EARLY DETECTION OF CHRONIC KIDNEY DISEASE USING MACHINE LEARNING

TEAM ID : PNT2022TMID0462

A PROJECT REPORT

Submitted by

Chaganti Praneeth Kumar
Shiek Mohideen M
Vallarasu S
Vishwa B

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SRI SAI RAM ENGINEERING COLLEGE

TABLE OF CONTENTS

1. INTRODUCTION

Project Overview

This article objects to predict Chronic Kidney Disease based on full features and important features of CKD dataset. Machine learning technique has become reliable for medical treatment. With the help of a machine learning classifier algorithms, the doctor can detect the disease on time. For this perspective, Chronic Kidney Disease prediction has been discussed in this article.. 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. 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%.

Purpose

This research article primarily aims to predict whether a person has Chronic Kidney Disease or not. In this perception, seven different machine learning classifiers were applied on the dataset. All the algorithms were running with both full features and selected features. SMOTE was used for over sampling and all the results were recorded. All the machine learning model results were also compared with one deep neural network algorithm. Deep learning neural network was used with two hidden layers.

2. LITERATURE SURVEY

Existing Problem

SURVEY 1 : AUTHORS: Himanshu Kriplani, Bhumi Patel and Sudipta Roy

TITLE : Prediction of Chronic Kidney Diseases Using Deep Artificial Neural Network Technique.

METHODS : This project presents a method to detect the chronic kidney disease and methodologies to diagnose chronic kidney disease is a challenging problem which can reduce the cost of treatment. We studied 224 records of chronic kidney disease available on the UCI machine learning repository named chronic kidney diseases dating back to 2015. Our proposed method is based on deep neural network which predicts the presence or absence of chronic kidney disease with an accuracy of 97%. Compared to other available algorithms, the model we built shows better results which is implemented using the cross-validation technique to keep the model safe from over fitting. This automatic chronic kidney disease treatment helps reduce the kidney damage progression, but for this chronic kidney disease detection at initial stage is necessary.

SURVEY 2 :

AUTHORS : Hongquan Peng , Haibin Zhu , Chi WaAo Ieong , Tao Tao , Tsung Yang Tsai , Zhi Liu

TITLE : A two-stage neural network prediction of chronic kidney disease

METHODS : This paper presents a method to detect chronic kidney disease (CKD) plays a pivotal role in early diagnosis and treatment. Measured glomerular

filtration rate (mGFR) is considered the benchmark indicator in measuring the kidney function. However, due to the high resource cost of measuring mGFR, it is usually approximated by the estimated glomerular filtration rate, underscoring an urgent need for more precise and stable approaches. With the introduction of novel machine learning methodologies, prediction performance is shown to be significantly improved across all available data, but the performance is still limited because of the lack of models in dealing with ultra-high dimensional datasets. This study aims to provide a two-stage neural network approach for prediction of GFR and to suggest some other useful biomarkers obtained from the blood metabolites in measuring GFR. It is a composite of feature shrinkage and neural network when the number of features is much larger than the number of training samples. The results show that the proposed method outperforms the existing ones, such as convolution neural network and direct deep neural network.

SURVEY 3 :

AUTHORS : Deepak K N , Adhwaidh P S , Akshay P D , Athira K S , Jisna Jayan

TITLE : Chronic Kidney Disease Prediction system using Machine Learning

METHODS : This paper reviews and analyzes the Chronic kidney disease (CKD)

is a global health issue that causes a high rate of morbidity and mortality, as well as the onset of additional diseases. Because there are no clear symptoms in the early stages of CKD, people frequently miss it. Early identification of CKD allows patients to obtain timely treatment to slow the disease's progression. Due to their rapid and precise recognition capabilities, machine learning models can successfully assist doctors in achieving this goal. We propose a machine learning framework for diagnosing CKD in this paper. The CKD data set was taken from kaggle, which has a substantial number of missing values. We employ multiple machine learning methods such as DT, SVM, and DNN to analyze data from CKD patients with 21 characteristics and 400 records. The dataset is preprocessed by filling in missing data and normalizing it. To increase accuracy and save training time, the most relevant features from the dataset are chosen. Image processing and letter recognition are used to automatically input the attributes.

References

1.Q.-L. Zhang and D. Rothenbacher, ‘ Prevalence of chronic kidney disease in population-based studies: Systematic review,’ BMC Public Health, vol. 8, no. 1, p. 117, Dec. 2008.

2. W. M. McClellan, D. G. Warnock, S. Judd, P. Muntner, R. Kewalramani, M.

Cushman, L. A. McClure, B. B. Newsome, and G. Howard, ‘ Albuminuria and racial disparities in the risk for ESRD,’ J. Amer. Soc. Nephrol., vol. 22, no. 9, pp. 1721–1728, Aug. 2011.

3. M. K. Haroun, ‘ Risk factors for chronic kidney disease: A prospective study of 23,534 men and women in Washington County, Maryland,’ J. Amer. Soc. Nephrol., vol. 14, no. 11, pp. 2934–2941, Nov. 20.

4. W. D. Souza, L. C. D. Abreu, L. G. D. SilvaI, and I. M. P. Bezerra, ‘ Incidence of chronic kidney disease hospitalisations and mortality in Espírito Santo between 1996 to 2017,’ WisitCheungpasitporn, Univ. Mississippi Medical Center, Rochester, MN, USA, Tech. Rep., 2019, doi: 10.1371/journal.pone.0224889.

5. W. Mula-Abed, K. A. Rasadi, and D. Al-Riyami, ‘ Estimated glomerular filtration rate (eGFR): A serum creatinine-based test for the detection of chronic kidney disease and its impact on clinical practice,’ Oman Med. J., vol. 27, no. 4, pp. 339–340, 2012.

6. A. S. Levey, D. Cattran, A. Friedman, W. G. Miller, J. Sedor, K. Tuttle, B. Kasiske, and T. Hostetter, ‘ Proteinuria as a surrogate outcome in CKD: Report of a scientific workshop sponsored by the national kidney foundation and the US food

and drug administration,' Amer. J. Kidney Diseases, vol. 54, no. 2, pp. 205–226, Aug. 2009.

Problem Statement Definition

To identify and manage patients who have early stages of chronic kidney disease may slow or prevent the progression to end stage kidney disease and reduce cardiovascular complications caused due to diabetes and high blood.

Who does the problem affect?

Chronic Kidney Disease is more common in people aged 65years or older (38%) than in people aged 45–64 years (12%) or 18–44 years (6%). CKD is slightly more common in women in (14%)than men (12%).

What are the boundaries of the problem?

A disease in GFR may also be a marker of kidney disease and precedes the onset of kidney failure . Below 60 ml/min/1.73 m , the prevalence of complications of CKD Increases , as does the risk of cardiovascular disease . Albuminuria (ACR ≥ 30 mg/g), Urine sediment abnormalities, Electrolyte and other abnormalities due to

tubular disorders, Abnormalities detected by histology, Structural abnormalities detected by imaging, History of kidney transplantation.

What is the issue?

Kidneys are damaged and can't filter blood the way they should. The disease is called “chronic” because the damage to your kidneys happens slowly over a long period of time. Some of the common health problems caused by kidney disease include gout, anemia, bone disease, heart disease and fluid buildup.

Where is the issue occurring?

The kidneys grow larger and gradually lose the ability to function as they should. Chronic kidney disease occurs when a disease or condition impairs kidney function, causing kidney damage and high blood pressure.

Why is it important that we fix the problem?

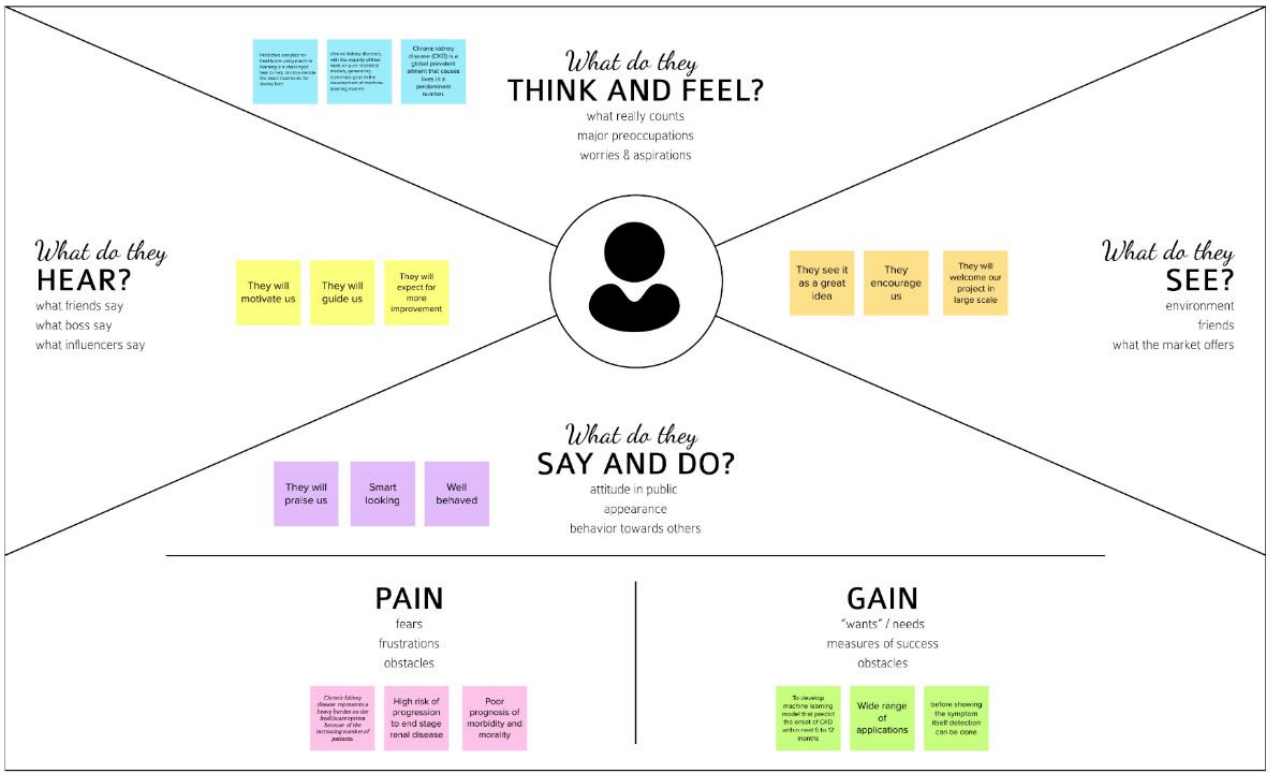
The older you get the more likely you are to have some degree of kidney disease.

This is important because CKD increases the risk of heart attack and stroke.

3. IDEATION & PROPOSED SOLUTION

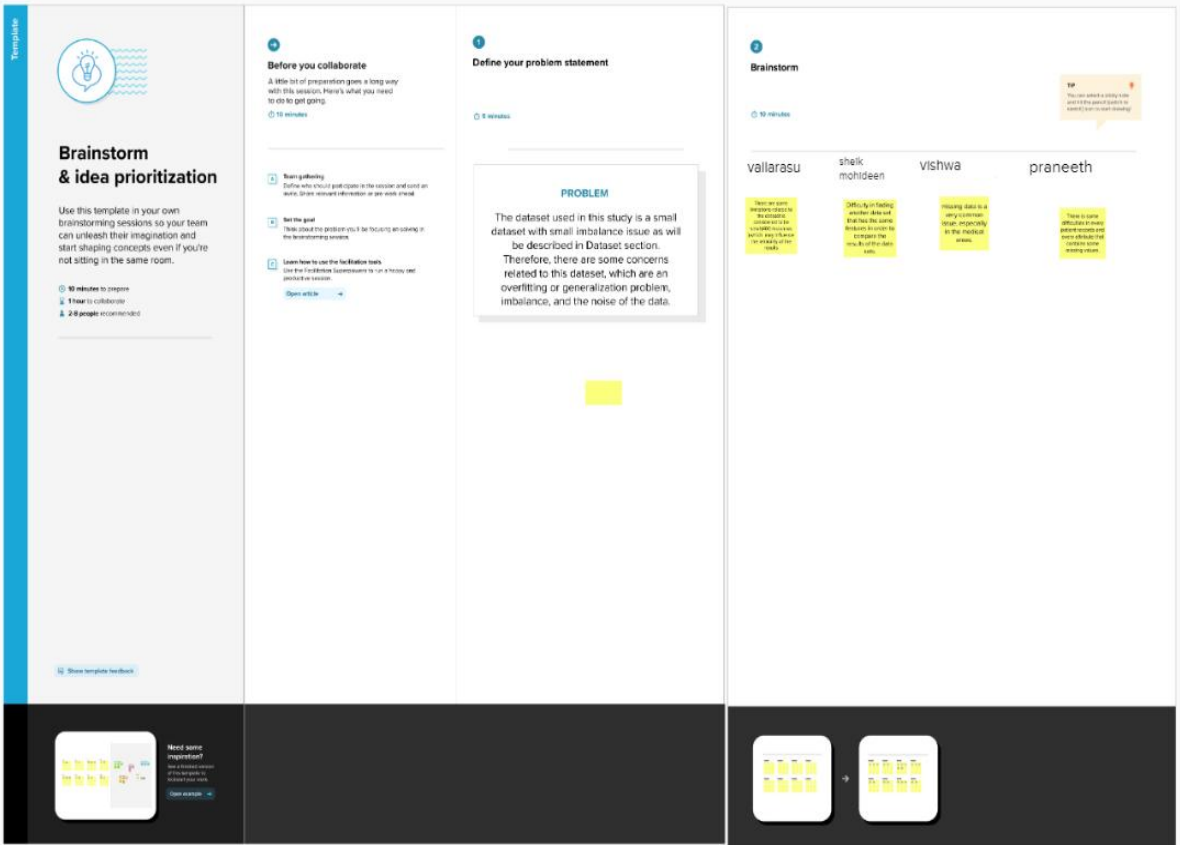
Empathy Map Canvas

Early Detection of Chronic Kidney Disease using Machine Learning



Ideation and Brainstorming:

Early Detection of Chronic Kidney Disease Using Machine Learning :



Proposed Solution

Problem Statement

Chronic Kidney Disease is a major concern for the global health care system. Chronic Kidney Disease is now wreaking havoc on society and is spreading at an alarming rate. Various efforts have been undertaken to advance early therapy to prevent the condition from progressing to CKD. Recent research suggest that some of the negative outcomes can be avoided with early identification and treatment.

Feasibility of idea

To predict the early onset of CKD, three Machine Learning techniques are used: ->Random Forest, Decision Tree, Support Vector Machines. Using these techniques, each algorithm's effectiveness is evaluated and the prediction of how many people have been affected by CKD is identified.

Novelty

The renal patient is recognized by undertaking two primary tests.

->A Blood Test to determine Glomerular FilterationRate(GFR).

->A Urine Test to determine Albumin.

Social Impact

As people don't undergo the general test of their health, early detection of CKD is not identified. This creates a great social impact of not being aware of CKD. As a result of this many people are getting affected by CKD.

Business Model

The widespread use of Machine Learning of predicting the CKD in the Medical Industries promotes medical innovation, lowers medical expenses, and improves medical quality. In order to cure the CKD patients, the hospitals have been gaining business profit in recovering the patients.

Scalability of solution

This Chronic Kidney Disease have been spreading widely now a days. Early prediction of CKD using Machine Learning that is more efficient to analyze the disease so that it can be cured on time.

Problem Solution Fit

1. CUSTOMER SEGMENT(S)

People who have symptoms of CKD and people who want to get their kidneys checked for staying healthy and fit.

Doctors and workers in the healthcare sectors.

6. CUSTOMER CONSTRAINTS

The test and treatments for advanced stages are highly priced.

The patients have to wait for a longer time to get their test results which may cause more complications in the meantime.

5. AVAILABLE SOLUTIONS

Proper medications

Regime diets

Dialysis.

2. JOBS-TO-BE-DONE / PROBLEMS:

Create an interface that is convenient for the users to operate and easy to understand.

Ensure that the predictions are of high accuracy.

Provide accurate and faster results by using the given data set so that the kidney disease can be detected earlier.

9. PROBLEM ROOT CAUSE

Ignorance of symptoms and lack of awareness about the disease.

Unhealthy diet.

Not having enough water.

Consumption of alcohol or tobacco frequently.

7. BEHAVIOUR

Notice the changes in your body and lookout for the symptoms of CKD.

Consult the doctors if you notice CKD symptoms.

Develop awareness and be cautious about the prevent disease. do not ignore the symptoms or be careless about those.

3. TRIGGERS

Usually, CKD tests takes longer time to predict the disease. These tests are expensive and are not easily affordable by many people.

4. EMOTIONS: BEFORE / AFTER

People are comfused and worried before taking up the prediction test. As they don't know why there are having the symptoms, they don't get proper treatment until they find out that they symptoms are leading to CKD. Without proper medications, the chances of complications are high.

After detecting the disease, patients gain more clarity and can be directed towards getting the proper medication. Hence, they can be more hopeful.

10. YOUR SOLUTION

Building a machine learning model that will predict CKD in its early stages by providing accurate and faster results.

8. CHANNELS OF BEHAVIOUR

ONLINE:

Consult a specialist online to follow the necessary medications.

Browse about the disease to be cautious.

OFFLINE:

Get dialysis and kidney transplations for advanced stage and medications for initial stages.

4. REQUIREMENTS ANALYSIS

Functional Requirements

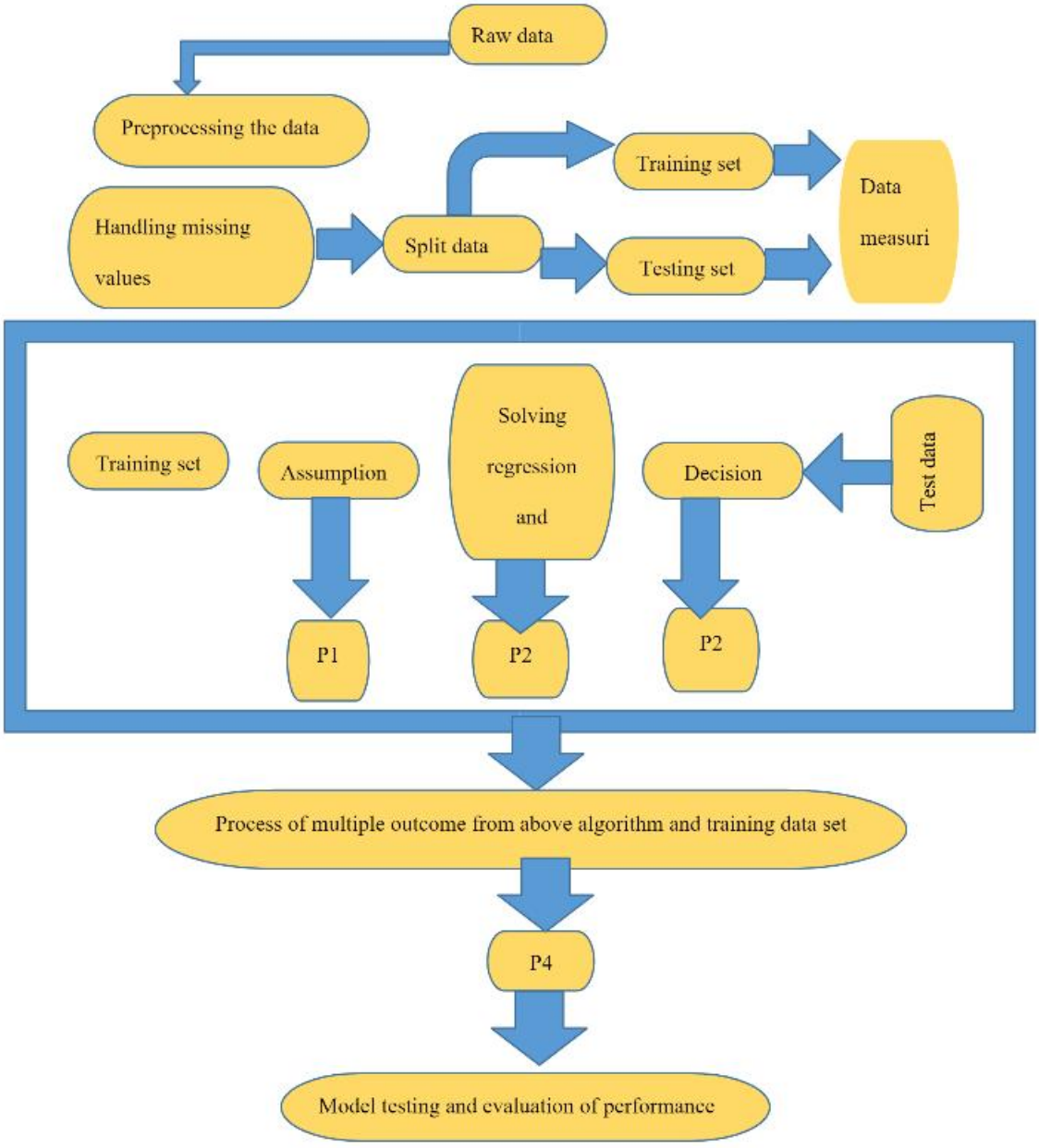
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR 1	User Page	Users can go to home pag
FR 2	Details	Users can know more details about the CKD.
FR 3	User Requirements	Store past records Generate report for presence of CKD Diagnostic remedies for symptoms
FR 4	User Entry	Input form for pre-diagnostic test results
FR 5	Business Requirements	Quick diagnosis for CKD
FR 6	User Feedback	Allows users to submit feedback

Non- Functional Requirements

FR NO.	**Non-Functional Requirement**	**Description**
NFR - 1	Usability	Simple user-friendly interface for communication
NFR - 2	Security	Safeguard the details shared by users and maintain confidentiality
NFR - 3	Reliability	Diagnosis based on probability predicted by ML model must be reliable
NFR - 4	Performance	Reduction in overall time taken for diagnosis
NFR - 5	Availability	Available at any time to users from various places
NFR - 6	Scalability	Needs to support numerous users at once

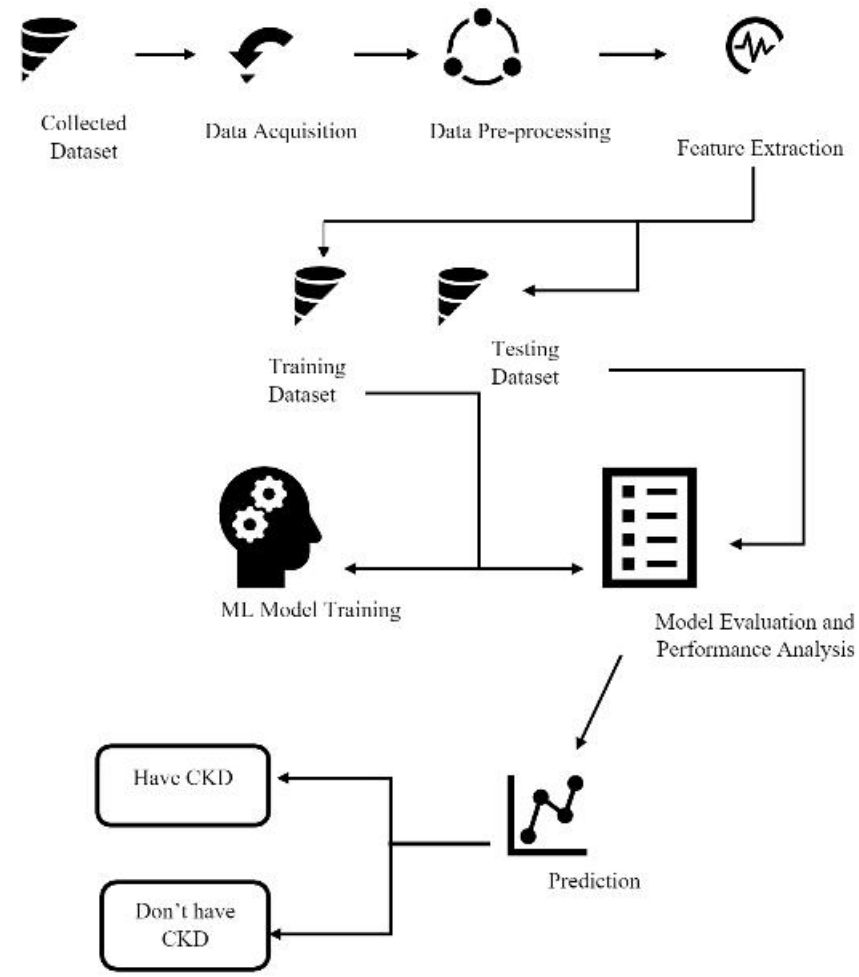
5. PROJECT DESIGN

Data Flow Diagrams



Solution & Technical Architecture

Technical Architecture:



User Stories

Create Home Page

As a user, I can collect the database of patients.

Preprocessing

As a user, I can preprocess the data from the database.

Testing

As a user, I can make the patients undergo two kinds of test (urine and blood test)

Analysis

As a user, I can analyse the test results

Prediction

As a user, I can predict CKD with the test results

Recovering Process

As a user, I can make the affected patients to take treatment

Awareness

As a user, I can advise the patients to maintain a healthy diet plan.

6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Collection of Dataset	USN-1	Collect dataset and clean the dataset	5	High	Praneeth kumar Vallarasu
Sprint-1		USN-2	Create, test and save the model	5	High	Vishwa Sheik mohideen
Sprint-2	Home page	USN-3	The user can enter into the home page	6	High	Praneeth kumar Vallarasu
Sprint-2		USN-4	User can use the prediction button to enter into the prediction page	4	Medium	Vishwa Sheik mohideen
Sprint-3	Prediction page	USN-5	The user will be presented with the prediction page where he can enter the values of report	3	Medium	Praneeth kumar Vallarasu
Sprint-3		USN-6	User should enter the blood glucose parameters	7	High	Vishwa Sheik mohideen
Sprint-4	Result	USN-7	The user will get the output	4	Medium	Praneeth kumar Vallarasu
Sprint-4		USN-8	Deploy into IBM CLOUD	6	High	Vishwa Sheik mohideen

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	6Days	24Oct2022	29Oct2022	20	29Oct2022
Sprint-2	20	6Days	31Oct2022	05Nov2022		
Sprint-3	20	6Days	07Nov2022	12Nov2022		

Sprint-4	20	6Days	14Nov2022	19Nov2022		

7. CODING AND SOLUTIONING

Feature 1:

In our website, we have additionally created some more pages:

Home Page.html

```
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>chronic</title>
    <link rel="stylesheet" href="/static/style.css" />
  </head>
  <body bgcolor="black">
      

    <div class="d1" align="right">
      <a href="/"> HOME </a>
         
      <a href="/index">PREDICTION</a>
       
    </div>

    <div class="glow" align="center">
      <h2>CHRONIC KIDNEY DISEASE PREDICTION</h2>
    </div>
  </body>
</html>
```

Index.html

```
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>pridiction</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="/static/indexStyle.css" />
```

```
  <script>
    function check() {
      document.getElementById("prediction").innerHTML = "";
    }
  </script>
</head>
<body>
    
  <div class="d1" align="right">
    <a href="/"> HOME </a>
       
    <a href="index.html">PREDICTION</a>
     
  </div>

  <h2>Enter the mentioned values</h2>

  <form class="container" action="/predict" method="post">
    <div class="tab" align="center">
      <table>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr>
          <td>
            <label for="age">Age : </label>
          </td>
          <td>
            <input id="age" type="number" name="age" required />
          </td>
        </tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr>
          <td>
            <label for="blood_Urea">Blood Urea : </label>
          </td>
          <td>
            <input id="blood_Urea" type="number" name="blood_Urea" required />
          </td>
        </tr>
```

```
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label for="BGR">Blood Glucose Random : </label>
  </td>
  <td>
    <input id="BGR" type="number" name="BGR" required />
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Are you Affected by Coronary Artery Disease : </label>
  </td>
  <td>
    <select name="CRD" id="CRD">
      <option for="CRD" value="1">YES</option>
      <option for="CRD" value="0">NO</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Do you have Anemia : </label>
  </td>
  <td>
    <select name="anemia" id="anemia">
      <option for="anemia" value="1">YES</option>
      <option for="anemia" value="0">NO</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
```

```
<tr></tr>
<tr>
  <td>
    <label>Pus cell : </label>
  </td>
  <td>
    <select name="pus_cell" id="pus_cell">
      <option for="pus_cell" value="1">NORMAL</option>
      <option for="pus_cell" value="0">ABNORMAL</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Red Blood Cell : </label>
  </td>
  <td>
    <select name="RBC" id="RBC">
      <option for="RBC" value="1">NORMAL</option>
      <option for="RBC" value="0">ABNORMAL</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Diabetesmellitus : </label>
  </td>
  <td>
    <select name="Diabete" id="Diabete">
      <option for="Diabete" value="1">YES</option>
      <option for="Diabete" value="0">NO</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
```

```
          <tr></tr>
          <tr>
            <td>
              <label>Pedal Edema : </label>
            </td>
            <td>
              <select name="P_edema" id="P_edema">
                <option for="P_edema" value="1">YES</option>
                <option for="P_edema" value="0">NO</option>
              </select>
            </td>
          </tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr>
            <td>
              <input type="submit" value="submit" />
            </td>
            <td>
              <input type="reset" value="clear" onclick="check()" />
            </td>
          </tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
        </table>
      </div>
    </form>

    <div align="center" class="hidden">
      <h2 id="prediction">{{pred}}</h2>
    </div>
  </body>
</html>
```

Indexstyle.css

```
* {
  margin: 0px;
  padding: 0px;
}
```

```
body {
  background-image: linear-gradient(
      rgba(206, 61, 174, 0.527),
      rgba(202, 78, 78, 0.5),
      rgba(67, 153, 173, 0.8)
    ),
    url("index_image.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  font-display: #1dbfca;
  user-select: none;
}
h2 {
  font-family: "Damion", cursive;
  font-weight: 400;
  color: black;
  font-size: 35px;
  text-align: center;
  position: flex;
}
.container {
  text-align: center;
  position: flex;
}
input {
  padding: 5px;
  background-color: transparent;
  border-color: rgb(11, 116, 116);
  font-size: 20px;
  color: white;
  display: flex;
  outline: none;
  border: 1px solid black;
  border-radius: 10px;
  text-align: center;
}

label {
  font-family: "Damion", cursive;
  font-weight: 200;
  color: black;

  font-size: 25px;
  text-align: left;
```

```
}

@media screen {
  label,
  .dl {
    position: flex;
  }
}
.tab {
  border-radius: 50px 0px 50px 0px;
  align-items: center;
  margin-left: 15%;
  padding: 10px;
  margin-right: 15%;
  background: linear-gradient(
    rgb(168, 69, 131),
    rgb(58, 58, 193),
    rgb(43, 150, 164)
  );
}
input[type="submit"],
input[type="reset"] {
  margin-top: 20px;
  padding: 10px;
  outline: none;
  border-radius: 10px;
  font-size: 14px;
  text-transform: uppercase;
}

input[type="submit"]:hover,
input[type="reset"]:hover {
  background-color: white;
  transition: 0.8s ease-in-out;
  color: black;
}
```

Style.css

```
    * {
    margin: 0;
    padding: 0;
}
.glow {
    font-family: cursive;
    text-align: center;
```

```
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);

    color: rgb(218, 218, 233);
    animation: glow 1s ease-in-out infinite alternate;
}
@keyframes glow {
    from {
        text-shadow: 0 0 10px rgb(20, 19, 19), 0 0 20px rgb(15, 2, 2),
        0 0 30px #090111, 0 0 40px #f3d9e6,
        0 0 50px #e60073, 0 0 60px #e60073,
        ;
    }
    to {
        text-shadow: 0 0 20px rgb(163, 85, 85),0 0 30px #92667c,
        0 0 40px #ff4da6;
    }
}
.glow h2 {
    font-size: 7vmin;
}
a {
    font-size: 3vmin;
    color: whitesmoke;
    text-decoration: dashed;
    font-family: 'Lato',sans-serif;
    padding: 2px 5px;


}
a:hover {

    color: #090111;
    text-shadow:  0 0 20px rgb(240, 233, 233),0 0 30px #f7f1f4,
    0 0 90px #ff4da6;

}
```

8. TESTING

Test Cases:

Test Case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps to execute	Expected Result	Actual Result	Status
Home_TC_001	UI	Home Page	When user clicks on my website, the user can see the home page I've created and it is verified successfully.	HTML , CSS and Flask for integration	1.Create HTML files 2. Run using python	Result was not as expected	Actual result was good	Pass
Prediction_TC_002	UI	Prediction Page	Verify the input values valid or not	Flask Server, Python, anaconda prompt	1.Create HTML files 2. Run using Flask	Working as expected	Satisfied as expected	Pass
Result_TC_003	UI	Result Page	Verify whether the user is able to predict	HTML,CSS, Visual Code	1.Create HTML files 2. Run using Flask			Pass

User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Early Detection of Chronic Kidney Disease] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	1	1	1	6
Duplicate	4	0	2	0	6
External	2	2	0	1	5
Fixed	1	1	1	1	4
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	10	4	4	3	21

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Home Screen	1	0	0	1
User Input	3	0	0	3
Chronic Kidney Disease testing	2	0	0	2
No Chronic Kidney Disease testing	2	0	0	2
Version Control	2	0	0	2

9. RESULTS:

Performance metrics

Category (# of Measures)	Subcategories	Measure Validity Rating — High Medium Low
CKD Prevention (7)	Hypertension, Diabetes	2 5 0
Slowing CKD Progression (2)	Hypertension/CKD	2 0
CKD Management (2)	Advance Care Planning, Lipid Testing	1 10
Advanced CKD/Kidney Replacement (1)	Dialysis Access	10
Dialysis Management (28)	Dialysis Access, Adequacy, Anemia, ESRD-related Complications, Transplant Referral, Advance Care Planning, Care Coordination	17 7 4
Broad Measures (18)	Preventive Care, Medication Reconciliation and Safety, Advance Care Planning, Falls, Complications/Misc.	6 8 4
PROMs (2)	PROMs	0 2 0

10. ADVANTAGES AND DISADVANTANGES

Advantages:

- Resulted in good accuracy.
- Customers can easily predict CKD.
- This machine learning model helps us in analysing the CKD.

Disadvantages:

- At some point algorithm may fail.
- Correct prediction may not be done.

11. CONCLUSION

This work examines the ability to detect CKD using machine learning algorithms while considering the least number of tests or features. We approach this aim by applying machine learning classifier logistic regression on a big dataset of around 4000 records. We found that haemoglobin, albumin, and specific gravity have the most impact to predict the CKD.

We conclude that by using Machine Learning algorithm , we can predict whether a person is suffering from CKD or not. This effective supervised learning algorithm helps us to analyze the datasets and helps us conclude about the disease. Thus, this project helps customers to easily predict the CKD by giving their data

12. FUTURE SCOPE

This work will be considered as basement for the healthcare system for CKD

patients. Also extension to this work is that implementation of deep learning since deep learning provides high-quality performance than machine learning algorithm. In order to help in reducing the prevalence of CKD, we plan to predict if a person with CKD risk factors such as diabetes, hypertension, and family history of kidney failure will have CKD in the future or not by using appropriate dataset.

1. APPENDIX

Source code:

HTML CODING:

Home.html

```
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>chronic</title>
    <link rel="stylesheet" href="/static/style.css" />
  </head>
  <body bgcolor="black">
      

    <div class="d1" align="right">
      <a href="/"> HOME </a>
         
      <a href="/index">PREDICTION</a>
       
    </div>

    <div class="glow" align="center">
      <h2>CHRONIC KIDNEY DISEASE PREDICTION</h2>
    </div>
  </body>
</html>
```

Index.html

```
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>pridiction</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="/static/indexStyle.css" />
    <script>
      function check() {
        document.getElementById("prediction").innerHTML = "";
      }
    </script>
  </head>
  <body>
      
    <div class="d1" align="right">
      <a href="/"> HOME </a>
         
      <a href="index.html">PREDICTION</a>
       
    </div>

    <h2>Enter the mentioned values</h2>

    <form class="container" action="/predict" method="post">
      <div class="tab" align="center">
        <table>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr>
            <td>
              <label for="age">Age : </label>
            </td>
            <td>
              <input id="age" type="number" name="age" required />
            </td>
          </tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr></tr>
          <tr>
            <td>
```

```
    <label for="blood_Urea">Blood Urea : </label>
  </td>
  <td>
    <input id="blood_Urea" type="number" name="blood_Urea" required />
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label for="BGR">Blood Glucose Random : </label>
  </td>
  <td>
    <input id="BGR" type="number" name="BGR" required />
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Are you Affected by Coronary Artery Disease : </label>
  </td>
  <td>
    <select name="CRD" id="CRD">
      <option for="CRD" value="1">YES</option>
      <option for="CRD" value="0">NO</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Do you have Anemia : </label>
  </td>
  <td>
    <select name="anemia" id="anemia">
      <option for="anemia" value="1">YES</option>
      <option for="anemia" value="0">NO</option>
```

```
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Pus cell : </label>
  </td>
  <td>
    <select name="pus_cell" id="pus_cell">
      <option for="pus_cell" value="1">NORMAL</option>
      <option for="pus_cell" value="0">ABNORMAL</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Red Blood Cell : </label>
  </td>
  <td>
    <select name="RBC" id="RBC">
      <option for="RBC" value="1">NORMAL</option>
      <option for="RBC" value="0">ABNORMAL</option>
    </select>
  </td>
</tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr></tr>
<tr>
  <td>
    <label>Diabetesmellitus : </label>
  </td>
  <td>
    <select name="Diabete" id="Diabete">
      <option for="Diabete" value="1">YES</option>
      <option for="Diabete" value="0">NO</option>
```

```
            </select>
          </td>
        </tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr>
          <td>
            <label>Pedal Edema : </label>
          </td>
          <td>
            <select name="P_edema" id="P_edema">
              <option for="P_edema" value="1">YES</option>
              <option for="P_edema" value="0">NO</option>
            </select>
          </td>
        </tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr>
          <td>
            <input type="submit" value="submit" />
          </td>
          <td>
            <input type="reset" value="clear" onclick="check()" />
          </td>
        </tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
        <tr></tr>
      </table>
    </div>
  </form>

  <div align="center" class="hidden">
    <h2 id="prediction">{{pred}}</h2>
  </div>
 </body>
</html>
```

CSS CODING:

Indexstyle.css

```
* {
  margin: 0px;
  padding: 0px;
}
body {
  background-image: linear-gradient(
      rgba(206, 61, 174, 0.527),
      rgba(202, 78, 78, 0.5),
      rgba(67, 153, 173, 0.8)
    ),
    url("index_image.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  font-display: #1dbfca;
  user-select: none;
}
h2 {
  font-family: "Damion", cursive;
  font-weight: 400;
  color: black;
  font-size: 35px;
  text-align: center;
  position: flex;
}
.container {
  text-align: center;
  position: flex;
}
input {
  padding: 5px;
  background-color: transparent;
  border-color: rgb(11, 116, 116);
  font-size: 20px;
  color: white;
  display: flex;
  outline: none;
  border: 1px solid black;
  border-radius: 10px;
  text-align: center;
}
```

```
label {
  font-family: "Damion", cursive;
  font-weight: 200;
  color: black;

  font-size: 25px;
  text-align: left;
}

@media screen {
  label,
  .d1 {
    position: flex;
  }
}
.tab {
  border-radius: 50px 0px 50px 0px;
  align-items: center;
  margin-left: 15%;
  padding: 10px;
  margin-right: 15%;
  background: linear-gradient(
    rgb(168, 69, 131),
    rgb(58, 58, 193),
    rgb(43, 150, 164)
  );
}
input[type="submit"],
input[type="reset"] {
  margin-top: 20px;
  padding: 10px;
  outline: none;
  border-radius: 10px;
  font-size: 14px;
  text-transform: uppercase;
}

input[type="submit"]:hover,
input[type="reset"]:hover {
  background-color: white;
  transition: 0.8s ease-in-out;
  color: black;
}
```

Style.css

```
* {
    margin: 0;
    padding: 0;
}
.glow {
    font-family: cursive;
    text-align: center;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);

    color: rgb(218, 218, 233);
    animation: glow 1s ease-in-out infinite alternate;
}
@keyframes glow {
    from {
        text-shadow: 0 0 10px rgb(20, 19, 19), 0 0 20px rgb(15, 2, 2),
        0 0 30px #090111, 0 0 40px #f3d9e6,
        0 0 50px #e60073, 0 0 60px #e60073,
        ;
    }
    to {
        text-shadow: 0 0 20px rgb(163, 85, 85),0 0 30px #92667c,
        0 0 40px #ff4da6;
    }
}
.glow h2 {
    font-size: 7vmin;
}
a {
    font-size: 3vmin;
    color: whitesmoke;
    text-decoration: dashed;
    font-family: 'Lato',sans-serif;
    padding: 2px 5px;


}
a:hover {
```

```
    color: #090111;
    text-shadow:  0 0 20px rgb(240, 233, 233),0 0 30px #f7f1f4,
    0 0 90px #ff4da6;

}
```

Flask Integrate With Scoring End Pints:

```
* {
    margin: 0;
    padding: 0;
}
.glow {
    font-family: cursive;
    text-align: center;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);

    color: rgb(218, 218, 233);
    animation: glow 1s ease-in-out infinite alternate;
}
@keyframes glow {
    from {
        text-shadow: 0 0 10px rgb(20, 19, 19), 0 0 20px rgb(15, 2, 2),
        0 0 30px #090111, 0 0 40px #f3d9e6,
        0 0 50px #e60073, 0 0 60px #e60073,
        ;
    }
    to {
        text-shadow: 0 0 20px rgb(163, 85, 85),0 0 30px #92667c,
        0 0 40px #ff4da6;
    }
}
.glow h2 {
    font-size: 7vmin;
}
a {
    font-size: 3vmin;
    color: whitesmoke;
    text-decoration: dashed;
    font-family: 'Lato',sans-serif;
    padding: 2px 5px;
```

```
}
a:hover {

    color: #090111;
    text-shadow:  0 0 20px rgb(240, 233, 233),0 0 30px #f7f1f4,
    0 0 90px #ff4da6;

}
```

NOTEBOOK .ipynb

Import Dataframe:

In [87]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='GenImWOC7pmDNtYSNujVb89Ra_uBfV7HW1eYmqCch1cr',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'ckd-donotdelete-pr-llwzpw51re9nxp'
object_key = 'chronickidneydisease.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )
```

Import Libraries:

In [131]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from collections import Counter
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import joblib
```

Load The Dataset:

LOAD DATASET

```
In [15]: df=pd.read_csv("chronickidneydisease.csv")
         pd.set_option("display.max_column",None)
         df
```

```
In [38]: df.shape
Out[38]: (400, 27)

In [39]: df.info
```

```
In [42]: df.sod.unique()
Out[42]: array([139. , 136. , 140. , 111. , 142. , 104. , 147. , 131. , 114. ,
                124. , 138. , 135. , 130. , 141. , 145. ,   4.5, 128. , 129. ,
                144. , 132. , 133. , 150. , 134. , 125. , 137. , 163. , 143. ,
                127. , 146. , 126. , 122. , 115. , 113. , 120. ])

In [43]: df.pcc.value_counts()
Out[43]: notpresent    354
         present        42
         Name: pcc, dtype: int64

In [18]: print(numerical_feature)
         ['id', 'age', 'bp', 'sg', 'al', 'su', 'bgr', 'bu', 'sc', 'sod', 'pot', 'hemo']

In [104]: df.isnull().any()
Out[104]: id               False
```

Visualization:

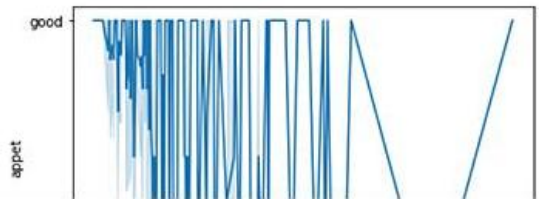
VISUALIZATION

```
In [44]: sns.displot(df.age)
Out[44]: <seaborn.axisgrid.FacetGrid at 0x28be2b13d30>
```

```
In [45]: sns.lineplot(df.bu,df.appet)
```

```
C:\Users\ELCOT\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the followi
ng variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `da
ta`, and passing other arguments without an explicit keyword will result in an error or misinterpreta
tion.
  warnings.warn(
```

```
Out[45]: <AxesSubplot:xlabel='bu', ylabel='appet'>
```




```
In [48]: sns.scatterplot(df.sg,df.bp)
```

```
C:\Users\ELCOT\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the followi
ng variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `da
ta`, and passing other arguments without an explicit keyword will result in an error or misinterpreta
tion.
  warnings.warn(
```

```
Out[48]: <AxesSubplot:xlabel='sg', ylabel='bp'>
```

```
In [55]: df.hist(figsize=(8,8))
```

```
Out[55]: array([[<AxesSubplot:title={'center':'id'}>,
                 <AxesSubplot:title={'center':'age'}>
```

Outlier Detection:

OUTLIER DETECTION ¶

```
In [57]: sns.boxplot(df.pot)
```

```
C:\Users\ELCOT\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the followi
ng variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data
`, and passing other arguments without an explicit keyword will result in an error or misinterpretati
on.
  warnings.warn(
```

```
Out[57]: <AxesSubplot:xlabel='pot'>
```

Outlier Using IQR:

OUTLIER REMOVAL USING IQR

```
In [58]: q1=df.hemo.quantile(0.25)  #(Q1)
         q3=df.hemo.quantile(0.75)  #(Q3)

In [65]: IQR=q3-q1

In [66]: upper_limit= q3 + 1.5*IQR

In [68]: uppper_limit= q3 + 1.5*IQR
         lower_limit=q1 - 1.5*IQR

In [69]: sns.boxplot(df.bu)
```

Outlier removal with percentile:

OUTLIER REMOVAL WITH PERCENTILE

```
In [71]: p99=df.hemo.quantile(0.99)
         p99
```

Z-Score:

Z-SCORE

```
In [73]: from scipy import stats

In [76]: hemo_zscore=stats.zscore(df.hemo)

In [77]: hemo_zscore
```

Encoding Techniques:

ENCODING TECHNIQUES

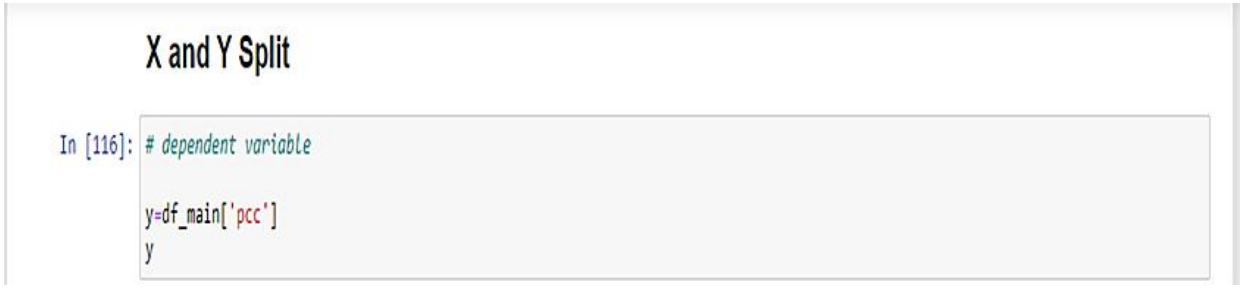
1. One hot encoding

```
In [78]: from sklearn.preprocessing import LabelEncoder

In [79]: le=LabelEncoder()

In [80]: df.age=le.fit_transform(df.age)
         df.bp=le.fit_transform(df.bp)
```

2. One hot encoding

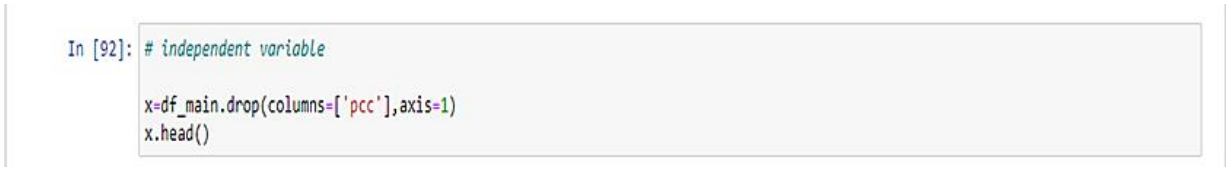
```
In [83]: df_main=pd.get_dummies(df,columns=['hemo'])
         df_main.head()
```

X and Y Split:



X and Y Split

```
In [116]: # dependent variable

          y=df_main['pcc']
          y
```



```
In [92]: # independent variable

         x=df_main.drop(columns=['pcc'],axis=1)
         x.head()
```

Linear Regression:

LINEAR REGRESSION

1.Simple Linear Regression

In [120]:
```
sns.displot(df.wc)
```

```
C:\Users\ELCOT\anaconda3\lib\site-packages\seaborn\axisgrid.py:88: UserWarning: Glyph 9 (       ) missing from current font.
  self._figure.tight_layout(*args, **kwargs)
```

In [141]:
```
x=df.drop(columns=['hemo'],axis=1)
x
```

Multi Linear Regression:

2. Multi Linear Regression

In [126]:
```
from sklearn.linear_model import LinearRegression
```

In [151]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [153]:
```
df.State=le.fit_transform(df.hemo)
df
```

In [155]:
```
df.corr().age.sort_values(ascending=False)
```

X ana Y split

```
In [157]: X=df.drop(columns=['age'],axis=1)
          X.head()  # independent variables
```

```
In [158]: y= df.age
          y.head()
```

Polynomial Regression:

Polynomial Regression

```
In [163]: x=df.iloc[:,1:2]
          x
```

```
In [166]: from sklearn.linear_model import LinearRegression
          lr=LinearRegression()
```

```
In [167]: lr.fit(xp,y)
```


Convert Normal Feature to Polynomial feature

```
In [164]: from sklearn.preprocessing import PolynomialFeatures
          pr=PolynomialFeatures(degree=4)
```

```
In [165]: xp=pr.fit_transform(x)
          xp
```

```
In [168]: # checking the prediction on a random value

          id=lr.predict(pr.fit_transform([[6]]))
          id
```

Logistic Regression:

LOGISTIC REGRESSION

```
In [170]: from sklearn.preprocessing import LabelEncoder

          le=LabelEncoder()

          df.age=le.fit_transform(df.age)

          df.head()
```

```
In [171]: df.id.value_counts()
```

```
In [237]: sns.displot(df.age)
```

Model Building:

```
In [243]: from sklearn.preprocessing import MinMaxScaler
          scale = MinMaxScaler()
```

Model Building

```
In [242]: ## Logistic Regression

          from sklearn.linear_model import LogisticRegression

In [244]: model = LogisticRegression()

In [245]: model.fit(x_train,y_train)
```

Evaluating Model:

Evaluating the Model

In [182]:
```
# accuracy score

from sklearn.metrics import accuracy_score,classification_report,confusion_matrix,roc_auc_score,roc_curve
```

In [247]:
```
accuracy_score(y_test,pred_test) # test accuracy
```

Out[247]: 0.06060606060606061

In [248]:
```
# Confusion matrix

pd.crosstab(y_test,pred_test)
```

In [249]:
```
# Classification report

print(classification_report(y_test,pred_test))
```

In [186]:
```
# Precision

#  TP/(TP+FP)

24/30
```

In [187]:
```
# Recall

# TP /(TP+FN)

24/32
```

In [188]:
```
# F1 score

# 2*precision*Recall / (precision+Recall)

2*0.8*0.75/(0.8+0.75)
```

In [229]:
```
# performs the split

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y)
```

In [223]:
```
# Display the shape

print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(297, 1)
(99, 1)
(297,)
(99,)
```

In [231]:
```
x_train.shape
```

Out[231]: (297, 1)

In [232]:
```
x_test.shape
```

Out[232]: (99, 1)

In [252]:
```
# ROC - AUC Score

probability = model.predict_proba(x_test)[:,1]
probability
```


Ridge and Lasso Regression:

RIDGE AND LASSO REGRESSION

In [256]:
```
# l1 Regularization technique- Lasso Regression
# l2 Regularization technique- Ridge Regression
```

In [259]:
```
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
```

In [260]:
```
r=Ridge()
l=Lasso()
```

In [261]:
```
r.fit(x_train,y_train)
```

Out[261]: Ridge()

In [262]:
```
l.fit(x_train,y_train)
```

Out[262]: Lasso()

In [265]:
```
pred1=r.predict(x_test)
pred1
```

In [266]:
```
pred2=l.predict(x_test)
pred2
```

Evaluation Metrics for Regression:

1.Evaluation metrics for Regression problem

In [267]:
```
# R-Square
# testing accuracy for both model

print(metrics.r2_score(y_test,pred1))
print(metrics.r2_score(y_test,pred2))
```

```
0.9999999998368482
0.9999856081783125
```

In [268]:
```
profit=pd.DataFrame({'Actual':y_test,'ridge_pred':pred1,'lasso_pred':pred2})
profit.head(10)
```

In [269]:
```
## MSE(Mean square error)

print(metrics.mean_squared_error(y_test,pred1))
print(metrics.mean_squared_error(y_test,pred2))
```

```
3.7522833790478166e-08
0.0033099339693025715
```

Decision Tree Classifier:

DECISION TREE CLASSIFIER

In [270]:
```
# X and y split

X=df.iloc[:,:-1]
X.head()
```

1. Model Building

```
In [277]: from sklearn.tree import DecisionTreeClassifier

          model = DecisionTreeClassifier(max_depth=4,splitter='best',criterion='entropy')

In [278]: model.fit(x_train,y_train)

Out[278]: DecisionTreeClassifier(criterion='entropy', max_depth=4)

In [279]: y_predict= model.predict(x_test)
          y_predict
```

Random Forest Classifier:

RANDOM FOREST CLASSIFIER

1. Model Building

```
In [282]: from sklearn.ensemble import RandomForestClassifier
          model = RandomForestClassifier(n_estimators=10,criterion='entropy')

In [283]: model.fit(x_train,y_train)

Out[283]: RandomForestClassifier(criterion='entropy', n_estimators=10)

In [284]: y_predict = model.predict(x_test)

In [285]: y_predict_train = model.predict(x_train)

In [286]: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

In [287]: print('Training accuracy: ',accuracy_score(y_train,y_predict_train))
          print('Testing accuracy: ',accuracy_score(y_test,y_predict))

          Training accuracy:  0.98989898989899
          Testing accuracy:  0.9797979797979798
```


Naive Baye's Classifier:

NAIVE BAYE'S CLASSIFIER

```
In [288]: # using for classification problem.
          # Naive baye's based on Baye's theorem fro the classification.

In [289]: from sklearn.preprocessing import MinMaxScaler
          scale = MinMaxScaler()

In [394]: x_scaled = pd.DataFrame(scale.fit_transform(x),columns= x.columns)
          x_scaled.head()
```

1. Model Building

In [294]:
```
# model building
from sklearn.naive_bayes import GaussianNB
model =GaussianNB()
```

In [295]:
```
model.fit(x_train,y_train)
```

Out[295]:
```
GaussianNB()
```

In [296]:
```
# evaluating the model
y_pred = model.predict(x_test)
```

In [297]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
```

In [298]:
```
accuracy_score(y_test,y_pred)
```

Out[298]:
```
0.9797979797979798
```

In [299]:
```
print(classification_report(y_test,y_pred))
```

Unsupervised K-Means:

UNSUPERVISED K-MEANS

In [300]:
```
from sklearn import cluster
```

In [332]:
```
error =[]
for i in range(11,1):
    kmeans=cluster.KMeans(n_clusters=i,init="k-means++",random_state=3)
    kmeans.fit(new_df)
    error.append(kmeans.inertia_)
```

In [333]:
```
error
```

Out[333]:
```
[]
```

In [334]:
```
import matplotlib.pyplot as plt
plt.plot(range(11,1),error)
plt.title('Elbow method')
plt.xlabel('no of clus')
plt.ylabel('error')
plt.show()
```

Model Building

```
In [340]: # model building
          from sklearn.neighbors import KNeighborsClassifier
          model = KNeighborsClassifier()

In [341]: model.fit(x_train,y_train)

Out[341]: KNeighborsClassifier()

In [395]: y_pred=model.predict(x_test)

In [396]: y_pred1=model.predict(x_train)
```

Model Evaluation:

Model Evaluation

```
In [345]: # model evaluation
          from sklearn.metrics import accuracy_score,confusion_matrix,classification_report

In [385]: # testing with a random observation

          model.predict([[1.1]])
```

IBM CODING:

```
* {
    margin: 0;
    padding: 0;
}
.glow {
    font-family: cursive;
    text-align: center;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);

    color: rgb(218, 218, 233);
    animation: glow 1s ease-in-out infinite alternate;
}
@keyframes glow {
    from {
        text-shadow: 0 0 10px rgb(20, 19, 19), 0 0 20px rgb(15, 2, 2),
        0 0 30px #090111, 0 0 40px #f3d9e6,
        0 0 50px #e60073, 0 0 60px #e60073,
```

```
        ;
    }
    to {
        text-shadow: 0 0 20px rgb(163, 85, 85),0 0 30px #92667c,
        0 0 40px #ff4da6;
    }
}
.glow h2 {
    font-size: 7vmin;
}
a {
    font-size: 3vmin;
    color: whitesmoke;
    text-decoration: dashed;
    font-family: 'Lato',sans-serif;
    padding: 2px 5px;


}
a:hover {

    color: #090111;
    text-shadow:  0 0 20px rgb(240, 233, 233),0 0 30px #f7f1f4,
    0 0 90px #ff4da6;

}
```

IBM DEPLOYMENT:

Building a Machine Learning model

```
In [34]: from sklearn.linear_model import LogisticRegression
         lgr=LogisticRegression()
         lgr.fit(x_train,y_train)

         C:\Users\DX\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed wh
         en 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)

Out[34]: LogisticRegression()
```

Predicting our output with the model which we build

```
In [35]: y_pred=lgr.predict(x_test)

In [37]: y_pred1=lgr.predict([[129,99,1,0,0,1,0,1]])
         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 1 1 0 1 1 0 0 0 0 1 0 1 1 0 0 1
Out[37]: Counter({0: 47, 1: 33})

In [39]: accuracy_score(y_test,y_pred)

Out[39]: 0.9125
```

Confusion Matrix of our model

In [40]:
```
conf_mat=confusion_matrix(y_test,y_pred)
conf_mat
```

Out[40]:
```
array([[47,  7],
       [ 0, 26]], dtype=int64)
```

IBM Deployment

In [2]:
```
!pip install -U ibm-watson-machine-learning
```

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-ma
chine-learning) (4.8.2)
```

Authenticate and Set Space

In [1]:
```
wml_credentials = {
    "apikey":"QXCDaXxG__YG-rHuYkTQwfeatR70SW8MTP9Lr29ORWNB",
    "url":"https://us-south.ml.cloud.ibm.com"
}
```

In [8]:
```
from ibm_watson_machine_learning import APIClient
import json
```

In [9]:
```
wml_client = APIClient(wml_credentials)
```

In [11]:
```
wml_client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
----------------------------------  ------------  ------------------------
```

In [57]:
```
SPACE_ID= "d307bf7a-276c-4443-b82c-f850214871c8"
```

In [58]:
```
wml_client.set.default_space(SPACE_ID)
```

Out[58]: 'SUCCESS'

In [60]:
```
wml_client.software_specifications.list(500)
```

```
------------------------------  ----------------------------------  ----
NAME                            ASSET_ID                            TYPE
```

In [37]:
```
## Save and Deploy the model
```

In [15]:
```
import sklearn
sklearn.__version__
```

Out[15]: '1.0.2'

In [ ]:
```
x = data.iloc[:, :-1]
y = data['id']
```

In [ ]:
```
x.head
```

In [34]:
```
y.tail
```

Out[34]: <bound method NDFrame.tail of 0 0

In [ ]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y)
```

In [44]:
```
## Build a machine learning model
```

In [ ]:
```
from sklearn.linear_model import LogisticRegression
lgr=LogisticRegression()
lgr.fit(x_train,y_train)
```

In [46]:
```
MODEL_NAME = 'CKD'
DEPLOYMENT_NAME = 'CKD Notebook'
DEMO_MODEL = model
```

In [64]:
```
# Set Python Version
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

In [65]:
```
# Setup model meta
model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
```

In [66]:
```
#Save model
model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=x_train,
    training_target=y_train
)
```

In [67]:
```
model_details
```

Out[67]:
```
{'entity': {'hybrid_pipeline_software_specs': [],
   'label_column': 'id',
   'schemas': {'input': [{'fields': [{'name': 'age', 'type': 'float64'},
      {'name': 'bp', 'type': 'float64'},
      {'name': 'sg', 'type': 'float64'},
```

[70]:
```
model_id = wml_client.repository.get_model_id(model_details)
model_id
```

t[70]:
```
'091254ff-fb60-4562-91c8-67b69e496e35'
```

[71]:
```
# Set meta
deployment_props = {
    wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
    wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
```

In [72]:
```
# Deploy
deployment = wml_client.deployments.create(
    artifact_uid=model_id,
    meta_props=deployment_props
)
```

```
#######################################################################################

Synchronous deployment creation for uid: '091254ff-fb60-4562-91c8-67b69e496e35' started

#######################################################################################


initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='e0e99a97-2d3d-482b-bedf-d1a5f97dfdc4'
------------------------------------------------------------------------------------------------
```

GitHub and Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-14189-1659544470

Demo Link :
https://drive.google.com/file/d/1UL1q7R4kwt2JojlMSYkG_ICaz1DDBscg/view?usp=drivesdk