PROJECT REPORT

EARLY DETECTION OF CHRONIC KIDNEY DISEASE USING MACHINE LEARNING

Team Leader	Febia Thomas
Team Members	Aryan TSB Girish Kumar S Madhav Hari
Problem Statement	Early Detection of Chronic Kidney Disease using Machine Learning

ABSTRACT

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. A machine learning powered web application model with the strong building of algorithm that helps to identify and predicts the disease with the identification of symptoms from various parameters entered by the user.

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the Nalaiya Thiran Project undertaken during B. Tech. Fourth Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to Dr. (Prof). V.M. Sivagami, Associate Professor, Department of IT for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to all faculty members, Department of IT, for their constant support and guidance throughout the course of our work. Their sincerity, thoroughness and perseverance have been a constant source of inspiration for us. They have showered us with all their extensively experienced ideas and insightful comments at virtually all stages of the project & have also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

ARYAN TSB

FEBIA THOMAS

GIRISH KUMAR S

MADHAV HARI V

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- CKD - Chronic Kidney Disease
- UI - User Interface
- UX - User Experience
- ML - Machine Learning
- CSV - Comma separated Values

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION

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

1.2 MOTIVATION

Attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease. The information says that it helps us to measure the severity of the problem and we make use of such information to build a machine learning model that predicts Chronic Kidney Disease.

1.3 PROPOSED SOLUTION

A machine learning powered web application model with the strong building of algorithm that helps to identify and predicts the disease with the identification of symptoms from various parameters entered by the user. The system offers flexible and comfortable UI experience for first time users - Usually, users visit the website and there is chance that they just leave or close the application due to some complicated features that is built in it. For to resolve such issues, the application's UI/UX design must be considered majorly of how it can be used easily by any users. So, our application can provide better UI experience. The system provides good classification ability in earlier prediction of kidney disease - The prediction must be extremely true and as expected by the user. User might be beginner in using the application and they may not be aware of true and fake results that some other application may show in front of them. Web application and Cloud Deployed ML Model - The application is connected with the machine learning model that is trained and is deployed in the cloud environment.

INDEX TERMS: Supervised and Unsupervised machine learning, Data mining , Python web application interface – Flask , IBM Cloud.

2. LITERATURE SURVEY

Table 1 - Literature Survey

S.NO	TITLE	AUTHOR	YEAR	FINDINGS
1.	Detailed Review of Chronic Kidney Disease	Kakitapalli Y, Ampolu J, Madasu S.D, Sai Kumar M.L.S	2020	A good approach for identifying CKD is to screen people, current recommendations suggest screening of individuals with structural diseases of the renal tract Hypertension, CVD diabetes family history of kidney disease autoimmune diseases with potential for kidney involvement during routine primary health encounters.
2.	Chronic Kidney Disease Prediction using Machine Learning	Reshma S , Salma Shaji , S R Ajina , Vishnu Priya S R , Janisha A	2020	The main objective of this study was to predict patients with CKD using less number attributes while maintaining a higher accuracy of 96%. A wrapper method used here for feature selection is ACO. ACO is a meta-heuristic optimization algorithm. Prediction is

				done using the machine learning technique, SVM.
3.	Chronic Kidney Disease Prediction using Machine Learning Models	S.Revathy, B.Bharathi, P.Jeyanthi, M.Ramesh	2019	Decision tree, Random Forest and Support Vector Machine learning models are constructed to carry outthe diagnosis of CKD. The results of the research showed that Random Forest Classifier model better predicts CKD in comparison to Decision trees and Support Vector machines.

4.	Prediction of kidney disease stages using data mining algorithms	El-Houssainy A.Rady , Ayman S.Anwar	2019	The Probabilistic Neural Networks algorithm gives the highest overall classification accuracy percentage of 96.7%, compared to other algorithmsin classifying the stages of CKD patients. On the other hand, the Multilayer Perceptron requires a minimum executiontime (3 s) whereas the Probabilistic Neural Network requires 12 s to finalize the analysis.

3. SOLUTION ARCHITECTURE

Data is collected and made in a common csv format. This data is then loaded, preprocessed in order to remove null values, segregate the dependent and independent variables, encode the needed columns, create analysis maps, split the data into training and testing data, choose the model which can suit this problem, train the model with the training data, test the accuracy with the test data against predicted data and save the model to integrate it with a web app. A web app is built which renders a form for the user to enter the attributes. The saved model is loaded and the entered values are fed into the loaded model and the predicted results are returned to the user. The model is then deployed into the cloud for the web app to request from the deployed model.

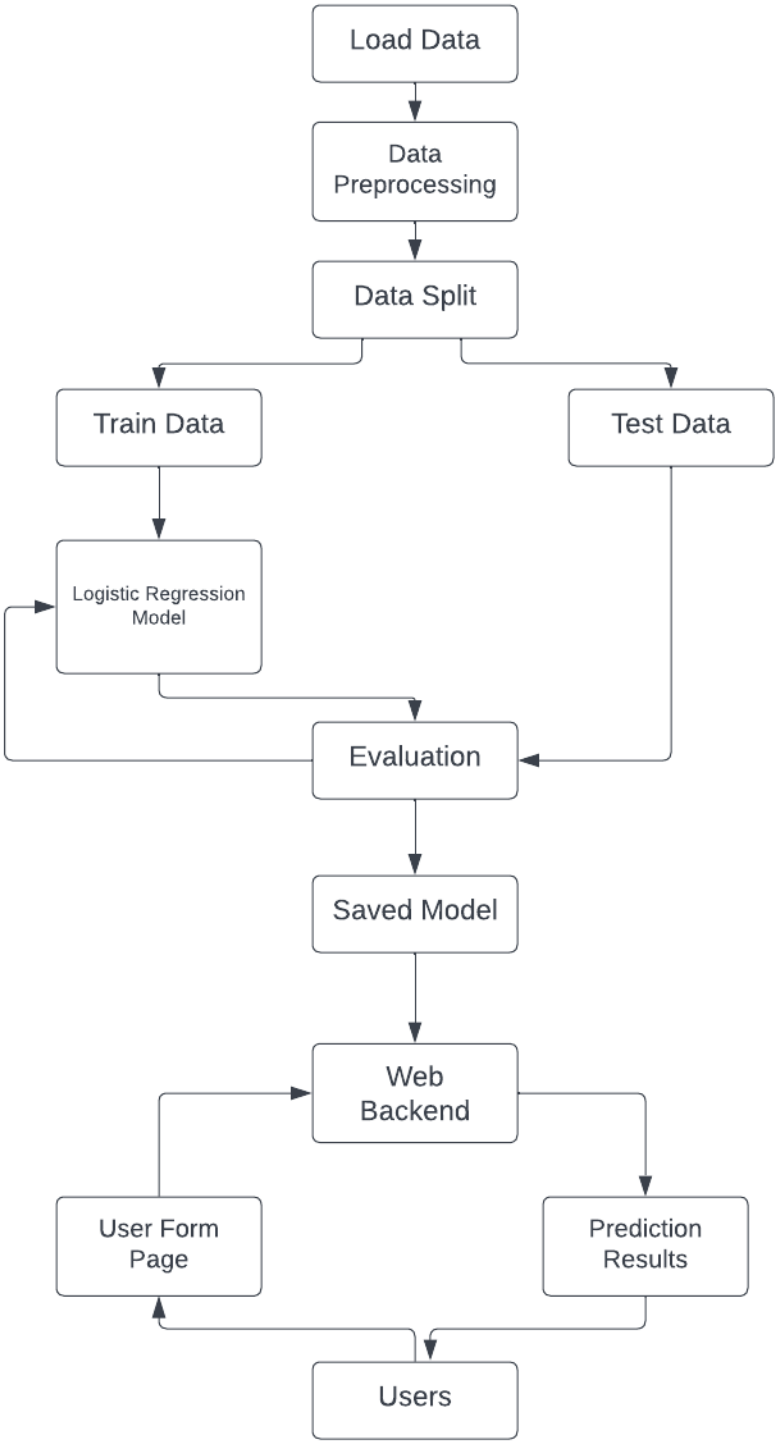
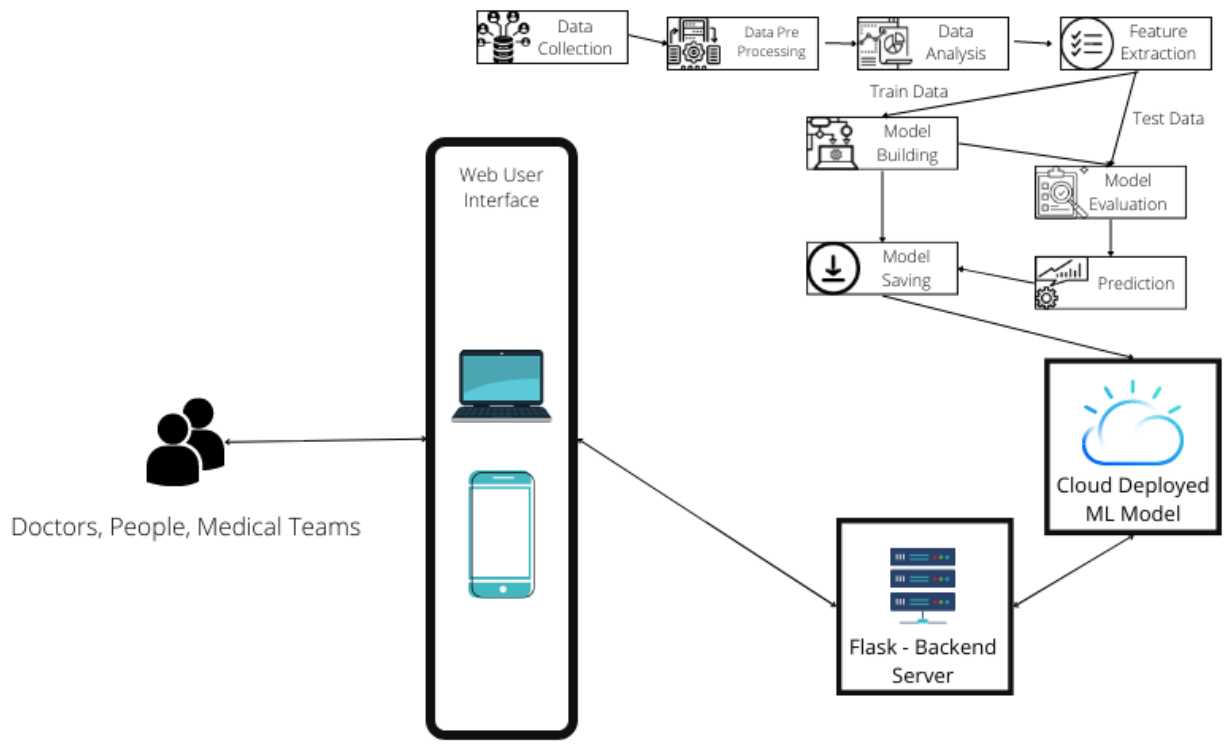


Figure 1 - Solution Architecture

4. TECHNOLOGY ARCHITECTURE



Doctor must manually examine and suggest medical diagnosis in which the symptoms might vary from person to person so suggesting medicine is also a challenge. So hence the disease examination varies at different instances of the medical operations. Here by using machine learning methods, the problem can be addressed with very less error rate. The dataset of Kidney disease is used as input. Also, our proposed system provides accurate results by a good accuracy percentage. We propose a web application for the medical team as well as normal users. This can be used direct by medical team for analyzing and offering the solutions at much positive scaling time. Since the machine learning model is saved and deployed in a cloud environment the app is fast to accept user requests, predict the result and return the response to the users. The web app is deployed in a auto scaling environment.

5. SYSTEM DESIGN

5.1 SOFTWARE REQUIREMENTS SPECIFICATION

Table 2 - Functional Requirements

FR NO.	FUNCTIONAL REQUIREMENT	SUB REQUIREMENT (STORY / SUB-TASK)
FR-1	Input Data	Users should be able to see the form in which they enter the medical test result values such as blood glucose level, blood urea level, RBC level, Anemia level etc when they enter the home url.
FR-2	Validation	Users should be able to submit the form with all the necessary attributes filled and should be appropriately notified when some values are missing.
FR-3	Prediction	The app should accept the test values and feed those inputs into the pre-saved trained machine learning model and should return the prediction result.
FR-4	User Interaction	The app should redirect the users to the appropriate page based on the prediction result.
FR-5	Data Extraction	User gets their personal disease report data from the application through email/downloadable form
FR-6	Medical Assistance	User receives the medical suggestions and assistance for to offer recovery

Table 3 - Non Functional Requirements

FR NO.	NON FUNCTIONAL REQUIREMENT	SUB REQUIREMENT (STORY / SUB-TASK)
NFR-1	Usability	The application can be used for accurate prediction to find the possibility of a kidney disease
NFR-2	Security	User’s data is well encrypted using stable

		machine learning algorithms
NFR-3	Reliability	The application is monitored periodically in terms of its constant prediction ability, quality, and availability towards the user
NFR-4	Performance	It classifies the input data and predicts the disease with careful accuracy output
NFR-5	Availability	The application is active throughout the day. While awaiting the prediction result, User can see medical suggestions
NFR-6	Scalability	It does not request money or bank details to setup their account and download their final medical result from the application

5.2 DATA FLOW DIAGRAM

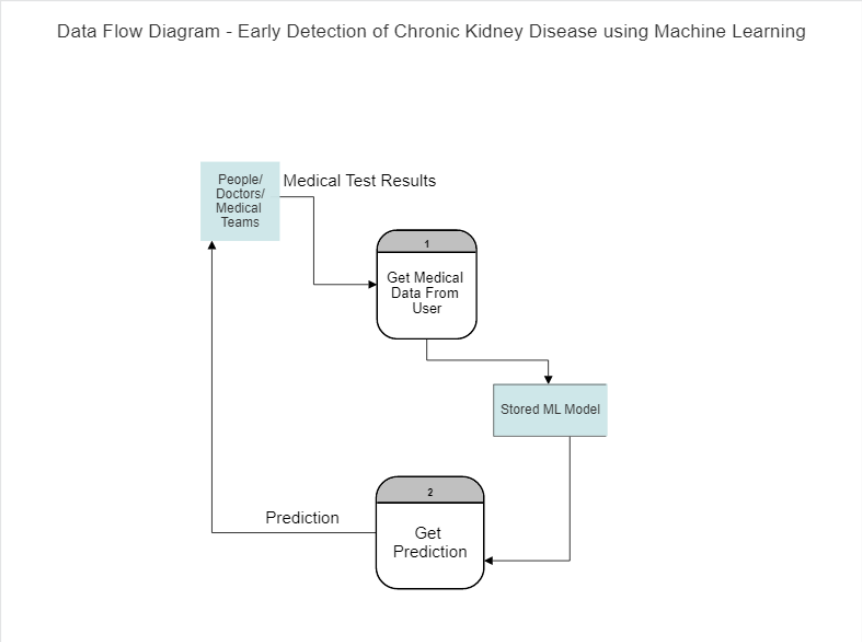


Figure 2 - DFD

5.3 USER STORIES

Table 4 - User Stories

User Type	Functional Requirement	User Story Number	User Story	Acceptanc ecriteria	Priority	Release
Public User/Doctors/Medical Teams	Adding Data	USN-1	As a user, I can feed my data as the input into the application for it to predict the possibility of kidney disease.	All the required data should be andwill be collected fromthe form	High	Sprint-1
Public User/Doctors/Medical Teams	Checking accuracy	USN-2	As a user, I can check theability and accuracy of the model in obtaining the required information	I can check the capability of the model	Medium	Sprint-1
Public User/Doctors/Medical Teams	Data classification	USN-3	As a user, I can examine the working action of the application model	I can view how the application works and responds tothe actions imposed	Medium	Sprint-2

Public User/Doctors/Medical Teams	Checking for the disease	USN-4	As a user, I can verify with the application that the given data is identified with the possibility of kidney disease with the help of the trained and tested data	I can confirm that the data shows the accurate result	High	Sprint-2
Public User/Doctors/Medical Teams	User interaction	USN-5	As a user, I can interact with the web app to process the accurate result in a meanwhile	I can see the results from the interaction	High	Sprint-3
Public User/Doctors/Medical Teams	Medical assistance	USN-6	As a user, I can get medical advises and recommendations for to boost the action of curing the disease	I can get enough assistance bygetting the suggestions for curing thedisease	Low	Sprint-3

Public User/Doctors/Medical Teams	Data extraction	USN-7	As a user, I can retrieve the result datafrom the application for data storage for further medical research uses.	I can download theresult in the form of data or email the result as a proof to showto medical teams	Medium	Sprint-4

5.4 CUSTOMER JOURNEY

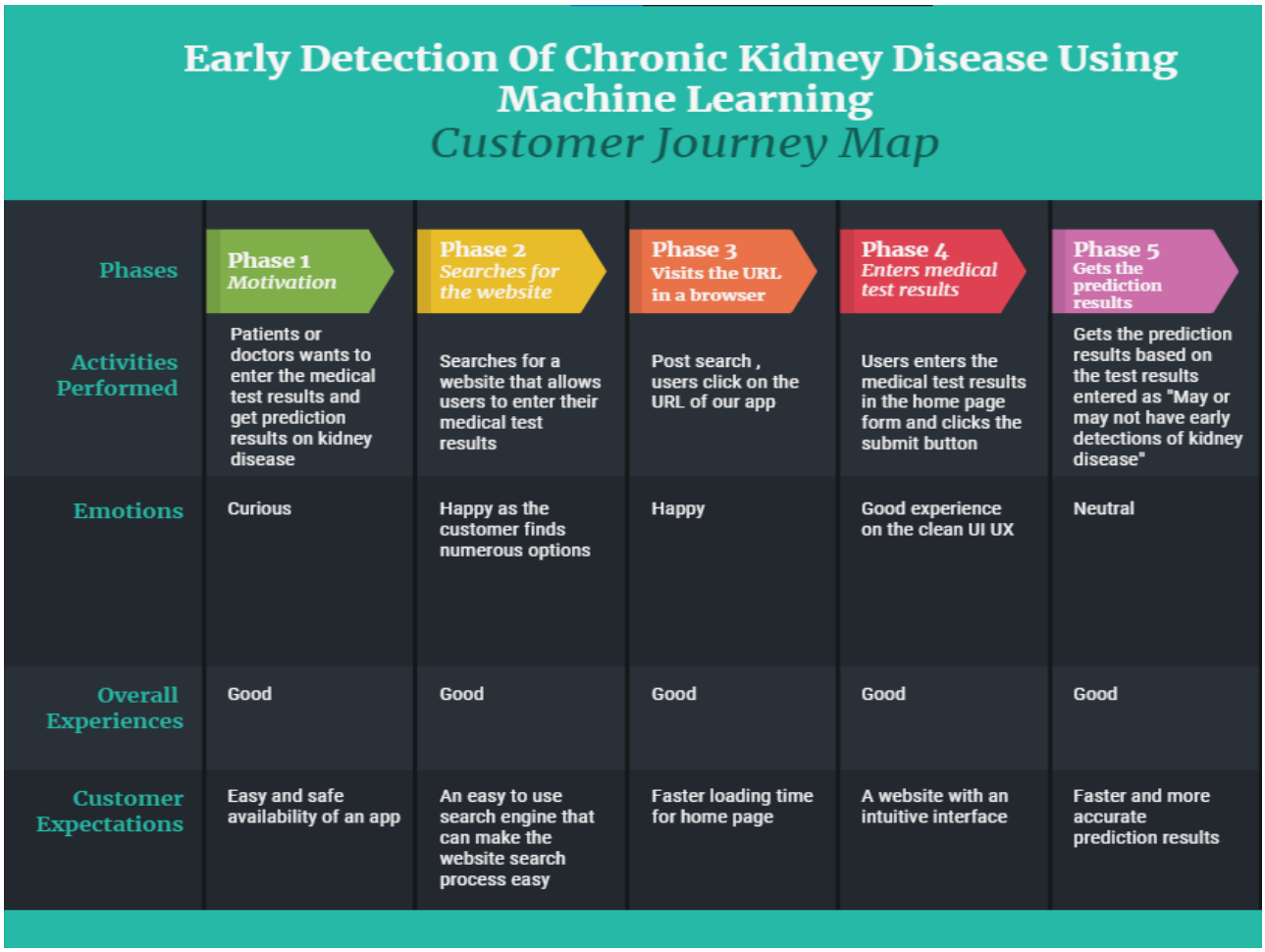


Figure 3 - Customer Journey Map

6. SYSTEM IMPLEMENTATION

6.1 DATA PROCESSING & MODEL BUILDING

- Install all the necessary modules
- Import all the necessary modules
- Read the csv dataset file into a data frame
- Drop the “id” column
- Rename the column names for better readability
- Check for null values and data type in each column
- Replace the mismatched column labels in categorical columns
- Create separate lists for categorical and columnar columns
- Clean the input labels
- Replace the null values in continuous columns with mean value
- Replace the null values in categorical columns with mode value
- Perform label encoding in the categorical columns
- Separate the necessary independent and dependent columns from the data frame
- Split the dataset into training and testing data
- Train the model with the training data frames
- Test the model
- Print the accuracy and confusion matrix
- Save the model for testing purposes

6.2 CLOUD DEPLOYMENT OF ML MODEL

- Signup for the IBM Cloud Service
- Create IBM Watson Studio along with IBM Machine Learning Service and IBM Storage
- Upload the python notebook, dataset in Watson studio and store the model in the attached machine learning service
- Generate the API Key from IAM service to get the auth token
- Get the URL for the deployed machine learning model

6.3 WEB INTERFACE

- Create a new virtual environment and install the flask, requests, smtp, email modules.
- Create 3 routes (“/”,”/prediction”,”/send_mail”). The “/” route will send the index.html page to user where the user can enter their medical test results.
- The /prediction route accepts only POST request and gets the input form values and forms the header, payload and sends the value to cloud ML model to get the prediction
- Based on the prediction the user will be redirected whether they have possibility of having chronic kidney disease or not
- The send_mail route is used to send a mail report of the prediction results

7. RESULT

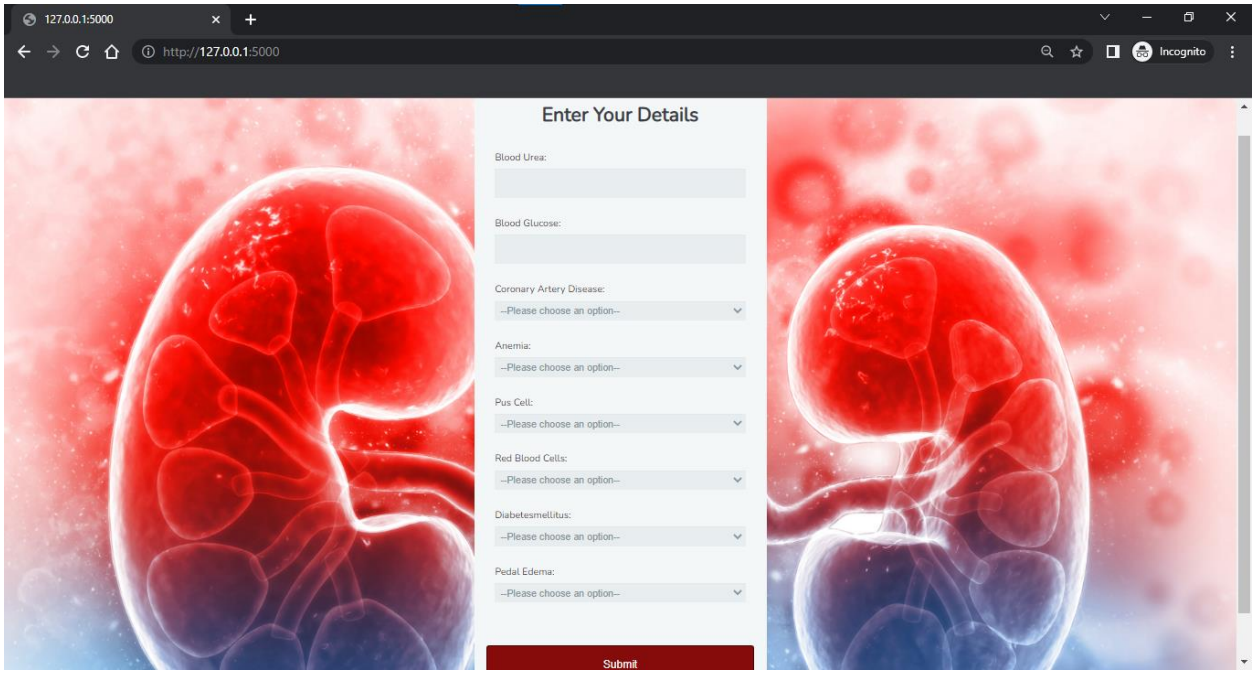


Figure 4 - Home Page

You're chances of having Chronic Kidney Disease is less with 89 % !!

girish26412@gmail.com Send Mail

Figure 5 - No CKD

You may have Chronic Kidney Disease with a 89 %!!!

girish26412@gmail.com Send Mail

Figure 6 - CKD

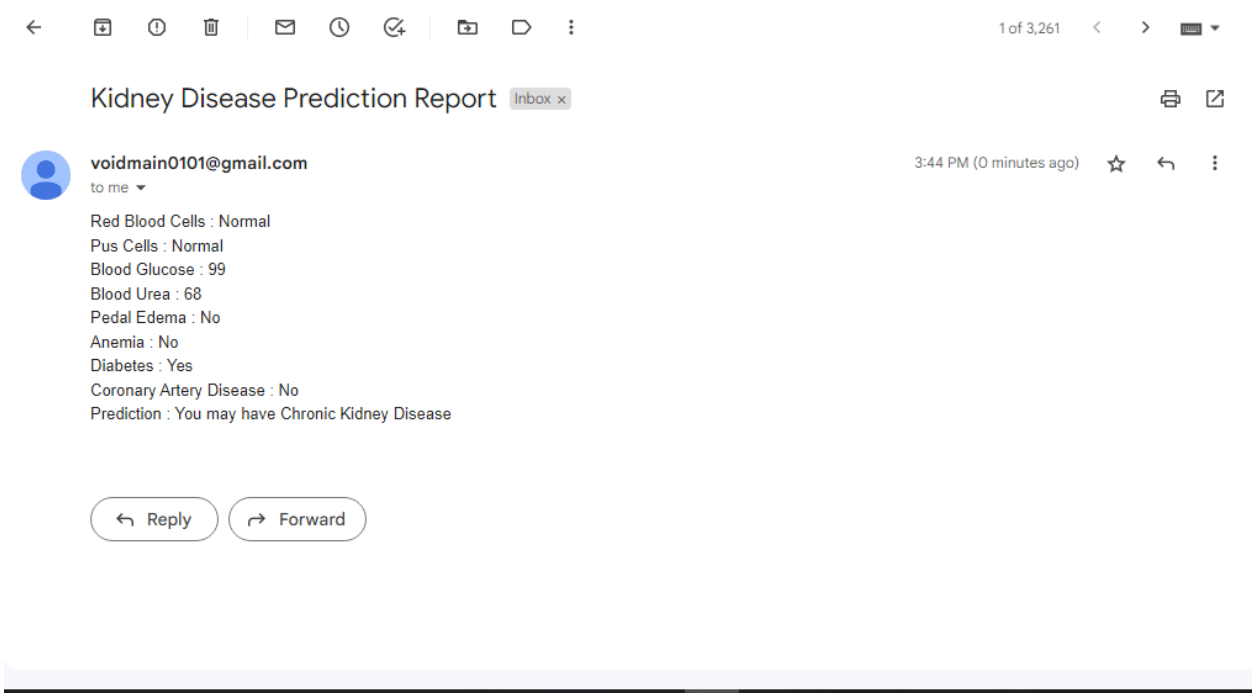


Figure 7 - Email Report

8. CONCLUSION

A machine learning based web application that all users to enter their medical details and get prediction on whether they have possibility of having chronic kidney disease. The problem is been analyzed to classify if it is a regression or a classification kind of problem. Pre-process on the data using different data pre-processing techniques has been applied and valuable insights have been gained. Different algorithms according to the dataset have been applied and accuracy is found. A web application using Flask framework is built to interact with the deployed machine learning model to get the prediction and get an email report.

REFERENCES

1. Detailed Review of Chronic Kidney Disease,Kakitapalli Y, Ampolu J, Madasu S.D, Sai Kumar M.L.S , 2020
2. Chronic Kidney Disease Prediction using Machine Learning, Reshma S , Salma Shaji , S R Ajina , Vishnu Priya S R , Janisha A , 2020
3. Chronic Kidney Disease Prediction using Machine Learning Models,S.Revathy, B.Bharathi, P.Jeyanthi, M.Ramesh,2019

4. Prediction of kidney disease stages using data mining algorithms , El-Houssainy A.Rady , Ayman S.Anwar , 2019

APPENDIX

A.1 LOGISTIC REGRESSION

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.

A.2 FLASK

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine.

A.3 IBM CLOUD

IBM Cloud Paks are software products for hybrid clouds that enable you to develop apps once and deploy them anywhere. Virtual Private Cloud (VPC) is available as a public cloud service that lets you establish your own private cloud-like computing environment on shared public cloud infrastructure.