PROJECT REPORT

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Chronic Kidney Disease (CKD) or chronic renal disease has become a major issue with a steady growth rate. A person can only survive without kidneys for an average time of 18 days, which makes a huge demand for a kidney transplant and Dialysis. It is important to have effective methods for early prediction of CKD. Machine learning methods are effective in CKD prediction.

The field of biosciences has advanced to a larger extent and has generated large amounts of information from Electronic Health Records. Data mining methods and machine learning play a major role in this aspect of biosciences. Chronic Kidney Disease (CKD) is a condition in which the kidneys are damaged and cannot filter blood as they always do.

An early detection of CKD can improve the quality of life to a greater extent. This calls for a good prediction algorithm to predict CKD at an earlier stage.

ABOUT CKD:

Chronic kidney Disease (CKD) means your kidneys are damaged and not filtering your blood the way it should. The primary role of kidneys is to filter extra water and waste from your blood to produce urine and if the person has suffered from CKD, it means that wastes are collected in the body.

This disease is chronic because of the damage gradually over a long period. It is flattering a common disease worldwide. Due to CKD may have some health troubles. There are many causes for CKD like diabetes, high blood pressure, and heart disease. Along with these critical diseases, CKD also depends on age and gender.

If your kidney is not working, then you may notice one or more symptoms like abdominal pain, back pain, diarrhoea, fever, nosebleeds, rash, vomiting.

There are two main diseases of CKD:

(i) diabetes

(ii) high blood pressure.

So, controlling these two diseases is the prevention of CKD. Usually, CKD does not give any sign till the kidney is damaged badly. CKD is being increased rapidly as per the studies hospitalization cases increase 6.23 per cent per year but the global mortality rate remains fixed.

There are few diagnostic tests to check the condition of CKD:

(i) estimated glomerular filtration rate(eGFR)

(ii) urine test

(iii) blood pressure

1.2 PURPOSE:

- Our study aims to develop a machine-learning model that predicts chronic kidney disease using data mining techniques.
- To predict positive CKD status and the stages of CKD machine learning can be used
- Data mining techniques include Getting the data, loading the data into the platform, processing data, and etc.,

2. LITERATURE SURVEY

2.1 References

01. LINTA ANTONY, et al. "A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction" ***IEEE Access*** **8 (2021):**

In this paper, Chronic Kidney Disease (CKD) indicates a condition where human kidneys that are damaged [1] and unable to filter the blood stream and get rid of the metabolic waste the way are supposed to. This research aims to build an intelligent machine learning model that can be used reliably to establish CKD diagnosis. It has implemented five unsupervised algorithms, K-Means Clustering, DB-Scan, I-Forest, and Autoencoder. And integrating them with various feature selection methods. Integrating feature reduction methods with K- Means Clustering algorithm has achieved an overall accuracy of 99% in classifying the clinical data of CKD and Non-CKD

02. Ahmed J. Aljaaf, et al. "Early Prediction of Chronic Kidney Disease Using Machine Learning Supported by Predictive Analytics." ***Computers in biology and medicine*** **109 (2018):**

In this paper, study starts with 24 parameters in addition to the class attribute, and ends up by 30% of them as ideal sub set to predict Chronic Kidney Disease. A total of 4 machine learning based classifiers have been evaluated within a supervised learning setting, achieving highest performance outcomes of AUC 0.995, sensitivity 0.9897, and specificity 1.

03. Rahul Gupta ., et al. "Performance Analysis of Machine Learning Classifier for Predicting Chronic Kidney Disease."INCET (2020):

In this paper, Chronic Kidney Disease (CKD) is a type of chronic disease which means it happens slowly over a period of time and persists for a long time thereafter. Chronic Kidney Disease (CKD) is a type of chronic disease which means it happens slowly over a period of time and persists for a long time thereafter. The main focus in this paper is on the classification techniques, that is, tree-based decision tree, random forest, and logistic regression has been analyzed. Different measure has been used for comparison between algorithms for the dataset collected from standard UCI repository.

04. Imesh Udara Ekanayake , et al. "Chronic Kidney Disease Prediction Using Machine Learning Methods." Moratuwa Engineering Research Conference (MERCon) 2020:

In this paper, —Chronic Kidney Disease (CKD) or chronic renal disease has become a major issue with a steady growth rate. It is important to have effective methods for early prediction of CKD. Out of the 11 machine learning methods considered, the extra tree classifier and random forest classifier are shown to result in the highest accuracy and minimal bias to the attributes. The research also considers the practical aspects of data collection and highlights the importance of incorporating domain knowledge when using machine learning for CKD status prediction.

05. Yedilkhan Amirgaliyev ., et al. "Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods." (2018).

In this paper ,the effects of using clinical features to classify patients with chronic kidney disease by using support vector machines algorithm is investigated. The chronic kidney disease dataset is based on clinical history, physical examinations, and laboratory tests. Experimental results showed over 93% of success rate in classifying the patients with kidney diseases based on three performance metrics i.e., accuracy, sensitivity and specificity.

06. Dulhare, Uma N., and Mohammad Ayesha. "Extraction of action rules for chronic kidney disease using Naïve bayes classifier." ***2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*****. IEEE, 2016.**

In this paper, the UCI dataset for CKD with 25 attributes. The missing values in the dataset were replaced using the mode value of the attributes takenacross the dataset.A naive Bayes Classifier was built to predictCKD. The 'One Rule' algorithm is used for dimensional reduction which reduced almost 80% of the attributes in the dataset. Normal Naive Bayes Classifier is compared with a Naive Bayes Classifier with OneR where the latterimproved the accuracyby 12.5%.

07. Bidri Deepika, Vasudeva Rao KR, Dharmaj N Rampure ., "Early Prediction of Chronic Kidney Disease by using Machine Learning Techniques" ***American Journal of Computer Science and Engineering Survey*** **(2020): 49-55.**

In this paper, Chronic Kidney Disease (CKD) is one of the types of kidney disease, which results in a gradual loss of kidney function. This phenomenon can be observed over a period of months

or years due to several living conditions of patients. The goal is to build a real time application by using the machine learning techniques (Naive Bayes and KNN algorithms), to detect the CKD at an early stage.

08. Pramila Arulanthu. " Predicting the Chronic Kidney Disease using Various Classifiers." ***International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*** **(2019):**

In this paper, we have tried the feature selection method which is used to reduce the attributes and select the more essential attributes only. We can classify the data, using four classifiers namely JRip, SMO, Naive Bayes and IBK. Finally we can compare the results of reduced attribute dataset and original dataset result using these four classifiers. Thus we can find the correct classifier and the best classifier. The classification is the most important part of the processes and it is done using data mining technique based on the machine learning. The classification can be used to predict group membership for data instances.

09. Veenita Kunwar. " Chronic Kidney Disease analysis using data mining classification techniques." ***IEEE access*** **(2016): 1-5.**

In this paper, an adopted K-means Clustering algorithm with a single mean vector of centroids, to classify and make clusters of varying probability of likeliness of suspect being prone to CKD. They observed and stated that the suspectsfalling in clustersK1 or K3 are surely suffering from CKD. The probability of a suspectlying in K2 cluster to fall in the class of CKD is 0.50545, which implies that the suspect cannot be classified by their L-factor classifier. However, suspects from clusters K1 & K3 were found to be falling in CKD class with full probability.

10. Akash Maurya , Rahul Wable et al. " Chronic Kidney Disease Prediction and Recommendation of Suitable Diet Plan by using Machine Learning" ***Journal of medical systems*** **39.10 (2019): 1-9.**

In this paper, the dataset consists 24 attributes, forming five main categories. Chronic kidney disease (CKD) is a type of kidney disease in which there is gradual loss of kidney function over a period of months or years. Prediction of this disease is one of the most important problems in

medical fields. So automated tool which will use machine learning techniques to determine the patient's kidney condition that will be helpful to the doctors in prediction of chronic kidney disease and hence better treatment. The proposed system extracts the features which are responsible for CKD, then machine learning process can automate the classification of the chronic kidney disease in different stages according to its severity. The objective is to use machine learning algorithm and suggest suitable diet plan for CKD patient using classification algorithm on medical test records. Diet recommendation for patient will be given according to potassium zone which is calculated using blood potassium level to slow down the progression of CKD.

2.2 PROBLEM STATEMENT DEFENITION:

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

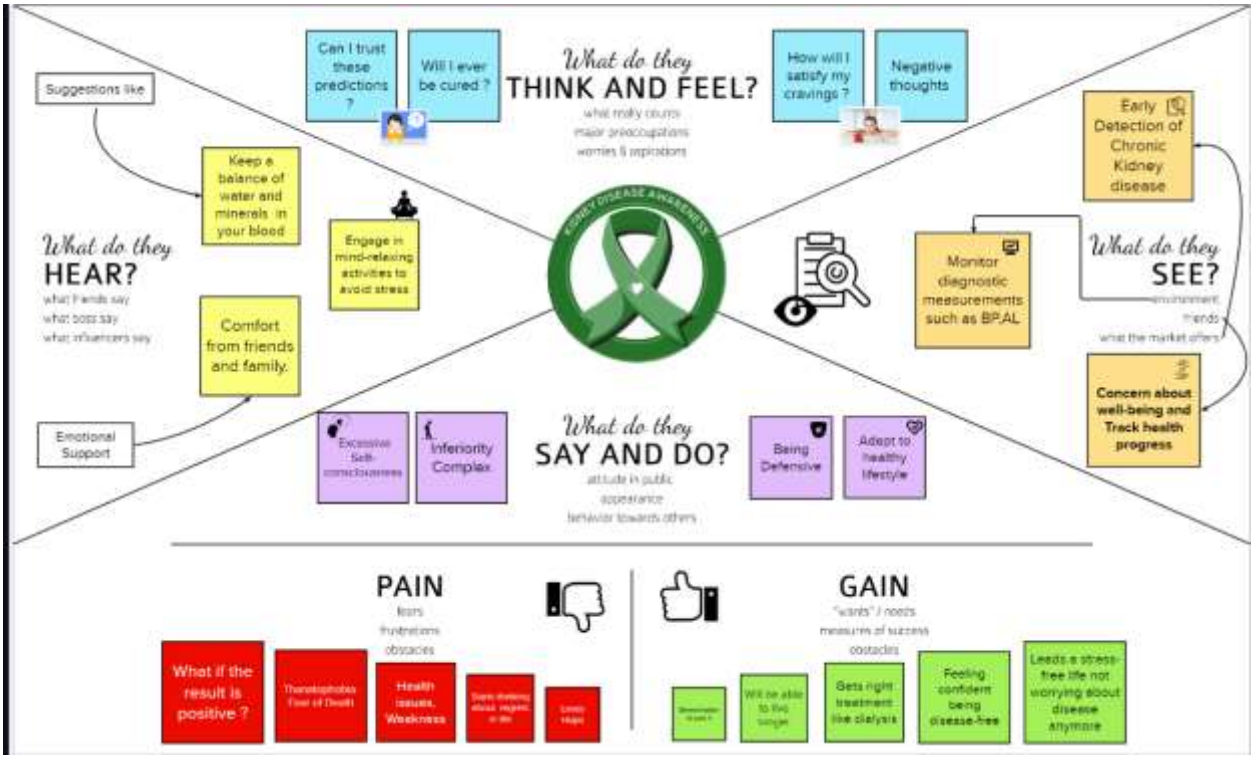
- Irregular lifestyle of current days leads to many disease chronic diseases. One of them being Chronic Kidney Disease (CKD), it has been of a growing concern, kidney is one of the most important organs in the body required for filtering blood, once a person has lost their kidneys, they could survive 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 AND BRAINSTROMING

3.2.1Define your problem statement :

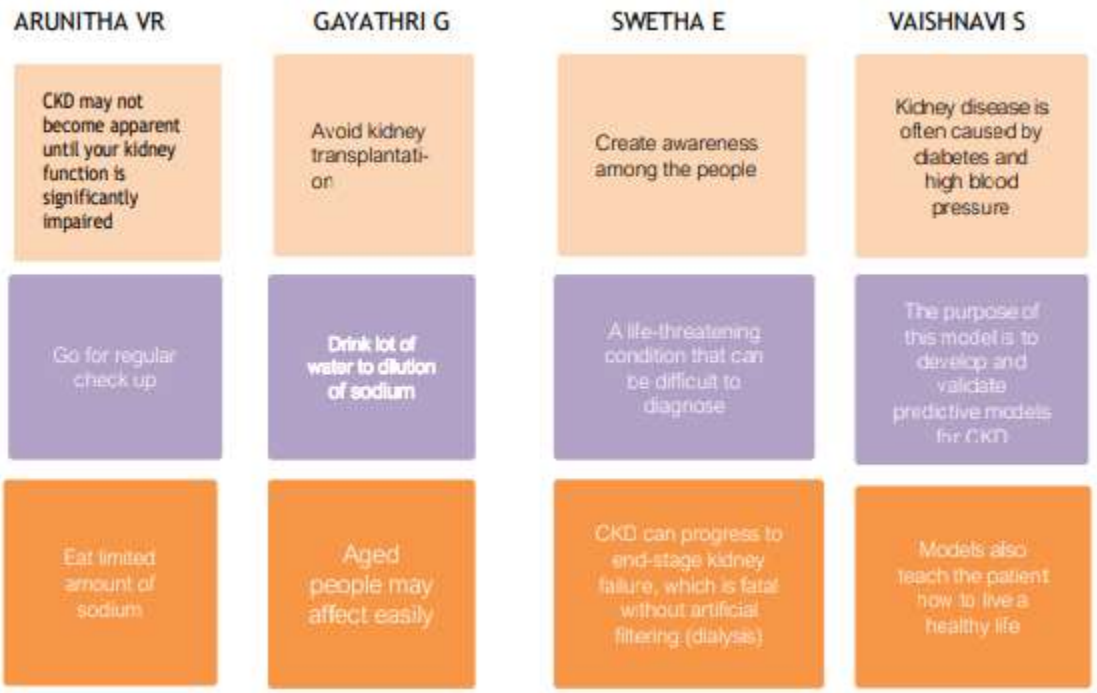
Chronic kidney disease (CKD) is a life -threatening condition that can be difficult to diagnose early because there are no symptoms. The purpose of the proposed study is to develop and validate a predictive model for the prediction of chronic kidney disease .

PROBLEM:

Chronic kidney disease (CKD) is a life -threatening condition that can be difficult to diagnose early because there are no symptoms. To avoid these cases, CKD should be predicted earlier by ML algorithms.

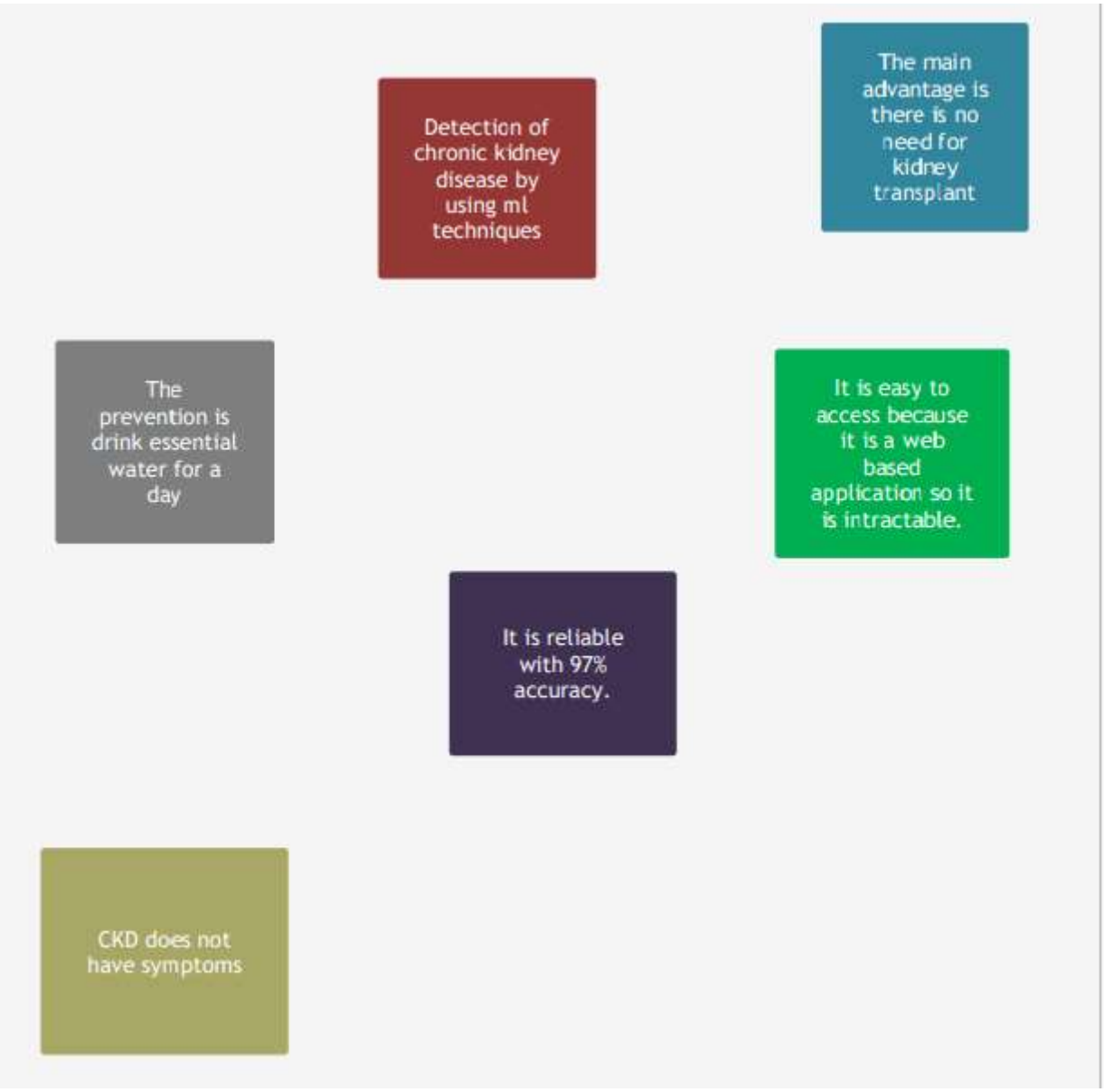
BRAINSTROM:

Chronic kidney disease, also called chronic kidney failure, describes the gradual loss of kidney function. Chronic kidney disease can progress to end -stage kidney failure, which is fatal without artificial filtering (dialysis) or a kidney transplant. To avoid these conditions , early prediction of chronic kidney disease helps the most to find the disease earlier and can make it curable.



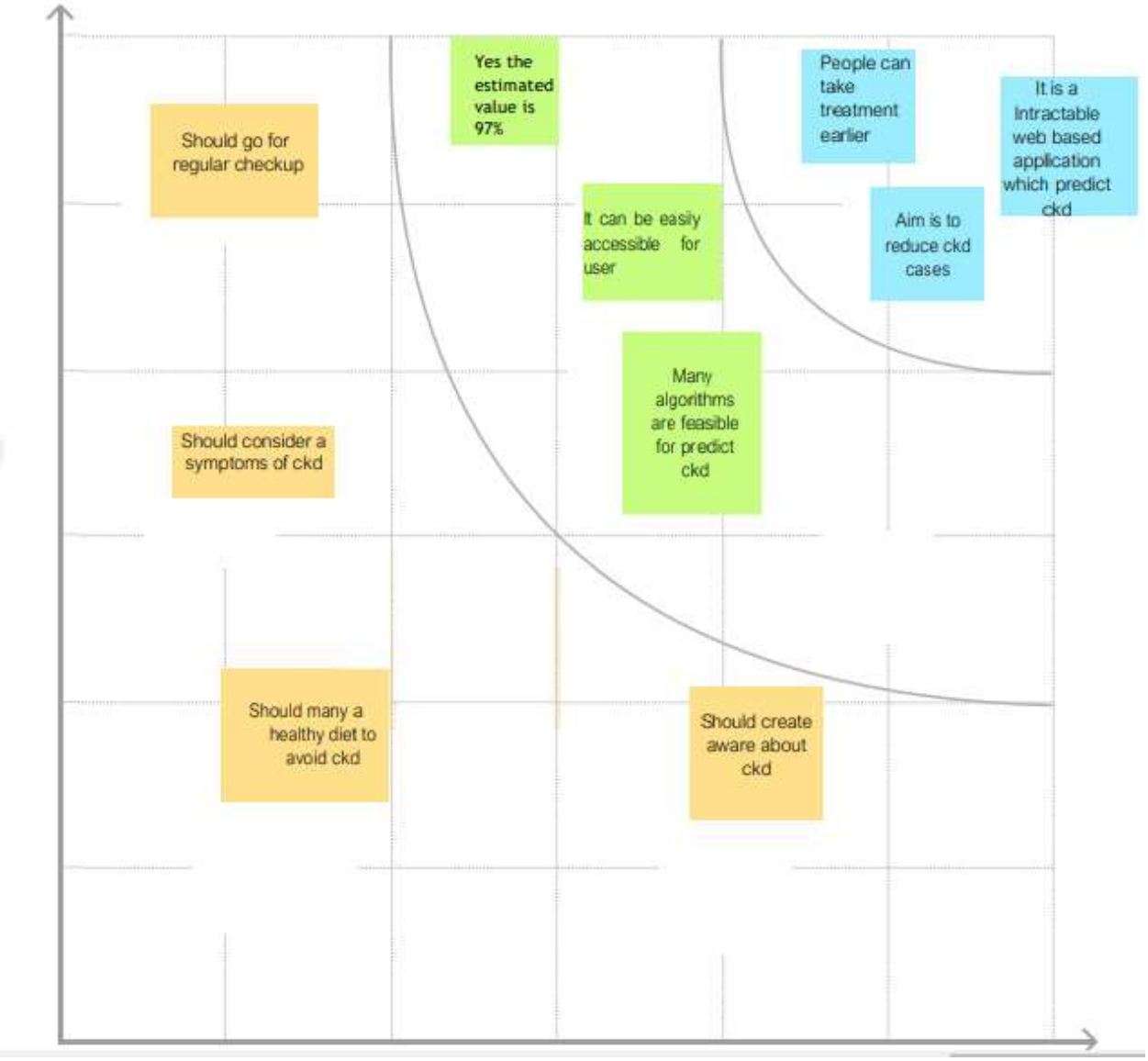
GROUP IDEAS:

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.



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 are important and feasible.



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Chronic kidney disease (CKD) is a condition in which the kidneys are damaged and cannot filterblood as well as theyshould. Chronic diseases, which are the cause of 71% of all deaths worldwide have devastating health consequences for individuals, families, and societies, as well as creating serious financial and economic risks to the economies of developed and developing countries and threatening the sustainability of health systems. It has been found that more than 60% of some chronic patient populations, such as those with cancer, cardiovascular disease, and stroke, have catastrophic health expenditures. Chronic diseases account for approximately 80% of all years of disability around the world
2.	Solution description	Either dipstick urinalysis or urine albumin-creatinine ratio (ACR) could also be used for the early detection of CKD. Various diagnostic measurements like Blood Pressure (Bp), Albumin (Al) etc., of the patients are collected and the data is processed and given to a machinelearning model that will predictif the patienthas CKD or not. 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.
3.	Novelty	We aim to findthe best machinelearning model forthe early prediction of chronic kidneydisease by analyzing the essential parameters and comparing their predictive accuracies. Then collaborate the best machinelearning model to an interactive user-interface which helps in the early detection of CKD and provide cure.
4.	CustomerSatisfaction	The main aim of this application is early prediction and proper treatments can possibly stop or slow the progression of this disease to the end stage. Consult a dietitian regarding useful changes in diet. Dietary changes may include limiting protein, eating foods that reduce blood cholesterol levels, and limiting sodium (salt) and potassium intake.

5.	Business Model (Revenue Model)	We can generate revenuethrough direct customers or can also collaborate with the health care sector and generate revenue from their customers. We can able to discuss the problem for the CKD through online or offline mode.
6.	Scalabilityof the Solution	We can build variousmodels using machine learning algorithms and compare them to find the best accurate model. We can also use image data and apply Deep Learning techniques, Probabilistic Neural Networks(PNN),and Multilayer Perceptron(MLP) etc., which will provide an improved accuracy than the machine learning techniques. Early detection and appropriate treatment are important in slowing the disease process, with the goal of preventing or delaying kidney failure. You will need to keep your medical appointments, take your medications as prescribed, stick to a healthy diet and monitor your blood pressure and blood sugar.

3.4 PROBLEM SOLUTION FIT

I.CUSTOMER SEGMENT: A Patient of Logesh Lokesh is a 70-year-old diabetic patient, and has been observing many unusual symptoms like vomiting, sleep problems, fatigue and weakness, Muscle cramps. He also has a family history of chronic kidney disease and wants to know if he is hereditarily affected by CKD.	**VII.BEHAVIOUR:** Lokesh has been trying to change his daily routines and he has decided to take healthier foods and also he had taken a decision of doing exercises regularly.
II.PROBLEMS OF CKD: People with diabetes, hypertension, heart disease and family history are at increased risk for developing CKD and depending on the region, it may also be important to consider comorbidities, environmental exposures or genetic risk factors.	**VIII.CHANGES OF BEHAVIOUR:** **ONLINE:** Lokesh browsed online about the unusual symptoms and developed his suspicions for CKD and tried to alleviate his ailments using home treatments **OFFLINE**: Lokesh attends awareness programs to find out more about CKD and tries to conclude if he is affected, he later approaches a doctor to confirm his suspicions of CKD.
III.TRIGGERS: Lokesh has noticed his change in his appearance which has caused his to develop insecurities. He has also lost his appetite, and developed insomnia.	**IX.CAUSES:** People with diabetes, hypertension, heart disease and family history are at increased risk for developing CKD and depending on the region, it may also be important to consider comorbidities, environmental exposures or genetic risk factors.

IV.CHANGES APPEARED: **BEFORE:** Lokesh was incompetent to decide if he has been affected by the disease and this caused him to feel distressed and uneasy. **AFTER:** Lokesh, after he has been identified has a better understanding of what the disease is and feels more secure about the decisions, he should take to face CKD.	**X.SOLUTION FOR CKD:** Targeted screening and detection of probable CKD can enable GPs to confirm a diagnosis and assess the level of risk of cardiovascular disease and progression of CKD. This in turn guides management. Early detection and subsequent timely management can slow down or even prevent deterioration in kidney function and help to improve cardiovascular outcomes

V.STEP TAKEN: Lokesh can consult a doctor, stating that he has been facing some unusual symptoms recently, following the prescription he could take tests to discover if he has CKD or not.	
VI.CUSTOMER NEEDS: People with less awareness get diagnosed with CKD only when the symptoms get worse and becomes more difficult to treat and is more deadly. CKD is a condition in which the kidneys are damaged and cannot filter blood as well as they should. Because of this, excess fluid and waste from blood remain in the body and may cause other health problems, such as heart disease and stroke.	

4. REQUIREMENT ANALYSIS

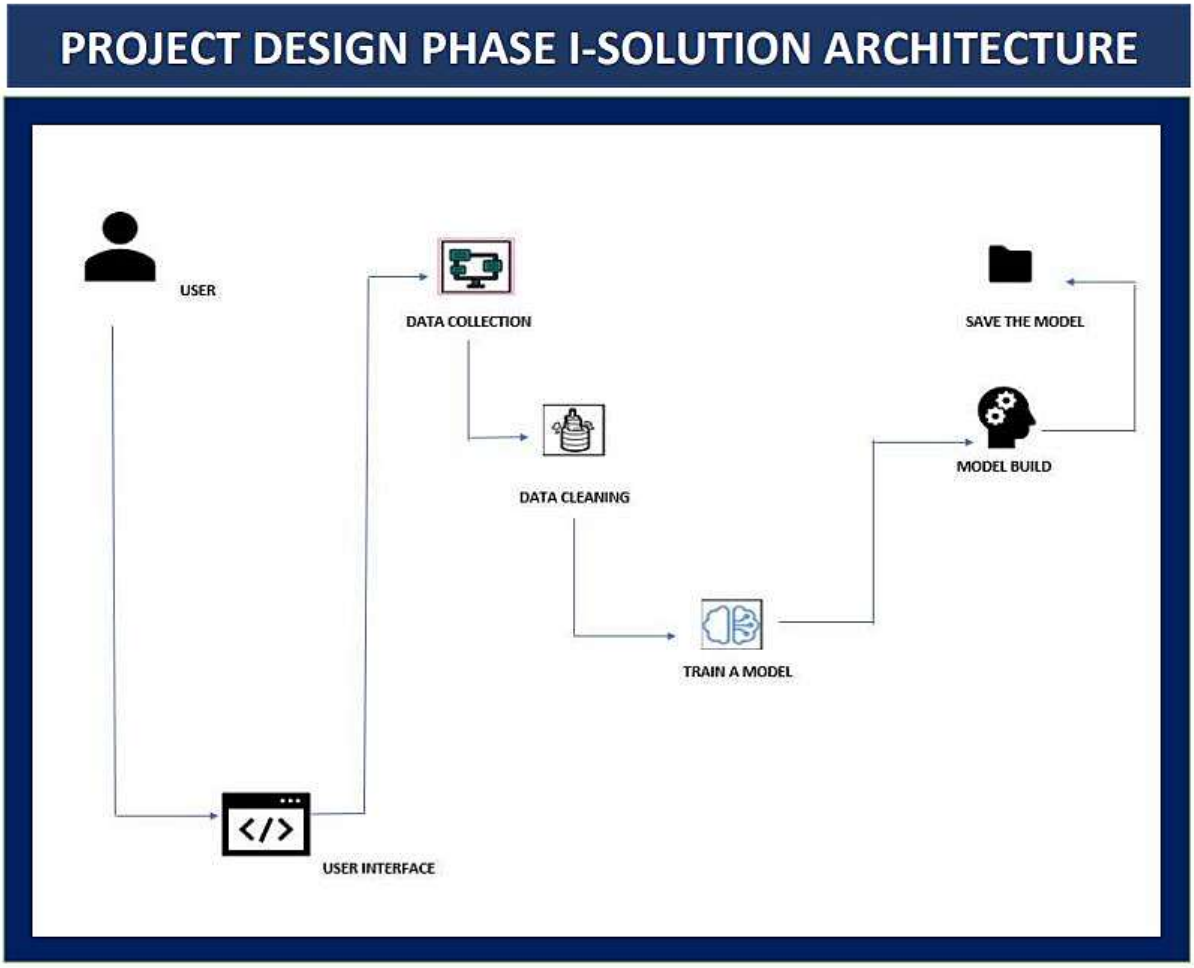
4.1 FUNCTIONAL REQUIREMENTS

S.No.	Functional Requirement	Sub Requirement
1	Home Page	The Home page consists of sign in and sign out page and also they can able to analyze the disease. Test vitals required for prediction of chronic kidneydisease. Sign Up and Login options for the user accordingly. If new user the user needs to SIGN UP and gets directed to the registration page else, redirected to the LOGINpage.
2	Registration	The user needs to enter fewaccount credentials andother personal details required for registering.
3	Confirmation	The user need to fill the details to confirm whether it is is corrreb
4	Login	If the customer would like to open into a login page, they needs to enterthe login credentials such as name, username, and password.
5	verification	It will verify the user details and verifies the email.
6	Symptoms Form	The user must entertheir symptoms andanswer the questions properly for prediction.
7	Result	The page displays the test report. Ifpositive - displays the test reportalong the necessary measures to be taken to cure the disease. If negative – displays the test reportalong with thepreventive measures forthe disease.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

PHASE 1

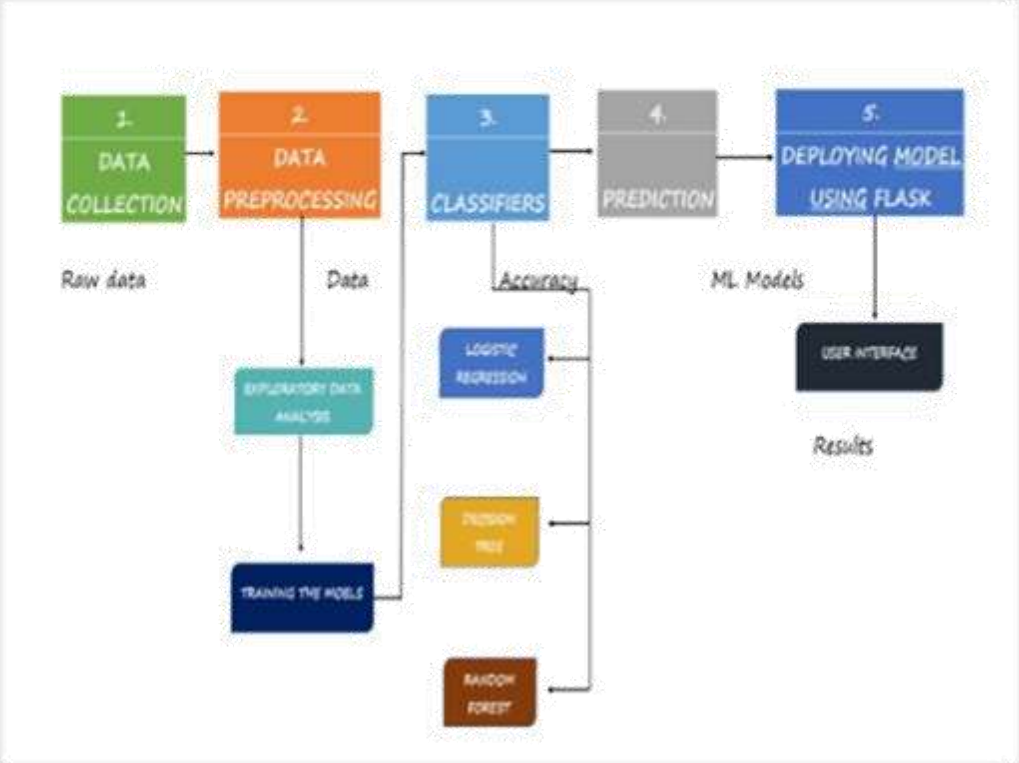


PHASE 2

Project Design Phase-II

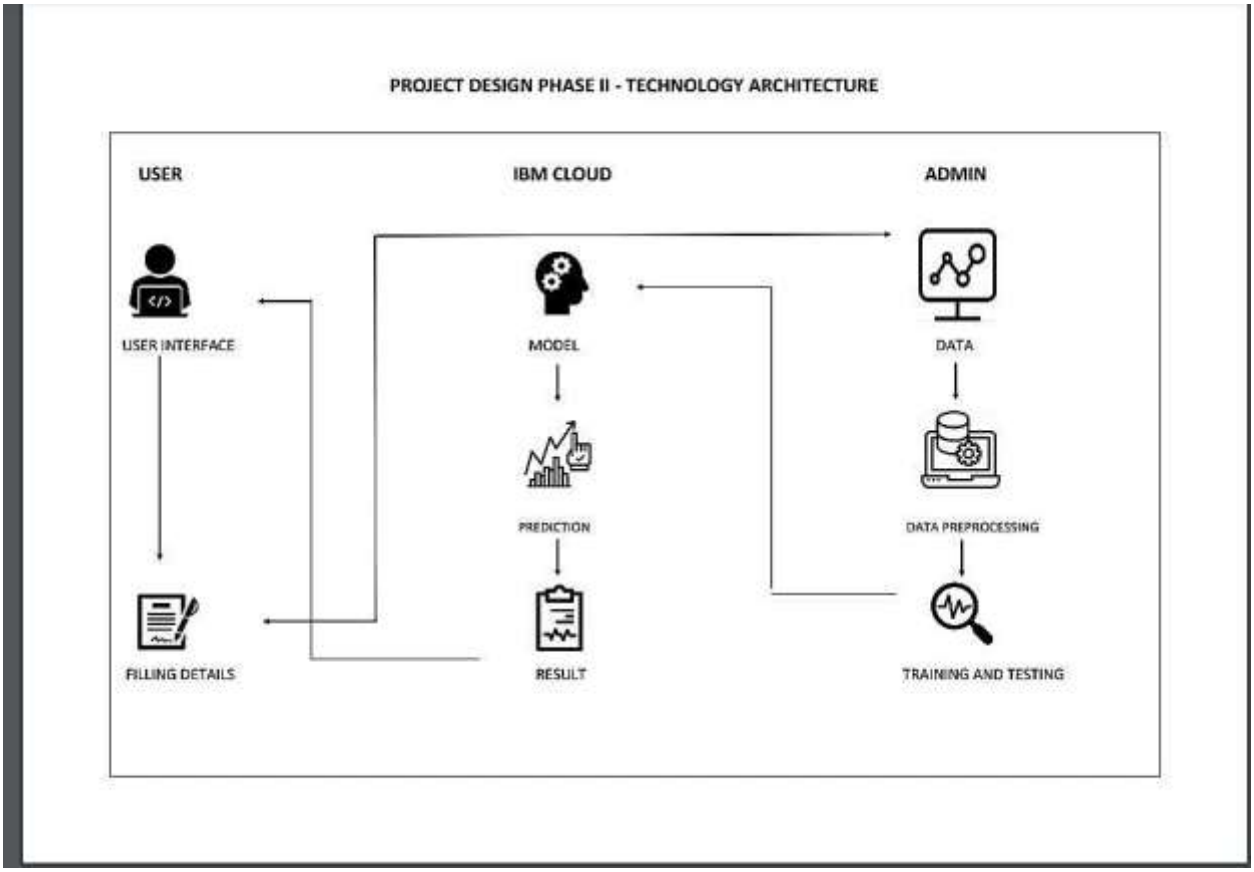
Data Flow Diagram

Early Detection of Chronic Kidney Disease using Machine Learning

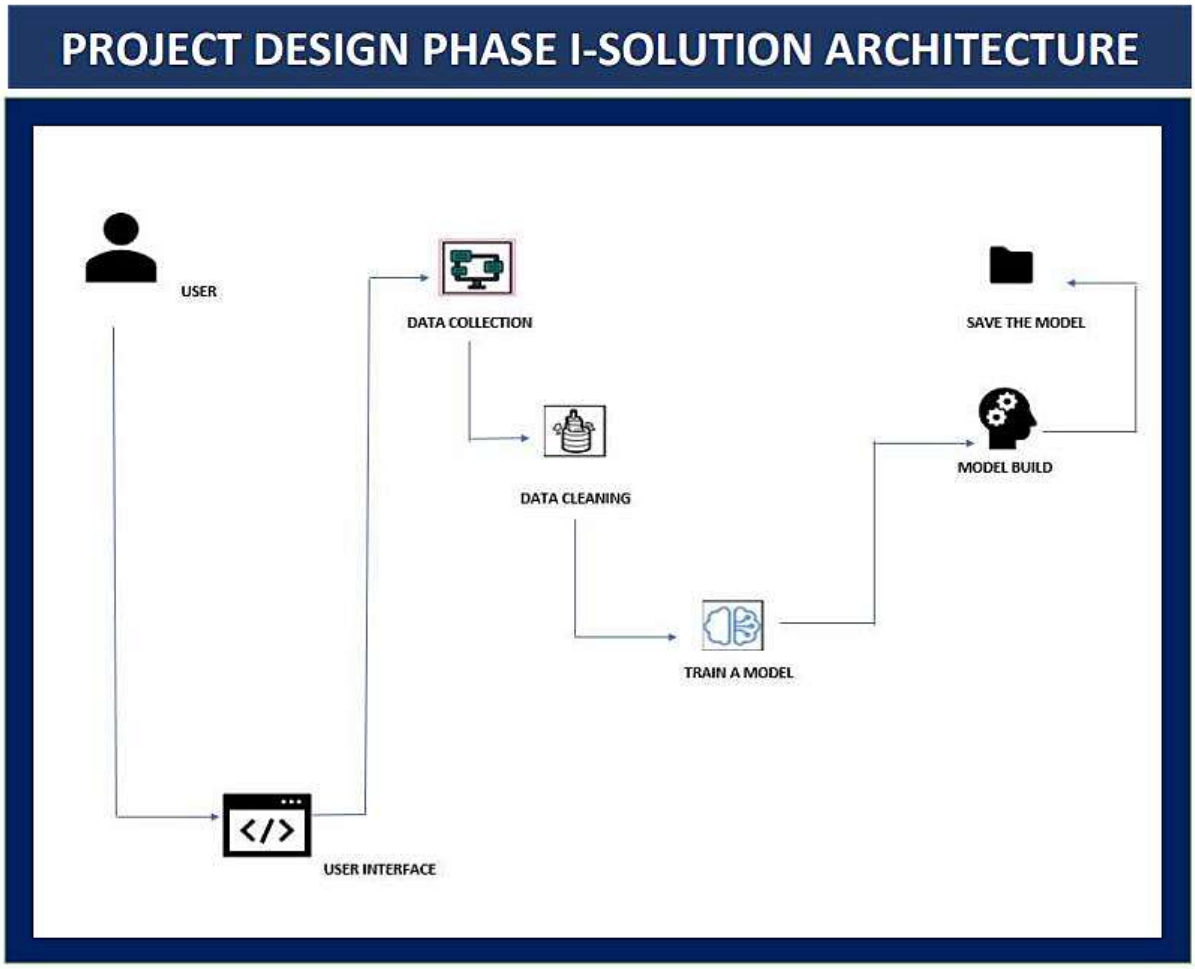


5.2 SOLUTION AND TECHNICAL ARCHITECTURE

PHASE 1



PHASE 2


PROJECT DESIGN PHASE I-SOLUTION ARCHITECTURE
USER
DATA COLLECTION
SAVE THE MODEL
DATA CLEANING
MODEL BUILD
TRAIN A MODEL
USER INTERFACE

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Task
Sprint 1 (15th to 19th October)	• Dataset is downloaded and data analysis is performed. • Data is cleaned, handled the missing values, and label encoding is performed.
Sprint 2 (20st to 28th October)	• Login in using login credentials. • Register into the diagnosis tool. • Dataset is split into train and test • Models are built and ready for prediction andclassification.
Sprint 3 (29th October to 3th November)	• Symptoms and test vitals viewed as user input. • Evaluating the model and saving the model. • Storing the login details.
Sprint 4 (4th to 10th November)	• Test results are built and saved. • ML model to be trained on IBM Watson. • Integrating the website and HTML model.

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	Task-1	To design an ML model, the dataset has to download and load into the platform for further analysis	4	Low	Swetha E
Sprint-1	Importing the packages	Task-2	We import the libraries and packages required for the data analysis.	3	High	Swetha E Vaishnavi S
Sprint-1	Data Pre-processing	Task-3	Data cleaning, handling missing values and label encoding were performed in this sprint	8	Medium	Swetha E Vaishnavi S
Sprint-2	Building Web application	USN-1	As a user, they can interact with the UI only to build a web page	5	High	Arunitha V R

Sprint -2	Get the input data	USN-2	As a new users, they have to fill in the details and parameters needed for the prediction	5	High	Arunitha VR
Sprint -2	Splitting the dataset	Task-4	Splitting dataset into train and test split.	3	Medium	Swetha E
Sprint -2	Training the model and testing the model	Task-5	Train the model and test that model	12	High	Swetha E Vaishnavi S
Sprint -3	Build the model	USN-3	Build the ML model	5	High	Swetha E Vaishnavi S
Sprint -3	Evaluating the model	Task-6	Evaluating model with the better accuracy	3	Low	Vaishnavi S

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-3	Creating User Database	Task-7	Storing the user's needed details in the database.	12	High	Vaishnavi S

Sprint-4	Prediction Page	USN-4	As a user, they have to view the predicted result in a interactive web page	5	Low	Arunitha V R
Sprint-4	Train model on IBM Cloud	Task-8	Train the ML model on IBM Watson studio.	7	Medium	Swetha E
Sprint-4	Flask Integration	Task-9	Integrating the HTML files with the ML model.	8	High	Arunitha V R Swetha E

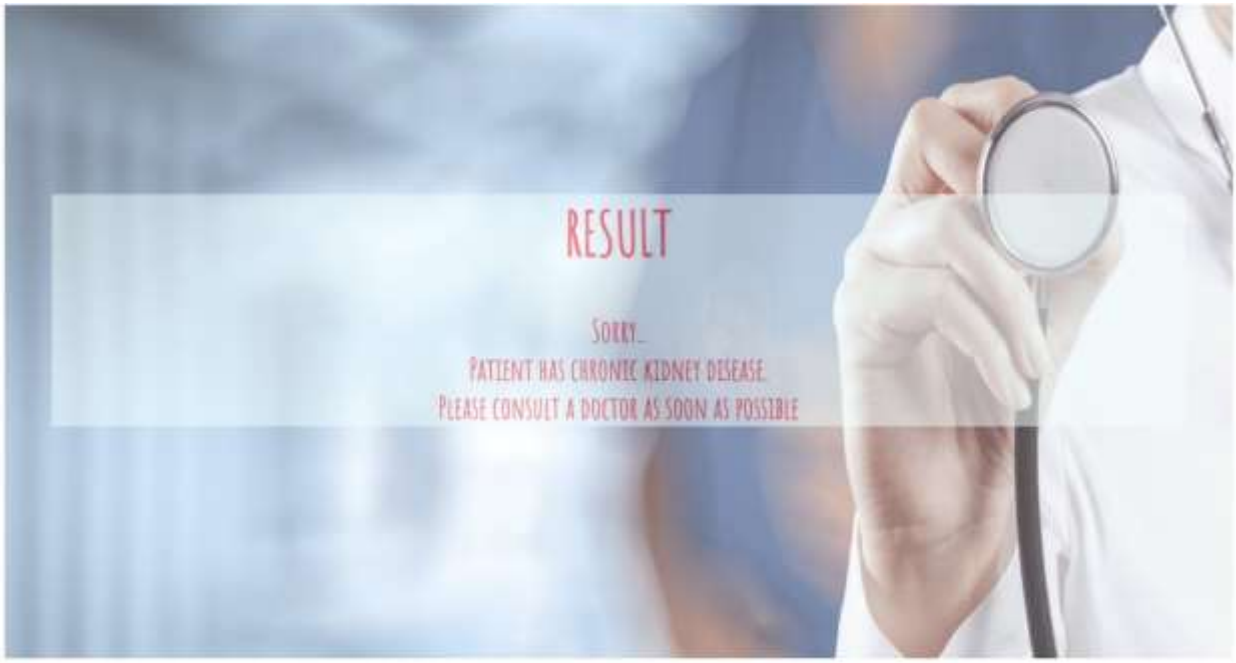
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	20	4 Days	15 Oct 2022	19 Oct 2022	20	19 Oct 2022
Sprint-2	20	9 Days	20 Oct 2022	28 Oct 2022	20	28 Oct2022
Sprint-3	20	6 Days	29 Oct 2022	3 Nov 2022	20	3 Nov 2022
Sprint-4	20	7 Days	4 Nov 2022	10 Nov 2022	20	10 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)

7. CODING & SOLUTIONING


RESULT
Sorry..
Patient has chronic kidney disease.
Please consult a doctor as soon as possible


RESULT
Hurray!!
Patient does not have chronic kidney disease.
Go home and celebrate!!

8. ADVANTAGES

This system helps to detect the presence of chronic kidney disease earlier by the help of analysing input data from the patient's blood samples.

By earlier prediction of chronic kidney disease, People can take treatment earlier as soon as possible.

With the help of the data obtained from previous patients that had chronic kidney disease or otherwise to aid in the creation of the prediction model, the model can predict the outcome of the data provided by the user before the subsequent onset of the disease before the occurrence of the same.

Thus, reducing the threats of this disease.

9. CONCLUSION

According to the findings of the study, the decision tree approach and logistic regression can be used to predict chronic kidney disease more accurately. According to the study, their precision was 96.25 percent, and their accuracy was 97 percent.

Compared to prior research, the accuracy percent of the models used in this investigation is considerably higher, indicating that the models used in this study are more reliable than those used in previous studies. When cross validation measurements are used in the prediction of chronic kidney disease, the LR method outperforms the other processes.

Future research may build on this work by developing a web application that incorporates these algorithms and using a bigger dataset than the one utilized in this study. This will aid in the achievement of improved outcomes as well as the accuracy and efficiency with which healthcare practitioners can anticipate kidney issues.

This will enhance the dependability of the framework as well as the framework's presentation. The hope is that it would encourage people to seek early treatment for chronic renal disease and to make improvements in their lives.

10.APPENDIX:

SOURCE CODE:

prediction/app.py

```
import NumPy as np
from flask import Flask, request, jsonify, render_template

import requests
import json

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "puYxlBXzc8bCOJBEKcpSk8ZC-w9D97xSMvQiOh2y1wQN"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line



app = Flask(__name__)

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

@app.route('/y_predict',methods=['POST'])
def y_predict():
```

```
    age=request.form["age"]
    bp=request.form["bp"]
    sg=request.form["sg"]
    al=request.form["al"]
    au=request.form["su"]
    rbc=request.form["rbc"]
    pc=request.form["pc"]
    pcc=request.form["pcc"]
    ba=request.form["ba"]
    bgr=request.form["bgr"]
    bu=request.form["bu"]
    sc=request.form["sc"]
    sod=request.form["sod"]
    pot=request.form["pot"]
    hemo=request.form["hemo"]
    pcv=request.form["pcv"]
    wc=request.form["wc"]
    rc=request.form["rc"]
    ht=request.form["htm"]
    dm=request.form["dm"]
    cad=request.form["cad"]
    appet=request.form["appet"]
    pe=request.form["pe"]
    ane=request.form["ane"]

    payload_scoring = {"input_data": [{"field": [['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr',
'bu',
        'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
        'appet', 'pe', 'ane']], "values":
[[25,60,1.02,0,0,0,0,0,0,119,27,0.5,137.53,4.63,15.2,40,9200,5.2,0,0,0,1,0,0
```

```
]]}]}

    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/23a45911-67a8-4070-be72-
46c6eaa3fda7/predictions?version=2022-11-12', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    prediction=response_scoring.json()
    print(prediction)

    pred=prediction['predictions'][0]['values'][0][0]

    if(pred == 0):
        output=False
        print("Patient does not have ckd")



    else:
        output=True
        print("Patient has  ckd")

    if(output ==False):
      return render_template('positive.html')
    else:
      return render_template('negative.html')


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

app.py

```
from flask import Flask, render_template, session, redirect
import pymongo

app = Flask(__name__)

# Database
# client = pymongo.MongoClient('localhost', 27017)
client = pymongo.MongoClient("mongodb+srv://admin-Vaishnavi:test1234@cluster0.cghftwz.mongodb.net/?retryWrites=true&w=majority")
db = client.ckdPatientDB
# collection = db.patients
# Routes
from user import routes

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

@app.route('/details/')
def details():
  users = db.patients.find()
  return render_template('details.html', users = users)
```

user/models.py

```
from flask import Flask, jsonify, request,session
from app import db
import uuid

class User:
```

```
def patient(self):
  print(request.form)
  # Create the user object
  user = {
    "_id": uuid.uuid4().hex,
    "age": request.form.get('age'),
    "bp": request.form.get('bp'),
    "sg": request.form.get('sg'),
    "al": request.form.get('al'),
    "su": request.form.get('su'),
    "rbc": request.form.get('rbc'),
    "pc": request.form.get('pc'),
    "pcc": request.form.get('pcc'),
    "ba": request.form.get('ba'),
    "bgr": request.form.get('bgr'),
    "bu": request.form.get('bu'),
    "sc": request.form.get('sc'),
    "sod": request.form.get('sod'),
    "pot": request.form.get('pot'),
    "hemo": request.form.get('hemo'),
    "pcv": request.form.get('pcv'),
    "wc": request.form.get('wc'),
    "rc": request.form.get('rc'),
    "htm": request.form.get('htm'),
    "dm": request.form.get('dm'),
    "cad": request.form.get('cad'),
    "appet": request.form.get('appet'),
    "pe": request.form.get('pe'),
    "ane": request.form.get('ane'),
  }
```

```
    db.patients.insert_one(user)
    return jsonify(user), 200
```

user/routes.py

```
from flask import Flask
from app import app
from user.models import User

@app.route('/user/patient', methods=['POST', 'GET'])
def patient():
  return User().patient()
```

templates/CDK.html

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chronic Kidney Disease</title>

    <link rel="stylesheet" href="{{ url_for('static',filename='css/style.css') }}">
    </link>

</head>

<body>
    <div class="header">
        <img src="{{ url_for('static',filename='css/CKD.jpeg') }}" class="thumbnail">
        <h1>CHRONIC KIDNEY DISEASE DETECTION</h1>
```

```
</div>
<div class="forms">
    <img src="{{ url_for('static',filename='css/background.jpeg') }}" class="image">
    <form action="{{ url_for('y_predict')}}" method="post">
        <div class="flex">
            <div class="forminput">
                <label>Age </label>
                <input type="number" step="any" id='age' name='age' />
                <br />
            </div>
            <div class="forminput">
                <label>Blood Pressure </label>
                <input type="number" step="any" id='bp' name='bp' />
                <br />
            </div>
            <div class="forminput">
                <label>Specific Gravity </label>
                <input type="number" step="any" id='sg' name='sg' />
                <br />
            </div>
            <div class="forminput">
                <label>Albumin </label>
                <input type="number" step="any" id='al' name='al' />
                <br />
            </div>
        </div>
        <div class="flex">

            <div class="forminput">
                <label>Sugar </label>
                <input type="number" step="any" id='su' name='su' />
```

```
        <br />
    </div>
    <div class="forminput">
        <label>Red Blood Cells </label>
        <input type="number" step="any" id='rbc' name='rbc' />
        <br />
    </div>
    <div class="forminput">
        <label>Pus cells </label>
        <input type="number" step="any" id='pc' name='pc' />
        <br />
    </div>
    <div class="forminput">
        <label>Pus Cell clumps </label>
        <input type="number" step="any" id='pcc' name='pcc' />
        <br />
    </div>
</div>
<div class="flex">

    <div class="forminput">
        <label>Bacteria </label>
        <input type="number" step="any" id='ba' name='ba' />
        <br />
    </div>
    <div class="forminput">
        <label>Blood Glucose Random </label>
        <input type="number" step="any" id='bgr' name='bgr' />
        <br />
    </div>
    <div class="forminput">
```

```
        <label>Blood Urea </label>
        <input type="number" step="any" id='bu' name='bu' />
        <br />
    </div>
    <div class="forminput">
        <label>Serum Creatinine </label>
        <input type="number" step="any" id='sc' name='sc' />
        <br />
    </div>
</div>
<div class="flex">

    <div class="forminput">
        <label>Sodium </label>
        <input type="number" step="any" id='sod' name='sod' />
        <br />
    </div>
    <div class="forminput">
        <label>Pottasium </label>
        <input type="number" step="any" id='pot' name='pot' />
        <br />
    </div>
    <div class="forminput">
        <label>Heamoglobin </label>
        <input type="number" step="any" id='hemo' name='hemo' />
        <br />
    </div>
    <div class="forminput">
        <label>Packed cells volume </label>
        <input type="number" step="any" id='pcv' name='pcv' />
        <br />
```

```
    </div>
</div>
<div class="flex">

    <div class="forminput">
        <label>White Blood Cells Count </label>
        <input type="number" step="any" id='wc' name='wc' />
        <br />
    </div>
    <div class="forminput">
        <label>Red Blood Cells Count </label>
        <input type="number" step="any" id='rc' name='rc' />
        <br />
    </div>
    <div class="forminput">
        <label>Hypertension </label>
        <input type="number" step="any" id='htm' name='htm' />
        <br />
    </div>
    <div class="forminput">
        <label>Diabetes Mellitus </label>
        <input type="number" step="any" id='dm' name='dm' />
        <br />
    </div>
</div>
<div class="flex">

    <div class="forminput">
        <label>Coronary artery disease </label>
        <input type="number" step="any" id='cad' name='cad' />
        <br />
```

```
            </div>
            <div class="forminput">
                <label>Appetite </label>
                <input type="number" step="any" id='appet' name='appet' />
                <br />
            </div>
            <div class="forminput">
                <label>Pedal edema </label>
                <input type="number" step="any" id='pe' name='pe' />
                <br />
            </div>
            <div class="forminput">
                <label>Anemia </label>
                <input type="number" step="any" id='ane' name='ane' />
                <br />
            </div>
        </div>

        <div class="flex">
            <div class="forminput">
                <button type="submit" class="btn btn-warning btn-block btn-Large"
name="Submit" placeholder="Submit"
                    onclick="prediction()">DIAGNOSE</button>
            </div>

        </div>
        {{ prediction_text }}
    </form>
  </div>

</body>
```

```
</html>
```

templates/positive.html

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Negative</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Amatic+SC&display=swap');

        .image {
            width: 100%;
            height: 690px;
            z-index: -1;
            opacity: 0.7;
            position: ABSOLUTE;
            top: 0px;
            left: 0px;
            margin: auto 50px;
        }

        .container {
            position: absolute;
            display: flex;
            flex-flow: row nowrap;
            width: 100%;
            height: fit-content;
        }
```

```
        .sub-container {
            position: relative;
            width: 90%;
            text-align: center;
        }

        .card {
            margin-top: 200px;
            text-align: center;
            background-color: azure;
            opacity: 0.5;
            margin-left: 100px;
        }

        body {
            font-family: 'Amatic SC', cursive;
            font-size: xx-large;
            font-weight: bold;
            color: red;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="sub-container">
            <img src="{{ url_for('static',filename='css/CKD1.jpeg') }}" class="image">
            <div class="card">
                <h1>RESULT</h1>
                <p>Sorry..<br />Patient has chronic kidney disease.
                    <br />Please consult a doctor as soon as possible
```

```
            </p>
          </div>
        </div>
    </div>
</body>
</html>
```

templates/negative.html

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Positive</title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Amatic+SC&display=swap');

        .image {
            width: 100%;
            height: 690px;
            z-index: -1;
            opacity: 0.7;
            position: ABSOLUTE;
            top: 0px;
            left: 0px;
            margin: auto 50px;
        }
```

```
        .container {
            position: absolute;
            display: flex;
            flex-flow: row nowrap;
            width: 100%;
            height: fit-content;
        }

        .sub-container {
            position: relative;
            width: 90%;
            text-align: center;
        }

        .card {
            margin-top: 200px;
            text-align: center;
            background-color: azure;
            opacity: 0.5;
            margin-left: 100px;
        }

        body {
            font-family: 'Amatic SC', cursive;
            font-size: xx-large;
            font-weight: bold;
            color: green;
        }
    </style>
</head>
```

```
<body>
    <div class="container">
        <div class="sub-container">
            <img src="{{ url_for('static',filename='css/CKD1.jpeg') }} "class="image">
            <div class="card">
                <h1>RESULT</h1>
                <p>Hurray!!<br />Patient does not have chronic kidney disease.
                    <br />Go home and celebrate!!
                </p>
            </div>
        </div>
    </div>
</body>

</html>
```

static/css/styles.css

```
@import url('https://fonts.googleapis.com/css2?family=Amatic+SC&display=swap');
@import
url('https://fonts.googleapis.com/css2?family=Urbanist:ital,wght@1,100&display=swap');
@import url('https://fonts.googleapis.com/css2?family=Amatic+SC&display=swap');

body {
    font-family: 'Amatic SC', cursive;
    font-weight: bolder;

}

h1 {
    width: 100%;
```

```
    font-family: 'Urbanist', sans-serif;
}

.header {
    display: flex;
    width: 100vw;
    box-shadow: 1px 1px 1px 1px;
    position: relative;
    top: -8px;
    left: -8px;
    text-align: center;
    margin: auto;
    background-color: azure
}

.forms {
    position: relative;
    top: 0px;
    margin: auto;
}

.forminput {
    margin: 2px auto;
    text-align: center;
    display: flex;
    flex-flow: column nowrap;
    width: 65%;
}

input {
    margin: auto;
```

```
    width: 80%;
    height: 4vh;
}

form {
    position: absolute;
    top: 25px;
    border: 1px solid black;
    height: 600px;
    width: 100%;
}

.image {
    width: 100%;
    height: 640px;
    z-index: -1;
    opacity: 0.5;
    position: relative;
    top: 0px;
    left: 0px;
    margin: auto;
}

.flex {
    display: flex;
    flex-direction: row;
    height: 100px;
    font-size: x-large;
}

button {
```

```
    width: 70%;
    height: 5vh;
    background-color: darkgray;
    border: none;
    font-size: 20px;
    font-family: 'Urbanist', sans-serif;
    border-radius: 8px;
    margin: 20px auto;
}

.thumbnail {
    height: auto;
    width: 100px;
    margin: 0px 0px 0px 20px ;}
```

GITHUB PROJECT LINK

https://github.com/IBM-EPBL/IBM-Project-9817-1659076624