CHRONIC KIDNEY DISEASE ANALYSIS USING MACHINE LEARNING

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Chronic diseases require long-term, continuous management. They take a long time to manifest and are difficult to cure. According to the Current Status and Future Development of Chronic Disease Management Project of the Korean Ministry of Health and Welfare, death by five major chronic diseases (hypertension, stroke, angina pectoris, myocardial infarction, and diabetes mellitus) constituted 63.1% of the total deaths in Korea in 2003. While the cost burden of diseases has increased annually, the number of deaths caused by chronic diseases also continues to increase. Various approaches have been introduced to prevent chronic diseases, and most of them focus on lifestyle. However, it is difficult for individuals to change their lifestyle to prevent chronic diseases, because many people do not know which chronic diseases, they may be susceptible to based on their physical condition and medical history. Although a few approaches have been used to predict the possibility of contracting these diseases, their performance was limited because relevant information on the physical condition and medical history was often omitted. Various studies on chronic diseases have received a lot of attention since the 1990s. A few studies were conducted if smoking, drinking, and high cholesterol levels cause chronic diseases. The cholesterol level with stroke and coronary heart disease using experimental groups. Other related studies have included reports investigating the effects of dietary supplements on preventing chronic diseases. One such dietary supplement is chlorella, which is reportedly effective in facilitating growth and improving stress-related ulcers in individuals at high risk for chronic disease. A study evaluating the effects of chlorella use found that this supplement improved fat metabolism and lowered blood glucose levels, suggesting that it may have beneficial effects in preventing chronic disease. Disease development may also be influenced by an individual's living environment. A recent study quantified many diseases and risk factors that correspond to environmental variables by conducting correlation analyses on stress related variables and chronic diseases. As more health information data become available, some machine learning approaches have been implemented to predict the characteristics of chronic disease potential using data as input variables and to predict these as individual medical histories. However, studies of chronic disease are usually experimental; hence, the resulting datasets tend to contain many missing values. Consequently, researchers are unlikely to obtain complete medical records and relevant information when analyzing chronic diseases. However, to the best of our knowledge, only a few approaches have predicted chronic diseases when there are missing values, and most of them have focused on handling them by imputation instead of implicit treatment.

1.2 PURPOSE

Chronic kidney disease (CKD) has become a global health issue and is an area of concern. It is a condition where kidneys become damaged and cannot filter toxic wastes in the body. The proposed system predominantly focuses on predicting this life-threatening disease Chronic Kidney Disease (CKD) using Classification algorithms (KNN and Naive Bayes). Proposed system is automation for chronic kidney disease prediction using classification techniques and supervised learning algorithms. The data for dataset is obtained from UCI machine learning repository which contains 25 parameters (features) including the class (CKD or NOT CKD).

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

A system that can predict multiple diseases with the help of various machine learning algorithms such as Naive Bayes, KNN, DT and SVM algorithms to bridge the gap among the patients and the doctors to achieve their own goals. The existing approaches in the field of automatic disease prediction lack the patient's trust in the model's prediction and also reduce the need for doctors, which makes the doctors get panic about their livelihood. But this method integrates a module for doctor recommendation that solves both the issues by making sure the patient to trust due to the intervention of doctors and also improves the business of doctors.

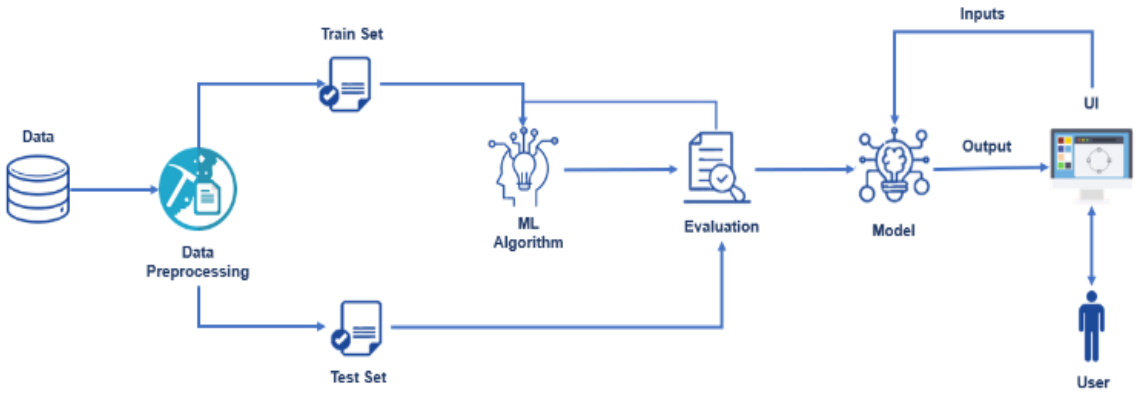
A system that enhances the risk prediction of a patient's health condition using a deep learning approach on big data and a revised fusion node model. This deep learning model for extracting the data and logical inference is made of the combination of complex machine learning algorithm such as Bayesian fusion and neural networks. The architecture of this system consists of five layers, namely, the data layer that is responsible for data collection, data aggregation layer for data acquisition from several data sources and desired format changing, analytics layer to do proper analytics on the data aggregated, information exploration layer to create the output that makes the results of analytics understandable for users, and big data governance layer that is responsible for managing the above layers.

2.2 PROPOSED SYSTEM

In order to solve the problems for the accuracy of the classification system, we proposed a new classification model. First, based on the pretrained models, the models were fine-tuned with the public dataset we used. Based on their performance, the best model was selected in order to further adjust the performance for high accuracy in classifying ships in inland river waterways. After selecting the best model, the model was adjusted, and classification was conducted based on the modification of the network.

3.THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE AND SOFTWARE DESIGNING

Software requirements:

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum , and first released on February 20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding , make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and mac OS.Conda is an open-source, cross platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Python specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Web framework used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user.

Hardware Requirements:

o Operating system: window7 and above with 64bit
o Processor Type -Intel Core i3-3220
o RAM: 4Gb and above
o Hard disk: min 100gb

4.EXPERIMENTAL INVESTIGATION

There are several Machine learning alogrithms to be used depending on the data you are going to process such as images, sound, text, and numerical values.The alorithms that you can choose according to the objective that you might have it may be Classification algorithms or Regression algorithms.

Examples:

1.Linear Regression.

2.Logistic Regression.

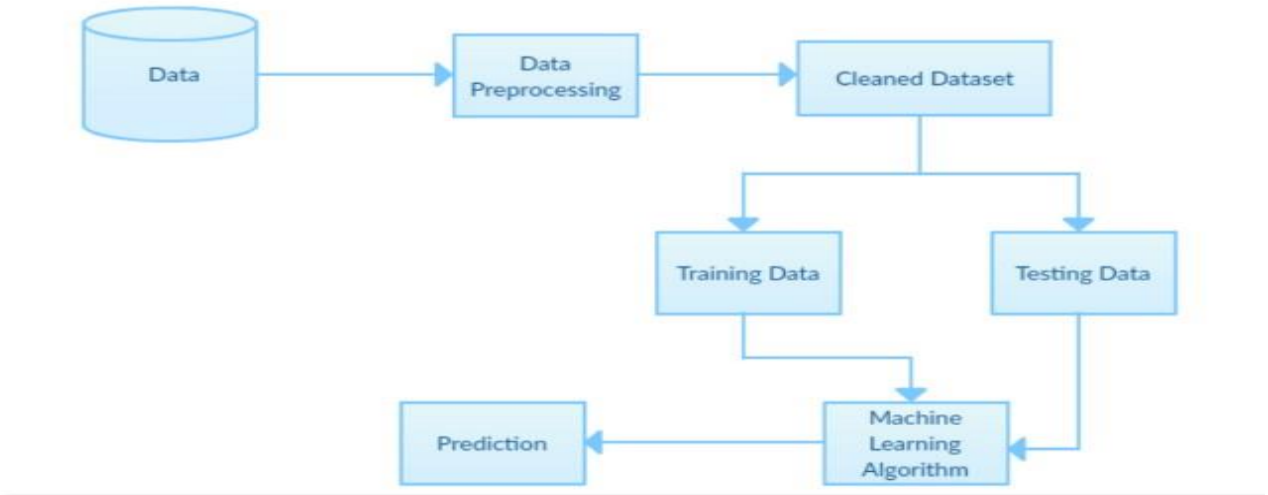
3.Random Forest Regression/Classification.

4.Decision Tree Regression/Classification.

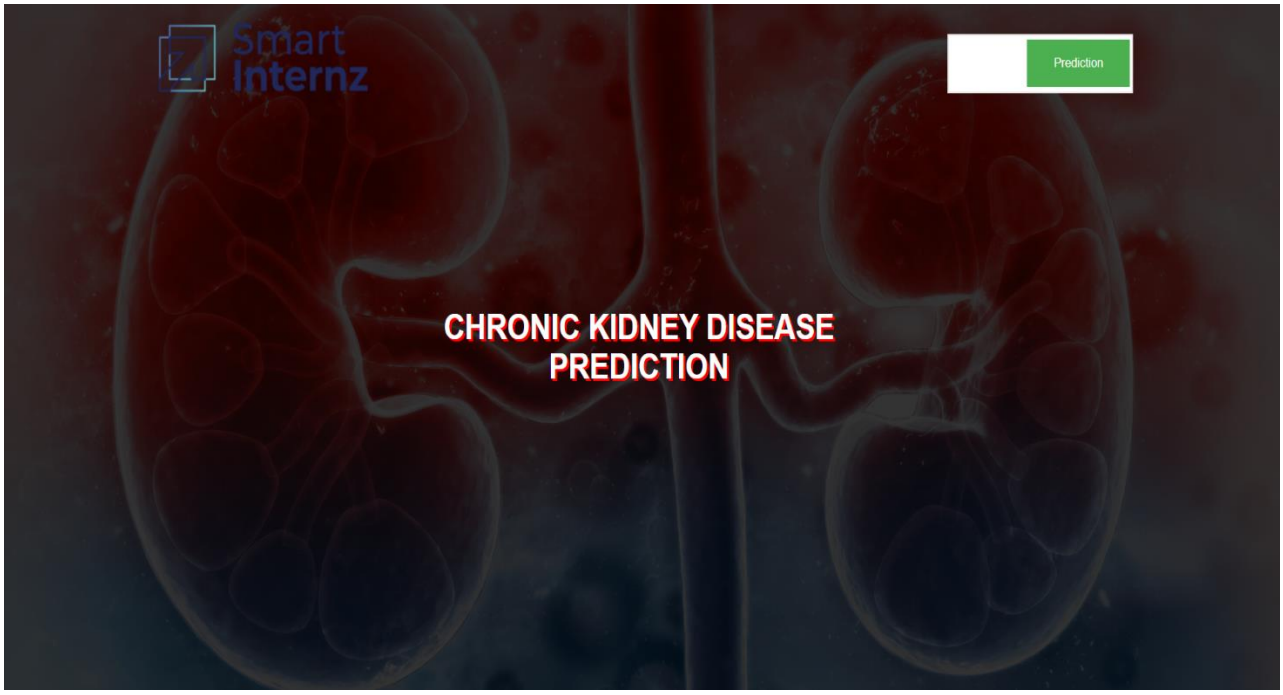
As the prediction for model is classification type, we apply a logistic regression algorithm on our dataset.

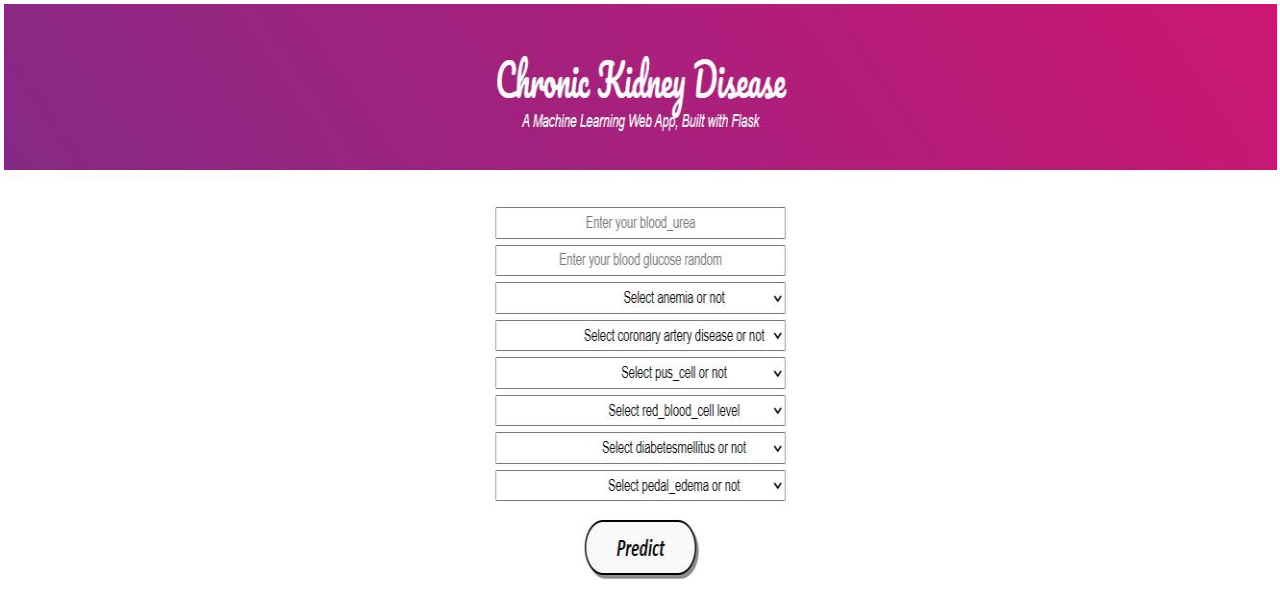
Logistic regression is the appropriate regression anaylsis to conduct when the dependent variable is binary, Like all regression analyses, the logistic regression is a predictive anaylsis. Logistic regression is used to describe data or more nominal ,ordinal, interval or ratio-level independent variables.

5.FLOWCHART



6.RESULTS


Smart Internz
Prediction
CHRONIC KIDNEY DISEASE PREDICTION


Chronic Kidney Disease
A Machine Learning Web App, Built with Flask
Enter your blood_urea
Enter your blood glucose random
Select anemia or not
Select coronary artery disease or not
Select pus_cell or not
Select red_blood_cell level
Select diabetesmellitus or not
Select pedal_edema or not
Predict

Chronic Kidney Disease

A Machine Learning Web App, Built with Flask

5	
5.8	
YES	
NO	
normal	
normal	
YES	
YES	

Predict

Chronic Kidney Disease

A Machine Learning Web App, Built with Flask

Prediction: Oops! You have Chronic Kidney Disease.

7.ADVANTAGES

Predictive modeling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data. It works by analyzing current and historical data and projecting what it learns on a model generated to forecast likely outcomes.We found that machine learning can predict the occurrence of individual chronic diseases, progression, and their determinants and in many contexts. The findings are original and relevant to improve clinical decisions and the organization of health care facilities.

8.DISADVANTAGES

In Chronic Disease prediction, for classification problem we get a very good accuracy but for regression, we get considerable error rate. So, we need to add some more data or change the machine algorithm or by using deep learning techniques for reducing the error in predicting the probability values.

9.CONCLUSION

The principal part of this work is to make an effective diagnosis system for chronic disease of patients.The application will have the option to predict chronic disease prior and advise the wellbeing condition. This application can be surprisingly gainful in low-salary nations where our absence of medicinal foundations and just as particular specialists. In our study, there are a few bearings for future work in this field. We just explored some popular supervised machine learning algorithms, more algorithms can be picked to assemble an increasingly precise model of chronic kidney disease prediction and performance can be progressively improved. Additionally, this work likewise ready to assume a significant role in health care research and just as restorative focuses to anticipate chronic disease.

10.FUTURE SCOPE

Diseases related to kidney is becoming more and more common with time. With continuous technological advancements, these are only going to increase in the future. Although people are becoming more conscious of health nowadays and are joining yoga classes, dance classes; still the sedentary lifestyle and luxuries that are continuously being introduced and enhanced; the problem is going to last long. So, in such a scenario, our project will be extremely helpful to the society. With the dataset that we used for this project, we got 89% accuracy for Random forest model, and though it might be difficult to get such accuracies with very large datasets, from this projects results, one can clearly conclude that we can predict the risk of chronic diseases with accuracy of 95 % or more. Also it can be incorporated into a wide range commercial website and these app and website will be highly beneficial for a large section of society.

11.BIBILOGRAPHY

1. https://www.researchgate.net/publication/335698017_Detection_of_Chronic_Kidney_Disease_using_Machine_Learning_Algorithms_with_Least_Number_of_Predictors

2. https://www.primescholars.com/articles/early-prediction-of-chronic-kidney-disease-by-using-machine-learning-techniques-92643.html

3. https://www.ijert.org/chronic-kidney-disease-prediction-using-machine-learning

11.APPENDIX

SOURCE CODE:

APP.PY

```
# importing the necessary dependencies
import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import pickle



app = Flask(__name__) # initializing a flask app
model = pickle.load(open('CKD.pkl', 'rb')) #loading the model

@app.route('/')# route to display the home page
def home():
    return render_template('home.html') #rendering the home page
@app.route('/Prediction',methods=['POST','GET'])
```

```
def prediction():
    return render_template('indexnew.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
    return render_template('home.html')

@app.route('/predict',methods=['POST'])# route to show the predictions in a web UI
def predict():

    #reading the inputs given by the user
    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', 'red_blood_cells',
      'diabetesmellitus', 'pedal_edema']

    df = pd.DataFrame(features_value, columns=features_name)

    # predictions using the loaded model file
    output = model.predict(df)

    # showing the prediction results in a UI# showing the prediction results in a UI
    return render_template('result.html', prediction_text=output)

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

HOME.HTML

```
<form action="/Prediction" method="[POST,GET]">
<header>
   <div class="wrapper">
      <div class="logo">
        <img src="static/logo.ico" class="w3-round" alt="Norway">
      </div>
      <link rel="stylesheet" type="text/css" href="style.css">
      <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}">

<ul class="nav-area">

<li><a href="#">Home</a></li>

   <input  type="submit" value="Prediction">

</ul>
</div>
<div class="welcome-text">
      <h1>CHRONIC KIDNEY DISEASE PREDICTION</h1>
 </div>
</header>



</form>
```

INDEX1.HTML

```
<html lang="en" dir="ltr">
   <head>
```

```
    <meta charset="utf-8">
    <title>CKD Predictor</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static/css', filename='styles1.css') }}">
    <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">

  </head>
 <div>

    <!-- Website Title -->
      <div class="container">
      <h2 class='container-heading'><span class="heading_font">Chronic Kidney Disease</span></h2>
      <div class='description'>
            <p>A Machine Learning Web App, Built with Flask</p>
        </div>
      </div>

    <!-- Text Area -->
   <div class="ml-container">
        <form action="{{ url_for('predict') }}" method="POST">

<style>
  select {
  text-indent: 29%;
  text-align: center;
      width: 350px;
      height: 25px;
```

```
        margin-bottom: 5px;
    }


</style>
            <input class="form-input" type="text" name="blood_urea" placeholder="Enter your blood_urea"><br>


<input class="form-input" type="text" name='blood glucose random' placeholder="Enter your blood glucose random"><br>


<select id="anemia" name="anemia">
<option value="">Select anemia or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>
<select id="coronary_artery_disease" name="coronary_artery_disease">
<option value="">Select coronary artery disease or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>
<select id="pus_cell" name="pus_cell">
<option value="">Select pus_cell or not</option>
    <option value="0">normal</option>
    <option value="1">abnormal</option>
  </select><br>
<select id="red_blood_cell" name="red_blood_cell">
<option value="">Select red_blood_cell level</option>
     <option value="0">normal</option>
    <option value="1">abnormal</option>
  </select><br>
<select id="diabetesmellitus" name="diabetesmellitus">
```

```
<option value="">Select diabetesmellitus or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>
<select id="pedal_edema" name="pedal_edema">
<option value="">Select pedal_edema or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>



                    <input type="submit" class="my-cta-button" value="Predict">
            </form>
            </div>
</div>
    </body>
</html>
```

INDEXNEW.HTML

```
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title>CKD Predictor</title>
        <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css') }}">
        <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
```

```
  </head>
 <body>


      <!-- Website Title -->
         <div class="container">
         <h2 class='container-heading'><span class="heading_font">Chronic Kidney Disease</span></h2>
         <div class='description'>
                        <p>A Machine Learning Web App, Built with Flask</p>
                </div>
         </div>


      <!-- Text Area -->
         <div class="ml-container">
                <form action="{{ url_for('predict') }}" method="POST">
<style>
   select {
   text-indent: 29%;
   text-align: center;
         width: 350px;
         height: 25px;
         margin-bottom: 5px;
   }


</style>
            <input class="form-input" type="text" name="blood_urea" placeholder="Enter your blood_urea"><br>


<input class="form-input" type="text" name='blood glucose random' placeholder="Enter your blood glucose random"><br>
```

```
<select id="anemia" name="anemia">
<option value="">Select anemia or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>
<select id="coronary_artery_disease" name="coronary_artery_disease">
<option value="">Select coronary artery disease or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>
<select id="pus_cell" name="pus_cell">
<option value="">Select pus_cell or not</option>
    <option value="0">normal</option>
    <option value="1">abnormal</option>
  </select><br>
<select id="red_blood_cell" name="red_blood_cell">
<option value="">Select red_blood_cell level</option>
     <option value="0">normal</option>
    <option value="1">abnormal</option>
  </select><br>
<select id="diabetesmellitus" name="diabetesmellitus">
<option value="">Select diabetesmellitus or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
  </select><br>
<select id="pedal_edema" name="pedal_edema">
<option value="">Select pedal_edema or not</option>
    <option value="1">YES</option>
    <option value="0">NO</option>
```

```
  </select><br>


                <input type="submit" class="my-cta-button" value="Predict">
        </form>
        </div>


    </body>
</html>
```

RESULT.HTML

```
<html lang="en" dir="ltr">
        <head>
                <meta charset="utf-8">
                <title>Chronic Kidney Disease</title>
                <link rel="shortcut icon" href="{{ url_for('static', filename='diabetes-favicon.ico') }}">
                <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css') }}">
                <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
                <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
        </head>

    <body>

       <!-- Website Title -->
                <div class="container">
```

```
    <h2 class='container-heading'><span class="heading_font">Chronic Kidney Disease</span></h2>
    <div class='description'>
        <p>A Machine Learning Web App, Built with Flask</p>
    </div>
    </div>


    <!-- Result -->
    <div class="results">
        {% if prediction_text==1 %}
            <h1>Prediction: <span class='danger'>Oops! You have Chronic Kidney Disease.</span></h1>
            <img class="gif" src="{{ url_for('static', filename='diabetes.webp') }}" alt="Diabetes Image">
        {% elif prediction_text==0 %}
            <h1>Prediction: <span class='safe'>Great! You DON'T have Chronic Kidney Disease</span></h1>
            <img class="gif1" src="{{ url_for('static', filename='no-diabetes.webp') }}" alt="Not Diabetes Image">
        {% endif %}
    </div>


    </body>
</html>
```