chapter 5**

### **Conclusions**

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, XG Boost, and K Nearest Neighbor (KNN), Support Vector Machine (SVM) to achieve maximum accuracy.



## **Chapter 6**

### **References**

- [1] Priyanka Sonar, Prof. K. Jaya Malini,” DIABETES PREDICTION USING DIFFERENT MACHINE LEARNING APPROACHES”, 2019 IEEE ,3rd International Conference on Computing Methodologies and Communication (ICCMC)
- [2] Archana Singh, Rakesh Kumar, “Heart Disease Prediction Using Machine Learning Algorithms”, 2020 IEEE, International Conference on Electrical and Electronics Engineering (ICE3)
- [3] A. Sivasangari, Baddigam Jaya Krishna Reddy, Annamareddy Kiran, P. Ajitha,” Diagnosis of Liver Disease using Machine Learning Models” 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)

## **Acknowledgements**

I am profoundly grateful to Prof. Mohammed Juned for his expert guidance and continuous encouragement throughout to see that this project rights its target.

I would like to express deepest appreciation towards Dr. Varsha Shah, Principal RCOE, Mumbai and Prof. Shiburaj Pappu HOD Computer Department whose invaluable guidance supported me in this project.

At last, I must express my sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

DANISH KHAN  
MUZAFFAR KHAN  
SOHAM MANJREKAR  
DANISH JAMADAR

# Publications

## Multiple Disease Prediction Webapp

**Soham Manjrekar<sup>1</sup>, Danish Khan<sup>2</sup>, Muzaffar Khan<sup>3</sup>, Danish Jamadar<sup>4</sup>,  
Mohammed Juned<sup>5</sup>**

Rizvi College of Engineering Department of Computer Engineering, Maharashtra, India

**Abstract** - Our point is to anticipate the various sorts of illness in a single stage by utilizing the inbuilt Python module Streamlit. In this task we are utilizing Naïve Bayes Algorithm, Random Forest, Decision Tree and SVM classifier are utilized for prediction of a particular disease. In this article we analyze Diabetes analysis, Heart disease and Parkinson's disease by using some of the basic parameters such as Pulse Rate, Cholesterol, Blood Pressure, Heart Rate, etc, and also the risk factors associated with the disease can be found using prediction model with good accuracy and Precision. The significance of this analysis is to analyze the maximum diseases to screen the patient's condition and caution the patients ahead of time to diminish mortality proportion. We have considered six diseases for now that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes and in the future, many more diseases can be added.

**Key Words:** Diabetes, Heart, Liver, KNN, Random Forest, XG Boost.

### 1. INTRODUCTION

In this digital world, data is an asset, and enormous data was generated in all the fields. Data in the healthcare industry consists of all the information related to patients. Here a general architecture has been proposed for predicting the disease in the healthcare industry. Many of the existing models are concentrating on one disease per analysis. Like one analysis for diabetes analysis, one for cancer analysis, one for skin diseases like that. There is no common system present that can analyze more than one disease at a time. Thus, we are concentrating on providing immediate and accurate disease predictions to the users about the symptoms they enter along with the disease predicted. So, we are proposing a system which used to predict multiple diseases by using streamlit. In this system, we are going to analyze Diabetes, Heart, and malaria disease analysis. Later many more diseases can be included 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. We are taking six diseases that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes. As all the six diseases are correlated to each other. To implement multiple disease analyses we are going to use machine learning algorithms and Streamlit. When the user is accessing this API, the user has to send the parameters of the disease along with the disease name. Our Model will invoke the corresponding model and return the status of the patient. Our basic idea is to develop a system which will predict and give the details of the disease predicted along with its severity which as symptoms are given as input by the user. The system

will compare the symptoms with the datasets provided in the database. If the symptom matches the datasets, then it should ask other relevant symptoms specifying the name of the symptom. If not, the symptom entered should be notified as the wrong symptom. After this a prompt will come up asking whether you want to still save the symptom in the database. If you click on yes, it will be saved in the database, if not it will go to the recycle bin. The main feature will be the machine learning, in which we will be using algorithms such as Naïve Bayes Algorithm, K-Nearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will predict accurate disease and also, will find which algorithm gives a faster and efficient result by comparatively-comparing. The importance of this system analysis is that while analyzing the diseases all the parameters which cause the disease are included so it is possible to detect the disease efficiently and more accurately. The final model's behavior will be saved as a python pickle file.

### 1.1 Description

A lot of analysis over existing systems in the healthcare industry considered only one disease at a time. For example, one system is used to analyze diabetes, another is used to analyze diabetes retinopathy, and another system is used to predict heart disease. Maximum systems focus on a particular disease. When an organization wants to analyze their patient's, health reports then they have to deploy many models. The approach in the existing system is useful to analyze only particular diseases. In multiple disease prediction systems, a user can analyze more than one disease on a single website. The user doesn't need to traverse different places in order to predict whether he/she has a particular disease or not. Main objective behind developing a system helps the doctors to cross verify their diagnosed results which gives promising solutions over existing death rates. By using our proposed work try to invent a unique platform and most promising solution for early diagnosis of multiple diseases. Existing work analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease wrong. So, we are giving more accurate solutions by using machine learning and Convolutional neural networks to detect diseases and make predictions.

### 1.2 Problem System

Many of the existing machine learning models for health care analysis are concentrating on one disease per analysis. For example, first is for liver analysis, one for cancer analysis, one for lung 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.

### 1.3 Proposed System

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. We are taking six diseases that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes. As all the six diseases are correlated to each other. To implement multiple disease

## **2. LITERATURE REVIEW**

1. According to the paper, diabetes is one of the dangerous diseases in the world, it can cause many varieties of disorders which includes blindness etc. In this paper they have used machine learning techniques to find out diabetes disease as it is easy and flexible to forecast whether the patient has illness or not. Their aim of this analysis was to invent a system that can help the patient to detect the diabetes disease of the patient with accurate results. Here they used mainly 4 main algorithms Decision Tree, Naïve Bayes, and SVM algorithms and compared their accuracy which is 85%, 77%, 77.3% respectively. They also used ANN algorithm after the training process to see the reactions of the network which states whether the disease is classified properly or not. Here they compared the precision recall and F1 score support and accuracy of all the models[1].

2. The main aim of the paper is, as the heart plays an important role in living organisms. So, the diagnosis and prediction of heart related disease should be perfect and correct because it is very crucial which can cause death cases related to heart.

So, Machine learning and Artificial Intelligence supports in predicting any kind of natural events. So, in this paper they calculate accuracy of machine learning for predicting heart disease using k-nearest neighbor, decision tree, linear regression and SVM by using UCI repository dataset for training and testing. They also compared the algorithm and their accuracy SVM 83 %, Decision tree 79%, Linear regression 78%, k-nearest neighbor 87% [2].

3. The system defines that liver diseases are causing a high number of deaths in India and is also considered as a life-threatening disease in the world. As it is difficult to detect liver disease at an early stage. So using automated programs using machine learning algorithms we can detect liver disease accurately. They used and compared SVM, Decision Tree and Random Forest algorithms and measured precision, accuracy and recall metrics for quantitative measurement. The accuracy is 95%, 87%, 92% respectively [3].

## **3. SYSTEM ANALYSIS**

### **3.1 Functional Requirement**

- The system allows the patient to predict the disease.
- The user adds the input for the particular disease and based on the trained model of the user input the output will be displayed.

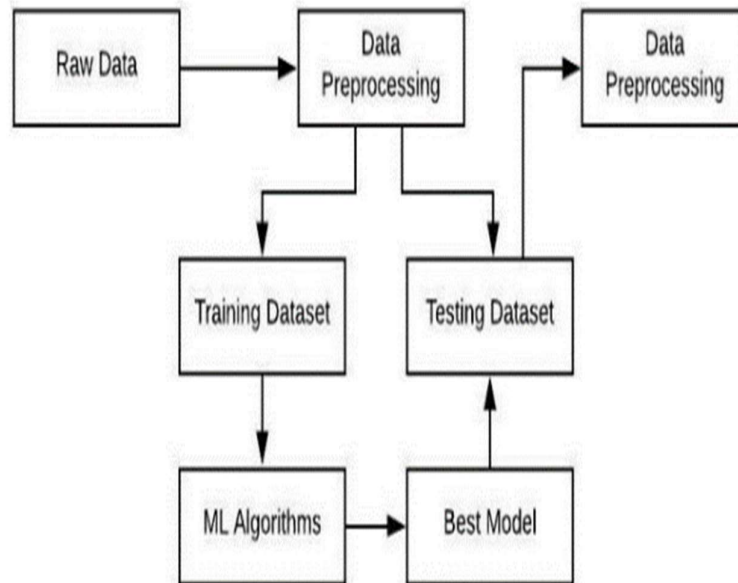
#### **3.1 Non-Functional Requirement**

- The website will provide a range of the values during the prediction of the disease.

- The website should be reliable and consistent.

## 4. DESIGN

### 4.1 Architecture Design



**Figure No4.1: Block Diagram**

In figure no 4.1 we have experimented on six diseases that are jaundice disease, hepatitis, Heart, Liver, Parkinson's disease and Diabetes as these are correlated to each other. The first step is to the dataset for heart disease, diabetes disease and liver disease we have imported the UCI dataset, PIMA dataset and Indian liver dataset respectively. Once we have imported the dataset then visualization of each inputted data takes place. After visualization pre-processing of data takes place where we check for outliers, missing values and also scale the dataset then on the updated dataset we split the data into training and testing. Next is on the training dataset we had applied KNN, XG Boost and random forest algorithm and applied knowledge on the classified algorithm using testing dataset. After applying knowledge, we will choose the algorithm with the best accuracy for each of the disease. Then we built a pickle file for all the diseases and then integrated the pickle file with the streamlit for the output of the model on the webpage.

## 5. IMPLEMENTATION

### 5.1 Algorithm

#### 5.1.1 KNN Algorithm

The working of the K-NN algorithm is as followed:

- Step-1: Start to select the K value for example k=5.
- Step-2: Then we will find the Euclidean distance between the points. It is calculated by the as:  

$$\text{Euclidean Distance} = \sqrt{[(X_2 - X_1)^2 + (Y_2 - Y_1)^2]}$$
- Step-3: Then we will calculate the Euclidean distance of the nearest neighbor.
- Step-4: Then count the number of the data points in each category .For example, find three values for Category A and two values for category B.
- Step-5: Then assign the new point to the category having the maximum number of neighbors. For example, Category A has the highest number of neighbors so we will assign the new data point to category A.
- Step-6: So finally, our KNN model is ready.

### 5.1.2. Random Forest Algorithm

Random Forest working is possible in two phases, first is to create the random forest by merging N decision trees, and second is making predictions for each tree created in the first phase.

The working of the random forest is as follows:

**Step-1:** Firstly, it will select random K data points from the training set.

**Step-2:** After selecting k data points then building the decision trees associated with the selected data points (Subsets).

**Step-3:** Then choose the number N for decision trees that you want to build.

**Step-4:** Repeating steps 1 and 2 .

**Step-5:** Finding the predictions of each decision tree, and assigning the new data points to the category that wins the majority votes.

### 5.1.3. XG Boost Algorithm

The working of XG Boost algorithm are as follows:

Step 1: Firstly, creating a single leaf tree.

Step 2: Then for the first tree, we have to 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:

$$\text{Similarity Score} = \text{Gradient} (\text{Gradient}^2 / \text{Hessian} + \lambda)$$

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 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:

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

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:

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

where  $\rho$  is the learning rate.

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

## 6. RESULT

In the system diabetes disease prediction model used KNN algorithm, heart disease uses the XG Boost algorithm and liver uses the random forest algorithm as these gave the best accuracy accordingly. There when the patient adds the parameter according to the disease it will show whether the patient has a disease or not according to the disease selected. The parameters will show the range of the values needed and if the value is not between the range or is not valid or is empty it will show the warning sign that adds a correct value.



## 1. User Interface:

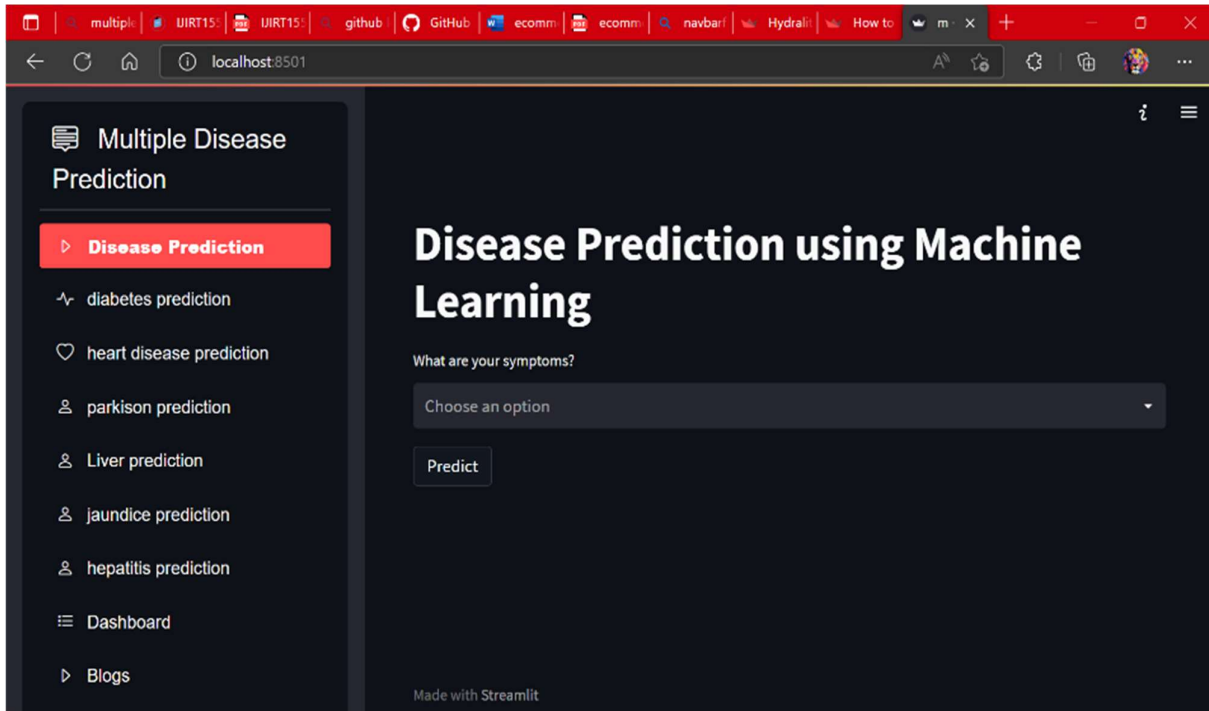


Figure No 6.1: User Interface

## 2. Diabetes Disease:

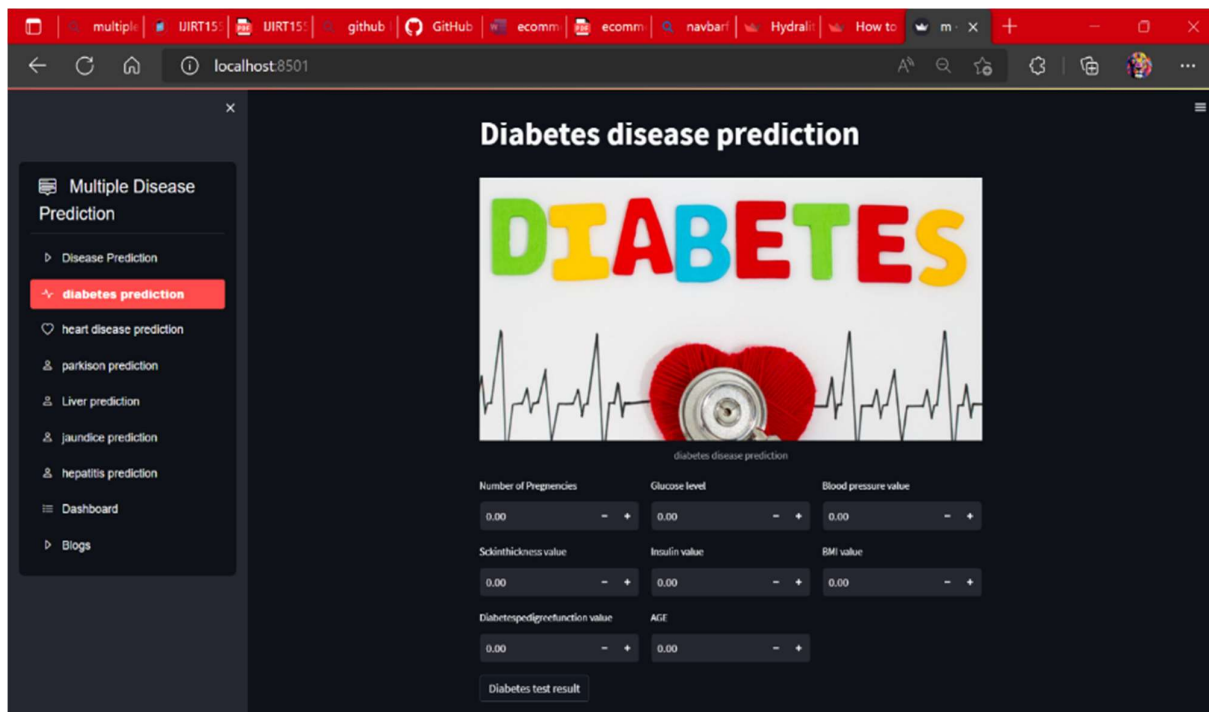


Figure No 6.2: Diabetes Disease Input Data

### 3. Heart Disease Prediction

The screenshot shows a web browser at localhost:8501 displaying the 'Heart disease prediction' application. The interface has a dark theme. On the left is a sidebar menu with options: 'Multiple Disease Prediction', 'Disease Prediction', 'diabetes prediction', 'heart disease prediction' (highlighted in red), 'parkinson prediction', 'Liver prediction', 'jaundice prediction', 'hepatitis prediction', 'Dashboard', and 'Blog'. The main content area is titled 'Heart disease prediction' and features a 3D anatomical illustration of a human torso with the heart highlighted in red. Below the illustration is a form titled 'heart failure' with input fields for various medical parameters, each with a value of 0.00 and a range selector (indicated by minus and plus signs). The parameters are: Age (18.00), sex (0.00), cp value (0.00), trestps value (0.00), chsl value (0.00), fbs value (0.00), resting value (0.00), thalach value (0.00), exang value (0.00), oldpeak value (0.00), slope value (0.00), ca value (0.00), and Thal value (0.00). At the bottom of the form is a button labeled 'Heart test result'.

Figure No 6.3: Heart Disease Prediction

### 4. Parkinson's Prediction

The screenshot shows the same web browser at localhost:8501, but now displaying the 'Parkinson prediction' application. The sidebar menu is identical, but 'parkinson prediction' is now highlighted in red. The main content area is titled 'Parkinson prediction' and features a 3D anatomical illustration of a human head in profile, showing the brain in yellow. Below the illustration is a form titled 'parkinson disease' with a grid of input fields for various parameters, each with a value of 0.00 and a range selector. The parameters are arranged in three columns: Column 1 includes MDP\_Fu1(s), MDP\_Uter1(s), MDP\_Glennar, ShimmerAPQ5, MMR, and upreal1; Column 2 includes MDP\_Fu1(s), MDP\_Uter1(s), MDP\_PFPQ, MDP1Shimmer(s), MDP1APQ, and MMR; Column 3 includes MDP\_Fu1(s), MDP1SRP, JitterDOP, ShimmerAPQ3, ShimmerAPQ3, and DFA. At the bottom of the grid is a button labeled 'DFA'.

Figure No 6.4: Parkinson's Prediction

## 5. Liver Disease Prediction

The screenshot shows a web browser at localhost:8501 displaying the 'Liver disease prediction' application. On the left is a sidebar menu with options: Disease Prediction, diabetes prediction, heart disease prediction, parkinson prediction, **Liver prediction** (highlighted), jaundice prediction, hepatitis prediction, Dashboard, and Blogs. The main content area features a title 'Liver disease prediction' above an illustration of a human torso with a highlighted liver. Below the illustration is a form titled 'Liver disease prediction' with the following input fields: 'Enter your Sex (male - 1 and female - 2)', 'Enter your age', 'Enter your Total\_Bilirubin', 'Enter your Direct\_Bilirubin', 'Enter your Alkaline\_Phosphatase', 'Enter your Alanine\_Aminotransferase', 'Enter your Aspartate\_Aminotransferase', 'Enter your Total\_Protiens', and 'Enter your Albumin\_and\_Globulin\_Ratio'. Each field has a numeric input with a value of 0.00 and a '+' button. At the bottom of the form is a 'Liver test result' button.

Figure No 6.5: Liver Disease Prediction

## 6. Jaundice Disease Prediction

The screenshot shows a web browser at localhost:8501 displaying the 'jaundice disease prediction' application. The sidebar menu is identical to the previous application, with 'jaundice prediction' highlighted. The main content area features a title 'jaundice disease prediction' above a diagram titled 'Causes of Jaundice'. The diagram shows a central red circle labeled 'Causes of Jaundice' surrounded by seven green circles: 'Cancer (pancreas or gall bladder carcinoma)', 'Hepatitis (infection in the liver)', 'Alcoholic liver disease', 'Liver cirrhosis', 'Gall stones in the gall bladder', 'Pancreatitis (inflammation in pancreas)', and 'Congenital disease (disease since birth)'. Below the diagram is a form titled 'jaundice test result' with the following input fields: 'Enter your age', 'Enter your Sex (male - 1 and female - 2)', 'Enter your Total\_Bilirubin', 'Enter your Direct\_Bilirubin', 'Enter your Alkaline\_Phosphatase', 'Enter your Alanine\_Aminotransferase', 'Enter your Aspartate\_Aminotransferase', 'Enter your Total\_Protiens', and 'Enter your Albumin'. Each field has a numeric input with a value of 0.00 and a '+' button. At the bottom of the form is a 'jaundice test result' button.

Figure No 6.6: Jaundice Disease Prediction

## 7. Hepatitis Prediction

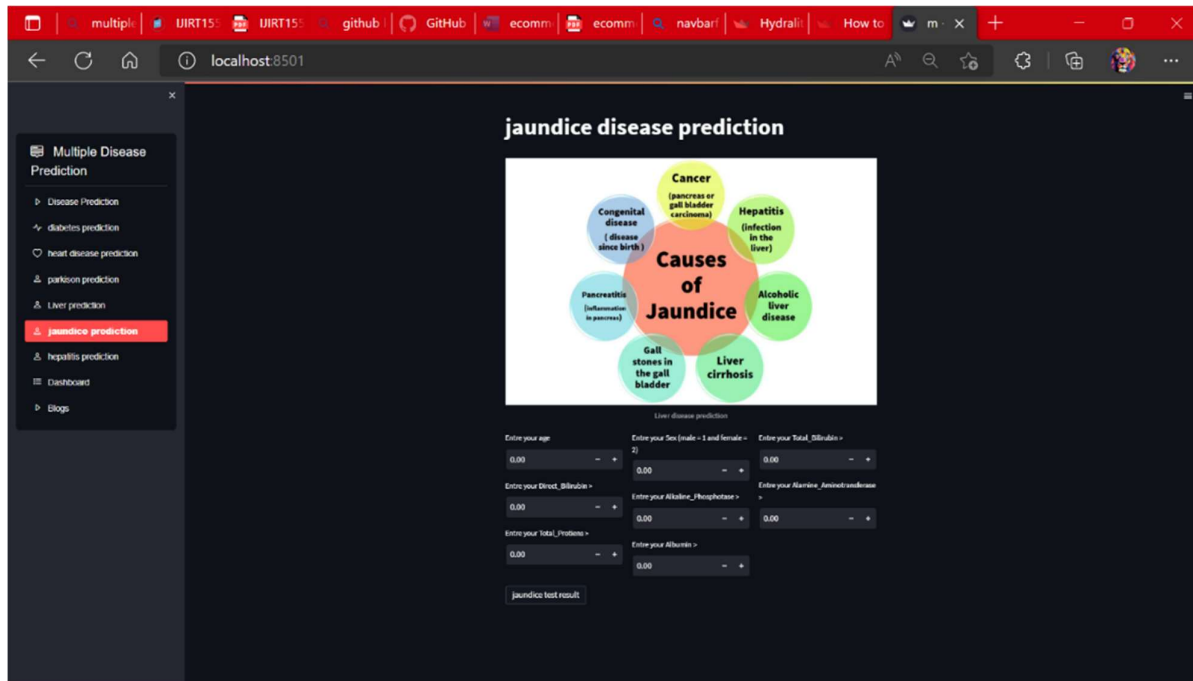


Figure No 6.7: Hepatitis Prediction

## 7. ACKNOWLEDGEMENT

We sincerely thank our college “**Rizvi College of Engineering**” for giving us a platform to prepare a project on the topic "Multiple Disease Prediction Webapp" and would like to thank our principal **Varsha Shah** for giving us the opportunities and time to conduct and research on the subject. We are sincerely grateful for **Prof. Mohammed Juned** as our guide, for providing help during our research, which would have seemed difficult without their motivation, constant support, and valuable suggestions.

## 8. REFERENCES

- [1] Priyanka Sonar, Prof. K. Jaya Malini,” DIABETES PREDICTION USING DIFFERENT MACHINE LEARNING APPROACHES”, 2019 IEEE ,3rd International Conference on Computing Methodologies and Communication (ICCMC)
- [2] Archana Singh, Rakesh Kumar, “Heart Disease Prediction Using Machine Learning Algorithms”, 2020 IEEE, International Conference on Electrical and Electronics Engineering (ICE3)
- [3] A. Sivasangari, Baddigam Jaya Krishna Reddy, Annamareddy Kiran, P. Ajitha,” Diagnosis of Liver Disease using Machine Learning Models” 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)