JEPPIAAR INSTITUTE OF TECHNOLOGY

A Project Report

on

EARLY DETECTION OF CHRONIC KIDNEY DISEASE USING MACHINELEARNING

Submitted in partial fulfilment of requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the guidance of

Mrs.W.NANCY,

Submitted By

TEAM ID : PNT2022TMID25227

210619106034 -NITHISH KUMAR.K
210619106045 – SAI THARUN.K.B
210619106040 – RAKESH,S
210619106013 - DHARANEESH.B

NAALAIYATIRAN - EXPERIENTIAL PROJECT BASED LEARNING INITATIVE

ABSTRACT

In recent years, the solitary reasons for mortality in the world are chronic diseases such as heart disease, diabetes, and chronic kidney disease. These diseases should be diagnosed earlier; however, the technique is costly as well as it leads to many complications. Considering the complexity, datamining performs a major part in accurately classifying chronic disease. A new approach to classify chronic disease is by using random forest (RF). The main goal is generating an efficient and heterogeneous decision trees, while determining the optimum training sets to run at the same time. Rather utilising traditional approach like bootstrap, multi-objective firefly optimisation algorithm and random forest algorithm are proposed in this method. As a result, to train random forest, various training sets are generated with alternative instances and attributes. As a result, the performance of random forests can be improved and thus the prediction accuracy. Here we have used RandomForest Classifier to detect whether the patient has chronic disease or not and also Random Forest Regressor is used to predict the probability of chronic disease. Ourmodel gives accuracy of nearly 89%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CKD	Chronic Kidney Disease
UCI	Unique Client Identifier
NFRs	Non Functional Requirements
EDA	Exploratory Data Analysis

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Chronic diseases require long-term, continuous management. They take a long time to manifest and are difficult to cure. According to the Current Status and Future Development of Chronic Disease Management Project of the Korean Ministry of Healthand Welfare, death by five major chronic diseases (hypertension, stroke, angina pectoris, myocardial infarction, and diabetes mellitus) constituted 63.1% of the total deaths in Korea in 2003. While the cost burden of diseases has increased annually, the number of deaths caused by chronic diseases also continues to increase. Various approaches have beenintroduced to prevent chronic diseases, and most of them focus on lifestyle. However, it is difficult for individuals to change their lifestyle to prevent chronic diseases, because many people do not know which chronic diseases, they may be susceptible to based on their physical condition and medical history. Although a few approaches have been used to predict the possibility of contracting these diseases, their performance was limited because relevant information on the physical condition and medical history was often omitted. Various studies on chronic diseases have received a lot of attention since the 1990s.

A few studies were conducted if smoking, drinking, and high cholesterol levels cause chronic diseases. The cholesterol level with stroke and coronary heart disease using experimental groups. Other related studies have included reports investigating the effects of dietary supplements on preventing chronic diseases. One such dietary supplement is chlorella, which is reportedly effective in facilitating growth and improving stress-related ulcers in individuals at high risk for chronic disease.

A study evaluating the effects of chlorella use found that this supplement improved fat metabolism and lowered blood glucose levels, suggesting that it may have beneficial effects in preventing chronic disease. Disease development may also be influenced by an individual’s living environment. A recent study quantified many diseases and risk factors that correspond to environmental variables by conducting correlation analyses on stress-related variables and chronic diseases. As more health information data become available, some machine learning approaches have been implemented to predict the characteristics of chronic disease potential using data as input variables and to predict these as individual medical histories. However, studies of chronic disease are usually experimental; hence, the resulting datasets tend to contain many missing values.

Consequently, researchers are unlikely to obtain complete medical records and relevant information when analyzing chronic diseases. However, to the best of our knowledge, only a few approaches have predicted chronic diseases when there are missing values, and most of them have focused on handling them by imputation instead of implicit treatment.

1.2 PURPOSE

Chronic kidney disease (CKD) has become a global health issue and is an area of concern. It is a condition where kidneys become damaged and cannot filter toxic wastes in the body. The proposed system predominantly focuses on predicting this life-threatening disease Chronic Kidney Disease (CKD) using Classification algorithms (KNN and Naive Bayes). Proposed system is automation for chronic kidney disease prediction using classification techniques and supervised learning algorithms. The data for dataset is obtained from UCI machine learning repository which contains 25 parameters (features) including the class (CKD or NOT CKD)

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

A system that can predict multiple diseases with the help of various machine learning algorithms such as Naive Bayes, KNN, DT and SVM algorithms to bridge the gap among the patients and the doctors to achieve their own goals. The existing approaches in the field of automatic disease prediction lack the patient's trust in the model's prediction and also reduce the need for doctors, which makes the doctors get panic about their livelihood. But this method integrates a module for doctor recommendation that solves both the issues by makingsure the patient to trust due to the intervention of doctors and also improves the business of doctors.

A system that enhances the risk prediction of a patient's health condition using a deep learning approach on big data and a revised fusion node model. This deep learning model for extracting the data and logical inference is made of the combination of complex machine learning algorithm such as Bayesian fusion and neural networks. The architecture of this system consists of five layers, namely, the data layer that is responsible for data collection, data aggregation layer for data acquisition from several data sources and desired format changing, analytics layer to do proper analytics on the data aggregated, information exploration layer to create the output that makes the results of analytics understandable for users, and big data governance layer that is responsible for managing the above layers

.

2.2 REFERENCES

1. https://www.researchgate.net/publication/335698017_Detection_of_Chronic_Kidney_Disease_using_Machine_Learning_Algorithms_with_Least_Number_of_Predictors

2. https://pubmed.ncbi.nlm.nih.gov/34211680/

3. https://www.primescholars.com/articles/early-prediction-of-chronic-kidney-disease-by-using-machine-learning-techniques-92643.html

4. https://www.webology.org/abstract.php?id=3038

5. https://www.ijert.org/chronic-kidney-disease-prediction-using-machine-learning

6. https://pubmed.ncbi.nlm.nih.gov/34372817/

7. https://www.researchgate.net/publication/330637326_Machine_Learning-Based_Prediction_System_For_Chronic_Kidney_Disease_Using_Associative_Classification_Technique

8. https://ijcsmc.com/docs/papers/July2014/V3I7201499a42.pdf

9. https://www.researchgate.net/publication/344335234_Chronic_Disease_Detection_Model_Using_Machine_Learning_Techniques

10. https://www.researchgate.net/publication/359618037_An_Augmented_Artificial_Intelligence_Approach_for_Chronic_Diseases_Prediction

2.3 PROBLEM STATEMENT DEFINITION

Kidney diseases avert the normal function of the kidney. Due to the large amount of alcohol consumption kidney disease arises. Early prediction of kidney disease using classification and regression algorithms are an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of kidney disease at an early stage is complex task for the doctors. The main objective of thisproject is to analyses the parameters of various classification algorithms and compare their predictive accuracy, to find out the best classifier for determining the kidney disease.This Project examines data from kidney patients concentrating on relationships between a key list of kidney enzymes, proteins, age and gender using them to try and predict the likeliness of kidney disease. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask-based web application.

I am	The person to predict the kidney disease using Machine Learning techniques.
I'm trying to	I'm trying to Use the recent technologies to predict the human kidney disease. .
But	I am unaware of the existing technology that can help me a lot to predict the disease and I don't know to use the correct technology.
Because	I don't want to waste the cost and time.
Which makes me feel	I want a best accuracy which can predict the disease so that the people can move with their necessary treatments.

Table 2.1 Problem Statement Definition

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative tool teams can use **to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was created by Dave Gray and has gained much popularity within the agile community.

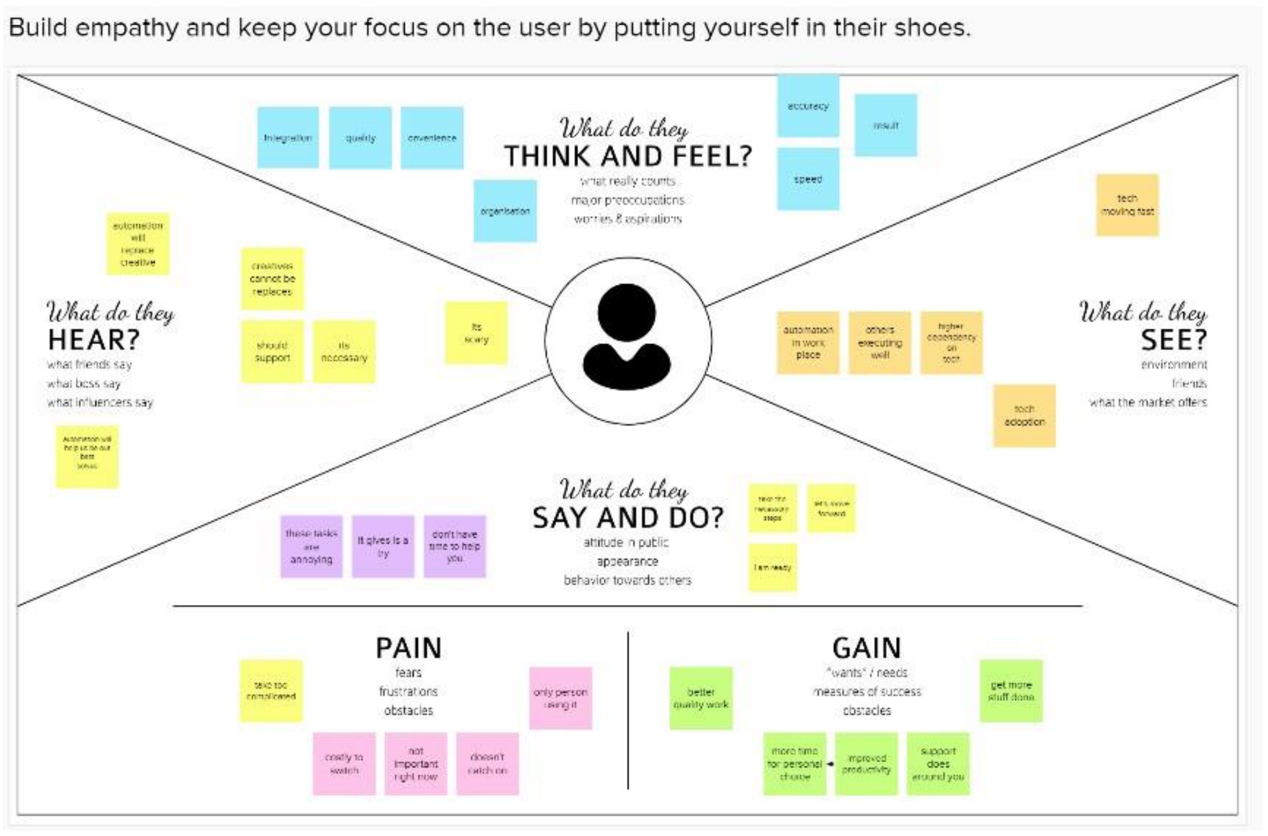


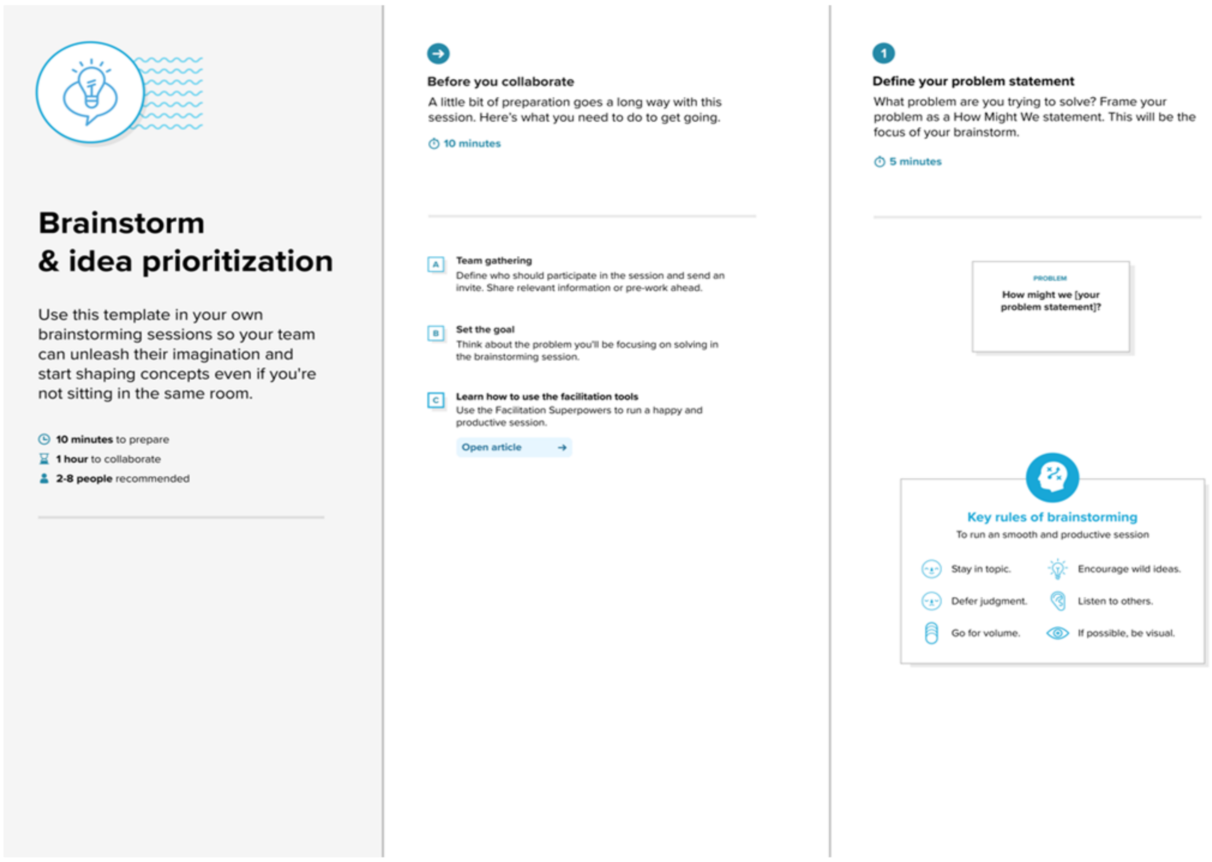
Figure 3.1 Empathy Map Canvas

The user will gets accurate results quickly.The user will see the webpage that they needs to enter user test details. The gain of this project is that it will increase the productivity but the pain is the illiterate people fells difficult to use.

3.2 IDEATION & BRAINSTORMING

Step-1: Team Gathering, Collaboration and Select the Problem Statement

A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we [your problem statement]?

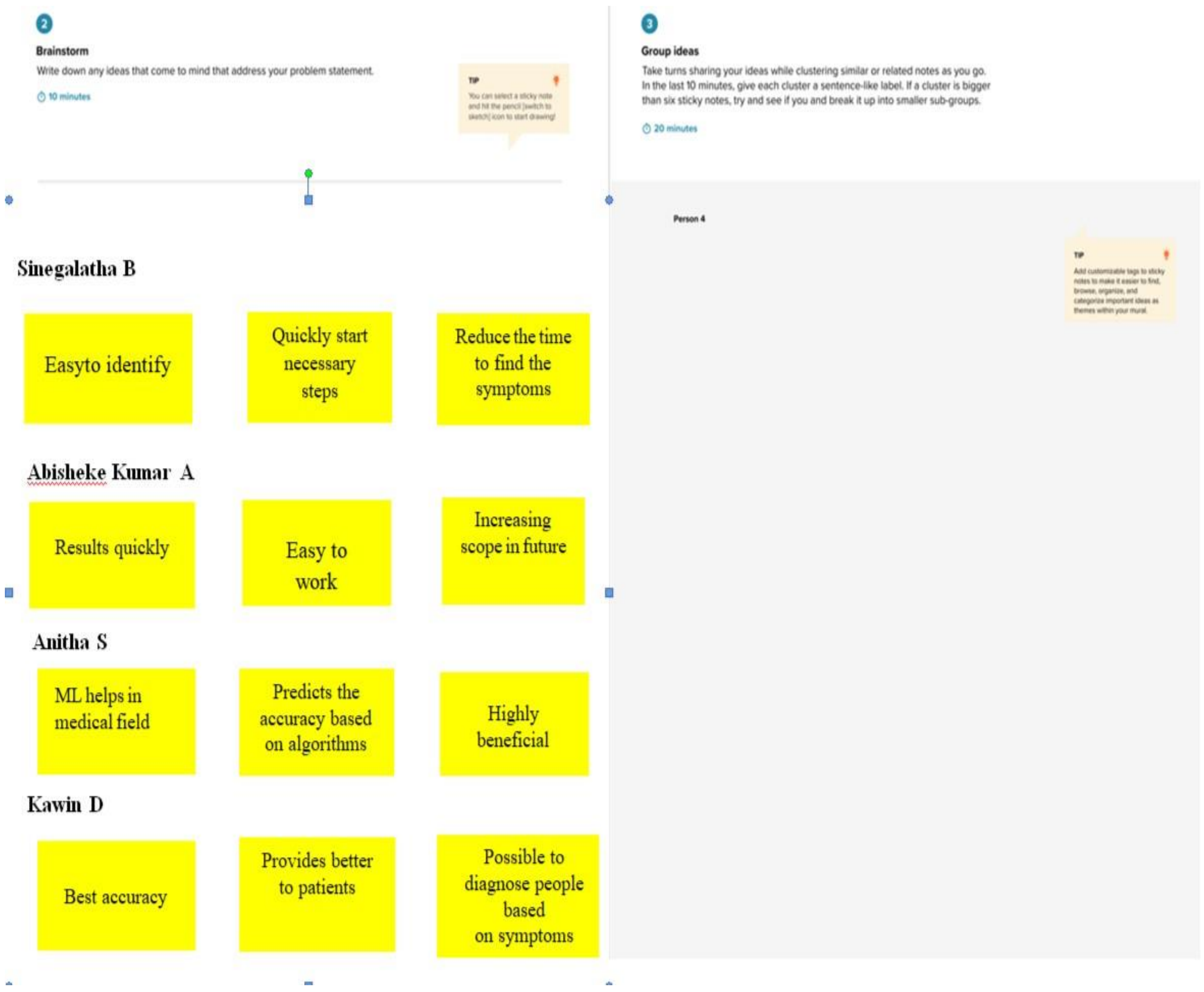
Key rules of brainstorming
To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.

Figure 3.2 Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping

The idea listing and grouping is used to organize and analyse large numbers of ideas by categorising them. By organising and reorganising ideas, students gain a better appreciation of, and dialogue about, their ideas. As students create idea clusters, new contexts and connections among themes emerge.


2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch[icon to start drawing!

Sinegalatha B

Easyto identify

Quickly start necessary steps

Reduce the time to find the symptoms

Abisheke Kumar A

Results quickly

Easy to work

Increasing scope in future

Anitha S

ML helps in medical field

Predicts the accuracy based on algorithms

Highly beneficial

Kawin D

Best accuracy

Provides better to patients

Possible to diagnose people based on symptoms

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Person 4

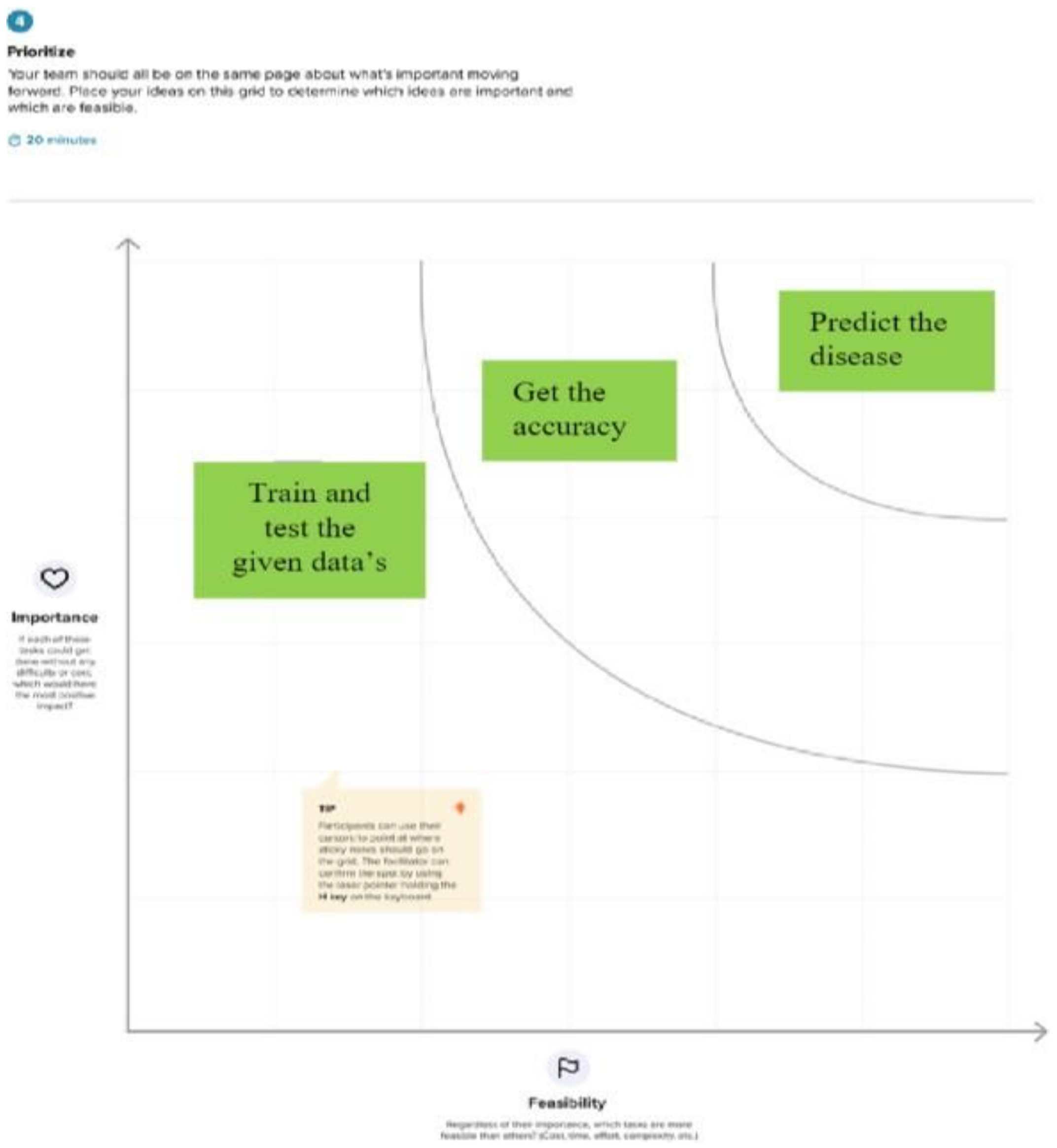
TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Figure 3.3 Brainstorm, Idea Listing and Grouping

Step-3: Idea Prioritization

Idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated to the everyday ways of working.



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Figure 3.4 Idea Prioritization

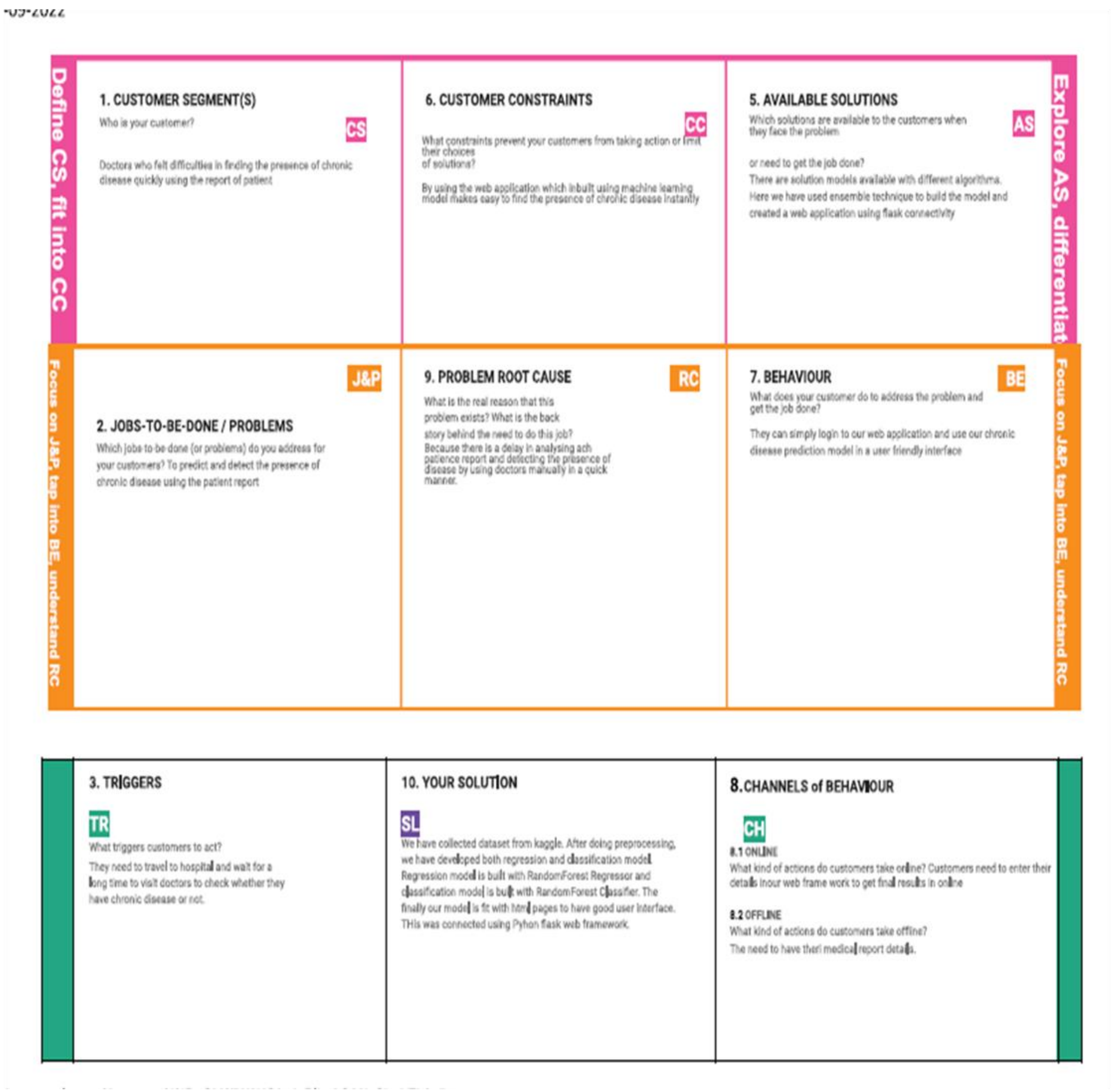
3.3 PROPOSED SOLUTION

The main treatments are: **lifestyle changes** – to help you stay as healthy as possible. medicine – to control associated problems, such as high blood pressure and high cholesterol. Dialysis – treatment to replicate some of the kidney's functions, which may be necessary in advanced (stage 5) CKD

S.No	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	Kidney diseases avert the normal function of the kidney. Early prediction of kidney disease using both classification and regression algorithms are an effective task that can help the doctors to diagnose the disease within a short duration of time.
2	Idea / Solution description	One of the easiest solutions to predict the kidney disease using Machine Learning techniques.
3	Novelty / Uniqueness	This project provides the best accuracy for predicting the kidney disease.
4	Social Impact / Customer Satisfaction	It helps to identify the kidney disease in effective way, reduce the cost and user friendly.
5	Scalability of the Solution	This project can be improved by giving medical suggestion for patients.

Table 3.1 Proposed Solution

3.4 PROBLEM SOLUTION FIT


-09-2022
Define CS, fit into CC
Explore AS, differentiat
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Doctors who felt difficulties in finding the presence of chronic disease quickly using the report of patient

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

By using the web application which inbuilt using machine learning model makes easy to find the presence of chronic disease instantly

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem

or need to get the job done?

There are solution models available with different algorithms. Here we have used ensemble technique to build the model and created a web application using flask connectivity

J&P

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers? To predict and detect the presence of chronic disease using the patient report

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

Because there is a delay in analysing ach patience report and detecting the presence of disease by using doctors manually in a quick manner.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

They can simply login to our web application and use our chronic disease prediction model in a user friendly interface

3. TRIGGERS

TR

What triggers customers to act?

They need to travel to hospital and wait for a long time to visit doctors to check whether they have chronic disease or not.

10. YOUR SOLUTION

SL

We have collected dataset from kaggle. After doing preprocessing, we have developed both regression and classification model. Regression model is built with RandomForest Regressor and classification model is built with RandomForest Classifier. The finally our model is fit with html pages to have good user interface. THis was connected using Pyhon flask web framework.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE

What kind of actions do customers take online? Customers need to enter their details inour web frame work to get final results in online

8.2 OFFLINE

What kind of actions do customers take offline?

The need to have theri medical report details.

Figure 3.5 Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks.

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Home Page	● Chronic Kidney disease description ● Information about Test Vitals required for prediction ● If new User, REGISTER ● If already exist, SIGN IN
FR-2	User Registration	● Enters Mail ID and other personal details required for Registering.
FR-3	User Login	● Uses Mail ID and Password for login
FR-4	Test Vitals Form	● Test Vitals should be entered for prediction
FR-5	Result	● If Positive – Test Result along with the Information about what is to be done next will be displayed. ● If Negative – Test result along with preventive measures to prevent themselves from getting chronic kidney disease will be displayed..

Table 4.1 Functional Requirement

4.2 NON - FUNCTIONAL REQUIREMENT

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Even Illiterates and people with no understanding of computer/mobile should be able to use the product.
NFR-2	Security	Access permission for particular system information may be changed by systems data administration.
NFR-3	Reliability	The database update process must roll back all related updates when any updates fails.
NFR-4	Performance	The Home-page load time must be no more than 2 seconds for users that access the website using an LTE mobile connection.
NFR-5	Availability	New Model Deployment must not impact home page, test page and result page availability and must not take longer than one hour.
NFR-6	Scalability	The website Traffic limit must be scalable enough to support 2000,000 users at a time.

Table 4.2 Non-Functional Requirement

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

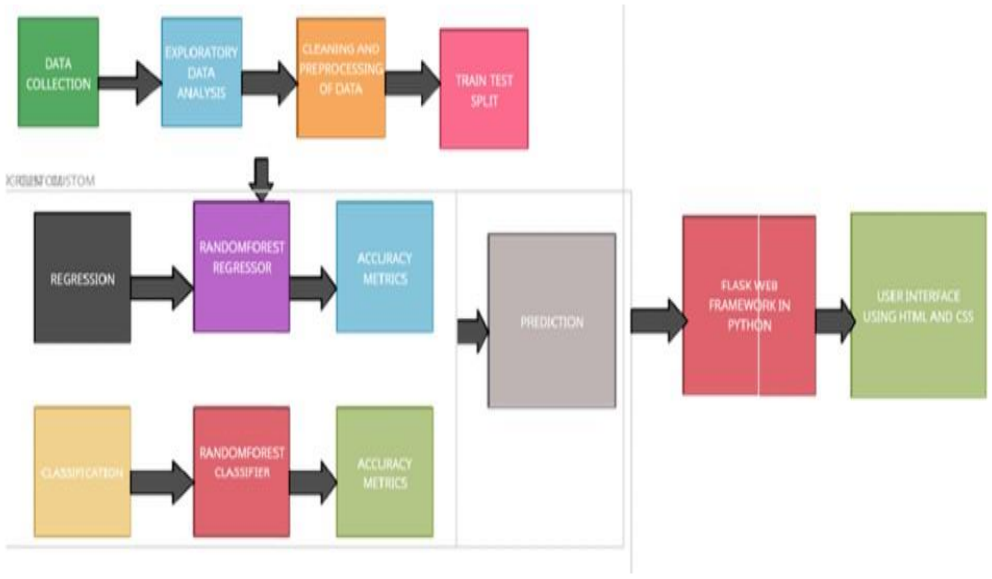


Figure 5.1 Data Flow of Chronic Disease Prediction

1. Medical data of patients is collected from Kaggle.
2. Exploratory data analysis done on the input dataset.
3. Then removal of null values, duplicates and outliers.
4. Then the dependent and independent variable is defined.
5. Train test split is done.
6. Both classification and regression model are built.
7. For Classification, the model is trained with Random Forest Classifier and tested with test dataset.
8. For Regression, the model is trained with Random Forest Regression and tested with test dataset.
9. Then the model is fitted with front end which is developed using HTML, CSS with the help of Python Flask Web Framework.
10. Finally, the output will be predicted for the user input data.

5.2 SOLUTION & TECHNICAL ARCHITECTURE

The solution architecture which can explains to collect the dataset and move to pre-processing, then need to train and test datas with the help of machine learning algorithms, to predict the output.

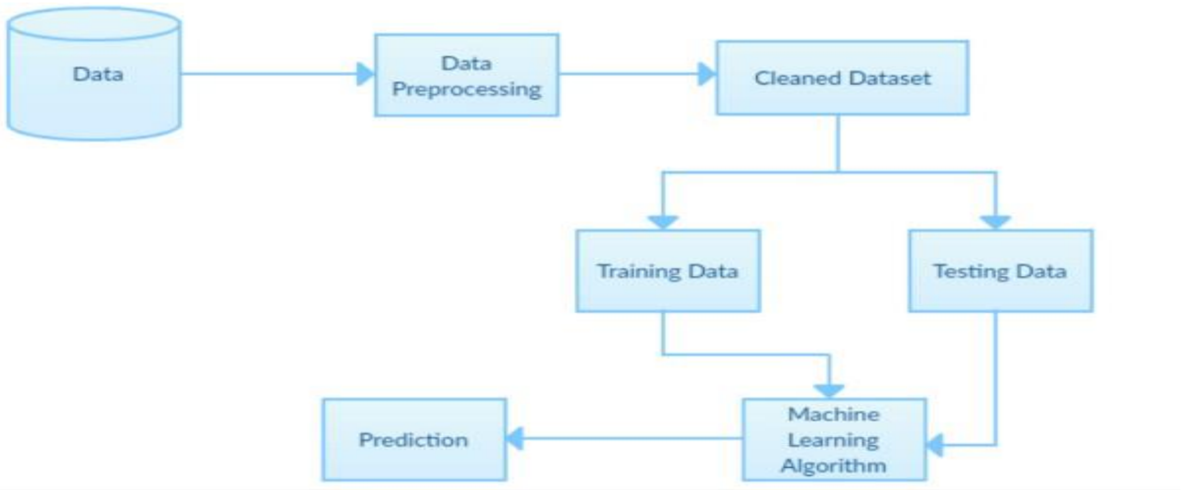


Figure 5.2 Solution Architecture

Technical architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goalof technical architects is to achieve all the business needs with an application that is optimized for both performance and security.

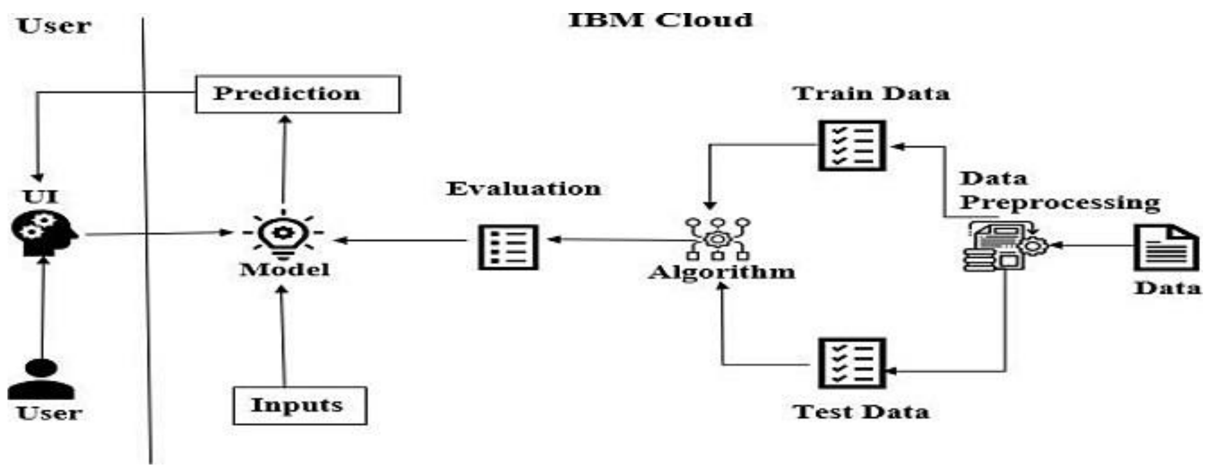


Figure 5.3 Technical Architecture

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web-user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email and password	Check whether password and email is correct	High	sprint-1

Customer (Web-user)	Dashboard	USN-4	If the email id and password is correct, the user can log in to application otherwise it shows 'incorrect password or Id’.	View the dashboard of user who is login	High	Sprint-1
Customer Care Executive	Help	USN-5	If the user faces any issues, he/she an report it to our mail id.	Report option will be available in web app	High	Sprint-2
Administrat or	Verification	USN-6	Administrator also has unique Id and password to login. He has additional users to organize the users of this web app	Check whether password and email is correct	High	Sprint-3

Table 5.1 User Story

CHAPTRE 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my name, mobile number, email, password, and confirming my password.	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-2		USN-2	As a user, I can register for the application through Gmail	5	Medium	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-1	User Confirmation	USN-3	As a user, I will receive confirmation email once I have registered for the application	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-2		USN-4	As a user, I will receive confirmation OTP to verify the identity.	5	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K

Sprint-2	User Confirmation	USN-5	As a user, I will enter the input data for disease prediction in the form	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-3	Provide output to the user	USN-6	As a user, I will get the result of disease prediction in the dashboard.	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-3	Data Analysis	USN-7	As the admin, I will develop modules to pre-process and store the data.	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-4	Prediction of disease	USN-8	As the admin, I will build a Machine Learning model to predict the disease	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K
Sprint-4	Final Delivery	USN-9	Deploy the application in IBM cloud and make it available for use.	10	High	SAI THARUN.K.B RAKESH S DHARANEESH.B NITHISH KUMAR.K

Table 6.1 Sprint Planning & Estimation

6.2 SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	3	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	2	05 Nov 2022
Sprint-3	20	6 Days	07 Oct 2022	12 Nov 2022	3	12 Nov 2022
Sprint-4	20	6 Days	14 Oct 2022	19 Nov 2022	3	19 Nov 2022

Table 6.2 Sprint Delivery Schedule

Velocity:

We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). The team’s average velocity (AV) per iteration unit (story points per day)

AV = Sprint duration / velocity = 20 / 6 = 3.33

6.3 REPORTS FROM JIRA

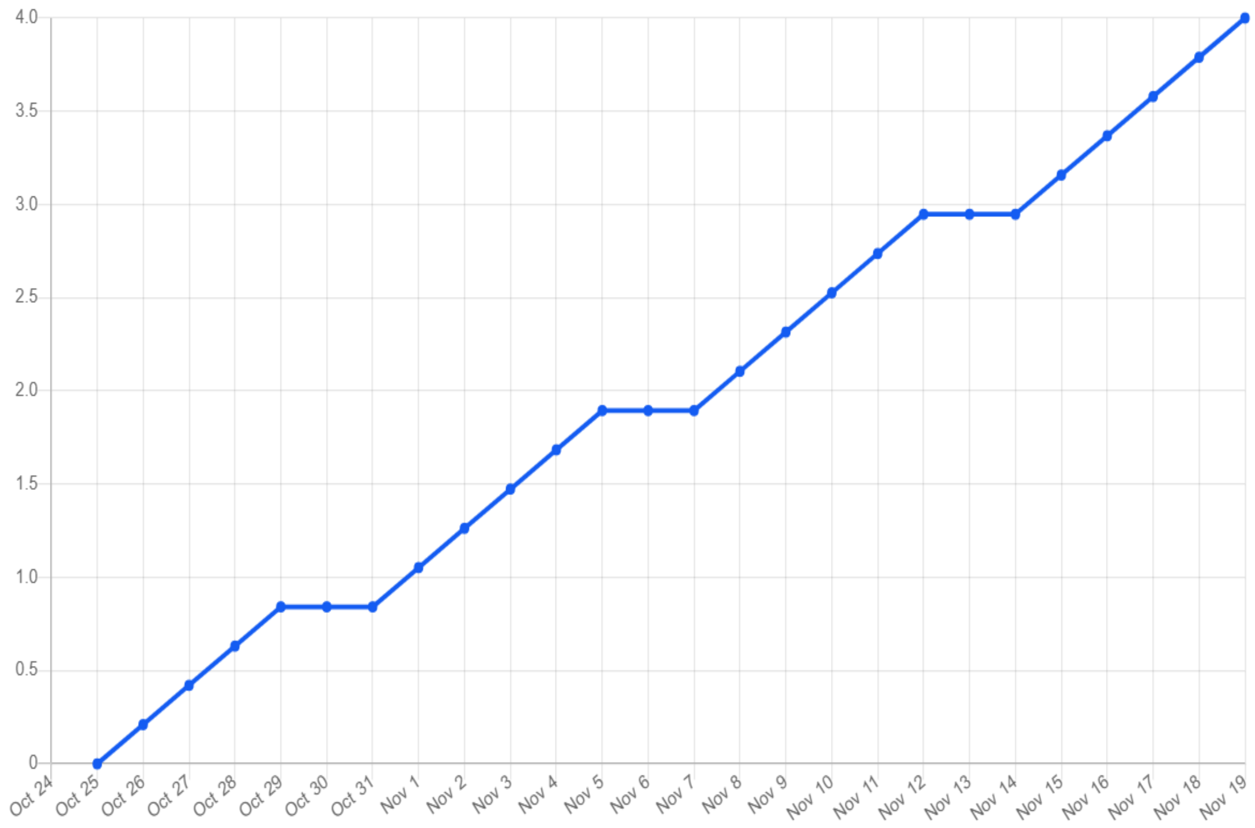


Figure 6.1 Reports from Jira

Sprint 1	**:** Oct 24,2022 - Oct 29,2022
Sprint 2	**:** Oct 31,2022 - Nov 05,2022
Sprint 3	**:** Nov 07,2022 - Nov 12,2022
Sprint 4	**:** Nov 14,2022 - Nov 19,2022

CHAPTER 7

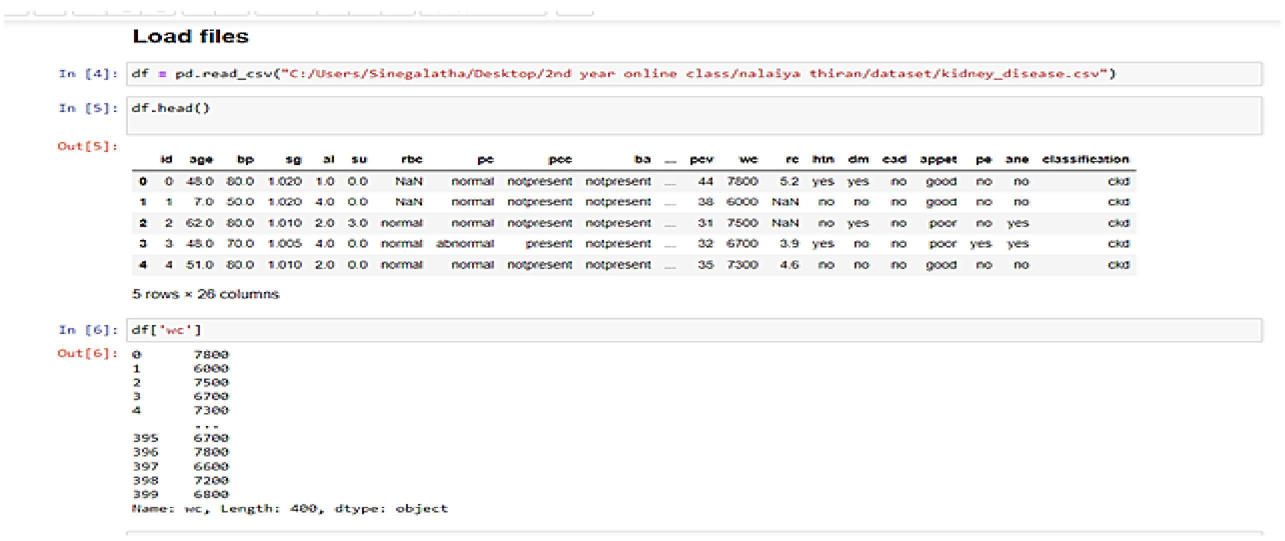
CODING AND SOLUTIONING

7.1 Feature 1

Different types of python libraries such as pandas, Sklearn, NumPy, matplotlib are used for processing the algorithms. Using exploration data analysis technique data was analyses in junketeer notebook.10-fold cross validation technique is used for spitting the data set into training and testing data.Then using random forest algorithm dataset was processed.

Collection of dataset

For the proposed study dataset was taken from Kaggle site. Then it was downloaded in excel file using comma separated format. Data has processed by python programming using Jupiter notebook. The data set contains 401 sample instances. The dataset contains 26 clinical features



Load files

```
In [4]: df = pd.read_csv("C:/Users/Sinegalatha/Desktop/2nd year online class/nalaiya thiran/dataset/kidney_disease.csv")

In [5]: df.head()

Out[5]:
```

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd

5 rows × 26 columns

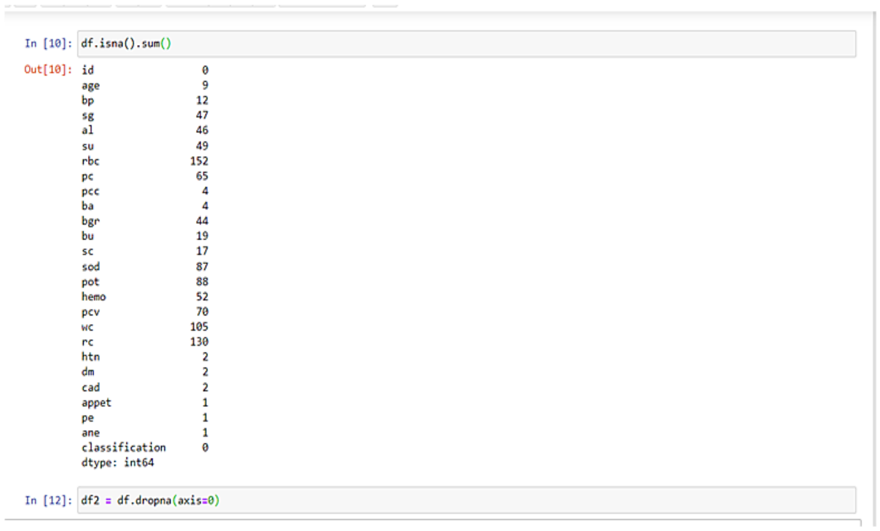
```
In [6]: df['wc']

Out[6]: 0       7800
        1       6000
        2       7500
        3       6700
        4       7300
                ...
        395     6700
        396     7800
        397     6600
        398     7200
        399     6800
        Name: wc, Length: 400, dtype: object
```

Figure 7.1 Collection of Dataset

Preprocessing - Data cleaning

Checking Null Entries

The most important step in EDA involving removing duplicate rows/columns, filling the void entries with values like mean/median of the data, dropping various values, removing null entries Here we have check for null values and drop the entries which contains null values as the percentage of null values in dataset is very less.

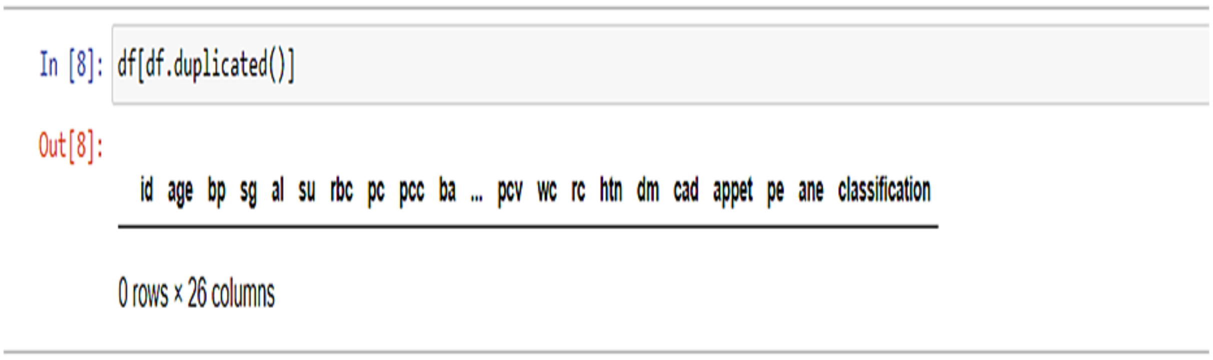


```
In [10]: df.isna().sum()

Out[10]: id                  0
         age                 9
         bp                 12
         sg                 47
         al                 46
         su                 49
         rbc               152
         pc                 65
         pcc                 4
         ba                  4
         bgr                44
         bu                 19
         sc                 17
         sod                87
         pot                88
         hemo               52
         pcv                70
         wc                105
         rc                130
         htn                 2
         dm                  2
         cad                 2
         appet               1
         pe                  1
         ane                 1
         classification      0
         dtype: int64

In [12]: df2 = df.dropna(axis=0)
```

Figure 7.2 Checking Null Entries

Checking Duplicates

In the dataset, there is no duplicate entries

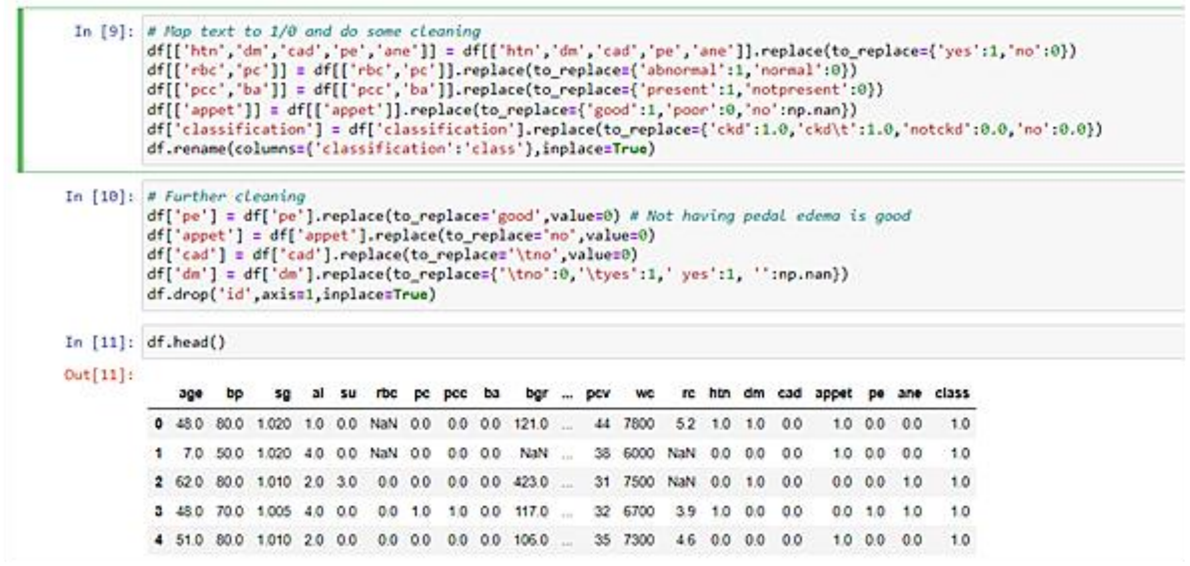


```
In [8]: df[df.duplicated()]

Out[8]:
   id age bp sg al su rbc pc pcc ba ... pcv wc rc htn dm cad appet pe ane classification

0 rows × 26 columns
```

Figure 7.3 Checking Duplicates

Encoding:

All the categorical columns('htn','dm','cad','pe','ane', 'rbc','pc', 'pcc','ba', 'appet', 'classification') in dataset is converted into numerical

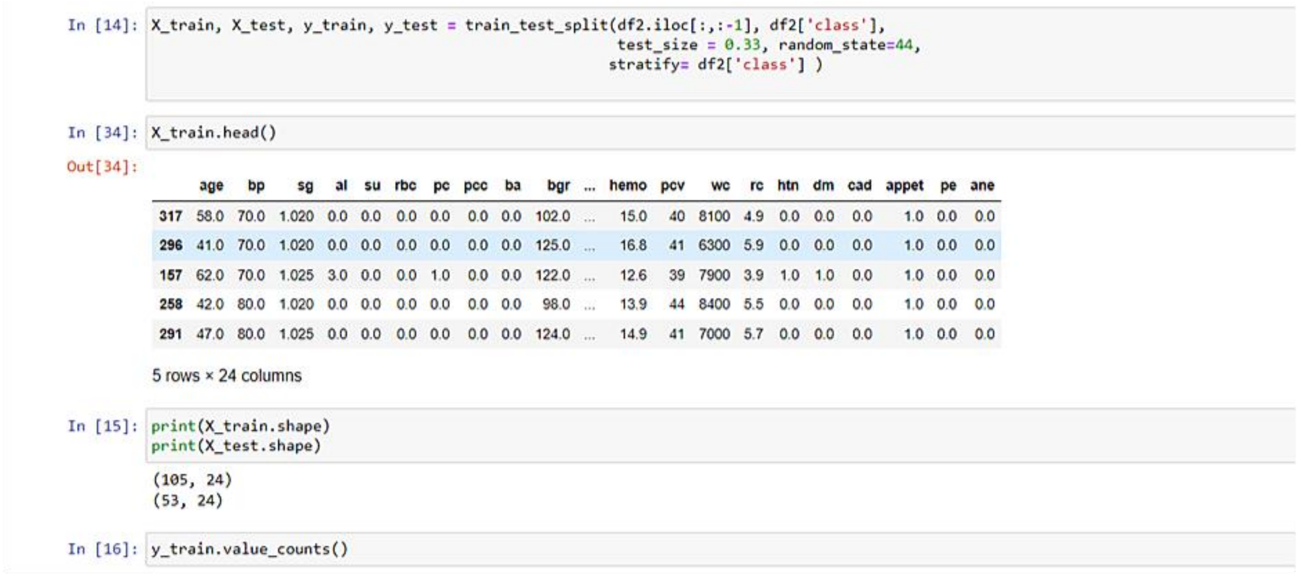


```
In [9]: # Map text to 1/0 and do some cleaning
        df[['htn','dm','cad','pe','ane']] = df[['htn','dm','cad','pe','ane']].replace(to_replace={'yes':1,'no':0})
        df[['rbc','pc']] = df[['rbc','pc']].replace(to_replace={'abnormal':1,'normal':0})
        df[['pcc','ba']] = df[['pcc','ba']].replace(to_replace={'present':1,'notpresent':0})
        df[['appet']] = df[['appet']].replace(to_replace={'good':1,'poor':0,'no':np.nan})
        df['classification'] = df['classification'].replace(to_replace={'ckd':1.0,'ckd\t':1.0,'notckd':0.0,'no':0.0})
        df.rename(columns={'classification':'class'},inplace=True)

In [10]: # Further cleaning
         df['pe'] = df['pe'].replace(to_replace='good',value=0) # Not having pedal edema is good
         df['appet'] = df['appet'].replace(to_replace='no',value=0)
         df['cad'] = df['cad'].replace(to_replace='\tno',value=0)
         df['dm'] = df['dm'].replace(to_replace={'\tno':0,'\tyes':1,' yes':1, '':np.nan})
         df.drop('id',axis=1,inplace=True)

In [11]: df.head()
Out[11]:
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	class
0	48.0	80.0	1.020	1.0	0.0	NaN	0.0	0.0	0.0	121.0	...	44	7800	5.2	1.0	1.0	0.0	1.0	0.0	0.0	1.0
1	7.0	50.0	1.020	4.0	0.0	NaN	0.0	0.0	0.0	NaN	...	38	6000	NaN	0.0	0.0	0.0	1.0	0.0	0.0	1.0
2	62.0	80.0	1.010	2.0	3.0	0.0	0.0	0.0	0.0	423.0	...	31	7500	NaN	0.0	1.0	0.0	0.0	0.0	1.0	1.0
3	48.0	70.0	1.005	4.0	0.0	0.0	1.0	1.0	0.0	117.0	...	32	6700	3.9	1.0	0.0	0.0	0.0	1.0	1.0	1.0
4	51.0	80.0	1.010	2.0	0.0	0.0	0.0	0.0	0.0	106.0	...	35	7300	4.6	0.0	0.0	0.0	1.0	0.0	0.0	1.0

Figure 7.4 Cleaning and preprocessing the data

Split of dependent and Independent Variables:



```
In [14]: X_train, X_test, y_train, y_test = train_test_split(df2.iloc[:,:-1], df2['class'],
                                                             test_size = 0.33, random_state=44,
                                                            stratify= df2['class'] )

In [34]: X_train.head()
Out[34]:
```

	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	...	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	ane
317	58.0	70.0	1.020	0.0	0.0	0.0	0.0	0.0	0.0	102.0	...	15.0	40	8100	4.9	0.0	0.0	0.0	1.0	0.0	0.0
296	41.0	70.0	1.020	0.0	0.0	0.0	0.0	0.0	0.0	125.0	...	16.8	41	6300	5.9	0.0	0.0	0.0	1.0	0.0	0.0
157	62.0	70.0	1.025	3.0	0.0	0.0	1.0	0.0	0.0	122.0	...	12.6	39	7900	3.9	1.0	1.0	0.0	1.0	0.0	0.0
258	42.0	80.0	1.020	0.0	0.0	0.0	0.0	0.0	0.0	98.0	...	13.9	44	8400	5.5	0.0	0.0	0.0	1.0	0.0	0.0
291	47.0	80.0	1.025	0.0	0.0	0.0	0.0	0.0	0.0	124.0	...	14.9	41	7000	5.7	0.0	0.0	0.0	1.0	0.0	0.0

5 rows × 24 columns

```
In [15]: print(X_train.shape)
         print(X_test.shape)

         (105, 24)
         (53, 24)

In [16]: y_train.value_counts()
```

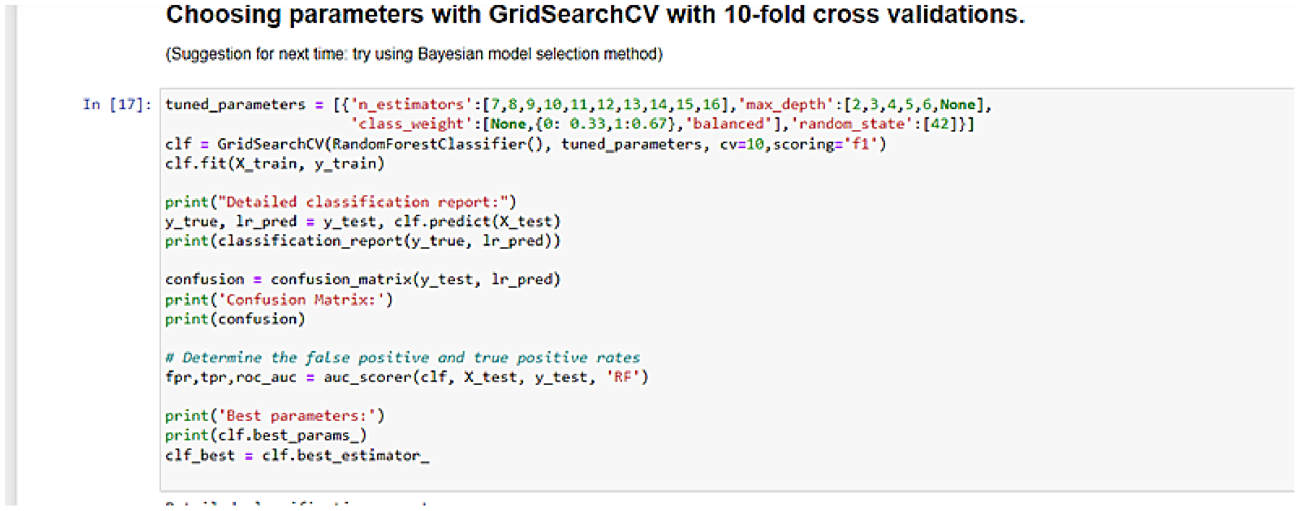
Figure 7.5 Split of dependent and Independent Variables

7.2 Feature 2

Both Classification and Regression models are built for this use case.

For classification:

RandomForest Classifier is used to detect whether the person has chronic disease or not GridSearchCV is used for hyper tuning.

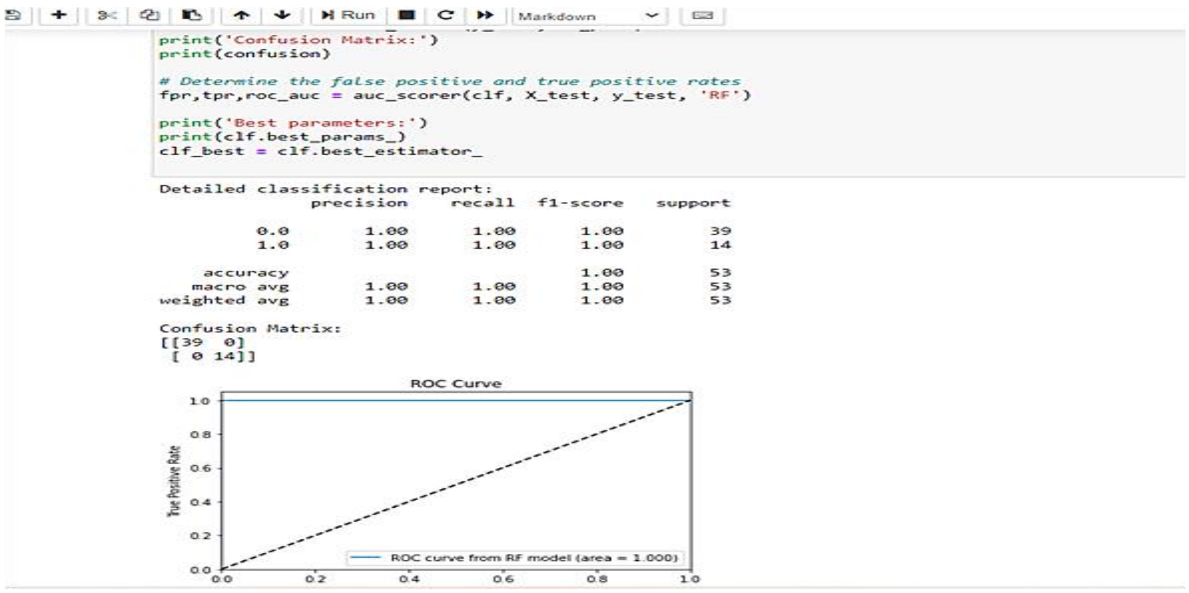


Choosing parameters with GridSearchCV with 10-fold cross validations.

(Suggestion for next time: try using Bayesian model selection method)

```
In [17]: tuned_parameters = [{'n_estimators':[7,8,9,10,11,12,13,14,15,16],'max_depth':[2,3,4,5,6,None],
                              'class_weight':[None,{0: 0.33,1:0.67},'balanced'],'random_state':[42]}]
         clf = GridSearchCV(RandomForestClassifier(), tuned_parameters, cv=10,scoring='f1')
         clf.fit(X_train, y_train)

         print("Detailed classification report:")
         y_true, lr_pred = y_test, clf.predict(X_test)
         print(classification_report(y_true, lr_pred))

         confusion = confusion_matrix(y_test, lr_pred)
         print('Confusion Matrix:')
         print(confusion)

         # Determine the false positive and true positive rates
         fpr,tpr,roc_auc = auc_scorer(clf, X_test, y_test, 'RF')

         print('Best parameters:')
         print(clf.best_params_)
         clf_best = clf.best_estimator_
```

Figure 7.6 Choosing Parameters



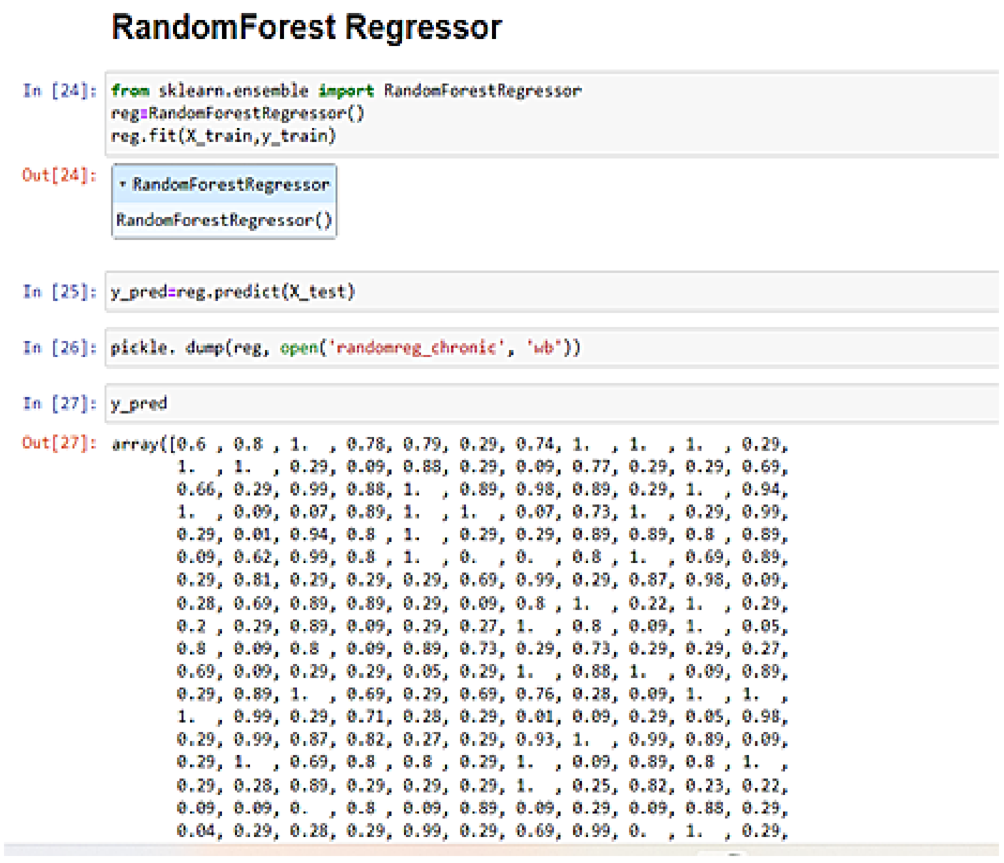
```
print('Confusion Matrix:')
print(confusion)

# Determine the false positive and true positive rates
fpr,tpr,roc_auc = auc_scorer(clf, X_test, y_test, 'RF')

print('Best parameters:')
print(clf.best_params_)
clf_best = clf.best_estimator_
```

```
Detailed classification report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        39
         1.0       1.00      1.00      1.00        14

    accuracy                           1.00        53
   macro avg       1.00      1.00      1.00        53
weighted avg       1.00      1.00      1.00        53

Confusion Matrix:
[[39  0]
 [ 0 14]]
```

Figure 7.7 Choosing Parameters

For Regression:

RandomForest Regression is used to find the probability level of patient to have chronic disease.



RandomForest Regressor

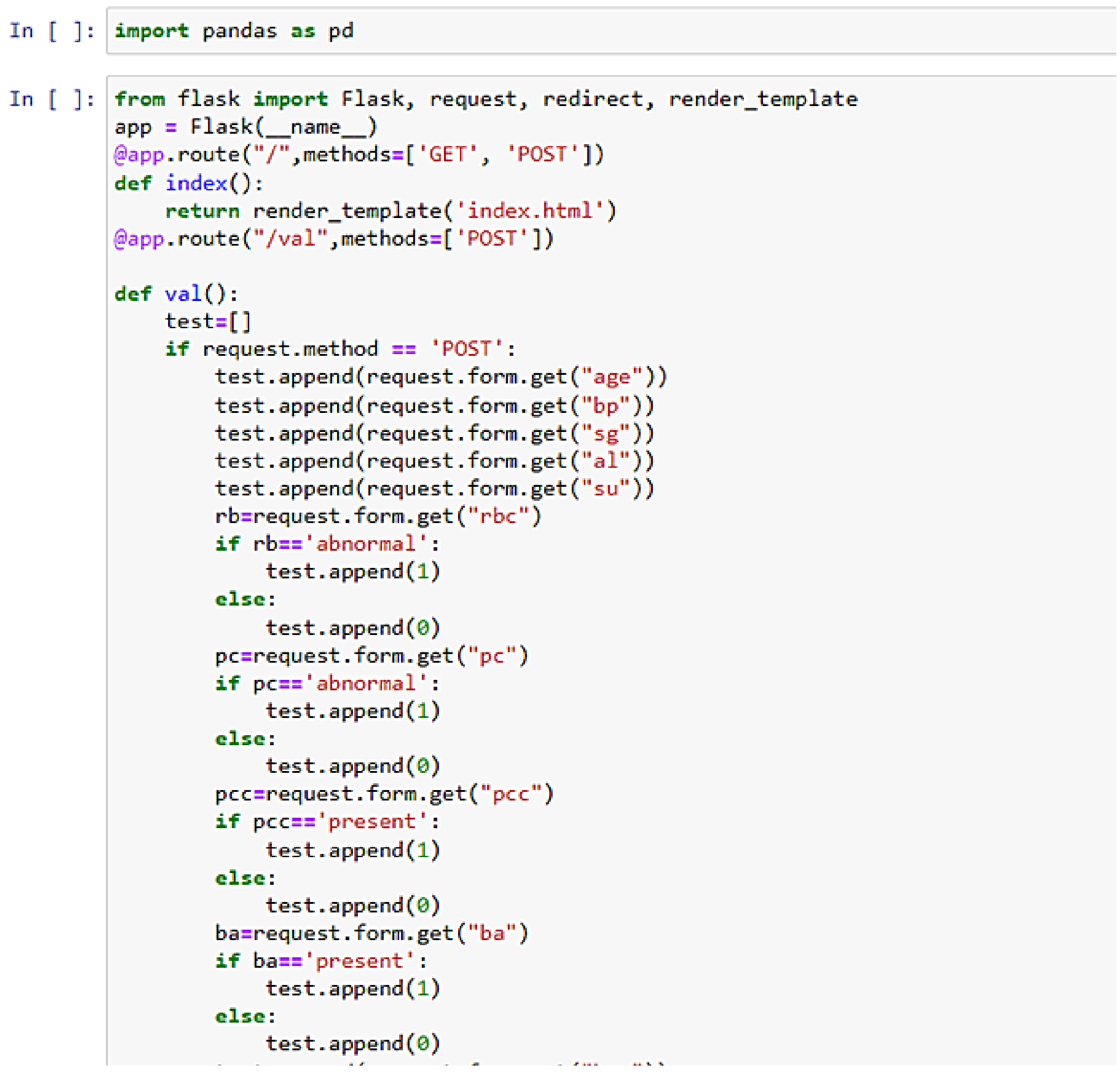
```
In [24]: from sklearn.ensemble import RandomForestRegressor
         reg=RandomForestRegressor()
         reg.fit(X_train,y_train)

Out[24]: ▾ RandomForestRegressor
         RandomForestRegressor()

In [25]: y_pred=reg.predict(X_test)

In [26]: pickle. dump(reg, open('randomreg_chronic', 'wb'))

In [27]: y_pred

Out[27]: array([0.6 , 0.8 , 1.  , 0.78, 0.79, 0.29, 0.74, 1.  , 1.  , 1.  , 0.29,
                1.  , 1.  , 0.29, 0.09, 0.88, 0.29, 0.09, 0.77, 0.29, 0.29, 0.69,
                0.66, 0.29, 0.99, 0.88, 1.  , 0.89, 0.98, 0.89, 0.29, 1.  , 0.94,
                1.  , 0.09, 0.07, 0.89, 1.  , 1.  , 0.07, 0.73, 1.  , 0.29, 0.99,
                0.29, 0.01, 0.94, 0.8 , 1.  , 0.29, 0.29, 0.89, 0.89, 0.8 , 0.89,
                0.09, 0.62, 0.99, 0.8 , 1.  , 0.  , 0.  , 0.8 , 1.  , 0.69, 0.89,
                0.29, 0.81, 0.29, 0.29, 0.29, 0.69, 0.99, 0.29, 0.87, 0.98, 0.09,
                0.28, 0.69, 0.89, 0.89, 0.29, 0.09, 0.8 , 1.  , 0.22, 1.  , 0.29,
                0.2 , 0.29, 0.89, 0.09, 0.29, 0.27, 1.  , 0.8 , 0.09, 1.  , 0.05,
                0.8 , 0.09, 0.8 , 0.09, 0.89, 0.73, 0.29, 0.73, 0.29, 0.29, 0.27,
                0.69, 0.09, 0.29, 0.29, 0.05, 0.29, 1.  , 0.88, 1.  , 0.09, 0.89,
                0.29, 0.89, 1.  , 0.69, 0.29, 0.69, 0.76, 0.28, 0.09, 1.  , 1.  ,
                1.  , 0.99, 0.29, 0.71, 0.28, 0.29, 0.01, 0.09, 0.29, 0.05, 0.98,
                0.29, 0.99, 0.87, 0.82, 0.27, 0.29, 0.93, 1.  , 0.99, 0.89, 0.09,
                0.29, 1.  , 0.69, 0.8 , 0.8 , 0.29, 1.  , 0.09, 0.89, 0.8 , 1.  ,
                0.29, 0.28, 0.89, 0.29, 0.29, 0.29, 1.  , 0.25, 0.82, 0.23, 0.22,
                0.09, 0.09, 0.  , 0.8 , 0.09, 0.89, 0.09, 0.29, 0.09, 0.88, 0.29,
                0.04, 0.29, 0.28, 0.29, 0.99, 0.29, 0.69, 0.99, 0.  , 1.  , 0.29,
```

\

Figure 7.8 Random Forest Regression

Flask Connectivity

The Backend Machine Learning model code is connect with HTML code by using python Flask Web Framework.



```
In [ ]: import pandas as pd
```

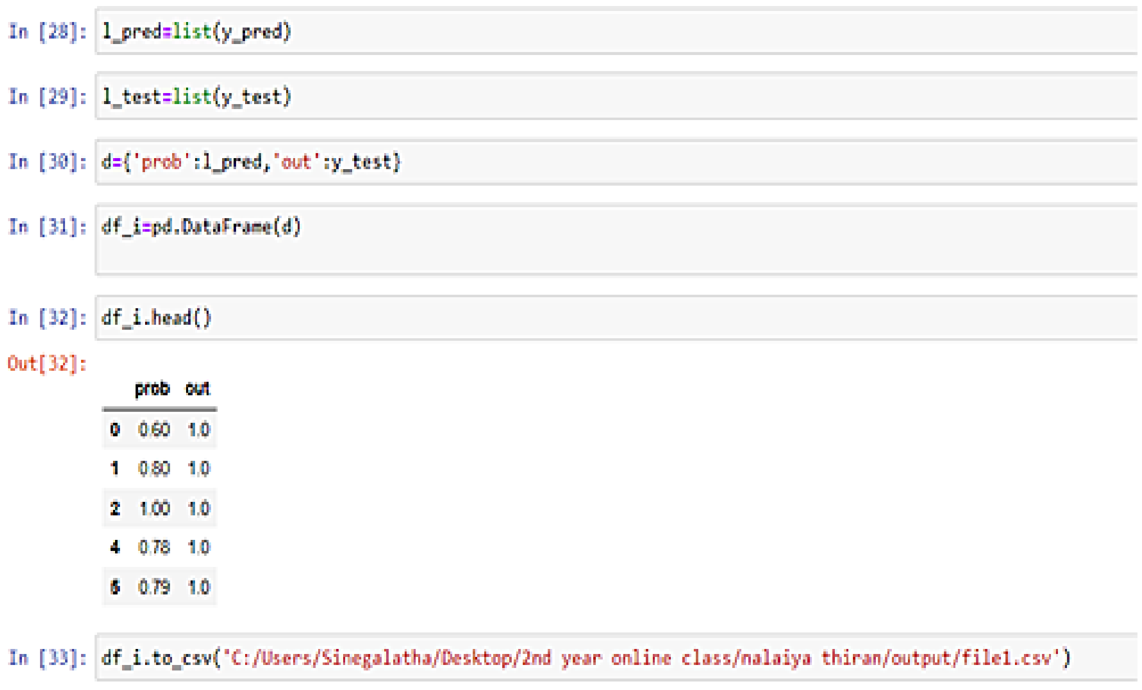
```
In [ ]: from flask import Flask, request, redirect, render_template
        app = Flask(__name__)
        @app.route("/",methods=['GET', 'POST'])
        def index():
            return render_template('index.html')
        @app.route("/val",methods=['POST'])

        def val():
            test=[]
            if request.method == 'POST':
                test.append(request.form.get("age"))
                test.append(request.form.get("bp"))
                test.append(request.form.get("sg"))
                test.append(request.form.get("al"))
                test.append(request.form.get("su"))
                rb=request.form.get("rbc")
                if rb=='abnormal':
                    test.append(1)
                else:
                    test.append(0)
                pc=request.form.get("pc")
                if pc=='abnormal':
                    test.append(1)
                else:
                    test.append(0)
                pcc=request.form.get("pcc")
                if pcc=='present':
                    test.append(1)
                else:
                    test.append(0)
                ba=request.form.get("ba")
                if ba=='present':
                    test.append(1)
                else:
                    test.append(0)
```

Figure 7.9 Flask connectivity

CHAPTER 8

TESTING

8.1 Test Cases



```
In [28]: l_pred=list(y_pred)

In [29]: l_test=list(y_test)

In [30]: d={'prob':l_pred,'out':y_test}

In [31]: df_i=pd.DataFrame(d)

In [32]: df_i.head()

Out[32]:
     prob  out
  0  0.60  1.0
  1  0.80  1.0
  2  1.00  1.0
  4  0.78  1.0
  6  0.79  1.0

In [33]: df_i.to_csv('C:/Users/Sinegalatha/Desktop/2nd year online class/nalaiya thiran/output/file1.csv')
```

Figure 8.1 Test Case

8.2 User Acceptance Testing

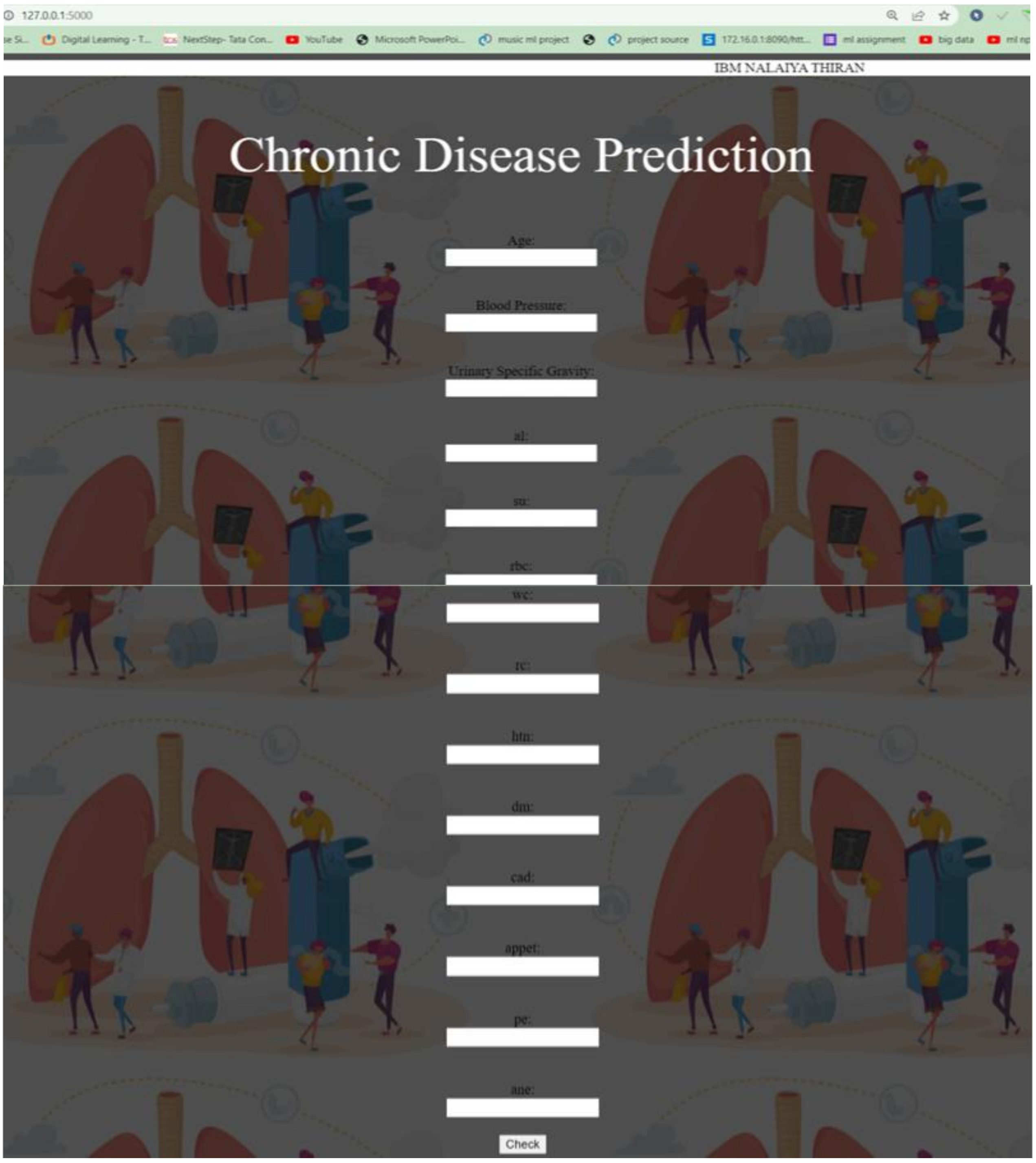
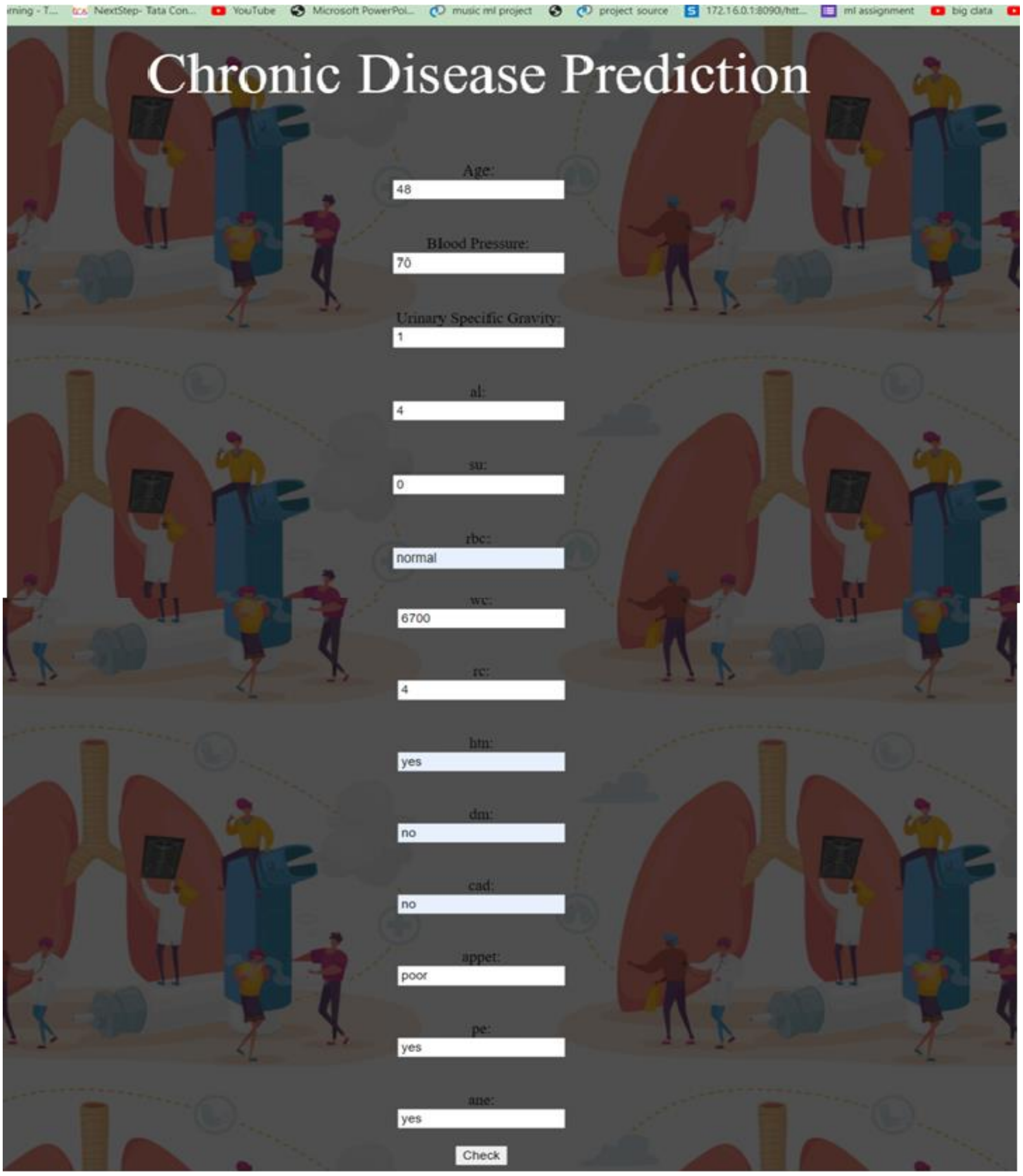


Figure 8.2 User input data acceptance



Chronic Disease Prediction

Age: 48

Blood Pressure: 70

Urinary Specific Gravity: 1

al: 4

su: 0

rbc: normal

wc: 6700

rc: 4

htn: yes

dm: no

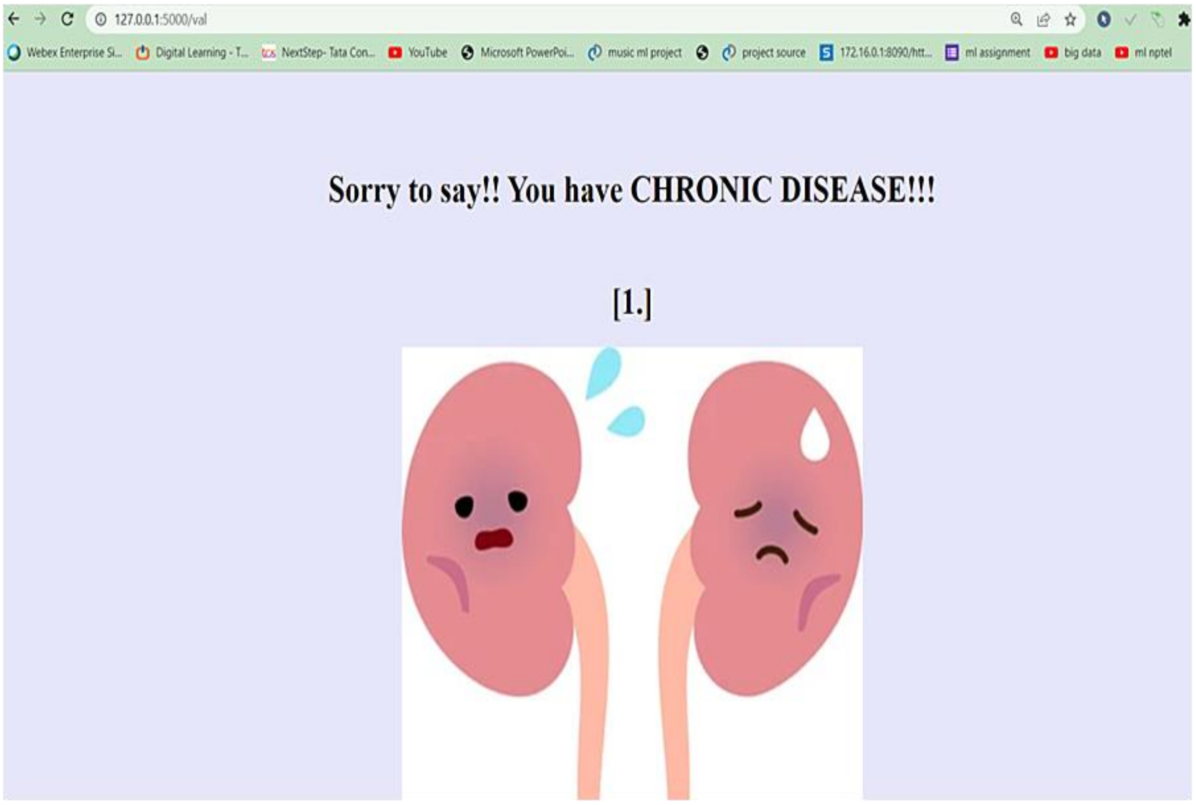
cad: no

appet: poor

pe: yes

ane: yes

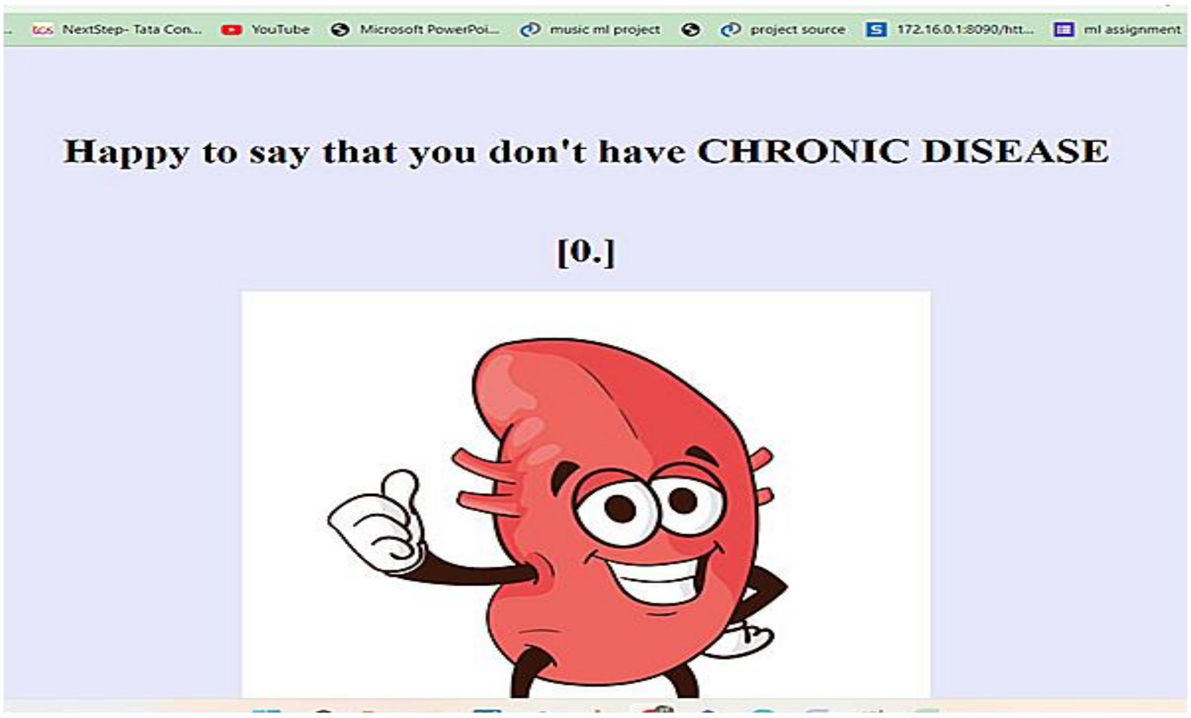
Figure 8.3 User data entry



Sorry to say!! You have CHRONIC DISEASE!!!

[1.]

Figure 8.4 Chronic disease presence output



Happy to say that you don't have CHRONIC DISEASE

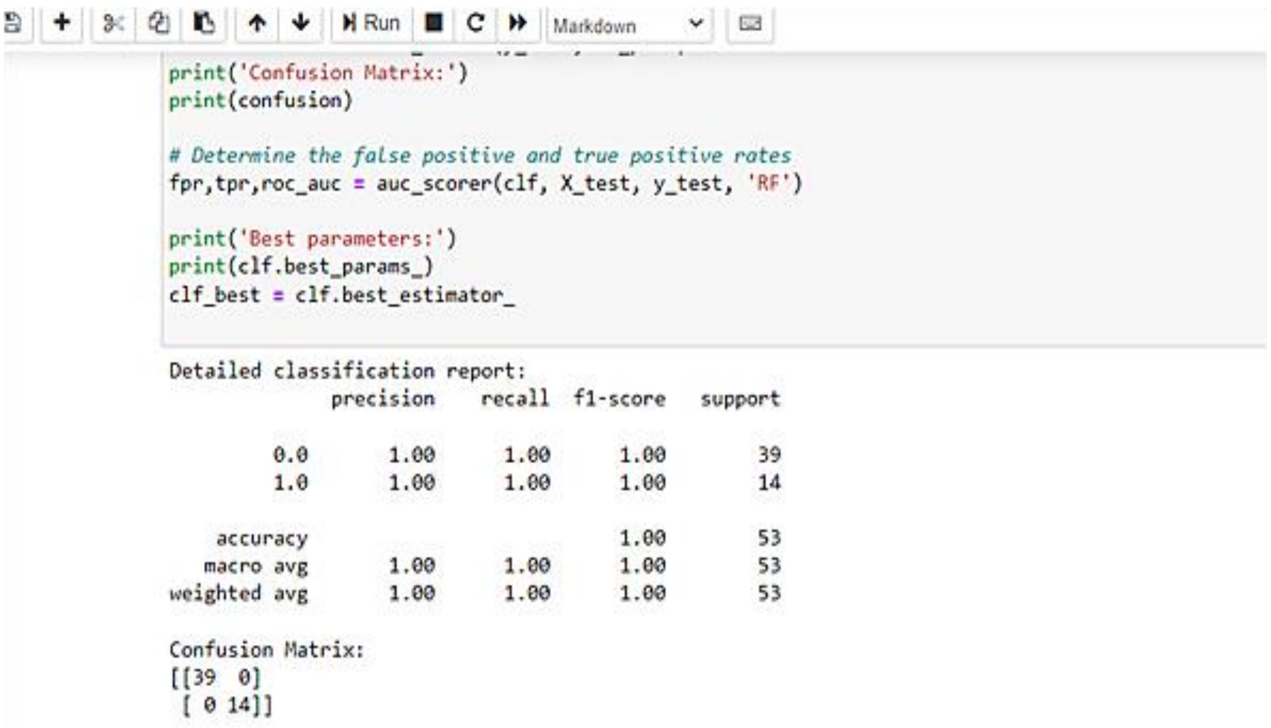
[0.]

Figure 8.5 Chronic disease not presence output

CHAPTER 9

RESULTS

9.1 Performance Metrics



```
print('Confusion Matrix:')
print(confusion)

# Determine the false positive and true positive rates
fpr,tpr,roc_auc = auc_scorer(clf, X_test, y_test, 'RF')

print('Best parameters:')
print(clf.best_params_)
clf_best = clf.best_estimator_
```

```
Detailed classification report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        39
         1.0       1.00      1.00      1.00        14

    accuracy                           1.00        53
   macro avg       1.00      1.00      1.00        53
weighted avg       1.00      1.00      1.00        53

Confusion Matrix:
[[39  0]
 [ 0 14]]
```

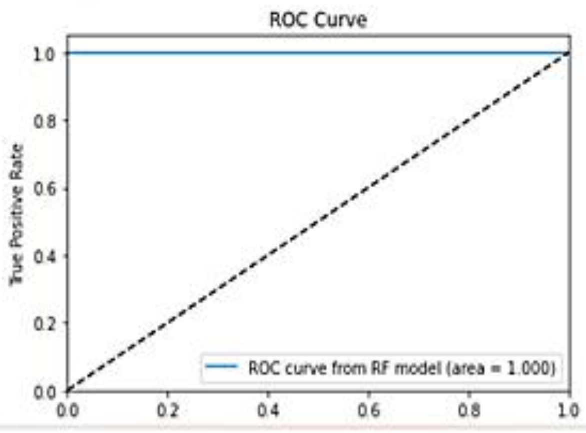


Figure 9.1 Performance Metrics

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

Predictive modeling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data. It works by analyzing current and historical data and projecting what it learns on a model generated to forecast likely outcomes.We found that machine learning can predict the occurrence of individual chronic diseases, progression, and their determinants and in many contexts. The findings are original and relevant to improve clinical decisions and the organization of health care facilities.

DISADVANTAGES

In Chronic Disease prediction, for classification problem we get a very good accuracy but for regression, we get considerable error rate. So, we need to add some more data or change the machine algorithm or by using deep learning techniques for reducing the error in predicting the probability values.

CHAPTER 11

CONCLUSION

The principal part of this work is to make an effective diagnosis system for chronic disease of patients.The application will have the option to predict chronic disease prior and advise the wellbeing condition. This application can be surprisingly gainful in low-salary nations where our absence of medicinal foundations and just as particular specialists. In our study, there are a few bearings for future work in this field. We just explored some popular supervised machine learning algorithms, more algorithms can be picked to assemble an increasingly precise model of chronic kidney disease prediction and performance can be progressively improved. Additionally, this work likewise ready to assume a significant role in health care research and just as restorative focuses to anticipate chronic disease.

CHAPTER 12

FUTURE SCOPE

Diseases related to kidney is becoming more and more common with time. With continuous technological advancements, these are only going to increase in the future. Although people are becoming more conscious of health nowadays and are joining yoga classes, dance classes; still the sedentary lifestyle and luxuries that are continuously being introduced and enhanced; the problem is going to last long. So, in such a scenario, our project will be extremely helpful to the society. With the dataset that we used for this project, we got 89% accuracy for Random forest model, and though it might be difficult to get such accuracies with very large datasets, from this projects results, one can clearly conclude that we can predict the risk of chronic diseases with accuracy of 95 % or more. Also it can be incorporated into a wide range commercial website and these app and website will be highly beneficial for a large section of society.

CHAPTER 13

APPENDIX

SOURCE CODE

Chronic disease Prediction.ipynb

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import roc_curve, auc, confusion_matrix, classification_report,accuracy_score
from sklearn.ensemble import RandomForestClassifier
import warnings warnings.filterwarnings('ignore') %matplotlib inline

def auc_scorer(clf, X, y, model): # Helper function to plot the ROC curve
    if model=='RF':
        fpr, tpr, _ = roc_curve(y, clf.predict_proba(X)[:,1])
    elif model=='SVM':
        fpr, tpr, _ = roc_curve(y, clf.decision_function(X))
    roc_auc = auc(fpr, tpr)

    plt.figure()    # Plot the ROC curve
    plt.plot(fpr, tpr, label='ROC curve from '+model+' model (area = %0.3f)' % roc_auc)
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('ROC Curve')
    plt.legend(loc="lower right")
    plt.show()

    return fpr,tpr,roc_auc

df = pd.read_csv("C:/Users/Sinegalatha/Desktop/2nd year online class/nalaiya
thiran/dataset/kidney_disease.csv")

df.head()

df['wc']

df.info()

df.describe()

df[df.duplicated()]

# Map text to 1/0 and do some cleaning
df[['htn','dm','cad','pe','ane']] = df[['htn','dm','cad','pe','ane']].replace(to_replace={'yes':1,'no':0})
df[['rbc','pc']] = df[['rbc','pc']].replace(to_replace={'abnormal':1,'normal':0})
df[['pcc','ba']] = df[['pcc','ba']].replace(to_replace={'present':1,'notpresent':0})
df[['appet']] = df[['appet']].replace(to_replace={'good':1,'poor':0,'no':np.nan})
df['classification'] = df['classification'].replace(to_replace={'ckd':1.0,'ckd\t':1.0,'notckd':0.0,'no':0.0})
df.rename(columns={'classification':'class'},inplace=True)

# Further cleaning
df['pe'] = df['pe'].replace(to_replace='good',value=0) # Not having pedal edema is good
df['appet'] = df['appet'].replace(to_replace='no',value=0)
df['cad'] = df['cad'].replace(to_replace='\tno',value=0)
df['dm'] = df['dm'].replace(to_replace={'\tno':0,'\tyes':1,' yes':1, '':np.nan})
df.drop('id',axis=1,inplace=True)

df.head()

df2 = df.dropna(axis=0)
df2['class'].value_counts()

corr_df = df2.corr()


# Generate a mask for the upper triangle
mask = np.zeros_like(corr_df, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True


# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))


# Generate a custom diverging colormap
```

```
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heat map with the mask and correct aspect ratio
sns.heatmap(corr_df, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.title('Correlations between different predictors')
plt.show()

X_train, X_test, y_train, y_test = train_test_split(df2.iloc[:,:-1], df2['class'],
                                                    test_size = 0.33, random_state=44,
                                                   stratify= df2['class'] )

X_train.head()

print(X_train.shape)
print(X_test.shape)

y_train.value_counts()

tuned_parameters = [{'n_estimators':[7,8,9,10,11,12,13,14,15,16],'max_depth':[2,3,4,5,6,None],
                    'class_weight':[None,{0: 0.33,1:0.67},'balanced'],'random_state':[42]}]
clf = GridSearchCV(RandomForestClassifier(), tuned_parameters, cv=10,scoring='f1')
clf.fit(X_train, y_train)

print("Detailed classification report:")
y_true, lr_pred = y_test, clf.predict(X_test)
print(classification_report(y_true, lr_pred))

confusion = confusion_matrix(y_test, lr_pred)
print('Confusion Matrix:')
print(confusion)

# Determine the false positive and true positive rates
fpr,tpr,roc_auc = auc_scorer(clf, X_test, y_test, 'RF')

print('Best parameters:')
print(clf.best_params_)
clf_best = clf.best_estimator_

plt.figure(figsize=(12,3))
```

```
features = X_test.columns.values.tolist()
importance = clf_best.feature_importances_.tolist()
feature_series = pd.Series(data=importance,index=features)
feature_series.plot.bar()
plt.title('Feature Importance')

list_to_fill = X_test.columns[feature_series>0]
print(list_to_fill)

# Are there correlation in missing values?
corr_df = pd.isnull(df).corr()

# Generate a mask for the upper triangle
mask = np.zeros_like(corr_df, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# Generate a custom diverging color map
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heat map with the mask and correct aspect ratio
sns.heatmap(corr_df, mask=mask, cmap=cmap, vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.show()


df2 = df.dropna(axis=0)
no_na = df2.index.tolist()
some_na = df.drop(no_na).apply(lambda x: pd.to_numeric(x,errors='coerce'))
some_na = some_na.fillna(0) # Fill up all Nan by zero.

X_test = some_na.iloc[:,:-1]
y_test = some_na['class']
y_true = y_test
lr_pred = clf_best.predict(X_test)
```

```
print(classification_report(y_true, lr_pred))


confusion = confusion_matrix(y_test, lr_pred)
print('Confusion Matrix:')
print(confusion)
print('Accuracy: %3f' % accuracy_score(y_true, lr_pred))
# Determine the false positive and true positive rates
fpr,tpr,roc_auc = auc_scorer(clf_best, X_test, y_test, 'RF')
import pickle
pickle. dump(clf_best, open('randomclass_chronic', 'wb'))
from sklearn.ensemble import RandomForestRegressor
reg=RandomForestRegressor()
reg.fit(X_train,y_train)
y_pred=reg.predict(X_test)

pickle. dump(reg, open('randomreg_chronic', 'wb'))
y_pred
l_pred=list(y_pred)


l_test=list(y_test)


d={'prob':l_pred,'out':y_test}
df_i=pd.DataFrame(d)
df_i.head()
df_i.to_csv('C:/Users/Sinegalatha/Desktop/2nd year online class/nalaiya thiran/output/file1.csv')
```

Flask web app.ipynb

```
import pickle
loaded_class = pickle. load(open('randomclass_chronic', 'rb'))
loaded_reg = pickle. load(open('randomreg_chronic', 'rb'))

import numpy as np

import pandas as pd
```

```
from flask import Flask, request, redirect, render_template
app = Flask(_name_)
@app.route("/",methods=['GET', 'POST'])
def index():
   return render_template('index.html')
@app.route("/val",methods=['POST'])
def val():
   test=[]

if request.method == 'POST':
      test.append(request.form.get("age"))
      test.append(request.form.get("bp"))
      test.append(request.form.get("sg"))
      test.append(request.form.get("al"))
      test.append(request.form.get("su"))
      rb=request.form.get("rbc")
      if rb=='abnormal':
         test.append(1)
      else:
         test.append(0)
      pc=request.form.get("pc")
      if pc=='abnormal':
         test.append(1)
      else:
         test.append(0)
      pcc=request.form.get("pcc")
      if pcc=='present':
         test.append(1)
      else:
         test.append(0)
      ba=request.form.get("ba")
      if ba=='present':
         test.append(1)
```

```
else:
    test.append(0)
    test.append(request.form.get("bgr"))
    test.append(request.form.get("bu"))
    test.append(request.form.get("sc"))
    test.append(request.form.get("sod"))
    test.append(request.form.get("pot"))
    test.append(request.form.get("hemo"))
    test.append(request.form.get("pcv"))
    test.append(request.form.get("wc"))
    test.append(request.form.get("rc"))
    ht=request.form.get("htn")
if ht=='yes':
    test.append(1)
else:
    test.append(0)
d=request.form.get("dm")
if d=='yes':
    test.append(1)
else:
    test.append(0)
ca=request.form.get("cad")
if ca=='yes':
    test.append(1)
else:
    test.append(0)
ap=request.form.get("appet")
if ap=='good':
    test.append(1)
elif ap=='poor':
    test.append(0)
```

```
else:
            test.append(np.nan)
        p=request.form.get("pe")
        if p=='yes':
            test.append(1)
        else:
            test.append(0)
        an=request.form.get("ane")
        if an=='yes':
            test.append(1)
        else:
            test.append(0)
    print(test)
    test_df=pd.DataFrame(test)
    test_df=np.array(test_df).reshape(1, -1)

    ans1=loaded_class.predict(test_df)
    ans2=loaded_reg.predict(test_df)
    if int(ans1)==1:
        answer1="Sorry to say!! You have CHRONIC DISEASE!!!"
        return render_template('rename.html',answer1=answer1,answer2=ans2)
    else:
        answer1="Happy to say that you don't have CHRONIC DISEASE"

        return render_template('rename2.html',answer1=answer1,answer2=ans2)

if__name__== "_main_":
    app.debug=True
    app.run(debug=False)
```

index.html

```
<html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
                <title>Flaskimio</title>
        <style>
        body {
        background: linear-gradient(
                        rgba(20,20,20, .75),
                        rgba(20,20,20, .75)),
                        url( 'https://pharmanewsintel.com/images/site/article_headers/_normal/Chronic_Disease
        _Manage.png');
        .container {
          border: 2px solid #ccc;
          padding: 10px;
          width: 20em;
        height:21em;
        background-color:white;
        }


        .hello{
        opacity: 0.5;
        }
        </style>
        </head>

        <body>
        <marquee bgcolor="white">IBM NALAIYA THIRAN</marquee>
        <center><p style="font-size:60px;color:white;">Chronic Disease Prediction</p></center>
        <form action="/val" method="post"><center>
          <label for="age">Age:</label><br>
          <input type="number" id="age" name="age"><br><br><br>


          <label for="bp">Blood Pressure:</label><br>
          <input type="number" id="bp" name="bp">
        <br>
```

```
<br>
<br>
  <label for="sg">Urinary Specific Gravity:</label><br>
  <input type="number" id="sg" name="sg">
<br>
<br>
<br>
  <label for="al">al:</label><br>
  <input type="number" id="al" name="al">
<br>
<br>
<br>
  <label for="su">su:</label><br>
  <input type="number" id="su" name="su">
<br>
<br>
<br>
  <label for="rbc">rbc:</label><br>
  <input type="text" id="rbc" name="rbc">
<br>
<br>
<br>
  <label for="pc">pc:</label><br>
  <input type="text" id="pc" name="pc">
<br>
<br>
<br>
  <label for="pcc">pcc:</label><br>
  <input type="text" id="pcc" name="pcc">
<br>
<br>
<br>
  <label for="ba">ba:</label><br>
  <input type="text" id="ba" name="ba">
<br>
<br>
<br>
  <label for="bgr">bgr:</label><br>
  <input type="number" id="bgr" name="bgr">
<br>
```

```
<br>
<br>
  <label for="bu">bu:</label><br>
  <input type="number" id="bu" name="bu">
<br>
<br>
<br>
  <label for="sc">sc:</label><br>
  <input type="number" id="sc" name="sc">
<br>
<br><br>
  <label for="sod">sod:</label><br>
  <input type="number" id="sod" name="sod">
<br>
<br>
<br>
  <label for="pot">pot:</label><br>
  <input type="number" id="pot" name="pot">
<br>
<br>
<br>
  <label for="hemo">hemo:</label><br>
  <input type="number" id="hemo" name="hemo">
<br>
<br>
<br>
  <label for="pcv">pcv:</label><br>
  <input type="text" id="pcv" name="pcv">
<br>
<br>
<br>
  <label for="wc">wc:</label><br>
  <input type="text" id="wc" name="wc">
<br>
<br>
<br>
  <label for="rc">rc:</label><br>
  <input type="text" id="rc" name="rc">
<br>
<br>
```

```
<br>
  <label for="htn">htn:</label><br>
  <input type="text" id="htn" name="htn">
<br>
<br>
<br>
  <label for="dm">dm:</label><br>
  <input type="text" id="dm" name="dm">
<br>
<br>
<br>
  <label for="cad">cad:</label><br>
  <input type="text" id="cad" name="cad">
<br>
<br>
<br>
  <label for="appet">appet:</label><br>
  <input type="text" id="appet" name="appet">
<br>
<br>
<br>
  <label for="pe">pe:</label><br>
  <input type="text" id="pe" name="pe">
<br>
<br>
<br>
  <label for="ane">ane:</label><br>
  <input type="text" id="ane" name="ane">
<br>
<br></center>
  <center><button type="submit">Check</button></center>
</form>
</body>
</html>
```

rename.html

```
<html>
              <head>
              <style>
              body {
                background-color: #E6E6FA;
              }
              </style>
              </head>
              <body >
              <br>
              <br>
              <br>
              <center><h1>{{answer1}}</h1></center>
              <br>
              <center><h1>{{answer2}}</h1></center>
              <center><img alt="Qries" src="https://image.shutterstock.com/
              image-vector/cute-illustration-sick-kidney-characters-260nw-1529381825.jpg"
                     width=500" height="500"></center>

              </body>
              </html>
```

rename2.html

```
<html>
            <head>
            <style>
            body {
              background-color: #E6E6FA;
            }
            </style>
            </head>
            <body >
            <br>
            <br>
            <br>
            <center><h1>{{answer1}}</h1></center>
            <br>
            <center><h1>{{answer2}}</h1></center>
```

```
<center><img alt="Qries" src="https://media.istockphoto.com/vectors/vector-kidney-
cartoon-human-body-health-organ-smiling-mascot-on-vector-id1136533246?k=20&m=1136
533246&s=612x612&w=0&h=eWjhtoVOfX3lBF4fcsDD4ZLLzjeJ_Ax4cgdgdUexG1Q="
        width=500" height="500"></center>
</body>
</html>
```

GITHUB & PROJECT DEMO LINK

1) GITHUB LINK : https://github.com/IBM-EPBL/IBM-Project-54572-1662303014
2) DEMO LINK : https://drive.google.com/file/d/1oJNeGNOs0TC3pQ927sFTGrlK47_AjfcO/view