INDEX

Early Detection of Chronic Kidney Disease using Machine Learning

1. INTRODUCTION

1.1 Project Overview

Chronic Kidney Disease prediction is one of the most important issues in healthcare analytics. The most interesting and challenging tasks in day-to-day life is prediction in medical field.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.Chronic kidney Disease can be cured, if treated in the early stages. 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(Al) using machine learning model.

1.2 Purpose

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.

2.LITERATURE SURVEY

2.1 Existing Problem

The existing solutions with machine learning approach are not extremely efficient as it is very difficult to train the model for predictive analysis due to unavailability of big data sets. Also it supports the medical community in just detecting the existence of the disease but not in identifying the stages of the disease. Few solutions aimed at getting accurate prediction of the disease but requires clinical tests to be done for detecting kidney disease. High accuracy was achieved when full feature set was used which makes it a costly approach rather than using selected important features.

2.2 References

S No	Title	Authors	Abstract	Drawbacks
1	Prediction of Chronic Kidney Disease Using Adaptive Hybridized Deep Convolutional Neural Network on the Internet of Medical Things Platform	Guozhen Chen, Chenguang Ding, Yang Li, Xiaojun Hu, Xiao Li, Li Ren, Xiaoming Ding, Puxun Tian, Wujun Xue	Chronic Kidney disease is a severe lifelong condition caused either by renal disease or by impaired functions of the kidneys. In the present area of research, Kidney cancer is one of the deadliest and crucial importance for the survival of the patients ’ diagnosis and classification. Early diagnosis and proper therapy can stop or delay the development of this chronic disease into the final stage where dialysis or renal transplantation is the only way of saving the life of the patient. The development of automated tools to accurately identify subtypes of kidney cancer is, therefore, an urgent challenge in the recent past. In this paper, to examine the ability of various deep learning methods an Adaptive hybridized Deep Convolutional Neural Network (AHDCNN) has been proposed for the early detection of Kidney disease efficiently and effectively. Classification technology efficiency depends on the role of the data set. To enhance the accuracy of the classification system by reducing the feature dimension an algorithm model has been developed using CNN. These high-level properties help to build a supervised tissue classifier that discriminates between the two types of tissue. The experimental process on the Internet of medical things platform (IoMT) concludes, with the aid of predictive analytics, that advances in machine learning which provides a promising framework for the recognition of intelligent solutions to prove their predictive capability beyond the field of kidney disease.	It is very difficult to train the model for predictive analysis due to unavailability of big data sets.

2	A Comprehensive Unsupervised Framework for Chronic Kidney Disease Prediction	LINTA ANTONY, SAMI AZAM, (Member, IEEE), EVA IGNATIOUS, RYANA QUADIR, ABHIJITH REDDY BEERAVOLU, MIRJAM JONKMAN,(Member, IEEE), FRISO DE BOER	The incidence, prevalence, and progression of chronic kidney disease (CKD) conditions have evolved over time. In most countries, diabetics and hypertension are the main causes of CKDs. The global guidelines classify CKD as a condition that results in decreased kidney function over time, as indicated by glomerular filtration rate (GFR) and markers of kidney damage. People with CKDs are likely to die at an early age. It is crucial for doctors to diagnose various conditions associated with CKD in an early stage because early detection may prevent or even reverse kidney damage. Early detection can provide better treatment and proper care to the patients. In many regional hospital/clinics, there is a shortage of nephrologists or general medical persons who diagnose the symptoms. This has resulted in patients waiting longer to get a diagnosis. Therefore, this research believes developing an intelligent system to classify a patient into classes of 'CKD' or 'Non-CKD' can help the doctors to deal with multiple patients and provide diagnosis faster. In time, organizations can implement the proposed machine learning framework in regional clinics that have lower medical expert retention, this can provide early diagnosis to patients in regional areas. Although, several researchers have tried to address the situation by developing intelligent systems using supervised machine learning methods, till date limited studies have used unsupervised machine learning algorithms. The primary aim of this research is to implement and compare the performance of various unsupervised algorithms and identify best possible combinations that can provide better accuracy and detection rate. This research has	This work aims to support the medical community in just detecting the existence of the disease but not in identifying the stages of the disease.

			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.	
3	Clinically Applicable Machine Learning Approach to Identify Attributes of Chronic Kidney Disease (CKD) for Use in Low-Cost Diagnostic Screening	MD. RASHED-AL-MAHFUZ, ABEDUL HAQUE, AKM AZAD, SALEM A. ALYAMI, JULIAN M. W. QUINN, MOHAMMAD ALI MONI	Objective: Chronic kidney disease (CKD) is a major public health concern worldwide. High costs of late-stage diagnosis and insufficient testing facilities can contribute to high morbidity and mortality rates in CKD patients, particularly in less developed countries. Thus, early diagnosis aided by vital parameter analytics using affordable computer-aided diagnosis could not only reduce diagnosis costs but improve patient management and outcomes. Methods: In this study, we developed machine learning models using selective key pathological categories to identify clinical test attributes that will aid in accurate early diagnosis of CKD. Such an approach will save time and costs for diagnostic screening. We have also evaluated the performance of several classifiers with k-fold cross-validation on optimized datasets derived using these selected clinical test attributes. Results: Our results suggest that the optimized datasets with important attributes perform well in diagnosis of CKD using our proposed machine learning models. Furthermore, we evaluated clinical test attributes based on urine and blood tests along with clinical parameters that have low costs of acquisition. The predictive models with the optimized and pathologically categorized attributes set yielded high levels of CKD diagnosis accuracy with random	This proposed work was aimed at getting accurate prediction of the disease but requires clinical tests to be done for detecting kidney disease.

			forest (RF) classifier being the best performing. Conclusions: Our machine learning approach has yielded effective predictive analytics for CKD screening which can be developed as a resource to facilitate improved CKD screening for enhanced and timely treatment plans.	
4	Prediction of Chronic Kidney Disease - A Machine Learning Perspective	PANKAJ CHITTORA,SANDEEP CHAURASIA,(Senior Member, IEEE), PRASUN CHAKRABARTI, (Senior Member, IEEE), GAURAV KUMAWAT,TULIKA CHAKRABARTI, ZBIGNIEW LEONOWICZ, (Senior Member, IEEE), MICHAŁ JASIŃSKI, (Member, IEEE), ŁUKASZ JASIŃSKI,	Chronic Kidney Disease is one of the most critical illness and proper diagnosis is required. With the help of a machine learning classifier algorithms, the doctor can detect the disease on time. 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	High accuracy was achieved when full feature set was used which makes it a costly approach rather than using selected important features.

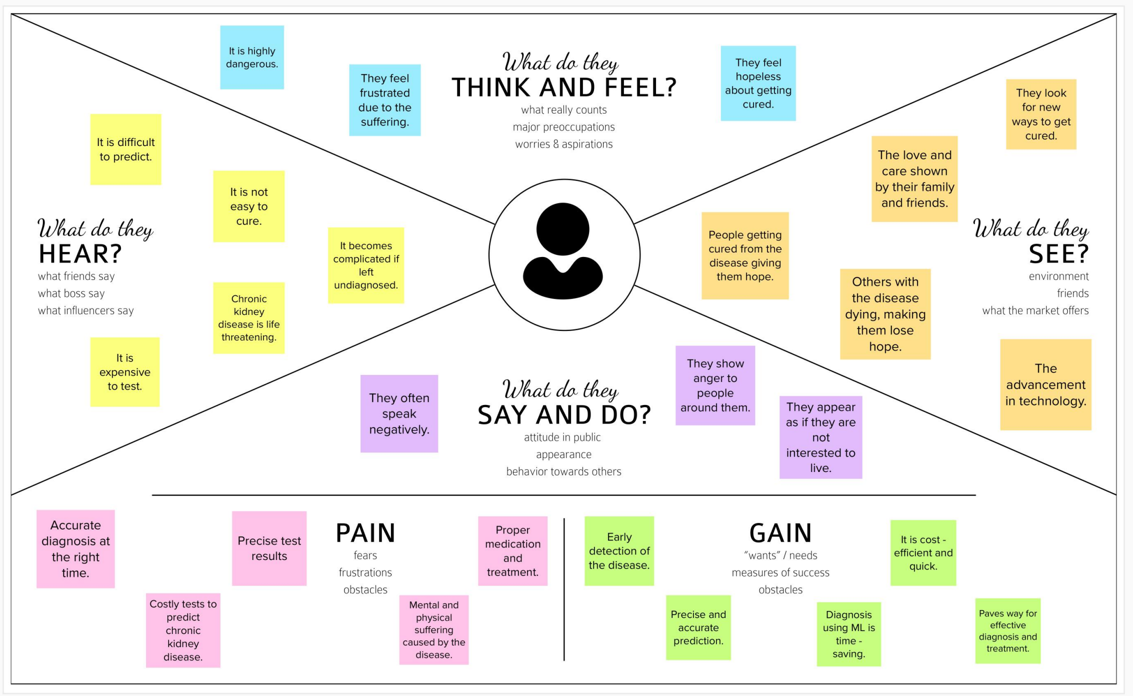
		RADOMIR GONO, (Senior Member, IEEE), ELŻBIETA JASIŃSKA ,VADIM BOLSHEV	shrinkage and selection operator regression selected features with synthetic minority over-sampling technique gave the best after synthetic minority over-sampling technique with full features. Linear support vector machine gave the 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%.	

2.3 Problem Statement Definition

Common people may have some illness and need to detect and predict the disease at an early stage so that effective treatment can be done at the right time as a result of which the patients may be cured at the earliest without having to undergo complex and costly procedures for testing and diagnosis.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

It is highly dangerous.
They feel frustrated due to the suffering.
They feel hopeless about getting cured.

What do they HEAR?
what friends say
what boss say
what influencers say

It is difficult to predict.
It is not easy to cure.
It becomes complicated if left undiagnosed.
Chronic kidney disease is life threatening.
It is expensive to test.

What do they SEE?
environment
friends
what the market offers

They look for new ways to get cured.
The love and care shown by their family and friends.
People getting cured from the disease giving them hope.
Others with the disease dying, making them lose hope.
The advancement in technology.

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

They often speak negatively.
They show anger to people around them.
They appear as if they are not interested to live.

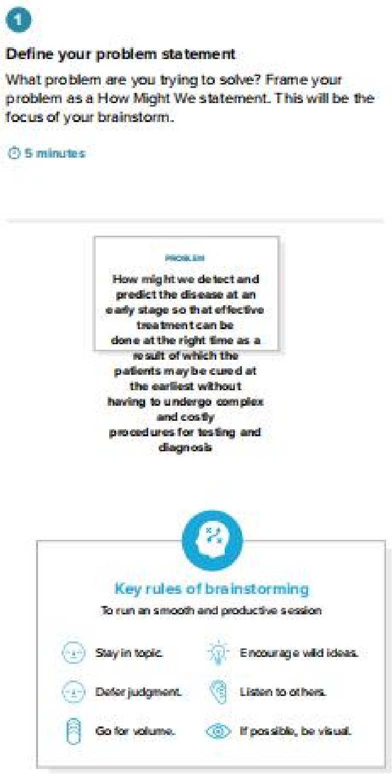
PAIN
fears
frustrations
obstacles

Accurate diagnosis at the right time.
Precise test results
Proper medication and treatment.
Costly tests to predict chronic kidney disease.
Mental and physical suffering caused by the disease.

GAIN
"wants" / needs
measures of success
obstacles

Early detection of the disease.
It is cost - efficient and quick.
Precise and accurate prediction.
Diagnosis using ML is time - saving.
Paves way for effective diagnosis and treatment.

3.2 Ideation and Brainstorming



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

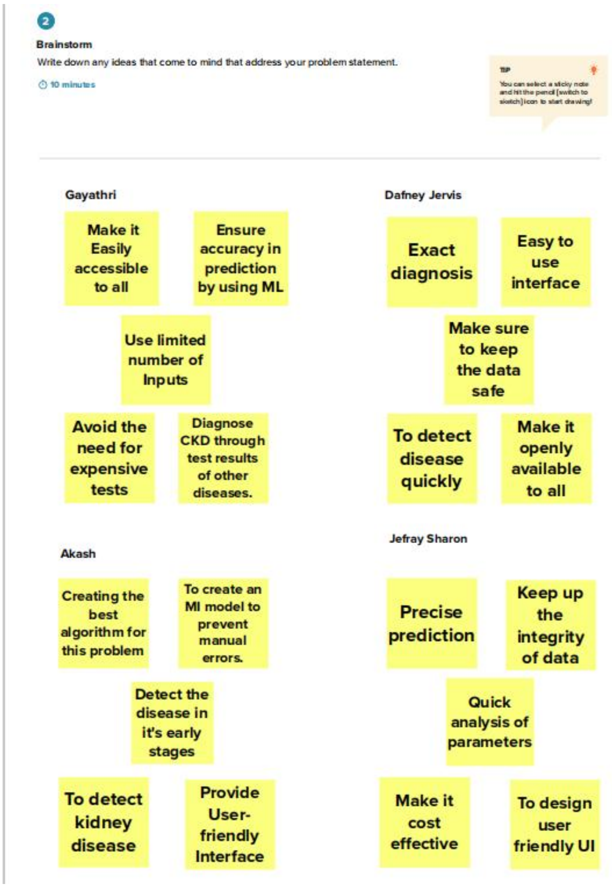
PROBLEM

How might we detect and predict the disease at an early stage so that effective treatment can be done at the right time as a result of which the patients may be cured at the earliest without having to undergo complex and costly procedures for testing and diagnosis

Key rules of brainstorming

To run an smooth and productive session

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



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!

Gayathri

- Make it Easily accessible to all
- Ensure accuracy in prediction by using ML
- Use limited number of Inputs
- Avoid the need for expensive tests
- Diagnose CKD through test results of other diseases.

Dafney Jervis

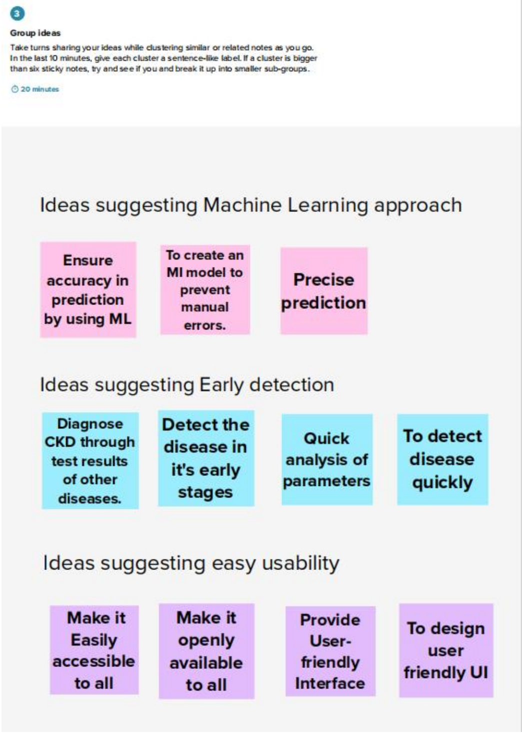
- Exact diagnosis
- Easy to use interface
- Make sure to keep the data safe
- To detect disease quickly
- Make it openly available to all

Akash

- Creating the best algorithm for this problem
- To create an MI model to prevent manual errors.
- Detect the disease in it's early stages
- To detect kidney disease
- Provide User-friendly Interface

Jefray Sharon

- Precise prediction
- Keep up the integrity of data
- Quick analysis of parameters
- Make it cost effective
- To design user friendly UI



3

Group ideas

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

20 minutes

Ideas suggesting Machine Learning approach

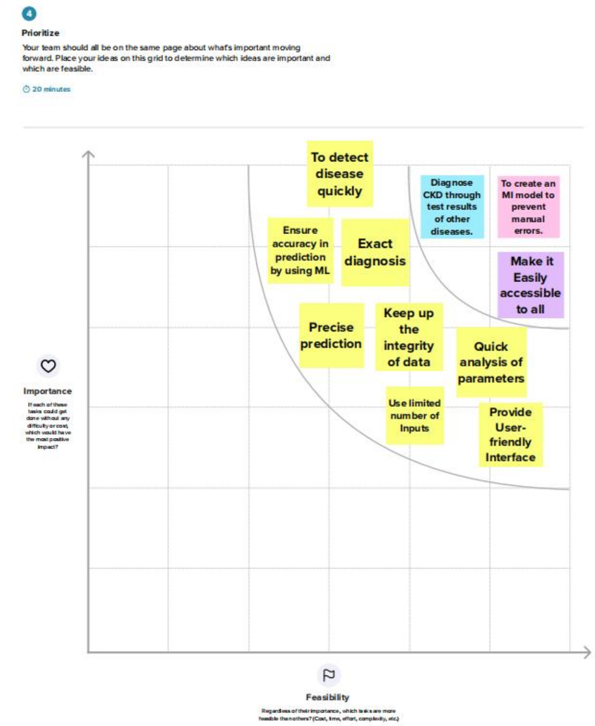
- Ensure accuracy in prediction by using ML
- To create an MI model to prevent manual errors.
- Precise prediction

Ideas suggesting Early detection

- Diagnose CKD through test results of other diseases.
- Detect the disease in it's early stages
- Quick analysis of parameters
- To detect disease quickly

Ideas suggesting easy usability

- Make it Easily accessible to all
- Make it openly available to all
- Provide User-friendly Interface
- To design user friendly UI



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

- To detect disease quickly
- Diagnose CKD through test results of other diseases.
- To create an MI model to prevent manual errors.
- Ensure accuracy in prediction by using ML
- Exact diagnosis
- Make it Easily accessible to all
- Precise prediction
- Keep up the integrity of data
- Quick analysis of parameters
- Use limited number of Inputs
- Provide User-friendly Interface

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.)

3.3 Proposed Solution

Problem Statement (Problem to be solved):

Many people suffer from Chronic Kidney Disease(CKD) which is life-threatening when identified in a serious stage and needs effective treatment at an early stage.

Idea / Solution description:

The aim of this project is to detect Chronic Kidney Disease even before there are symptoms so that effective treatment can be given at the right time.

Novelty / Uniqueness :

This project uses parameters from various test results other than those for CKD to diagnose the disease.

Social Impact / Customer Satisfaction:

As this solution predicts the disease at an early stage the patients can get the right treatment at the right time. Also It helps doctors to quickly diagnose the disease.

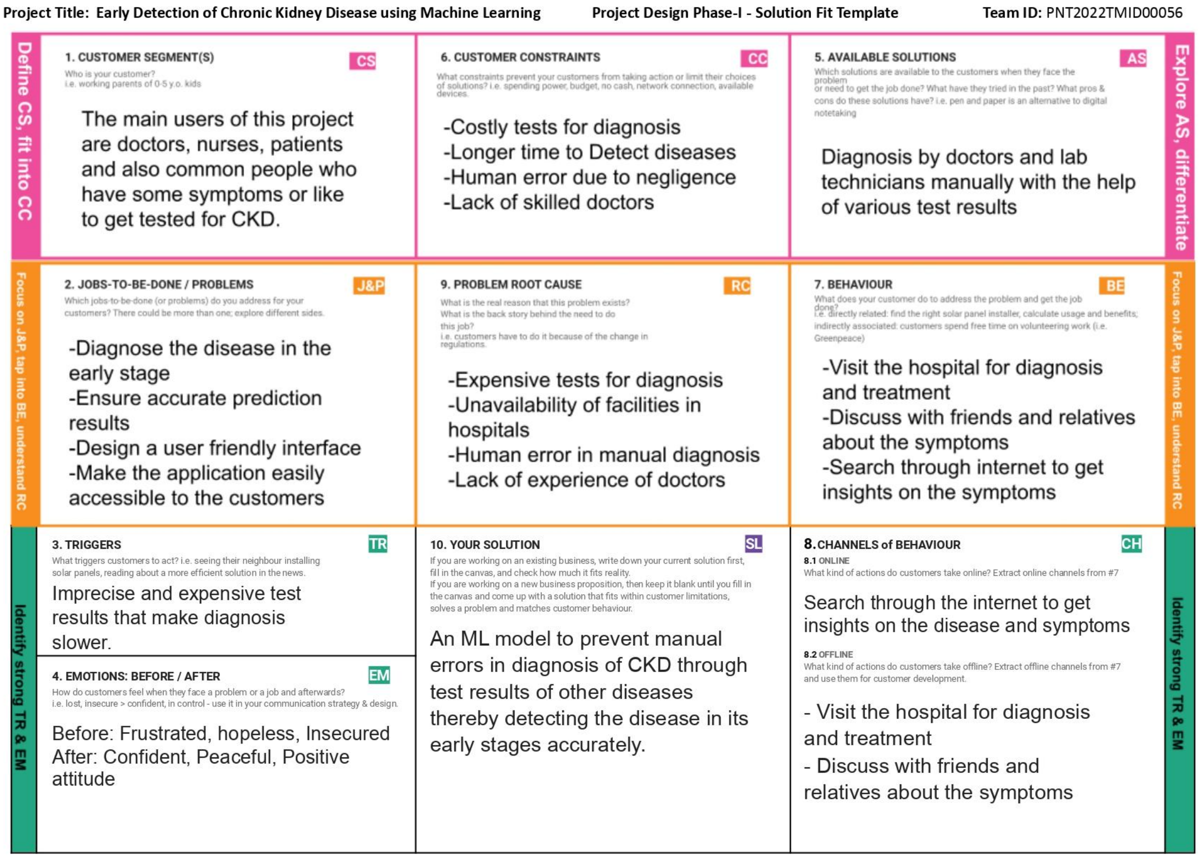
Business Model (Revenue Model):

This project can be kept open for people to have easy access as a website so that anyone can get quick diagnosis at the earliest. Hence revenue can be generated from the use of this website as a tool by Doctors and common public as well.

Scalability of the Solution:

It can be improved by reducing the number of parameters needed for this prediction and also increase the accuracy. Further this solution can be applied to detect various other diseases in the future.

3.4 Problem Solution Fit



Project Title: Early Detection of Chronic Kidney Disease using Machine Learning **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID00056

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

The main users of this project are doctors, nurses, patients and also common people who have some symptoms or like to get tested for CKD.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

-Costly tests for diagnosis
-Longer time to Detect diseases
-Human error due to negligence
-Lack of skilled doctors

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Diagnosis by doctors and lab technicians manually with the help of various test results

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

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

-Diagnose the disease in the early stage
-Ensure accurate prediction results
-Design a user friendly interface
-Make the application easily accessible to the customers

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

-Expensive tests for diagnosis
-Unavailability of facilities in hospitals
-Human error in manual diagnosis
-Lack of experience of doctors

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

-Visit the hospital for diagnosis and treatment
-Discuss with friends and relatives about the symptoms
-Search through internet to get insights on the symptoms

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Imprecise and expensive test results that make diagnosis slower.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before: Frustrated, hopeless, Insecured
After: Confident, Peaceful, Positive attitude

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

An ML model to prevent manual errors in diagnosis of CKD through test results of other diseases thereby detecting the disease in its early stages accurately.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

Search through the internet to get insights on the disease and symptoms

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

- Visit the hospital for diagnosis and treatment
- Discuss with friends and relatives about the symptoms

Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

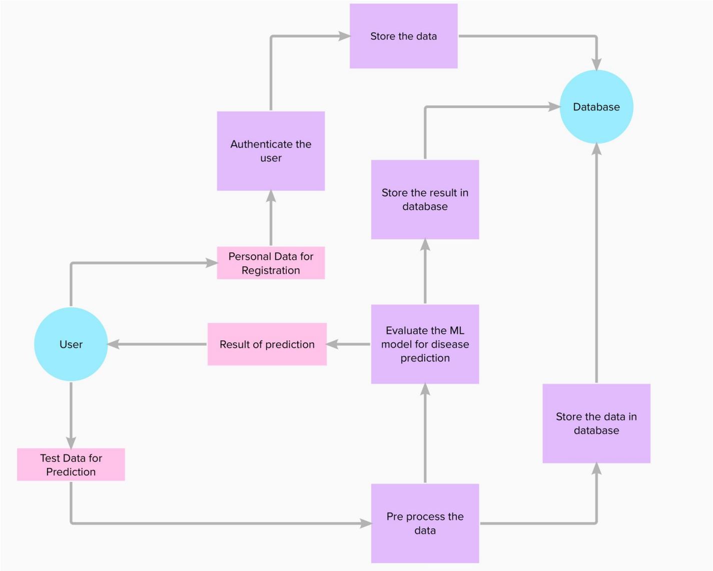
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Store the credentials in database
FR-2	User Confirmation	Confirmation via verification of credentials from database
FR-3	Data Collection	Input data through Form
FR-4	Data Analysis	Pre-process the data Check correct format of Input
FR-5	Prediction of disease	Evaluate the ML model with the data
FR-6	Provide output to the user	Display the prediction result in UI

4.2 Non- Functional Requirements

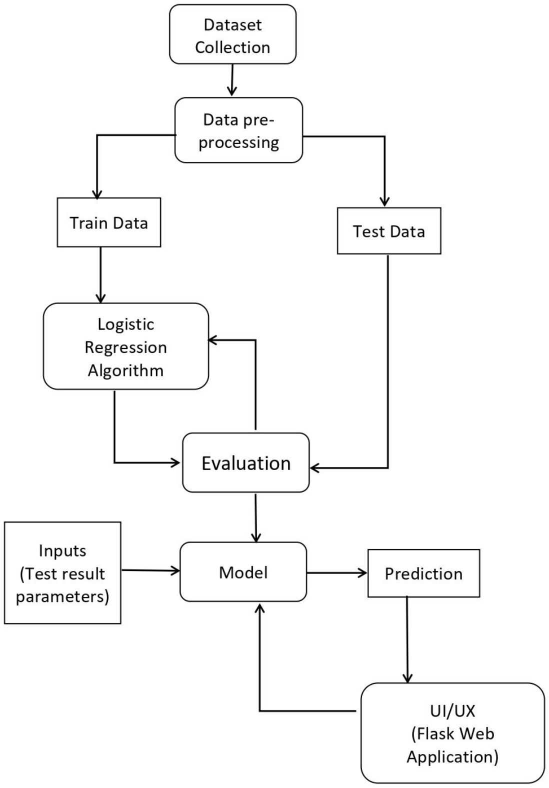
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The Web UI is simple and easy for navigation and understanding even by common people with the help of known labels and descriptions.
NFR-2	**Security**	The users are required to register so that only verified users can use the application.
NFR-3	**Reliability**	The prediction result is expected to be accurate and free from error.
NFR-4	**Performance**	The webpage must load quickly even with slow internet connection. The ML model must predict the result with great accuracy and speed.
NFR-5	**Availability**	The website should be without any downtime for updates.
NFR-6	**Scalability**	This simple web application must have high scalability for supporting large number of users.

5. PROJECT DESIGN

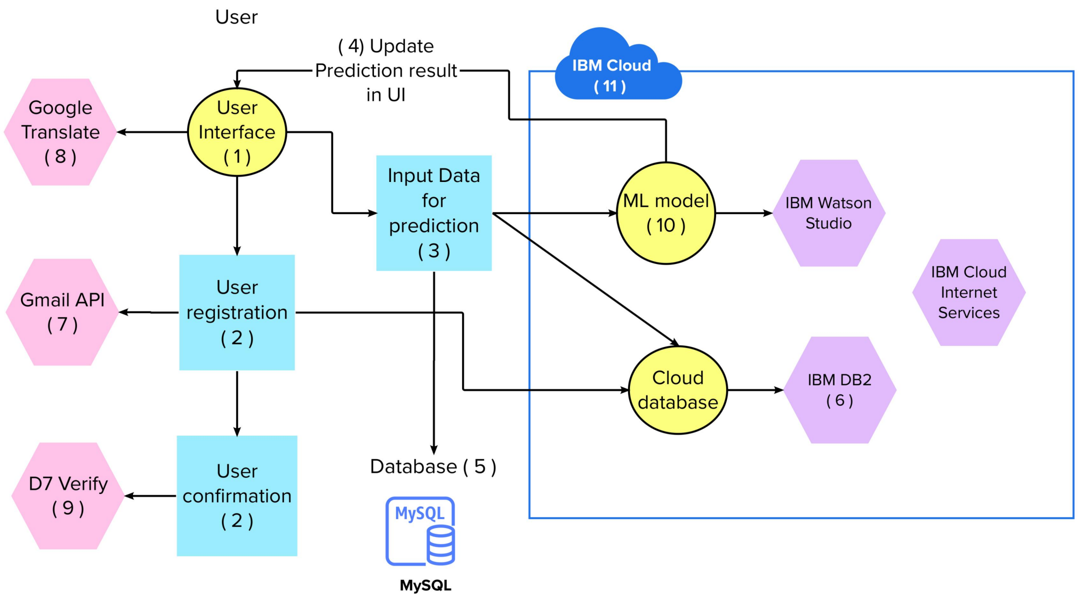
5.1 Data Flow Diagram


Store the data
Database
Authenticate the user
Store the result in database
Personal Data for Registration
User
Result of prediction
Evaluate the ML model for disease prediction
Store the data in database
Test Data for Prediction
Pre process the data

5.2 Solution Architecture


Dataset Collection
Data pre-processing
Train Data
Test Data
Logistic Regression Algorithm
Evaluation
Inputs (Test result parameters)
Model
Prediction
UI/UX (Flask Web Application)

Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User (Common people and doctors)	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 .	I can access my dashboard	High	Sprint-1
	User Confirmation	USN-2	As a user, I will receive confirmation once I have registered for the application	I can receive confirmation message in dashboard	High	Sprint-1
	Data Collection	USN-3	As a user, I will enter the input data for disease prediction in the form.	I can give the data to predict.	High	Sprint-1
	Provide output to the user	USN-4	As a user, I will get the result of disease prediction in the dashboard.	I can know if I am affected or not.	High	Sprint-1
Administrator	Data Analysis	USN-5	As the admin, I will develop modules to preprocess and store the data.	I can change the data into right format.	High	Sprint-1
	Prediction of disease	USN-6	As the admin, I will build a Machine Learning model to predict the disease.	I can deploy the model within a website	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my name, email, password.	10	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Confirmation	USN-2	As a user, I will receive confirmation in the dashboard once I have registered for the application	10	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E
Sprint-2	Prediction of disease	USN-6	As the admin, I will build a Machine Learning model to predict the disease	10	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E
Sprint-2	Data Collection	USN-3	As a user, I will enter the input data for disease prediction in the form	10	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E
Sprint-3	Data Analysis	USN-5	As the admin, I will develop modules to preprocess and store the data.	10	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E
Sprint-3	Provide output to the user	USN-4	As a user, I will get the result of disease prediction in the dashboard.	10	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E
Sprint-4	Final Delivery	USN-7	Deploy the application in IBM cloud and make it available for use.	20	High	Gayathri G Dafney Jervis W Akash R Jefray Sharon E

6.2 Sprint Delivery Schedule

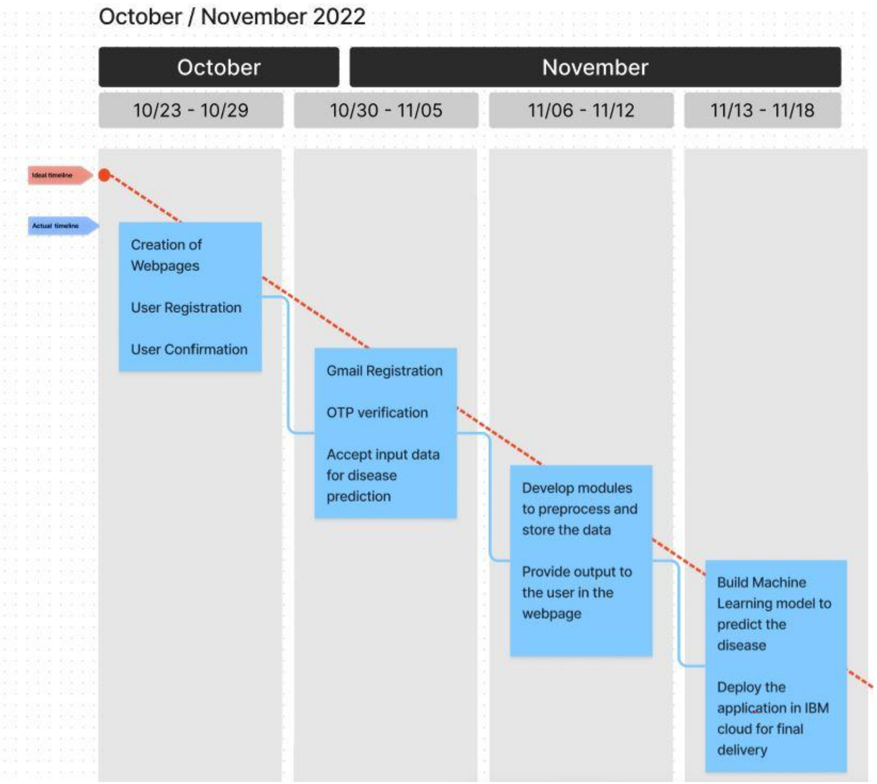
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

Velocity:

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

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

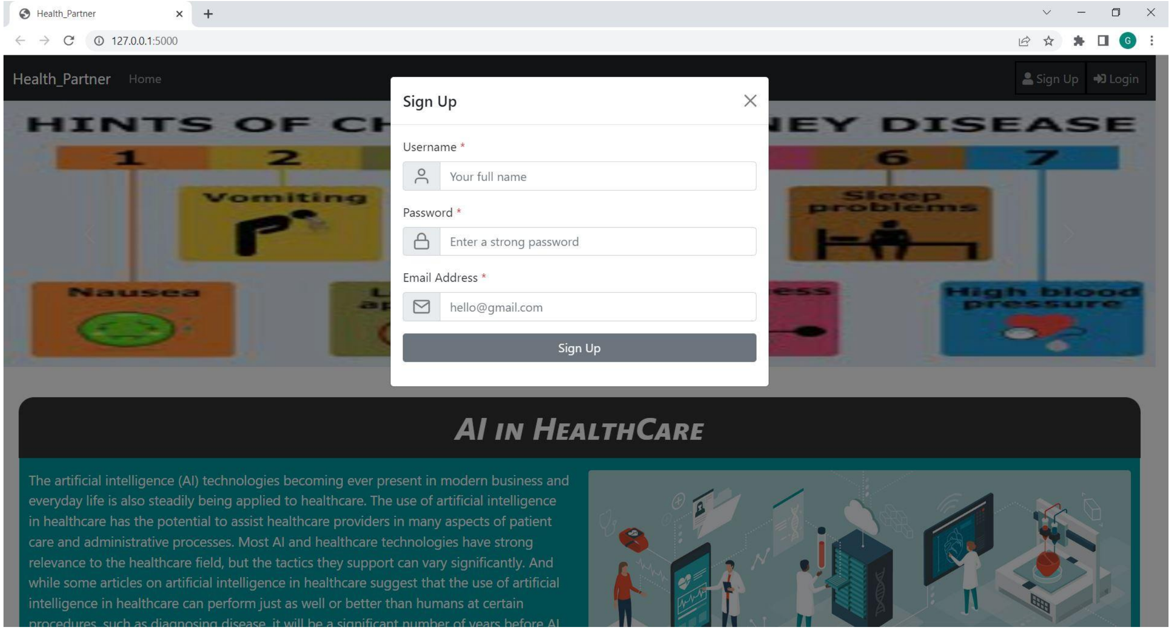
Burndown Chart:



7. CODING & SOLUTIONING

7.1 Register

A sign-up button in the home-screen's navigation bar offers a way for users to register with their username, password and email address. When the button is clicked, a modal pops up as an overlay on the home page where the user can enter the details and click the Sign up button at the end of the form. Now the details are submitted to the database and stored for further use. In case the username already exists in the database, then the user is notified with a message in the home page itself. Otherwise the new user is registered successfully.



Sign Up

Username *

Your full name

Password *

Enter a strong password

Email Address *

hello@gmail.com

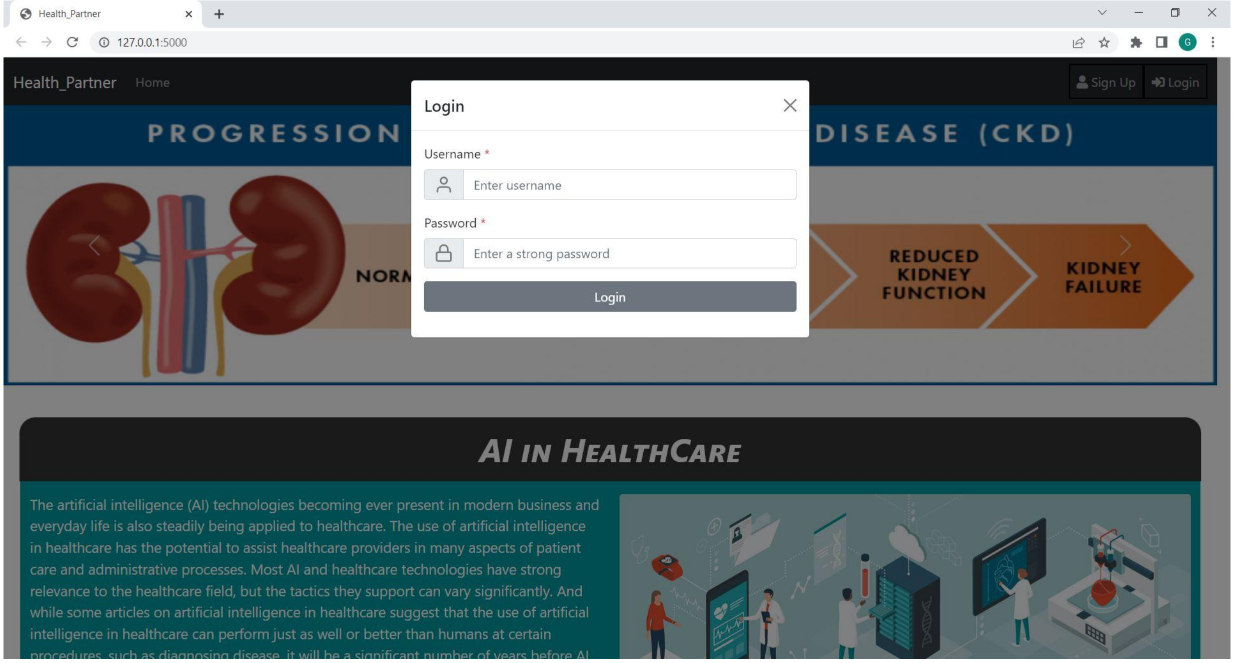
Sign Up

AI IN HEALTHCARE

The artificial intelligence (AI) technologies becoming ever present in modern business and everyday life is also steadily being applied to healthcare. The use of artificial intelligence in healthcare has the potential to assist healthcare providers in many aspects of patient care and administrative processes. Most AI and healthcare technologies have strong relevance to the healthcare field, but the tactics they support can vary significantly. And while some articles on artificial intelligence in healthcare suggest that the use of artificial intelligence in healthcare can perform just as well or better than humans at certain procedures, such as diagnosing disease, it will be a significant number of years before AI

7.2 Login

A login button in the home-screen’s navigation bar offers a way for users to login with their username and password. When the button is clicked, a modal pops up as an overlay on the home page where the user can enter the details and click the Login button at the end of the form. Now the details are checked with the data in the database. In case the username and password exists in the database, then the user is taken to the dashboard page with the welcome message. Otherwise the user is notified with a error message in the home page itself.



Login

Username *

Enter username

Password *

Enter a strong password

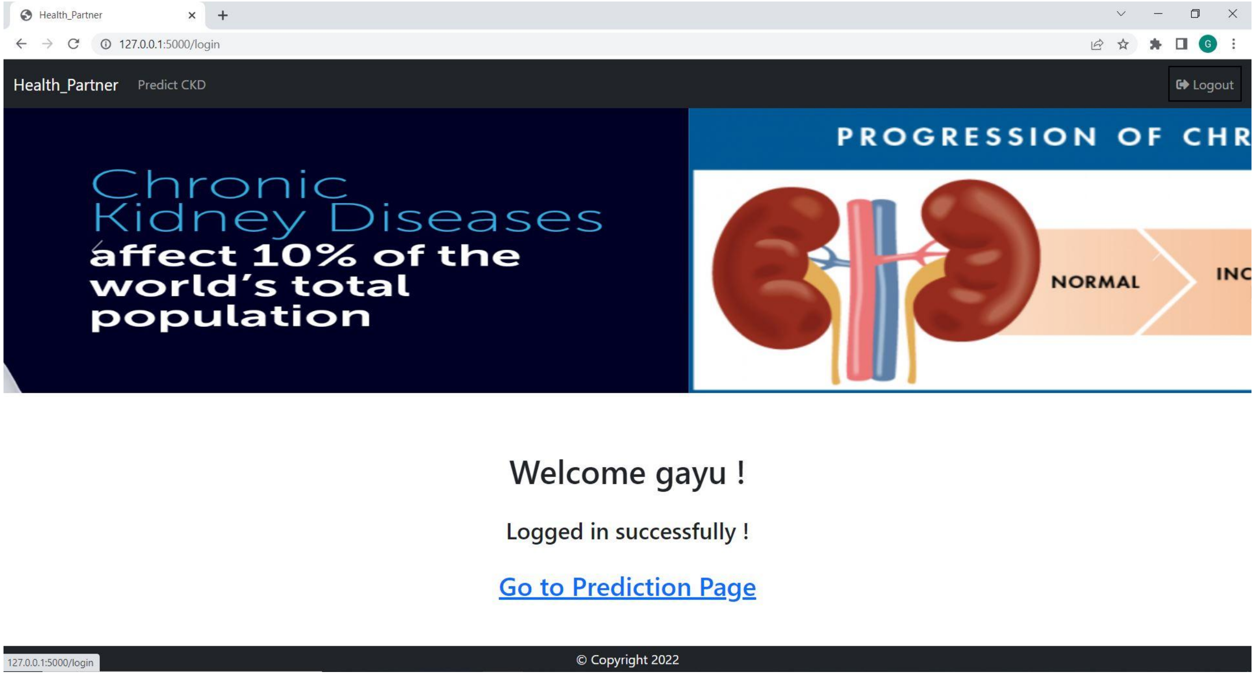
Login

AI IN HEALTHCARE

The artificial intelligence (AI) technologies becoming ever present in modern business and everyday life is also steadily being applied to healthcare. The use of artificial intelligence in healthcare has the potential to assist healthcare providers in many aspects of patient care and administrative processes. Most AI and healthcare technologies have strong relevance to the healthcare field, but the tactics they support can vary significantly. And while some articles on artificial intelligence in healthcare suggest that the use of artificial intelligence in healthcare can perform just as well or better than humans at certain procedures, such as diagnosing disease, it will be a significant number of years before AI

7.3 Dashboard

The user enters the dashboard after logging in. It shows a welcome message with a link to the prediction page.



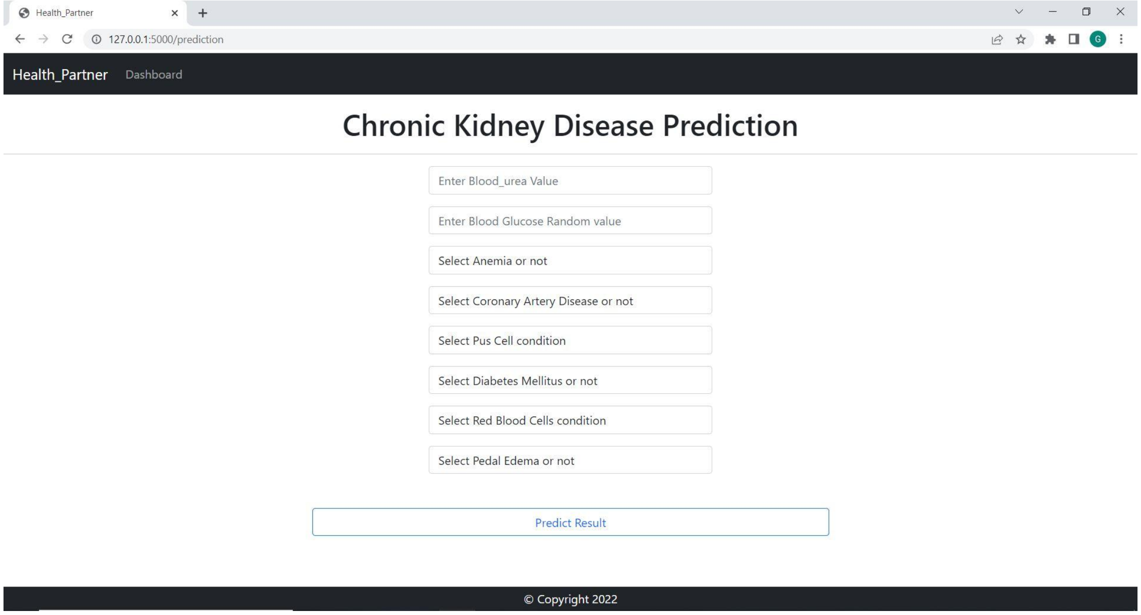
Welcome gayu !

Logged in successfully !

Go to Prediction Page

7.4 Predict

The prediction page contains a form for the users to enter the input for various parameters for prediction. After entering the data and selecting from the drop down list, the user can click the Predict Result button for the output.



Chronic Kidney Disease Prediction

Enter Blood_urea Value

Enter Blood Glucose Random value

Select Anemia or not

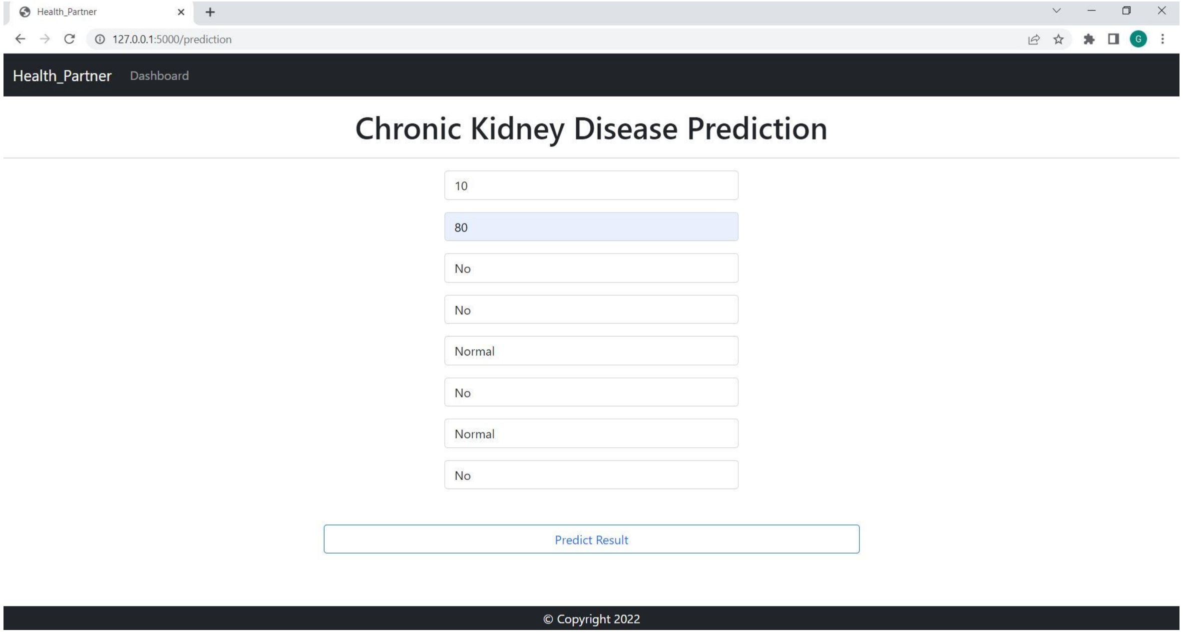
Select Coronary Artery Disease or not

Select Pus Cell condition

Select Diabetes Mellitus or not

Select Red Blood Cells condition

Select Pedal Edema or not



Chronic Kidney Disease Prediction

10

80

No

No

Normal

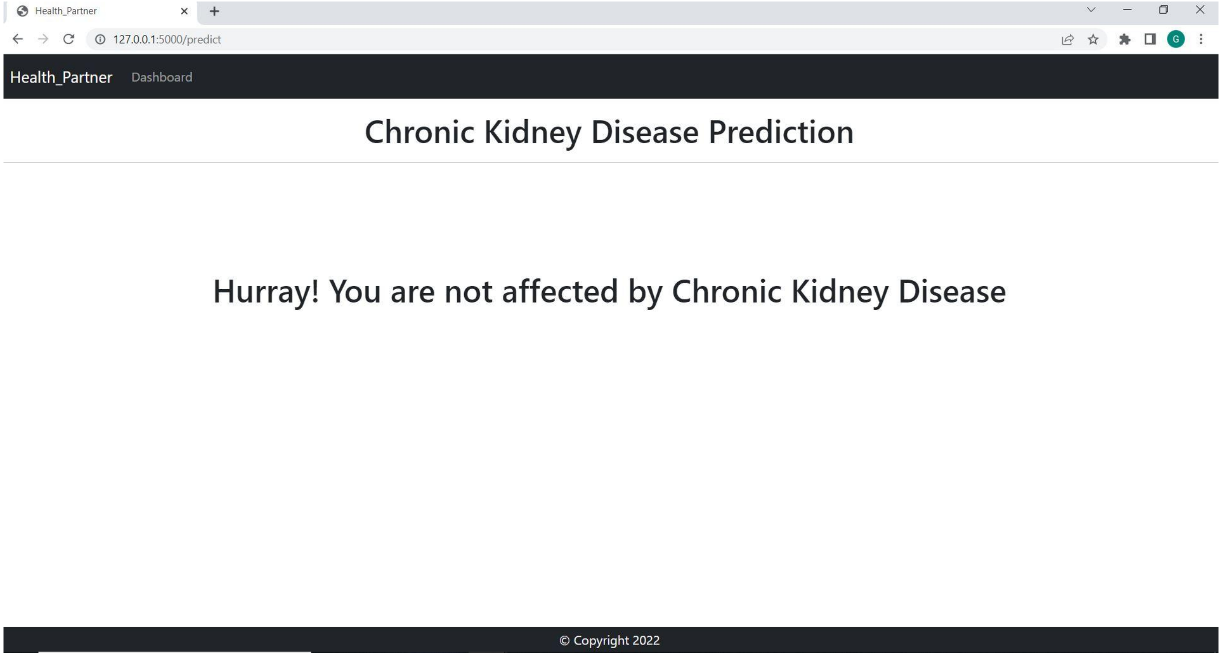
No

Normal

No

7.5 Result

Once the data has been given for prediction, the Machine Learning model predicts the result and displays it in the web page.



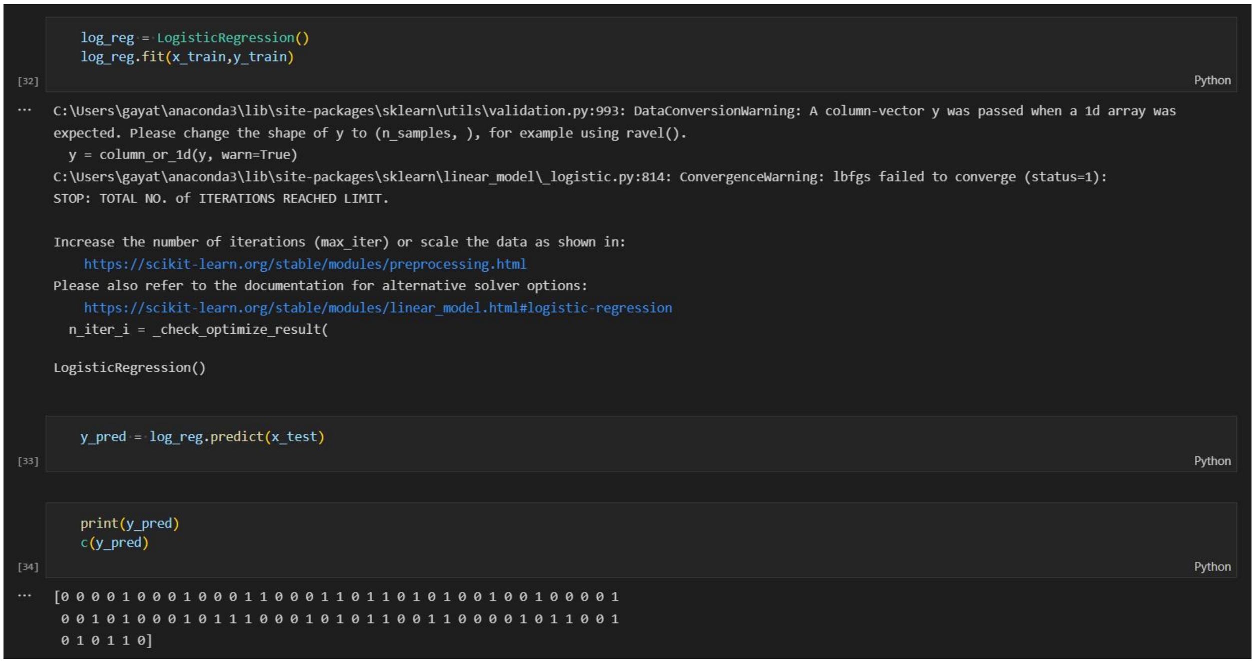
Chronic Kidney Disease Prediction

Hurray! You are not affected by Chronic Kidney Disease

7.6 Machine Learning Model

As the prediction for our model is classification type, we apply a logistic regression algorithm on our dataset.

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.



```
log_reg = LogisticRegression()
log_reg.fit(x_train,y_train)
```

```
C:\Users\gayat\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\gayat\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(

LogisticRegression()
```

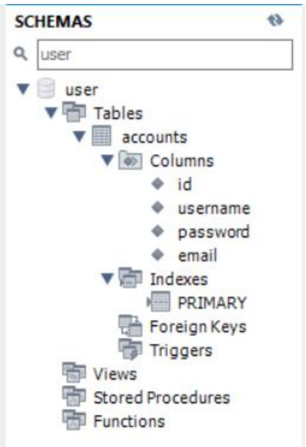
```
y_pred = log_reg.predict(x_test)
```

```
print(y_pred)
c(y_pred)
```

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

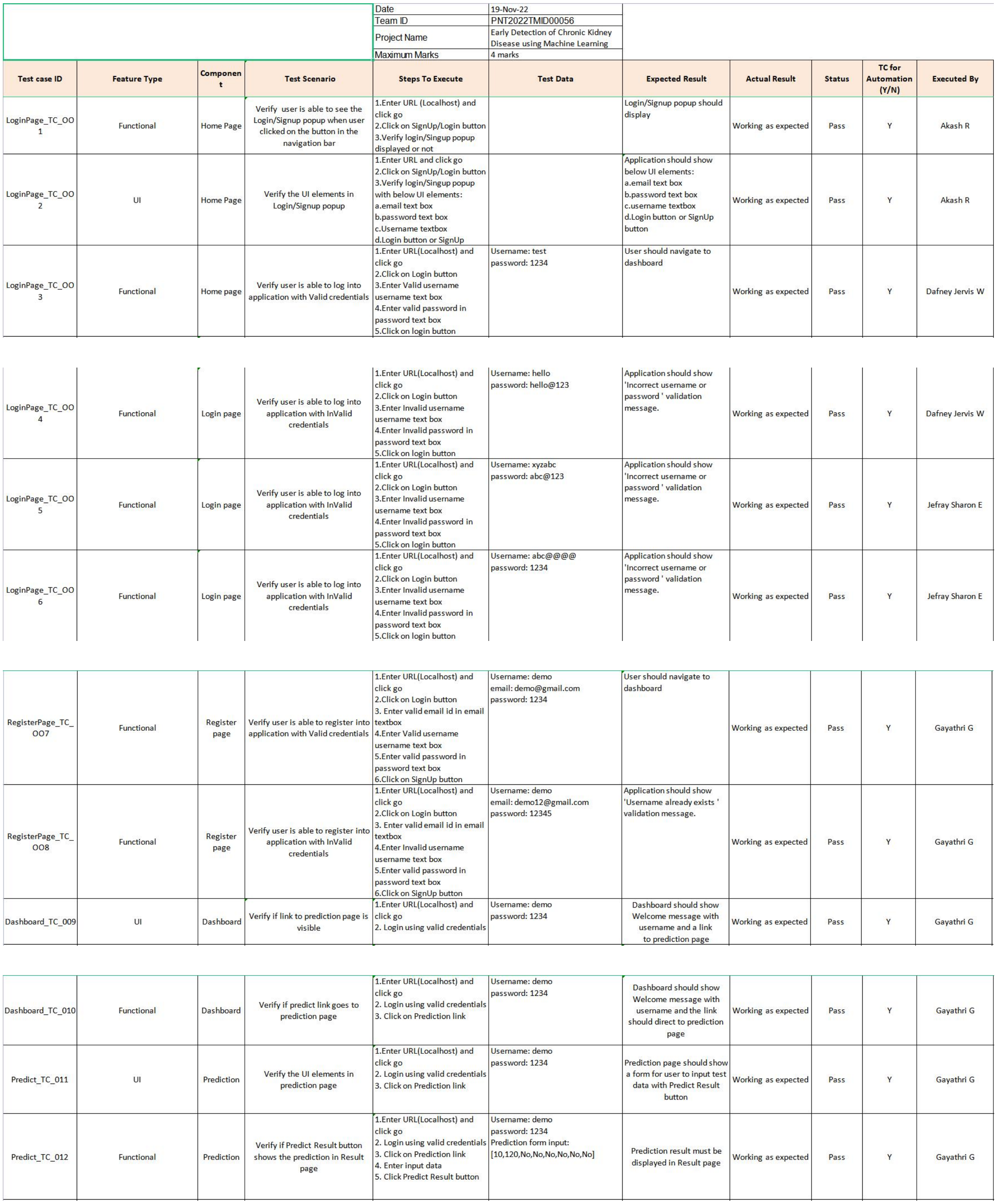
7.7 Database Schema

A database named user is created with a table accounts with the columns id, username, password and email to store the credentials of the users.



8. TESTING

8.1 Test Cases



Date	19-Nov-22
Team ID	PNT2022TMID00056
Project Name	Early Detection of Chronic Kidney Disease using Machine Learning
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	TC for Automation (Y/N)	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on the button in the navigation bar	1.Enter URL (Localhost) and click go 2.Click on SignUp/Login button 3.Verify login/Singup popup displayed or not		Login/Signup popup should display	Working as expected	Pass	Y	Akash R
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	1.Enter URL and click go 2.Click on SignUp/Login button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Username textbox d.Login button or SignUp		Application should show below UI elements: a.email text box b.password text box c.username textbox d.Login button or SignUp button	Working as expected	Pass	Y	Akash R
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL(Localhost) and click go 2.Click on Login button 3.Enter Valid username username text box 4.Enter valid password in password text box 5.Click on login button	Username: test password: 1234	User should navigate to dashboard	Working as expected	Pass	Y	Dafney Jervis W
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on Login button 3.Enter Invalid username username text box 4.Enter Invalid password in password text box 5.Click on login button	Username: hello password: hello@123	Application should show 'Incorrect username or password ' validation message.	Working as expected	Pass	Y	Dafney Jervis W
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on Login button 3.Enter Invalid username username text box 4.Enter Invalid password in password text box 5.Click on login button	Username: xyzabc password: abc@123	Application should show 'Incorrect username or password ' validation message.	Working as expected	Pass	Y	Jefray Sharon E
LoginPage_TC_OO6	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on Login button 3.Enter Invalid username username text box 4.Enter Invalid password in password text box 5.Click on login button	Username: abc@@@@ password: 1234	Application should show 'Incorrect username or password ' validation message.	Working as expected	Pass	Y	Jefray Sharon E
RegisterPage_TC_OO7	Functional	Register page	Verify user is able to register into application with Valid credentials	1.Enter URL(Localhost) and click go 2.Click on Login button 3. Enter valid email id in email textbox 4.Enter Valid username username text box 5.Enter valid password in password text box 6.Click on SignUp button	Username: demo email: demo@gmail.com password: 1234	User should navigate to dashboard	Working as expected	Pass	Y	Gayathri G
RegisterPage_TC_OO8	Functional	Register page	Verify user is able to register into application with InValid credentials	1.Enter URL(Localhost) and click go 2.Click on Login button 3. Enter valid email id in email textbox 4.Enter Invalid username username text box 5.Enter valid password in password text box 6.Click on SignUp button	Username: demo email: demo12@gmail.com password: 12345	Application should show 'Username already exists ' validation message.	Working as expected	Pass	Y	Gayathri G
Dashboard_TC_009	UI	Dashboard	Verify if link to prediction page is visible	1.Enter URL(Localhost) and click go 2. Login using valid credentials	Username: demo password: 1234	Dashboard should show Welcome message with username and a link to prediction page	Working as expected	Pass	Y	Gayathri G
Dashboard_TC_010	Functional	Dashboard	Verify if predict link goes to prediction page	1.Enter URL(Localhost) and click go 2. Login using valid credentials 3. Click on Prediction link	Username: demo password: 1234	Dashboard should show Welcome message with username and the link should direct to prediction page	Working as expected	Pass	Y	Gayathri G
Predict_TC_011	UI	Prediction	Verify the UI elements in prediction page	1.Enter URL(Localhost) and click go 2. Login using valid credentials 3. Click on Prediction link	Username: demo password: 1234	Prediction page should show a form for user to input test data with Predict Result button	Working as expected	Pass	Y	Gayathri G
Predict_TC_012	Functional	Prediction	Verify if Predict Result button shows the prediction in Result page	1.Enter URL(Localhost) and click go 2. Login using valid credentials 3. Click on Prediction link 4. Enter input data 5. Click Predict Result button	Username: demo password: 1234 Prediction form input: [10,120,No,No,No,No,No,No]	Prediction result must be displayed in Result page	Working as expected	Pass	Y	Gayathri G

8.2 User Acceptance Testing

8.2.1 Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	4	2	3	14
Duplicate	1	0	0	0	1
External	1	2	0	0	3
Fixed	7	6	2	3	18
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	7	6	2	3	18

8.2.2 Test Case Analysis

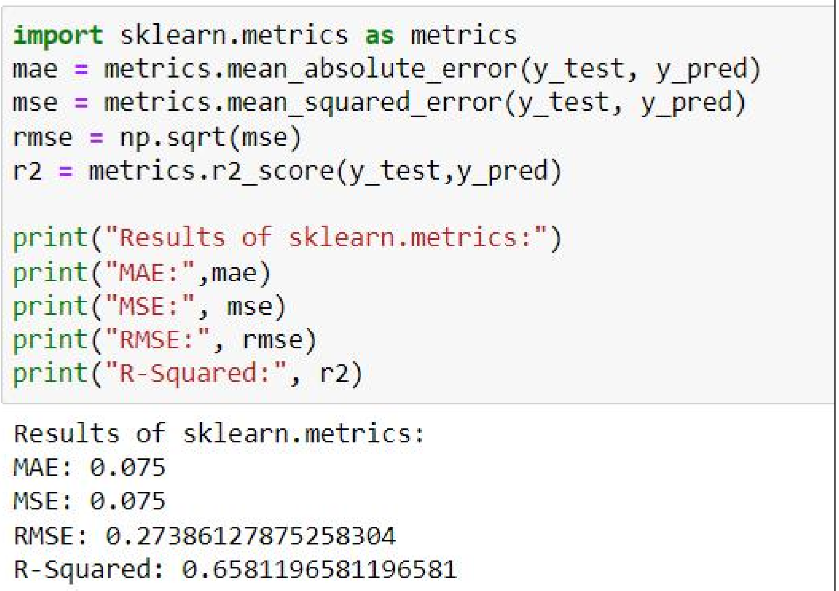
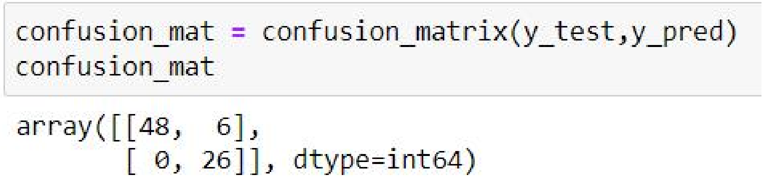
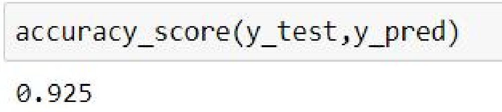
This report shows the number of test cases that have passed, failed, and untested

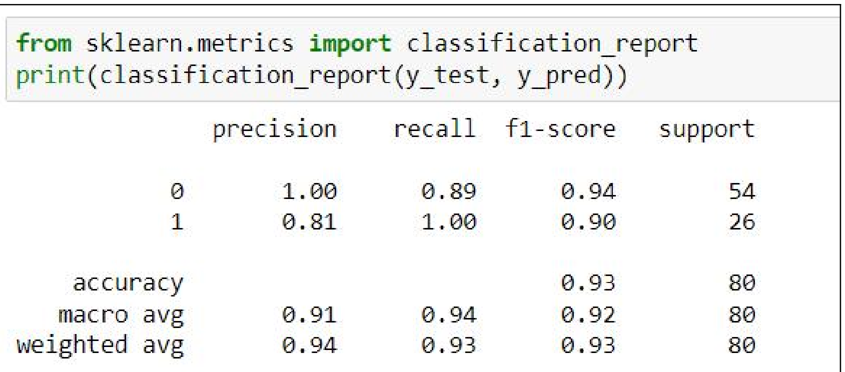
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	10	0	0	10
Security	2	0	0	2
Outsource Shipping	0	0	0	0
Exception Reporting	4	0	0	4
Final Report Output	1	0	0	1

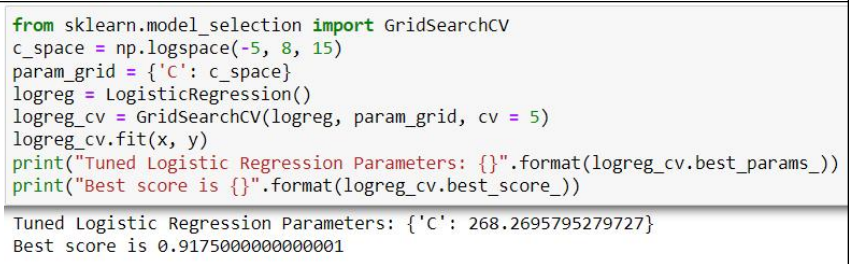
9. RESULTS

9.1 Performance Metrics

S.No	Parameter	Values	Screenshot

1.	Metrics	**Regression Model:**
		MAE - 0.075
		MSE - 0.075
		RMSE - 0.27386
		R2 score - 0.65811
		Classification Model:
		Confusion Matrix
		Accuracy Score
			
			`import sklearn.metrics as metrics`
			`mae = metrics.mean_absolute_error(y_test, y_pred)`
			`mse = metrics.mean_squared_error(y_test, y_pred)`
			`rmse = np.sqrt(mse)`
			`r2 = metrics.r2_score(y_test,y_pred)`
			`print("Results of sklearn.metrics:")`
			`print("MAE:",mae)`
			`print("MSE:", mse)`
			`print("RMSE:", rmse)`
			`print("R-Squared:", r2)`
			`Results of sklearn.metrics:`
			`MAE: 0.075`
			`MSE: 0.075`
			`RMSE: 0.27386127875258304`
			`R-Squared: 0.6581196581196581`
			
			`confusion_mat = confusion_matrix(y_test,y_pred)`
			`confusion_mat`
			`array([[48,  6],`
			`       [ 0, 26]], dtype=int64)`
			
			`accuracy_score(y_test,y_pred)`
			`0.925`

		Classification Report	
			from sklearn.metrics import classification_report
			print(classification_report(y_test, y_pred))
			
			`              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 - GridSearch CV with best score 0.9175	
			from sklearn.model_selection import GridSearchCV
			c_space = np.logspace(-5, 8, 15)
			param_grid = {'C': c_space}
			logreg = LogisticRegression()
			logreg_cv = GridSearchCV(logreg, param_grid, cv = 5)
			logreg_cv.fit(x, y)
			print("Tuned Logistic Regression Parameters: {}".format(logreg_cv.best_params_))
			print("Best score is {}".format(logreg_cv.best_score_))
			
			Tuned Logistic Regression Parameters: {'C': 268.2695795279727}
			Best score is 0.9175000000000001

10. ADVANTAGES AND DISADVANTAGES

10.1 Advantages

- User friendly and simple user interface
- Get instantaneous results for prediction
- High accuracy prediction
- Simple registration with minimal details
- Saves time and cost in manual prediction

10.2 Disadvantages

- Need more datasets to train the model
- Needs evaluation of multiple parameters

11. CONCLUSION

In this project we have built a logistic regression model that predicts whether the user is affected by Chronic Kidney Disease(CKD) or not. This model makes use of only 8 important parameters for prediction thereby minimizing the need for costly tests to predict CKD manually. These parameters can be taken from normal test results that are done in regular check-ups rather than some specific test for kidney disease. This model is integrated with a flask based web application so that the website can be deployed for the users to make use of it. The website is designed with user friendly interface and easy to use controls. Thus early detection of chronic kidney disease has been made possible by using machine learning.

12. FUTURE SCOPE

This project does early detection of chronic kidney disease by using machine learning algorithm. The accuracy of the logistic regression model used can be improved in the future by training the model with more data. The number of parameters used for prediction can be reduced further. Also, the project can be improved to predict the severity or stage of chronic kidney disease in the future.

13. APPENDIX

Source Code

app.py

```
import flask
import numpy as np
import pandas as pd
from flask import Flask, request, render_template, url_for, session, redirect
import pickle
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re
app = flask.Flask(__name__,static_folder='')
model = pickle.load(open('./CKD.pkl','rb'))
@app.route('/')
def homePage():
   return render_template('index.html')
@app.route('/dashboard')
def dashboard():
   return render_template('dashboard.html',name=session['username'])
```

```
@app.route('/prediction', methods=['POST','GET'])
def predictCKD():
    return render_template('predict.html')
@app.route('/predict', methods=['POST','GET'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    features_name = ['blood_urea', 'blood_glucose_random', 'anemia',
'coronary_artery_disease','pus_cell','diabetes_mellitus','red_blood_cells','pedal_edema']
    df = pd.DataFrame(features_value, columns=features_name)
    output = model.predict(df)
    if(output==0):
        text="Oops! You are detected with Chronic Kidney Disease."
    else:
        text="Hurray! You are not affected by Chronic Kidney Disease"
    return render_template('result.html',prediction_text=text)
app.secret_key = '123'
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'root'
app.config['MYSQL_DB'] = 'user'

mysql = MySQL(app)

@app.route('/login', methods =['GET', 'POST'])
def login():
    msg = "
    #if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
    username = request.form['username']
    password = request.form['password']
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM accounts WHERE username = % s AND password = % s',
(username, password, ))
```

```
    account = cursor.fetchone()
    if account:
        session['loggedin'] = True
        session['id'] = account['id']
        session['username'] = account['username']
        msg = 'Logged in successfully !'
        return render_template('dashboard.html', msg = msg, name=username)
    else:
        msg = 'Incorrect username / password !'
    return render_template('index.html', msg = msg)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect('/')

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    #if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :
    username = request.form['username']
    password = request.form['password']
    email = request.form['email']
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM accounts WHERE username = % s', (username, ))
    account = cursor.fetchone()
    if account:
        msg = 'Username already exists !'
        return render_template('index.html', msg = msg)
    else:
        cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username,
password, email ))
        mysql.connection.commit()
        msg = 'You have successfully registered !'
        return render_template('dashboard.html', msg = msg, name=username)

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

app_ibm.py

```
   import requests
   import flask
   import numpy as np
   import pandas as pd
   from flask import Flask, request, render_template, url_for, session, redirect
   from flask_mysqldb import MySQL
   import MySQLdb.cursors
   import re
   API_KEY = "I1JHnBDQ7KxJ_pRHg948Nit3rjLLPZGz0bdhTKSTizAU"
   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}

   app = flask.Flask(__name__,static_folder='')
   @app.route('/')
   def homePage():
       return render_template('index.html')

   @app.route('/dashboard')
   def dashboard():
       return render_template('dashboard.html',name=session['username'])
```

```
@app.route('/prediction', methods=['POST','GET'])
def predictCKD():
    return render_template('predict.html')
@app.route('/predict', methods=['POST','GET'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    features_name = ['blood_urea', 'blood_glucose_random', 'anemia',
'coronary_artery_disease','pus_cell','diabetes_mellitus','red_blood_cells','pedal_edema']
    payload_scoring = {"input_data": [{"fields": ['blood_urea', 'blood_glucose_random',
'anemia', 'coronary_artery_disease','pus_cell','diabetes_mellitus','red_blood_cells','pedal_edema'],
"values": [input_features]}]}

    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/fcd0538f-628a-4e12-94af-
b523e0495be5/predictions?version=2022-11-11', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    predictions=response_scoring.json()
    print(predictions)
    output = predictions['predictions'][0]['values'][0][0]
    if(output==0):
        text="Oops! You are detected with Chronic Kidney Disease."
    else:
        text="Hurray! You are not affected by Chronic Kidney Disease"
    return render_template('result.html',prediction_text=text)
#extra code
app.secret_key = '123'

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'root'
app.config['MYSQL_DB'] = 'user'

mysql = MySQL(app)
```

```
@app.route('/login', methods =['GET', 'POST'])
def login():
    msg = ''
    #if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
    username = request.form['username']
    password = request.form['password']
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM accounts WHERE username = % s AND password = % s', (username, password, ))
    account = cursor.fetchone()
    if account:
        session['loggedin'] = True
        session['id'] = account['id']
        session['username'] = account['username']
        msg = 'Logged in successfully !'
        return render_template('dashboard.html', msg = msg, name=username)
    else:
        msg = 'Incorrect username / password !'
    return render_template('index.html', msg = msg)

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return redirect('/')

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    #if request.method == 'POST' and 'username' in request.form and 'password' in request.form and 'email' in request.form :
```

```
    username = request.form['username']
    password = request.form['password']
    email = request.form['email']
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM accounts WHERE username = % s', (username, ))
    account = cursor.fetchone()
    if account:
        msg = 'Username already exists !'
        return render_template('index.html', msg = msg)
    else:
        cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username, password, email ))
        mysql.connection.commit()
        msg = 'You have successfully registered !'
        return render_template('dashboard.html', msg = msg, name=username)

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

Git Repo Link:

https://github.com/IBM-EPBL/IBM-Project-11080-1659260039

Demo Video Link:

https://drive.google.com/file/d/18G_VgU-ct-to7mFwFz9zwvA-wiSh-in0/view?usp=sharing