

# **EARLY PREDICTION OF CHRONIC KIDNEY DISEASE USING MACHINE LEARNING**

**IBM-Project-41546-1660642789**

*Nalaiya Thiran project based learning on Professional readlines for  
innovation,employment and entrepreneurship*

**A PROJECT BY**

**KAVIYA P - 612719104033**

**SATHIYA PRABA.S - 612719104059**

**KANTHA.N - 612719104031**

**SWETHA.R - 612719104066**

**BACHELOR OF ENGINEERING COMPUTER SCIENCE AND  
ENGINEERING**

**THE KAVERY ENGINEERING COLLEGE**

**MECHERI,SALEM - 636 453**

## Table of contents

Chapter No	Title	Page No
<b>1</b>	<b>Introduction</b>	<b>05</b>
	1.1 Project Overview	<b>05</b>
	1.2 Purpose	<b>06</b>
<b>2</b>	<b>Literature Survey</b>	<b>07</b>
	2.1 Existing Problem	<b>07</b>
	2.2 References	<b>07</b>
	2.3 Problem Statement Definition	<b>09</b>
<b>3</b>	<b>Ideation and Proposed solution</b>	<b>10</b>
	3.1 Empathy map canvas	<b>10</b>
	3.2 Ideation and Brainstorming	<b>11</b>
	3.3 Proposed Solution	<b>14</b>
	3.4 Problem Solution Fit	<b>16</b>
<b>04</b>	<b>Requirement Analysis</b>	<b>17</b>
	4.1 Functional Requirement	<b>17</b>
	4.2 Non-Functional Requirements	<b>18</b>
<b>05</b>	<b>Project Design</b>	<b>19</b>
	5.1 DataFlow Diagram	<b>19</b>
	5.2 solution & Technical Architecture	<b>20</b>

	<b>5.3 User Stories</b>	<b>21</b>
<b>06</b>	<b>Project Planning &amp; Scheduling</b>	<b>22</b>
	<b>6.1 Sprint Planning &amp; Estimation</b>	<b>22</b>
	<b>6.2 Sprint Delivery Schedule</b>	<b>23</b>
	<b>6.3 Report From JIRA</b>	<b>26</b>
<b>07</b>	<b>Coding &amp; Solution</b>	<b>27</b>
	<b>7.1 Accuracy of Decision tree</b>	<b>27</b>
	<b>7.2 Accuracy of SVA</b>	<b>27</b>
	<b>7.3 Accuracy of Random Forest</b>	<b>27</b>
<b>08</b>	<b>Testing</b>	<b>28</b>
	Unit testing	<b>28</b>
	Functional testing	<b>28</b>
	Integration testing	<b>28</b>
	System testing	<b>29</b>
	<b>8.1 Test cases</b>	<b>29</b>
	<b>8.2 User Acceptance testing</b>	<b>30</b>
<b>09</b>	<b>Result</b>	<b>31</b>
	<b>9.1 Performance Metrics</b>	<b>31</b>
	<b>1)Confusion Matrix Description</b>	<b>31</b>
	<b>2)Classification Accuracy</b>	<b>31</b>
	<b>3) Classification Error</b>	<b>31</b>

	<b>4)Precision</b>	<b>32</b>
	<b>5)Recall</b>	<b>32</b>
	<b>6)F-Measure</b>	<b>32</b>
	<b>7)Roc and AUC</b>	<b>32</b>
<b>10</b>	<b>Advantages &amp; Disadvantages</b>	<b>33</b>
	<b>10.1 Advantages</b>	<b>33</b>
	<b>10.2 Disadvantages</b>	<b>33</b>
<b>11</b>	<b>conclusion</b>	<b>34</b>
<b>12</b>	<b>Future Scope</b>	<b>35</b>
<b>13</b>	<b>Source Code</b>	<b>36</b>

# 1. INTRODUCTION

## 1.1 PROJECT OVERVIEW

In today's era everyone is trying to be conscious about health although due to workload and busy schedule one gives attention to the health when it shows any symptoms of some kind. But CKD is a disease which doesn't show symptoms at all or in some cases it doesn't show any disease specific symptoms it is hard to predict, detect and prevent such a disease and this could lead to permanent health damage, but machine learning can be a hope in this problem it is best in prediction and analysis. By using data of CKD patients with 14 attributes and 400 records we are going to use various machine learning techniques like Decision Tree, SVM, etc. To build a model with maximum accuracy of predicting whether CKD or not and if yes then its severity. Chronic Kidney Disease is one of the most critical illnesses nowadays and proper diagnosis is required as soon as possible. Machine learning technique has become reliable for medical treatment. With the help of machine learning classifier algorithms, the doctor can detect the disease on time. For this perspective, Chronic Kidney Disease prediction has been discussed in this article. Chronic Kidney Disease dataset has been taken from the UCI repository. Seven classifier algorithms have been applied in this research such as artificial neural network, C5.0, Chi-square Automatic interaction detector, logistic regression, linear support vector machine with penalty L1 & with penalty L2 and random tree. The important feature selection technique was also applied to the dataset. For each classifier, the results have been computed based on (i) full features, (ii) correlation-based feature selection, (iii) Wrapper method feature selection, (iv) Least absolute shrinkage and selection operator regression, (v) synthetic minority over-sampling technique with least absolute shrinkage and selection operator regression selected features, (vi) synthetic minority oversampling technique with full features. From the results, it is marked that LSVM with penalty L2 is giving the highest accuracy of 98.86% in synthetic minority over-sampling technique with full features. Along with accuracy, precision, recall, F-measure, area under the curve and GINI coefficient have been computed and compared

results of various algorithms have been shown in the graph. Least absolute shrinkage and selection operator regression selected features with synthetic minority over-sampling technique gave the best after synthetic minority over-sampling technique with full features. In the synthetic minority over-sampling technique with least absolute shrinkage and selection operator selected features, again linear support vector machine gave the highest accuracy of 98.46%. Along with machine learning models one deep neural network has been applied on the same dataset and it has been noted that deep neural network achieved the highest accuracy of 99.6%.

## **1.2 PURPOSE**

Your kidneys, each just the size of a computer mouse , filter all the blood in your body every 30 minutes They work hard to remove wastes, toxins, and excess fluid. They also help control blood pressure, stimulate production of red blood cells , keep your bones healthy , and regulate blood chemicals that are essential to life. Kidneys that function properly are critical for maintaining good health, however, more than one in seven American adults are estimated to have chronic kidney disease (CKD).some other health consequences of CKD include, anemia or low number of red blood cells, increased occurrence of infections, low calcium levels , high potassium levels, and high phosphorus levels in the blood , loss of appetite or eating less, depression or lower quality of life CKD has varying levels of seriousness. So avoid this disease by using machine learning methods. While we using machine learning method we can early predict the disease with the help of the dataset .The dataset includes age,bb,Albumin,rbc,pc,pcc,sc,hb,wbcc,rbcc, Anaemia,classification,packed cell volume

## **2. LITERATURE SURVEY**

### **2.1 EXISTING PROBLEM**

Majority of the time the disease is detected in its final stage and which sometimes leads to kidney failure. The existing system of diagnosis is based on the examination of urine with the help of serum creatinine level. Many medical methods are used for this purpose such as screening, ultrasound method. Chronic kidney disease occurs when a disease or condition impairs kidney function, causing kidney damage to worsen over several months or years. Diseases and conditions that cause chronic kidney disease include: Type 1 or type 2 diabetes. high blood pressure.

### **2.2 REFERENCES**

#### **[1] Author Name: Tomas E**

The small dataset of 400 records have been collected from Apollo Hospital, India in 2015 taken over a two-month period. ANOVA test, the Pearson's correlation, and the Cramer's V test are applied and removed the redundant features in dataset. By using filter feature selection method, three features (hemoglobin, albumin, and specific gravity) are selected and trained using Logistic regression, support vector machines, random forest, and gradient boosting algorithm and reached an accuracy of 99.1% accuracy. Data used in this research is small. So, in future, need to validate the results by using big dataset and for reducing the prevalence of CKD, planned to predict if a person with CKD risk factors such as diabetes, hypertension, and family history of kidney failure will have CKD in the future or not by using appropriate dataset.

#### **[2] Author Name: Ebrahime Mohammed Senan, Nizar Alsharif**

The dataset was collected from 400 patients containing 24 features. The dataset was divided into 75% training and 25% testing and validation. The dataset was processed to remove outliers and replace missing numerical and nominal values using mean and mode statistical measures respectively.

**[3] Author Name: Bidri Deepik, Vasudeva Rao KR, Dharmaj N Rampure, Prajwal P and Devanand Gowda G**

Early Prediction of Chronic Kidney Disease by using Naive bayes, K-Nearest neighbor. KNN algorithm takes CKD parameters as input and predicts the disease based on old CKD patient's data. To develop user interface for create easier communication between doctors and patients.

**[4] Author Name: Sreeji S, Balamurugan Balusamy**

In Design System For Early Detection And Prediction Of Chronic Kidney Disease Using Machine Learning Techniques, The SVM, and Naive Bayes classification algorithms were applied on the processed data. The information assortment contains 400 patient records, for certain qualities missing. It comprises of 24 clinical highlights that show up in the anticipation of ongoing kidney illness, with one class quality demonstrating the event of persistent renal disappointment in the patient. Data used in this research is small. In future by using better algorithms the accuracy can be increased and also need to validate the results by using big dataset.

**[5] Author Name: Reshma S , Salma Shaji , S R Ajina, Vishnu Priya S R , Janisha A**

In this research, The dataset contains 400 samples of two different classes. Out of 25 attributes, 11 are numeric and 13 are nominal and one is class attribute. The data set contains number of missing values. Here the information of dataset uses the patient's data like age, blood pressure, specific gravity, albumin, sugar, red blood cells etc. The sample Data used in this research is small. So, in future by using by using better deep learning concepts the accuracy can be increased.



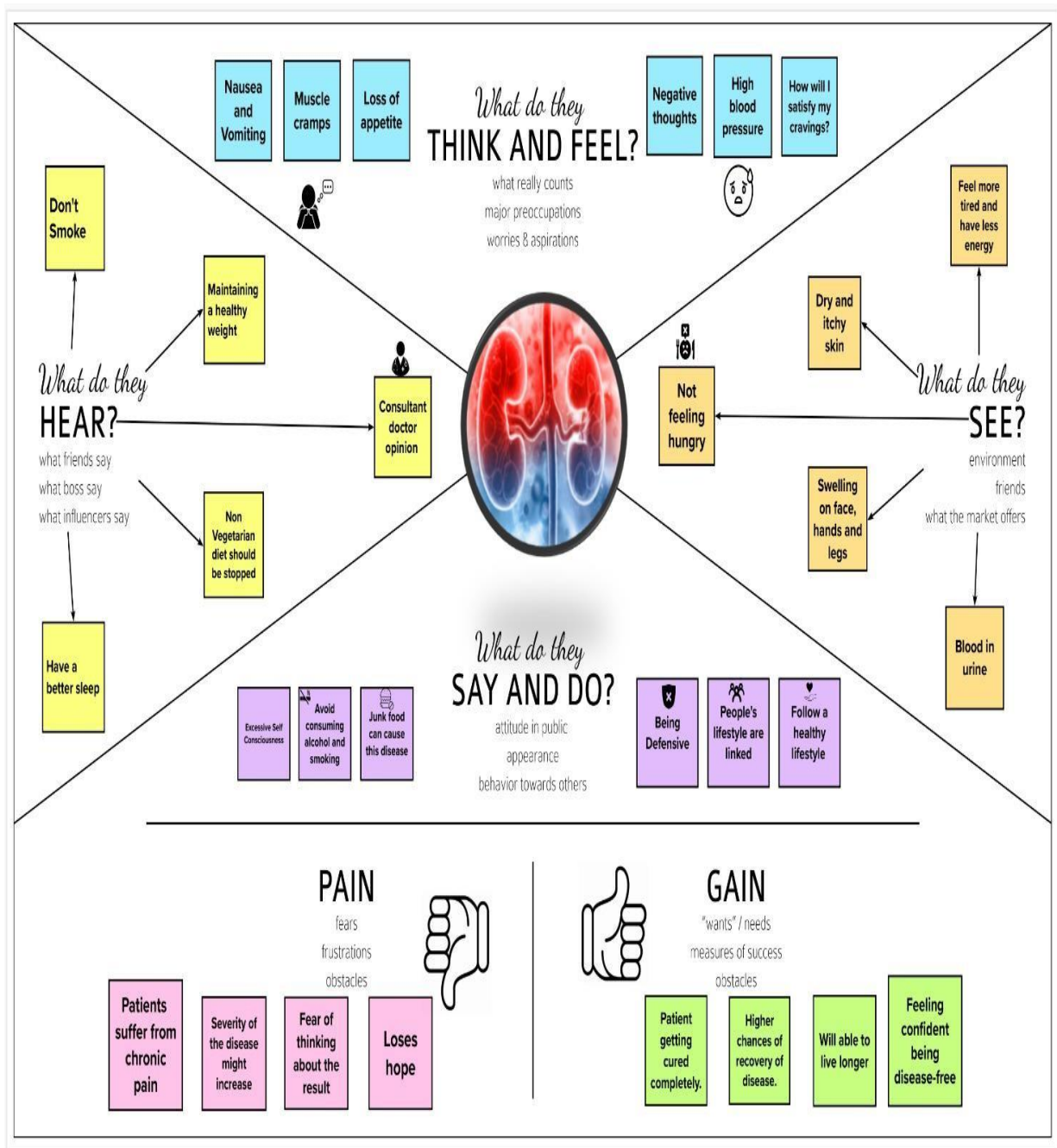
## 2.3 PROBLEM STATEMENT DEFINITION

Irregular lifestyle of current days leads to many disease chronic diseases. One of them being Chronic Kidney Disease(CKD), it has been of a growing concern, kidney is one of the most important organs in the body required for filtering blood, once a person has lost their kidneys, they could survive only for 18 days without their kidneys, it would take a fortune to just keep the person alive, with treatments like dialysis, transplant etc. 10% of the population worldwide is affected by chronic kidney disease (CKD), and millions die each year because they do not have access to affordable treatment. People usually don't realize that the medical tests we perform for various purposes can contain valuable information related to kidney disease. Subsequently, the attributes of various medical tests are examined to distinguish which attributes may contain useful information about the disease. The information, they say, helps us gauge the severity of the problem, and we use that information to build a machine learning model that predicts chronic kidney disease. Early detection of kidney disease can help in treatment which could save lives. Analyzing various medical tests, would give us an idea about which attributes help us distinguish the disease. The main aim of this project is to predict whether the patient have chronic kidney disease or not, in more accurate and faster way based on certain diagnostic measurements like Blood Pressure (Bp), Albumin(AI).




### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 EMPATHY MAP CANVAS



## 3.2 IDEATION & BRAINSTORMING

Step 1: Team gathering, collaboration and select the problem statement:



### 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 a smooth and productive session

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

## Step : 2 Brainstorm, idea Listing and Grouping:

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 icon to quickly start to start drawing!

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 can break it up into smaller sub-groups.  
20 minutes

Person 4

Tip

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

Cost efficient for patient

Easy identification using algorithms

User friendly

Save patient time

**Kaviya p**

Easy to identify

Quickly start necessary steps

Reduce the time to find the symptoms

**Sathiya praba S**

Results quickly

Easy to work

Increasing scope in future

**Kantha N**

ML helps in medical field

Predicts the accuracy based on algorithms

Highly beneficial

**Swetha R**

Best accuracy

Provides better to patients

Possible to diagnose people based on symptoms

Activate Windows  
Go to Settings to activate Windows.

## Step : 3 idea prioritization

3

### 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



### 3.3 PROPOSED SOLUTION

S.NO	Parameter	Description
1	Problem statement	Chronic Kidney Disease is a major concern for the global health care system. Chronic Kidney Disease is now wreaking havoc on society and is spreading at an alarming rate. Various efforts have been undertaken to advance early therapy to prevent the condition from progressing to CKD. Recent research suggest that some of the negative outcomes can be avoided with early identification and treatment.
2	Feasibility study	To predict the early onset of CKD, three Machine Learning techniques are used: effectiveness is evaluated and the prediction of how many people have been affected by CKD is identified. 23 October 2022 PNT2022TMID41072 *Random Forest, Decision Tree, Support Vector

		Machines. Using these techniques, each algorithm's
3	Novelty	<p>The renal patient is recognized by undertaking two primary tests. *A Blood Test to determine Glomerular Filtration Rate(GFR).</p> <p>*A Urine Test to determine Albumin.</p>
4	Social Impact	<p>As people don't undergo the general test of their health, early detection of CKD is not identified. This creates a great social impact of not being aware of CKD. As a result of this many people are getting affected by CKD.</p>

### 3.3 PROBLEM SOLUTION FIT

Problem-Solution Fit canvas		Early detection of chronic kidney disease using machine learning	Version:
<b>1. CUSTOMER SEGMENT(S)</b> 1. Medical Researcher 2. Truck drivers 3. Diabetics patients 4. Elderly women	<b>6. CUSTOMER LIMITATIONS</b> <small>EG. BUDGET, DEVICES</small> 1. Accuracy is less 2. Cannot handle more data 3. Feature selection is not applied	<b>5. AVAILABLE SOLUTIONS</b> <small>PLUSES &amp; MINUSES</small> 1. Prediction model using machine learning methodologies	Define CS, fit into CL Explore AS, differentiate
<b>2. PROBLEMS / PAINS + ITS FREQUENCY</b> 1. Complex for large datasets 2. Disassociation between classes occurs 3. More iteration required	<b>9. PROBLEM ROOT / CAUSE</b> 1. Algorithm is not efficient for large datasets 2. Feature selection is not done	<b>7. BEHAVIOR + ITS INTENSITY</b> 1. Loads the dataset, Analysis the dataset 2. Implement random forest 3. Predicts the stages of CKD algorithm	Focus on PR, map into BE, understand BC Focus on PR, map into BE, understand and BC
<b>3. TRIGGERS TO ACT</b> 1. High blood 2. High glucose 3. Glomerular filtration rate <b>4. EMOTIONS</b> <small>BEFORE / AFTER</small> 1. Anxiety 2. Sorrow and anger 3. Stress	<b>10. YOUR SOLUTION</b> Implement feature embedding method to reduce the unwanted feature to increase model performance	<b>8. CHANNELS of BEHAVIOR</b> ONLINE 1. Anaconda navigator 2. Jupyter notebook 3. Google colab OFFLINE 1. Excel/spreadsheet 2. Rattle GUI 3. Rapid miner-Machine learning	Identify strong TR, B, EM Extract online B, offline CH of BE



## 4. REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENT

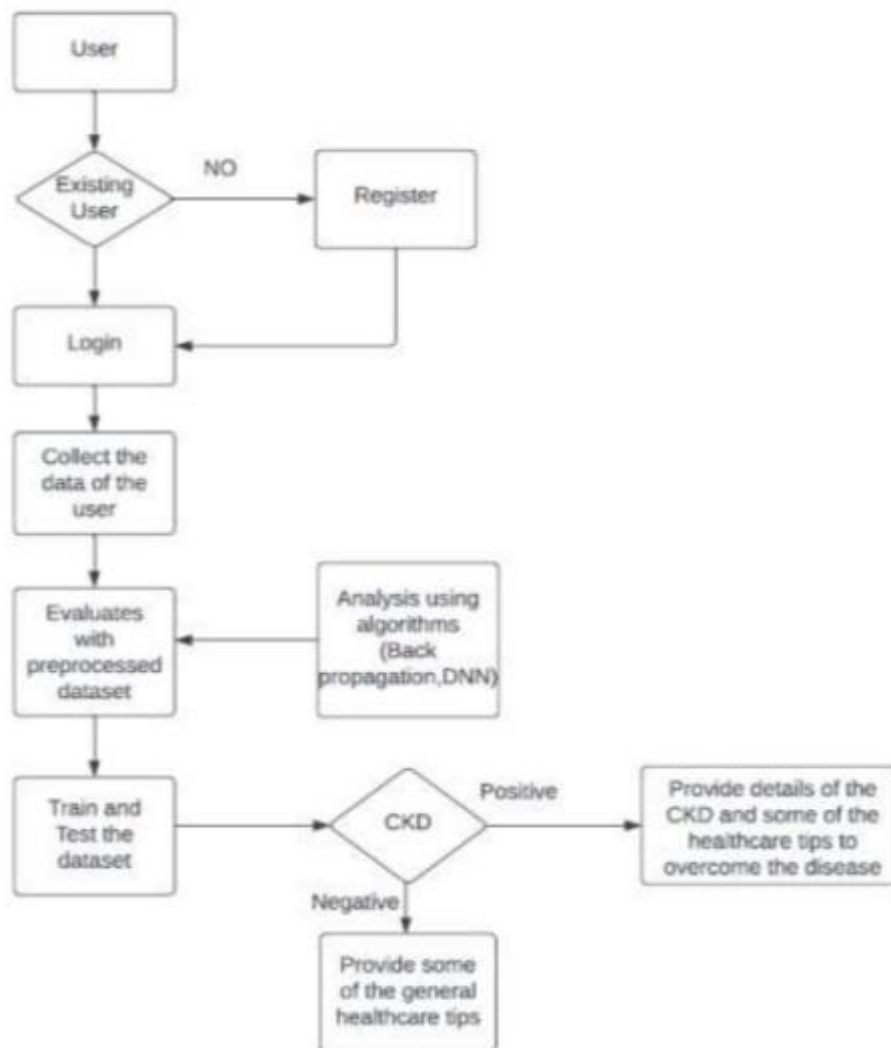
FR . NO	FUNCTIONAL REQUIREMENT (Epic)	SUB REQUIREMENT (STORY/SUB-TASK)
FR-1	Home page (login page)	Introduction page of the website. Symptoms and steps to cure will be displayed.If the user already exists asks to login or else redirects to Sign Up.
FR-2	User sign up page	The user had to enter the username, phone number and password.
FR-3	User verification	After getting the phone number the OTP will be sent via SMS and it will be verified.
FR-4	Dataset collection	Collect the data set of Chronic Kidney Disease patients and pre-process the data.
FR-5	Training the model	By using the pre-processed data, we can train the model by using Deep Neural Networks.

## 4.2 NON FUNCTIONAL REQUIREMENTS

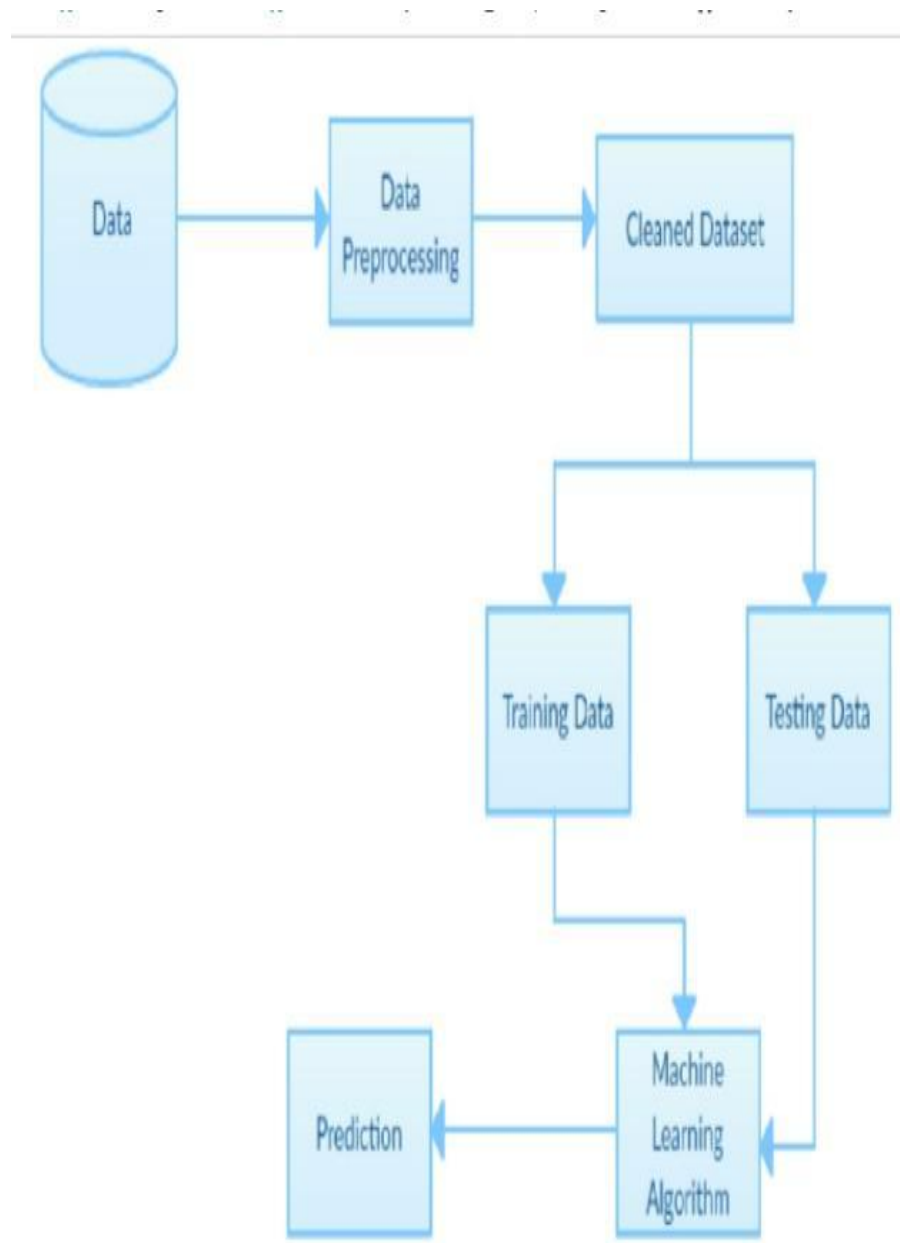
FR. NO	NON FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	Usability	Creating a machine learning model that uses the attributes of medical tests taken for different purposes to detect chronic kidney disease at early stage
NFR-2	Security	The reports are maintained confidentially to the customer.
NFR-3	Reliability	Earlier prediction can save the life of many users who may be affected by the CKD, hence this model produces the reliable results.
NFR-4	Performance	By using DNN, we can predict the chronic kidney disease with more than 98% of accuracy. In the DNN we have more hidden layers . hence its accuracy.

## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAMS



## 5.2 SOLUTION & TECHNICAL ARCHITECTURE



### 5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	Can access account	High	Sprint-1
	Login	USN-2	As a user, I can log into the application	Login into account	High	Sprint-1
Data Entry	Enter data	USN-3	Enter symptoms and clinical data	Enter clinical data	High	Sprint-2
Customer  View results	View  Results	USN-4	Result can be viewed by the user	View Result	Medium	Sprint-3

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 SPRINT PLANNING & ESTIMATION

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers ,research publications etc.	25 OCTOBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	19 OCTOBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	16 OCTOBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	15 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	19 OCTOBER 2022

Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	31 OCTOBER 2022
Project Development	Develop & submit the developed code by testing it.	5 NOVEMBER 2022
Delivery of Sprint-1, 2, 3 & 4	Submit the source code	16 NOVEMBER 2022

## 6.2 SPRINT DELIVERY SCHEDULE

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	KAVIYA P SATHIYA PRABA S KANTHA N SWETHA R
Sprint-2		USN-2	As a user, I can register for the application through Gmail	5	medium	KAVIYA P SATHIYA PRABA S KANTHA N SWETHA R
Sprint-1	User Confirmation	USN-3	As a user, I will receive confirmation	10	High	KAVIYA P SATHIYA PRABA S KANTHA N

			email once I have registered for the application			SWETHA R
Sprint-2		USN-4	As a user, I will receive confirmation otp to verify the identity	5	High	KAVIYA P SATHIYA PRABA S KANTHAN SWETHA R
Sprint-2	Data collection	USN-5	As a user, I will enter the input data for disease prediction in the form	10	High	KAVIYA P SATHIYA PRABA S KANTHAN SWETHA R
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	KAVIYA P SATHIYA PRABA S KANTHAN SWETHA R
Sprint-3	Data Analysis	USN-7	As the admin, I will develop modules to preprocess and store the data.	10	High	KAVIYA P SATHIYA PRABA S KANTHAN SWETHA R
Sprint-4	Prediction of disease	USN-8	As the admin, I will build a Machine Learning model to predict the disease	10	High	KAVIYA P SATHIYA PRABA S KANTHAN SWETHA R

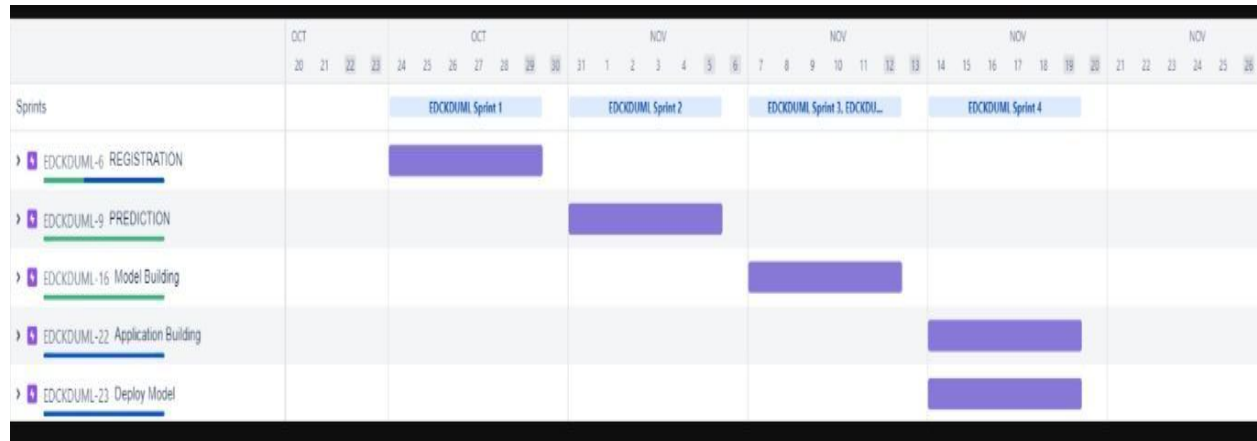


Sprint-4	Final Delivery	USN-9	Deploy the application in IBM cloud and make it available for use.	10	High	KAVIYA P SATHIYA PRABA S KANTHA N SWETHA R
----------	----------------	-------	--	----	------	---

### PROJECT TRACKER, VELOCITY & BURNDOWN CHART:

<b>Sprint</b>	<b>Total Story Points</b>	<b>Duration</b>	<b>Sprint Start Date</b>	<b>Sprint End Date (Planned)</b>	<b>Story Points Completed (as on Planned End Date)</b>	<b>Sprint Release Date (Actual)</b>
<b>Sprint-1</b>	20	6 Days	24 Oct 2022	29 Oct 2022	2 Nov 2022	3 Nov 2022
<b>Sprint-2</b>	20	6 Days	31 Oct 2022	05 Nov 2022	5 Nov 2022	7 Nov 2022
<b>Sprint-3</b>	20	6 Days	07 Nov 2022	12 Nov 2022	9 Nov 2022	12 Nov 2022
<b>Sprint-4</b>	20	6 Days	14 Nov 2022	12 Nov 2022	13 Nov 2022	16 Nov 2022

### 6.3 REPORT FROM JIRA



## 7. CODING & SOLUTIONING

models have been constructed using a training data set (280 instances) which is 70% of the original CKD data set . constructed models have been validated using test data which is 30% of original data with respect to the parameter accuracy. here , accuracy has been calculated using a confusion matrix . The best classifier model is the one with highest accuracy.

### **Accuracy of Decision tree**

Confusion Matrix has been generated by decision tree model for the test data (120 instances)with class (values:CKD,NON CKD)as the target variable is given by table 1.The confusion matrix clearly says that 7 instances are not classified properly and 113 instances have been classified accurately and the accuracy of this classifier model is **94.16%.**

### **Accuracy of SVM**

Confusion Matrix has been generated by SVM model for the test data(120 instances) with class (values:CKD,NON CKD)as the target variable is given by table 1.The confusion matrix clearly says that 2 instances are not classified properly and 118 instances have been classified accurately and the accuracy of this classifier model is **98.33%**

### **Accuracy of Random Forest**

Confusion Matrix has been generated by SVM model for the test data (120 instances) with class (values:CKD,NONCKD)as the target variable is given by table 1.The confusion matrix clearly says that 1 instances are not classified properly and 119 instances have been classified accuracy.

## **8. TESTING**

### **UNIT TESTING**

When the testing happens for some individual group or some related units then that type of testing is called as Unit Testing. It is often done by a programmer to test the part of the program he or she has implemented.

Unit Testing is successful means all the modules have been successfully tested and it can proceed further.

### **FUNCTIONAL TESTING**

This type of testing is tested because to check the functional components or the functionality required from the system is gained or not .It actually falls under the testing of the Black Box testing of Software Engineering. This part includes the feeding of the inputs in the system or the project and to check if that system or the project is getting the same value or not as expected if not then calculate the error as wanted and check for more .Functional Testing of this project mainly involves below things. All of these are tested successfully and errors are also calculated.

### **INTEGRATION TESTING**

In a total project or the system, many groups of components are getting added or summed up in the purpose of the project query. Integration testing is about to check the interaction between various modules of the project or the system. This module also includes the hardware and the software requirements of the project. All the individual modules are integrated and tested together.

All the best and extreme cases that the modules are interacting or not are successfully checked and passed, errors are calculated for the machine learning platforms.

## SYSTEM TESTING

This type of testing is actually meant for the system or the project and also the platform and the integrated software and tools, technologies are also tested. The idea or purpose behind the system testing is to check all the requirements that will be provided by the system.

This application of the project along with the tools and technologies has been tested in both Windows and Linux platform and also uncertified online apple MAC platform to check the requirements. It passed successfully.

This is a type of system or software testing where a system has been tested for availability. The purpose of this test is to check the business requirements and assess whether it will be accepted for delivery. In this part ADRIAN of Pyrimag search has been referred to, who worked with the same platform and to check if this project was accepted by the delivery partner or not.

### 8.1 TEST CASES

Attribute	Value Used
Age	Discrete Integer Values
Blood Pressure	Discrete Integer Values
Albumin	Nominal Values
Red Blood cells	Nominal Values(Normal, Abnormal)
Pus cell	Nominal Values(Normal, Abnormal)
Pus cells clumps	Nominal Values(Present, Not-Present)
Serum creatinine	Numeric Values
Haemoglobin	Numeric Values
White blood cell count	Discrete Integer Values
Red blood cell count	Numeric Values
Anaemia	Nominal Values(Yes, No)
Classification	Nominal Values(CKD, Not CKD)
Appetite	Nominal Values(Good, Poor)
Packed cell volume	Discrete Integer Values

## **8.2 USER ACCEPTANCE TESTING**

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps, or missing requirements in contrast to the actual requirements.

## 9. RESULT

### 9.1 PERFORMANCE METRICS

#### 1) CONFUSION MATRIX DESCRIPTION

TP: True Positive means output as positive such that the predicted result is correctly classified.

TN: True Negative means output as negative such that the predicted result is correctly classified.

FP: False Positive means output as positive such that predicted result is incorrectly classified.

FN: False Negative means output as negative such that predicted result is incorrectly classified.

#### 2) CLASSIFICATION ACCURACY

Classification accuracy shows the correct rate of prediction results. It computes the confusion matrix. The classification accuracy is found by equation

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100$$

#### 3) CLASSIFICATION ERROR

Classification error shows the incorrect rate of prediction results. It computes the confusion matrix. The classification error is found by equation 3:

$$Error = \frac{FP + FN}{TP + TN + FP + FN} * 100$$

#### 4) PRECISION

Precision is an important model performance evaluation matrix. It is the fraction of related instances among the total retrieved instances. It is a positive predicted value. The precision is calculated as follows in equation 4:

$$Precision = \frac{TP}{TP + FP} * 100$$

#### 5) RECALL

Recall is also an important model performance evaluation matrix. It is the fraction of related instances among the total number of retrieved instances. The recall is calculated as follows in equation 5:

$$Recall = \frac{TP}{TP + FN} * 100$$

#### 6) F-MEASURE

It is also known as F Score. F-measure is calculated so as to measure the accuracy of the test. It is calculated from the precision and recall by equation 6:

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

#### 7) ROC AND AUC

The performance of the classification model is measured from the Receiver operating a characteristic curve (ROC). ROC is a graph that is created for true positive rate vs. false positive rate at different classifications threshold. The entire area under the ROC curve is known as area of the curve (AOC). It gives a collective measure of performance across all achievable classification's threshold.



## **10. ADVANTAGES & DISADVANTAGES**

### **ADVANTAGES**

- With using of ANN algorithm, easy for small parameters, multiple processing units are connected to each other very effectively and efficiently errorless.
- With the KNN algorithm , there is no need to separate linear and nonlinear at no cost to learn good for large numbers of records fast to train the model.
- With the use of the DT algorithm, a nonparametric learning model simple to know & visualize .It can produce sets of rules to transfer knowledge .
- With using of the NB algorithm,it can contain numerical & categorical data independent between attributes . It is very fast algorithm.

### **DISADVANTAGES**

- With the ANN algorithm, It is complex and the time is long to train, sometimes overlapping with slow learning.
- With the KNN algorithm, It requires more memory space ,is slow to test , and depends on a number of dimensions sensitive to noise .
- With using of DT algorithm, complex for large dataset , unstable , difficult to learn
- With the use of the NB algorithm, in a small data set to reduce precision , it requires very range.

## 11. CONCLUSION

This paper presented a prediction algorithm to predict CKD at an early stage. The dataset shows input parameters collected from the CKD patients and the models are trained and validated for the given input parameters. Decision tree, Random Forest and Support Vector Machine learning models are constructed to carry out the diagnosis of CKD. The performance of the models are evaluated based on the accuracy of prediction. The results of the research showed That Random Forest Classifier model better predicts CKD in comparison to Decision trees and Support Vector machines. The comparison can also be done based on the time of execution, feature set selection as the improvisation of this research. The prediction of chronic kidney disease is very important and now-a-days it is the leading cause of death. The performance of the Decision tree method was found to be 99.25% accurate compared to the naive Bayes method. Classification algorithm on chronic kidney disease dataset the performance was obtained as 99.33% Specificity and 99.20% Sensitivity. We are also further working on enhancing the performance of prediction system accuracy in neural networks and deep learning algorithm.

## **12.FUTURE SCOPE**

- This would help detect the chances of a person having CKD further on in his life which would be really helpful and cost-effective people.
- This model could be integrated with normal blood report generation, which could automatically flag out if there is a person at risk. Patients would not have to go to a doctor unless they are flagged by the algorithms.
- This would make it cheaper and easier for the modern busy person.

## 13. APPENDIX

### SOURCE CODE

```
from flask import Flask, render_template, request, flash, redirect

import pickle

import numpy as np

from PIL import Image

from tensorflow.keras.models import load_model

app = Flask(__name__)

def predict(values, dic):

    if len(values) == 8:

        model = pickle.load(open('models/diabetes.pkl','rb'))

        values = np.asarray(values)

        return model.predict(values.reshape(1, -1))[0]

    elif len(values) == 26:

        model = pickle.load(open('models/breast_cancer.pkl','rb'))

        values = np.asarray(values)

        return model.predict(values.reshape(1, -1))[0]

    elif len(values) == 13:

        model = pickle.load(open('models/heart.pkl','rb'))

        values = np.asarray(values)

        return model.predict(values.reshape(1, -1))[0]
```

```
elif len(values) == 18:
```

```
    model = pickle.load(open('models/kidney.pkl','rb'))
```

```
    values = np.asarray(values)
```

```
    return model.predict(values.reshape(1, -1))[0]
```

```
elif len(values) == 10:
```

```
    model = pickle.load(open('models/liver.pkl','rb'))
```

```
    values = np.asarray(values)
```

```
    return model.predict(values.reshape(1, -1))[0]
```

```
@app.route("/")
```

```
def home():
```

```
    return render_template('home.html')
```

```
@app.route("/diabetes", methods=['GET', 'POST'])
```

```
def diabetesPage():
```

```
    return render_template('diabetes.html')
```

```
@app.route("/cancer", methods=['GET', 'POST'])
```

```
def cancerPage():
```

```
    return render_template('breast_cancer.html')
```

```
@app.route("/heart", methods=['GET', 'POST'])
```

```
def heartPage():
```

```
    return render_template('heart.html')
```

```
@app.route("/kidney", methods=['GET', 'POST'])
```

```
def kidneyPage():
```

```
    return render_template('kidney.html')
```

```
@app.route("/liver", methods=['GET', 'POST'])
```

```
def liverPage():
```

```
    return render_template('liver.html')
```

```
@app.route("/malaria", methods=['GET', 'POST'])
```

```
def malariaPage():
```

```
    return render_template('malaria.html')
```

```
@app.route("/pneumonia", methods=['GET', 'POST'])
```

```
def pneumoniaPage():
```

```
    return render_template('pneumonia.html')
```

```
@app.route("/predict", methods = ['POST', 'GET'])
```

```
def predictPage():
```

```
    try:
```

```
        if request.method == 'POST':
```

```
            to_predict_dict = request.form.to_dict()
```

```
            to_predict_list = list(map(float, list(to_predict_dict.values())))
```

```
            pred = predict(to_predict_list, to_predict_dict)
```

```
    except Exception as e:
```

```

print(e)

message = "Please enter valid Data"

return render_template("home.html", message = message)

return render_template('predict.html', pred = pred)

@app.route("/malariapredict", methods = ['POST', 'GET'])

def malariapredictPage():

    if request.method == 'POST':

        try:

            if 'image' in request.files:

                img = Image.open(request.files['image'])

                img = img.resize((36,36))

                img = np.asarray(img)

                img = img.reshape((1,36,36,3))

                img = img.astype(np.float64)

                model = load_model("models/malaria.h5")

                pred = np.argmax(model.predict(img)[0])

            except:

                message = "Please upload an Image"

                return render_template('malaria.html', message = message)

        return render_template('malaria_predict.html', pred = pred)

```

```
@app.route("/pneumoniapredict", methods = ['POST', 'GET'])

def pneumoniapredictPage():

    if request.method == 'POST':

        try:

            if 'image' in request.files:

                img = Image.open(request.files['image']).convert('L')

                img = img.resize((36,36))

                img = np.asarray(img)

                img = img.reshape((1,36,36,1))

                img = img / 255.0

                model = load_model("models/pneumonia.h5")

                pred = np.argmax(model.predict(img)[0])

            except:

                message = "Please upload an Image"

                return render_template('pneumonia.html', message = message)

        return render_template('pneumonia_predict.html', pred = pred)

if __name__ == '__main__':

    app.run(debug = True)
```