Early Detection of Chronic Kidney Disease using Machine Learning

1. INTRODUCTION

Chronic kidney disease occurs due to improper functioning of kidneys. Diabetes, high blood pressure and several heart disease problems which causes chronic kidney disease are taken as primary features. Detection of chronic kidney disease in early stage is a challenging task because there are no early-stage symptoms to be seen, so when kidney disease worsens, we could see more waste built up in your blood which may result in many complications like nerve damage, high blood pressure and anemia. Diabetes and high blood pressure are responsible for two-thirds of chronic kidney disease. Chronic kidney disease could affect any human being irrespective of age. Most importantly person with a family history of kidney failure is more prone to this disease. At early stage of chronic kidney disease there won't be severe symptoms at start later on we could notice trouble sleeping, swollen feet and urinate more often especially at night. Chronic kidney disease gets worse over time which can progress to kidney disease. Often this disease could be predicted in later stage. Medical researchers use Urine and blood sample to detect the chronic kidney disease at much earlier rate.

1.1 Project Overview

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if treated in the early stages. Usually, people are not aware that medical tests we take for different purposes could contain valuable information concerning kidney diseases. Consequently, attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease. The information says that it helps us to measure the severity of the problem and we make

use of such information to build a machine learning model that predicts chronic kidney disease.

1.2 Purpose

Early detection and cure of CKD is extremely desirable as it can lead to the prevention of unwanted consequences. Machine learning methods are being extensively advocated for early detection of symptoms and diagnosis of several diseases recently. With the same motivation, the aim of this study is to predict the various stages of CKD using machine learning classification algorithms on the dataset obtained from the medical records of affected people. Specifically, we have used the Random Forest and J48 algorithms to obtain a sustainable and practicable model to detect various stages of CKD with comprehensive medical accuracy.

2. LITERATURE SURVEY

2.1 Existing problem

Gunarathne W.H.S.D et.al. [1] Has compared results of different models. and finally they concluded that the Multiclass Decision forest algorithm gives more accuracy than other algorithms which is around 99% for the reduced dataset of 14 attributes. S Ramya and Dr N Radha [2] worked on diagnosis time and improvement of diagnosis accuracy using different classification algorithms of machine learning. The proposed work deals with classification of different stages of CKD according to its gravity. By analyzing different algorithms like Basic Propagation Neural Network, RBF and RF. The analysis results indicates that RBF algorithm gives better results than the other classifiers and produces 85.3% accuracy. S Dilli Arasu and Dr R Thirumalai Selvi has worked on missing values in a dataset of chronic kidney disease. Missing values in dataset will reduce the accuracy of our model as well as prediction results. They find solution over this problem that they performed a recalculation process on CKD stages and by doing so they got up with unknown values. They replaced missing values with recalculated values.as if Salekin and john Stankovic they use novel approach to detect CKD using machine learning algorithm. They get result on dataset which having 400 records and 25 attributes which gives result of patient having CKD or not CKD. They use k-nearest neighbors, random forest and neural network to get results. For feature reduction they use wrapper method which detect CKD with high accuracy. Pinar Yildirim searches the effect of class imbalance when we train the data by using development of neural network algorithm for making medical decision on chronic kidney disease. In this proposed work, a comparative study was performed using sampling algorithm. This study reveals that the performance of classification algorithms can be improved by using the sampling algorithms. It also

reveals that the learning rate is a crucial parameter which significantly effect on multilayer perceptron. Sahil Sharma, Vinod Sharma, and Atul Sharma, has assessed 12 different classification Algorithm on dataset which having 400 records and 24 attributes. They had compared their calculated results with actual results for calculating the accuracy of prediction results. They used assessment metrics like accuracy, sensitivity, precision and specificity. They find that the decision tree technique gives accuracy up to 98.6%, sensitivity of 0.9720, and precision of 1 and specificity of 1.

2.2 References

Gunarathne W H S D Perera K D M Kahandawaarachchi K A D C P, "Performance Evaluation on Machine Learning Classification Techniques for Disease Classification and Forecasting through Data Analytics for Chronic Kidney Disease (CKD)",2017 IEEE 17th International Conference on Bioinformatics and Bioengineering.

S Ramya, Dr. N Radha, "Diagnosis of Chronic Kidney Disease Using Machine Learning Algorithms," Proc. International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 1, January 2016.

S Dilli Arasu and Dr. R Thirumalai Selvi, "Review of Chronic Kidney Disease based on Data Mining Techniques", International Journal of Applied Engineering Research ISSN 0973- 4562 Volume 12, Number 23 (2017) pp. 13498-13505

L. Rubini, "Early stage of chronic kidney disease UCI machine learning repository", 2015.

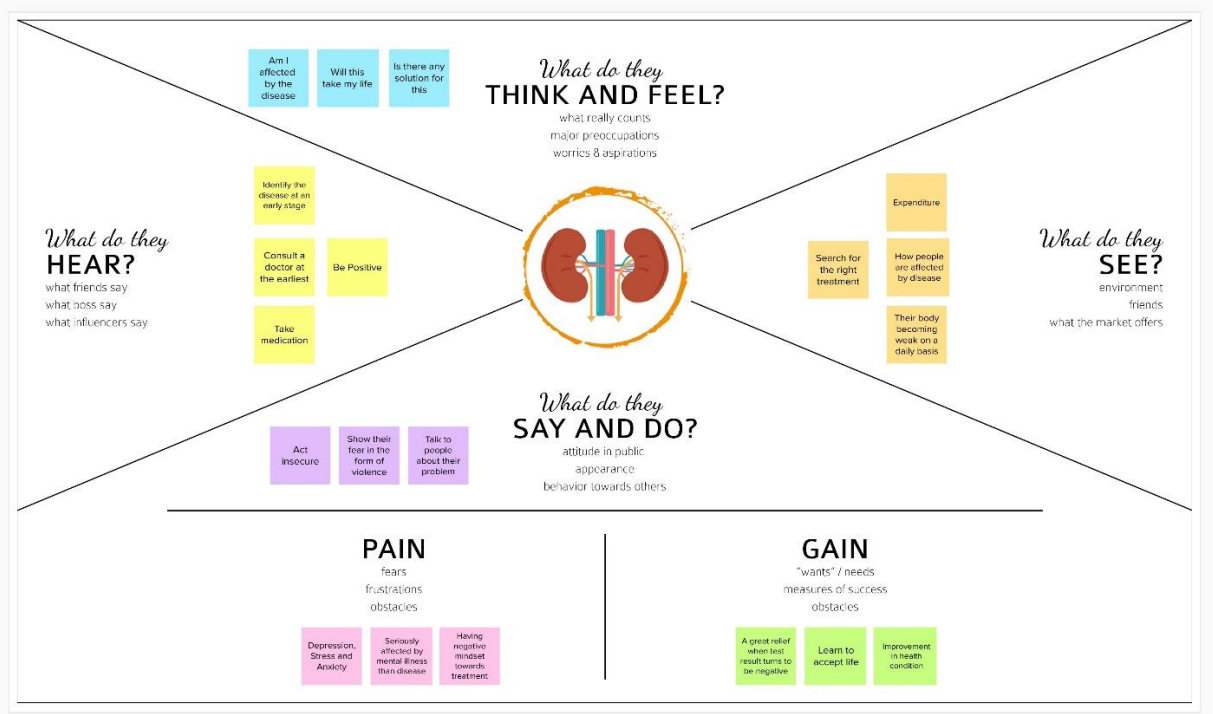
2.3 Problem Statement Definition

Chronic kidney disease occurs due to improper functioning of kidneys. Diabetes, high blood pressure and several heart disease problems which causes chronic kidney disease are taken as primary features. Detection of chronic kidney disease in early stage is a challenging task because there are no early-stage symptoms to be seen, so when kidney disease worsens, we could see more waste built up in your blood which may result in many complications like nerve damage, high blood pressure and aneamia. Diabetes and high blood pressure are responsible for two-thirds of chronic kidney disease. Chronic kidney disease could affect any human being irrespective of

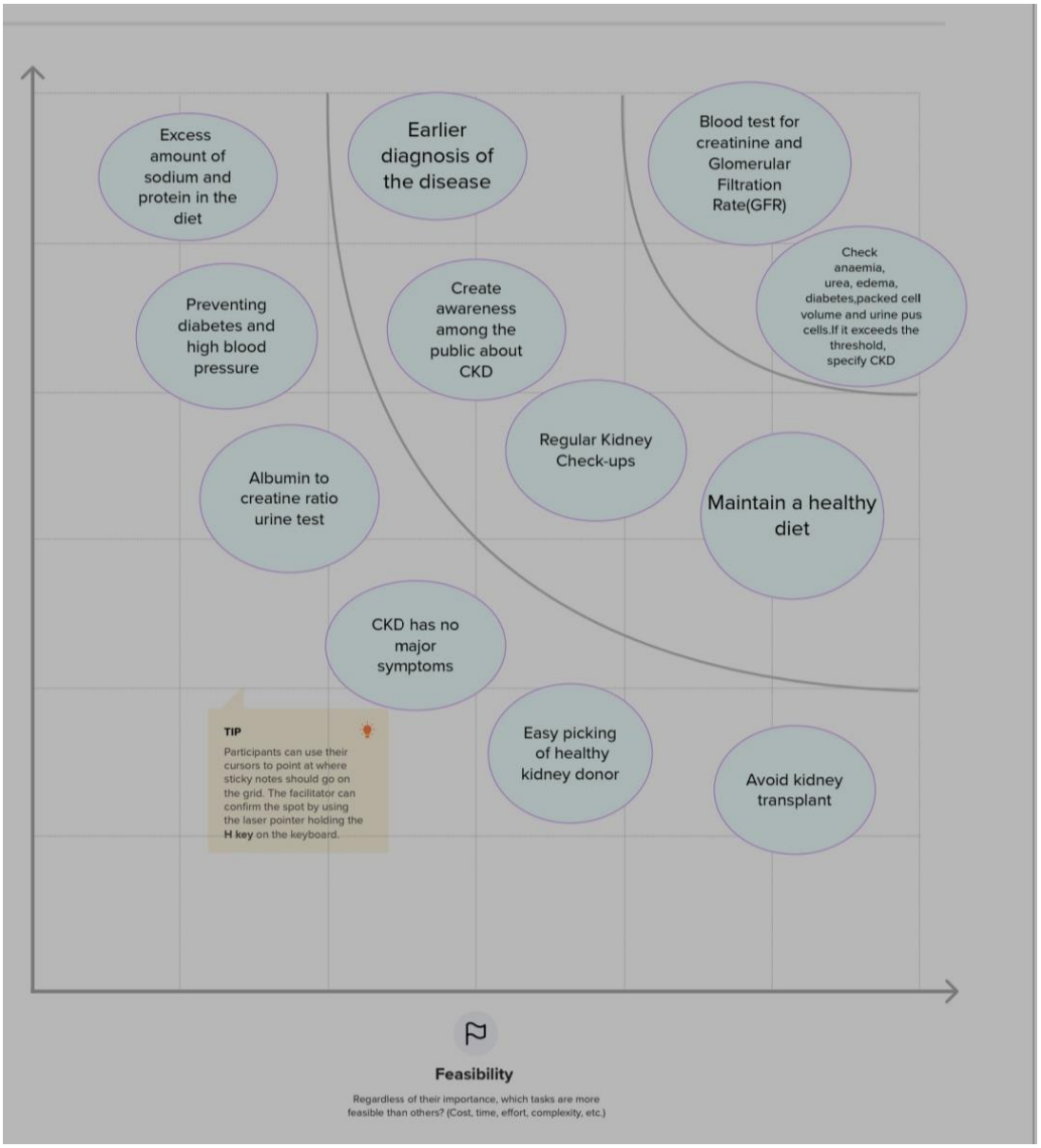
age. Most importantly person with a family history of kidney failure is more prone to this disease. At early stage of chronic kidney disease there won't be severe symptoms at start later on we could notice trouble sleeping, swollen feet and urinate more often especially at night. Chronic kidney disease gets worse over time which can progress to kidney disease. Often this disease could be predicted in later stage. Medical researchers use Urine and blood sample to detect the chronic kidney disease at much earlier rate.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Brainstorm and Ideation



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!

NITHISH KUMAR

- None of the ML Models provide 100% accuracy. So, we to choose the one which best suits the dataset
- Aged people should have regular kidney check-ups
- Statistically analyzing the collected data using visualization tools to get more accurate results
- Avoid doing Kidney Transplant as its a burden for both donor and user

MOHAMMED FAZIL

- Diabetes and high blood pressure are responsible for two-thirds of CKD cases
- To detect CKD, we need Albumin to creatine ratio urine test, Blood test for creatinine, Glomerular Filtration Rate(GFR)
- Restricting the amount of protein in the diet may help slow down the progression of CKD
- By limiting sodium, everyone can help lower their risk for developing CKD

SANJAI

- Create awareness among the public about CKD
- Traditional diagnostic tools are believed by everyone, so no one will have a fear of accuracy on that
- Check anaemia, urea, edema, albumin, packed cell volume, diabetes and urine pus cells. If it exceeds the threshold, specify CKD
- Stay Hydrated!!

MARZOUK

- Interview and fnd chronic kidney disease thereby judging the remaining people having CKD
- Drink specifed amount of water and then check urine output
- Diet is the key to healthy life. Always follow a regular diet
- Check anaemia, urea, edema, albumin, diabetes and urine pus cells. If it exceeds the threshold, specify CKD

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

DETECTION

- Check anaemia, urea, edema, albumin,packed cell volume, diabetes and urine pus cells. If it exceeds the threshold, then specify CKD
- Albumin to creatine ratio urine test
- Blood test for creatinine and Glomerular Filtration Rate(GFR)

TIP

Add customizable t
notes to make it ea
browse, organize, a
categorize importa
themes within your

CAUSES

- Diabetes and high Blood pressure are the major cause
- Abnormal properties of urine may result in CKD
- Excess amount of sodium and protein in the diet

PRECAUTION..

- Maintaining a healthy diet
- Exercise regularely
- Regular kidney checkup

PREVENTIVE MEASURES

- Earlier diagnosis of the disease
- Drink atlest 3litre water everyday
- Check urine salt everyday

ADVANTAGE

- Avoids doing kidney transplant
- Easy picking of healthy kidney donor

DRAWBACK

- No ML technique has 100 percent accuracy
- CKD has no major symptoms

Project Design Phase-I

Proposed solution

S.No	Parameter	Description
1.	Problem statement (problem to be solved)	Our Aim is to predict patients with Chronic Kidney Failure (CKD) disease and patients who do not(not CKD).
2.	Idea/Solution Description	We are building a Machine Learning model to predict the compressive strength of concrete.
3.	Novelty/Uniqueness	With the measure of Glomerular Filtration Rate Early Detection of kidney failure is accurate. Deep neural networks have been proposed to detect and diagnose CKD.
4.	Social impact /customer satisfaction	Unlike MRI which uses radiation, we detect the people with CKD through blood and urine tests itself.
5.	Business model (revenue model)	It is cost efficiency and also provides best results.
6.	Scalability of the solution	This model can be expanded to include more attributes for more accurate Detection. Training the model with more attributes will increase the efficiency further.

Early Detection of Chronic Kidney Disease using Machine Learning

Project Design Phase-I - Solution Fit Template

Team id:PNT2022TMID32437

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

People with CKD

6. CUSTOMER CONSTRAINTS CC

Patients are afraid of using new technology.
Budget and Unaware about the symptoms.

5. AVAILABLE SOLUTIONS AS

Never intake Alcohol.
Reduce salt intake.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Problem related to identifying the chronic kidney disease.
Accuracy of patients test result.
Time taken to produce test result.

9. PROBLEM ROOT CAUSE RC

The root cause of the problem is inaccurate result.
The test takes much time to evaluate the result.

.

7. BEHAVIOUR BE

They blindly trust the inaccurate result and become anxious and sad.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

The Dilemma and confusion whether they really have chronic disease or not.

10. YOUR SOLUTION SL

Predict faster and accurately. Time and cost of the test is drastically reduced. Helps to take treatment at right time.

8. CHANNELS of BEHAVIOUR CH

Online: Aware of symptoms of chronic kidney failure.

Offline: people must take the treatment once CKD is detected

Project Design Phase-II
Requirement Analysis (Functional & Non-functional)

Functional Requirements:

Following are the functional requirements of the proposed solution.

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.

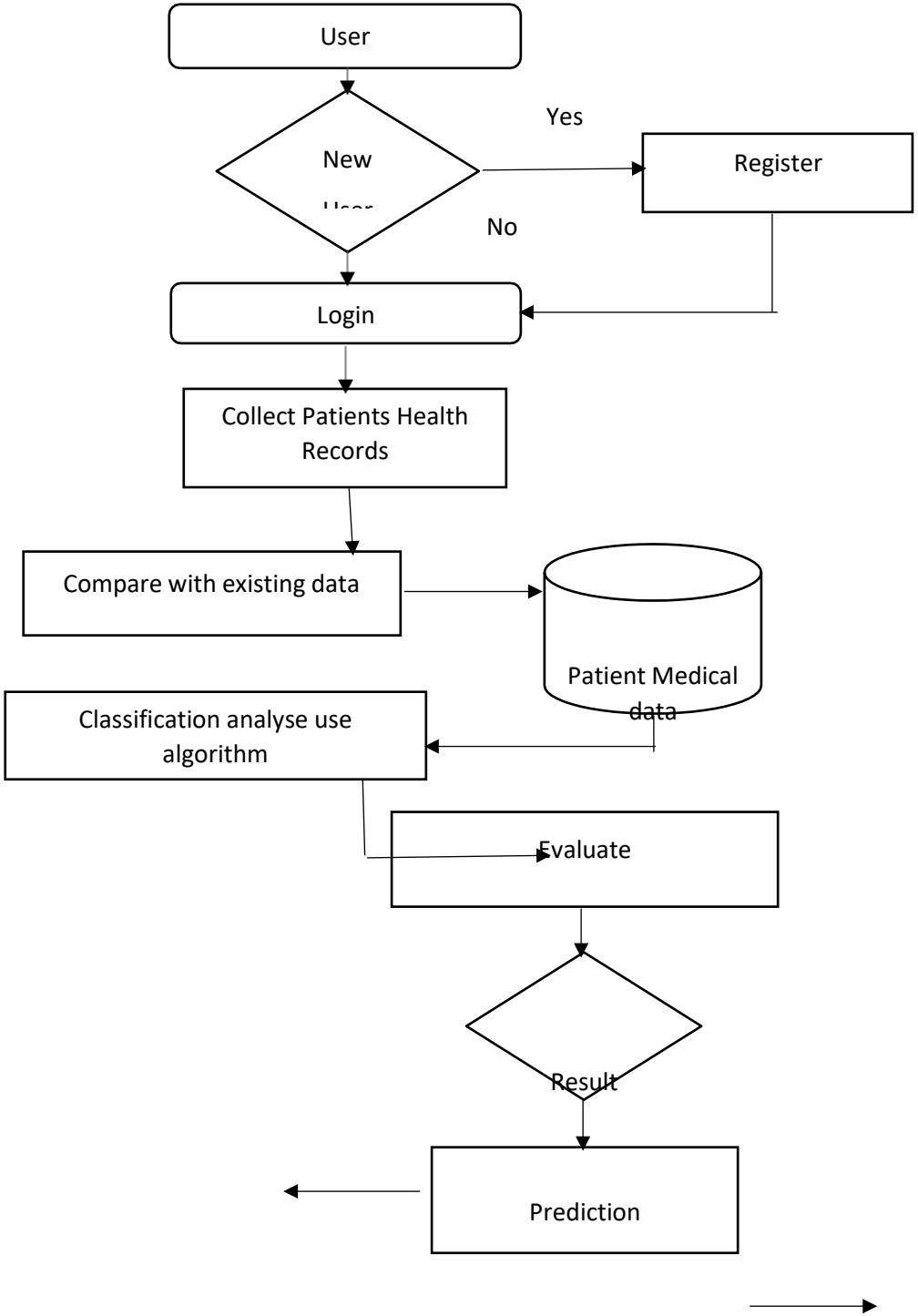
PROJECT DESIGN PHASE- II

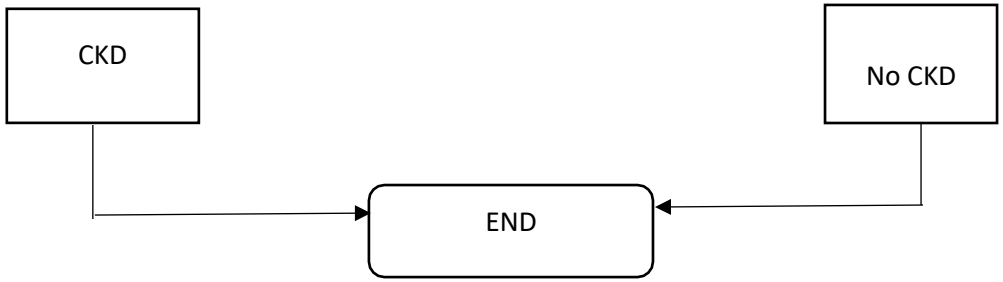
DATA FLOW DIAGRAM & USER STORIES

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

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 1 hour.
NFR-6	**Scalability**	The website Traffic limit must be scalable enough to support 2000,000 users at a time.


User
New
Yes
Register
No
Login
Collect Patients Health Records
Compare with existing data
Patient Medical data
Classification analyse use algorithm
Evaluate
Result
Prediction



Yes **No**

Technical Architecture

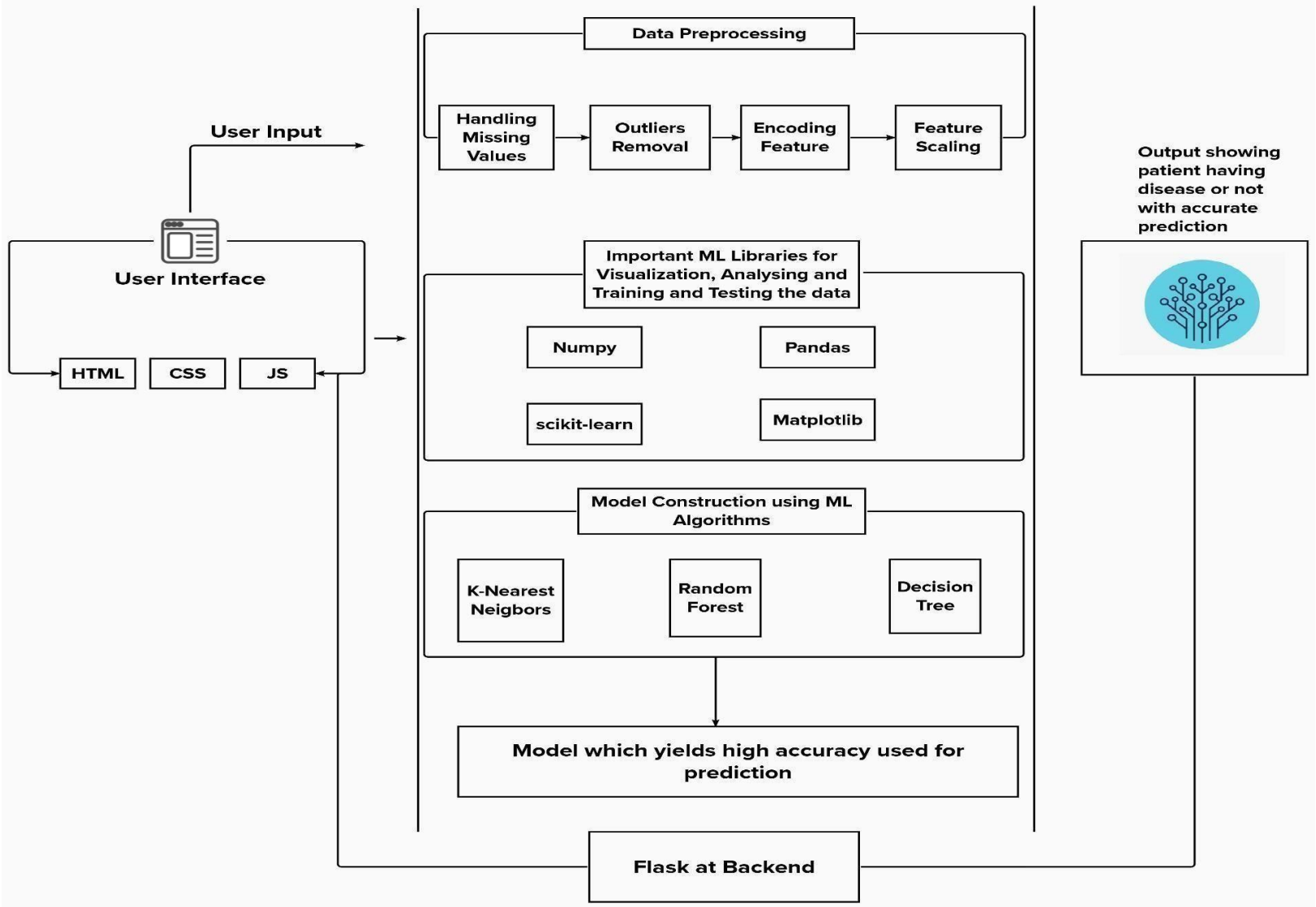


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. WebUI	HTML, CSS, Javascript
2.	Application Logic-1	Clicking on the register button will direct to registration page where they will enter their detailsto get registered	HTML, CSS, Flask
3.	Application Logic-2	Clicking on the login button will direct to login page where they can enter the login credentials if registered already	HTML, CSS, Flask
4.	Application Logic-3	After successful login, the form will get the vitaldetails from the user for predicting the disease	HTML, CSS, Flask
5.	Data	Data type	Comma Separated Value File(.CSV)
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	Local Filesystem
8.	External API-1	Purpose of External API used in the application	NIL
9.	External API-2	Purpose of External API used in the application	NIL
10.	Machine Learning Model	Model is developed to find the patterns or make decisions. It will predict whether the user has disease or not	Supervised Machine Learning Algorithms
11.	Infrastructure (Server / Cloud)	Cloud Deployment	IBM Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks used for data pre-processing, application development and deployment	Visual Studio Code, Anaconda Navigatorand IBM Cloud etc.
2.	Security Implementations	User profile and test result details will be secured	Encryptions and OWASP
3.	Scalable Architecture	Accurate details will be displayed	Supervised Machine Learning Algorithms such as Random Forest Classifier, K-Nearest Neighbor and Decision Tree etc.
4.	Availability	Available at any cost of time	IBM Load Balancer
5.	Performance	User will be able to know in-depth information aboutthe disease and its severity. Capable of performing faster classification	Supervised Machine Learning Algorithms such as Random Forest Classifier, K-Nearest Neighbor and Decision Tree etc.

Project Planning Phase Milestone and Activity List

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers ,research publications etc.	28 OCTOBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	17 OCTOBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	17 OCTOBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	17 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	17 OCTOBER 2022

Solution Architecture	Prepare solution architecture document.	17 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	17 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	17 OCTOBER 2022
Technology Architecture	Architecture diagram.	17 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	28 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	New user enters into the System. He/ Shecan register into the Application by entering user details such as username and mobile number	2	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-2	User Verification	USN-2	The user will receive OTP through SMS.	3	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-1	Login	USN-3	After Successful registration the user canLog into the application by entering the registered Username and Password	2	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		USN-4	CAPTCHA will be provided to reduce the network traffic.	2	Medium	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-2	Dashboard	USN-5	User can get into the Dashboard only when the Verification Successful. Afterthe user can access the displayed information in the Dashboard	3	Medium	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-3	Data collection	USN-6	Diagnosed result data will be entered bythe user.	2	Medium	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-4	Prediction result	USN-7	By the collected data the trained model will predict and display the result.	2	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-4		USN-8	Based on the result the suggestion varies.	2	Low	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-1	Dataset Collection	USN-9	Chronic Kidney Disease dataset identification	2	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Clean the Dataset	USN-10	The dataset had to be cleaned. Cleaning process includes removing null values, Replacing missing values, segregation of testand train data.	3	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-2	Train ML Model inIBM	USN-11	The model will be trained in IBM.	4	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-3	Model Testing	USN-12	The model will be tested using the test data	3	High	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-3	Integration	USN-14	HTML file and python Code Integration	2	Medium	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-4	Deployment	USN-15	The model will be deployed in Cloud	3	Medium	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI
Sprint-4	Further Clarification	USN-16	The problems which are faced by the user while using the application can be clarified	2	Medium	G NITHISH KUMAR M MOHAMMED FAZIL K MARZOUK S SANJAI

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	7	6 Days	07 Nov 2022	12 Nov 2022	7	07 Nov 2022
Sprint-4	9	6 Days	14 Nov 2022	19 Nov 2022	9	14 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Sprint 1 AV = Sprint duration/velocity =

11/6 = 1.83Sprint 2 AV = Sprint

duration/velocity = 10/6 = 1.67Sprint 3

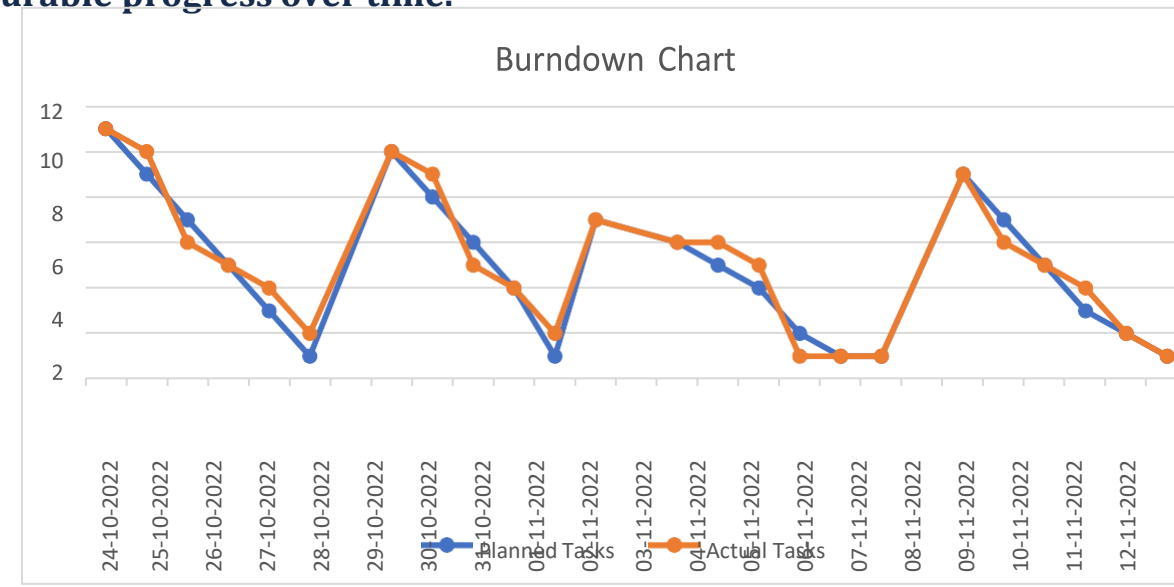
AV = Sprint duration/velocity = 7/6 =

1.16 Sprint 4 AV = Sprint

duration/velocity = 9/6 = 1.5

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CSS Coding:

```
@import url(https://fonts.googleapis.com/css?family=Open+Sans);
.btn { display: inline-block; *display: inline; *zoom: 1; padding: 4px 10px 4px; margin-bottom: 0; font-size: 13px; line-height: 18px; color: #333333; text-align: center;text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75); vertical-align: middle; background-color: #f5f5f5; background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6); background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6); background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6)); background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6); background-image: -o-linear-gradient(top, #ffffff, #e6e6e6); background-image: linear-gradient(top, #ffffff, #e6e6e6); background-repeat: repeat-x; filter: progid:dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0); border-color: #e6e6e6 #e6e6e6 #e6e6e6; border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25); border: 1px solid
```

```
#e6e6e6; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px; -
webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
-moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
cursor: pointer; *margin-left: .3em; }

.btn:hover, .btn:active, .btn.active, .btn.disabled, .btn[disabled] { background-color:
#e6e6e6; }

.btn-large { padding: 20px 0px; font-size: 20px; line-height: normal; -webkit-border-
radius: 5px; -moz-border-radius: 5px; border-radius: 5px; }

.btn:hover { color: #333333; text-decoration: none; background-color: #e6e6e6;
background-position: 0 -15px; -webkit-transition: background-position 0.1s linear; -moz-
transition: background-position 0.1s linear; -ms-transition: background-position 0.1s
linear; -o-transition: background-position 0.1s linear; transition: background-position
0.1s linear; }

.btn-primary, .btn-primary:hover { text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25); color:
#ffffff; }

.btn-primary.active { color: rgba(255, 255, 255, 0.75); }

.btn-primary { background-color: #4a77d4; background-image: -moz-linear-gradient(top,
#6eb6de, #4a77d4); background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);
background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4));
background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4); background-image: -o-
linear-gradient(top, #6eb6de, #4a77d4); background-image: linear-gradient(top, #6eb6de,
#4a77d4); background-repeat: repeat-x; filter:
progid:dximagetransform.microsoft.gradient(startColorstr=#6eb6de, endColorstr=#4a77d4,
GradientType=0);  border: 1px solid #3762bc; text-shadow: 1px 1px 1px rgba(0,0,0,0.4);
box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5); }

.btn-primary:hover, .btn-primary:active, .btn-primary.active, .btn-primary.disabled,
.btn-primary[disabled] { filter: none; background-color: #4a77d4; }

.btn-block { width: 10%; display:block; }

* { -webkit-box-sizing:border-box; -moz-box-sizing:border-box; -ms-box-sizing:border-box;
-o-box-sizing:border-box; box-sizing:border-box; }

html { width: 100%; height:100%; overflow:hidden}

body {
    width: 100%;
    height:100%;
    font-family: 'Open Sans', sans-serif;
    background: #092756;
    color: #fff;
    font-size: 18px;
    text-align:center;
    letter-spacing:1.2px;
```

```
    background: -moz-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%),-moz-linear-gradient(top,  rgba(57,173,219,.25) 0%,
rgba(42,60,87,.4) 100%), -moz-linear-gradient(-45deg,  #670d10 0%, #092756 100%);
    background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), -webkit-linear-gradient(top,  rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -webkit-linear-gradient(-45deg,  #670d10 0%,#092756 100%);
    background: -o-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), -o-linear-gradient(top,  rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -o-linear-gradient(-45deg,  #670d10 0%,#092756 100%);
    background: -ms-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), -ms-linear-gradient(top,  rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), -ms-linear-gradient(-45deg,  #670d10 0%,#092756 100%);
    background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104,128,138,.4)
10%,rgba(138,114,76,0) 40%), linear-gradient(to bottom,  rgba(57,173,219,.25)
0%,rgba(42,60,87,.4) 100%), linear-gradient(135deg,  #670d10 0%,#092756 100%);
    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D',
endColorstr='#092756',GradientType=1 );

}



.login h1 { color: #fff; text-shadow: 0 0 10px rgba(0,0,0,0.3); letter-spacing:1px; text-
align:center; }

input {
    width: 20%;
    margin: 16px 16px;
    background: rgba(0,0,0,0.3);
    border: none;
    outline: none;
    padding: 1rem 1rem;
    font-size: 15px;
    color: #fff;
    text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
    border: 1px solid rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}

input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px
rgba(255,255,255,0.2); }

button{
    display: block;
    margin: 2rem auto;
```

```
    border-radius: 20rem;
    padding: 1rem 4rem;
    font-size: 1.5rem;
    font-weight: 600;
    letter-spacing: 1px;
    font-family: sans-serif;
background-color: #FFC107;
border: none;
transition: all .2s;
}
.login input {
    display: inline-block;
    font-family: cursive;
    margin: 1rem 1rem;
}
::placeholder {
  color: #858585;
}
```

HTML Coding:

```
<!DOCTYPE
html>
            <html >
            <!--From https://codepen.io/frytyler/pen/EGdtg-->
            <head>
              <meta charset="UTF-8">
              <title>ML API</title>
              <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
            type='text/css'>
            <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
            type='text/css'>
            <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
            type='text/css'>
            <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
            rel='stylesheet' type='text/css'>
            <link rel="stylesheet" href="../static/css/style.css">

            </head>


            <body>
             <div class="login">
                   <h1>Chronic Kidney Disease Detection</h1>


                 <!-- Main Input For Receiving Query to our ML -->
                <form action="{{ url_for('predict')}}"method="post">
                  <input type="text" name="age" placeholder="Age" required="required" >
```

```
        <input type="text" name="bp" placeholder="Blood Pressure in mm/Hg"
required="required" >
        <input type="text" name="bu" placeholder="Blood Urea(mgs/dl)" required="required"
>
               <input type="text" name="sc" placeholder="Serum Creatinine(mgs/dl)"
required="required" >
               <input type="text" name="sod" placeholder="Sodium(mEq/L)"
required="required" >
               <input type="text" name="hemo" placeholder="Hemoglobin(gms)"
required="required" >
               <input type="text" name="wc" placeholder="White Blood Cell
Count(cells/cumm)" required="required" >
               <input type="text" name="rc" placeholder="Red Blood Cell
Count(millions/cmm)" required="required" >
        <input type="text" name="bgr" placeholder="Blood Glucose Random(mgs/dl)"
required="required" >
               <input type="text" name="al" placeholder="Albumin (Nominal: 0-5)"
required="required" >
               <input type="text" name="su" placeholder="Sugar (Nominal: 0-5)"
required="required" >
        <input type="text" name="appet" placeholder="Appetite(Good or Poor)"
required="required" >
               <input type="text" name="rbc" placeholder="Red Blood Cells(Normal or
Abnormal)" required="required" >
               <input type="text" name="pc" placeholder="Pus Cell(Normal or Abnormal)"
required="required" >
               <input type="text" name="pcc" placeholder="Pus Cell Clumps(Present or Not
Present)" required="required" >
               <input type="text" name="ba" placeholder="Bacteria(Present or Not Present)"
required="required" >
               <input type="text" name="htn" placeholder="Hypertension(Yes or No)"
required="required" >
               <input type="text" name="dm" placeholder="Diabetes Mellitus(Yes or No)"
required="required" >
               <input type="text" name="cad" placeholder="Coronary Artery Disease(Yes or
No)" required="required" >
               <input type="text" name="pe" placeholder="Pedal Edema(Yes or No)"
required="required" >
               <input type="text" name="ane" placeholder="Anemia(Yes or No)"
required="required" >


        <button type="submit" class="btn btn-primary btn-block btn-
large">Predict</button>
    </form>


   <br>
    <font size="6">
```

```
        <b>{{ prediction_text }}</b>


      </div>




    </body>
    </html>
```

Python Coding(app.py):

```
import numpy as np
import pickle
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
import pandas as pd


app = Flask(__name__)
model = pickle.load(open('model_rf.pkl','rb'))
scaler = pickle.load(open('scaler.pkl','rb'))


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/predict',methods=['POST'])
def predict():
    one = ['yes', 'present', 'good', 'normal', 'Yes', 'Present', 'Good', 'Normal', 'YES',
'PRESENT', 'GOOD', 'NORMAL']
    zero = ['no', 'notpresent', 'not present', 'poor', 'abnormal', 'No', 'Notpresent',
'NotPresent', 'Not Present', 'Poor', 'Abnormal', 'AbNormal', 'NO', 'NOTPRESENT', 'NOT
PRESENT', 'POOR', 'ABNORMAL']
    int_features = []
    for i in request.form.values():
        if i in one:
            int_features.append(1.0)
        elif i in zero:
            int_features.append(0.0)
        else:
            int_features.append(float(i))

    final_features = [np.array(int_features)]
```

```
            final_features = scaler.transform(final_features)
            prediction = model.predict(final_features)

            output = prediction

            if output == [0]:
                output = "Kidney Disease Not Detected"
            elif output == [1]:
                output = "Kidney Disease Detected"

            return render_template('index.html', prediction_text='Diagnosis Result:
        {}'.format(output))


        if __name__ == "__main__":
            app.run(debug=True)
```

Scalar Coding:

```
Import panda as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
df = pd.read_csv("kidney.csv")


df[['pcv', 'wc', 'rc', 'dm', 'cad', 'classification']] = df[['pcv', 'wc', 'rc', 'dm', 'cad',
'classification']].replace(to_replace={'\t8400':'8400', '\t6200':'6200', '\t43':'43', '\t?':np.nan,
'\tyes':'yes', '\tno':'no', 'ckd\t':'ckd'})


df.fillna(method='ffill', inplace=True)
df.fillna(method='bfill', inplace=True)


df[['pcv', 'wc', 'rc']] = df[['pcv', 'wc', 'rc']].astype('float64')


df.drop(['id', 'sg', 'pcv', 'pot'],axis=1,inplace=True)


col = ['rbc', 'pcc', 'pc', 'ba', 'htn', 'dm', 'cad', 'pe', 'ane']
encoder = LabelEncoder()
for col in col:
    df[col] = encoder.fit_transform(df[col])
```

```
df[['appet', 'classification']] = df[['appet', 'classification']].replace(to_replace={'good':'1',
'ckd':'1', 'notckd':'0', 'poor':'0'})
df[['classification', 'appet']] = df[['classification', 'appet']].astype('int64')

X = df.drop("classification", axis=1)
y = df["classification"]


scaler = StandardScaler()
features = scaler.fit_transform(X)


import pickle
pickle.dump(scaler, open('scaler.pkl','wb'))
scaler = pickle.load(open('scaler.pkl','rb'))
```

Scalar Coding(Kidney Model):

```
Import panda pd
import numpy as np


from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier


df = pd.read_csv("kidney.csv")


df[['pcv', 'wc', 'rc', 'dm', 'cad', 'classification']] = df[['pcv', 'wc', 'rc', 'dm', 'cad',
'classification']].replace(to_replace={'\t8400':'8400', '\t6200':'6200', '\t43':'43',
'\t?':np.nan, '\tyes':'yes', '\tno':'no', 'ckd\t':'ckd'})


df.fillna(method='ffill', inplace=True)
df.fillna(method='bfill', inplace=True)


df[['pcv', 'wc', 'rc']] = df[['pcv', 'wc', 'rc']].astype('float64')


df.drop(['id', 'sg', 'pcv', 'pot'],axis=1,inplace=True)


col = ['rbc', 'pcc', 'pc', 'ba', 'htn', 'dm', 'cad', 'pe', 'ane']
encoder = LabelEncoder()
for col in col:
```

```
    df[col] = encoder.fit_transform(df[col])

df[['appet', 'classification']] = df[['appet',
'classification']].replace(to_replace={'good':'1', 'ckd':'1', 'notckd':'0', 'poor':'0'})
df[['classification', 'appet']] = df[['classification', 'appet']].astype('int64')

X = df.drop("classification", axis=1)
y = df["classification"]


scaler = StandardScaler()
features = scaler.fit_transform(X)


x_train, x_test, y_train, y_test = train_test_split(features, y, test_size=0.3,
random_state=42)


rf = RandomForestClassifier(n_estimators=50, random_state=42)
rf.fit(x_train,y_train)


import pickle
pickle.dump(rf, open('model_rf.pkl','wb'))
model_rf = pickle.load(open('model_rf.pkl','rb'))
```

Project

Development Phase

Model Performance

Test

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** MAE - , MSE - , RMSE - , R2 score - **Classification Model:** Confusion Matrix - , Accuracy Score- & Classification Report -	See Below
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	See Below

1. Metrics

Model: Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train , y_train)
prediction = model.predict(x_test)
print(prediction)
from sklearn.metrics import confusion_matrix
print('RandomForest\n')
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

```
[0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1
 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1
 0 0 0 0 1 0]
RandomForest

confusion_matrix
[[52  1]
 [ 2 25]]


accuracy_score
0.9625
```

2. Tune the Model

Hyperparameter Tuning:

- The number of features is important and should be tuned in random forestclassification.
- Initially all parameters in the dataset are taken as independent values to arrive at thedependent decision of Chronic Kidney Disease or No Chronic Kidney Disease.
- But the result was not accurate so used only 9 more correlated values as independentvalues to arrive at the dependent decision of Chronic Kidney Disease or not.

Validation Method:

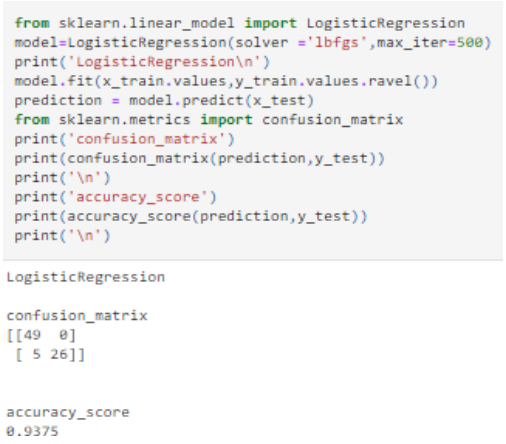
- It involves partitioning the training data set into subsets, where one subset is held out to testthe performance of the model. This data set is called the validation data set.
- Cross validation is to use different models and identify the best:

Logistic Regression Model performance values:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression(solver ='lbfgs',max_iter=500)
print('LogisticRegression\n')
model.fit(x_train.values,y_train.values.ravel())
prediction = model.predict(x_test)
from sklearn.metrics import confusion_matrix
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

```
LogisticRegression

confusion_matrix
[[49  0]
 [ 5 26]]


accuracy_score
0.9375
```

Hence we tested with Logistic regression and Random Forest Classification wherein theaccuracy ofRandom Forest classification is 95% compared with Logistic Regression.

Metric	Logistic Regression	Random Forest Classification
Accuracy	0.9375	0.9625
Other metrics		

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression(solver ='lbfgs',max_iter=500)
print('LogisticRegression\n')
model.fit(x_train.values,y_train.values.ravel())
prediction = model.predict(x_test)
from sklearn.metrics import confusion_matrix
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

```
LogisticRegression

confusion_matrix
[[49  0]
 [ 5 26]]


accuracy_score
0.9375
```

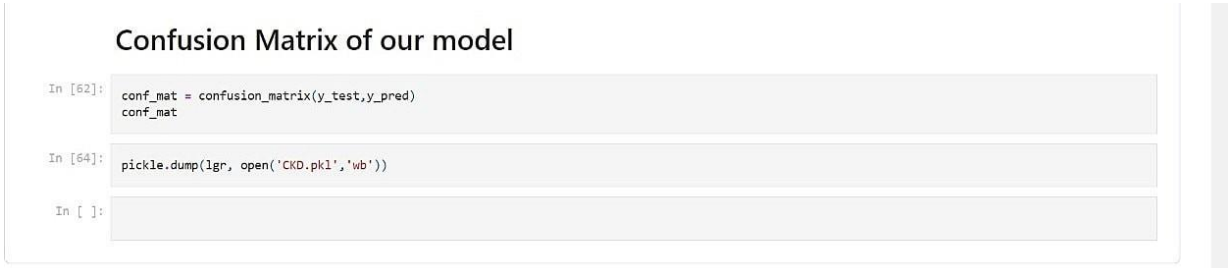
```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(x_train , y_train)
prediction = model.predict(x_test)
print(prediction)
from sklearn.metrics import confusion_matrix
print('RandomForest\n')
print('confusion_matrix')
print(confusion_matrix(prediction,y_test))
print('\n')
print('accuracy_score')
print(accuracy_score(prediction,y_test))
print('\n')
```

```
[0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1
 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1
 0 0 0 0 1 0]
RandomForest

confusion_matrix
[[52  1]
 [ 2 25]]


accuracy_score
0.9625
```

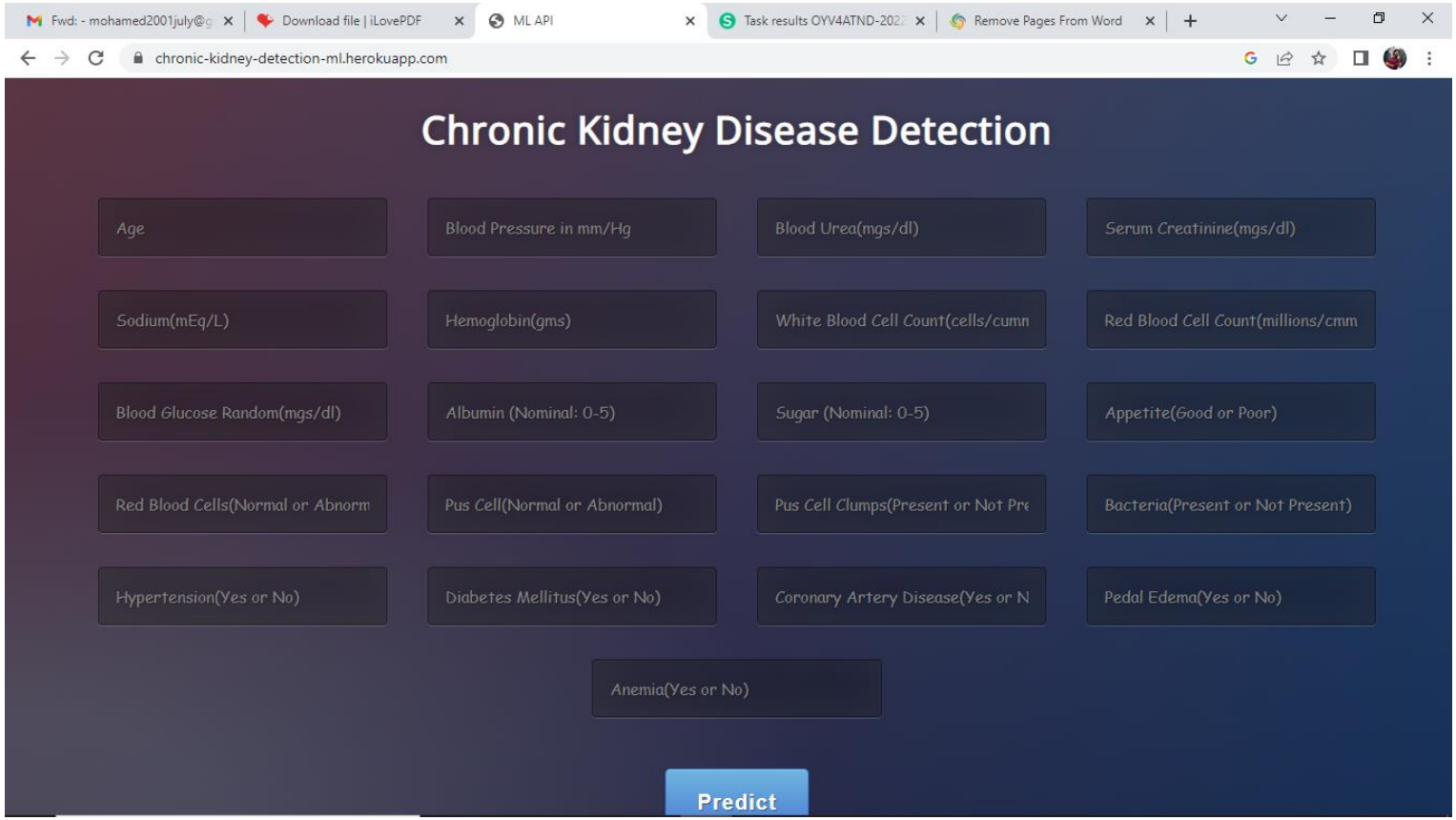
The above table shows that Random Forest Classification gives better results over Logistic regression.

Result:

Accuracy



Chronic kidney disease Application:



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 andadvise 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

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 got89% 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 thatwe 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 highlybeneficial for a large section of society.