Project Report

On

Doctor Specific Heart Disease and Patient Specific Different Disease Prediction Framework using Machine Learning

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Amit Yadav
2018BITE079

Amit Kumar
2018BITE055

Under the supervision of

Dr. Prabal Verma

Department of Information Technology

National Institute of Technology Srinagar

June 2022

CERTIFICATE

This is to certify that the project titled **Doctor Specific Heart Disease and Patient Specific Different Disease Prediction Framework using Machine Learning** has been completed by **Amit Kumar (2018BITE055)** and **Amit Yadav (2018BITE079)** under my supervision in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology**. It is also certified that the project has not been submitted or produced for the award of any other degree.

Dr. Prabal Verma
Supervisor
Dept. of Information Technology
NIT, Srinagar

STUDENTS' DECLARATION

We, hereby declare that the work, which is being presented in the project entitled **Doctor Specific Heart Disease and Patient Specific Different Disease Prediction Framework using Machine Learning** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology**, in the session 2022, is an authentic record of our own work carried out under the supervision of **Dr. Prabal Verma**, Department of Information Technology, National Institute of Technology, Srinagar. The matter embodied in this project has not been submitted by us for the award of any other degree.

Dated: _____

(i) Name: Amit Yadav
Signature: _____

(ii) Name: Amit Kumar
Signature: _____

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Dr Prabal Verma**, Department of Information Technology for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr Shabir Sofi, Head of the Department, Information Technology**, for providing us with the required facilities for the completion of the project work. We are very much thankful to the Principal and Management, National Institute of Technology, Srinagar, for their encouragement and cooperation to carry out this work. We express our gratitude to Project Coordinator **Dr. Shabir Sofi**, for his continuous support and encouragement. We thank all teaching faculty of Department of Information Technology, whose suggestions during reviews helped us in accomplishment of our project. We are also thankful to our parents, friends, and classmates for their encouragement throughout our project period.At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

Amit Yadav (2018BITE079)

Amit Kumar (2018BITE055)

Abstract

The development of contemporary technologies such as data science and machine learning has paved the way for healthcare organisations and medical facilities to identify ailments as early as feasible and improve patient care. Lack of comprehensive medical data reduces the accuracy of detecting potential diseases. Additionally, the regional nature of some diseases may make disease prediction difficult. When something goes wrong inside of us, our bodies will exhibit symptoms. Sometimes these symptoms will indicate a little issue, but other times they may indicate a serious illness. If we do not address these symptoms at an early stage, it may be too late to treat the illness. It saves the time needed to finish the patient's full diagnosis because we can only diagnose the patient for the diseases that need to be diagnosed based on the system's recommendations. In this study, we employ different state of art classification techniques to attempt an accurate disease prediction.The method has amazing potential for better accurately predicting potential ailments. This study's primary goal is to assist nontechnical people and new clinicians in forming accurate opinions on diseases.

Keywords: SVM, Naive Bayes, Decision Tree, Random Forest, GradientBoost, KNN, confusion matrix

Contents

List of Figures

Chapter 1

Heart Disease Prediction Model

1.1 Introduction

12 million people every year die from heart disease, according to the World Health Organization.. Heart desease is one of the primary reason of death. Global population mortelity and morbidity rates. Cardiovascular disease evaluation is one of the most crucial topics in this domain. knowledge analysis. In recent years, cardiovascular disease has been spreading rapidly over the globe. In an effort to pinpoint the most important heart disease risk factors and correctly forecast the overall danger, numerous research have been conducted. Even heart disease, which kills a person without showing any outward signs of illness. Early detection of heart disease in high-risk persons is essential for assisting them in determining whether to change their lifestyle, which lessens repercussions. Machine learning facilitates decision-making and prediction from the massive amounts of data produced by the healthcare industry. This study aims to forecast future cases of heart disease by analysing patient data that employs a machine-learning system to categorise whether a patient has heart disease or not. In this case, machine learning techniques can be of great assistance.

1.1.1 Problem Definition

The healthcare sector has grown significantly in size. The healthcare sector generates enormous amounts of health-care data every day, from which it is possible to extract knowledge for diagnosing diseases that can affect a patient in the future utilising treatment information and This cover

Later, information from the healthcare data will be used to help patients make effective decisions. Additionally, this part has to be improved using the useful healthcare data. The biggest problem is figuring out how to get information out of these data since there is so much of it. Large enough to allow the application of some machine learning techniques. The anticipated The goal of this project is to see whether disease may be predicted and whether early diagnosis and treatment dispensed to patients which can lower the risk of death, save patients lives, and, to some extent, minimise the cost of disease treatment through early detection. For this issue, we have created a system that can predict disease based on their symptoms.

1.1.2 Objective

There is a need to study and make a system which will make it easy for an end users to predict heart disease. To detect the Heart Disease through the examining parameters of patient's using different state of the art classification techniques of Machine Learning Models. The Proposed system will give the prediction based on symptoms.

1.1.3 Need Of The System

There is always a need of a system that will provide the heart disease information according to parameters for early diagnosis shared by doctor.
There is always a need of a system that will provide the heart disease severity level of heart disease according to medical values.
This model will help the doctor and user to find heart disease severity level according to their medical values.
We always need such a system, if we are having high severity then send a notification to the doctor of our city.

1.2 Literature Survey

There is number of projects has been done related to disease prediction systems using various machine learning algorithms in medical field.

Senthil Kumar Mohan, et al. [3] suggested Efficient Heart Disease Prediction in his paper. By implementing machine learning and hybrid algorithms, this strategy's goal is to identify important elements, improving the accuracy of cardiovascular prediction. With many combinations of highlights and a few well-known arrangement techniques, the expectation model is created. With hybrid random forest with a linear model (HRFLM), which was similarly trained on a variety of data mining approaches and expectation techniques, including KNN, LR, SVM, NN, and Vote, which have become increasingly popular in recent years for differentiating and predicting heart disease, we were able to improve the accuracy level of our prediction framework for heart disease to 88.7 percent.

In recent years, several experiments and researches have been conducted in the area of medical science and machine learning, resulting in the publication of important publications. In the publication Kanishk et al. [1], it is suggested to use WEKA software along with KStar, J48, SMO, and Bayes Net to forecast heart disease. Using k-fold cross validation, SMO (89 percent accuracy) and Bayes Net (87 percent accuracy) produce the best results when compared to KStar, Multilayer Perceptron, and J48 methods. These algorithms' accuracy performance still falls short of expectations. Therefore, if accuracy performance is increased, it will be possible to diagnose diseases more accurately.

Researchers found that Gaussian Naive Bayes and Random Forest had the highest accuracy of 91.2 percent in a study utilising the Cleveland dataset for heart illnesses, which included 303 instances and employed 10-fold Cross Validation, examining 13 attributes, and applying 4 different algorithms.

Four models were used in the studies, which were conducted using a similar dataset from Framingham, Massachusetts, and were trained and tested to the highest possible accuracy. K Neighbors Classifier performed with an accuracy of 87 percent, Support Vector Classifier at 83 percent, Decision Tree Classifier at 79 percent, and Random Forest Classifier at 84 percent.

Utilizing decision tree and hill climbing algorithms, Purushottam et al. [4] proposed a ”Efficient Heart Disease Prediction System” in their study. They used the Cleveland dataset, and before using classification methods, data was preprocessed. Evolutionary Learning (KEEL), an open-source data mining programme that fills in the missing values in the data set, provides the basis for the Knowledge Extraction process. A decision tree operates in a top-down fashion. At each level, a node is chosen by a test for every actual node chosen by the hill-climbing algorithm. Confidence are the variables and their corresponding values. Its confidence level is at least 0.25. About 86.7 percent of the time, the system is accurate.

In their study ”Prediction of Heart Disease Using Machine Learning Algorithms,” Santhana Krishnan, J., et al. [3] advocated using decision trees and the Naive Bayes method to predict heart disease. The decision tree algorithm builds the tree based on specific circumstances that result in True or False choices. The outcomes of algorithms like SVM and KNN are based on split conditions that can be vertical or horizontal depending on the dependent variables. However, a decision tree is a structure that resembles a tree with a root node, leaves, and branches, and it is based on the decisions made in each tree. The value of the attributes in the dataset is also explained by the decision tree. Additionally, they used the Cleveland data set. Using various techniques, the dataset is divided into a training portion 70 percent and a testing portion 30 percent. This method provides accuracy of 91 percent.

Few data mining techniques are utilised in ”Heart Disease Prediction Using Effective Machine Learning Techniques,” a proposal by Avi Golande et al. [2] that helps doctors distinguish between different types of heart disease. Frequently used techniques are Naive Bayes, Decision Tree, and K-Nearest Neighbor. other original Packing calculation, Part thickness, and other characterization-based techniques are used. following minimal neural networks and simplification, direct kernel self-arranging a manual and SVM (support Vector Machine) .

1.3 Software Requirements

Technology: Python Django

IDE: Visual Studio Code

Client Side Technologies: HTML, CSS, JavaScript , Bootstrap

Server Side Technologies: Python

Data Base Server: Sqlite

OperatingSystem: Ubuntu/Windows

1.4 System Analysis

1.4.1 Purpose

Doctor can predict heart disease severity level for early diagnosis.

User can search for doctor's help at any point of time.

User can talk about their illness and get instant diagnosis.

Informs the user about the heart disease,severity level of heart disease.

Doctors get more clients online.

Our system will notify the doctors, about the patients in stage 3 of heart disease.

1.4.2 Proposed System

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. This system is implemented using the following modules.

1. Collection of Dataset

2. Selection of attributes
3. Data Pre-Processing
4. Balancing of Data
5. Disease Prediction

Collection Of Dataset

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, there are 76 attributes; out of which, 14 attributes are used for the system.

Selection Of Attributes

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum , cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

Data Pre-Processing

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

Balancing Of Data

Imbalanced datasets can be balanced in two ways. They are Under Sampling and Over Sampling (a) Under Sampling: In Under Sampling, dataset balance is done by the reduction of the size of the

sample class. This process is considered when the amount of data is adequate.

(b) Over Sampling: In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

Disease Prediction

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

1.4.3 System Overview

Login:- User(Patient or Doctor) Login to the system using his credential.

SignUp:- If User is a new user he will enter his personal information, user credential through which he can login to the system.

About:- Patient can view his personal information.

Feedback:- A patient can look up a doctor by name, address, or specialty.

History:- The admin will be informed of the patients feedback.

Heart Prediction:- Doctor and user will access the system using his User ID and Password.

1.4.4 ER Diagram

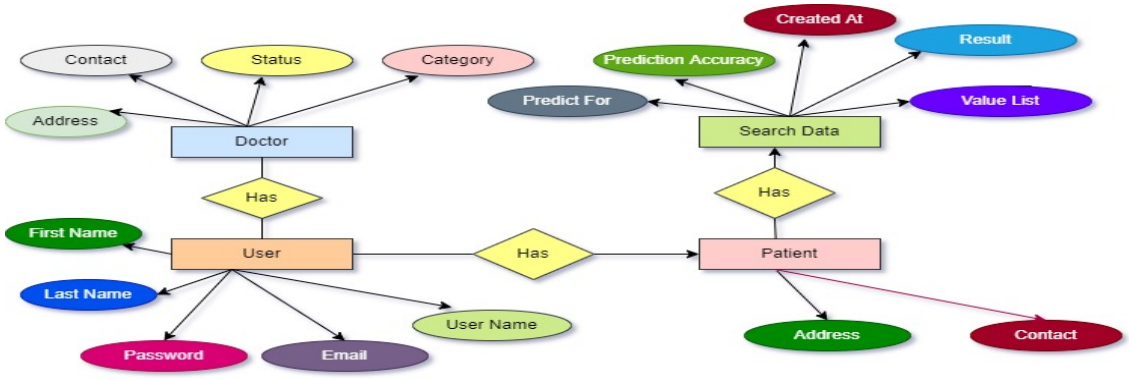


Figure 1.1: **ER Diagram**

1.4.5 Activity Diagram Of Heart Disease Prediction

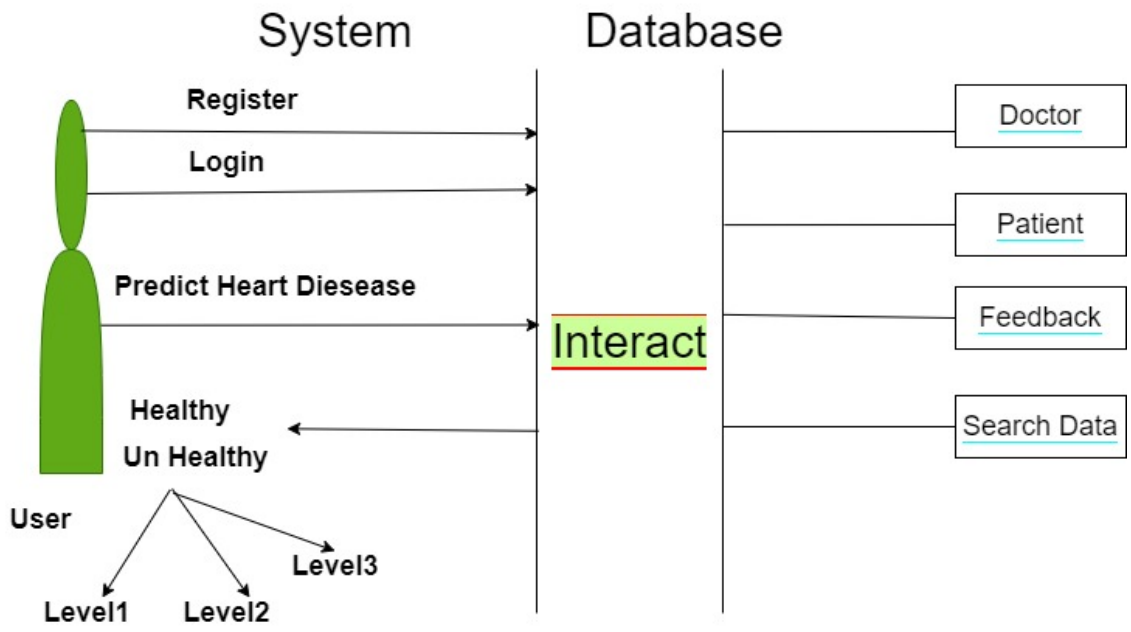


Figure 1.2: **Activity Diagram**

1.4.6 Connectivity Between User And Database

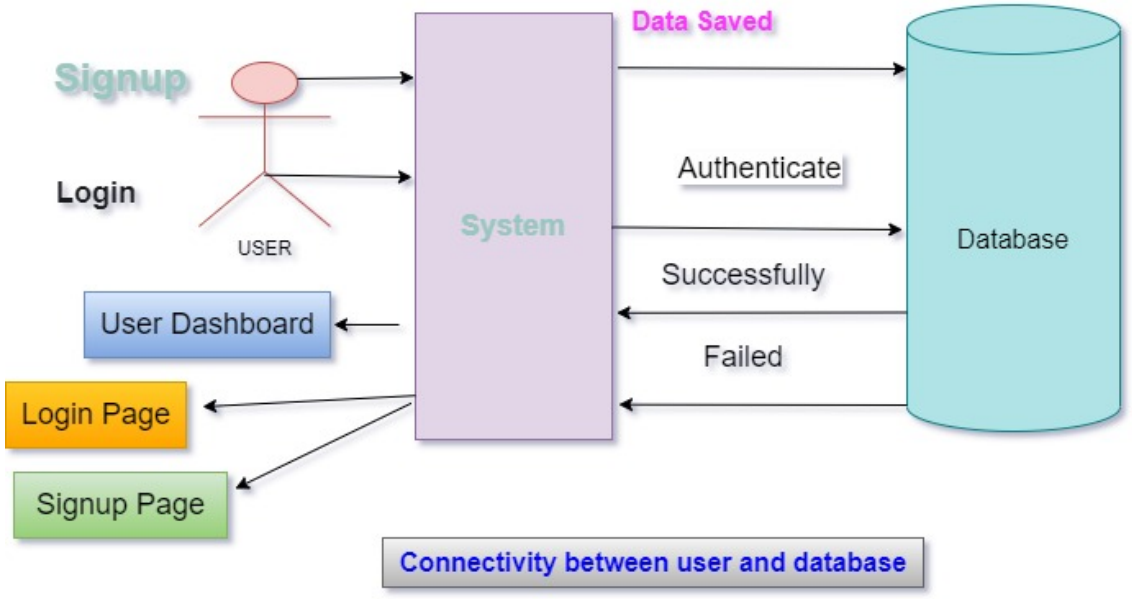


Figure 1.3: **Connectivity Between User And Database**

1.4.7 Flow Diagram For Heart Disease Prediction Model

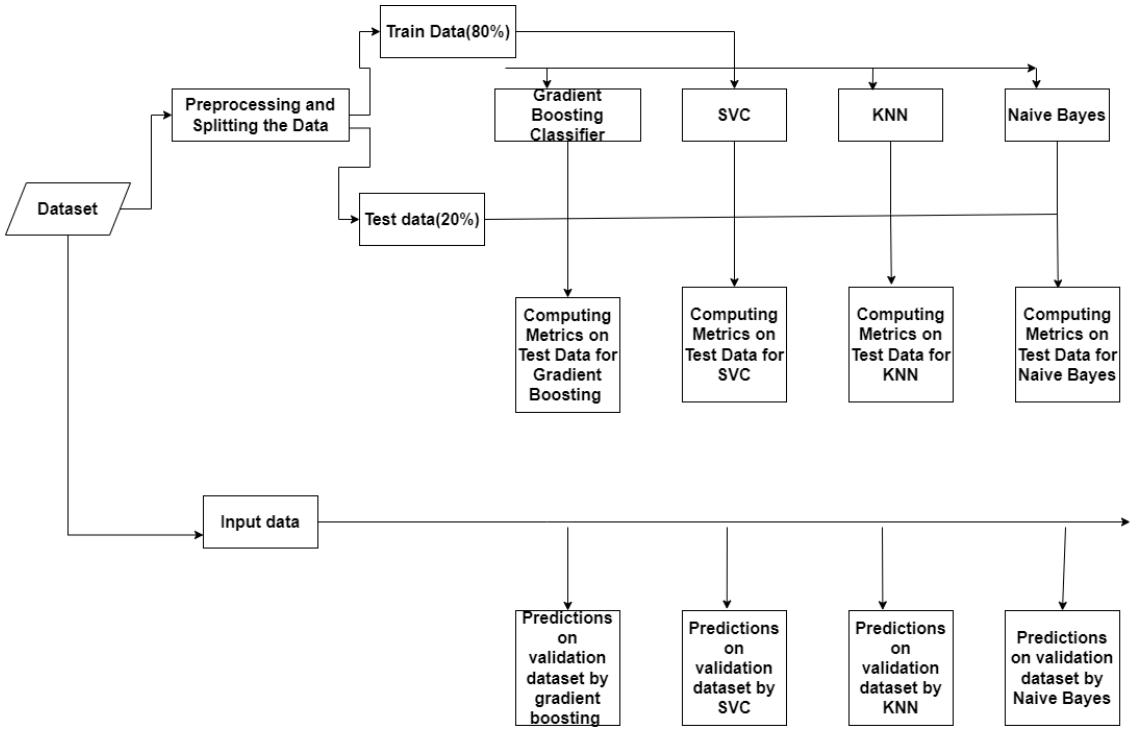


Figure 1.4: **Flow Diagram For Heart Disease Prediction Model**

1.5 Output Screens

1.5.1 Home page

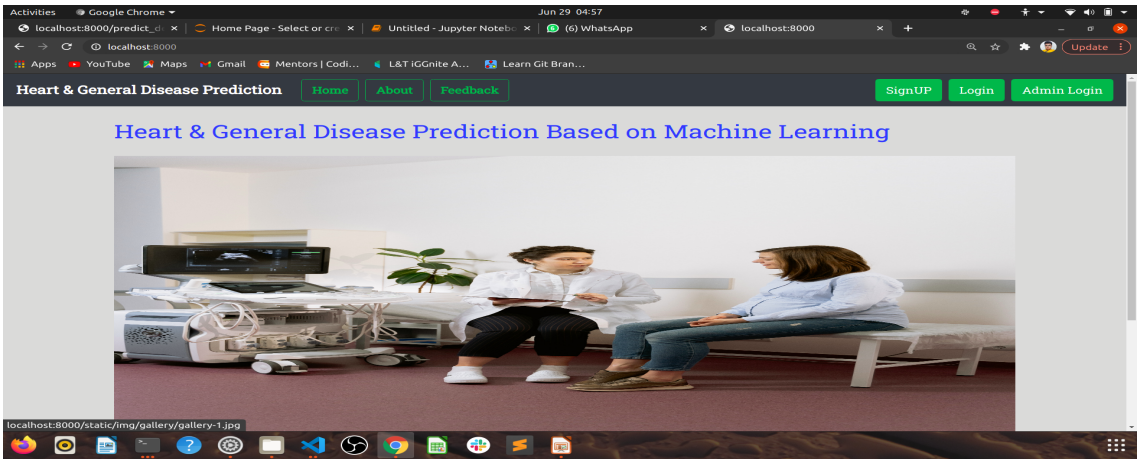


Figure 1.5: **Home Page**

1.5.2 User(patient) Login page

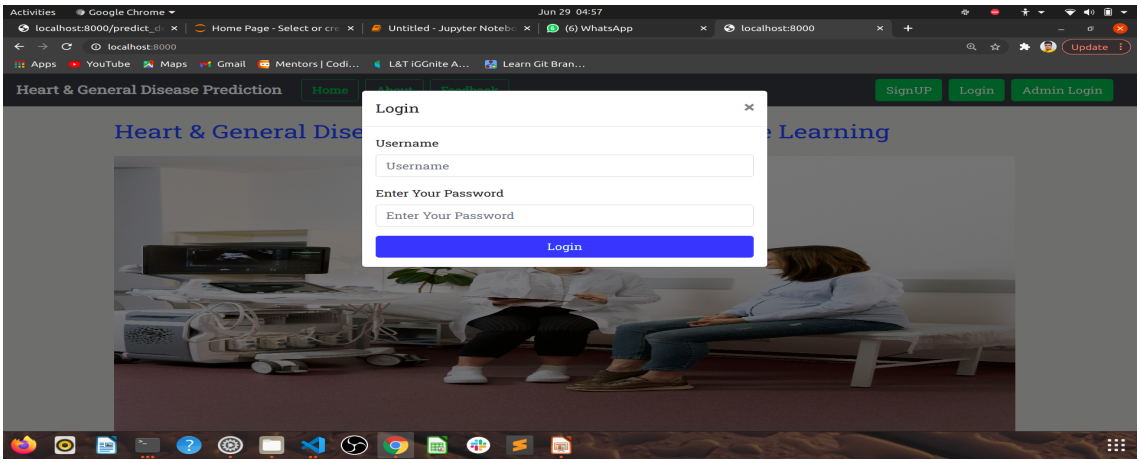


Figure 1.6: **User Login Page**

1.5.3 SignUp

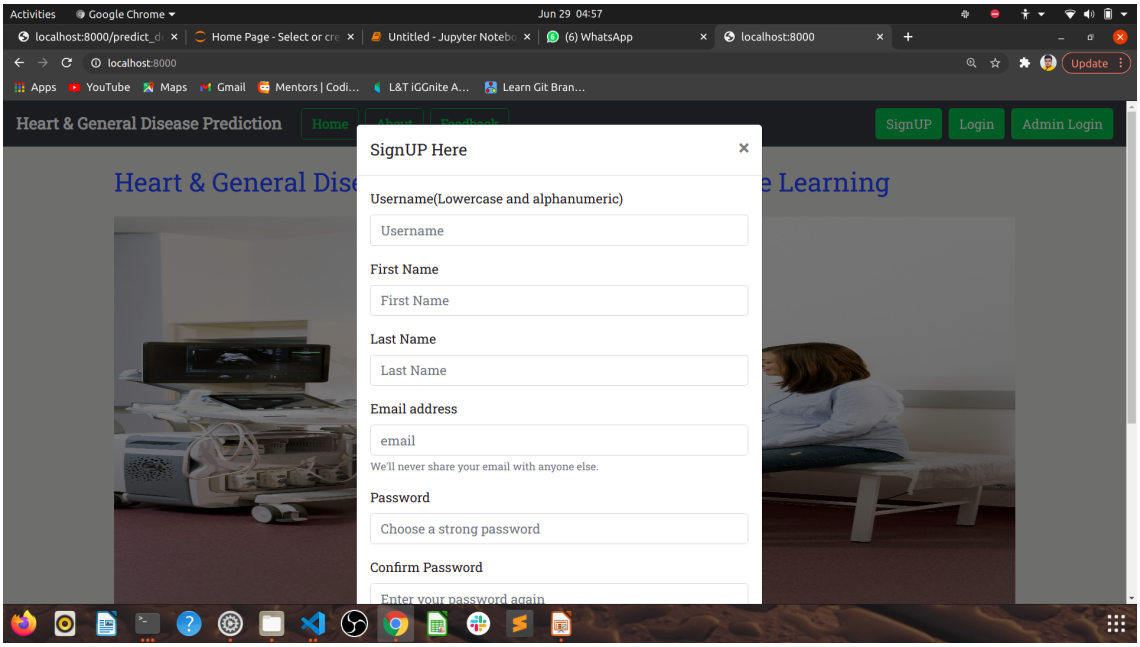


Figure 1.7: **signup Page**

1.5.4 Patient Home Page

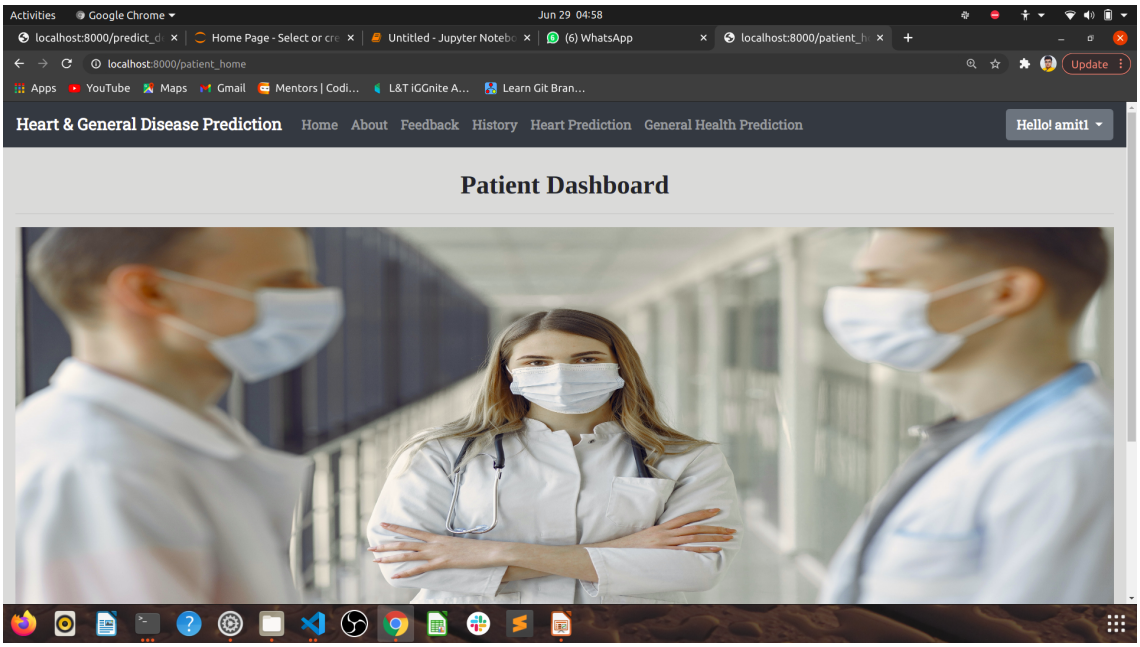


Figure 1.8: **Patient Home Page**

1.5.5 User Profile Page

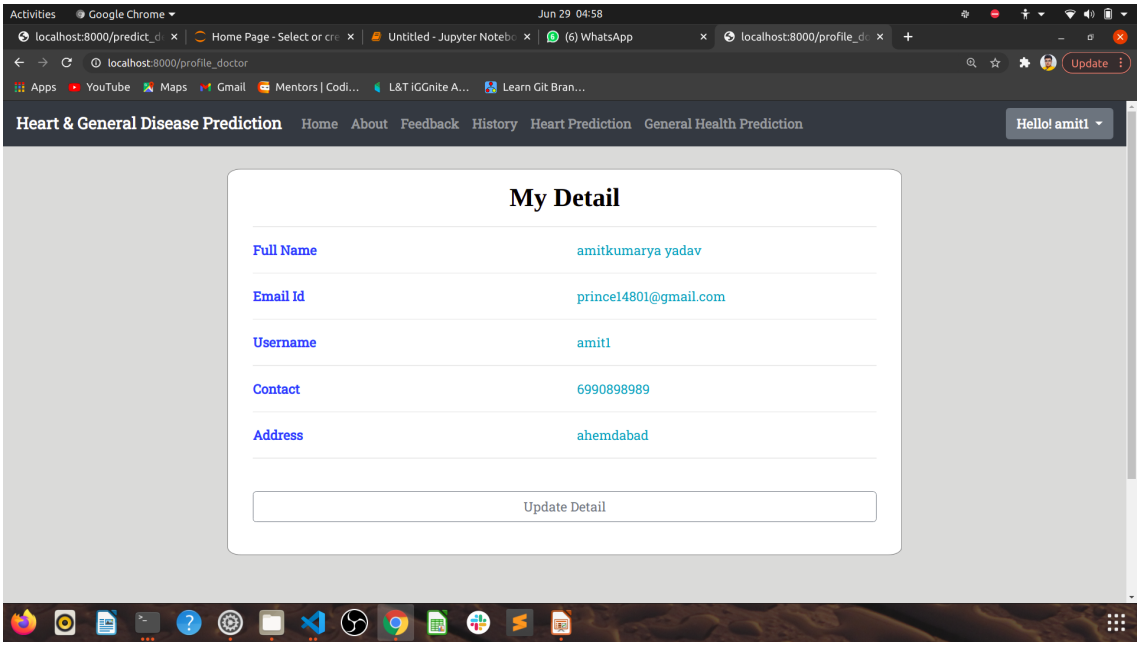


Figure 1.9: **User Profile Page**

1.5.6 Send Feedback page

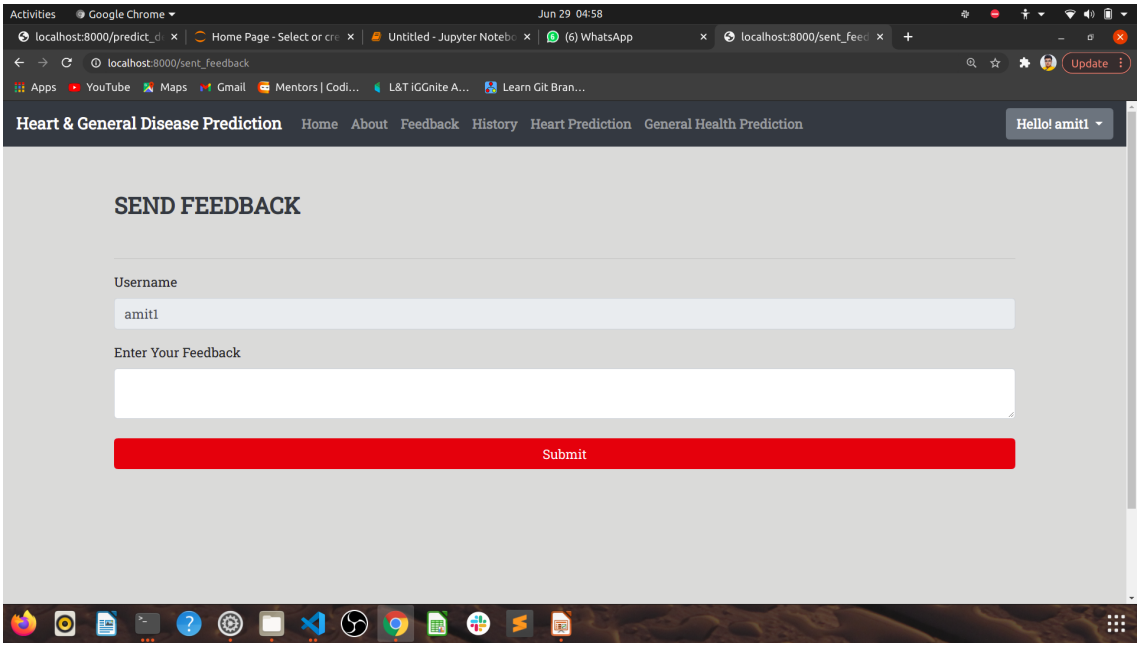


Figure 1.10: **Send Feedback Page**

1.5.7 View History page

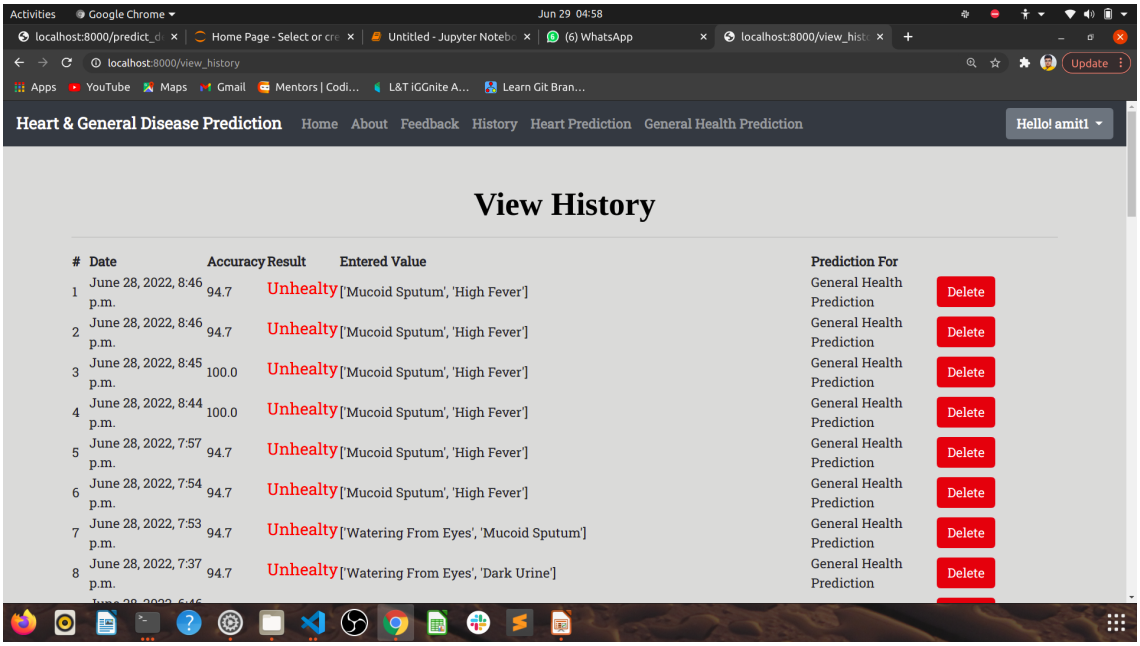


Figure 1.11: **View History Page**

1.5.8 Input User Heart Disease Parameters Page

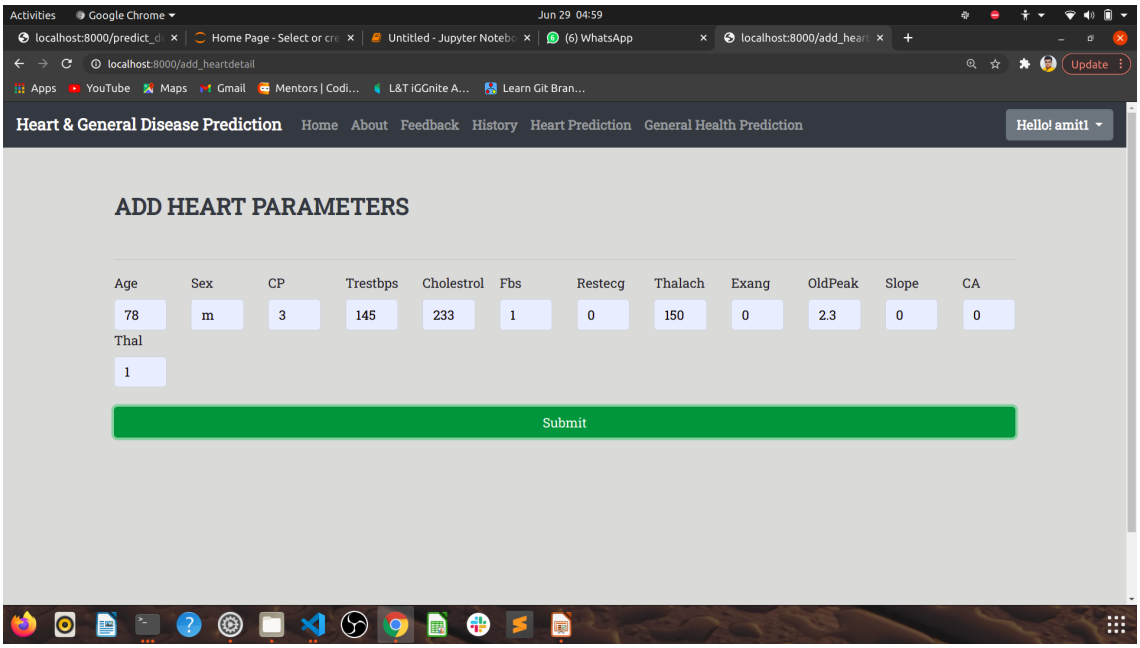


Figure 1.12: **Input User Heart Disease Parameter**

1.5.9 Result Page for Heart Disease Prediction

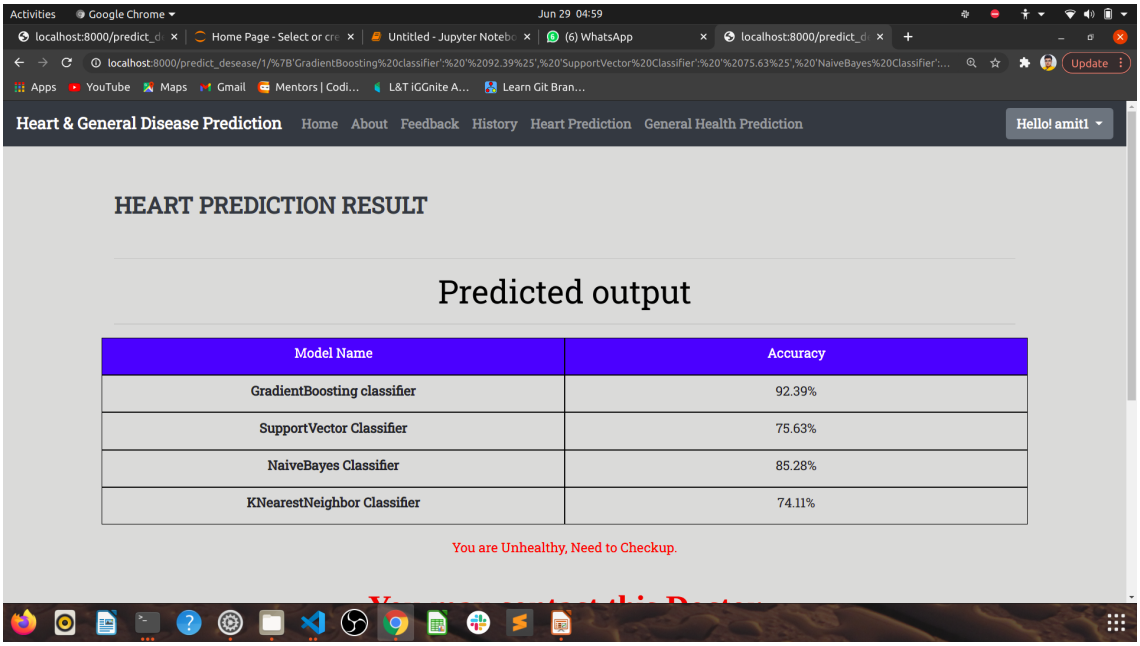


Figure 1.13: **Result Page For Heart Disease Prediction**

1.5.10 Admin Login Page

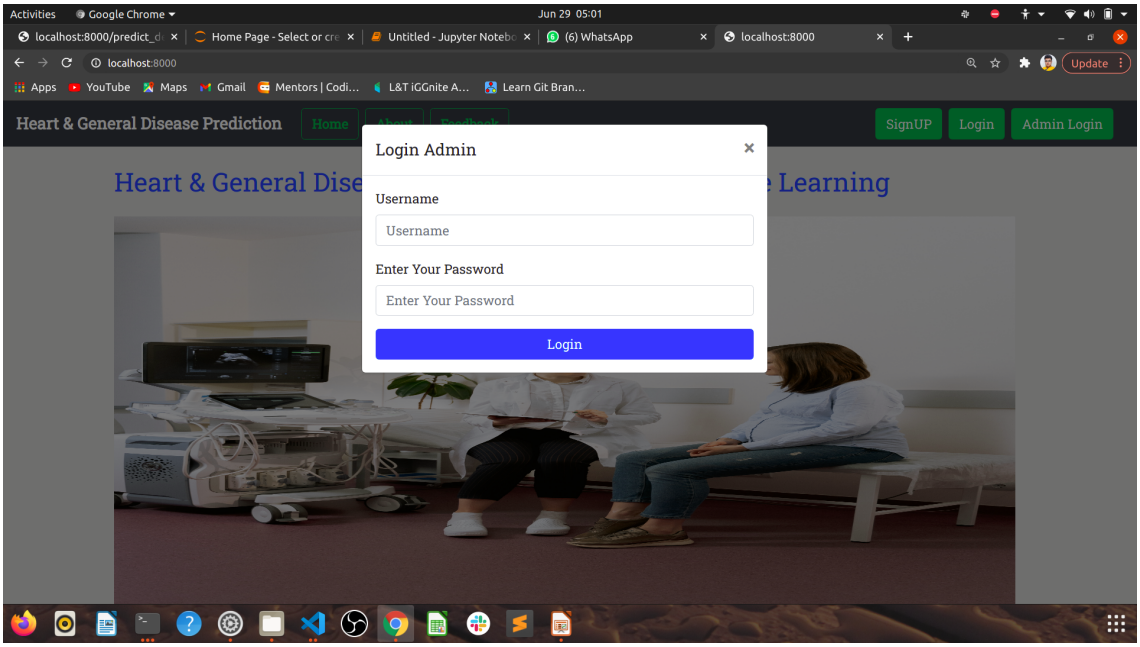


Figure 1.14: **Admin Login Page**

1.5.11 Admin Home Page

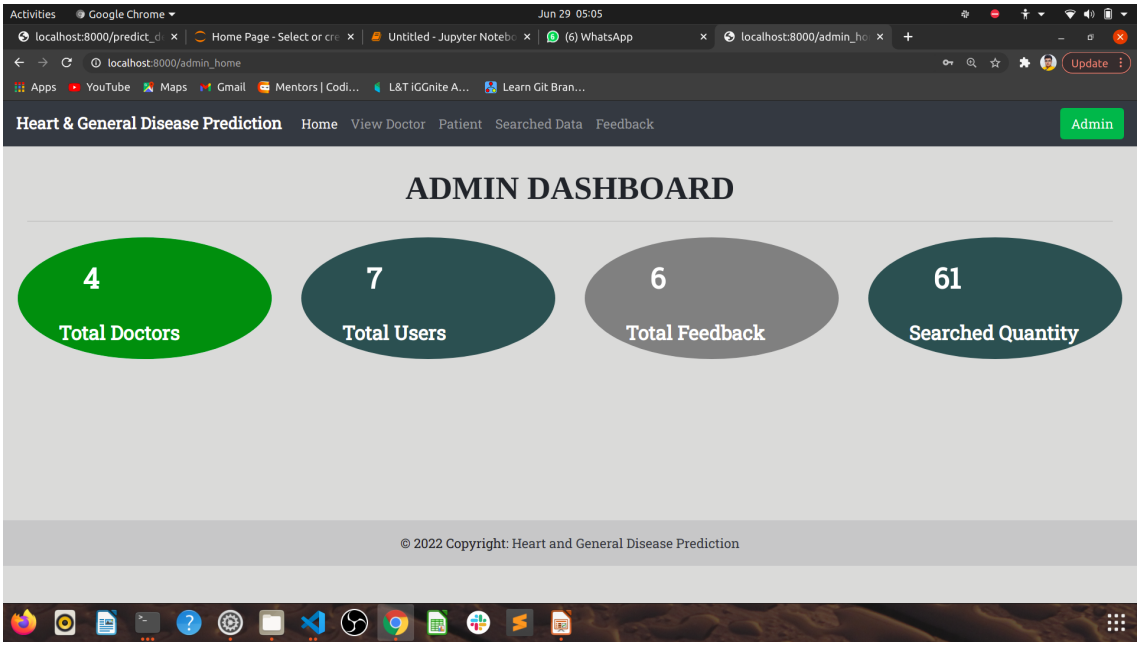


Figure 1.15: **Admin Home Page**

1.5.12 View Doctor Page

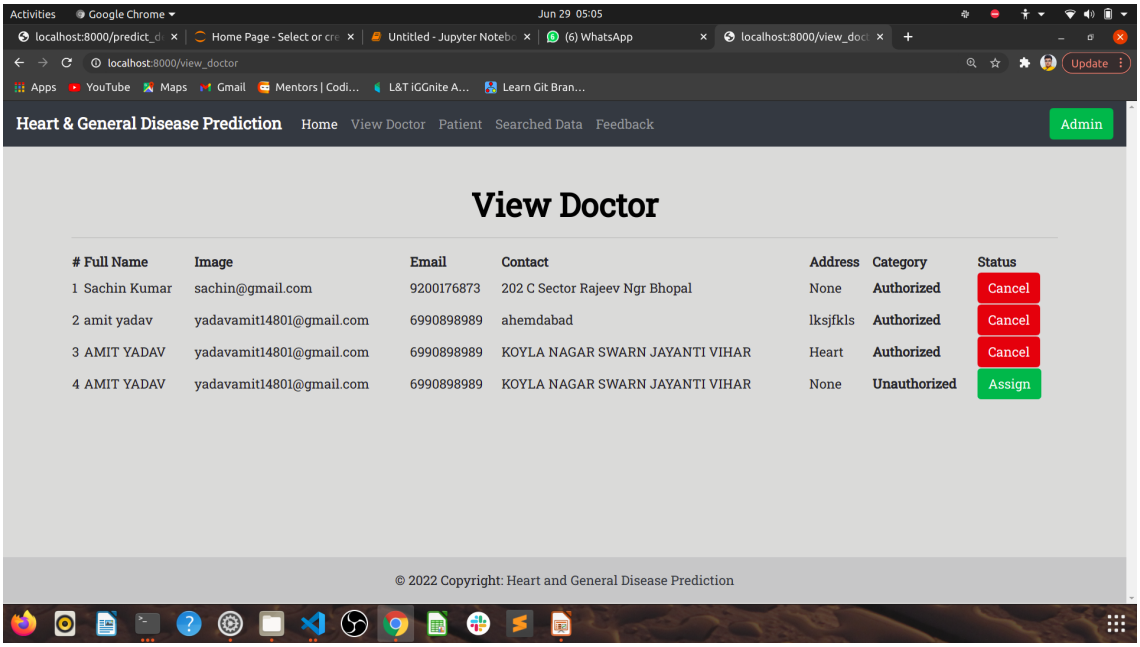


Figure 1.16: **View Doctor Page**

1.5.13 View All Patient Page

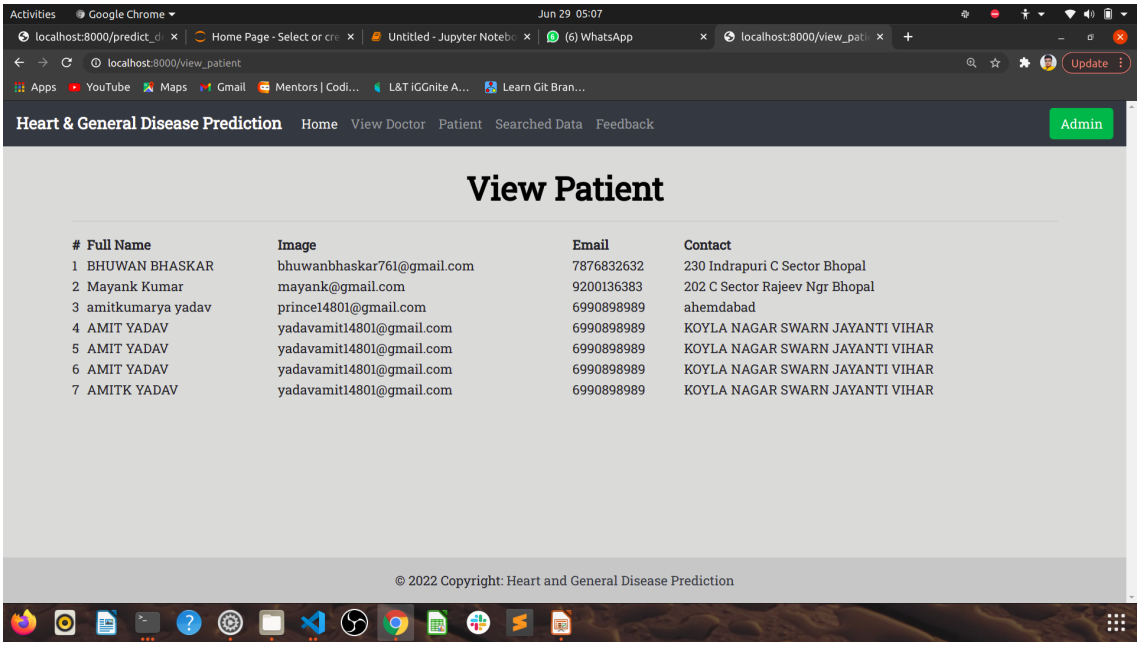


Figure 1.17: **View All Patient Page**

1.5.14 Patient History

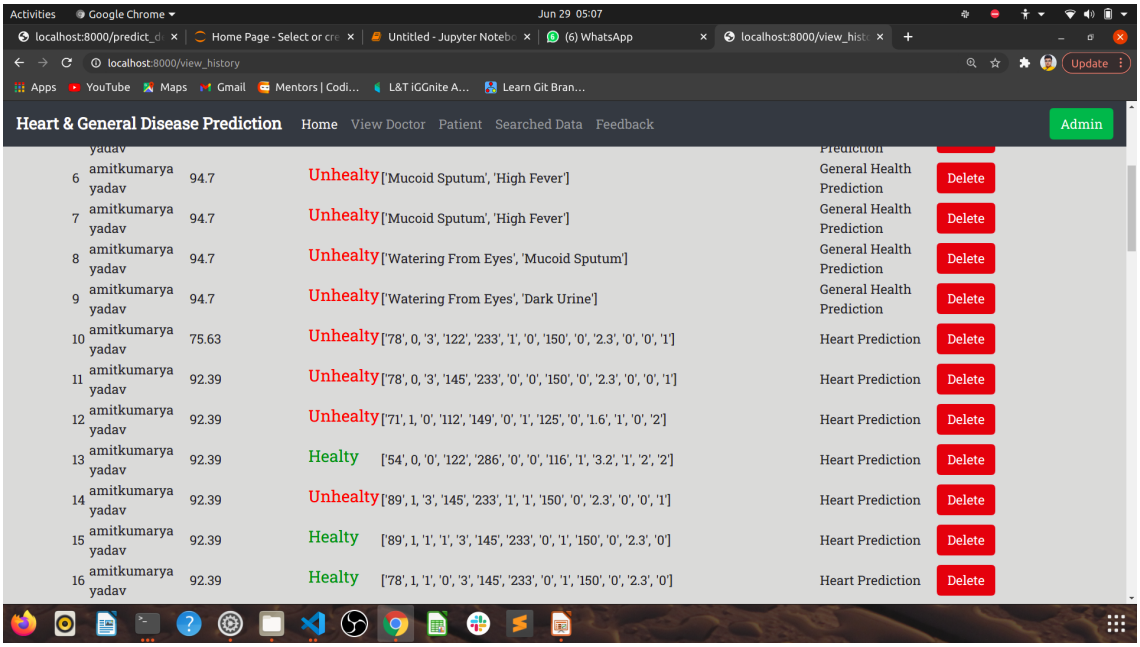


Figure 1.18: **Patient History**

1.5.15 View Patient Details Page

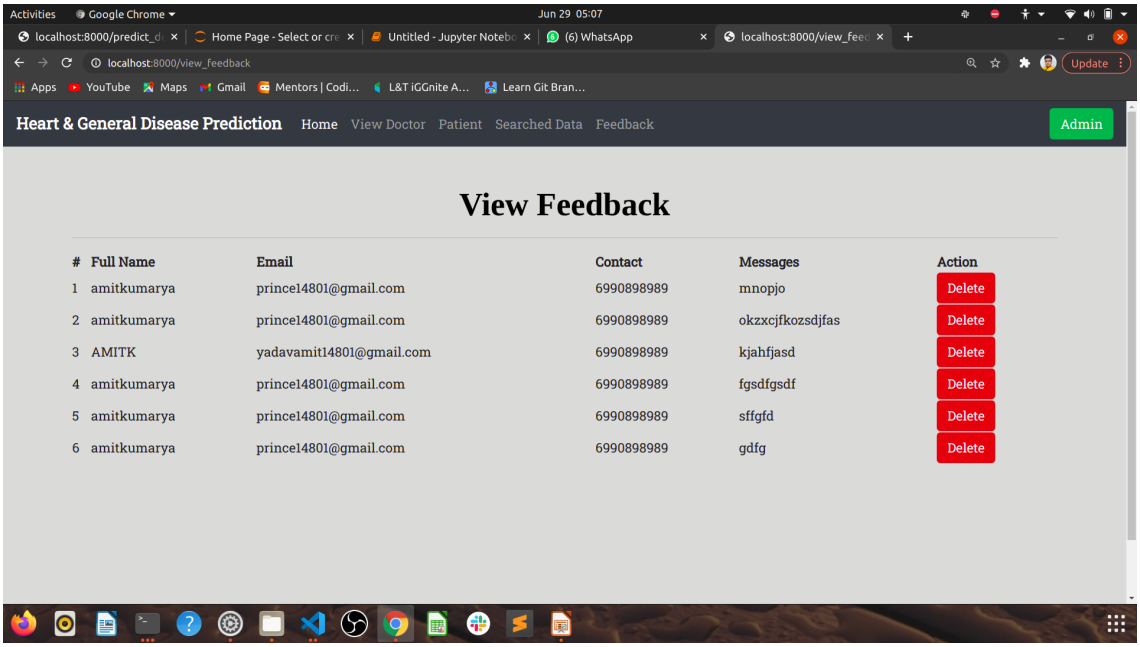


Figure 1.19: **View Patient Details Page**

1.6 Algorithm Used

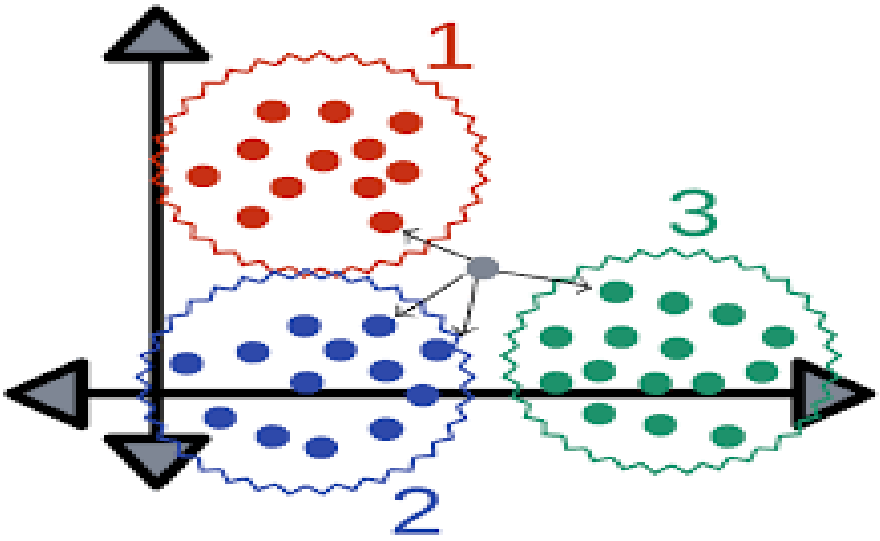
1.6.1 Ensemble Learning Approach

Ensemble learning models combines decisions from multiple models to improve their overall performance Basic ensemble learning techniques are: Max voting:- predictions we get from the majority are as the final prediction. Averaging:- in this final prediction is the average of prediction of all the models. weighted average:- according to the significance different models are assigned different weights and then predictions are made.

1.6.2 K Nearest Neighbour Classifier

K-Nearest Neighbor is one of the most basic supervised learning-based machine learning algorithms. The K-NN algorithm places the new instance in the category that resembles the current categories the most, presuming that the new case and the previous cases are comparable. After storing all the previous data, a new data point is categorised using the K-NN algorithm based on similarity. This indicates that new data can be reliably and quickly categorised using the K-NN approach. K-NN is a non- parametric technique, thus it makes no assumptions about the data39;s underlying structure, even though classification issues are where it is most frequently used.

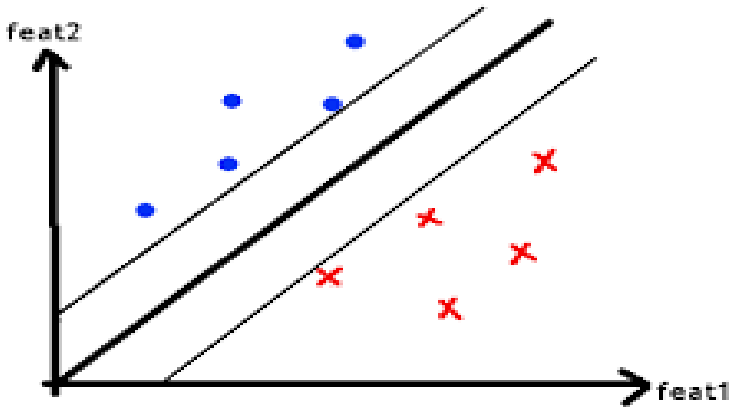
Example: Let's say we have a picture of a species that resembles both cats and dogs, but we aren't sure if it is one or the other. Therefore, since the KNN algorithm is based on a similarity metric, we can utilise it for this identification. Our KNN model will look for similarities between the new dataset's features and those in the photos of cats and dogs, and based on those similarities, it will classify the new data set as either cat or dog-related.



1.6.3 Support Vector Classifier

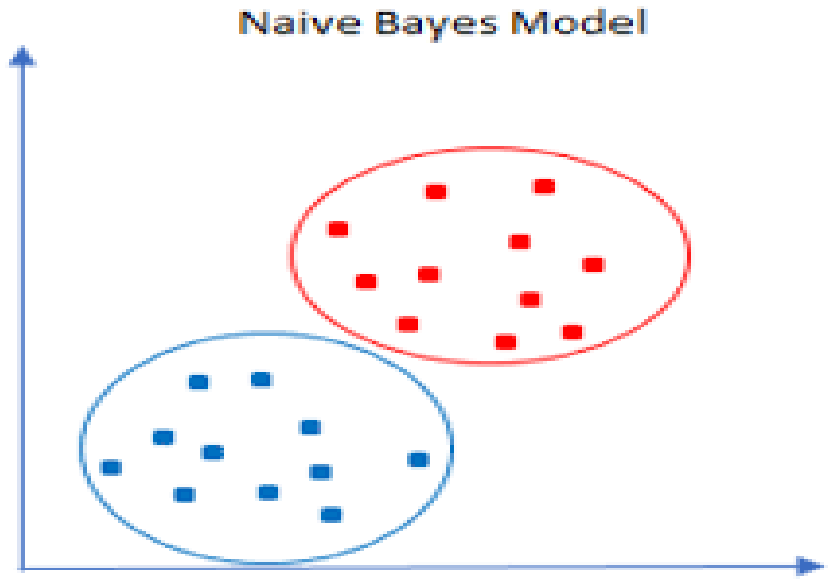
Support vector classifier(SVC) are robust yet adaptable supervised machine learning classifier used for outlier detection, regression, and classification. SVMs are frequently employed in clas- sification issues and are extremely effective in large dimensional spaces. Because they only use a portion of the training points in the decision function, SVMs are well- liked and memory-efficient algorithms. SVMs' primary objective is to classify the datasets into several groups in order to identify the maximum marginal hyperplane (MMH).

Example: The example we used for the KNN classifier can be utilised to understand SVM. If we want a model that can correctly distinguish between a cat and a dog, let's say we observe an unusual cat that also resembles a dog. We can build such a model by utilising the SVM algorithm. Prior to testing it with this weird animal, we will first train our model with several photographs of cats and dogs so that it can become familiar with the various attributes of cats and dogs. When a result, the extreme cases of cats and dogs will be seen by the support vector as it draws a judgement border between these two sets of data (cat and dog).



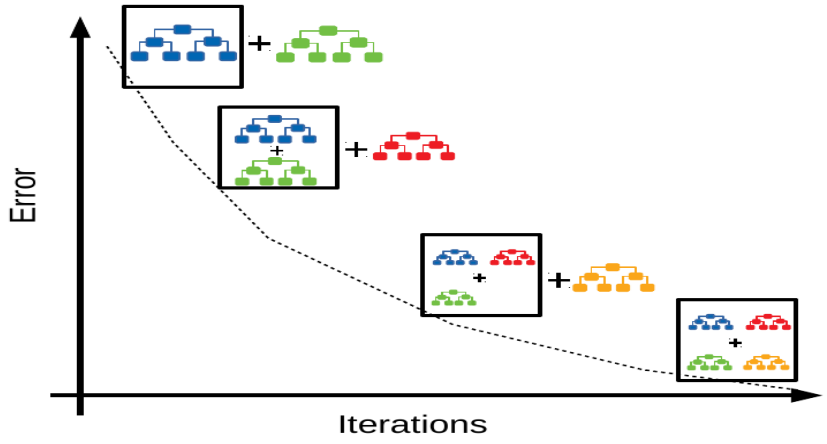
1.6.4 Naive Bayes Classifier

A group of classification algorithms built on the Bayes' Theorem are known as naive Bayes classifiers. It is a family of algorithms rather than a single method, and they are all based on the idea that every pair of features being classified is independent of the other. It is mostly employed in text categorization with a large training set. This state of art Classifier is one of the most straightforward and efficient classifier available today. It aids in the development of quick machine learning models capable of making accurate predictions. Being a probabilistic classifier, it makes predictions based on the likelihood that an object will occur. Spam filtration, Sentimental analysis, and article classification are a few examples of Naive Bayes algorithms that are frequently used.



1.6.5 Gradient Boosting Classifier

A class of machine learning techniques known as gradient boosting classifiers combines a number of weak learning models to produce a powerful predicting model. Gradient boosting frequently makes use of decision trees. Due to their success in categorising large datasets, gradient boosting models are gaining popularity and have lately been successful in numerous Kaggle data science challenges. The Scikit-Learn machine learning toolkit for Python offers a variety of XGBoost implementations of gradient boosting classifiers.



1.7 Coding

Dataset used in Heart Disease

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

1.7.1 Naive Bayes Implementation For Heart disease Prediction Model

Naive Bayes implementatioin

In [93]:
```
from sklearn.naive_bayes import GaussianNB
nb = train_model(X_train, Y_train, X_test, Y_test, GaussianNB)

nb.fit(X_train, Y_train)

y_pred_nb = nb.predict(X_test)
```

```
Train accuracy: 82.07%
Test accuracy: 85.37%
```

In [94]:
```
score_nb = round(accuracy_score(y_pred_nb,Y_test)*100,2)

print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")
```

```
The accuracy score achieved using Naive Bayes is: 85.37 %
```

In [95]:
```
#Gaussian Naive Bayes
from sklearn.naive_bayes import GaussianNB
model = train_model(X_train, Y_train, X_test, Y_test, GaussianNB)
```

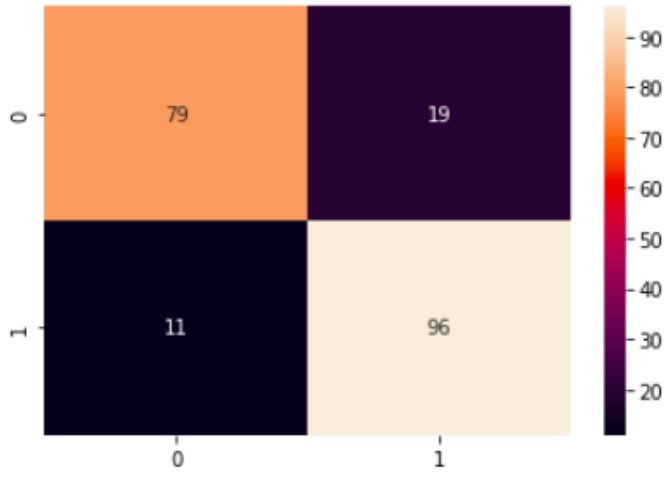
```
Train accuracy: 82.07%
Test accuracy: 85.37%
```

Confusion Matrix of Naive Bayes

In [96]:
```
from sklearn.metrics import confusion_matrix
```

In [97]:
```
matrix= confusion_matrix(Y_test, y_pred_nb)
```

In [98]:
```
sns.heatmap(matrix,annot = True, fmt = "d")
```



Precision Score

In [99]:
```
from sklearn.metrics import precision_score
```

In [100]:
```
precision = precision_score(Y_test, y_pred_nb)
```

In [101]:
```
print("Precision: ",precision)
```

```
Precision:  0.8347826086956521
```

Recall

In [102]:
```
from sklearn.metrics import recall_score
```

In [103]:
```
recall = recall_score(Y_test, y_pred_nb)
```

In [104]:
```
print("Recall is: ",recall)
```

```
Recall is:  0.897196261682243
```

F Score

In [105]:
```
print((2*precision*recall)/(precision+recall))
```

```
0.8648648648648648
```

1.8 Comparative Analysis

Author	Dataset	Classification Technique used	Best Technique found	Accuracy Achieved	Application Domain
Otoom et al.[1]	Cleveland (UCI) 303 cases, 76 attributes	Naïve Bayes, Support vector machine, and FT	SVM	88.3%	Heart Disease
Parthiban et al. [2]	Chennai Research Institute (500 patients data)	Naive Bayes, SVM	SVM	91.60%	Heart Disease
Chaurasia et al.[3]	UCI machine learning laboratory	Naive Bayes, J48 and Bagging	Bagging	85.03%	Heart Disease
Vembandasamy et al. [4]	Chennai Research Institute (500 patients data)	Naive Bayes	Naïve Bayes	86.419%	Heart Disease
X. Liu et al. [5]	UCI machine learning laboratory	Relief and Rough set (RFRS) method	cross-validation scheme 9	91.59%	Heart Disease
Our Model	Kaggle Dataset	Svc,Gradient Boosting,knn,Naïve Bayes	Gradient Boosting	92.3%	Heart Disease

1.9 Analysis For Heart Disease

```
In [8]: data.head()
```

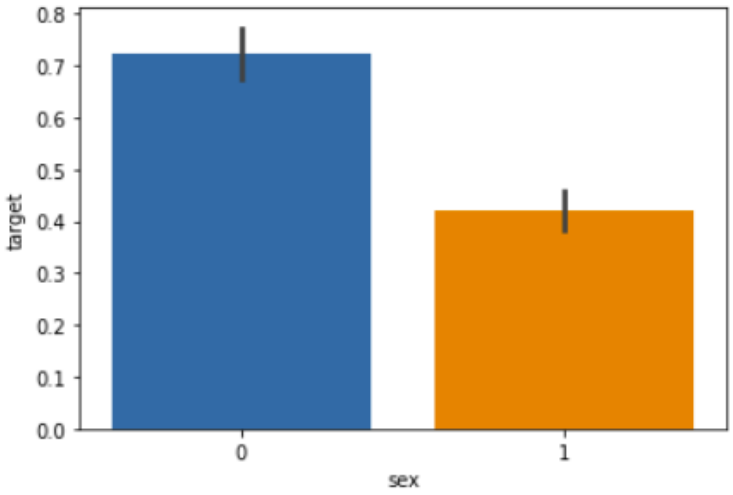
Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [14]:
```
data["sex"].unique()
```

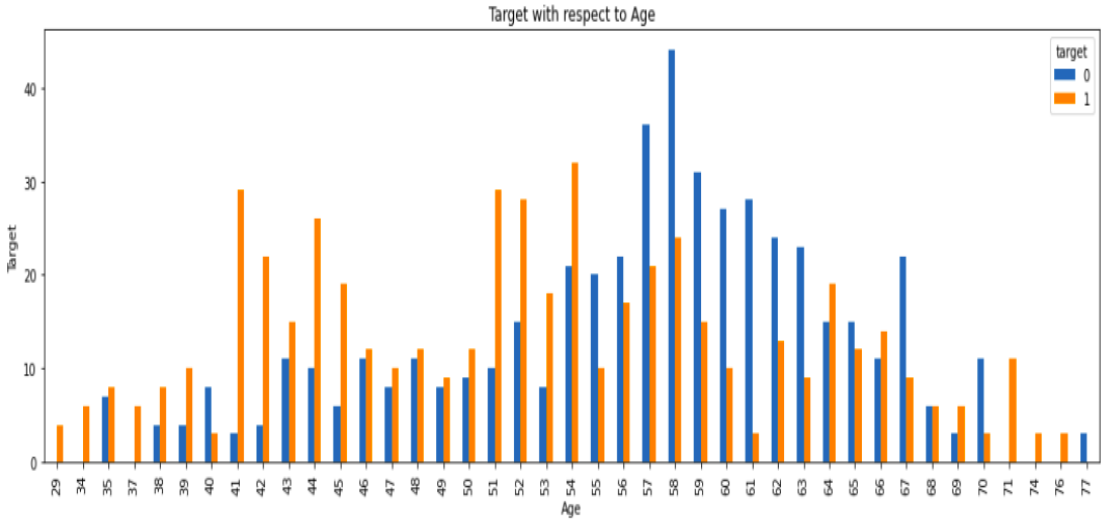
Out[14]:
```
array([1, 0])
```

In [15]:
```
sns.barplot(data["sex"],data["target"])
```



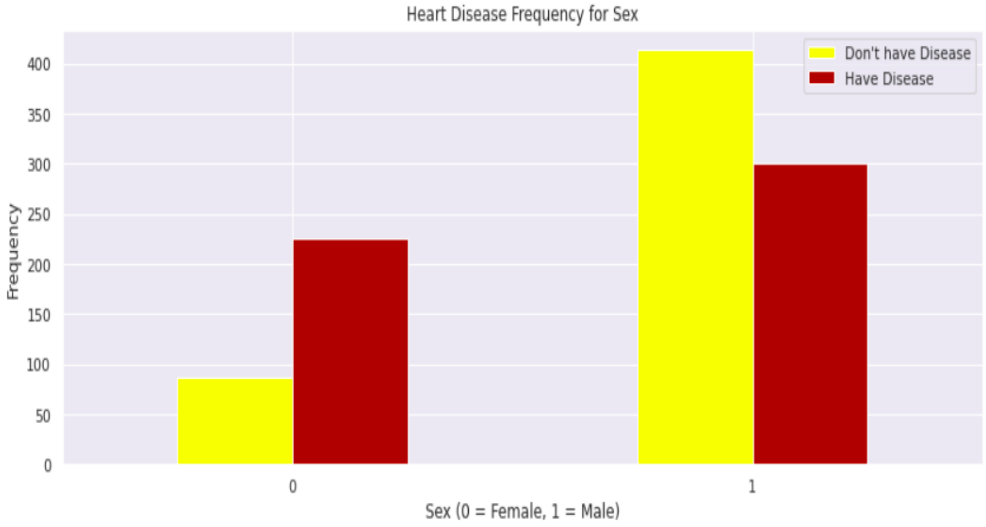
Heart Disease Frequency For Ages

In [20]:
```
pd.crosstab(data.age,data.target).plot(kind="bar",figsize=(18,5))
plt.title('Target with respect to Age')
plt.xlabel('Age')
plt.ylabel('Target')
# plt.savefig('heartDiseaseAndAges.png')
plt.show()
```



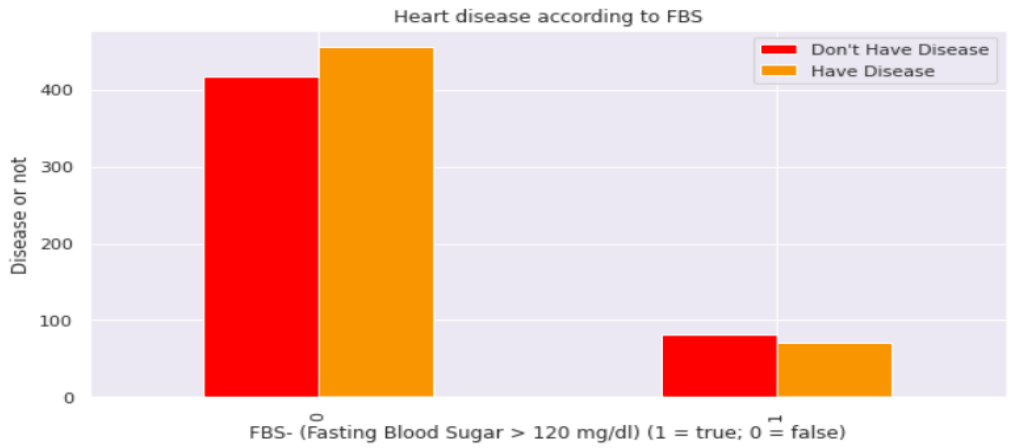
Heart Disease frequency for sex (where 0 is female and 1 is male and "red" is have heart disease and "yellow" is don't have heart disease)

In [148]:
```
pd.crosstab(data.sex,data.target).plot(kind="bar",figsize=(14,5),color=['yellow','#AA1111' ])
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Don't have Disease", "Have Disease"])
plt.ylabel('Frequency')
plt.show()
```



Heart Disease According To Fasting Blood Sugar

1]:
```
pd.crosstab(data.fasting_blood_sugar,data.target).plot(kind="bar",
                                       figsize=(10,5),color=['red','#f49242'])
plt.title("Heart disease according to FBS")
plt.xlabel('FBS- (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation=90)
plt.legend(["Don't Have Disease", "Have Disease"])
plt.ylabel('Disease or not')
plt.show()
```



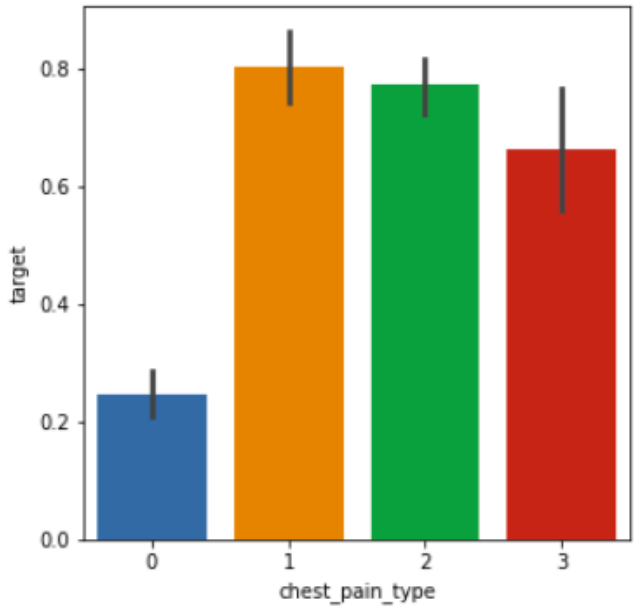
Analysing the chest pain (4 types of chest pain)

[Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic]

Graph between chest pain type and target value

```
In [26]: data["chest_pain_type"].unique()
Out[26]: array([0, 1, 2, 3])
```

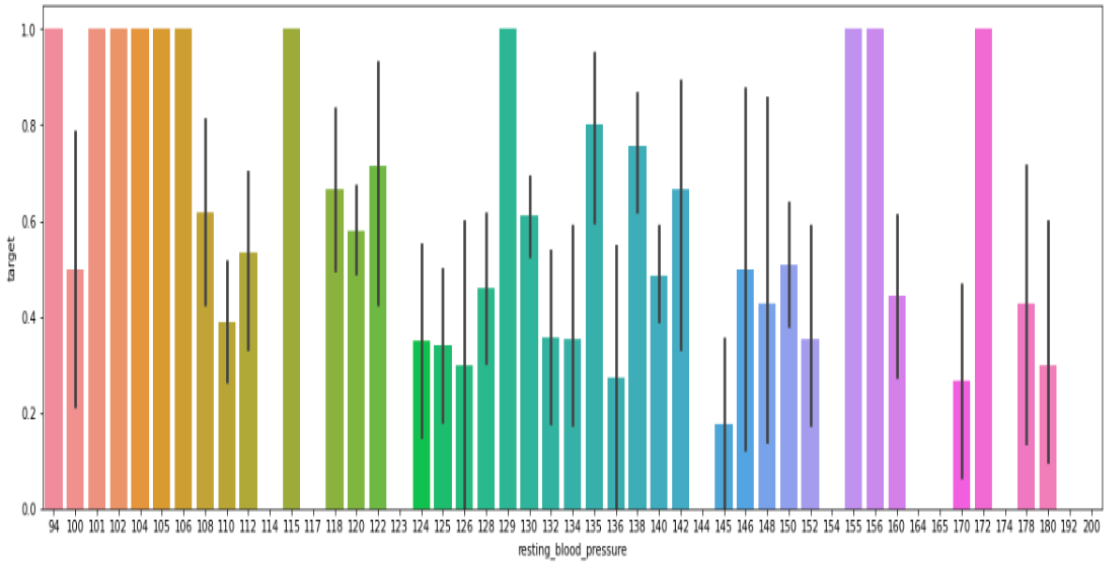
```
In [27]: plt.figure(figsize=(5, 5))
         sns.barplot(data["chest_pain_type"],y)
```



Analysing The person's resting blood pressure (mm Hg on admission to the hospital)

```
In [30]: data["resting_blood_pressure"].unique()
Out[30]: array([125, 140, 145, 148, 138, 100, 114, 160, 120, 122, 112, 132, 118,
                128, 124, 106, 104, 135, 130, 136, 180, 129, 150, 178, 146, 117,
                152, 154, 170, 134, 174, 144, 108, 123, 110, 142, 126, 192, 115,
                 94, 200, 165, 102, 105, 155, 172, 164, 156, 101])
```

```
In [32]: plt.figure(figsize=(20, 6))
         sns.barplot(data["resting_blood_pressure"],y)
```

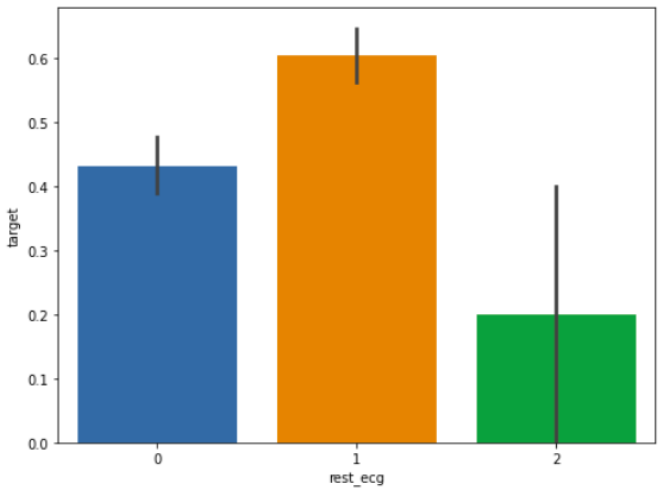


Analysing the Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)

In [33]:
```
data["rest_ecg"].unique()
```

Out[33]:
```
array([1, 0, 2])
```

In [34]:
```
plt.figure(figsize=(8, 6))
sns.barplot(data["rest_ecg"],y)
```



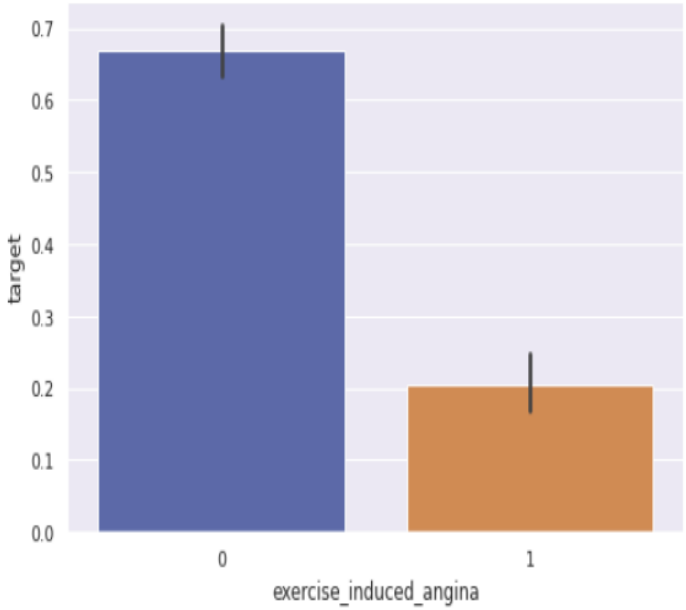
Analysing Exercise induced angina (1 = yes; 0 = no)

```
data["exercise_induced_angina"].unique()
```

```
array([0, 1])
```

```
plt.figure(figsize=(8, 5))
sns.barplot(data["exercise_induced_angina"],y)
```

```
<AxesSubplot:xlabel='exercise_induced_angina', ylabel='target'>
```



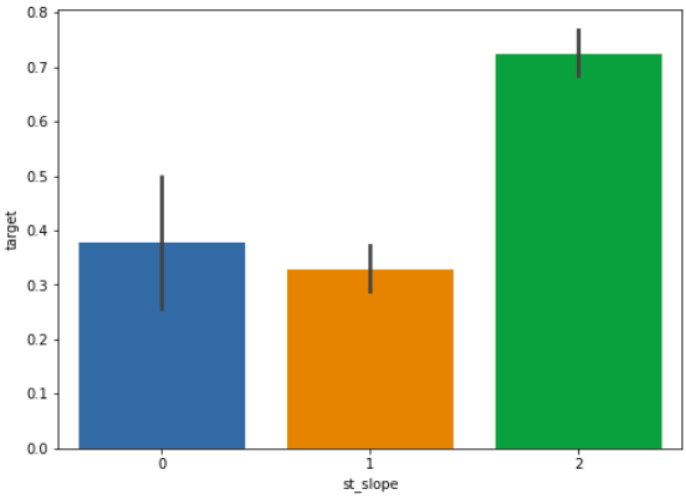
Analysing The Slope Of The Peak Exercise ST Segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)

```
data["st_slope"].unique()
```

```
array([2, 0, 1])
```

```
plt.figure(figsize=(8, 6))
sns.barplot(data["st_slope"],y)
```

```
<AxesSubplot:xlabel='st_slope', ylabel='target'>
```



Analysing Number Of Major Vessels (0-3) Colored By Flourosopy

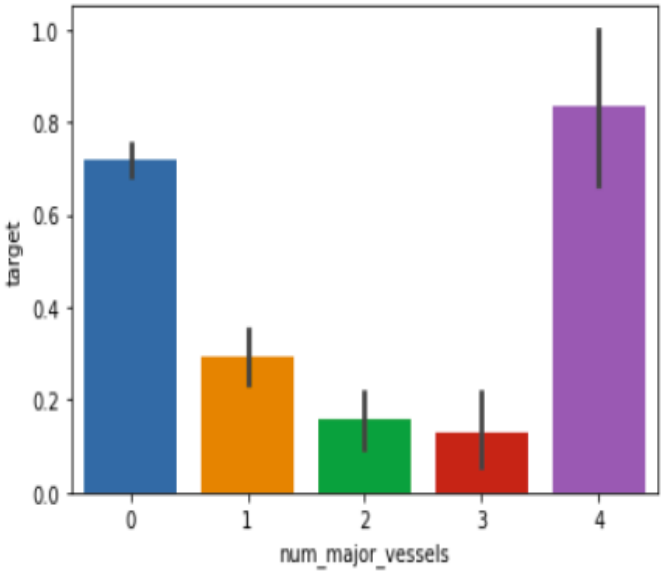
```
data["num_major_vessels"].unique()
```

```
array([2, 0, 1, 3, 4])
```

comparing with target

```
sns.barplot(data["num_major_vessels"],y)
```

```
<AxesSubplot:xlabel='num_major_vessels', ylabel='target'>
```



Analysing A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

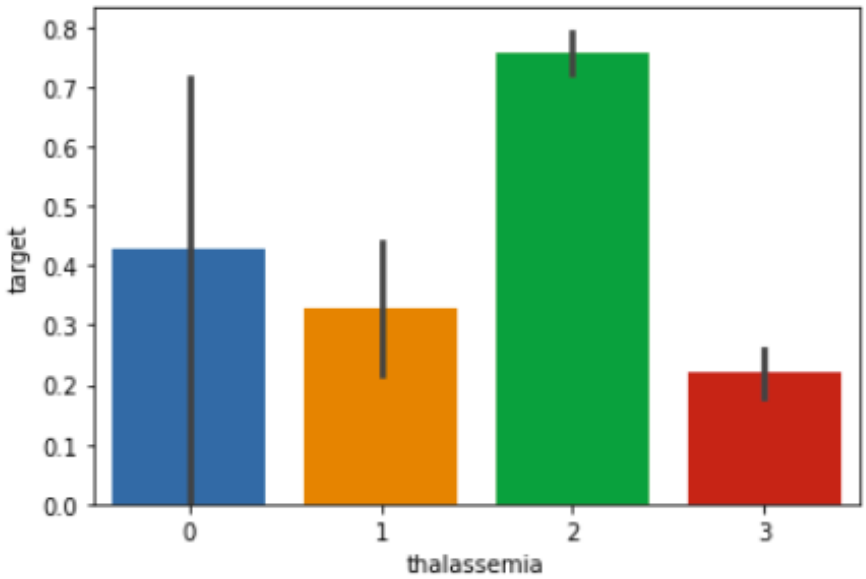
```
: data["thalassemia"].unique()
```

```
: array([3, 2, 1, 0])
```

comparing with target

```
: sns.barplot(data["thalassemia"],y)
```

```
: <AxesSubplot:xlabel='thalassemia', ylabel='target'>
```



Correlation plot

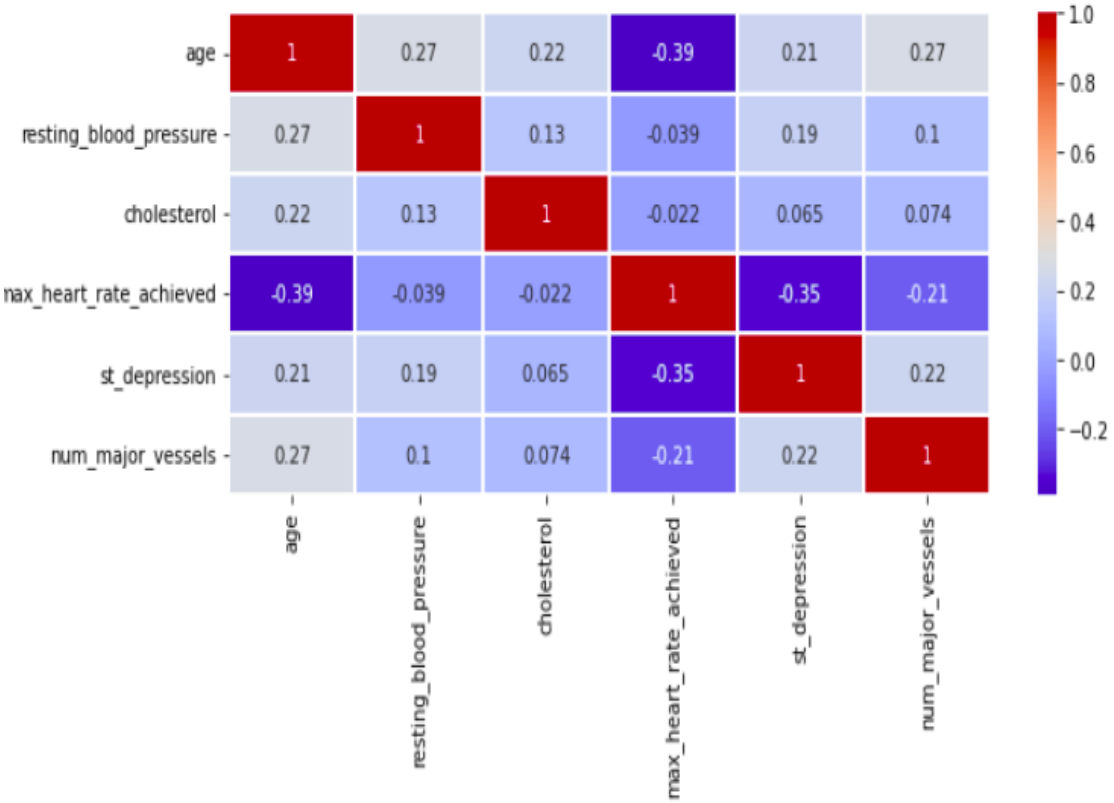
Correlation plot between some parameters using heatmap

```
# store numeric variables in cnames
cnames=['age','resting_blood_pressure','cholesterol','max_heart_rate_achieved','st_depression','num_major_vessels']
```

```
#Set the width and height of the plot
f, ax = plt.subplots(figsize=(10, 4))

#Correlation plot
df_corr = data.loc[:,cnames]
#Generate correlation matrix
corr = df_corr.corr()

#Plot using seaborn library
sns.heatmap(corr, annot = True, cmap='coolwarm',linewidths=.1)
plt.show()
```



Chapter 2

Different Disease Prediction Model

2.1 Introduction

Due to the status of the environment and their lifestyle choices, humans are today suffering from a wide range of diseases. To stop such diseases from

developing to their final stages, it is essential to identify and predict them early on. Most of the time, doctors find it difficult to accurately diagnose illnesses by hand. A variety of machine learning approaches are employed in different industries to do predictive analytics on enormous amounts of data. Although it is a very challenging process overall, predictive analytics helps clinicians make appropriate decisions about a patient's health and treatment based on the large amount of data accessible. numerous illnesses, including diabetes and cancer.are starting to kill people all around the world, and the main reason for this is because these diseases are not being detected early enough. Another significant reason is the dearth of a strong medical infrastructure and a low doctor-to-population ratio. According to WHO recommendations, there should be one doctor for every 1000 patients, however India has a ratio of one doctor for every 1456 people, indicating a doctor shortage in the country. Early detection and identification of these disorders is therefore essential to saving many lives. The main target of this research is to use machine learning classifier to forecast diseases.The goal of this study is to detect and forecast patients with more common diseases. Modern machine learning approaches may be used to achieve this, making sure that the categorization correctly identifies those who have ailments. Since many diseases might cause the same type of symptoms, predicting diseases is also a challenging task.

2.1.1 Problem Definition

A patient would visit a doctor, go through a number of tests, and then receive a diagnosis using the conventional method. This procedure takes a long time. In order to reduce the time needed for the initial disease

prediction process, which depends on user input, this research presents an automated disease prediction system. The user inputs data into the system, and the system returns to the user a list of likely ailments.

2.1.2 Objective

There is a need to study and make a system which will make it easy for an end users to predict diseases without visiting physician or doctor for diagnosis.To detect the Various Diseases through the examining Symptoms of patient's using different techniques of Machine Learning Models. The Proposed system will give the prediction based on symptoms .

2.1.3 Need of the system

There is always a need of a system that will provide the disease information according to symptoms shared by user.
This system will help the user to find disease according to their symptoms.
We always need such a system, if we are unwell then send a notification to the doctor of our city

2.2 Literature Survey

There is number of projects has been done related to disease prediction systems using various machine learning algorithms in medical field.

Narander Kumar and Sabita Khatri et al. [5], the project's authors, have investigated and compared various algorithms, including k-NN, Naive Bayes, Random Forest, and J48, using performance metrics like ROC, kappa statistics, RMSE, and MAE in WEKA tools, and have also compared the classifiers on various accuracy metrics. The study's findings indicated that Random

Forest has superior accuracy for the dataset utilised for chronic kidney disease.

Using structured and unstructured hospital data, M. Chen, Y. Hao at el. [3] developed and proposed a new convolutional neural network-based multimodal disease risk prediction (CNNMDRP) algorithm. To the best of our knowledge, no study has been done in the field of medical big data analytics that specifically addressed both data types. Our suggested algorithm's prediction accuracy is 94.8 percent, higher than several commonly used prediction algorithms, and it converges more quickly than the CNN-based unimodel disease risk prediction (CNN-UDRP) algorithm.

The data gathering method suggested by A. N. Repaka, S. D. Ravikanti at el. [6] is carried out utilising a variety of sources that are major causes of any type of heart disease, and as a result, using a structure, the database is built. The research focuses on developing SHDP (Smart Heart Disease Prediction), which addresses the problem of heart disease prediction by taking into account the method of NB (Naive Bayesian) classification and AES (Advanced Encryption Standard) algorithm. It is shown that the dominant approach outperforms the Naive Bayes in terms of accuracy by producing an accuracy of 89.77 percent while lowering the attributes. In comparison to PHEA (Parallel Homomorphic Encryption Algorithm), AES exhibits high security performance.

By putting up a computerized diagnostic model to support standardization and wide use of traditional Chinese medicine (TCM) diagnosis, Huiyan Wang [7] has sought out a niche in this sector (Huiyan Wang 2008). This is how the system functions. First, a database of cases with mutual information is used to learn the Bayesian network topology and choose the symptoms.theory. The target variable's Markov blanket is chosen as the symptom set in the structure. The mapping relationship is the second Based on naive Bayesian classifiers, the relationship between symptom sets and diagnostic outcomes.

2.3 Software Requirements

Technology:Python Django

IDE:Visual Studio Code

Client Side Technologies:HTML, CSS, JavaScript , Bootstrap

Server Side Technologies:Python

Data Base Server:Sqlite

OperatingSystem:Ubuntu/Windows

2.4 System Analysis

2.4.1 Purpose

User can search for doctor's help at any point of time.

User can talk about their illness and get instant diagnosis.

Informs the user about the type of disease or disorder it feels.

Doctors get more clients online.

Our system will notify the doctor about disease of patient

Our system will also predict severity level of heart disease.

Our system will also provide doctor list to the patient according to the patient city.

2.4.2 Flow Diagram For General Disease Prediction Model

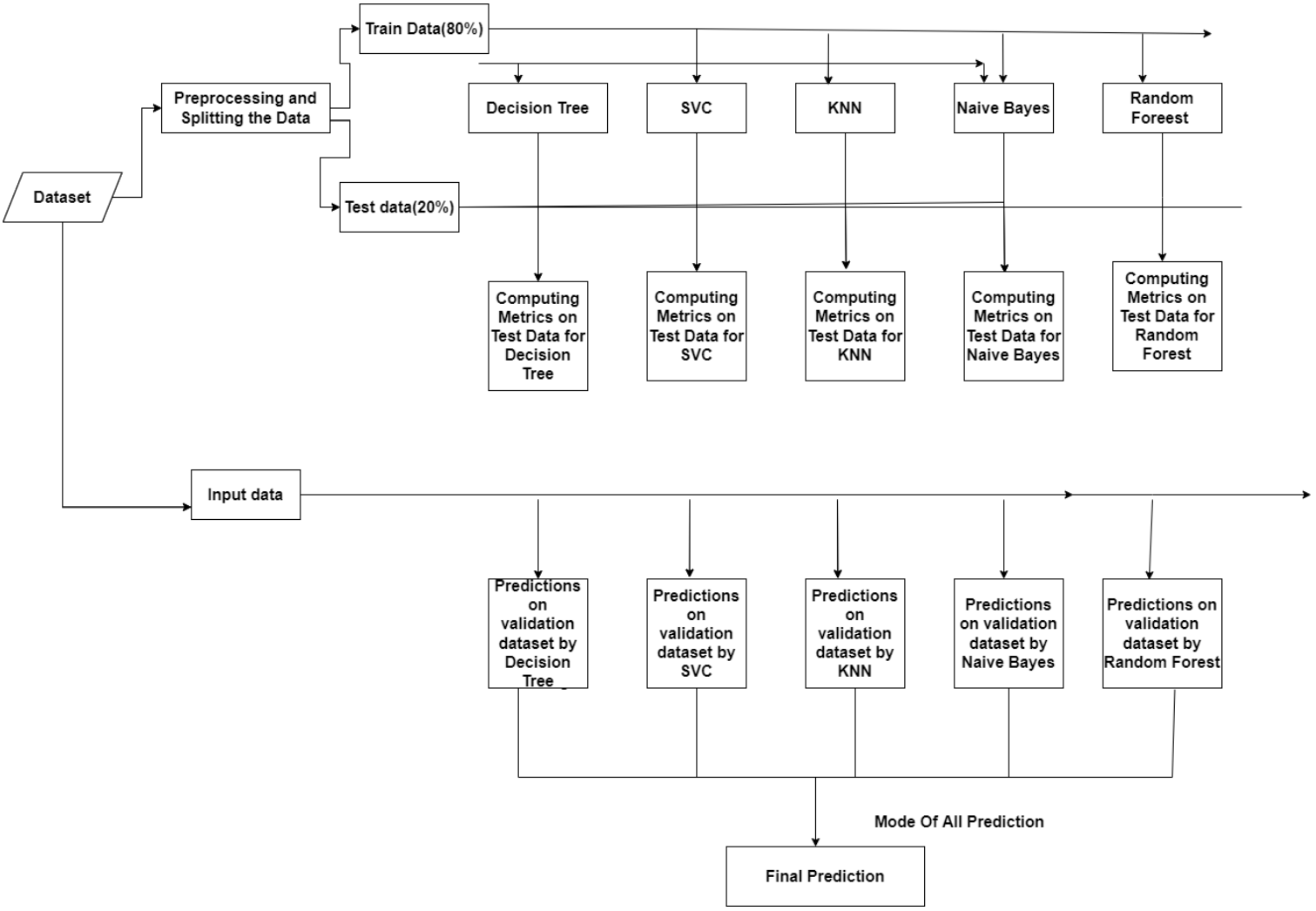


Figure 2.1: **Flow Diagram For General Disease Prediction Model**

2.4.3 Input User Health Parameter Page

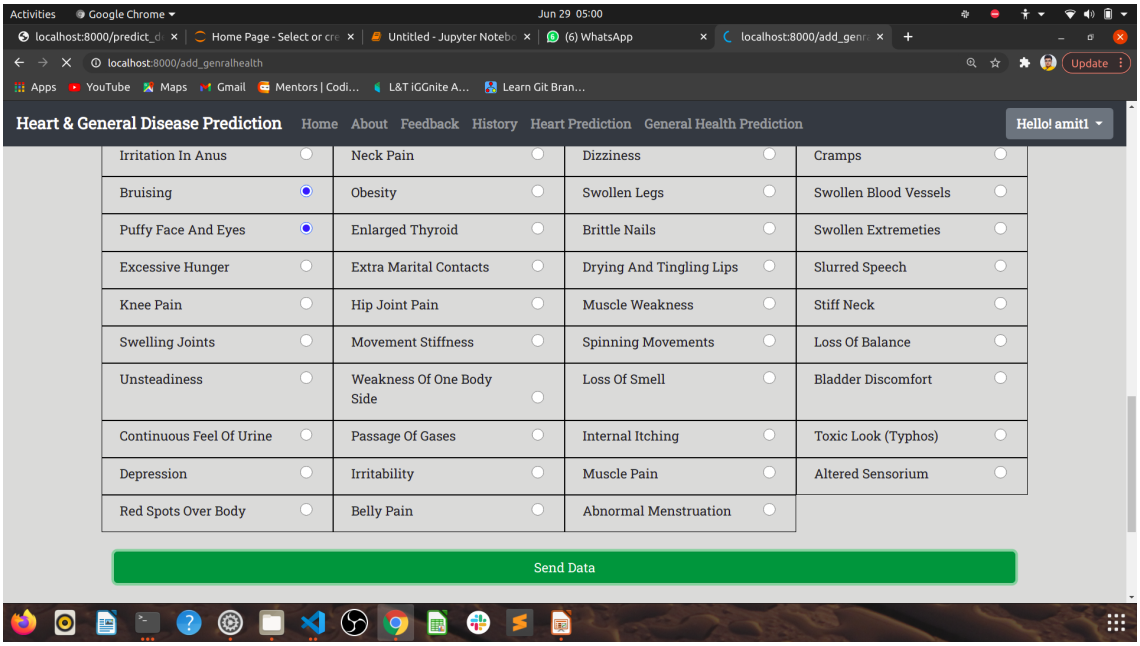


Figure 2.2: **Input User Health Parameter Page**

2.4.4 Result For General Disease Prediction Page

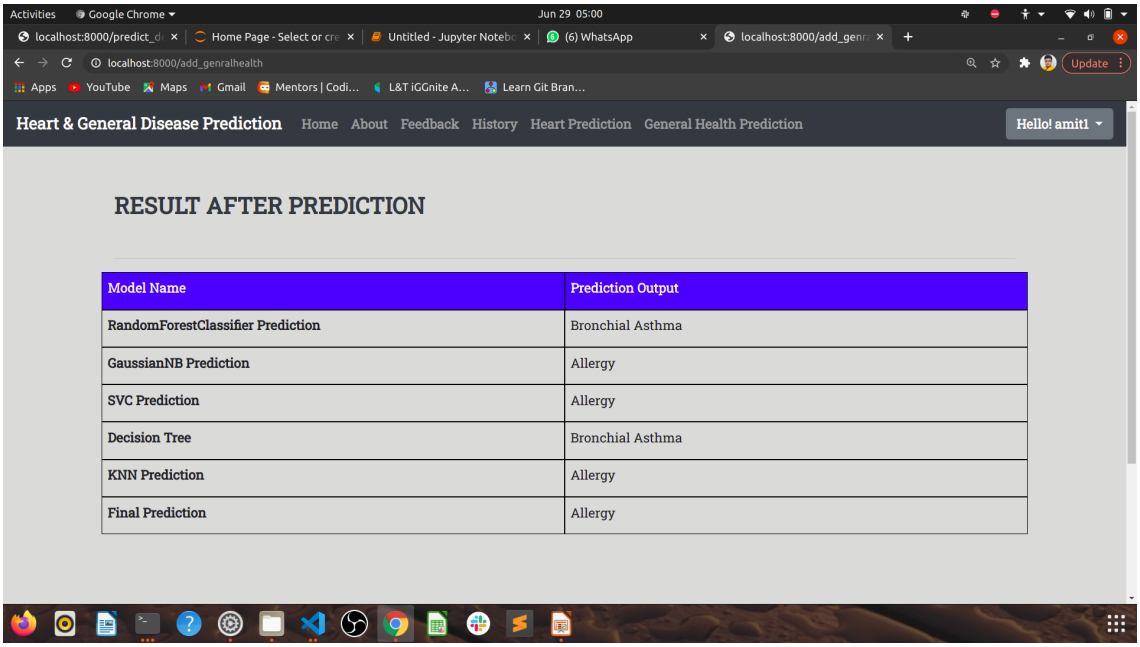


Figure 2.3: **Result Page For General Disease**

2.5 Other State Of Art Classifier Used In Our Framework

2.5.1 Random Forest Classifier

Popular machine learning algorithm Random Forest is a part of the supervised learning methodology. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance.

Random Forest, as the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the dataset's predictive accuracy. Instead than depending on a single decision tree, the random forest uses forecasts from each tree and predicts the result based on the votes of the majority of predictions.

there are some applications of random forest classifier

Banking Industry

Credit Card Fraud Detection

Customer Segmentation

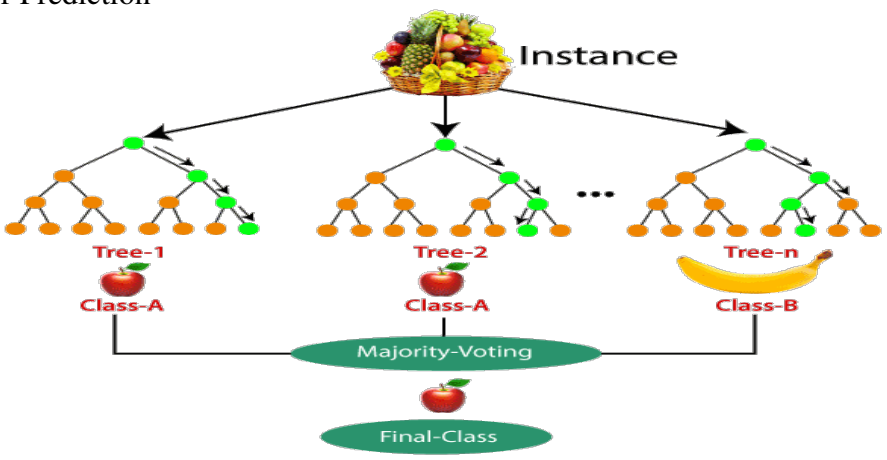
Predicting Loan Defaults on LendingClub.com

Healthcare and Medicine

Cardiovascular Disease Prediction

Diabetes Prediction

Breast Cancer Prediction



2.5.2 Decision Tree

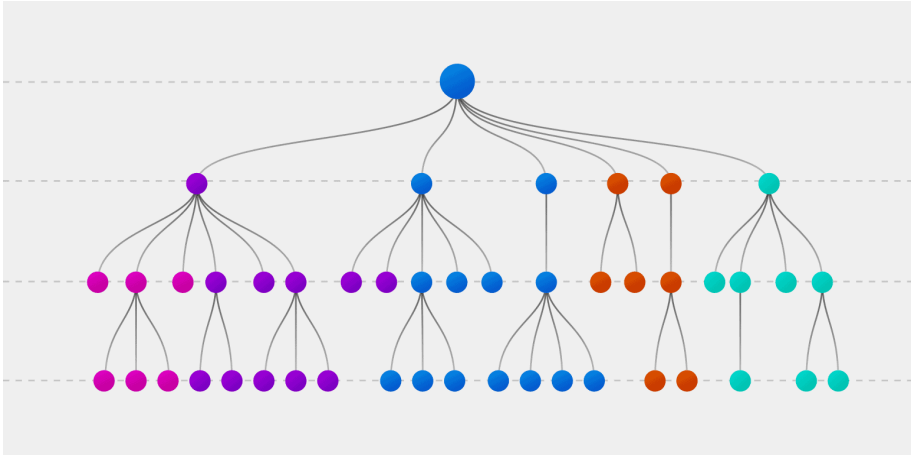
The supervised learning algorithms family includes the decision tree algorithm. The decision tree technique, in contrast to other supervised learning methods, is capable of handling both classification and regression issues.By learning straightforward decision rules derived from previous data, a Decision Tree is used to build a training model that may be used to predict the class or value of the target variable (training data).In decision trees, we begin at the tree's root when anticipating a record's class label. We contrast the root attribute's values with that of the attribute on the record. We follow the branch that corresponds to that value and go on to the next node based on the comparison.

Types of Decision Trees Types of decision trees are based on the type of target variable we have. It can be of two types:

Categorical Variable Decision Tree:A categorical variable decision tree is a decision tree with a categorical target variable.

Continuous Variable Decision Tree: If the decision tree's target variable is continuous, it is referred to as a continuous variable decision tree.

Example: Consider the situation where we must determine whether a consumer will pay his renewal premium to an insurance firm (yes/no). The insurance business does not have information on every client's income, despite the fact that we are aware that customer income is a crucial component in this situation. Now that we are aware of how crucial this variable is, we can create a decision tree to forecast client income based on their occupation, the product, and a number of other characteristics. We are speculating here on the values of the continuous variables.



2.6 Coding

2.6.1 KNN implementation for General Disease Prediction Model

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
```

In [89]:
```
df = pd.read_csv('/home/amityadav/Desktop/HealthDesease/media/Training.csv').dropna(axis = 1)
# print(df)
encoder = LabelEncoder()
df["prognosis"] = encoder.fit_transform(df["prognosis"])
```

In [3]:
```
df.head()
```

Out[3]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scu
0	0	0	0	0	0	0	0	0	0	0	...	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	

5 rows × 133 columns

In [4]:
```
df.describe()
```

Out[4]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_
count	4947.000000	4947.000000	4947.000000	4947.000000	4947.000000	4947.000000	4947.000000	4947.000000	4947.000000	4947.
mean	0.141096	0.162725	0.024257	0.044876	0.021831	0.161310	0.138266	0.046291	0.044876	0.
std	0.348155	0.369152	0.153862	0.207052	0.146148	0.367854	0.345213	0.210135	0.207052	0.
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.

8 rows × 133 columns

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4947 entries, 0 to 4946
Columns: 133 entries, itching to prognosis
dtypes: int64(133)
memory usage: 5.0 MB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
itching                 0
skin_rash               0
nodal_skin_eruptions    0
continuous_sneezing     0
shivering               0
                       ..
inflammatory_nails      0
blister                 0
red_sore_around_nose    0
yellow_crust_ooze       0
prognosis               0
Length: 133, dtype: int64
```

KNN implementation for Disease prediction

In [95]:
```
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier

knn = train_model(x_train, y_train, x_test, y_test, KNeighborsClassifier, n_neighbors=3)

knn.fit(x_train, y_train)

y_pred_knn = knn.predict(x_test)
print(y_pred_knn)
```

```
Train accuracy: 99.65%
Test accuracy: 99.71%
[25 16 25 ... 25  7  5]
```

In [96]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns

import matplotlib.pyplot as plt
```

In [105]:
```
matrix= confusion_matrix(y_test, y_pred_knn)
sns.heatmap(matrix,annot = True, fmt = "d")
```

Specificity , Sensitivity

In [98]:
```
total1=sum(sum(matrix))
Accuracy = (matrix[0,0]+matrix[1,1])/total1
Specificity = matrix[0,0]/(matrix[0,0]+matrix[0,1])
Sensitivity = matrix[1,1]/(matrix[1,0]+matrix[1,1])

print(Accuracy)
print(Specificity)
print(Sensitivity)
#df.loc[i] =[i,Accuracy,Sensitivity,Specificity]

#cutoff_df = pd.DataFrame( columns = ['Probability','Accuracy','Sensitivity','Specificity'])

plt.show()
```

```
0.049868766404199474
1.0
1.0
```

Precision Score

n [108]:
```
from sklearn.metrics import precision_score
precision = precision_score(y_test, y_pred_knn,pos_label='positive',average='micro')
print("Precision: ",precision)
```

```
Precision:  0.9971128608923885
```

Recall

n [109]:
```
from sklearn.metrics import recall_score
recall = recall_score(y_test, y_pred_knn,pos_label='positive',average='micro')
print("Recall is: ",recall)
```

```
Recall is:  0.9971128608923885
```

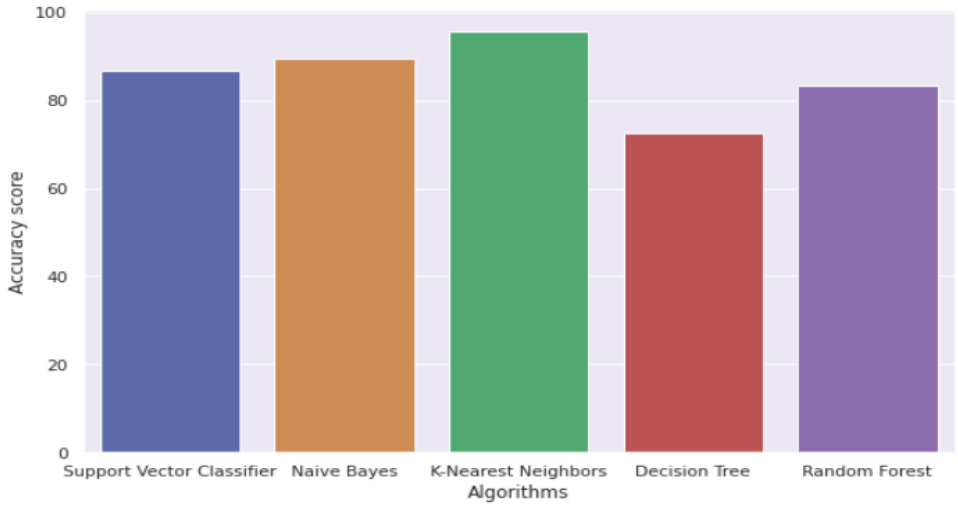
f score

n [110]:
```
print((2*precision*recall)/(precision+recall))
```

```
0.9971128608923885
```

2.6.2 Comparison between all algorithm used in general disease prediction

```
scores = [score_svc,score_nb,score_knn,score_dt,score_rf]
algorithms = ["Support Vector Classifier","Naive Bayes","K-Nearest Neighbors","Decision Tree","Random Forest"]
sns.set(rc={'figure.figsize':(10,6)})
plt.xlabel("Algorithms")
plt.ylabel("Accuracy score")
sns.barplot(algorithms,scores)
```



2.7 Comparative Analysis

Author	Dataset	Classification Technique used	Best Technique found	Accuracy Achieved	Application Domain
M chen and Y hao [6]	Hospitals	CNN-MDRP, CNN-UDRP	CNN-MDRP	94.8%	Disease Prediction
J. Gao and L. Tian [7]	Kaggle Dataset	Logistic Regression,SVC ,Adaboost	Adaboost	92.4%	Disease Prediction
A N Repaka and S.D RaviKanti [8]	UCI machine learning laboratory	Naïve Bayes	Naïve Bayes	89.71%	Disease Prediction
Our Model	Kaggle Dataset	Naive Bayes,Random Forest,Decisio	KNN	94.6%	Disease Prediction

Chapter 3

Advantage And Limitation Of Our Framework

3.1 Advantage

Framework provides various features, which complement the information system and increase the productivity of the system. These features make the system easily usable and convenient. Some of the important features included are listed as follows:

Intelligent User Forms Design

Data access and manipulation through same forms

Access to most required information

Data Security

Restrictive data access, as per login assigned only.

Organized and structured storage of facts.

Strategic Planning made easy.

No decay of old Records.

Exact financial position of the Business.

User can search for doctor's help at any point of time.

User can talk about their illness and get instant diagnosis.

Doctors get more clients online.

3.2 Limitations

Besides the above achievements and the successful completion of the project, we still feel the project has some limitations, listed as below:

It is not a large scale system.

Only limited information provided by this system.

Since it is an online project, users need internet connection.

People who are not familiar with computers can't use this software.

The system is not fully automated; it needs doctors for full diagnosis

Chapter 4

Conclusion

I have gained a lot of knowledge from this project. I have greatly enjoyed working on this fascinating and difficult subject. This project worked out well for me because it gave me practical programming experience in Python and Sqlite web applications. Additionally, it offers information on client server technology, which will be in high demand in the future, and the most recent technology employed in creating web-enabled applications. This will give future project developers better opportunities and direction for working autonomously.

Bibliography

[1] Verma, Prabal, and Sandeep K. Sood. ”Fog assisted-IoT enabled patient health monitoring in smart homes.” IEEE Internet of Things Journal 5.3 (2018): 1789-1796.

[2] Yahaya, Lamido, Nathaniel David Oye, and Etemi Joshua Garba. ”A comprehensive review on heart disease prediction using data mining and machine learning techniques.” American Journal of Artificial Intelligence 4.1 (2020): 20-29.

[3] Vijayashree, J., SrimanNarayanaIyengar, N. C. (2016). Heart disease prediction system using data mining and hybrid intelligent techniques: A review. International Journal of Bio-Science and Bio-Technology, 8(4), 139-148.

[4] Barik, S., Mohanty, S., Rout, D., Mohanty, S., Patra, A. K., Mishra, A. K. (2020). Heart disease prediction using machine learning techniques. In Advances in Electrical Control and Signal Systems (pp. 879-888). Springer, Singapore.

[5] Yekkala, I., Dixit, S. (2018). Prediction of heart disease using random forest and rough set based feature selection. International Journal of Big Data and Analytics in Healthcare (IJBDAH), 3(1), 1-12.

[6] Yang, L., Wu, H., Jin, X., Zheng, P., Hu, S., Xu, X., ... Yan, J. (2020). Study of cardiovascular disease prediction model based on random forest in eastern China. Scientific reports, 10(1), 1-8.

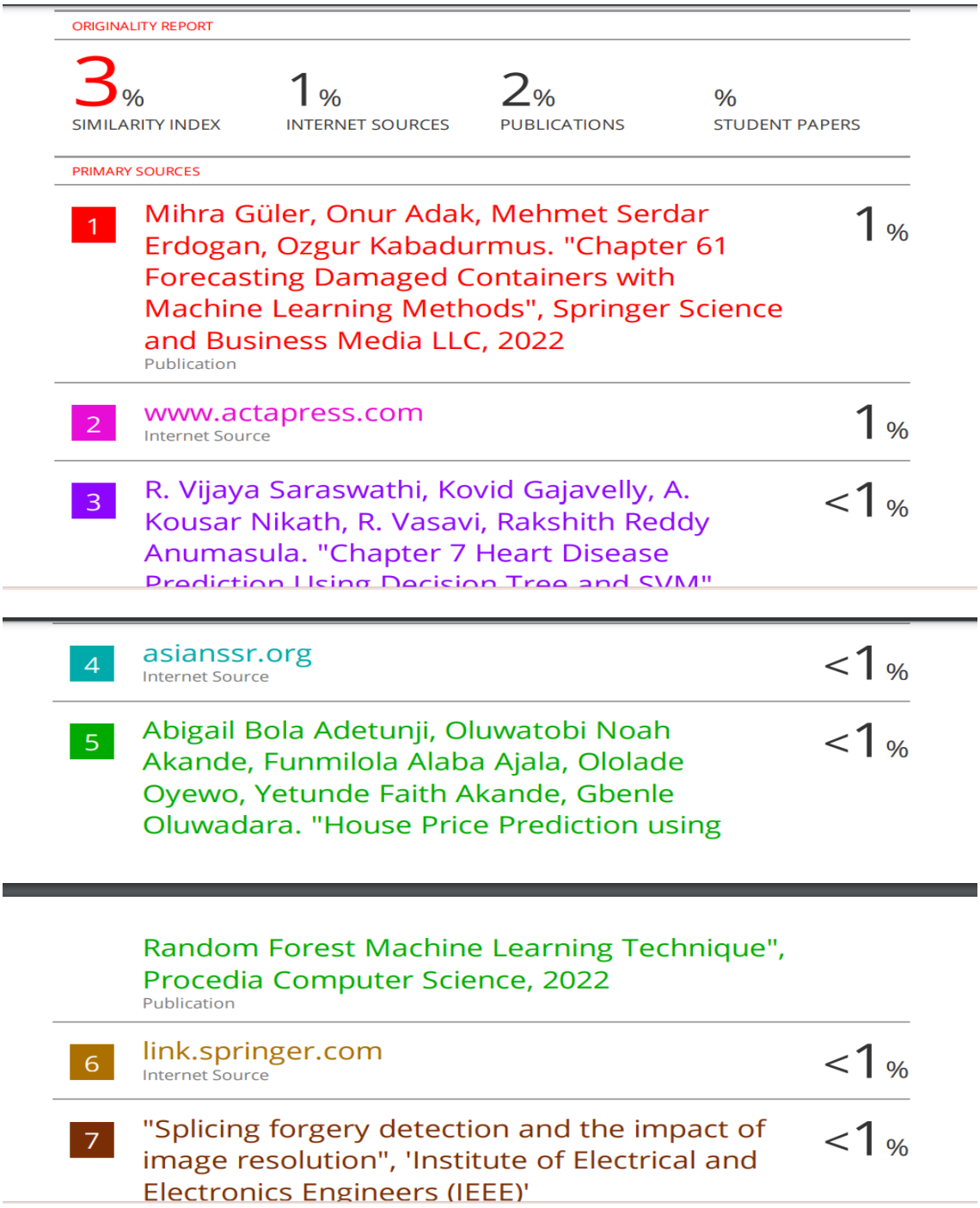
[7] Kim, K. M., Kim, B. T., Lee, D. J., Park, S. B., Joo, N. S., Kim, K. N. (2012). Serum gamma-glutamyltransferase as a risk factor for general cardiovascular disease prediction in Koreans. Journal of Investigative Medicine, 60(8), 1199-1203.

[8] Yu, X., Zhang, J., Sun, S., Zhou, X., Zeng, T., Chen, L. (2017). Individual-specific edge-network analysis for disease prediction. Nucleic acids research, 45(20), e170-e170.

[9] Zhan, Y., Holtfreter, B., Meisel, P., Hoffmann, T., Micheelis, W., Dietrich, T., Kocher, T. (2014). Prediction of periodontal disease: modelling and validation in different general German populations. Journal of Clinical Periodontology, 41(3), 224-231.

[10] Ramalingam, V. V., Dandapath, A., Raja, M. K. (2018). Heart disease prediction using machine learning techniques: a survey. International Journal of Engineering Technology, 7(2.8), 684-687.

4.1 Plagrism Report



ORIGINALITY REPORT

3%	1%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Mihra Güler, Onur Adak, Mehmet Serdar Erdogan, Ozgur Kabadurmus. "Chapter 61 Forecasting Damaged Containers with Machine Learning Methods", Springer Science and Business Media LLC, 2022 Publication	1%
2	www.actapress.com Internet Source	1%
3	R. Vijaya Saraswathi, Kovid Gajavelly, A. Kousar Nikath, R. Vasavi, Rakshith Reddy Anumasula. "Chapter 7 Heart Disease Prediction Using Decision Tree and SVM"	<1%
4	asianssr.org Internet Source	<1%
5	Abigail Bola Adetunji, Oluwatobi Noah Akande, Funmilola Alaba Ajala, Ololade Oyewo, Yetunde Faith Akande, Gbenle Oluwadara. "House Price Prediction using Random Forest Machine Learning Technique", Procedia Computer Science, 2022 Publication	<1%
6	link.springer.com Internet Source	<1%
7	"Splicing forgery detection and the impact of image resolution", 'Institute of Electrical and Electronics Engineers (IEEE)'	<1%