Chapter 1

Introduction

1.1 Introduction

In this digital world, most people are prone to diseases, due to a lack of healthy food, proper sleep, and daily exercise. It is very crucial to know if we are suffering from a disease, at an early stage rather than discovering it at a later stage.

According to recent reports many young people from the age groups of 25-35 are suffering from heart attack. Hence heart disease prediction system plays an important role as it predicts diseases based on symptoms. This cardiovascular disease prediction system uses machine learning algorithms like Random Forest, Logistic regression, SVM, Naïve Bayes, Decision tree classifier, Neural Network, MLP, Perceptron, KNN. This system also suggests the amount of risk that a person has.

With the advancement in technology, Machine Learning is becoming more popular and commonly used technology by industry experts for solving problems faced in real life. Machine Learning is the scientific study of algorithms and statistical models that computer uses to perform a specific task without using explicit instructions, relying on patterns and inference instead. Machine Learning is also used by the healthcare industry to bring advancement in their techniques so that they can provide better services to their patients. The heart disease prediction system predicts the severity of the diseases based on the patient's symptoms.

1.2 Choice of Topic with reasoning/Need of Project

The rationale for choosing this topic is that Heart Disease Prediction systems have the potential to greatly help the society to predict any kind of heart disease. Additionally, Heart Disease prediction systems can provide valuable insights to user. Nowadays, the younger generation is also facing the Heart problems due to lack of awareness and knowledge related to heart disease.

The idea behind the project is that this system can help people discover a disease that they are suffering from. Additionally, Heart Disease Prediction systems can help to promote the awareness of dangerous Diseases. Prediction systems are a valuable tool for both medical field professionals and Society.

The study will identify and analyze symptoms and identify major heart diseases from symptoms which people mostly neglect. In addition, the study will also identify and analyze the minor issues and challenges associated with disease prediction. The project aims to predict the severity of heart disease that a person is suffering from.

1.3 Problem Statement

The WHO reports that heart-related disorders are on the rise. Due to this, 17.9 million individuals pass away annually. Many people neglect early signs of illnesses that, in the long run, can be fatal. There are instruments that can predict heart disease, but they are either expensive or ineffective at estimating the likelihood that heart disease will occur in humans. They also do not provide a risk percentage related to the disease. Calculating the risk of the disease is just as crucial as diagnosing it. The project provides an implementation of machine learning for identifying heart diseases and understanding the risk percentage at an early stage.

Chapter 2
Proposed System

2.1 Objectives

1. To Study existing system.
2. To make use of common clinical data to create a high-performing and economical ML-based heart disease prediction system
3. To provide risk percentage.
4. To compare the performance between previous and existing system.

2.2 Requirement Engineering

To study the system, you need to collect facts. Facts are expressed in qualitative form called as data. Success of any requirement any investigation depends upon availability of accurate and reliable data. These depend on appropriate method chosen for data collection. The specific methods used for collecting data are fact finding techniques.

The different methods used by analyst are:

Interview

Onside

Observation

Record

Review

Questionary

In this project I am using the method of:

Interview:

Interview technique is used to collect information from individual or from groups. Analyst should select respondent how are related to system under study. In this method interviewer that is analyst seats face to face with respondent and record his responses.

The information collected is likely to be more accurate and reliable because the interviewer can clear up their doubts and crass check the despondence. This method also helpsto find the area of misunderstanding, unrealistic expectations and future problems of the prosesystem.

Observation:

Unlike the other fact-finding technique, in this method the analyst himself visits the organization on observes and understands the flow of document, working of requirement system, the users of the system etc. For this method to be adopted it takes and analyst to perform this job as he knows which points should be noticed and highlighted. In analyst may observe the unwanted things as well and simply cause delay in the development of the new system.

2.3 Requirement Gathering

The waterfall model is a sequential (non-iterative) design process, used in softwaredevelopment process, in which process is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation & maintenance. Despite the development of new software development process models, the waterfall model is still the dominant process model with over a third of software developers still using it.

2.4 Software Requirement

The software requirements are description of features and functionalities of the target system. SRS defines how the intended software will interact with hardware, external interfaces,speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc. It is the responsibility of system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team.

SRS should come up with following features:

- User Requirements are expressed in natural language.
- Technical requirements are expressed in structured language, which is sued inside theorganizations.
- Design description should be written in Pseudo code.
- Format of Forms and GUI screen prints.
- Conditional and mathematical notations for DFDs etc.
- Technical requirements are expressed in structured language.
- Format of Forms and GUI screen prints.

Broadly software requirements should be categorized in two categories:

Functional Requirements:

Requirements, which are related to functional aspect of software fall into this category. Theydefine functions and functionality within and from the software system.

Non-Functional Requirements:

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of software, which users make assumption of.

Software Requirement:

What is Flask?

Flask is a web application framework written in Python. It was developed by Armin Ronacher,who led a team of international Python enthusiasts called Pocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects.

Flask is a web framework, it's a Python module that lets you develop web applications easily.It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features.

It does have many cool features like URL routing, template engine. It is a WSGI web app framework.

WSGI: The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

Werkzeug: Werkzeug is a WSGI toolkit that implements requests, response objects, and utilityfunctions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

Database Requirement:

Introduction to MySQL server:

- MySQL is a relational database management system.
- MySQL is open-source.
- MySQL is free.
- MySQL is ideal for both small and large applications.
- MySQL is very fast, reliable, scalable, and easy to use.
- MySQL is cross-platform.
- MySQL is compliant with the ANSI SQL standard.
- MySQL was first released in 1995.

Features of MYSQL Server:

Open Source

Quick and Reliable

Scalable

Data Types

Character Sets

Secure

Supports Large Databases

Chapter 3
System Analysis

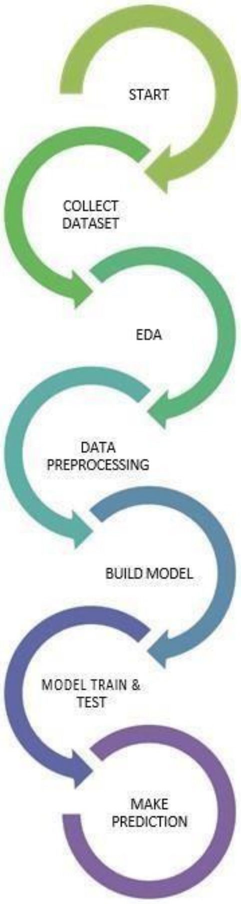
3.1 System Designs

The system design consists of machine learning model in the backend while the frontend is a website wherein users can enter their details in order to get the output.

System Architecture

- **Chatbot Service***:* Using this service users will register with a web application and have the option to use a chatbot to get an automatic response from the trained question and answer data. The chatbot will be trained using LSTM and dialog flow.
- **Online Analysis***:* The users will receive an analysis of their reports with the help of the website. Once they enter their report details into the interface, they will receive the analysis and the risk percentage.

The system design is as follows –



- **Collecting dataset** – This step involves collecting dataset manually which I referred from various sources. The dataset contains 7k values.

- **EDA** – This phase involves understanding the dataset and using libraries like matplotlib and seaborn to visualize the variables in the dataset.

- **Data preprocessing** – In this phase we removed and cleaned the dataset to remove all the null values. The processing of categorial values is also done in this step

- **Building model** – In this step we build a model using several algorithms. The algorithms that we have used in this project are Logistic Regression, SVM, KNN, Random Forest, Naïve Bayes, Neural Network, MLP, Perceptron and Decision Tree Classifier.

- **Model train and test** – In this step we train and test the dataset by firstly splitting it in a specific ratio. We have split our dataset using sklearn library in the ratio of 1:4, i.e., 80% of data for training and remaining 20% for testing.

- **Making prediction** – After making sure that the model works properly for while testing it, we can deploy the model for making predictions. For this, we are going to create a website containing fields for taking information from the user. From the values entered by the user the model will make further predictions.

3.2 Methodology/Algorithm

3.2.1 Logistic Regression

Independent variables are analyzed to determine the binary outcome with the results falling into one of two categories. The independent variables can be categorical or numeric, but the dependent variable is always categorical.

In logistic regression, we fit a "S" shaped logistic function, which predicts two maximum values, rather than a regression line (0 or 1).

The logistic function's curve shows the possibility of several things, including whether or not the cells are malignant, whether or not a mouse is obese depending on its weight, etc.

Because it can classify new data using both continuous and discrete datasets, logistic regression is a key machine learning approach.

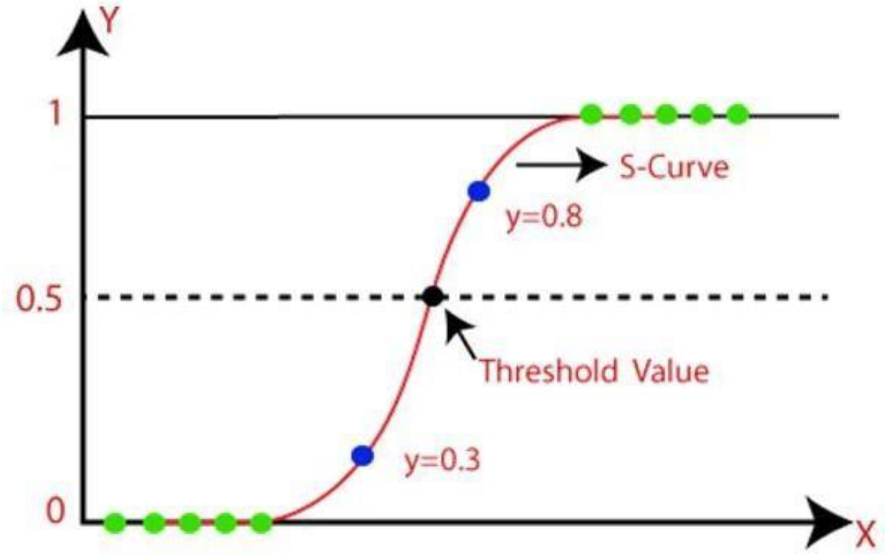


Fig. Logistic Regression

3.2.2 K-Nearest Neighbors

K-nearest neighbors (k-NN) is a pattern recognition algorithm that uses training datasets to find the k closest relatives in future examples. When k-NN is used in classification, you calculate to place data within the category of its nearest neighbor. If k =1, then it would be placed in the class nearest 1. K is classified by a plurality poll of its neighbors.

The K-NN algorithm assumes that the new case and the existing cases are comparable, and it places the new instance in the category that is most like the existing categories.

A new data point is classified using the K-NN algorithm based on similarity after all the existing data has been stored. This means that utilizing the K-NN method, fresh data can be quickly and accurately sorted into a suitable category.

It is also known as a lazy learner algorithm since it saves the training dataset rather than learning from it immediately. Instead, it uses the dataset to perform an action when classifying data.

The KNN method simply saves the information during the training phase, and when it receives new data, it categorizes it into a category that is quite like the new data.

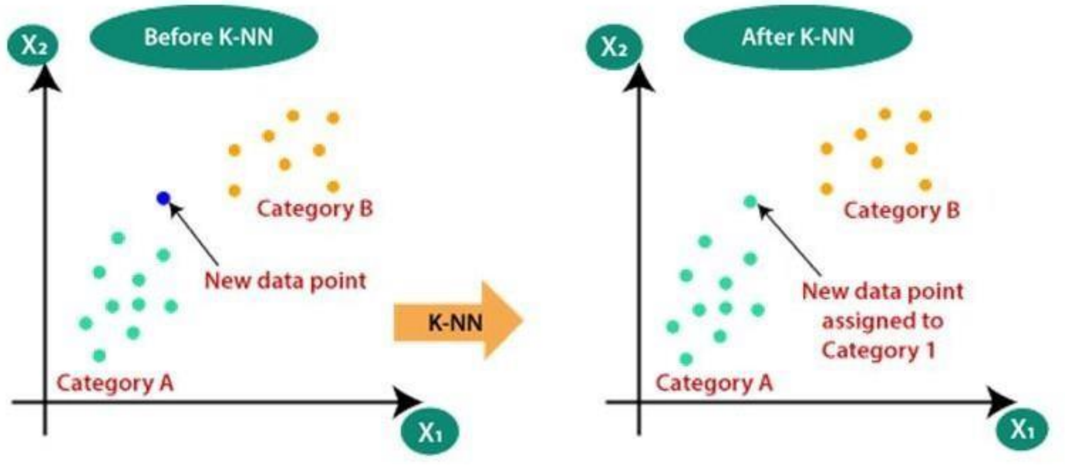


Fig. K-Nearest Neighbors

3.2.3 Random Forest

The random forest algorithm is an expansion of decision tree, in that you first construct a multitude of decision trees with training data, then fit your new data within one of the trees as a "random forest". It, essentially, averages your data to connect it to the nearest tree on the data scale.

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, then 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.

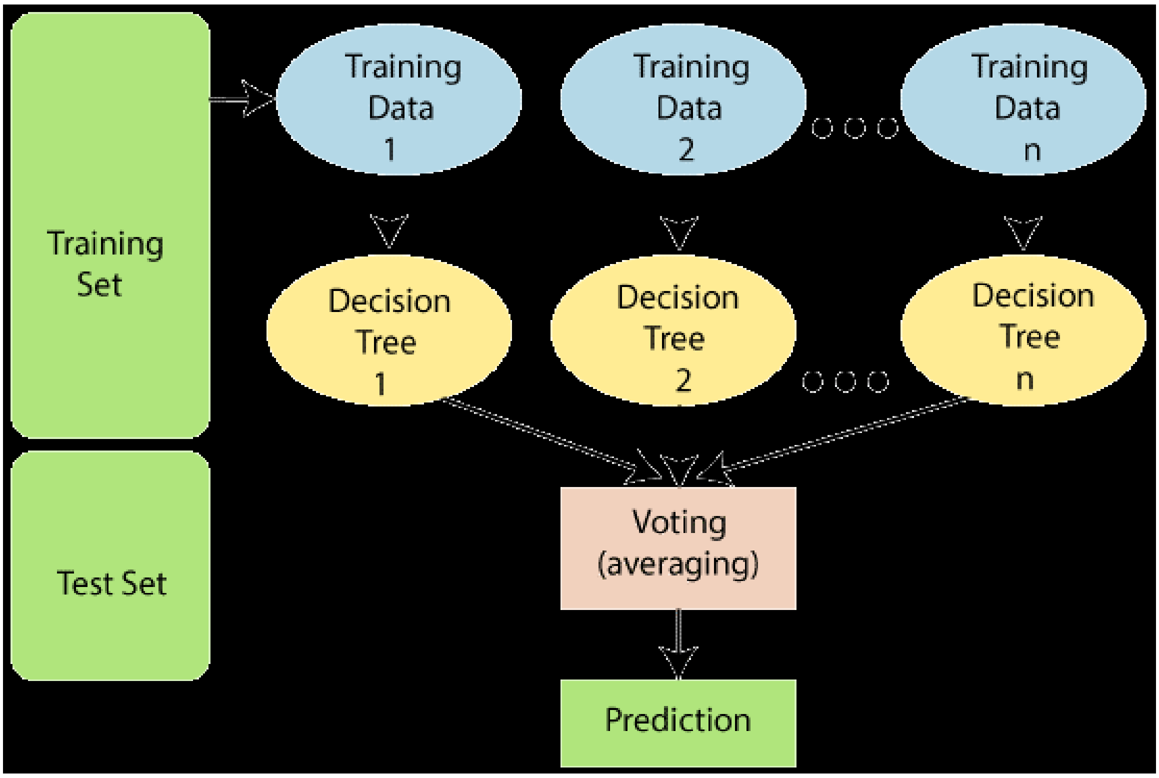


Fig. Random Forest

For the dataset's feature variable to predict true outcomes rather than a speculated result, there should be some actual values in the dataset.

3.2.4 Decision Tree

A decision tree is a supervised learning algorithm that is perfect for classification problems, as it’s able to order classes on a precise level. It works like a flow chart, separating data points into two similar categories at a time from the “tree trunk” to “branches,” to “leaves,” where the categories become more finitely similar. This creates categories within categories, allowing for organic classification with limited human supervision.

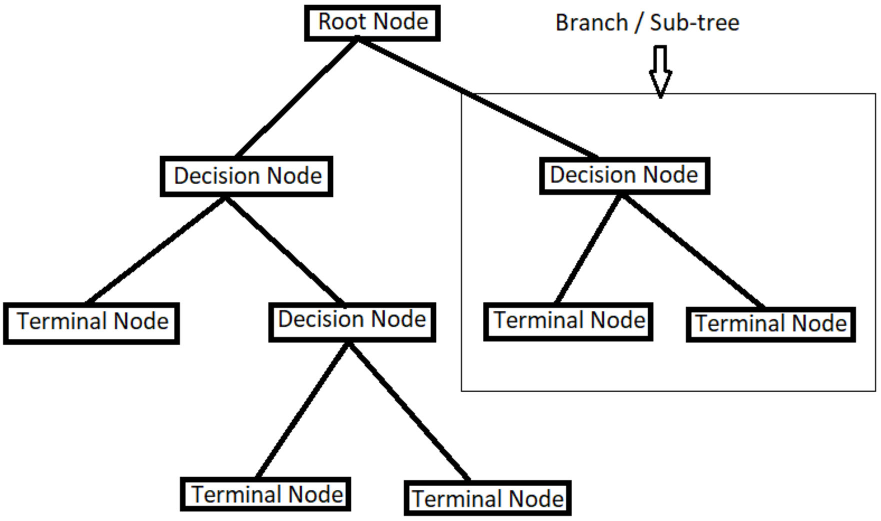


Fig. Decision Tree

3.2.5 Support Vector Machine

A support vector machine (SVM) uses algorithms to train and classify data within degrees of polarity, taking it to a degree beyond X/Y prediction. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method.

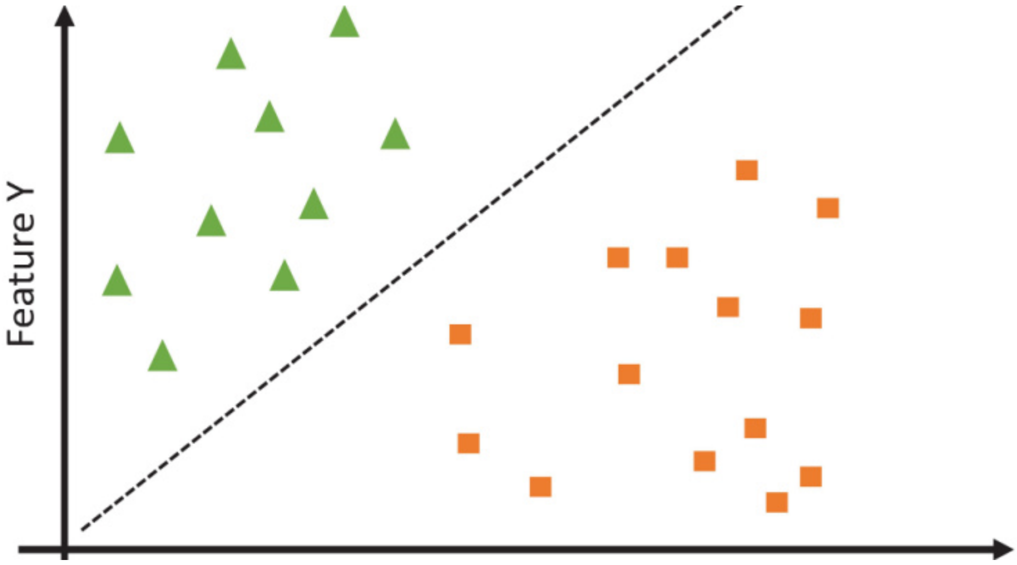


Fig. Support Vector Machine

3.2.6 Naïve Bayes

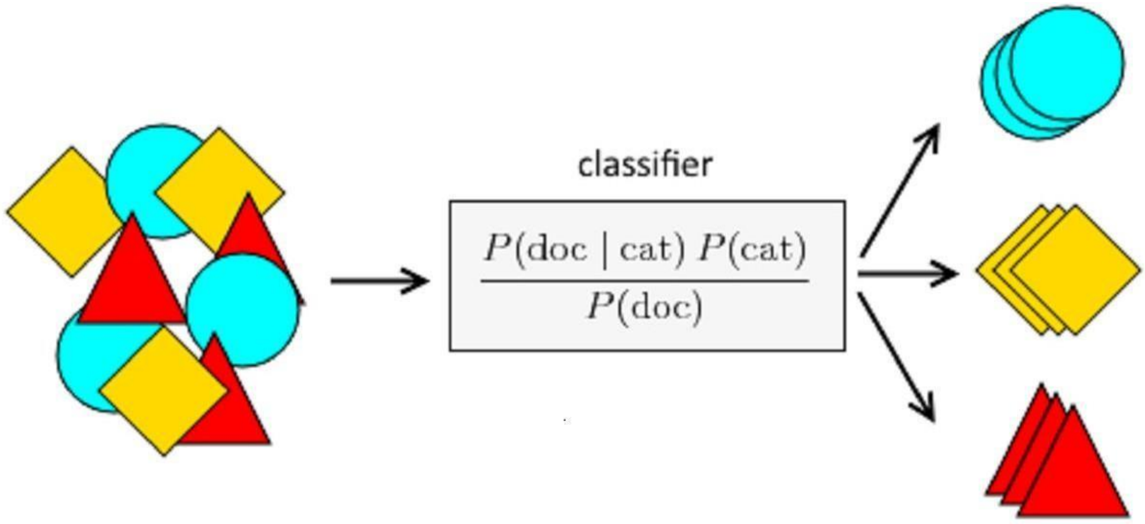

classifier
$\frac{P(\mathrm{doc} \mid \mathrm{cat})\, P(\mathrm{cat})}{P(\mathrm{doc})}$

Fig. Naïve Bayes

This algorithm is a supervised learning method for classification issues that is based on the Bayes theorem. It is mostly employed in text categorization with a large training set. The Naive Bayes Classifier is one of the most straightforward and efficient classification algorithms 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. The Bayes theorem, also referred to as Bayes' Rule or Bayes' law, is used to calculate the likelihood of a hypothesis given some prior information. The conditional probability determines this.

$$\mathrm{P(A|B)} = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

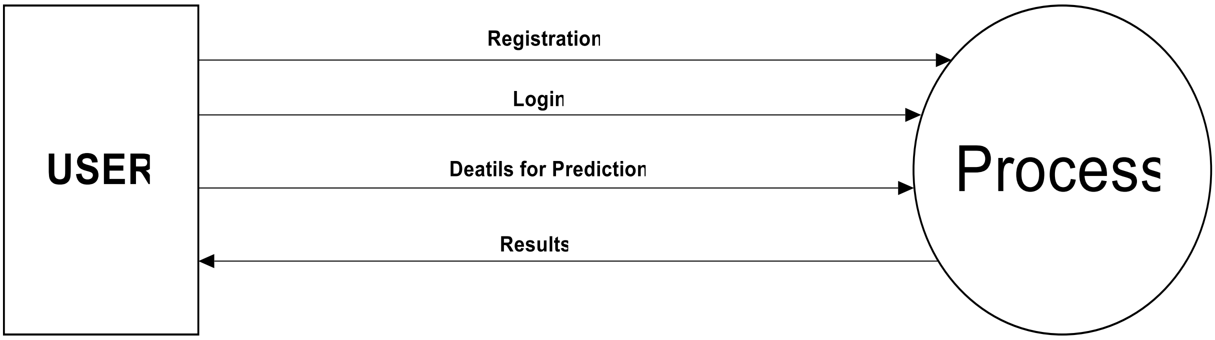
P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

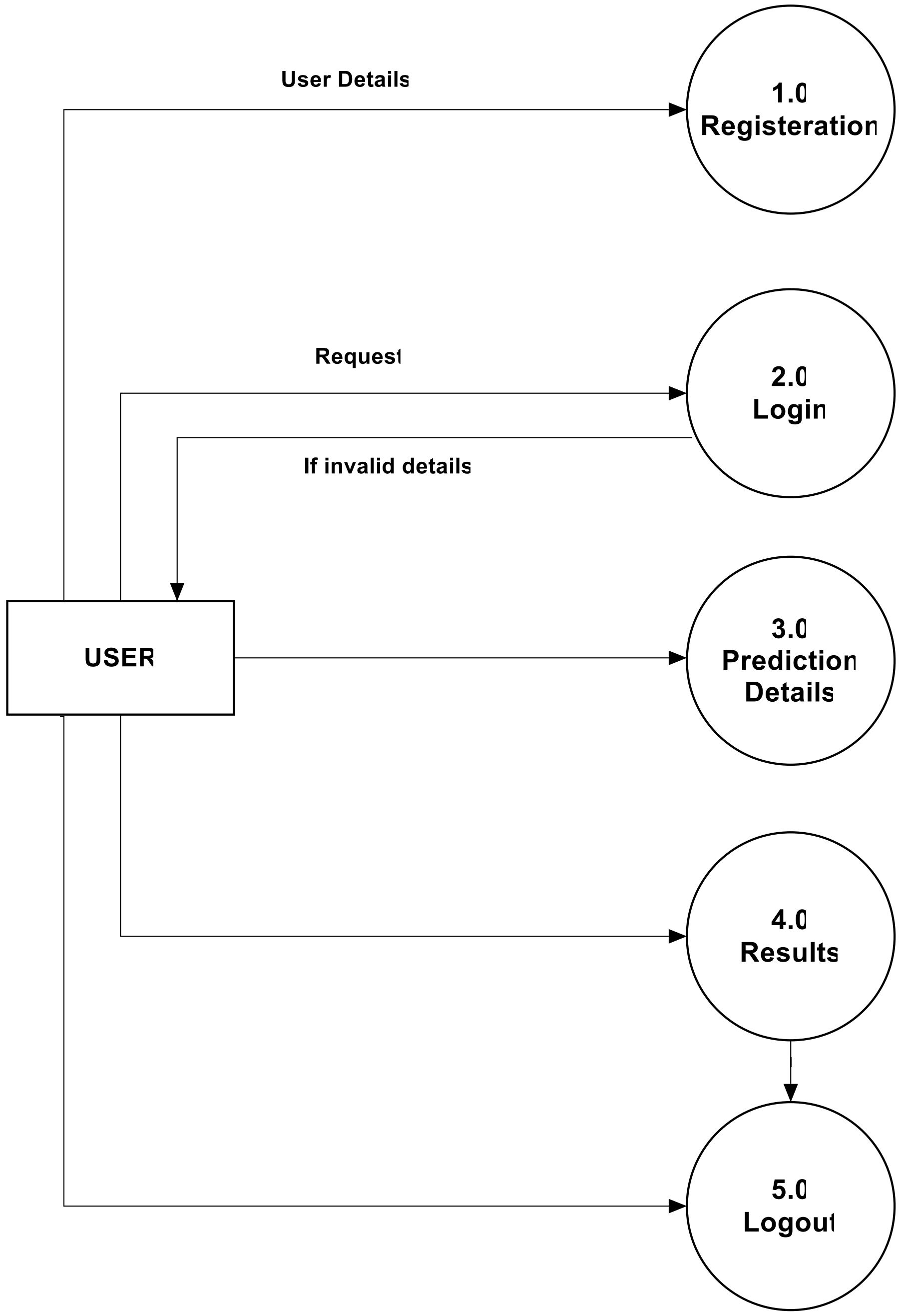
We are going to implement the above given algorithms and choose one that gives the highest accuracy for making predictions.

3.3 Data Flow Diagram

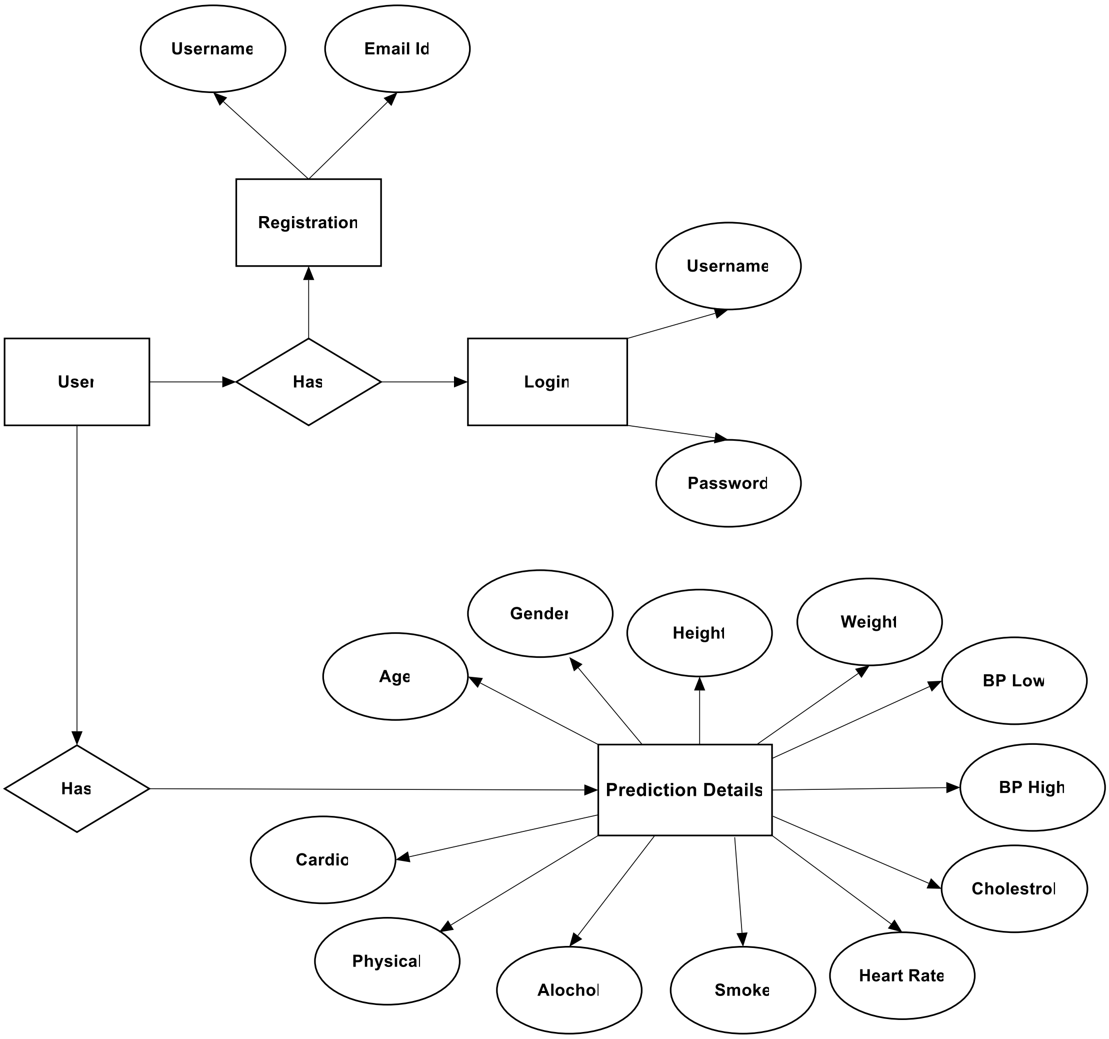
3.3.1 Context Level DFD



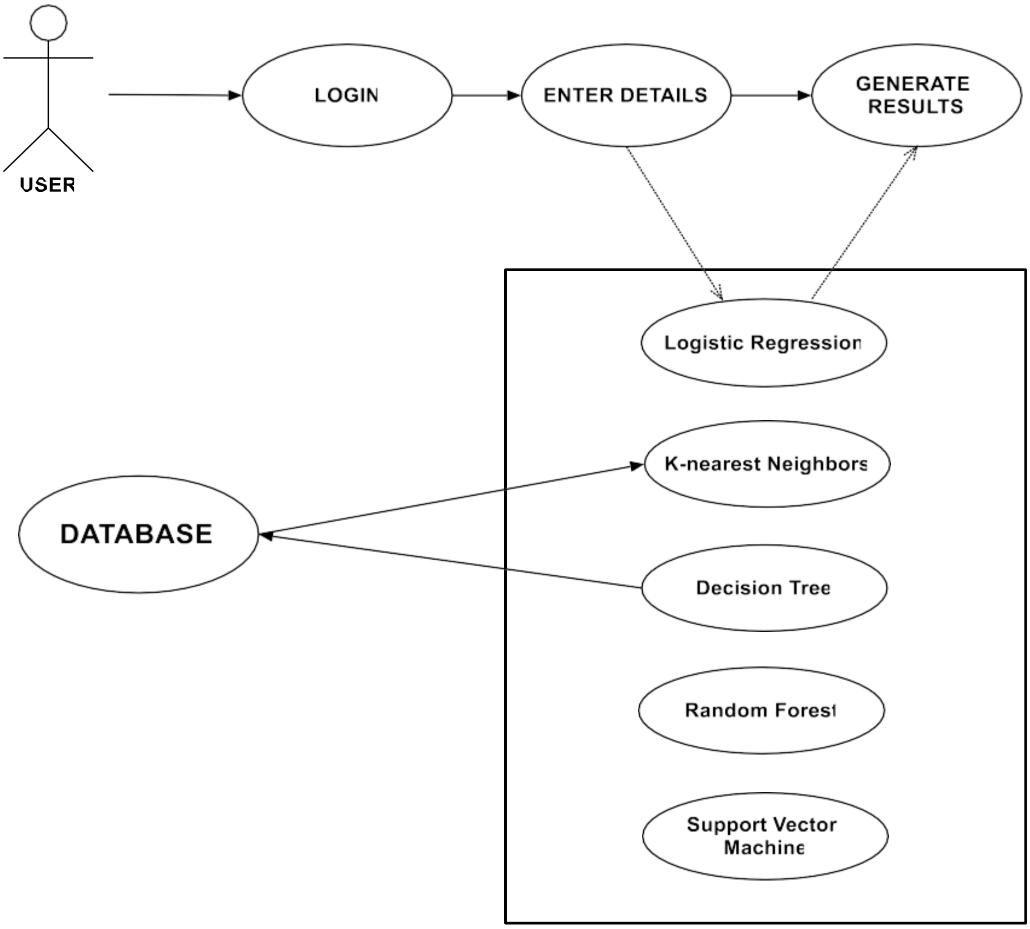
3.3.2 First Level DFD



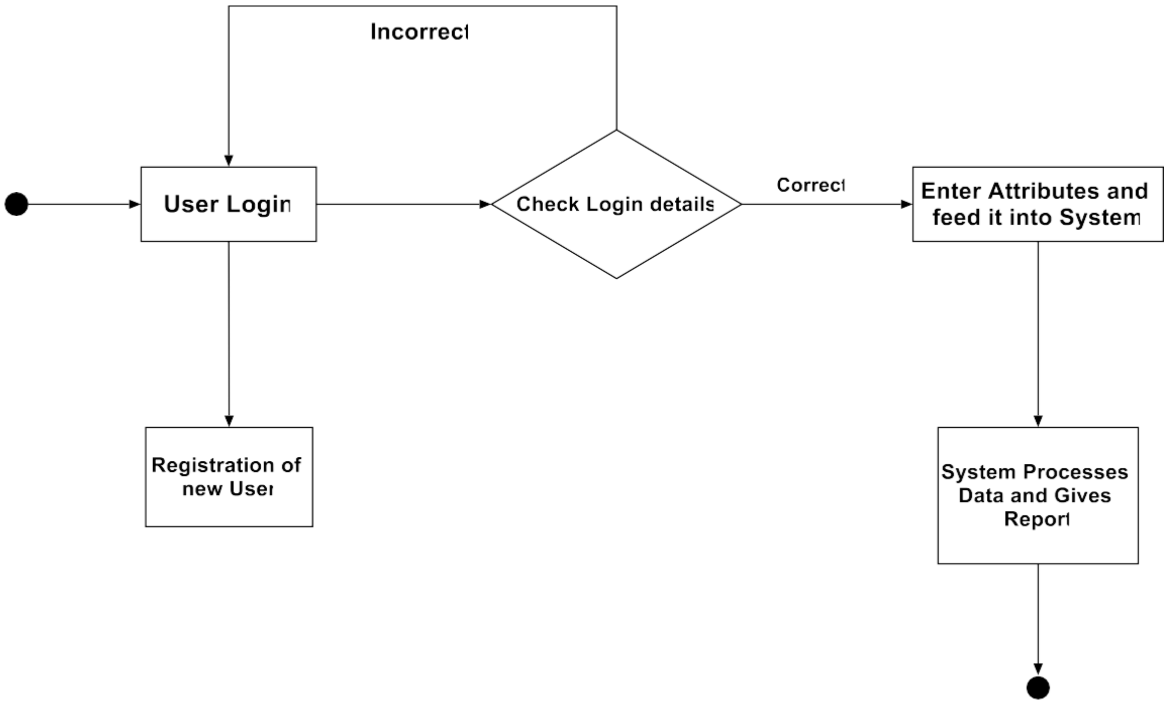
3.4 Entity Relationship Diagram (ERD)



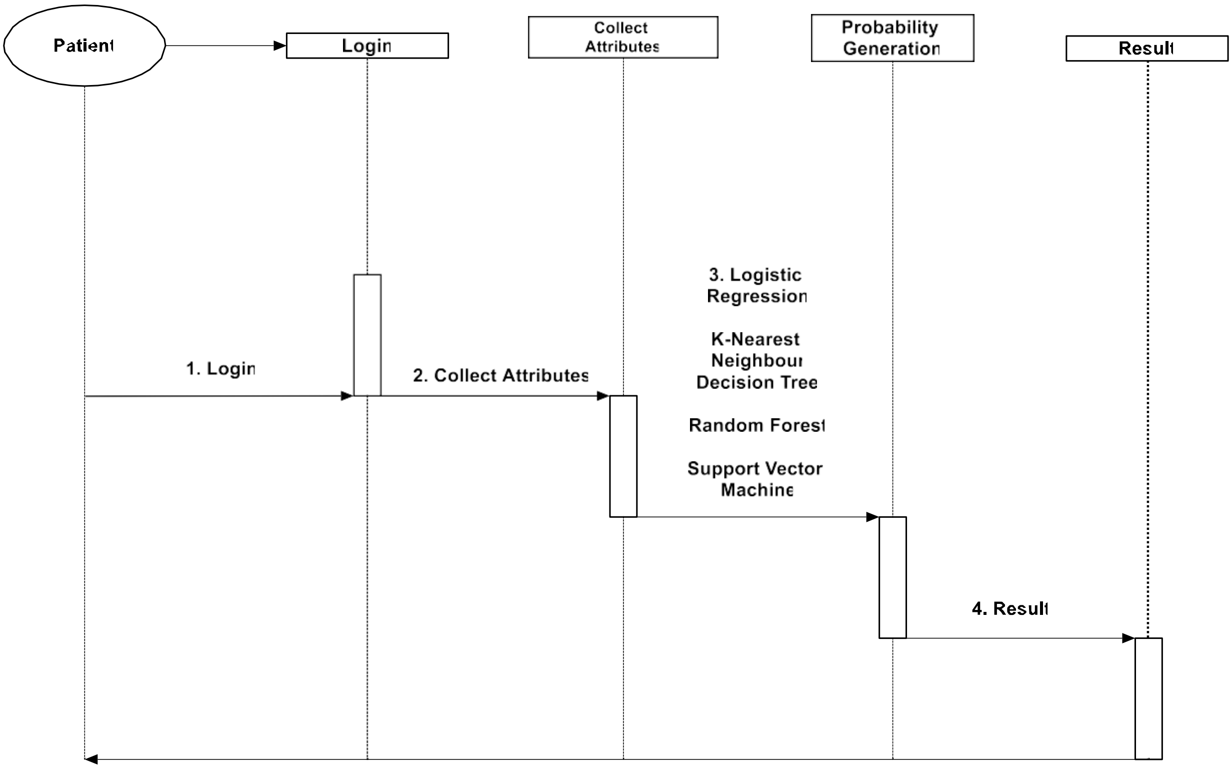
3.5 Use Case Diagram



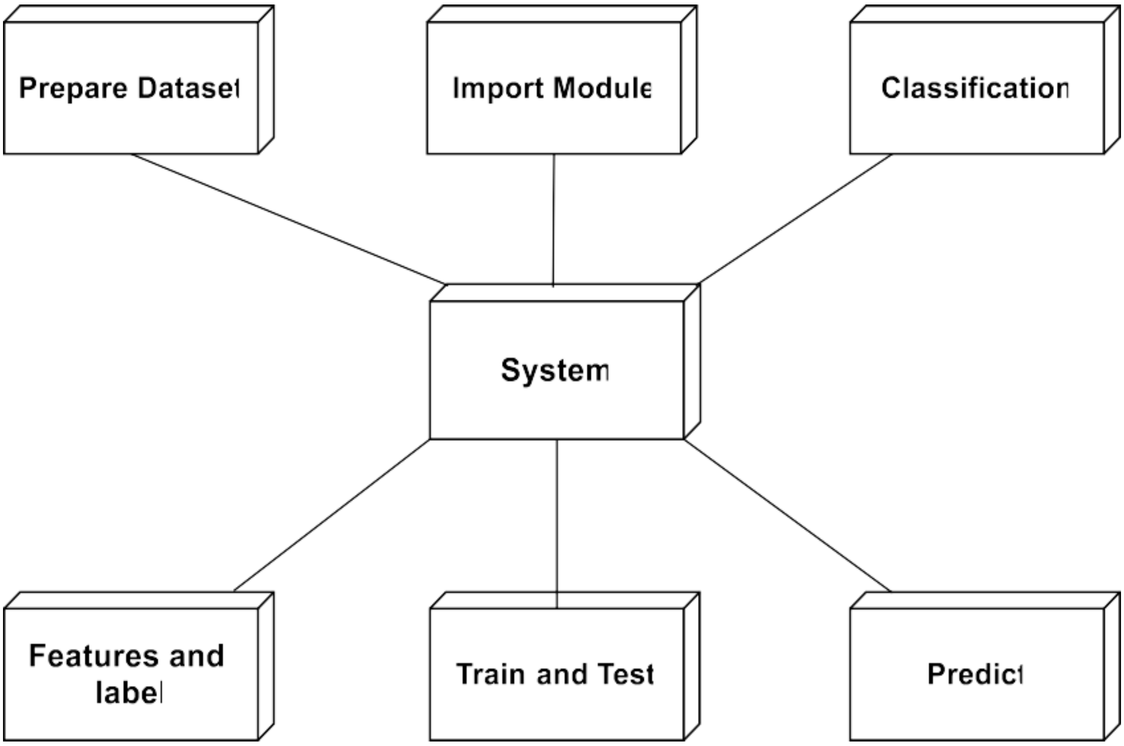
3.6 State Diagram


Incorrect
User Login
Check Login details
Correct
Enter Attributes and feed it into System
Registration of new User
System Processes Data and Gives Report

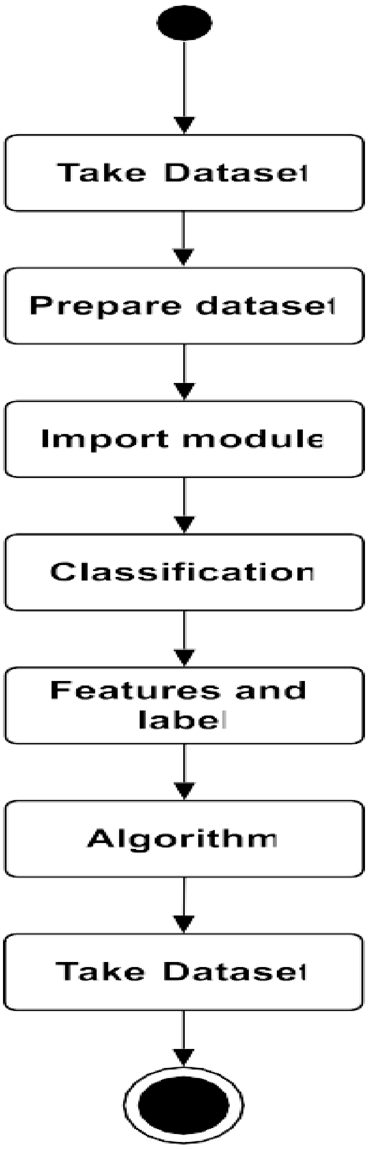
3.7 Sequence Diagram



3.8 Deployment Diagram



3.9 Activity Diagram



Chapter 4
System Design

4.1 Database Design

1. User Registration

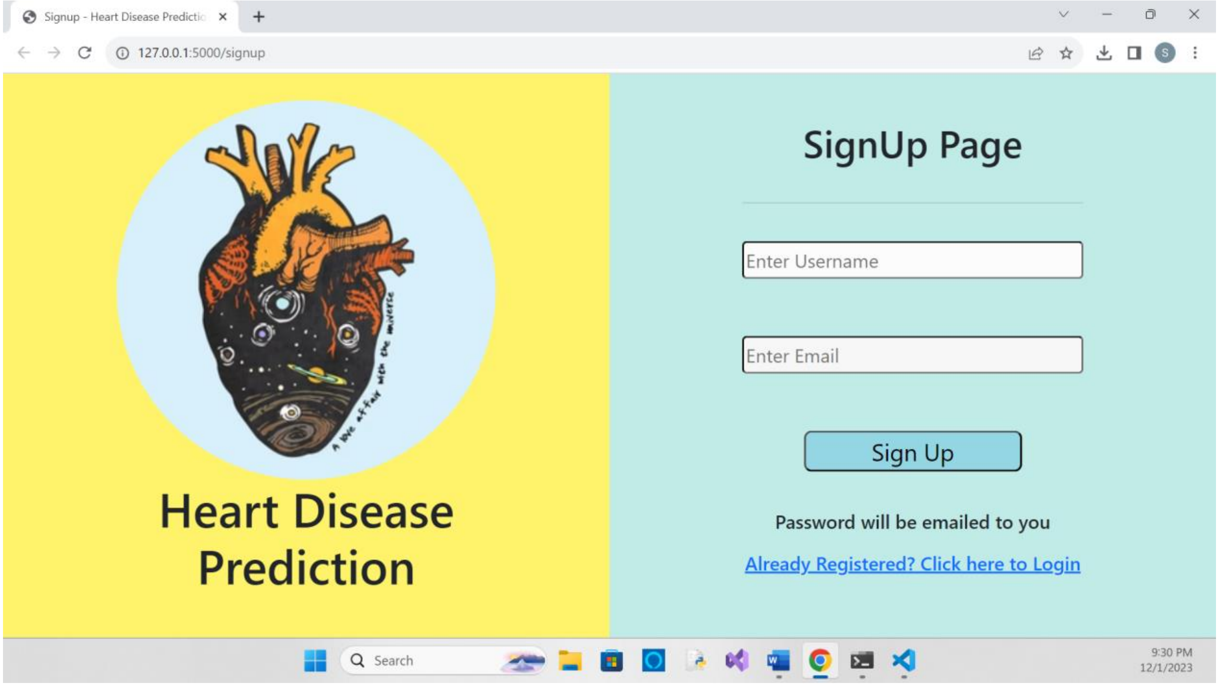
Field	Type	Null	Key	Default	Extra
Username	Varchar(255)	No	Primary Key	Null	
Emailid	Varchar(255)	Yes		Null	
Password	Varchar(255)	Yes		Null	

2. Details

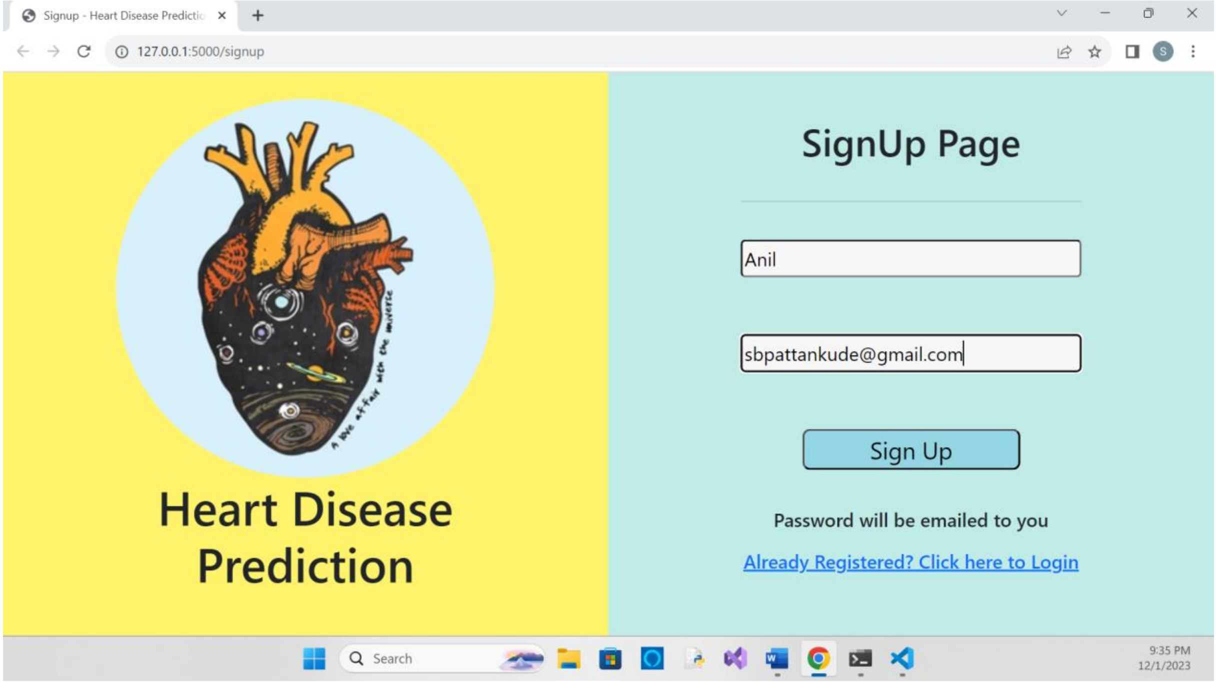
Field	Type	Null	Key	Default	Extra
Did	Int	No	Primary Key	Null	Auto_increment
Age	Int	Yes		Null	
Gender	Int	Yes		Null	
Height	Int	Yes		Null	
Weight	Int	Yes		Null	
BP Lo	Int	Yes		Null	
Bp Hi	Int	Yes		Null	
Cholesterol	Int	Yes		Null	
Heartrate	Int	Yes		Null	
Smoke	Int	Yes		Null	
Alcohol	Int	Yes		Null	
Physical	Int	Yes		Null	
Cardio	Int	Yes		Null	

4.2 Input Design

1. Registration:



Providing the Inputs:

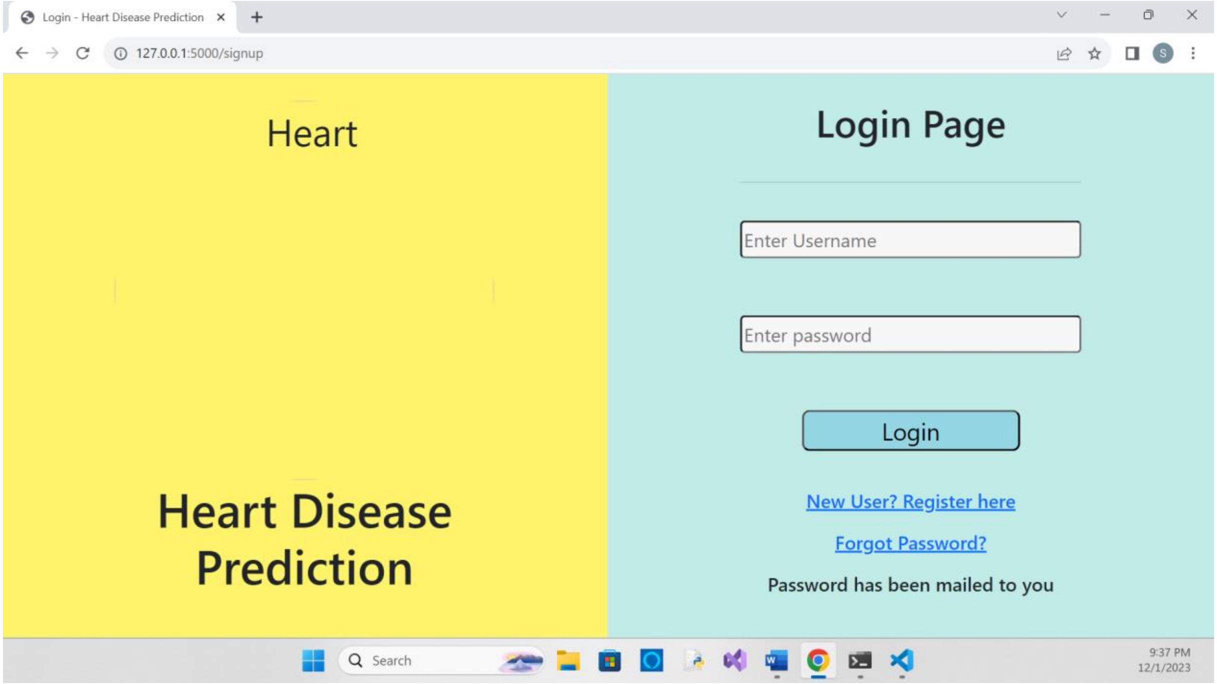

Signup - Heart Disease Predictio
127.0.0.1:5000/signup
Heart Disease Prediction
SignUp Page
Sign Up
Password will be emailed to you
Already Registered? Click here to Login
Search
9:35 PM
12/1/2023

Anil

sbpattankude@gmail.com

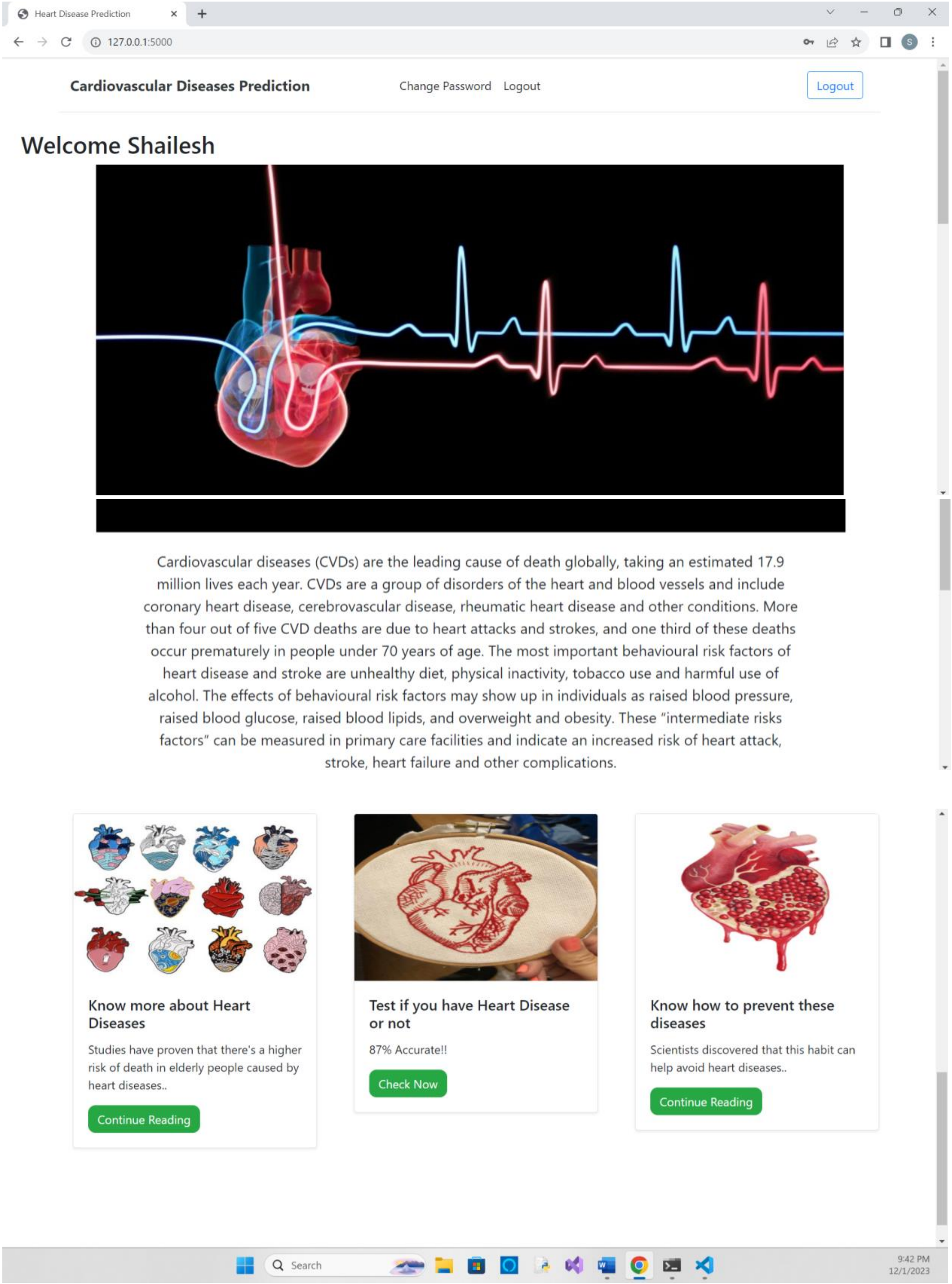
After submitting the input:

(Redirected to Login page as password was mailed on given email id)



Password has been mailed to you

2. Home page:



Cardiovascular Diseases Prediction

Welcome Shailesh

Cardiovascular diseases (CVDs) are the leading cause of death globally, taking an estimated 17.9 million lives each year. CVDs are a group of disorders of the heart and blood vessels and include coronary heart disease, cerebrovascular disease, rheumatic heart disease and other conditions. More than four out of five CVD deaths are due to heart attacks and strokes, and one third of these deaths occur prematurely in people under 70 years of age. The most important behavioural risk factors of heart disease and stroke are unhealthy diet, physical inactivity, tobacco use and harmful use of alcohol. The effects of behavioural risk factors may show up in individuals as raised blood pressure, raised blood glucose, raised blood lipids, and overweight and obesity. These "intermediate risks factors" can be measured in primary care facilities and indicate an increased risk of heart attack, stroke, heart failure and other complications.

Know more about Heart Diseases

Studies have proven that there's a higher risk of death in elderly people caused by heart diseases..

Continue Reading

Test if you have Heart Disease or not

87% Accurate!!

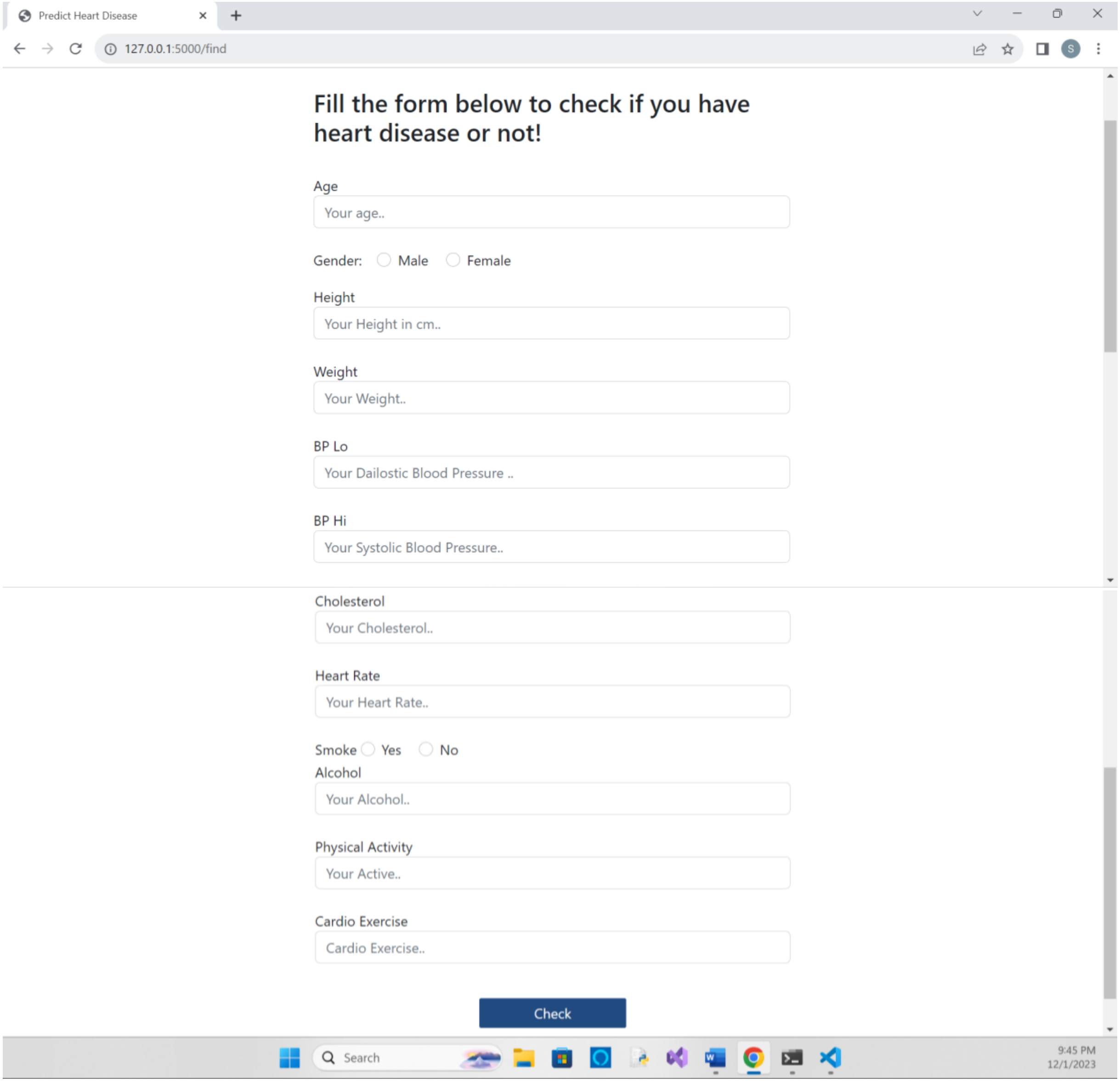
Check Now

Know how to prevent these diseases

Scientists discovered that this habit can help avoid heart diseases..

Continue Reading

3. Prediction Form:



Fill the form below to check if you have heart disease or not!

Age

Your age..

Gender: ○ Male ○ Female

Height

Your Height in cm..

Weight

Your Weight..

BP Lo

Your Dailostic Blood Pressure ..

BP Hi

Your Systolic Blood Pressure..

Cholesterol

Your Cholesterol..

Heart Rate

Your Heart Rate..

Smoke ○ Yes ○ No

Alcohol

Your Alcohol..

Physical Activity

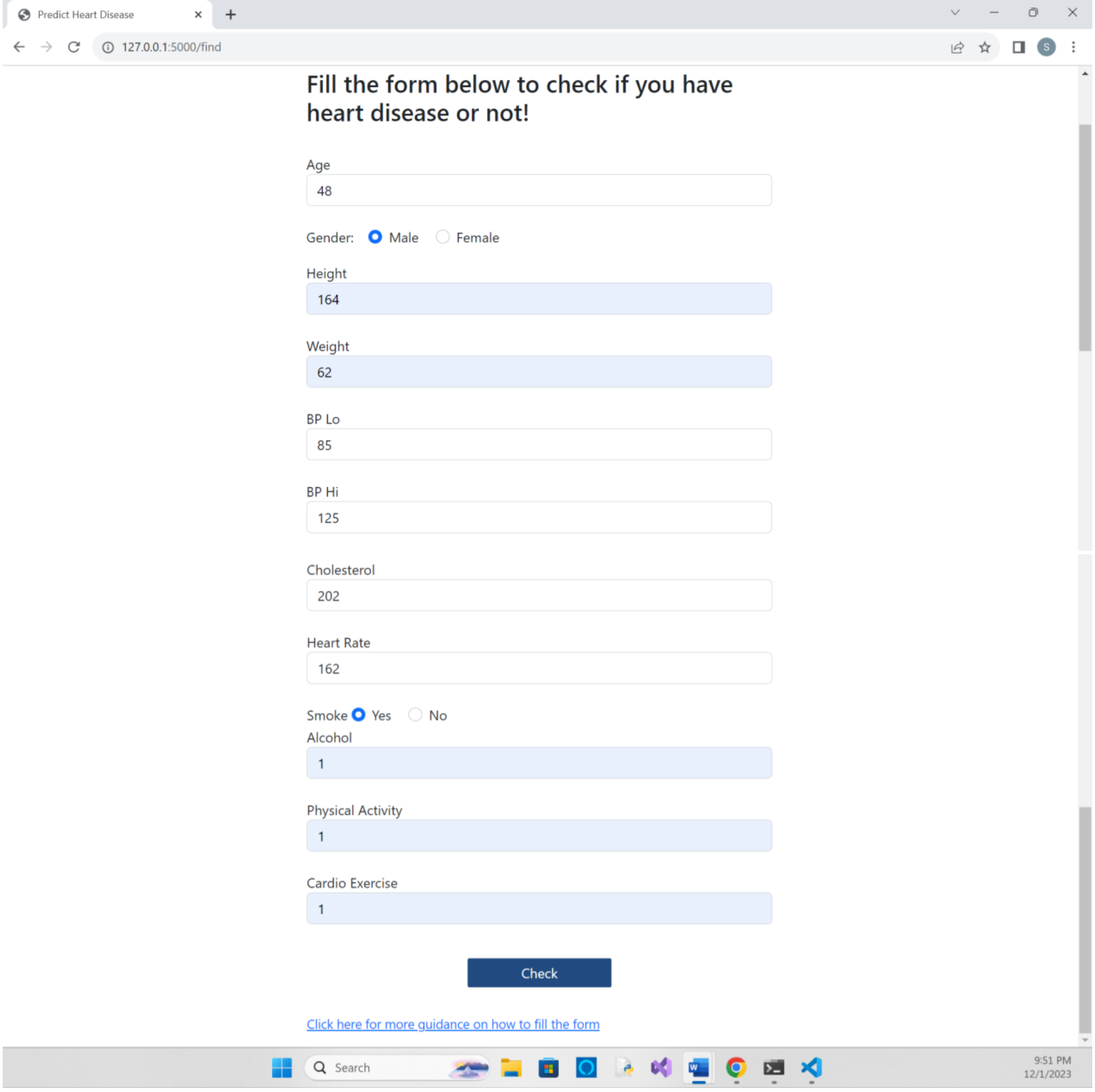
Your Active..

Cardio Exercise

Cardio Exercise..

Check

4. Providing required inputs:
(Input with high-risk inputs)



Fill the form below to check if you have heart disease or not!

Age
48

Gender: ● Male ○ Female

Height
164

Weight
62

BP Lo
85

BP Hi
125

Cholesterol
202

Heart Rate
162

Smoke ● Yes ○ No

Alcohol
1

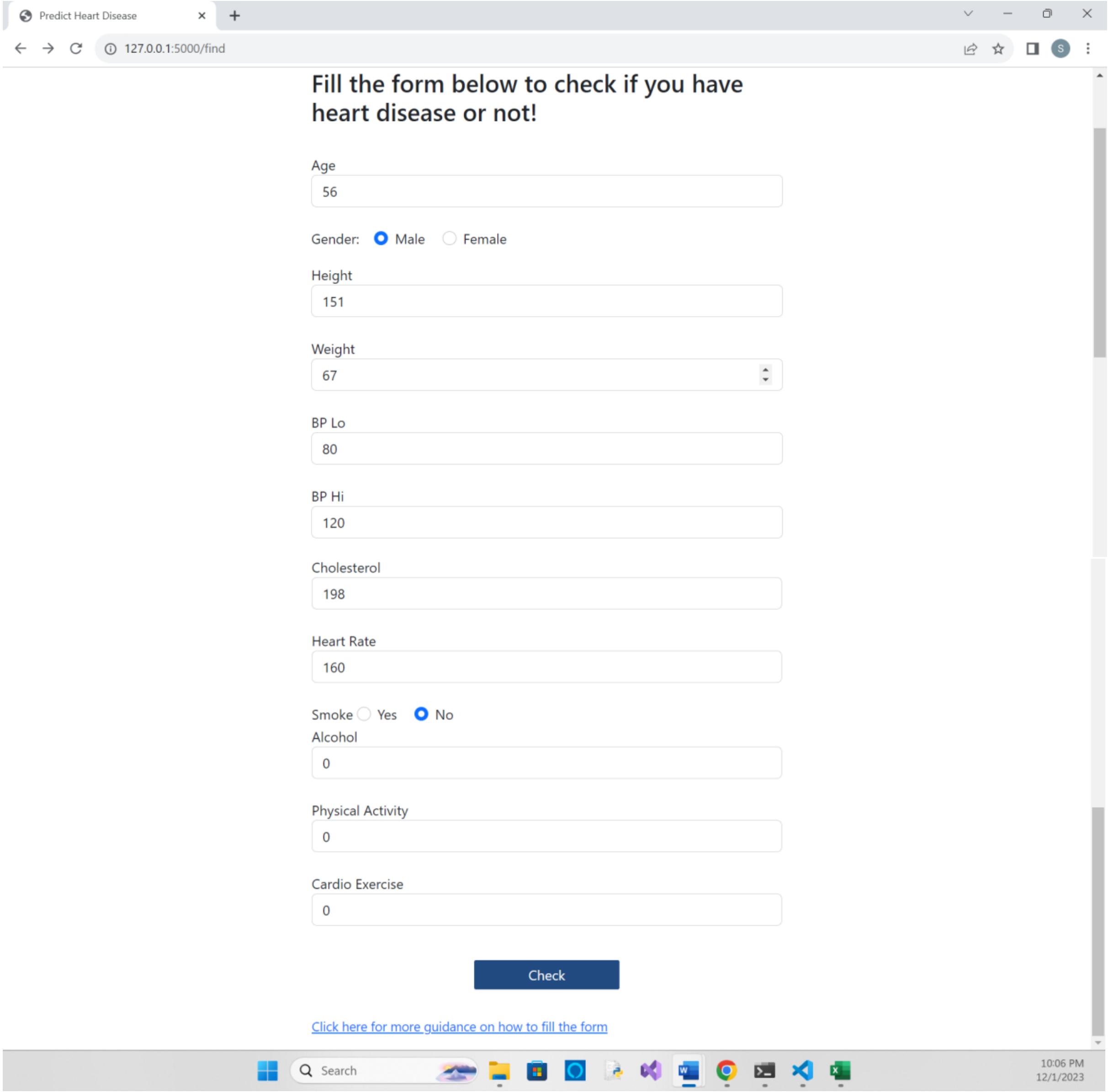
Physical Activity
1

Cardio Exercise
1

Check

Click here for more guidance on how to fill the form

(Input with low-risk input)



Fill the form below to check if you have heart disease or not!

Age
56

Gender: ● Male ○ Female

Height
151

Weight
67

BP Lo
80

BP Hi
120

Cholesterol
198

Heart Rate
160

Smoke ○ Yes ● No

Alcohol
0

Physical Activity
0

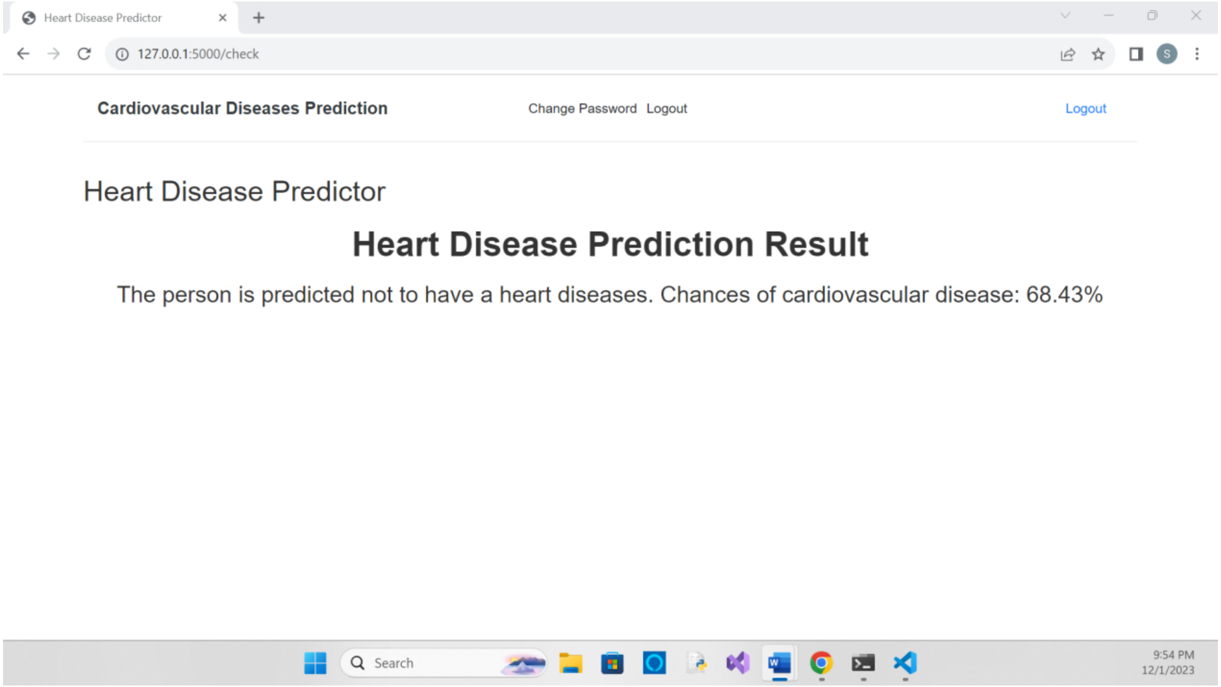
Cardio Exercise
0

Check

Click here for more guidance on how to fill the form

4.3 Outputs

(With high-risk inputs)

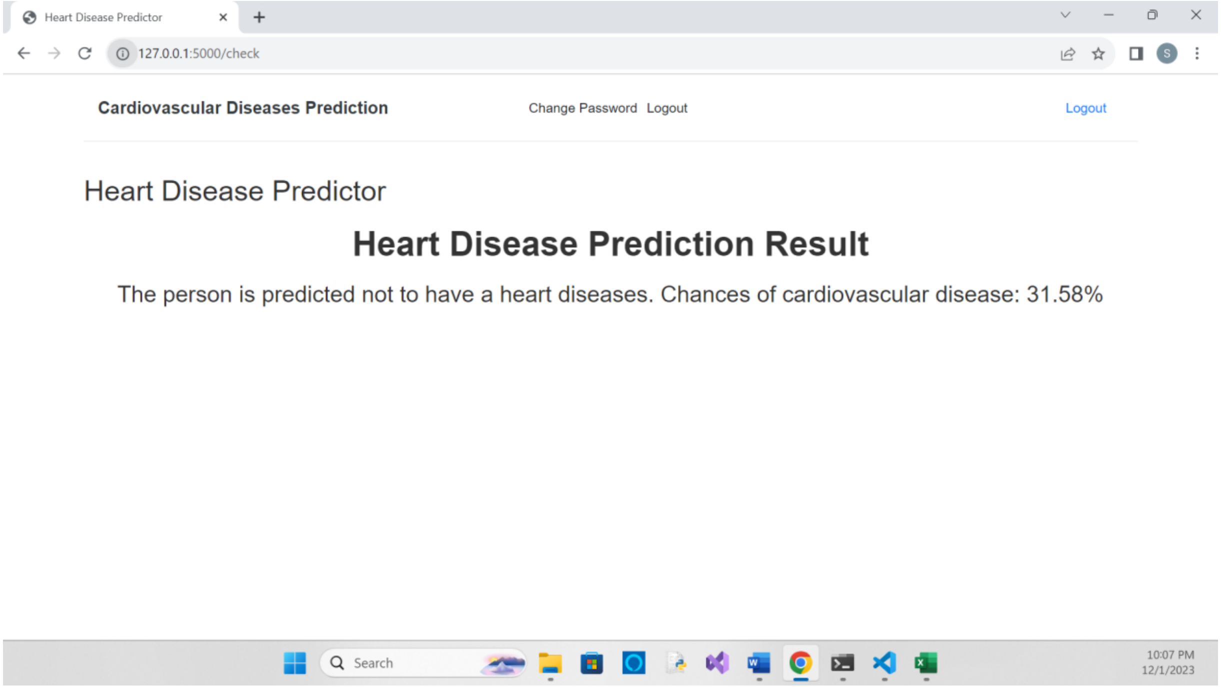


Heart Disease Predictor

Heart Disease Prediction Result

The person is predicted not to have a heart diseases. Chances of cardiovascular disease: 68.43%

(With low-risk inputs)



Heart Disease Predictor

Heart Disease Prediction Result

The person is predicted not to have a heart diseases. Chances of cardiovascular disease: 31.58%

Chapter 5
System Requirements
and
Implementation

5.1 System Requirement

Hardware:

- Graphics Card: - 1650 or 1660TI

  Performs ML or DL training and (often) inference, which is the capacity to automatically classify data based on learning, and is frequently an Nvidia P100 (Pascal), V100 (Volta), or A100 (Ampere) GPU for training and a V100, A100, or T4 (Turing) GPU for inference.

- CPU: - i5 9th or 10th Gen.

  It is accountable for managing I/O, running the VM or container subsystem, and sending code to the GPUs. With the addition of features that considerably speed up ML and DL inference procedures, current-generation CPUs are now ready for production AI workloads using models that were previously trained on GPUs.

- Storage IOPS

  Another performance barrier for AI workloads is the transfer of data between the storage and compute subsystems. As a result, local NVMe drives are more common than SATA SSDs in systems.

- RAM: - 16GB
- Intel's ML Hardware Evolution
- PC

Software:

- Windows 10 and 11 (Intel/AMD 64-bit)
- Google Collaboratory
- Visual Studio Code
- Microsoft Excel
- Python IDLE

5.2 Implementation

Phase I

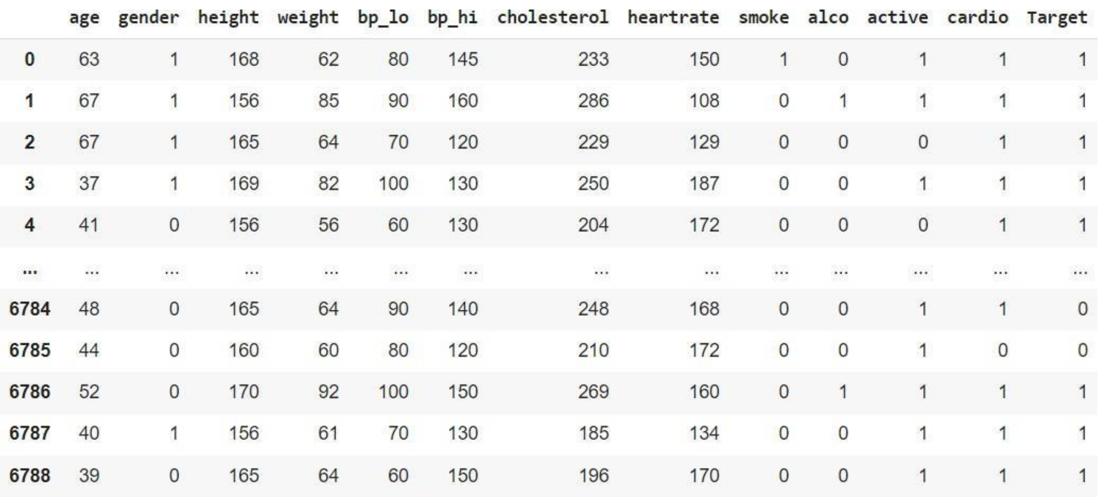
1. Dataset Study:

In order to train and test the model we have used a dataset comprising of almost 70k attributes. In the final project, the patients will have to provide data as per their test reports.

The dataset comprises of the following columns –

- Objective: factual information.
- Examination: results of medical examination.
- Subjective: information given by the patient.

Dataset:



	age	gender	height	weight	bp_lo	bp_hi	cholesterol	heartrate	smoke	alco	active	cardio	Target
0	63	1	168	62	80	145	233	150	1	0	1	1	1
1	67	1	156	85	90	160	286	108	0	1	1	1	1
2	67	1	165	64	70	120	229	129	0	0	0	1	1
3	37	1	169	82	100	130	250	187	0	0	1	1	1
4	41	0	156	56	60	130	204	172	0	0	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6784	48	0	165	64	90	140	248	168	0	0	1	1	0
6785	44	0	160	60	80	120	210	172	0	0	1	0	0
6786	52	0	170	92	100	150	269	160	0	1	1	1	1
6787	40	1	156	61	70	130	185	134	0	0	1	1	1
6788	39	0	165	64	60	150	196	170	0	0	1	1	1

Attributes:

Age	Objective	Age	Int (days)
Gender	Objective	Gender	Categorical code
Height	Objective	Height	Int (cm)
Weight	Objective	Weight	Float (kg)
Diastolic blood pressure	Examination	Bp_lo	Int
Systolic blood pressure	Examination	Bp_hi	Int
Cholesterol	Examination	Cholesterol	Int
Heartrate	Examination	Heartrate	Int
Smoking	Subjective	Smoke	Binary
Alcohol intake	Subjective	Alco	Binary
Physically Active	Subjective	Active	Binary
Cardio Exercise	Subjective	Cardio	Binary
Presence or absence of CVD	Target variable	Target	Binary

After taking input from the user, we are going to pass it to the model that we have trained in order to receive accurate results regarding presence or absence of the heart disease.

2. Performing EDA (exploratory data analysis)

The dataset requires cleaning and modifying to facilitate easy access of data. For that purpose, we will perform EDA on null values and categorial values in the dataset.

3. Feature selection

Two of the 13 features in the data set—one each for age and sex—are used to identify the patient's personal information. The 11 remaining qualities are significant because they include crucial clinical records. Clinical data are essential for heart disease diagnosis and severity assessment. We chose blood pressure to be our target value based on whom we will carry out our analysis. We are also going to consider cholesterol and other parameters for more accurate results.

4. Splitting data into train and test

The project considered two main ways of data splitting one being using sklearn library and another approach is using cross validation or k-fold cross validation. The Sklearn train_test_split function helps us create our training data and test data. Whereas on the other hand, Cross-Validation or K-Fold Cross-Validation is a more robust technique for data splitting, where a model is trained and evaluated "K" times on different samples.

The value of k may be set as per the programmer's choice.

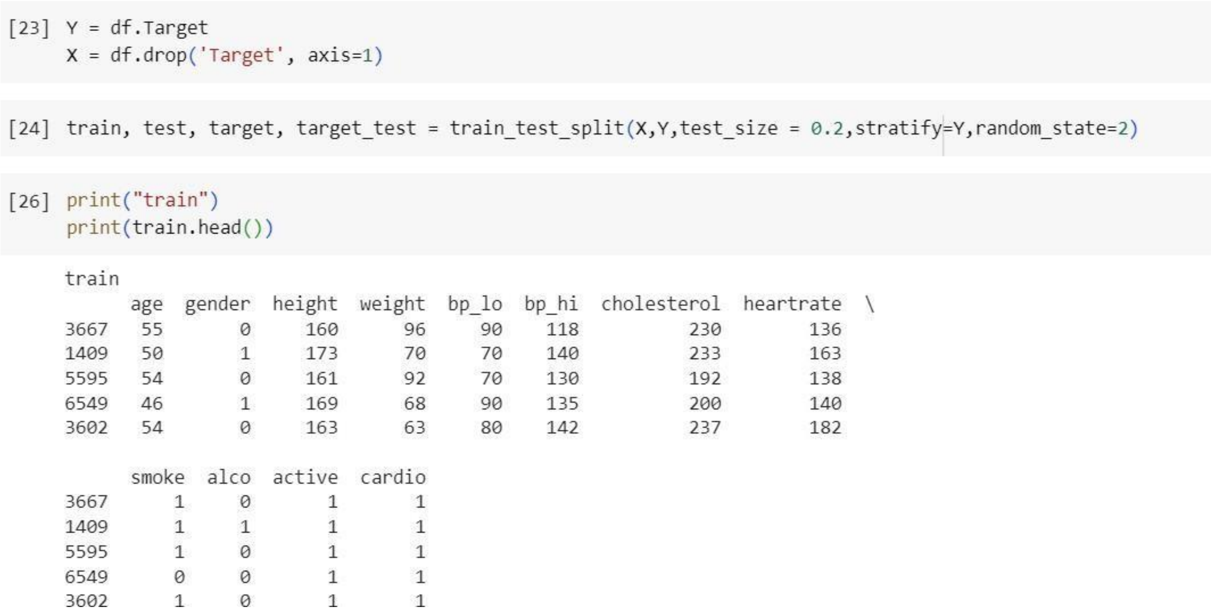
The project mainly implements K-fold cross validation, the example for understanding which is as follows –

Suppose we have a balanced, 2-class dataset consisting of 1000 images of raccoons and ringtails (to be used for training and validation only). Now, we want to perform a 5-Fold cross-validation. We first split the datasets into 5 equal and non-overlapping parts: each consisting of 200 images; label them as Parts 1, 2, 3, 4, and 5. Each of these subsets of 200 images consists of mutually different samples.

Now, we will create 5 complete datasets (labeled as Datasets 1-5) using Parts 1-5 in the following manner: For Dataset-1, use Part-1 as the validation set, and consolidate Parts 2-5 to create the training set; for Dataset-2, use Part-2 as the validation set, and consolidate Parts 1, 3, 4 and 5 to create the training set, and so on. Notice that since each part consists of 20% of the data of the original dataset, each of Datasets 1-5 has an 80%-20% train-validation split ratio. Generalizing, each K-Fold cross-validation dataset has (100/K) % data in its validation set (here, 100/5 = 20% was in validation set).

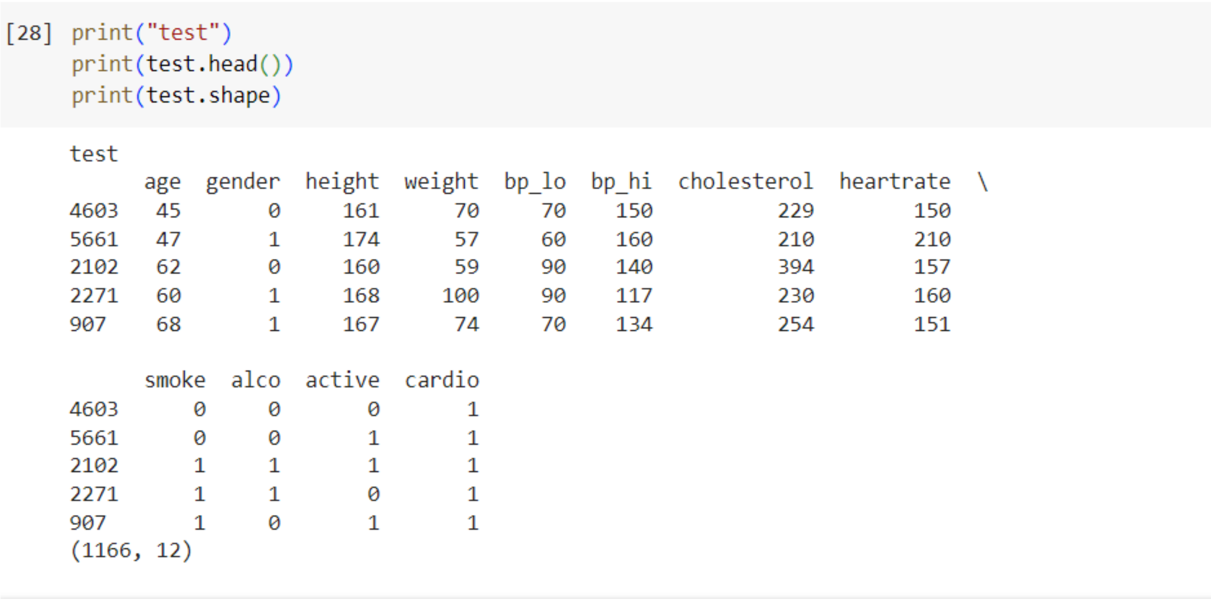
The images of the trained and tested dataset are as below –

Train –



```
[23] Y = df.Target
     X = df.drop('Target', axis=1)

[24] train, test, target, target_test = train_test_split(X,Y,test_size = 0.2,stratify=Y,random_state=2)

[26] print("train")
     print(train.head())

     train
           age  gender  height  weight  bp_lo  bp_hi  cholesterol  heartrate  \
     3667   55       0     160      96     90    118          230        136
     1409   50       1     173      70     70    140          233        163
     5595   54       0     161      92     70    130          192        138
     6549   46       1     169      68     90    135          200        140
     3602   54       0     163      63     80    142          237        182

           smoke  alco  active  cardio
     3667      1     0       1       1
     1409      1     1       1       1
     5595      1     0       1       1
     6549      0     0       1       1
     3602      1     0       1       1
```

Test –



```
[28] print("test")
     print(test.head())
     print(test.shape)

     test
           age  gender  height  weight  bp_lo  bp_hi  cholesterol  heartrate  \
     4603   45       0     161      70     70    150          229        150
     5661   47       1     174      57     60    160          210        210
     2102   62       0     160      59     90    140          394        157
     2271   60       1     168     100     90    117          230        160
     907    68       1     167      74     70    134          254        151

           smoke  alco  active  cardio
     4603      0     0       0       1
     5661      0     0       1       1
     2102      1     1       1       1
     2271      1     1       0       1
     907       1     0       1       1
     (1166, 12)
```

We have employed the sklearn library for splitting the dataset and the train dataset is 80% of the total while test is the remaining 20%.

5. Evaluate and improve model accuracy

Accuracy is a measure to know how well or badly a model is doing on an unseen validation set. Based on the current learning, evaluate the model on validation sets. We train and test the model with the help of the dataset that we have split. For that we have used algorithms like –

1 Logistic regression
2 K nearest neighbor
3 Random Forest
4 Neural Network
5 Perceptron
6 MLP
7 Decision Tree Classifier
8 SVM
9 Naïve Bayes

Phase II

Front-end Part

1. HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as

<p> and </p> surround and provide information about document text and may include sub-element tags. Browsers do not display the HTML tags but use them to interpret the content of the page.

2. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

3. Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is an HTML, CSS and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components which do not require other libraries like jQuery. They provide additional user interface elements such as dialog boxes, tooltips, progress bars, navigation drop-downs, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

4. JavaScript

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

5. Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

6. Prediction

Test the model on unknown data or real-time data, After the system starts working properly, the model is complete.

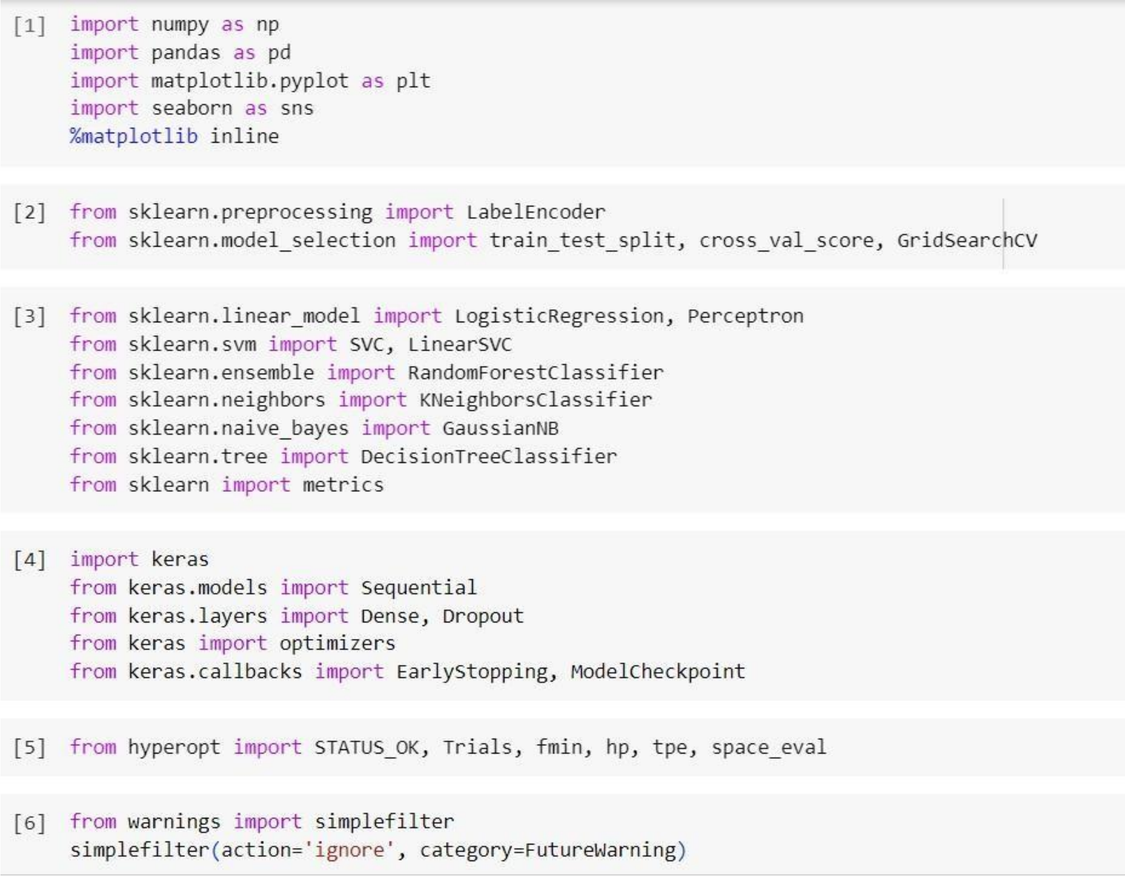
Back-end Part (Model Part)

For the backend, that is for the actual cardiovascular disease prediction system, we developed a model where we have developed a model. The model is a classification model which will predict whether or not a person has heart disease.

However, this is not the objective of our major project. We wanted to develop a model which can also show the risk percentage along with the possibility of having a heart disease.

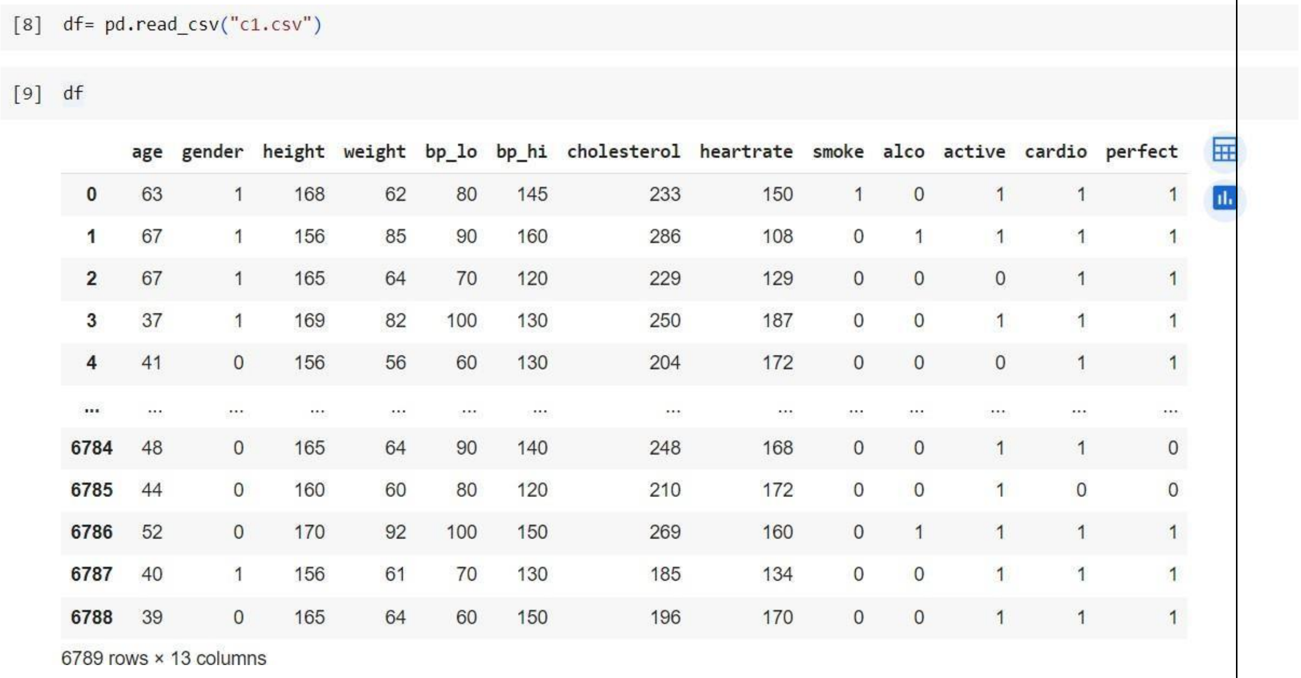
For that purpose, we have developed a full stack project, and have coded the risk prediction part separately which will be explained further.

Libraries used:



```
[1]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     %matplotlib inline

[2]  from sklearn.preprocessing import LabelEncoder
     from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV

[3]  from sklearn.linear_model import LogisticRegression, Perceptron
     from sklearn.svm import SVC, LinearSVC
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.naive_bayes import GaussianNB
     from sklearn.tree import DecisionTreeClassifier
     from sklearn import metrics

[4]  import keras
     from keras.models import Sequential
     from keras.layers import Dense, Dropout
     from keras import optimizers
     from keras.callbacks import EarlyStopping, ModelCheckpoint

[5]  from hyperopt import STATUS_OK, Trials, fmin, hp, tpe, space_eval

[6]  from warnings import simplefilter
     simplefilter(action='ignore', category=FutureWarning)
```

Using the pandas library, we loaded the dataset which is as follows –



```
[8]  df= pd.read_csv("c1.csv")
```

```
[9]  df
```

	age	gender	height	weight	bp_lo	bp_hi	cholesterol	heartrate	smoke	alco	active	cardio	perfect
0	63	1	168	62	80	145	233	150	1	0	1	1	1
1	67	1	156	85	90	160	286	108	0	1	1	1	1
2	67	1	165	64	70	120	229	129	0	0	0	1	1
3	37	1	169	82	100	130	250	187	0	0	1	1	1
4	41	0	156	56	60	130	204	172	0	0	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6784	48	0	165	64	90	140	248	168	0	0	1	1	0
6785	44	0	160	60	80	120	210	172	0	0	1	0	0
6786	52	0	170	92	100	150	269	160	0	1	1	1	1
6787	40	1	156	61	70	130	185	134	0	0	1	1	1
6788	39	0	165	64	60	150	196	170	0	0	1	1	1

6789 rows × 13 columns

Chapter 6
Reports

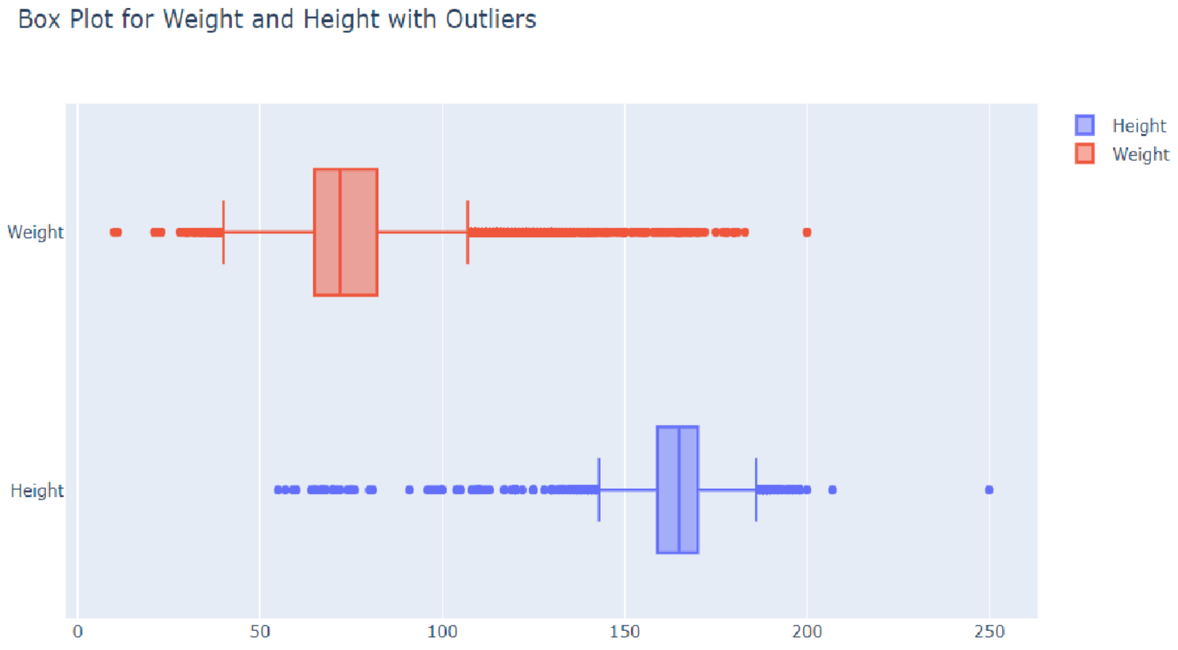
I carried out some study of the dataset to understand the data better. I have plotted some graphs to study the same. The graphs and the codes for them are as follows.

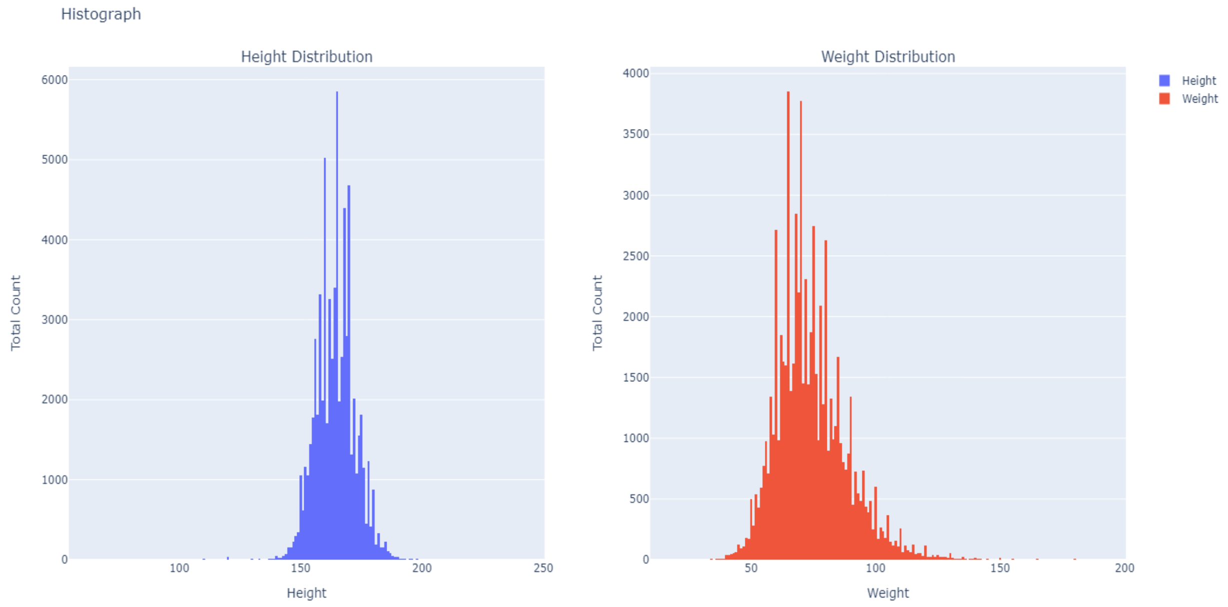
Graphs related to height and weight

i. For detecting outliers in height and weight

Code

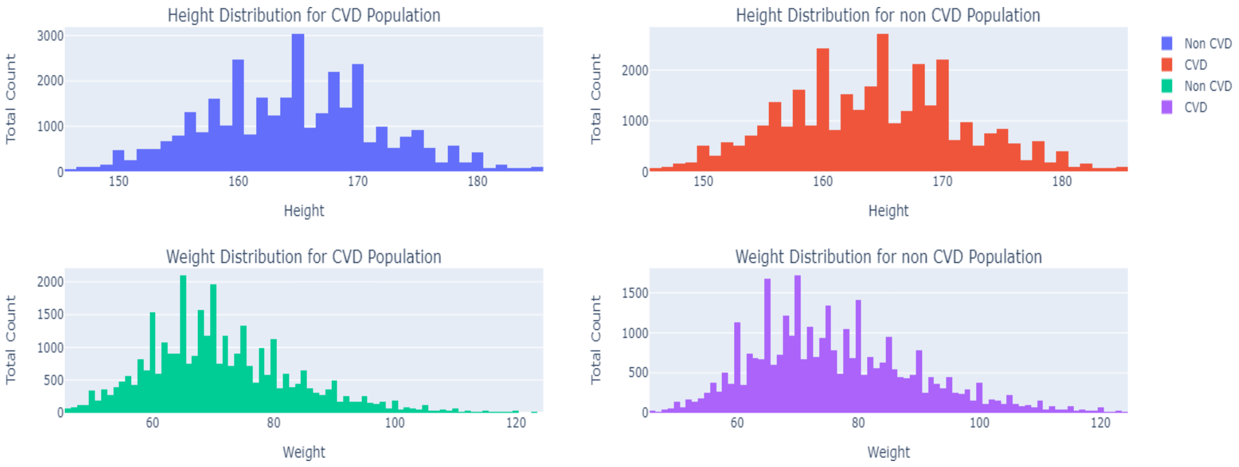
```
def outliers(df_out, drop = False):
    for each_feature in df_out.columns:
        feature_data = df_out[each_feature]
        Q1 = np.percentile(feature_data, 25.) # 25th percentile of the data of the given feature
        Q3 = np.percentile(feature_data, 75.) # 75th percentile of the data of the given feature
        IQR = Q3-Q1 #Interquartile Range
        outlier_step = IQR * 1.5 #That's we were talking about above
        outliers = feature_data[~((feature_data >= Q1 - outlier_step) &
            (feature_data <= Q3 + outlier_step))].index.tolist()
            print('For the feature {}, No of Outliers is {}'.format(each_feature, len(outliers)))
outliers(df[['height', 'weight']])
```


Box Plot for Weight and Height with Outliers
Height
Weight
Weight
Height
0
50
100
150
200
250


Histograph
Height Distribution
Weight Distribution
Height
Weight
Total Count
6000
5000
4000
3000
2000
1000
0
100
150
200
250
Height
Total Count
4000
3500
3000
2500
2000
1500
1000
500
0
50
100
150
200
Weight

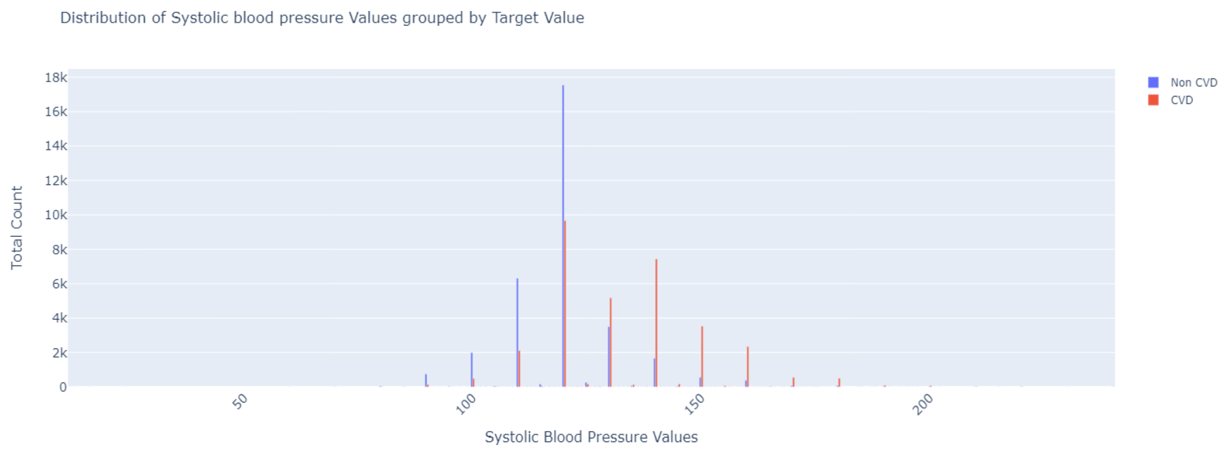
ii. Graph for distribution of height and weight of a person with cardiovascular disease and without cardiovascular disease

```
fig = make_subplots(rows=2, cols=2, subplot_titles=("Height Distribution for CVD
Population", "Height Distribution for non CVD Population", "Weight Distribution for
CVD Population", "Weight Distribution for non CVD Population"))

trace0 = go.Histogram(x=np.exp(df[df['cardio'] == 0]['height']), name = 'NonCVD')
trace1 = go.Histogram(x=np.exp(df[df['cardio'] == 1]['height']), name = 'CVD')

trace2 = go.Histogram(x=np.exp(df[df['cardio'] == 0]['weight']), name = 'NonCVD')
trace3 = go.Histogram(x=np.exp(df[df['cardio'] == 1]['weight']), name = 'CVD')
fig.append_trace(trace0, 1, 1)

fig.append_trace(trace1, 1, 2)

fig.append_trace(trace2, 2, 1)

fig.append_trace(trace3, 2, 3)

fig.update_xaxes(title_text="Height",  row=1,  col=1)
fig.update_yaxes(title_text="Total Count", row=1, col=1)

fig.update_xaxes(title_text="Height",  row=1,  col=2)
fig.update_yaxes(title_text="Total Count", row=1, col=2)

fig.update_xaxes(title_text="Weight",  row=2,  col=1)
fig.update_yaxes(title_text="Total Count", row=2, col=1)

fig.update_xaxes(title_text="Weight",  row=2,  col=2)
fig.update_yaxes(title_text="Total Count", row=2, col=2)

fig.show()
```

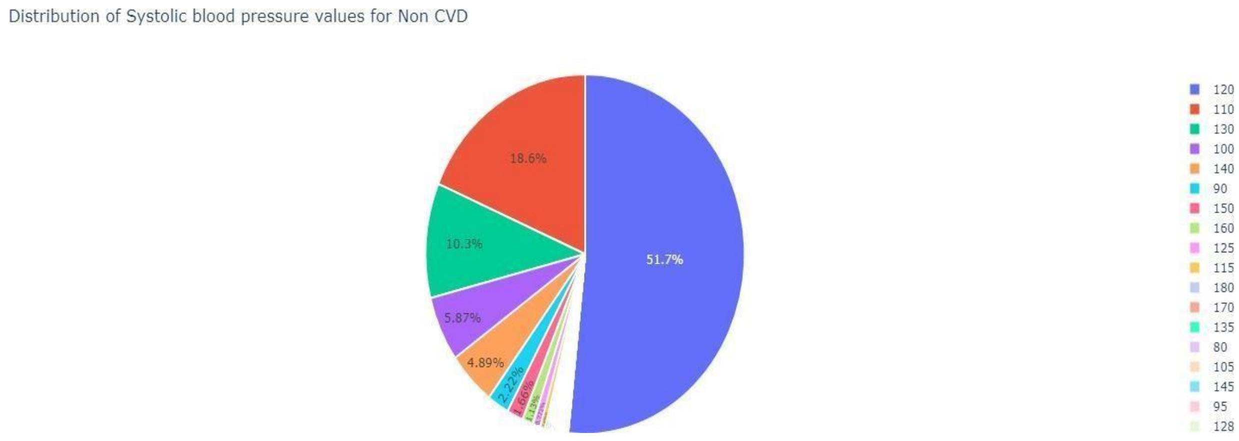


Graphs related to blood pressure

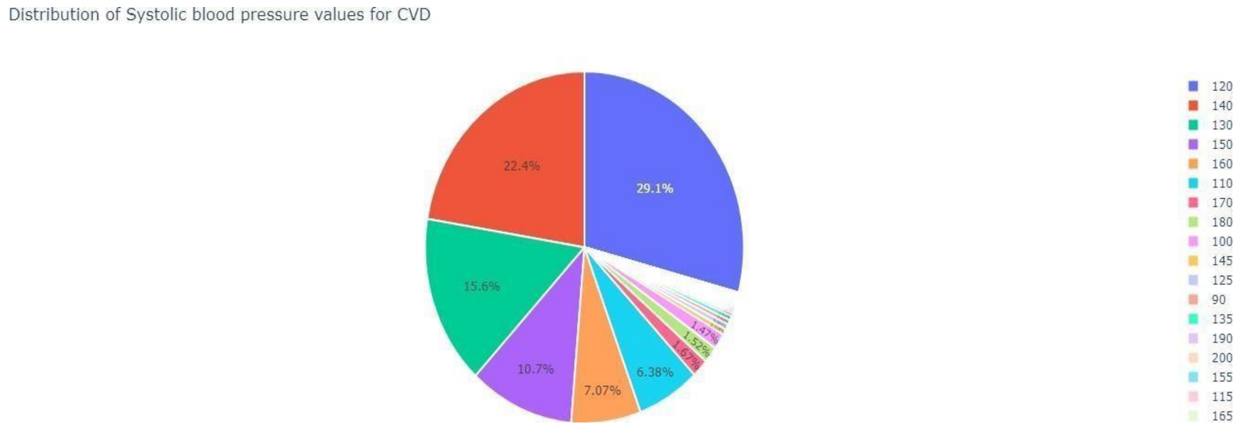
iii. Graph to understand the distribution of people with and without cardiovascular disease against their systolic and diastolic blood pressures



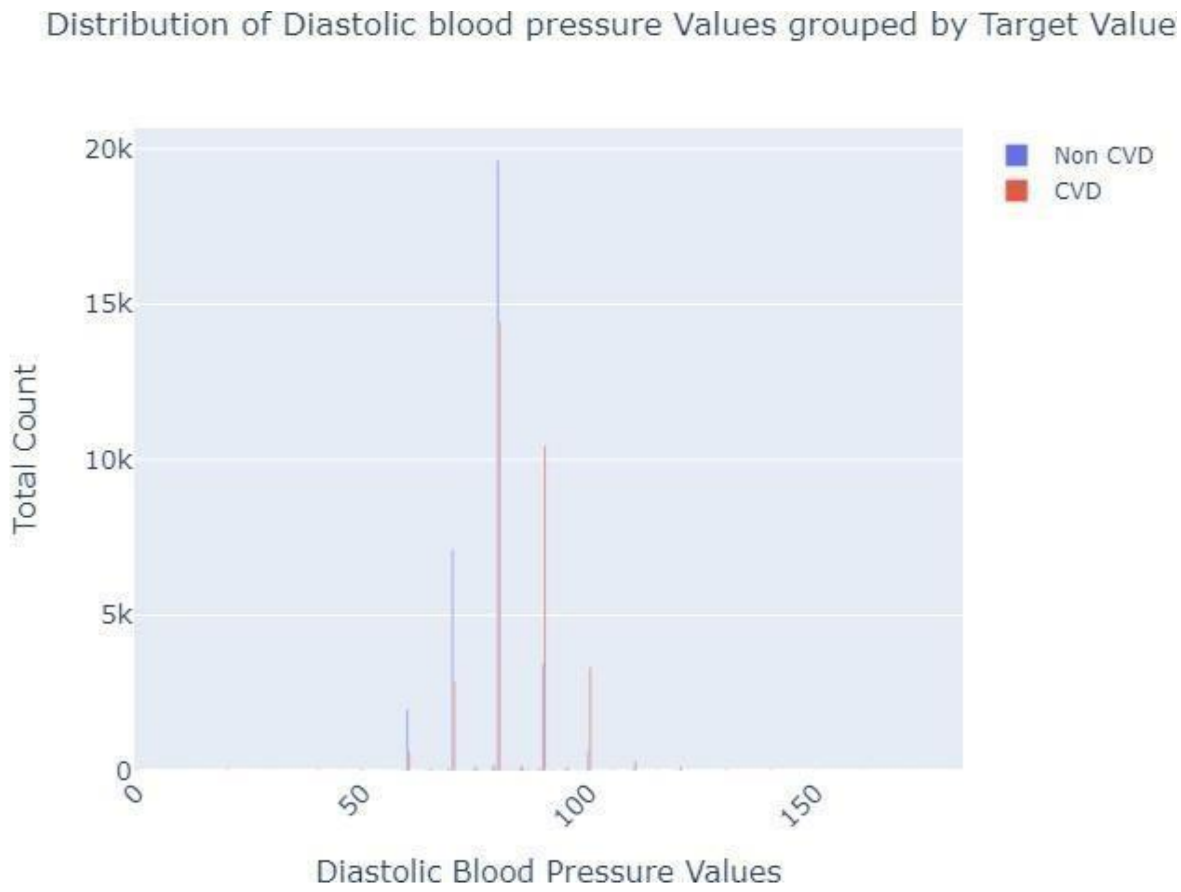
iv. Graph of distribution of systolic blood pressure for people without cardiovascular disease



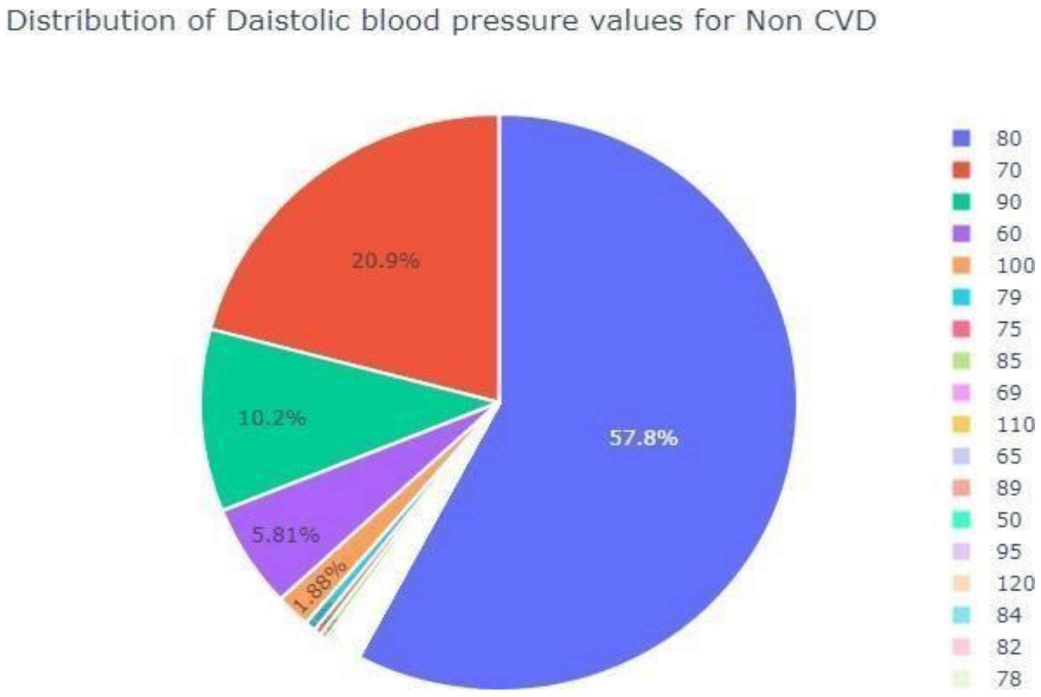
v. **Graph of distribution of diastolic blood pressure for people with cardiovascular disease**



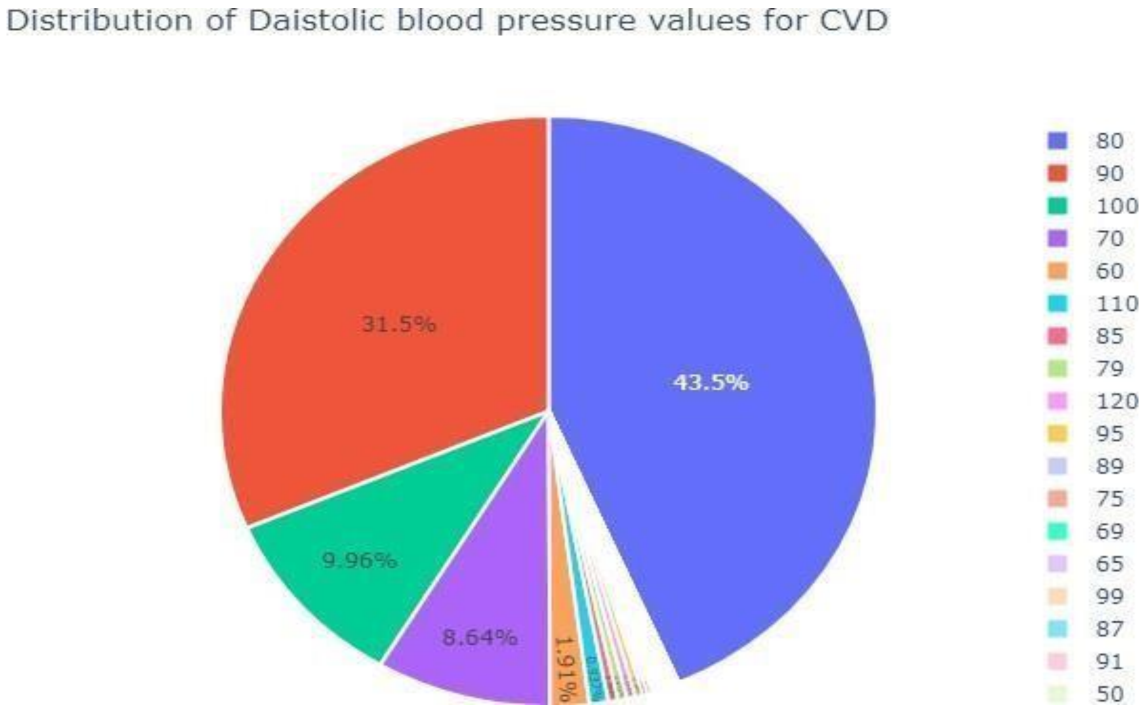
vi. **Graph of count of people to understand whether or not they have cardiovascular disease with the normal values – (diastolic)**



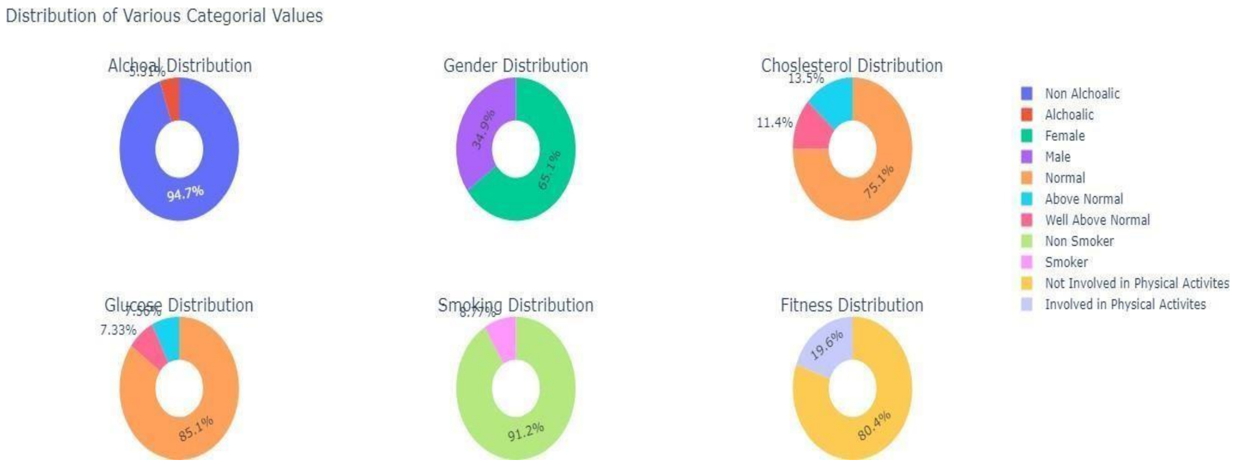
vii. **Graph to understand the diastolic blood pressure ranges for people that do not have cardiovascular disease – (diastolic)**



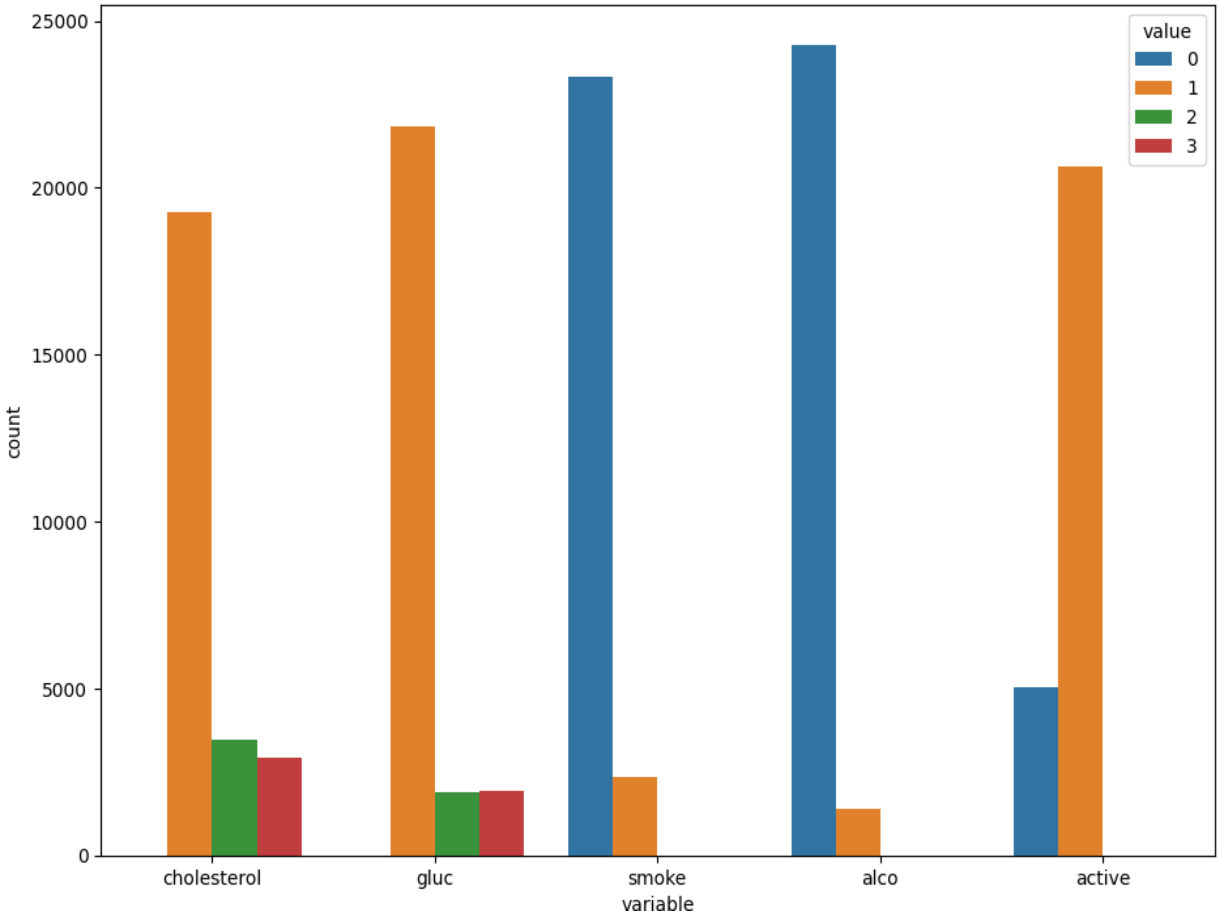
viii. **Graph to understand the diastolic blood pressure ranges for people that have cardiovascular disease – (diastolic)**



ix. Graph to show the distribution of all the categorical values in the dataset considered

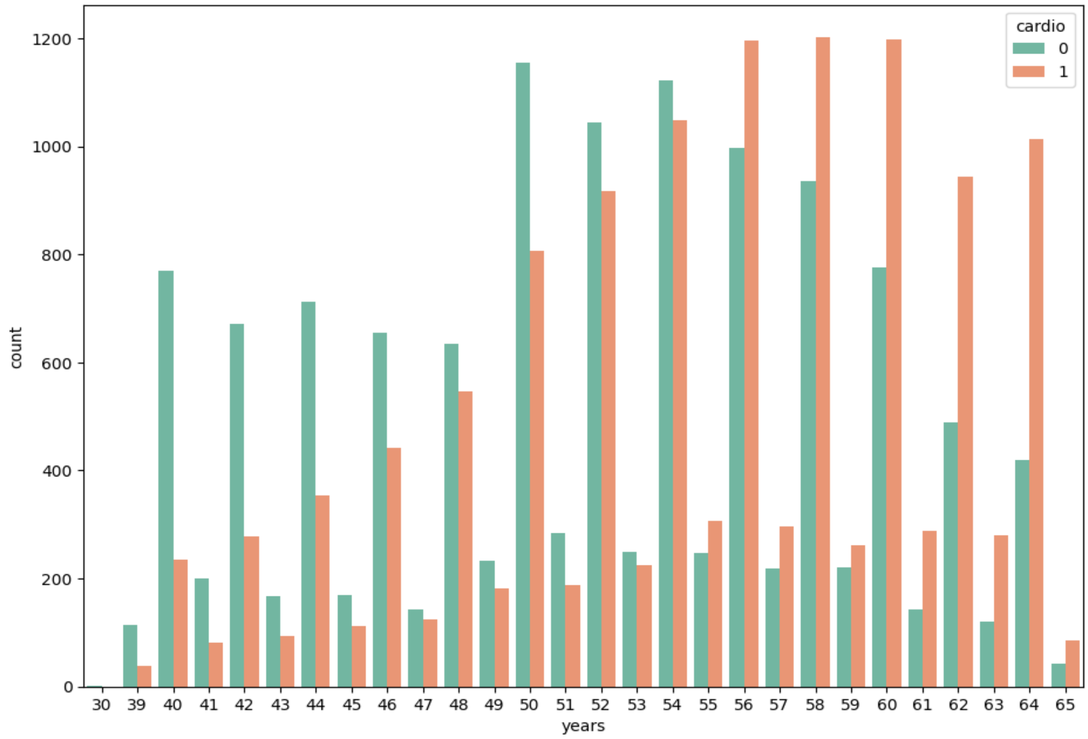


x. Graph of severity level of cholesterol, glucose, smoke, alcohol intake and physical activity with the count of people in that severity range

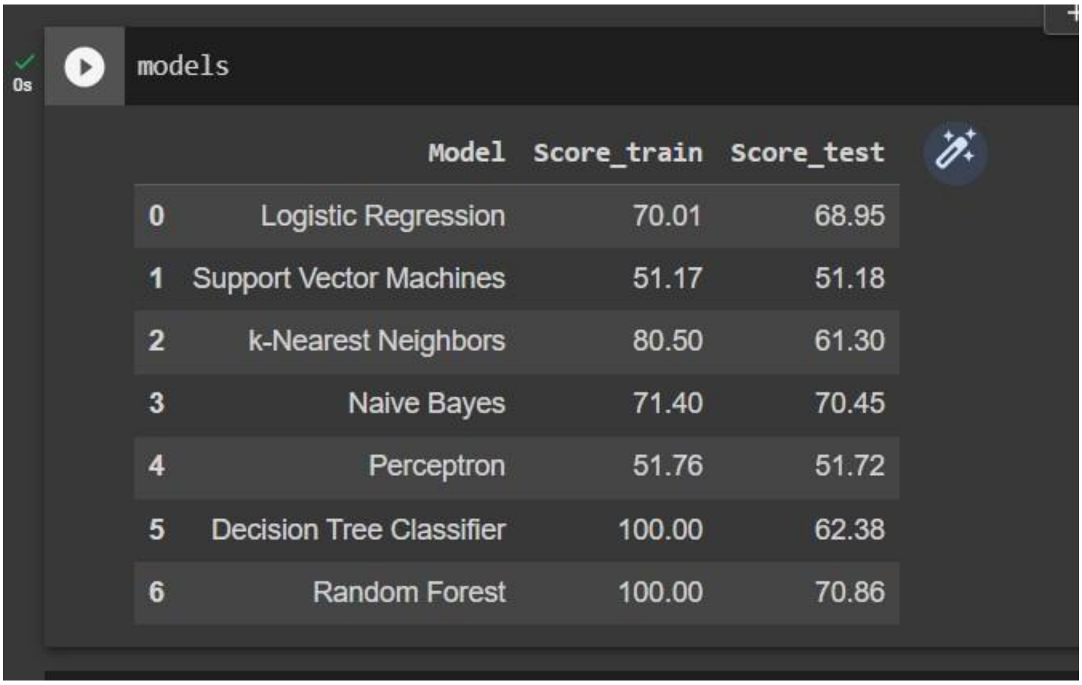


Graphs related to age factor

xi. Graph to study the age group and the gender which mainly suffer from cardiovascular disease.



Accuracy of all the models can be understood through the following –



```
models
```

	Model	Score_train	Score_test
0	Logistic Regression	70.01	68.95
1	Support Vector Machines	51.17	51.18
2	k-Nearest Neighbors	80.50	61.30
3	Naive Bayes	71.40	70.45
4	Perceptron	51.76	51.72
5	Decision Tree Classifier	100.00	62.38
6	Random Forest	100.00	70.86

Line Graph:

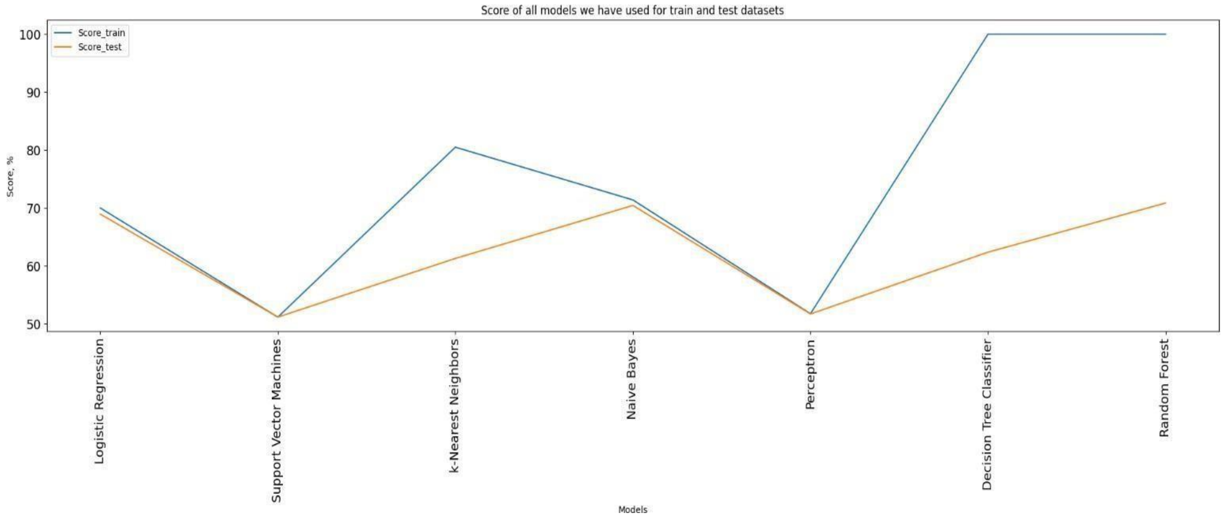


Fig. line graph

Bar Graph:

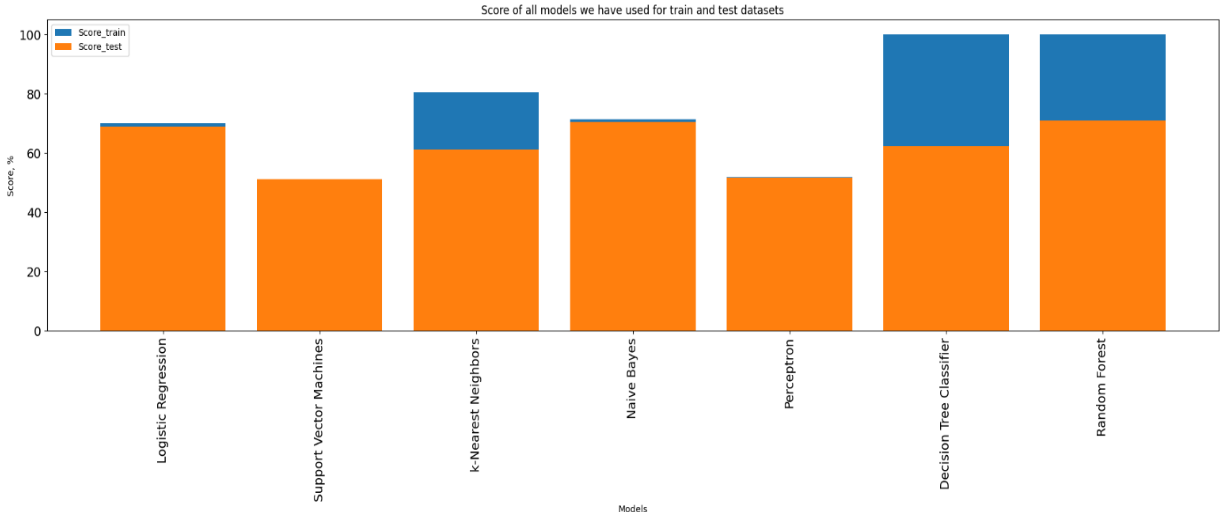


Fig. bar graph

We have finally selected Random Forest to be the model which we will use to make predictions because it gave us the highest accuracy –

For training we received – 100 %

For test we received – 70.86 %

The graph of random forest is as follows –

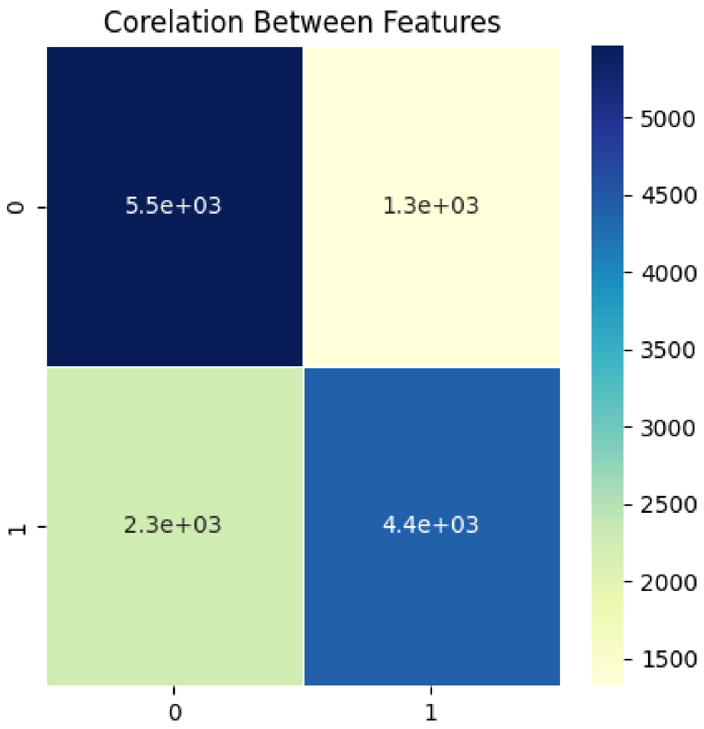


Fig. confusion matrix of random forest model

Chapter 7
Conclusion
And
Future Work

Identifying the processing of raw healthcare data of heart information will help in the long-term saving of human lives and early detection of abnormalities in heart conditions. Machine learning techniques were used in this work to process raw data and provide a new and novel discernment towards heart disease. Heart disease prediction is challenging and very important in the medical field. However, the mortality rate can be drastically controlled if the disease is detected at the early stages and preventative measures are adopted as soon as possible.

Through this project we have made an honest effort to use various machine learning algorithms in order to get maximum accuracy which is very important in healthcare sector.

There are various applications of Heart Disease Prediction using Machine Learning. Some are mentioned below:

In the medical industry, when practitioners or patients need to monitor a patient's heart rate for a variety of reasons.

This model might be applied when it's necessary to assess heart rate or other heart related conditions prior to major surgery. Furthermore, even before the patient receives the necessary medical attention, this model can be used to identify the risk and severity of cardiac disease.

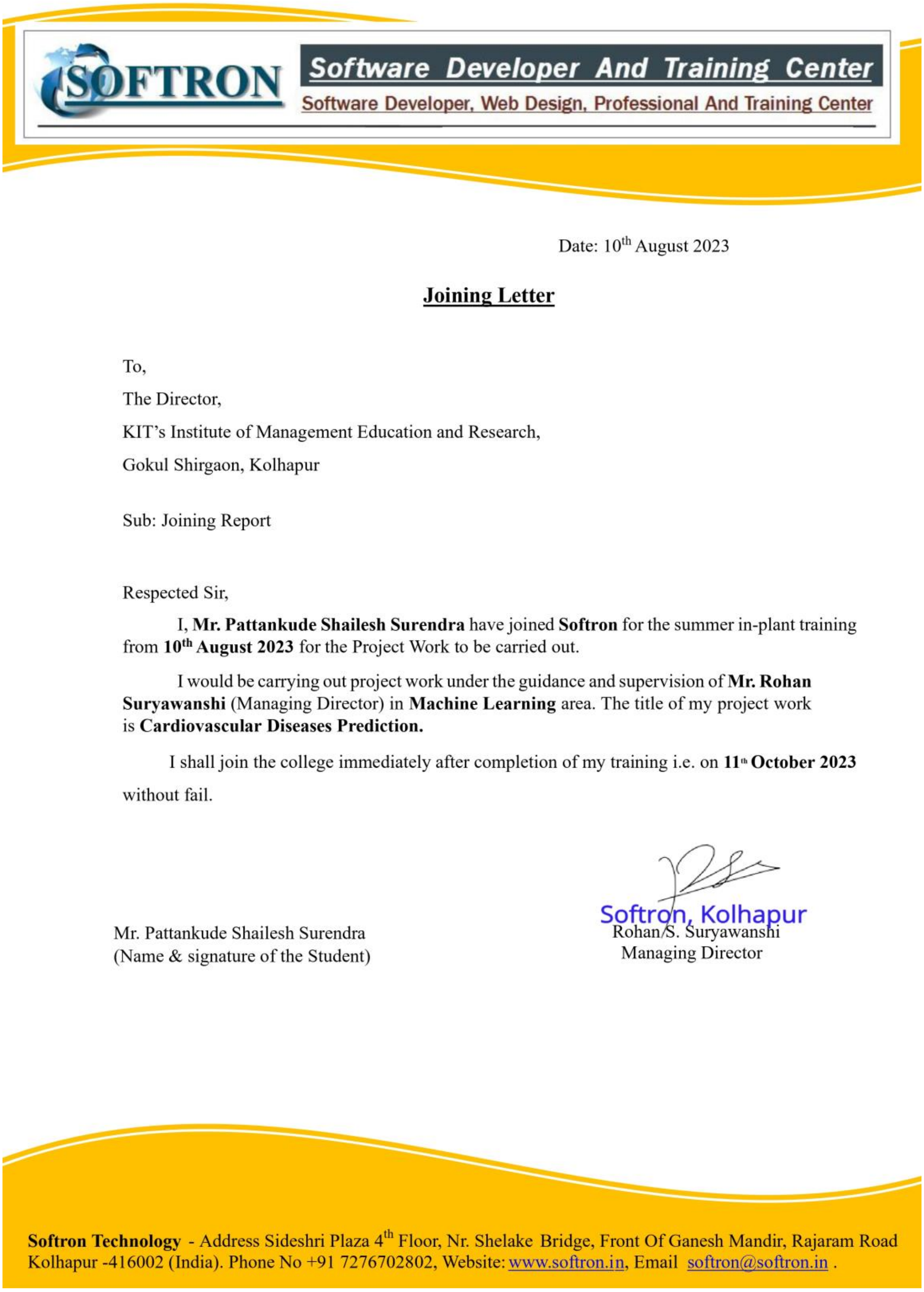
Using the website users can also see whether they are having any risk of heart disease by entering their report details on the website.

In the future a more elaborate project can be build using deep learning techniques and data mining techniques not only for heart disease prediction but also for covering other diseases.

Various other techniques of ML can also be implemented in order to develop a more efficient model. In this project we have only created a website but a mobile application can also be created for easier access.

Chapter 8
Annexure

1. Joining Report



SOFTRON Software Developer And Training Center

Software Developer, Web Design, Professional And Training Center

Date: 10th August 2023

Joining Letter

To,
The Director,
KIT's Institute of Management Education and Research,
Gokul Shirgaon, Kolhapur

Sub: Joining Report

Respected Sir,

I, **Mr. Pattankude Shailesh Surendra** have joined **Softron** for the summer in-plant training from **10th August 2023** for the Project Work to be carried out.

I would be carrying out project work under the guidance and supervision of **Mr. Rohan Suryawanshi** (Managing Director) in **Machine Learning** area. The title of my project work is **Cardiovascular Diseases Prediction.**

I shall join the college immediately after completion of my training i.e. on **11th October 2023** without fail.

Softron, Kolhapur

Mr. Pattankude Shailesh Surendra
(Name & signature of the Student)

Rohan S. Suryawanshi
Managing Director

Softron Technology - Address Sideshri Plaza 4th Floor, Nr. Shelake Bridge, Front Of Ganesh Mandir, Rajaram Road Kolhapur -416002 (India). Phone No +91 7276702802, Website: www.softron.in, Email softron@softron.in .

2. Weekly Progress Report

Weekly Progress Report 1

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	10th Aug 2023 To 16th Aug 2023
Period of progress Report	7 Days

Progress:

1. Introduction To HTML, CSS, JavaScript, Python
2. Problem Identification.
3. Project Topic Finalization.
4. Submission of Synopsis.

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 2

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	17th Aug 2023 To 23th Aug 2023
Period of progress Report	7 Days
Progress: 1.Introduction to MySQL. 2.Basic Commands in MySQL 3.Function of MySQL 4. Introduction to Flask	

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 3

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	24th Aug 2023 To 30th Aug 2023
Period of progress Report	7 Days

Progress:

1. Collecting dataset
2. Installing Flask
3. Overview of HTML, CSS, JavaScript
4. Introduction to Bootstrap

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 4

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	31st Aug 2023 To 6th Sep 2023
Period of progress Report	7 Days

Progress:

1. SRS Submission and Approval.
2. Cleaning Dataset
3. Task on Frontend development.

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 5

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	7th Sep 2023 To 13th Sep 2023
Period of progress Report	7 Days

Progress:

1. Training Dataset
2. Use case Diagram
3. State Diagram
4. Sequence Diagram

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 6

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	14th Sep 2023 To 20th Sep 2023
Period of progress Report	7 Days

Progress:

1. Testing Dataset
2. Deployment Diagram
3. Activity Diagram
4. Completed Database Design in MySQL

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 7

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	21st Sep 2023 To 27th Sep 2023
Period of progress Report	7 Days

Progress:

1. Creating Pickle File
2. Input Design Completed.

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 8

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	28th Sep 2023 To 4th Oct 2023
Period of progress Report	7 Days

Progress:

1. Output Design Completed.
2. Completed Report Generation.

Signature of Student **Signature of Industry Guide**

Weekly Progress Report 9

Name of Student	Mr. Pattankude Shailesh Surendra
Title of the Project	Cardiovascular Diseases Prediction
Name of Guide	Mr. Rohan S. Suryawanshi
Organization	Softron
Date of joining Organization	10th Aug 2023
Date of Progress Report	5th Oct 2023 To 10th Oct 2023
Period of progress Report	6 Days

Progress:

1. Project Presentation to the company.

Signature of Student **Signature of Industry Guide**

3. Student Guide Meeting Record

GUIDE STUDENT MEETING RECORD

Student Name: - Mr. Pattankude Shailesh Surendra

Contact No.: - +91 7721916221

Guide Name: - Mrs. Navni P. Chougale

Contact No.: - +91 9850908991

Topic: - Cardiovascular Diseases Prediction

Industry Name: - Softron Technology

Industry Guide Name: - Mr. Rohan Suryawanshi

Designation: - CEO

Contact No: - +91 7276702802

Sr. No.	Date	Description	Signature of Institute Guide	Signature of Student
1		Problem identification, Topic finalization Submission of synopsis. (First week of Inplant training)		
2		SRS submission and approval (Fourth week of Inplant training)		
3		Logical design of system (DFD, System flowchart, ERD, UML diagram, Decision tables, Decision tree etc. which is applicable) (Fifth week of Inplant training)		
4		Database Design (Sixth week of Inplant training)		
5		I/O Design (Eight week of Inplant training)		
6		Submission of First Draft (Second week of Sem III)		
7		Submission of Second Draft (Fifth week of Sem III)		
8		Submission of Final Draft (Tenth week of Sem III)		

Sr. No.	Date	Description of Discussion	Signature of Institute Guide	Signature of Student
1				
2				
3				
4				
5				
6				
7				
8				

Director

Chapter 9
References

References

- http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6864427&newsearch=true&queryText=disease%20prediction

- http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6395001&newsearch=true&queryText=disease%20prediction