Early Detection Of Chronic Kidney Disease Using Machine Learning

TEAM ID:PNT2022TMID13778

1. **INTRODUCTION**
    1.1. Project Overview
    1.2. Purpose
2. **LITERATURE SURVEY**
    2.1. Existing problem
    2.2. References
    2.3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    3.1. Empathy Map Canvas
    3.2. Ideation & Brainstorming
    3.3. Proposed Solution
    3.4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    4.1. Functional requirement
    4.2. Non-Functional requirements
5. **PROJECT DESIGN**
    5.1. Data Flow Diagrams
    5.2. Solution & Technical Architecture
    5.3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    6.1. Sprint Planning & Estimation
    6.2. Sprint Delivery Schedule
    6.3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    7.1. Feature 1
    7.2. Feature 2

8. **TESTING**

8.1. Test Cases

8.2. Performance Testing

9. **RESULTS**

9.1. Machine Learning Accuracy

9.2. Chronic kidney disease Application

10. **ADVANTAGES & DISADVANTAGES**

11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project Demo Link

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 anaemia. 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 analysing 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. Thirumalaiselvi [3] 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. Asif salekin and john stankovic [7] 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 neighbours, random forest and neural network to get results. For feature reduction they use wrapper method which detect CKD with high accuracy. Pinar Yildirim [8] 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 [9], 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

[1]. 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.

[2]. 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.

[3]. S.Dilli Arasu and Dr. R.Thirumalaiselvi, “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

[4]. L. Rubini, “Early stage of chronic kidney disease UCI machine learning repository,”2015. [Online].Available:http://archive.ics.uci.edu/ml/datasets/Chronic Kidney Disease.

[5]. S. A. Shinde and P. R. Rajeswari, “Intelligent health risk prediction systems using machine learning : a review,” IJET, vol. 7, no. 3, pp. 1019– 1023, 2018.

[6]. Himanshu Sharma,M A Rizvi,”Prediction of Heart Disease using Machine Learning Algorithms: A Survey”,International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169,Volume: 5 Issue: 8

[7]. Asif Salekin, John Stankovic, "Detection of Chronic Kidney Disease and Selecting Important Predictive Attributes," Proc. IEEE International Conference on Healthcare Informatics (ICHI), IEEE, Oct. 2016, doi:10.1109/ICHI.2016.36.

[8]. Pinar Yildirim, "Chronic Kidney Disease Prediction on Imbalanced Data by Multilayer Perceptron: Chronic Kidney Disease Prediction," Proc. 41st IEEE International Conference on Computer Software and Applications (COMPSAC), IEEE, Jul. 2017, doi: 10.1109/COMPSAC.2017.84

[9]. Sahil Sharma, Vinod Sharma, Atul Sharma, “Performance Based Evaluation of Various Machine Learning Classification Techniques for Chronic Kidney Disease Diagnosis,” July18,2016

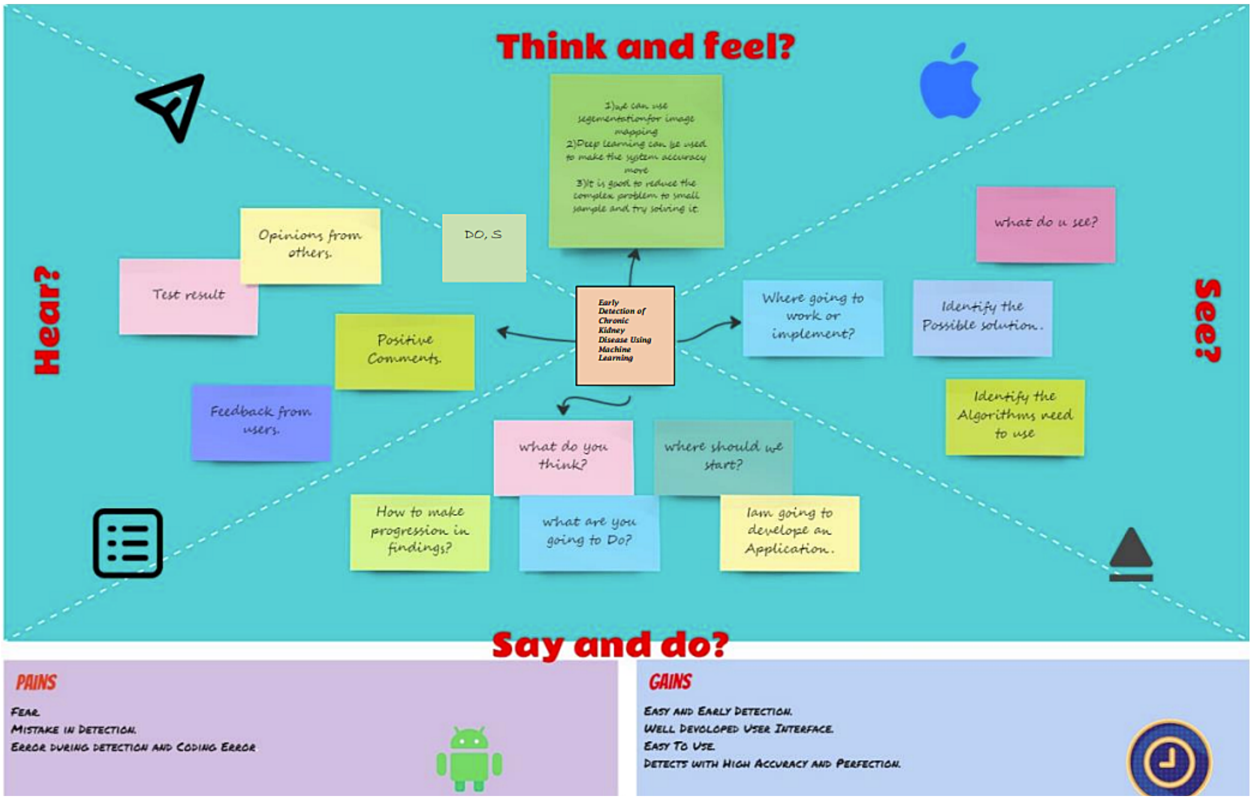
2.3 Problem Statement Definition

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 anaemia. 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

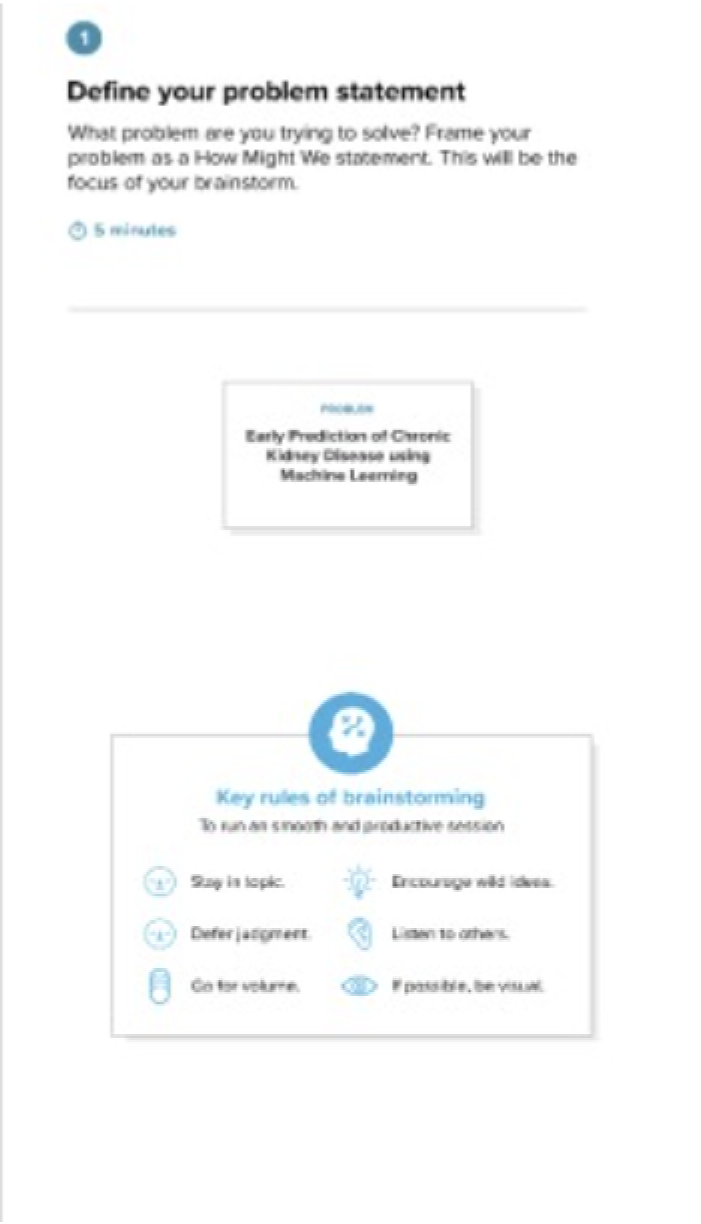
3.1 Empathy Map Canvas

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user's needs and pain points.


Think and feel?
1)we can use segementationfor image mapping
2)Deep learning can be used to make the system accuracy more
3)It is good to reduce the complex problem to small sample and try solving it
DO, S
Opinions from others.
Test result
Hear?
Positive Comments.
Feedback from users.
Early Detection of Chronic Kidney Disease Using Machine Learning
Where going to work or implement?
Identify the Possible solution.
what do u see?
See?
Identify the Algorithms need to use
what do you think?
where should we start?
How to make progression in findings?
what are you going to Do?
Iam going to develope an Application.
Say and do?
PAINS
Fear
Mistake in Detection.
Error during detection and Coding Error
GAINS
Easy and Early Detection.
Well Devoloped User Interface.
Easy To Use.
Detects with High Accuracy and Perfection.

3.2 Ideation & Brainstorming

Organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.



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

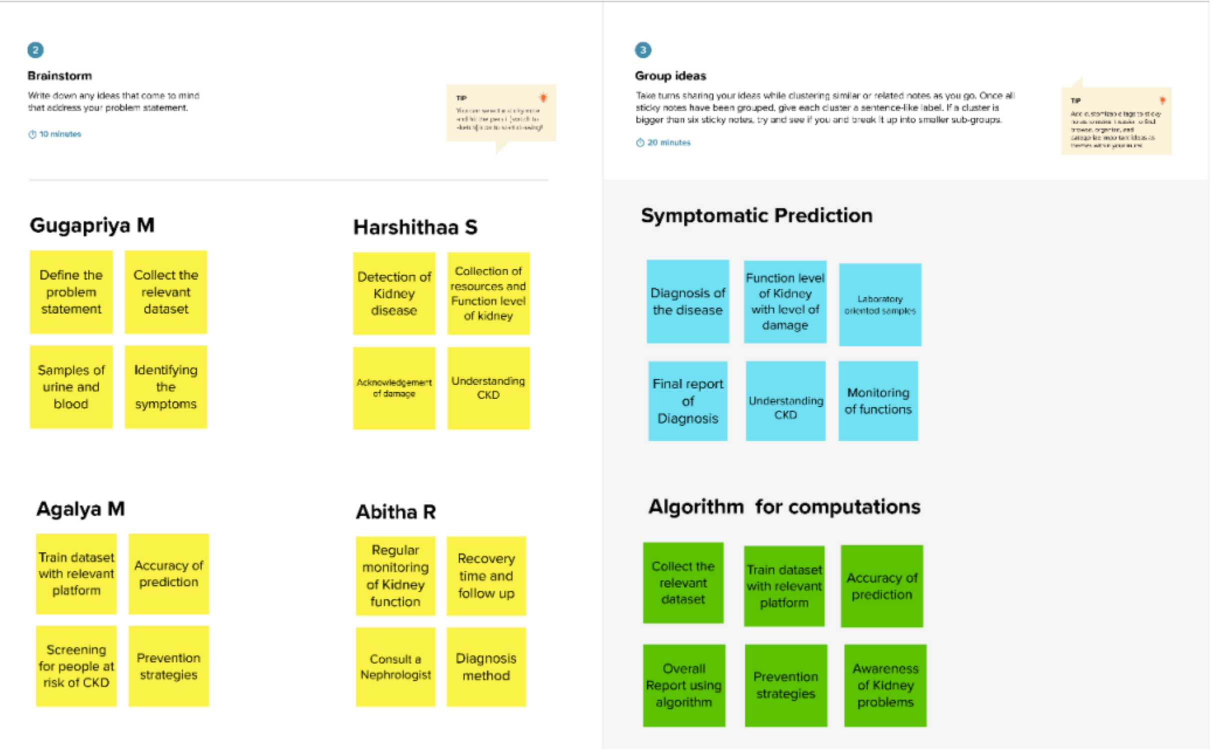
Early Prediction of Chronic Kidney Disease using Machine Learning

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Defer judgment.
- Go for volume.
- Encourage wild ideas.
- Listen to others.
- 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 [switch to sketch] icon to start drawing!

Gugapriya M

- Define the problem statement
- Collect the relevant dataset
- Samples of urine and blood
- Identifying the symptoms

Harshithaa S

- Detection of Kidney disease
- Collection of resources and Function level of kidney
- Acknowledgement of damage
- Understanding CKD

Agalya M

- Train dataset with relevant platform
- Accuracy of prediction
- Screening for people at risk of CKD
- Prevention strategies

Abitha R

- Regular monitoring of Kidney function
- Recovery time and follow up
- Consult a Nephrologist
- Diagnosis method

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

TIP

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

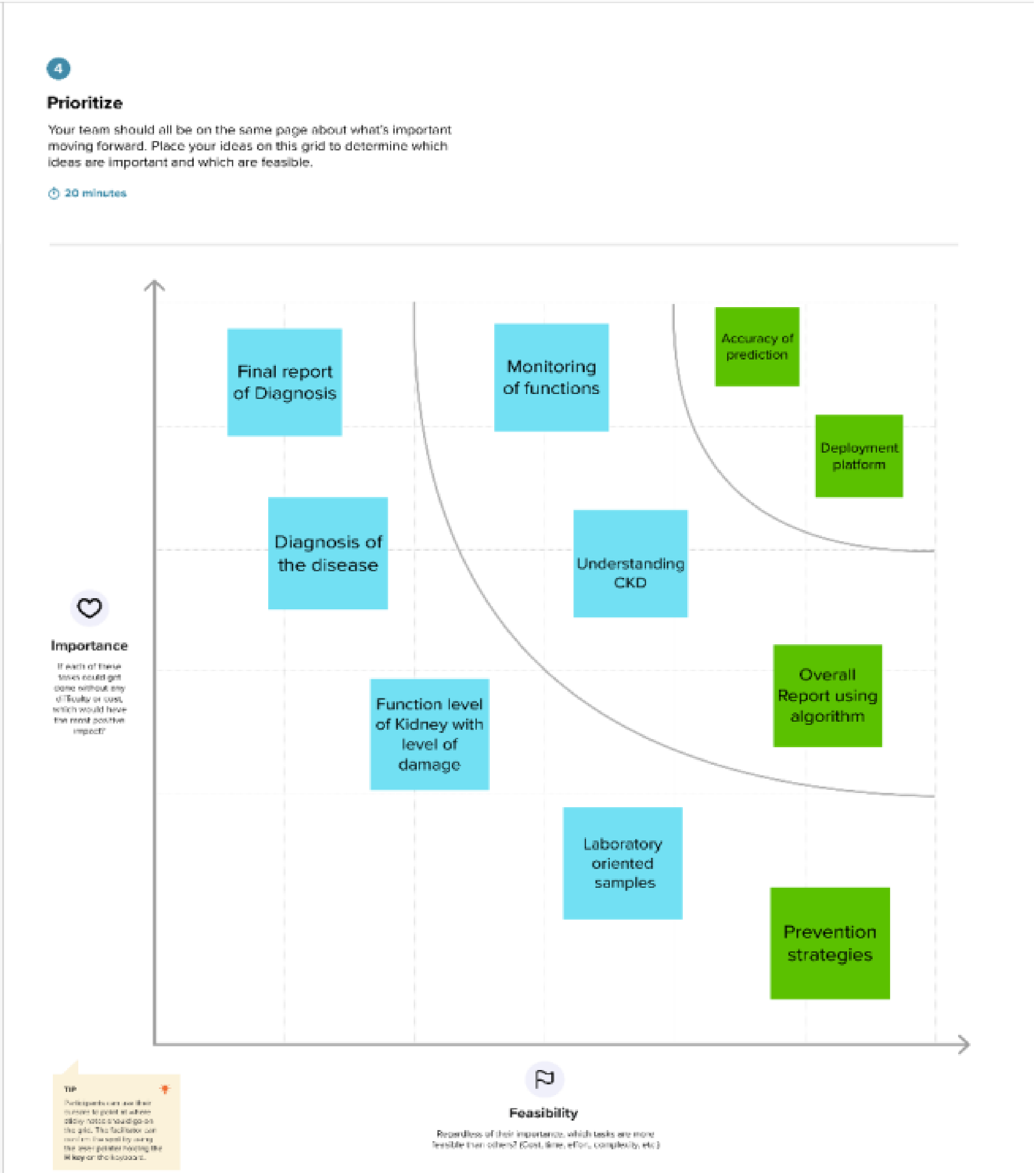
Symptomatic Prediction

- Diagnosis of the disease
- Function level of Kidney with level of damage
- Laboratory oriented samples
- Final report of Diagnosis
- Understanding CKD
- Monitoring of functions

Algorithm for computations

- Collect the relevant dataset
- Train dataset with relevant platform
- Accuracy of prediction
- Overall Report using algorithm
- Prevention strategies
- Awareness of Kidney problems

Step-3: Idea Prioritization



4

Prioritize

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

⏱ 20 minutes

Importance

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

Feasibility

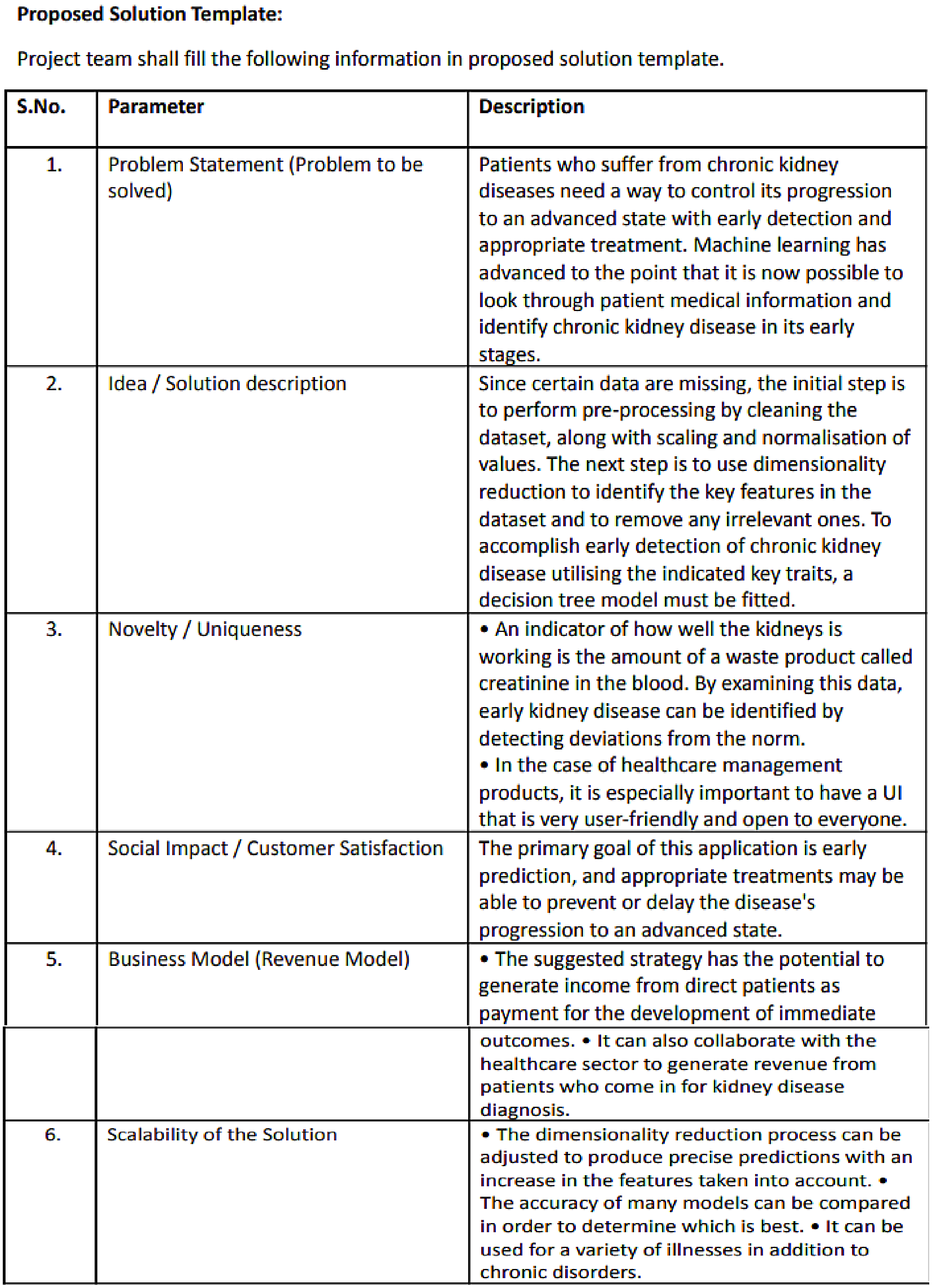
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

TIP

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

Final report of Diagnosis
Monitoring of functions
Accuracy of prediction
Deployment platform
Diagnosis of the disease
Understanding CKD
Overall Report using algorithm
Function level of Kidney with level of damage
Laboratory oriented samples
Prevention strategies

3.3 Proposed Solution

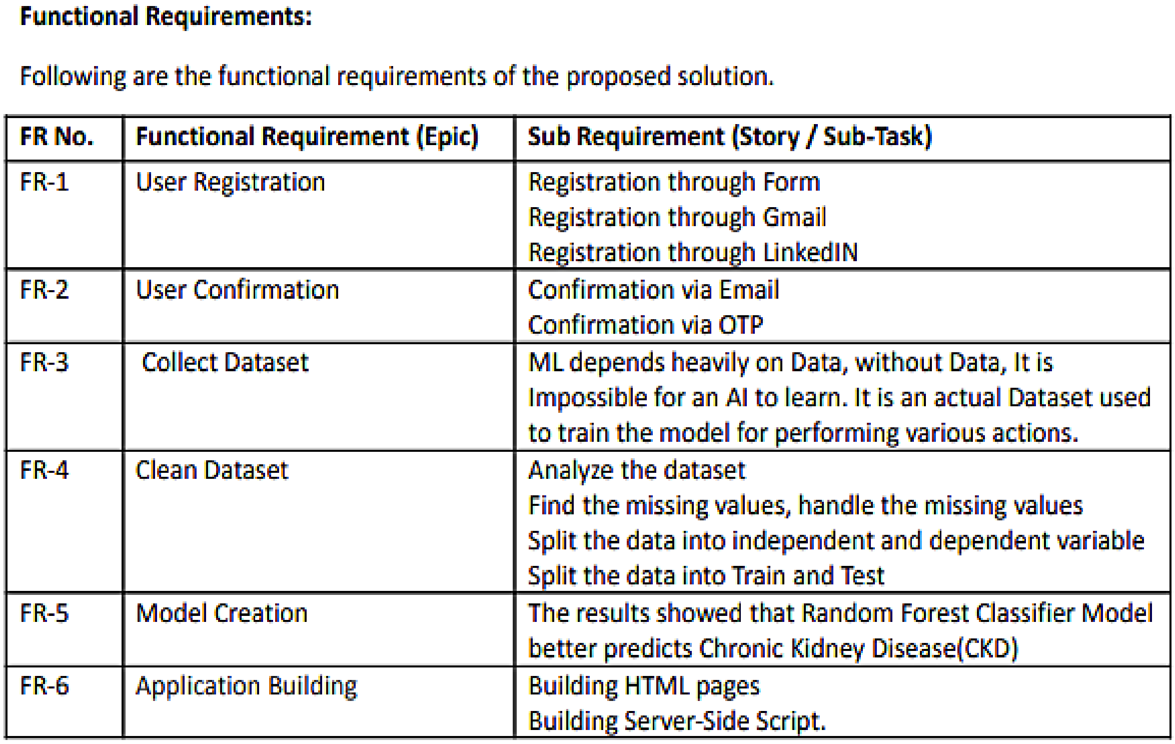


Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Patients who suffer from chronic kidney diseases need a way to control its progression to an advanced state with early detection and appropriate treatment. Machine learning has advanced to the point that it is now possible to look through patient medical information and identify chronic kidney disease in its early stages.
2.	Idea / Solution description	Since certain data are missing, the initial step is to perform pre-processing by cleaning the dataset, along with scaling and normalisation of values. The next step is to use dimensionality reduction to identify the key features in the dataset and to remove any irrelevant ones. To accomplish early detection of chronic kidney disease utilising the indicated key traits, a decision tree model must be fitted.
3.	Novelty / Uniqueness	• An indicator of how well the kidneys is working is the amount of a waste product called creatinine in the blood. By examining this data, early kidney disease can be identified by detecting deviations from the norm. • In the case of healthcare management products, it is especially important to have a UI that is very user-friendly and open to everyone.
4.	Social Impact / Customer Satisfaction	The primary goal of this application is early prediction, and appropriate treatments may be able to prevent or delay the disease's progression to an advanced state.
5.	Business Model (Revenue Model)	• The suggested strategy has the potential to generate income from direct patients as payment for the development of immediate outcomes. • It can also collaborate with the healthcare sector to generate revenue from patients who come in for kidney disease diagnosis.
6.	Scalability of the Solution	• The dimensionality reduction process can be adjusted to produce precise predictions with an increase in the features taken into account. • The accuracy of many models can be compared in order to determine which is best. • It can be used for a variety of illnesses in addition to chronic disorders.

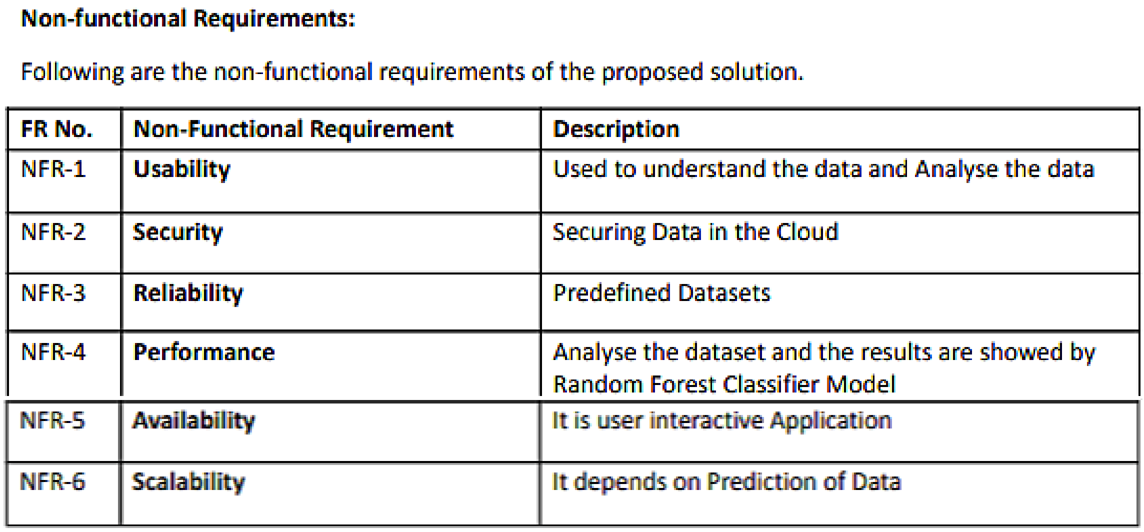
4. REQUIREMENT ANALYSIS



Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Collect Dataset	ML depends heavily on Data, without Data, It is Impossible for an AI to learn. It is an actual Dataset used to train the model for performing various actions.
FR-4	Clean Dataset	Analyze the dataset Find the missing values, handle the missing values Split the data into independent and dependent variable Split the data into Train and Test
FR-5	Model Creation	The results showed that Random Forest Classifier Model better predicts Chronic Kidney Disease(CKD)
FR-6	Application Building	Building HTML pages Building Server-Side Script.



Non-functional Requirements:

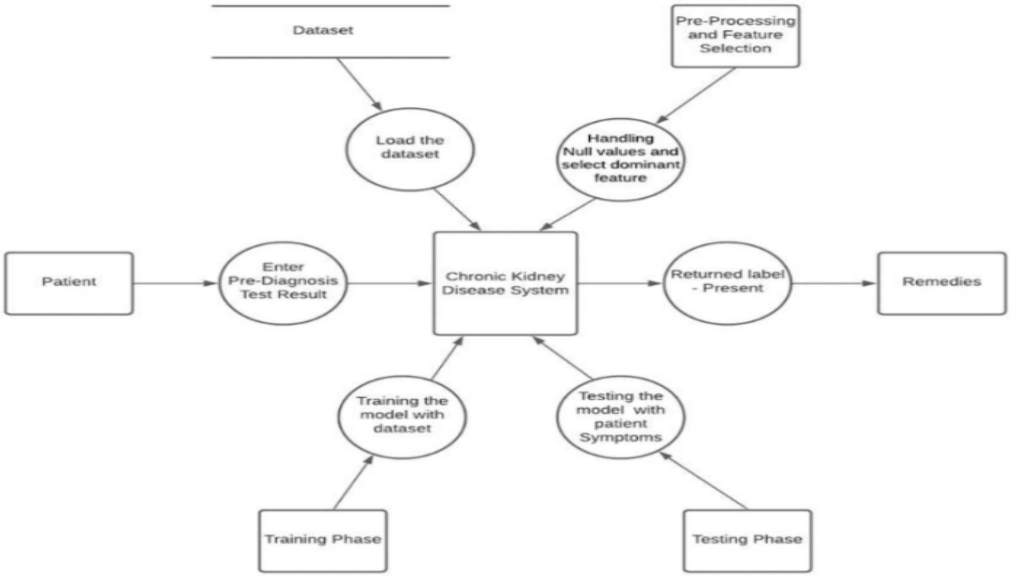
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Used to understand the data and Analyse the data
NFR-2	**Security**	Securing Data in the Cloud
NFR-3	**Reliability**	Predefined Datasets
NFR-4	**Performance**	Analyse the dataset and the results are showed by Random Forest Classifier Model
NFR-5	**Availability**	It is user interactive Application
NFR-6	**Scalability**	It depends on Prediction of Data

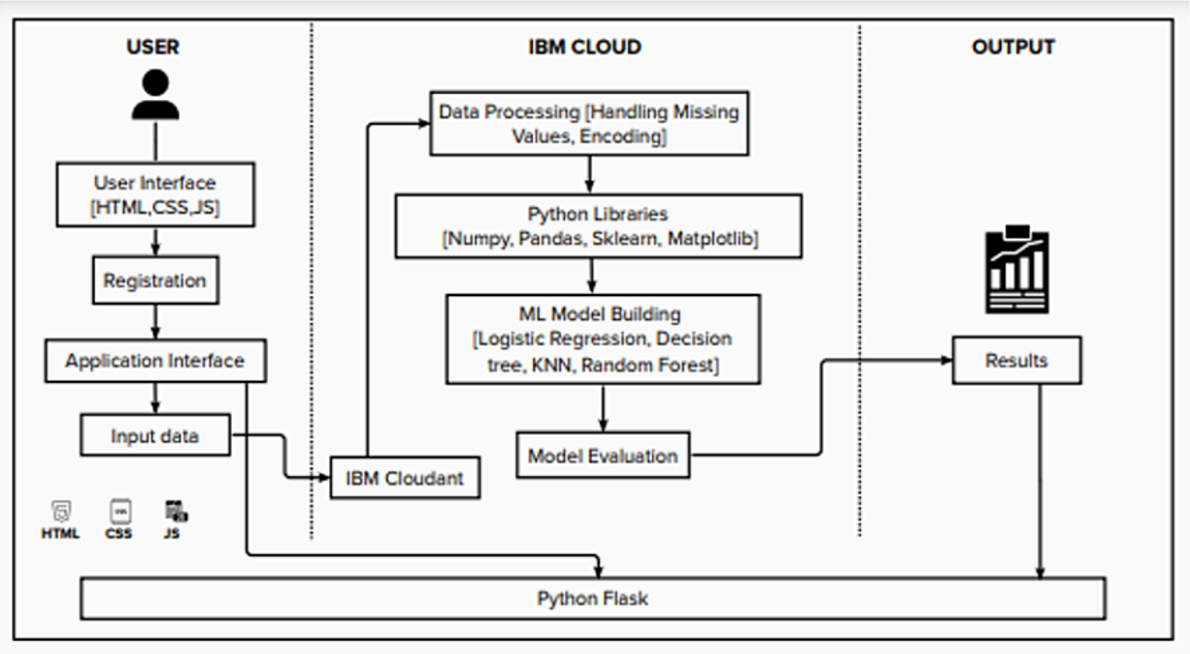
5. PROJECT DESIGN

5.1 Data Flow Diagrams

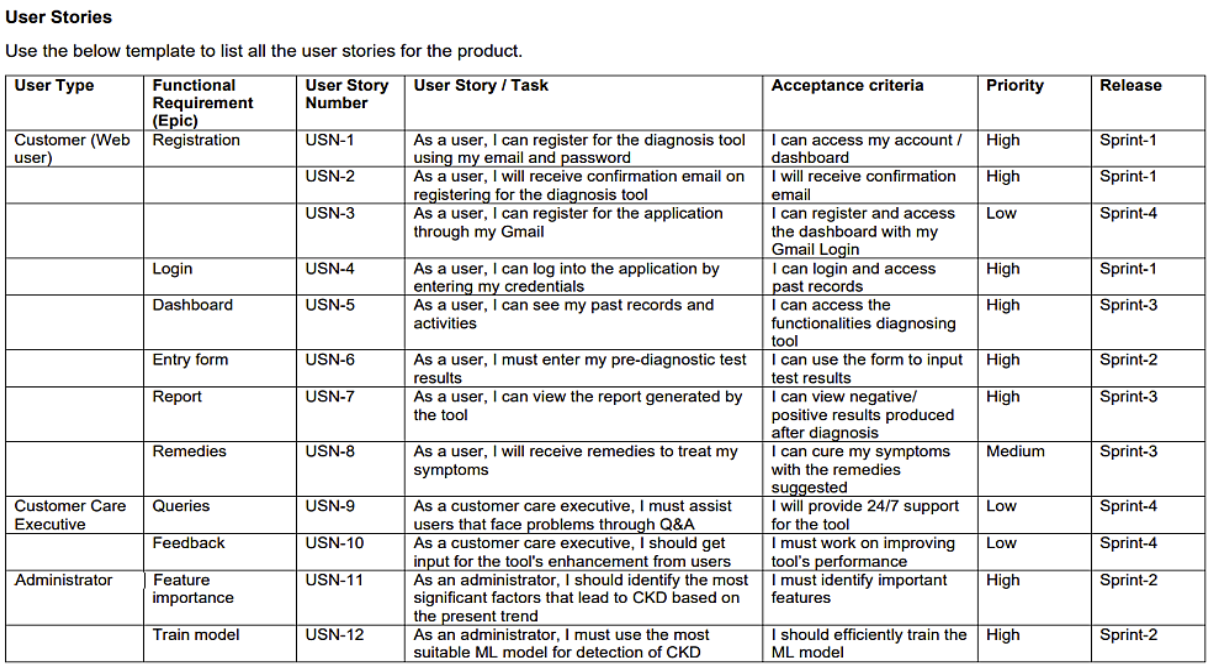
Data Flow Diagram:


Dataset
Pre-Processing and Feature Selection
Load the dataset
Handling Null values and select dominant feature
Patient
Enter Pre-Diagnosis Test Result
Chronic Kidney Disease System
Returned label - Present
Remedies
Training the model with dataset
Testing the model with patient Symptoms
Training Phase
Testing Phase

5.2 Solution & Technical Architectu


USER
IBM CLOUD
OUTPUT
User Interface [HTML,CSS,JS]
Registration
Application Interface
Input data
HTML
CSS
JS
IBM Cloudant
Data Processing [Handling Missing Values, Encoding]
Python Libraries [Numpy, Pandas, Sklearn, Matplotlib]
ML Model Building [Logistic Regression, Decision tree, KNN, Random Forest]
Model Evaluation
Results
Python Flask

5.3 User Stories



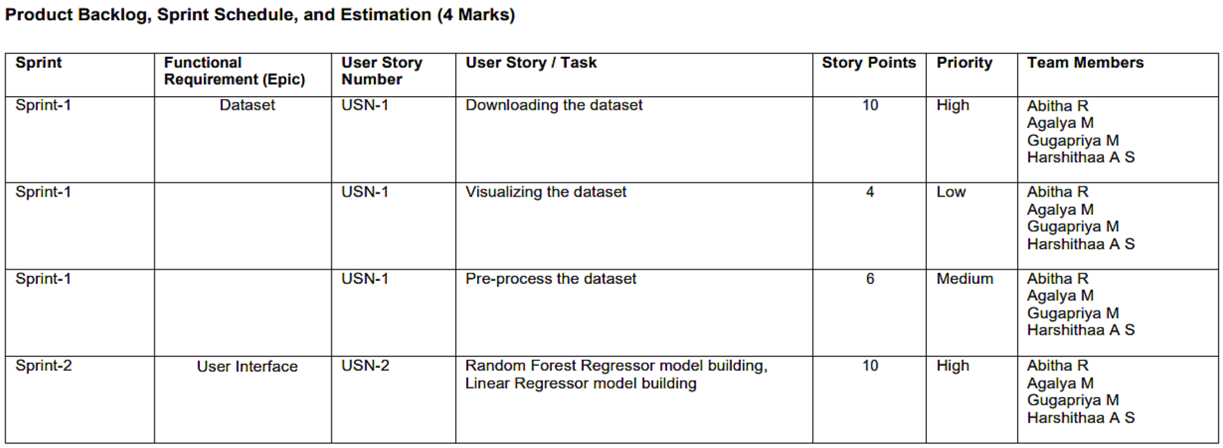
User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the diagnosis tool using my email and password	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email on registering for the diagnosis tool	I will receive confirmation email	High	Sprint-1
		USN-3	As a user, I can register for the application through my Gmail	I can register and access the dashboard with my Gmail Login	Low	Sprint-4
	Login	USN-4	As a user, I can log into the application by entering my credentials	I can login and access past records	High	Sprint-1
	Dashboard	USN-5	As a user, I can see my past records and activities	I can access the functionalities diagnosing tool	High	Sprint-3
	Entry form	USN-6	As a user, I must enter my pre-diagnostic test results	I can use the form to input test results	High	Sprint-2
	Report	USN-7	As a user, I can view the report generated by the tool	I can view negative/ positive results produced after diagnosis	High	Sprint-3
	Remedies	USN-8	As a user, I will receive remedies to treat my symptoms	I can cure my symptoms with the remedies suggested	Medium	Sprint-3
Customer Care Executive	Queries	USN-9	As a customer care executive, I must assist users that face problems through Q&A	I will provide 24/7 support for the tool	Low	Sprint-4
	Feedback	USN-10	As a customer care executive, I should get input for the tool's enhancement from users	I must work on improving tool's performance	Low	Sprint-4
Administrator	Feature importance	USN-11	As an administrator, I should identify the most significant factors that lead to CKD based on the present trend	I must identify important features	High	Sprint-2
	Train model	USN-12	As an administrator, I must use the most suitable ML model for detection of CKD	I should efficiently train the ML model	High	Sprint-2

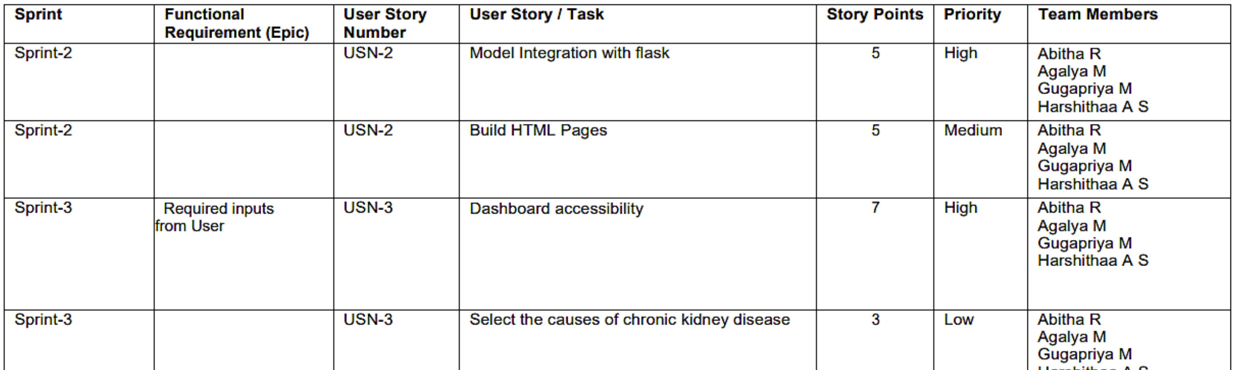
6. PROJECT PLANNING & SCHEDULING6

6.1 Sprint Planning & Estimation



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

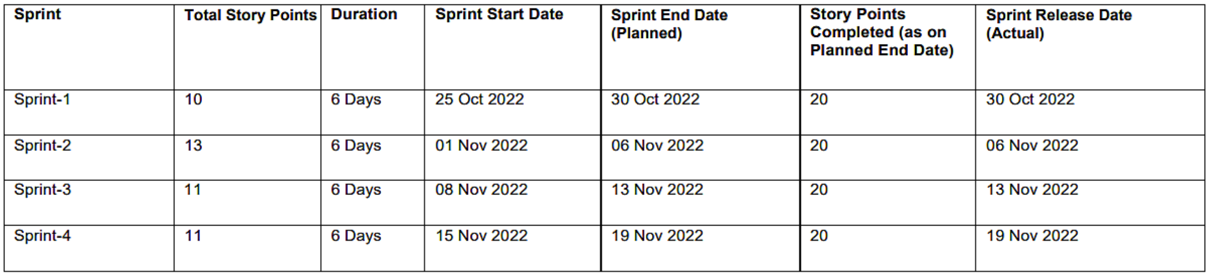
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset	USN-1	Downloading the dataset	10	High	Abitha R Agalya M Gugapriya M Harshithaa A S
Sprint-1		USN-1	Visualizing the dataset	4	Low	Abitha R Agalya M Gugapriya M Harshithaa A S
Sprint-1		USN-1	Pre-process the dataset	6	Medium	Abitha R Agalya M Gugapriya M Harshithaa A S
Sprint-2	User Interface	USN-2	Random Forest Regressor model building, Linear Regressor model building	10	High	Abitha R Agalya M Gugapriya M Harshithaa A S



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-2	Model Integration with flask	5	High	Abitha R Agalya M Gugapriya M Harshithaa A S
Sprint-2		USN-2	Build HTML Pages	5	Medium	Abitha R Agalya M Gugapriya M Harshithaa A S
Sprint-3	Required inputs from User	USN-3	Dashboard accessibility	7	High	Abitha R Agalya M Gugapriya M Harshithaa A S
Sprint-3		USN-3	Select the causes of chronic kidney disease	3	Low	Abitha R Agalya M Gugapriya M

6.2 Sprint Delivery Schedule

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

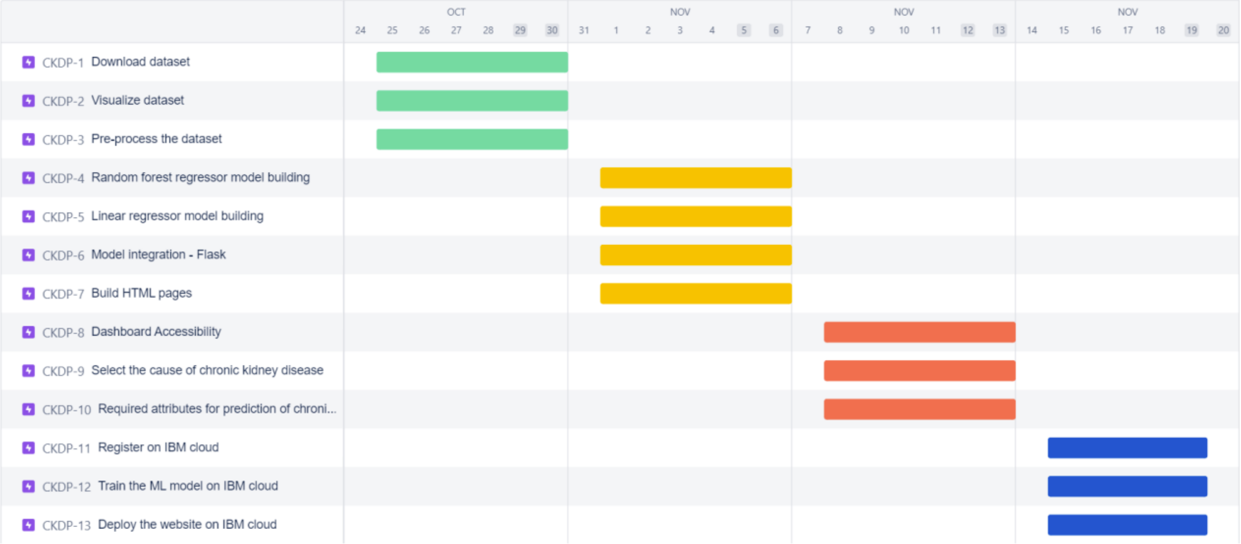


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	25 Oct 2022	30 Oct 2022	20	30 Oct 2022
Sprint-2	13	6 Days	01 Nov 2022	06 Nov 2022	20	06 Nov 2022
Sprint-3	11	6 Days	08 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-4	11	6 Days	15 Nov 2022	19 Nov 2022	20	19 Nov 2022

The following table shows the sprint works assigned to the members along with the priority and story points assigned with the functional requirements with regards to user story.

6.3 Reports from JIRA

Burndownchart



7. CODING & SOLUTION

7.1 .FEATURE 1

Collecting , Visualizing, and Preprocessing the Dataset

1.Importing the packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter as c
import seaborn as sns
import missingno as msng
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
#Data Collections
data=pd.read_csv("/content/drive/MyDrive/chronickidneydisease.csv")
data.head()
data.drop(['id'],axis=1,inplace=True)
data.columns
data.columns=['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
    'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',
    'appet', 'pe', 'ane', 'classification']
data.columns
data['classification'].unique()
data.info()
```

2. Data visualization

```
from matplotlib import pyplot
data.plot
data.plot.hist()
data.plot.box()
data.boxplot()

data.plot.area()
data.plot.scatter(x='age',y='bp')
pie = data
pie
pie.plot();
data.plot.bar()
data.corr()
plt.figure(figsize=(15,8));
plt.title("Correlation",color="green")
sns.heatmap(data.corr(),linewidth=1,annot=True);
sns.set_theme(style="white")
fig, ((ax1, ax2,ax3,ax4,ax5), (ax6, ax7,ax8,ax9,ax10))= plt.subplots(nrows=2, ncols=5, figsize=(18,14))
sns.boxplot(data=data,x="age",ax=ax1)
sns.boxplot(data=data,x="bp",ax=ax2)
sns.boxplot(data=data,x="sg",ax=ax3)
sns.boxplot(data=data,x="al",ax=ax4)
sns.boxplot(data=data,x="bgr",ax=ax5)
sns.boxplot(data=data,x="bu",ax=ax6)
sns.boxplot(data=data,x="sc",ax=ax7)
sns.boxplot(data=data,x="sod",ax=ax8)
sns.boxplot(data=data,x="pot",ax=ax9)
sns.boxplot(data=data,x="hemo",ax=ax10)
```

3. Data Preprocessing

```
data['classification']=data['classification'].replace("ckd\t",'ckd')
catcols=set(data.dtypes[data.dtypes=='O'].index.values)
print(catcols)
```

```
for i in catcols:
    print("columns:",i)


    print(c(data[i]))
    print('*'*120+'\n')
catcols.remove('rbc')
catcols.remove('pcv')
catcols.remove('wc')
catcols
contcols=set(data.dtypes[data.dtypes!='O'].index.values)
contcols
for i in catcols:
    print("continuous columns :",i)
    print(c(data[i]))
    print('*'*120+'\n')
contcols.remove('sg')
contcols.remove('al')
contcols.remove('su')
print(contcols)
contcols.add('rbc')
contcols.add('pc')
contcols.add('wc')
print(contcols)
catcols.add('sg')
catcols.add('al')
catcols.add('su')
print(catcols)
data['cad']=data.cad.replace('\tno','no')
c(data['cad'])
```

```
data['dm']=data.dm.replace(to_replace={'\tno':'no','\tyes':'yes',' yes':'yes'})
c(data['dm'])
data.isna().any()
data.isna().sum()
data.pcv=pd.to_numeric(data.pcv,errors='coerce')
data.wc=pd.to_numeric(data.wc,errors='coerce')
data.rc=pd.to_numeric(data.rc,errors='coerce')
data['bgr'].fillna(data['bgr'].mean(),inplace=True)
data['bp'].fillna(data['bp'].mean(),inplace=True)
data['bu'].fillna(data['bu'].mean(),inplace=True)
data['hemo'].fillna(data['hemo'].mean(),inplace=True)
data['pcv'].fillna(data['pcv'].mean(),inplace=True)
data['pot'].fillna(data['pot'].mean(),inplace=True)
data['rc'].fillna(data['rc'].mean(),inplace=True)
data['sc'].fillna(data['sc'].mean(),inplace=True)
data['sod'].fillna(data['sod'].mean(),inplace=True)
data['wc'].fillna(data['wc'].mean(),inplace=True)
data['age'].fillna(data['age'].mode()[0],inplace=True)
data['htn'].fillna(data['htn'].mode()[0],inplace=True)
data['pcc'].fillna(data['pcc'].mode()[0],inplace=True)
data['appet'].fillna(data['appet'].mode()[0],inplace=True)
data['al'].fillna(data['al'].mode()[0],inplace=True)
data['pc'].fillna(data['pc'].mode()[0],inplace=True)
data['rbc'].fillna(data['rbc'].mode()[0],inplace=True)
data['cad'].fillna(data['cad'].mode()[0],inplace=True)
data['ba'].fillna(data['ba'].mode()[0],inplace=True)
data['ane'].fillna(data['ane'].mode()[0],inplace=True)
data['su'].fillna(data['su'].mode()[0],inplace=True)
data['dm'].fillna(data['dm'].mode()[0],inplace=True)
```

```
data['pe'].fillna(data['pe'].mode()[0],inplace=True)
data['sg'].fillna(data['sg'].mode()[0],inplace=True)
```

SPRINT 2

ML MODEL CREATION

Importing the packages

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msng
from sklearn.metrics import accuracy_score,confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
```

1.splitting the dataset

```
for i in catcols:
    print("LABEL ENCODING OF :",i)
    le=LabelEncoder()
    print(c(data[i]))
    data[i]=le.fit_transform(data[i])
    print(c(data[i]))
    print('*'*100)
 data['rbc']=le.fit_transform(data['rbc'])
selcols=['rbc','pc','bgr','bu','pe','ane','dm','cad']
x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['classification'])
 print(x.shape)
print(y.shape)
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=2)
```

```
xtrain
```

2. Model creation

```
lgr=LogisticRegression()
lgr.fit(xtrain.values,ytrain.values)
ypred=lgr.predict(xtest)
ypred1=lgr.predict([[129,99,1,0,0,1,0,1]])
print(ypred1)
c(ypred)
```

3. Accuracy , Confustion Matrix , Classification Report

```
print(accuracy_score(ytest,ypred)*100)
confmat=confusion_matrix(ytest,ypred)
confmat
from sklearn.metrics import classification_report
print(classification_report(ytest, ypred))
from sklearn.model_selection import cross_val_score
scores = cross_val_score(lgr, xtrain, ytrain, cv=50)
print('Cross-Validation Accuracy Scores', scores)
```

FEATURE 2

SPRINT 3

1.FrontEnd Development

Frontend consists of 3 pages

Index page

Prediction page

Output page

Technology used in Frontend

1. HTML
2. CSS
3. JS

1.Index.html

```
<!DOCTYPE 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>Document</title>
    <style>
      * {
        padding: 0;
        margin: 0;
      }
      .background {
        background-image: url("/images/bg4.jpeg");
        background-repeat: no-repeat;
        background-size: cover;
      }
      .header {
        display: flex;
        flex: 100%;
        flex-direction: row;
        justify-content: flex-end;
        height: 60px;
        padding-right: 30px;
        background-color: #2e6e82;
        align-items: center;
      }
    . btn{
        color: #fff;
        font-size: large;
        text-decoration: none;
     }
     .titleWrapper{
        height: 500px;
        display: flex;
        justify-content: center;
        align-items: center;
     }
     .title{
        background-color: #2e6e82;
        border-radius: 5px;
        padding: 20px 90px;
     }
    </style>
  </head>
  <body class="background">
    <div class="header">
      <a href="inputs.html" class="btn"> Predict
        </button>
    </div>
    <div class="titleWrapper">
        <h1 class="title">Chronic Kidney disease predicition</h1>
    </div>
  </body>
</html>
```

Index page

2.Prediction Page

Inputs.html

```
<!DOCTYPE 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>Document</title>
    <style>
      .header {
        display: flex;
        justify-content: center;
        align-items: center;
      }
      .title {
        background-color: #2e6e82;
        border-radius: 5px;
        padding: 20px 90px;
        color: white;
      }
      .inputs {
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
        margin-top: 10px;
        background-color: #2e6e82;
        padding: 50px;
        border-radius: 50px;
      }
      input {
        width: 300px;
        height: 25px;
        text-align: center;
        margin-bottom: 5px;
        font-size: large;
      }
      select {
        width: 310px;
        height: 25px;
        text-align: center;
        margin-bottom: 5px;
        font-size: large;
      }
      .btn {
        display: flex;
        justify-content: center;
        align-items: center;
        margin-top: 30px;
      }
      button {
        position: relative;
        font-size: 14px;
        letter-spacing: 3px;
        height: 3em;
        padding: 0 3em;
        border: none;
        background-color: #2e6e82;
        color: #fff;
        text-transform: uppercase;
        overflow: hidden;
        border-radius: 5px;
      }

      button::before {
        content: "";
        display: block;
        position: absolute;
        z-index: 0;
        bottom: 0;
        left: 0;
        height: 0px;
        width: 100%;
        background: rgb(46, 110, 130);
        background: linear-gradient(
          90deg,
          rgba(46, 110, 130, 1) 20%,
          rgba(46, 110, 130, 1) 100%
        );
        transition: 0.2s;
      }

      button .label {
        position: relative;
      }
```

```
    button .icon {
      display: flex;
      align-items: center;
      justify-content: center;
      height: 3em;
      width: 3em;
      position: absolute;
      top: 3em;
      right: 0;
      opacity: 0;
      transition: 0.4s;
    }

    button:hover::before {
      height: 100%;
    }

    button:hover .icon {
      top: 0;
      opacity: 1;
    }
  </style>
</head>
<body>
  <div class="header">
    <h1 class="title">Chronic Kidney disease predicition</h1>
  </div>
  <div class="inputs">
  <input type="number" placeholder="Blood Urea" />
  <input type="number" placeholder="Blood Glucose Random" />
  <select name="Coronary Artery Disease" id="">
    <option value="Coronary Artery Disease">Coronary Artery Disease</option>
    <option value="yes">Yes</option>
    <option value="no">No</option>
  </select>
  <select name="anemia" id="">
    <option value="anemia">Anemia</option>
    <option value="yes">Yes</option>
    <option value="no">No</option>
  </select>
  <select name="pus cell" id="">
    <option value="pus cell">Pus Cell</option>
    <option value="normal">Normal</option>
    <option value="abnormal">Abnormal</option>
  </select>
  <input type="number" placeholder="Red Blood Cell Count" />
  <select name="diabetes mellitus" id="">
    <option value="diabetes mellitus">Diabetes Mellitus</option>
    <option value="yes">Yes</option>
    <option value="no">No</option>
  </select>
  <select name="pedal edema" id="">
    <option value="pedal edema">Pedal Edema</option>
    <option value="yes">Yes</option>
    <option value="no">No</option>
  </select>
</div>
```

```
    <div class="btn">
      <a href="results.html">
        <button>
          <span class="label">Predict</span>
          <span class="icon">
            <svg
              xmlns="http://www.w3.org/2000/svg"
              viewBox="0 0 24 24"
              width="24"
              height="24"
            >
              <path fill="none" d="M0 0h24v24H0z"></path>
              <path
                fill="currentColor"
                d="M16.172 11l-5.364-5.364 1.414-1.414L20 12l-7.778 7.778-1.414-1.414L16.172 13H4v-2z"
              ></path>
            </svg>
          </span>
        </button>
      </a>
    </div>
  </body>
</html>
```

Inputs page Or Prediction page

SPRINT 4

FrontEnd and Backend connection

1.FrontEnd Development

OUTPUT PAGE

Result.html

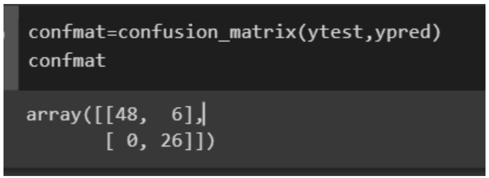
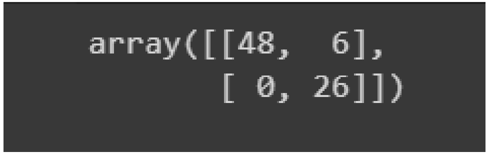
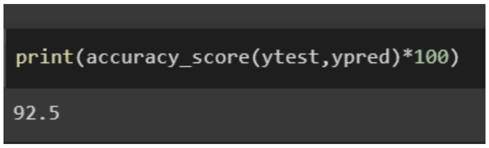
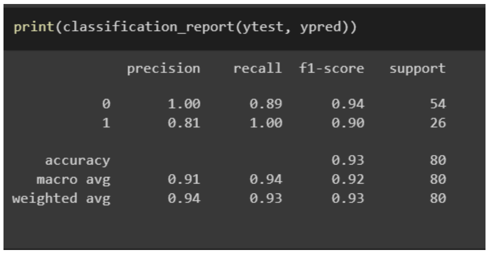
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      .header {
        display: flex;
        justify-content: center;
        align-items: center;
      }
      .title {
        background-color: #2e6e82;
        border-radius: 5px;
        padding: 20px 90px;
        color: white;
      }
      .resultWrapper {
        display: flex;
        height: 200px;
        justify-content: center;
        align-items: center;
      }
      .result {
        border-radius: 10px;
        padding: 10px 30px;
      }
      .result-positive {
        color: red;
        font-size: larger;
      }
      .result-negative {
        color: blue;
        font-size: larger;
      }
      h2 {
        color: #2e6e82;
      }
    </style>
  </head>
  <body>
    <div class="header">
      <h1 class="title">Chronic Kidney disease predicition</h1>
    </div>
    <div class="resultWrapper">
      <div class="result">
        <h2>
          Prediction:
          <samp class="result-positive">You have Chronic Kidney Disease</samp>
        </h2>
        <!-- <h2>
          Prediction:
          <samp class="result-negative"> You Don't Chronic Kidney Disease</samp>
        </h2> -->
      </div>
    </div>
  </body>

</html>
```

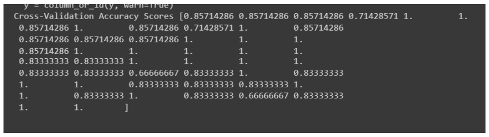
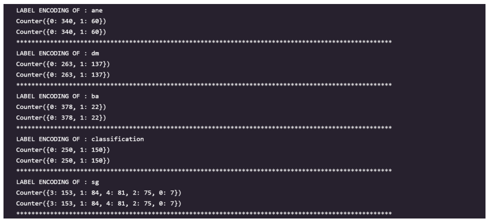
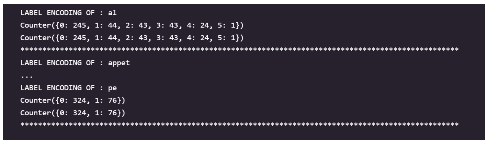
8. TESTING:

Model Performance Testing:

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

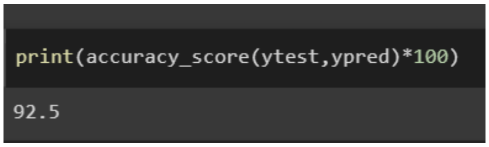
S.No.	Parameter	Values	Screenshot

1.	Metrics	**Classification Model:**
		Confusion Matrix - , Accuray Score- & Classification Report -
			Classification Model :
			Confusion Matrix :
			
			`confmat=confusion_matrix(ytest,ypred)`
			`confmat`
			`array([[48,  6],`
			`       [ 0, 26]])`
			
			`array([[48,  6],`
			`       [ 0, 26]])`
			Accuracy Score :
			
			`print(accuracy_score(ytest,ypred)*100)`
			`92.5`
			Classification Report :
			
			`print(classification_report(ytest, ypred))`
			`              precision    recall  f1-score   support`
			`           0       1.00      0.89      0.94        54`
			`           1       0.81      1.00      0.90        26`
			`    accuracy                           0.93        80`
			`   macro avg       0.91      0.94      0.92        80`
			`weighted avg       0.94      0.93      0.93        80`

2.	Tune the Model	Hyperparameter Tuning - Validation Method -	Validation Method :  y = column_or_1d(y, warn=True) Cross-Validation Accuracy Scores [0.85714286 0.85714286 0.85714286 0.71428571 1. 1. 0.85714286 1. 0.85714286 0.71428571 1. 0.85714286 0.85714286 0.85714286 0.85714286 1. 1. 1. 0.85714286 1. 1. 1. 1. 1. 0.83333333 0.83333333 1. 1. 1. 1. 0.83333333 0.83333333 0.66666667 0.83333333 1. 0.83333333 1. 1. 0.83333333 0.83333333 0.83333333 1. 1. 0.83333333 1. 0.83333333 0.66666667 0.83333333 1. 1.] Hyperparameter Tuning :  LABEL ENCODING OF : ane Counter({0: 340, 1: 60}) Counter({0: 340, 1: 60}) ** LABEL ENCODING OF : dm Counter({0: 263, 1: 137}) Counter({0: 263, 1: 137}) ** LABEL ENCODING OF : ba Counter({0: 378, 1: 22}) Counter({0: 378, 1: 22}) ** LABEL ENCODING OF : classification Counter({0: 250, 1: 150}) Counter({0: 250, 1: 150}) ** LABEL ENCODING OF : sg Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7}) Counter({3: 153, 1: 84, 4: 81, 2: 75, 0: 7}) **  LABEL ENCODING OF : al Counter({0: 245, 1: 44, 2: 43, 3: 43, 4: 24, 5: 1}) Counter({0: 245, 1: 44, 2: 43, 3: 43, 4: 24, 5: 1}) ** LABEL ENCODING OF : appet ... LABEL ENCODING OF : pe Counter({0: 324, 1: 76}) Counter({0: 324, 1: 76}) **

9. RESULTS

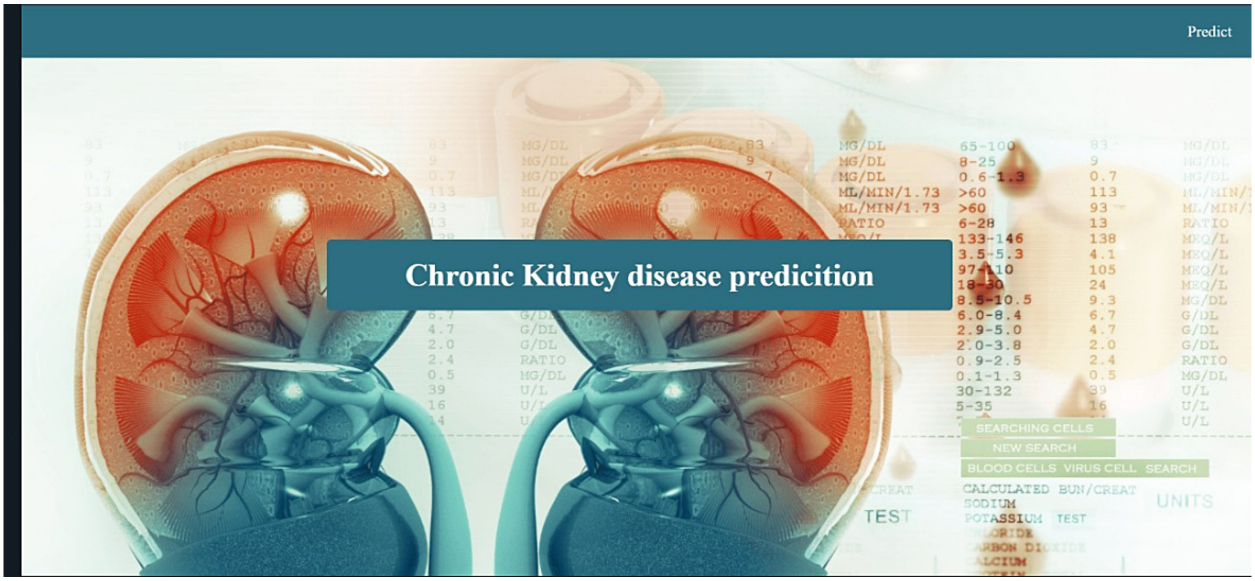
9.1 Machine Learning model accuracy



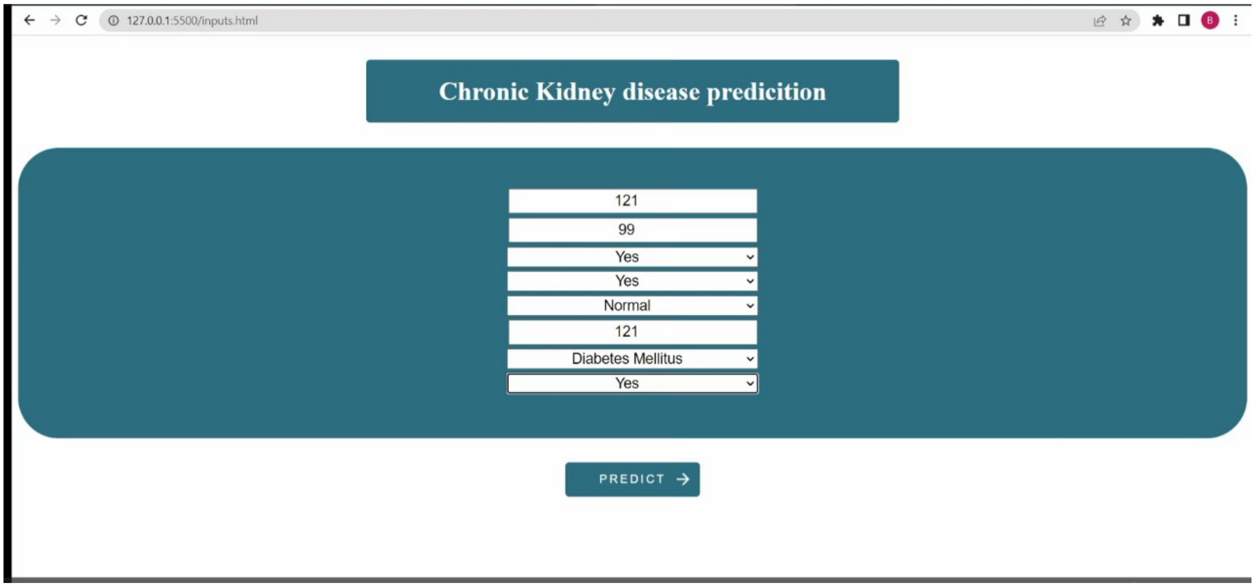
```
print(accuracy_score(ytest,ypred)*100)
92.5
```

9.2 Chronic Kidney Disease Prediction Application

Home page



Prediction Page



127.0.0.1:5500/inputs.html

121

99

Yes

Yes

Normal

121

Diabetes Mellitus

Yes

Resu

10. ADVANTAGES & DISADVANTAGES

Advantages

- Code optimization
- Better accuracy
- Less execution time
- Feature selection makes the work simpler

Disadvantages

- The correct dataset should be given inorder to gain the accuracy output
- patient cannot access it on their own

11.CONCLUSION

Our ideology is practicable and, if possible, might be developed as an application. Making image detection easier by segmenting and creating images from all conceivable angles .Other than binary classification, various other features may be used to discover further classifications such as low, mild, medium, and high (chances).We committed such a concept for detection using both Machine Learning , which is almost always used in various Domains and is primarily known for its accuracy in detection. We conclude here that our system can be adapted for any kind of Environment and it supports well with any Ideology to make change.

12.FUTURE SCOPES

- It can be developed as a Web or Android Application.
- In future Alternate Advanced technologies can be Implemented.
- It can be predicted based on X rays or MRI scans.
- The Identification and tracking system can be implemented if possible.

14. APPENDIX

Source Code

SourceCode:https://github.com/IBM-EPBL/IBM-Project-33833-1660227727/tree/main/Final%20Deliverables

GitHub & Project Demo Link

Github: https://github.com/IBM-EPBL/IBM-Project-33833-1660227727

DemoLink: https://drive.google.com/file/d/1q0ZlVTT98YBmrnjV8qiCdxI3APuUnZBj/view