

**CO2 EMISSION RATING BY VEHICLES USING DATA SCIENCE TECHNIQUE**

#### A PROJECT REPORT

***Submitted by***

### AVANTIKA.V 211419205023

### HARRIAT LINDA.L 211419205067

### NIVESHAA.R 211419205121

***in partial fulfillment for the award of the degree of***

### BACHELOR OF TECHNOLOGY

***in***

#### INFORMATION TECHNOLOGY

**PANIMALAR ENGINEERING COLLEGE, POONAMALLEE**

### ANNA UNIVERSITY: CHENNAI 600 025

**APRIL 2023**

## ANNA UNIVERSITY: CHENNAI 600 025

### BONAFIDE CERTIFICATE

Certified that this project report **“CO2 EMISSION RATING BY VEHICLES USING DATA SCIENCE TECHNIQUE”** is the bonafide work of **“AVANTIKA.V(211419205023), HARRIAT LINDA.L(211419205067), NIVESHAA.R(211419205121) ”** who carried out the project under my supervision.

#### SIGNATURE SIGNATURE

**Dr. M. HELDA MERCY M.E., Ph.D., MS.HARITHA M.E.,**

**HEAD OF THE DEPARTMENT SUPERVISOR**

Department of Information TechnologyDepartment of Information Technology Panimalar Engineering College Panimalar Engineering College

Poonamallee, Chennai - 600 123 Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### SIGNATURE SIGNATURE

**INTERNAL EXAMINER EXTERNAL EXAMINER**

#### DECLARATION

I hereby declare that the project report entitled “**CO2 EMISSION RATING BY VEHICLES USING DATA SCIENCE TECHNIQUE**” which is being submitted in partial fulfilment of the requirement of the course leading to the award of the ‘Bachelor Of Technology in Information Technology’ in **Panimalar Engineering College, Autonomous institution Affiliated to Anna university- Chennai** is the result of the project carried out by me under the guidance of **Ms.HARITHA M.E.,in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

Date**: AVANTIKA.V**

Place**:** Chennai **HARRIAT LINDA.L**

#### NIVESHAA.R

It is certified that this project has been prepared and submitted under my guidance.

Date: Ms.HARITHA

Place: Chennai (DESIGNATION / IT)

#### ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Beloved Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E.,M.B.A.,Ph.D.,**and **Dr.saranya**

**sree sakthikumar.,B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project co-ordinator **Mr. M. DILLI BABU**, **M.E.,(P.hD.,)** Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our supervisor **Ms.HARITHA** for providing the support to carry out the project successfully. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

#### 4

**ABSTRACT**

Our personal vehicles are a major cause of global warming. Collectively, cars account for nearly one-fifth of all emissions, emitting around 24 pounds of carbon dioxide and other global-warming gases for every gallon of gas. About five pounds comes from the extraction, production, and delivery of the fuel, while the great bulk of heat-trapping emissions-more than 19 pounds per gallon-comes right out of a car's tailpipe. It is assumed the average gasoline vehicle on the road today has a fuel economy of about 22.0 miles per gallon and drives around 11,500 miles per year. Every gallon of gasoline burned creates about 8,887 grams of CO2. More than Twenty years ago (1998) the car industry agreed to a voluntary commitment to reduce new car emissions by 25% by 2008. 10 Then, CO2 emissions on the road from new cars were around 203g/km. 11 Today, they are still around 170g/km and unlikely to reach 140g/km until after 2020. A typical passenger vehicle emits about 4.6 metric tons of carbon dioxide per year. This number can vary based on a vehicle's fuel, fuel economy, and the number of miles driven per year. The higher the number of the controlled and uncontrolled effect variables that influence the co2 properties, the lesser the predicted accuracy. Despite this, a few experimental designs have been suggested by considering the controllable effect variables and interaction terms between them. To predict the emission of gas from cars we develop a model which uses the attributes of the car to specify whether the car has exceeded the threshold value of CO2 if it exceeded the threshold then (Road and Transport Authority) RTA will seize that particular car. Supervised machine learning technique is one of the great technique for predicting the CO2 emission rating.

5

#### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO** | **TITLE** | **PAGE NO** |
|  | **Abstract** | 5 |
|  | **List of Tables** | 8 |
|  | **List of Figures** | 9 |
|  | **List of abbreviation** | 10 |
| **1** | **Introduction** | 11 |
|  | 1.1 Overview of the project | 12 |
|  | 1.2 Need of the system | 12 |

* 1. [Objective of the project 13](#_TOC_250009)
  2. [Scope of the project 13](#_TOC_250008)

1. Literature Survey 14
   1. Chronic Kidney Disease And its Complications 15
   2. Attributable causes of chronic kidney disease in adults 15

a five-year retrospective study in a tertiary-care hospital in the northeast of the Malaysian Peninsula

* 1. Detection of Chronic Kidney Disease Using Machine 15

Learning Algorithms with Least Number of Predictors

* 1. Detection and diagnosis of chronic kidney disease 16

using deep learning-based heterogeneous modified artificial neural network

* 1. Diagnosis of Chronic Kidney Disease Using 16

Effective Classification Algorithms and Recursive Feature Elimination Techniques

* 1. Prediction of Chronic Kidney Disease – A Machine 16

Learning Perspective

* 1. Neural network and support vector machine for the 17

prediction of chronic kidney disease

1. System Design 18
   1. [Architecture Diagram 19](#_TOC_250007)
2. Requirement Specification 23
   1. System Analysis 24
   2. [Software Requirements 25](#_TOC_250006)
   3. Hardware Requiremwnts 27
3. Implementation 30
   1. [Flow Diagram 31](#_TOC_250005)
   2. [Working 32](#_TOC_250004)
   3. Code 33
   4. Output 43
4. Uml Diagram And Testing 48
   1. [Uml Diagrams 49](#_TOC_250003)
   2. [Testing 51](#_TOC_250002)
   3. Testing cases 52
5. Conclusion And Future Enhancement 53
   1. [Conclusion 54](#_TOC_250001)
   2. [Future Enhancement 54](#_TOC_250000)

Appendices

#### LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of the Table** | **Page No.** |
| 6.3.1  6.3.2 | Test cases for Unit Testing  Test case Results | 79  79 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name Of the Figure** | **Page No.** |
| 3.1.1 | Proposed System Architecture | 23 |
| 3.2.1 | Process Data Flow Diagram For Clustering Phase | 24 |
| 3.2.2 | Process Data Flow Diagram For Filtering Phase | 25 |
| 3.3.2 | Agglomerative Clustering Algorithm | 27 |
| 3.3.5 | UML Class Diagram | 30 |
| 3.3.6 | UML Sequence Diagram For Clustering Phase | 31 |
| 3.3.7 | UML Sequence Diagram For Filtering Phase | 32 |
| 3.3.8 | UML Collaboration Diagram For Clustering Phase | 33 |
| 3.3.9 | UML Collaboration Diagram For Filtering Phase | 33 |
| 3.3.10 | UML Use Case Diagram | 34 |
| 5.2.1 | Screen Shot For Loading Dataset | 86 |
| 5.2.2 | Screen Shot Displaying Cluster Formation | 86 |
| 5.2.3 | Screen Shot Displaying Matrix Generation | 87 |
| 5.2.4 | Screen Shot For Login | 87 |
| 5.2.5 | Screen Shot For Searching | 88 |
| 5.2.6 | Screen Shot For Software Querying | 88 |
| 5.2.7 | Screen Shot which Displays Filtered Content | 89 |
| 5.2.8 | Screen Shot Displaying Predicted Result | 89 |
| 5.2.9 | Screen Shot Displaying Viewing Description | 90 |
| 5.2.10 | Screen Shot For Rating Item | 90 |
| 5.2.11 | Screen Shot which displays Posted Comment | 91 |
| 5.2.12 | Screen Shot For New User Registration | 91 |
| 5.2.13 | Screen Shot For New User Search | 92 |
| 5.2.14 | Screen Shot Displaying Recommendations for New User | 92 |

**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| CKD  NCKD SC | Chronic Kidney Disease  No Chronic Kidney Disease Serum Creatinine |
| SOD | Sodium |
| POT | Potassium |
| HEMO | Hemoglobin |
| RBC | Red Blood Cell |
| WBC | White Blood Cell |
| BP | Blood Pressure |

# CHAPTER 1 INTRODUCTION

#### OVERVIEW OF THE PROJECT

Using machine learning, we choose the subject of knowledge set to improve diagnosis of chronic kidney disease (CKD) and kidney stones. We are all aware that CKD is one of the most common diseases among individuals nowadays. By gathering a large amount of CKD and stone data. We are working hard to improve detection in people. It usually affects the Adults above the age of 30 and continues indefinitely. Males between the ages of 30 and 35 have stage 5 CKD. without a transplant, one could expect to live for 14 years. For women of the same age,

13 years are predicted to be their lifespan. Everybody has a four-year life expectancy between the ages of 70 and 75, regardless of gender. GFR Rate is a popular tool for identifying the majority of renal disorders (Glomerular filtration rate). the following three stages of chronic kidney disease: Stage 1 with normal or high GFR (GFR > 90 mL/min) Stage 2 Mild CKD (GFR= 60-89 mL/min) Stage 3A Moderate CKD (GFR = 45-59 mL/min) Stage 3B Moderate CKD (GFR = 30-44 mL/min). We should not smoke. Restrict the alcohol consumption. Maintain a

blood pressure of less than 140/90 mm Hg (or the target set by your doctor). If we have diabetes, we should try to maintain our blood sugar levels within the ideal range. We should get vaccinated for the flu during flu season. They are crucial in the fight against kidney stones and CKD.

#### NEED OF THE SYSTEM

As the kidney disease cases are increasing and need for the system also increases. When the patient goes for diagnosis first, He/she has to take an blood test then they gets the blood test report . The report generation can take days due to the demand and as the number of patients are also increasing. After the report is generated sometimes it takes time to visit the doctor so it may lead to delay in knowing the result whether your kidney is healthy or not , so to avoid that problem this project helps the patients or common people to get the idea whether they have kidney disease or not by entering their values in the website according to the report with that we will be able to find whether kidney disease is present or not and can also able to find the stages of their disease without consulting the doctor . After knowing the result if they come know that they have severe kidney disease they will be to know to consult doctor without delaying any further. The main problem is that the report generation can delay as it is a tedious process and due the increase in patients and this can be prevented with our system as it detects the stages of kidney

#### 12

disease and whether kidney stone is present or not. It also states the required treatment 3 type depending upon the stage. It reduces the waiting time, till the patient gets official confirmation from the doctor it can be used a reference which helps them get prepared mentally. By using this system, a patient can avoid anxiety during a duration for wait of result.

#### OBJECTIVE OF THE PROJECT

The main objective of the system is to reduce the time period for the patients to know the result whether they have chronic kidney or not and if there what is their stage and can also know whether kidney stone is present or not. Blood test will given within 1-2 days. Our objective is to detect stages of chronic kidney disease in an early stage by just giving values that they get after blood test report like albumin, sodium, potassium, creatinine value as input. In output the stages are specified for the input breast values given. By this we can confirm whether the kidney is healthy or sick. This causes the patient to suffer as treatment should be started as fast as they can. This gap can be avoided using our system. This reduces the anxiety of the patients as they get to know what they have been diagnosed. It helps the patient to get mentally prepared before the official report comes.

#### SCOPE OF THE PROJECT

As the cases for chronic liver disease increases, the duration of diagnosis also increases. The delay in diagnosis may result in the loss of patient life. As there are many patients are diagnosed with kidney disease the report generation gets delayed. This creates panic among the patients. The scope of this project is to investigate a dataset of hospital records for the medical sector using machine learning techniques. To identify a patient is affected with CKD or not. We can also make diagnoses as quickly as possible.

# CHAPTER 2 LITERATURE SURVEY

* 1. **TITLE:** Chronic Kidney Disease And its Complications. (2008)

**AUTHOR:** Robert Thomas, MD, Abbas Kans, John R, Sedor

**DESCRIPTION:** When aberrant albumin excretion or impaired kidney function last for longer than three months, as determined by a measured or estimated glomerular filtration rate (GFR), CKD is present. According to the stage of the disease, treatments are suggested for CKD and dialysis patients. These treatments could lower these patients' morbidity and fatality rates. But it has disadvantage that is it may not been determined for sure, it is prudent to abide by FDA instructions.

* 1. **TITLE:** Attributable causes of chronic kidney disease in adults: a five-year retrospective study in a tertiary-care hospital in the northeast of the Malaysian Peninsula (2015)

**AUTHOR:** Muhammad SalmanI, Amer Hayat Khan, Azreen Syazril Adnan, Syed Azhar Syed Sulaiman, Khalid Hussain,Naureen Shehzadi, Fauziah Jummaat.

**DESCRIPTION:** They set out to describe the adult patients' demographics, clinical profile, and potentially causal factors for CKD at a tertiary-care hospital in Malaysia. There advantage is to study alerts the general public to the likelihood that putting more emphasis on diabetes and hypertension primary prevention will significantly reduce hospital admissions due to CKD in Malaysia. And the disadvantage is that the data might not paint a complete picture of the CKD-related causes and Certain data in this study were not completely available due to the back data.

* 1. **TITLE**: Detection of Chronic Kidney Disease Using Machine Learning Algorithms with Least Number of Predictors. (2013)

**AUTHOR:** Marwa Almasoud , Tomas E Ward.

**DESCRIPTION:** In this study, we use the lowest subset of features to investigate how well machine learning algorithms predict chronic kidney disease. The ANOVA test, Pearson's correlation, and Cramer's V test are only a few examples of the statistical tests that have been performed to eliminate redundant features. The pros of this paper are the classifiers have been trained, tested, and verified using 10-fold cross-validation. The gradient boosting approach improved performance in terms of F1-measure (99.1%), sensitivity (98.8%), and specificity

#### 15

(99.3%).And the cons are they want to compare the results with another dataset that has the same

attributes or validate our results using a large dataset. Moreover, to aid in lowering the occurrence of CKD**.**

* 1. **TITLE:** Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network (2016)

#### AUTHOR:

**DESCRIPTION:** Computer vision and machine learning are the fields that develop methods to extract meaningful meanings from digital images. To create the reference standard for segmentation from the validation and training datasets, an experienced radiologist was evaluated to divide the abdomen MR and CT scan pictures into the left and right half. Pros are the proposed HMANN approach helps to segment the kidney picture and lowers noise for precise placement of the kidney stone diagnosis. To successfully solve this issue, tested. CONS: Early kidney stone detection is essential because renal damage might endanger life. To undertake surgery to remove a kidney stone, the location of the kidney must be determined.

* 1. **TITLE:** Diagnosis of Chronic Kidney Disease Using Effective Classification Algorithms and Recursive Feature Elimination Techniques (2021)

#### AUTHOR:

**DESCRIPTION:** The originality of this study is in creating a system for diagnosing chronic renal illnesses. This study supports specialists in studying preventive methods for CKD through early diagnosis utilizing machine learning approaches. This study's main objective was to assess a dataset made up of 400 patients and 24 attributes. The missing nominal and numerical data were replaced using the mean and mode statistical analysis methods. Recursive Feature Elimination (RFE) was used to select the most crucial characteristics. PROS: Our systems' accuracy ranged from 100% with random forest to 97.3% with SVM. CONS: These papers don't accurately depict chronic renal disease.

* 1. **TITLE:** Prediction of Chronic Kidney Disease – A Machine Learning Perspective ( 2021)

#### AUTHOR:

**DESCRIPTION:** This article has examined chronic kidney disease prediction from this angle. In this study, seven classifier methods were used. The results have been calculated for each classifier based on the following features: (i) full features; (ii) correlation-based feature selection;

#### 16

(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; and (vi) synthetic features. PROS: Again, the linear support vector machine provided the maximum accuracy of 98.46% in the synthetic minority

over-sampling technique with the least absolute shrinkage and selection operator selected features. CONS: Logistic and KNN were not employed in SMOTE since they did not produce the desired results.

* 1. **TITLE:** Neural network and support vector machine for the prediction of chronic kidney disease (2019)

#### AUTHOR:

**DESCRIPTION:** Using a dataset of 400 patients and 24 variables linked to the diagnosis of chronic kidney disease, we concentrate on using several machine learning classification methods in this article. In this study, artificial neural networks (ANN) and support vector machines were employed as classification methods (SVM). PROS: One of the most well-known machine learning methods is the support vector machine (SVM), while another is the artificial neural network (ANN). Both methods are beneficial and have a history of producing outstanding results across a range of industries. In comparison to SVM, which has shown the highest accuracy in prior studies, ANN has been presented as a novel model to more accurately predict CKD. The missing values were initially replaced after pre processing the dataset. CONS: The results in this paper were not accurate.

**17**

# CHAPTER 3 SYSTEM DESIGN

#### 18

#### ARCHITECTURE DIAGRAM:

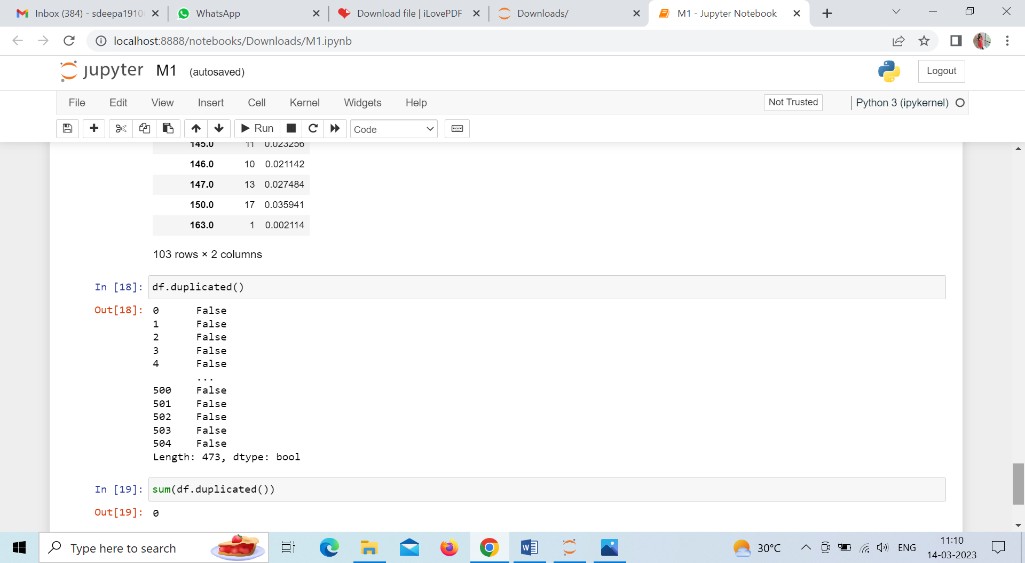
#### Fig 3.1 Architecture Diagram

**DESCRIPTION:** The basic Architecture diagram of the “CHRONIC KIDNEY DISEASE AND KIDNEY STONE PREDICTION USING SML TECNIQUE” is shown in the above figure. Mainly this diagram consists of the following essential Blocks:

* + 1. Pre processing
    2. Visualization
    3. Algorithm Implementation
    4. Deploy

1. **Data Preprocessing:** - Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the important process in machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data so to verify that and to get a clean dataset we do data preprocessing so after that we will be able to obtain clean and formatted dataset and then we can proceed further with our project implementation. **19**

* In this the input values will be in the form of raw dataset and after pre processing the output will be clean and formatted dataset.
* To get pre processed data first we will import libraries like pandas and numpy where pandas is used to analyze big data and to make conclusions and numpy is used for working with arrays. After importing the libraries we will link the dataset with the notebook for processing with the help of df and read function. Then we use shape , size and column function to know more about the dataset for processing . isnull function is used to return a dataframe object where values are replaced with true for null values and false for non-null values. We use dropna function to remove rows which contain null values. And also we use groupby and duplicate function for aggregation , analysis and remove duplicate values.



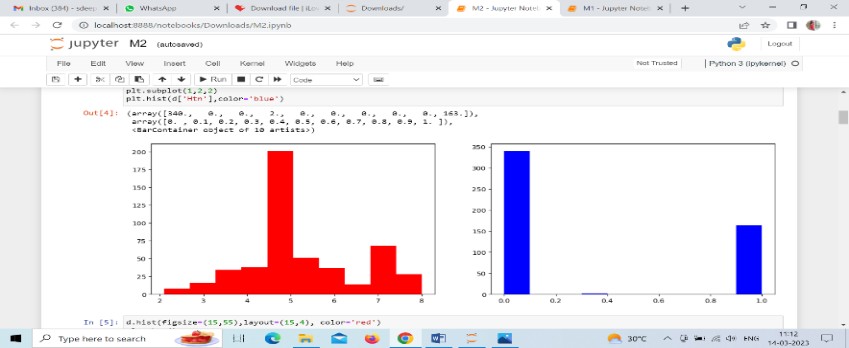
#### Fig 3.2 Data Preprocessing

1. **Data Visualization: -** In this data visualization module the data’s are displayed has visualized data using graphs and charts like histogram , pie-chart, bar graph etc.Data visualization is a valuable tool for gaining a qualitative understanding. This can help with identifying patterns, corrupt data, outliers, and other issues when exploring and getting to know a dataset.

* In this module the input data will be pre-processed data and output will be visualized data which means the data will be represented in the form of graph or plot.

#### 20

* In this module also we will import libraries like numpy and pandas but along with that we will also import libraries like seaborn and matplotlib where Seaborn uses Matplotlib underneath to plot graphs and It will be used to visualize random distributions. And Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. In this module we use functions like plt.subplot, plt.hist, plt.bar, plt.boxplot, sns.displot, sns.heatmap etc. to obtain the visualized data in the form of graph or plot.



#### Fig 3.3 Data Visualization

1. **ALGORITHM IMPLEMENTATION:** - In this project we totally use 4 algorithms to predict the output they are:
   1. DECISION TREE CLASSIFIER: It only poses a question and divides the tree into subtrees according to the response ( Yes/No). There are 2 nodes they are decision node and leaf node. Decision node is used to make decision and leaf node is the outcome of such decisions.
   2. LOGISTIC REGRESSION: It is used for predicting the categorical dependent variable using a given set of independent variables. Outcome will be either true or false, 1 or 0, yes or no. But instead of giving exact values it provides probabilistic value.
   3. KNN : It makes the assumption that new cases and existing cases are comparable and it places the new instance in the category that is most likely the existing categories.
   4. Naïve Bayes : It makes predictions using the probabilities of each attribute belonging each

class.

* In this the input will be data and the output will be accuracy
* Each and every algorithm is implemented separately to find accuracy then after finding the best accuracy algorithm will be considered for predicting the output i.e., stages of ckd and whether kidney stone is present or not.
* To find the accuracy for each and every algorithm first libraries are imported then dataset are linked, dropna function is performed. Then sklearn module is used to assess the quality of our prediction.

#### 21

In this data’s are trained and tested they are x\_train, x\_test, y\_train and y\_test. Then confusion matrix is performed to find where errors is made in the module. We also perform cros\_val\_score function on dataset to test whether model can generalize over the whole dataset and cros\_val\_score returns the accuracy value. To display the confusion matrix in visual manner imshow and image interpolation functions are used. Finally the output will be displayed displayed in plot manner.

* Same procedure is followed for all the 4 algorithms to find the accuracy .

1. **DEPLOYMENT**: In this module the trained deep learning model is converted into hierarchical data format file (.h5 file) which is then deployed in our Django framework for providing better user interface and predicting the stages of chronic kidney disease and whether kidney stone is present or not according to the data.

**22**

**CHAPTER – 4**

**REQUIREMENT SPECIFICATION**

#### 23

* 1. **SYSTEM ANALYSIS PROPOSED SYSTEM**

The proposed strategy is to create a machine learning model for categorizing renal disease stages and stones. The procedure begins with data gathering, in which previous information about kidney disease stages and stones is gathered. In the healthcare arena, data mining is a widely utilized technique for processing massive amounts of data. The stages of kidney illness and stones, if detected early enough, can save lives. Machine learning is increasingly widely employed in health care, where it minimizes manual labor and, with a better model, errors are reduced, potentially saving lives. The dataset is analyzed and accurate variable identification is performed, which means that both dependent variables and independent variables are discovered. Then appropriate machine learning methods are used which is applied to the dataset where the data pattern is discovered. Following the use of many algorithm, a better algorithm is utilized to predict the outcome.

#### ADVANTAGES:

* + 1. To forecast the stages of kidney illness and stone formation, we are employing machine learning techniques.
    2. For improved prediction, algorithms are compared, and the best model is assessed.
    3. Several algorithms' performance indicators are evaluated in order to provide a more accurate prediction.
    4. Excellent performance and accuracy.
    5. We completed the deployment procedure and compared various algorithms to improve accuracy.

#### SOFTWARE AND HARDWARE REQUIREMENTS

#### SOFTWARE REQUIREMENTS:

* + 1. Windows 10
    2. Anaconda
    3. Jupyter Notebook
    4. Visual Studio

#### WINDOWS 10:



**Fig 4.1 Windows**

* Windows is a graphical operating system developed by Microsoft.
* It allows users to view and store files, run the software, play games, watch videos,and provides a way to connect to the internet.
* It was released for both home computing and professional works.

#### ANACONDA:



**Fig 4.2 Anaconda**

Anaconda is a [distribution](https://en.wikipedia.org/wiki/Software_distribution) of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) [programming languages](https://en.wikipedia.org/wiki/Programming_language) for [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing)

([data science](https://en.wikipedia.org/wiki/Data_science), [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications, large-scale [data processing](https://en.wikipedia.org/wiki/Data_processing), [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics), etc.), that aims to simplify [package management](https://en.wikipedia.org/wiki/Package_management) and [deployment](https://en.wikipedia.org/wiki/Deployment_environment). The distribution includes data-science packages suitable for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), and [macOS](https://en.wikipedia.org/wiki/MacOS). It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) in 2012.As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.Package versions in Anaconda are managed by the package management system [*conda*](https://en.wikipedia.org/wiki/Conda_(package_manager)).This package manager was spun out as a separate [open-source](https://en.wikipedia.org/wiki/Open_source) package as it ended up being useful on its own and for things other than Python.There is also a small, [bootstrap](https://en.wikipedia.org/wiki/Bootstrapping) version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

#### JUPYTER NOTEBOOK:



**Fig 4.3 Jupyter notebook**

Jupyter notebook is an open-source, interactive web application that allows users to create and share documents that contain interactive calculations, code, images, etc. Users can combine data, coe, and visualizations into a single notebook, and create interactive “stories” that they can edit and share.

Notebooks are documents which contain both computer code (such as Python) and other text elements such as paragraph, markdown,figures, links, etc.The Jupyter notebook is widely used and well documented and offers an easy to use interface for creating, editing, and running notebooks. The notebook runs as a web application called the “Dashboard” or “control panel” that shows local files and allows users to open notebook documents and run snippets of code. The outputs are neatly formatted and displayed on the browser.The other component of the notebook is the kernel. The kernel is a “computational engine” that executes the code written in the Notebook. It is similar to the back-end of the application.The IPython kernel(Jupyter was previously called IPython notebook) is used to execute Python code in the Jupyter notebook. There are kernels for other languages as well, but in this article, we will explore running Python code in the notebook.

#### ISUAL STUDIO:



**Fig 4.4 visual studio**

The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to enhance the software development process.

#### HARDWARE REQUIREMENTS:

* + 1. PROCESSOR: PENTIUM IV/III
    2. HARD DISK: 80GB
    3. RAM: 2GB

#### 27

1. **PROCESSOR: PENTIUM IV/III:**



#### Fig 4.5 Pentium

* + Pentium 4 was a series of single-core central processing units (CPU) for desktop PCs and laptops. The series was designed by Intel and launched in November 2000. Pentium 4 clock speeds were over 2.0 GHz.
  + Intel shipped Pentium 4 processors until August 2008. Pentium 4 variants included code named Willamette, Northwood, Prescott and Cedar Mill with clock speeds that varied from 1.3-3.8 GHz.
  + The Pentium 4 processor replaced the Pentium III via an embedded seventh-generation x86 microarchitecture, known as Netburst Microarchitecture, which was the first new chip architecture launched after the P6 microarchitecture in the 1995 Pentium Pro CPU model.

#### 28

1. **HARD DISK: 80GB:**



#### Fig 4.6 Hard disk

Hard disk, also called hard disk drive or hard drive, [magnetic](https://www.britannica.com/science/magnetism) storage medium for a [computer](https://www.britannica.com/technology/computer). Hard disks are flat circular plates made of [aluminum](https://www.britannica.com/science/aluminum) or [glass](https://www.britannica.com/technology/glass) and coated with a magnetic material. Hard disks for [personal computers](https://www.britannica.com/technology/personal-computer) can store terabytes (trillions of bytes) of information. [Data](https://www.britannica.com/dictionary/Data) are stored on their surfaces in concentric tracks. A small [electromagnet](https://www.britannica.com/science/electromagnet), called a magnetic head, writes a [binary](https://www.britannica.com/technology/binary-code) [digit](https://www.britannica.com/technology/binary-code) (1 or 0) by magnetizing tiny spots on the spinning disk in different directions and reads digits by detecting the magnetization direction of the spots. A computer’s hard drive is a device consisting of several hard disks, read/write heads, a drive motor to spin the disks, and a small amount of circuitry, all sealed in a metal case to protect the disks from dust. In addition to referring to the disks themselves, the term *hard disk* is also used to refer to the whole of a computer’s internal data storage. Beginning in the early 21st century, some personal computers and laptops were produced that used solid-state drives (SSDs) that relied on [flash memory](https://www.britannica.com/technology/flash-memory) chips instead of hard disks to store information.

#### RAM: 2GB:

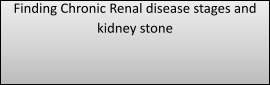
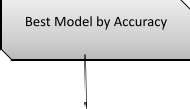
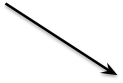
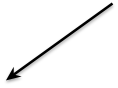


- 2GB of RAM is the sweet spot for the majority of users, providing enough RAM fovirtually all productivity tasks and less demanding games.

## CHAPTER- 5

**IMPLEMENTATION**

#### FLOW DIAGRAM:



**Fig 5.1 Flow Diagram**

#### WORKING

According to the blood test report, the values for each attributes will be entered in the website which we have given. Then the User will click the "PREDICT" button . After then the process will be carried out in the backend to produce the results according to the data which the user entered. In that the first process will be "DATA PREPROCESSING". After Data Preprocessing, Data visualization will be performed for better understanding of the data. After Data visualization, We are separating the dataset into training dataset and testing dataset. The output of the training dataset will be classification of ML algorithms. So with ML algorithms, we would test the dataset for the accuracies of different algorithms. After the accuracy report of each algorithms, we consider the best accuracy algorithm to find the stages and kidney stone according to the values entered by the user.

For finding the stages and kidney stones, the average of attributes values are considered to predict the result as such like Class 0 ,Class 1 , Class 2.

#### CODE MODULE 1 :

import pandas as p import numpy as n import warnings

warnings.filterwarnings('ignore') df=p.read\_csv('A.csv')

df.head()

df.tail() df.shape df.size df.columns df.isnull()

df['Class'].unique() df = df.dropna() df.describe() df.corr()

#### 32

df.info()

p.crosstab(df["Wbcc"], df["Rbcc"])

df.groupby(["Hemo","Htn"]).groups df["Class"].value\_counts() p.Categorical(df["Sod"]).describe() df.duplicated()

sum(df.duplicated())

#### MODULE 2:

import pandas as pd import numpy as n

import matplotlib.pyplot as plt import seaborn as sns

d = pd.read\_csv('A.csv') d.head()

d.columns plt.figure(figsize=(15,5)) plt.subplot(1,2,1) plt.hist(d['Rbcc'],color='red') plt.subplot(1,2,2) plt.hist(d['Htn'],color='blue')

d.hist(figsize=(15,55),layout=(15,4), color='red') plt.show() d['Hemo'].hist(figsize=(10,5),color='red')

plt.bar(d['Su'],d['Sc'], color='red') # scatter, plot, triplot, stackplot plt.boxplot(d['Rbc'])

d['Bu'].plot(kind='density') sns.displot(d['Class'], color='red')

sns.residplot(d['Sc'],d['Sod'], color='red') # residplot, scatterplot sns.pairplot(d) **33**

fig, ax = plt.subplots(figsize=(20,15))

sns.heatmap(d.corr(),annot = True, fmt='0.2%',cmap = 'autumn',ax=ax) def plot(d, variable):

dataframe\_pie = d[variable].value\_counts()

ax = dataframe\_pie.plot.pie(figsize=(9,9), autopct='%1.2f%%', fontsize = 10) ax.set\_title(variable + ' \n', fontsize = 10)

return n.round(dataframe\_pie/d.shape[0]\*100,2) plot(d, 'Class')

#### MODULE 3:

import pandas as pd

import matplotlib.pyplot as plt import warnings warnings.filterwarnings('ignore') data=pd.read\_csv('A.csv') data.head()

df=data.dropna() df

df.columns

x = df.drop(labels='Class', axis=1) y = df.loc[:,'Class']

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20, random\_state=1, stratify=y)

print("Number of training dataset : ", len(x\_train)) print("Number of test dataset : ", len(x\_test))

print("Total number of dataset : ", len(x\_train)+len(x\_test))

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score from sklearn.model\_selection import cross\_val\_score

#### 34

from sklearn.tree import DecisionTreeClassifier DTC = DecisionTreeClassifier() DTC.fit(x\_train,y\_train)

predicted = DTC.predict(x\_test)

cr = classification\_report(y\_test,predicted)

print('Classification report of DecisionTreeClassifier Result is:\n',cr) print("\n")

cm = confusion\_matrix(y\_test,predicted)

print('Confusion Matrix result of DecisionTreeClassifier is:\n',cm) print("\n")

accuracy = cross\_val\_score(DTC, x, y, scoring='accuracy') print('Cross validation test results of accuracy:', accuracy) print("\n")

a = accuracy.mean() \* 100

print("Accuracy Result of DecisionTreeClassifier is:",a)

ef plot\_confusion\_matrix(cm, title='Confusion matrix-DecisionTreeClassifier', cmap=plt.cm.cool):

plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)

plt.colorbar() cm1=confusion\_matrix(y\_test, predicted)

print('Confusion matrix-DecisionTreeClassifier:') print(cm)

plot\_confusion\_matrix(cm) import matplotlib.pyplot as plt df2 = pd.DataFrame() df2["y\_test"] = y\_test df2["predicted"] = predicted

#### 35

df2.reset\_index(inplace=True) plt.figure(figsize=(20, 5))

plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red') plt.plot(df2["y\_test"][:100], marker='o', linestyle='dashed', color='green') plt.show()

import joblib joblib.dump(DTC, 'model.pkl') **MODULE 4:**

import pandas as pd

import matplotlib.pyplot as plt import warnings warnings.filterwarnings('ignore') data=pd.read\_csv('A.csv') data.head()

df=data.dropna() df

df.columns

x = df.drop(labels='Class', axis=1) y = df.loc[:,'Class']

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20, random\_state=1, stratify=y)

print("Number of training dataset : ", len(x\_train)) print("Number of test dataset : ", len(x\_test))

print("Total number of dataset : ", len(x\_train)+len(x\_test))

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score from sklearn.model\_selection import cross\_val\_score

from sklearn.linear\_model import LogisticRegression

LR = LogisticRegression() LR.fit(x\_train,y\_train) predicted = LR.predict(x\_test)

cr = classification\_report(y\_test,predicted)

print('Classification report of LogisticRegression Result is:\n',cr) print("\n")

cm = confusion\_matrix(y\_test,predicted)

print('Confusion Matrix result of LogisticRegression is:',cm) print("\n")

accuracy = cross\_val\_score(LR, x, y, scoring='accuracy') print('Cross validation test results of accuracy:', accuracy) print("\n")

a = accuracy.mean() \* 100

print("Accuracy Result of LogisticRegression is:",a)

def plot\_confusion\_matrix(cm, title='Confusion matrix-LogisticRegression', cmap=plt.cm.cool):

plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)

plt.colorbar() cm1=confusion\_matrix(y\_test, predicted) print('Confusion matrix-LogisticRegression:') print(cm)

plot\_confusion\_matrix(cm) import matplotlib.pyplot as plt df2 = pd.DataFrame() df2["y\_test"] = y\_test df2["predicted"] = predicted df2.reset\_index(inplace=True)

plt.figure(figsize=(20, 5))

plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red') plt.plot(df2["y\_test"][:100], marker='o', linestyle='dashed', color='green') plt.show()

#### MODULE 5:

import pandas as pd

import matplotlib.pyplot as plt import warnings warnings.filterwarnings('ignore') data=pd.read\_csv('A.csv') data.head()

df=data.dropna() df

df.columns

x = df.drop(labels='Class', axis=1) y = df.loc[:,'Class']

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20, random\_state=1, stratify=y)

print("Number of training dataset : ", len(x\_train)) print("Number of test dataset : ", len(x\_test))

print("Total number of dataset : ", len(x\_train)+len(x\_test))

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score from sklearn.model\_selection import cross\_val\_score

from sklearn.neighbors import KNeighborsClassifier KNN = KNeighborsClassifier() KNN.fit(x\_train,y\_train)

predicted = KNN.predict(x\_test)

cr = classification\_report(y\_test,predicted)

print('Classification report of KNeighborsClassifier Result is:\n',cr) print("\n")

cm = confusion\_matrix(y\_test,predicted)

print('Confusion Matrix result of KNeighborsClassifier is:',cm) print("\n")

accuracy = cross\_val\_score(KNN, x, y, scoring='accuracy') print('Cross validation test results of accuracy:', accuracy) print("\n")

a = accuracy.mean() \* 100

print("Accuracy Result of KNeighborsClassifier is:",a)

def plot\_confusion\_matrix(cm, title='Confusion matrix-KNeighborsClassifier', cmap=plt.cm.cool):

plt.imshow(cm, interpolation='nearest', cmap=cmap) plt.title(title)

plt.colorbar() cm1=confusion\_matrix(y\_test, predicted)

print('Confusion matrix-KNeighborsClassifier:') print(cm)

plot\_confusion\_matrix(cm) import matplotlib.pyplot as plt df2 = pd.DataFrame() df2["y\_test"] = y\_test df2["predicted"] = predicted df2.reset\_index(inplace=True) plt.figure(figsize=(20, 5))

plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red')

plt.plot(df2["y\_test"][:100], marker='o', linestyle='dashed', color='green') plt.show()

#### MODULE 6:

import pandas as pd

import matplotlib.pyplot as plt import warnings warnings.filterwarnings('ignore') data=pd.read\_csv('A.csv') data.head()

df=data.dropna() df

df.columns

x = df.drop(labels='Class', axis=1) y = df.loc[:,'Class']

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.20, random\_state=1, stratify=y)

print("Number of training dataset : ", len(x\_train)) print("Number of test dataset : ", len(x\_test))

print("Total number of dataset : ", len(x\_train)+len(x\_test))

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score from sklearn.model\_selection import cross\_val\_score

from sklearn.naive\_bayes import GaussianNB

GN = GaussianNB() GN.fit(x\_train,y\_train) predicted = GN.predict(x\_test)

cr = classification\_report(y\_test,predicted) print('Classification report of GaussianNB Result is:\n',cr) print("\n")

cm = confusion\_matrix(y\_test,predicted) print('Confusion Matrix result of GaussianNB is:',cm) print("\n")

accuracy = cross\_val\_score(GN, x, y, scoring='accuracy') print('Cross validation test results of accuracy:', accuracy) print("\n")

a = accuracy.mean() \* 100

print("Accuracy Result of GaussianNB is:",a)

def plot\_confusion\_matrix(cm, title='Confusion matrix-GaussianNB', cmap=plt.cm.cool): plt.imshow(cm, interpolation='nearest', cmap=cmap)

plt.title(title) plt.colorbar()

cm1=confusion\_matrix(y\_test, predicted) print('Confusion matrix-GaussianNB:') print(cm)

plot\_confusion\_matrix(cm) import matplotlib.pyplot as plt df2 = pd.DataFrame() df2["y\_test"] = y\_test df2["predicted"] = predicted df2.reset\_index(inplace=True) plt.figure(figsize=(20, 5))

plt.plot(df2["predicted"][:100], marker='x', linestyle='dashed', color='red') plt.plot(df2["y\_test"][:100], marker='o', linestyle='dashed', color='green') plt.show()

#### MODULE 7:

from django.shortcuts import render

from django.shortcuts import render, redirect import numpy as np

import joblib

model = joblib.load('C:/Users/DELL/Downloads/PROJECT/PROJECT/deploy/app/model.pkl') # Create your views here.

def home(request):

return render(request, "index.html") def predict(request):

if request.method == "POST":

int\_features = [x for x in request.POST.values()] int\_features = int\_features[1:] print(int\_features)

final\_features = [np.array(int\_features, dtype=object)] print(final\_features)

prediction = model.predict(final\_features) print(prediction)

output = prediction[0]

print(f'output{output}') if output == 0:

return render(request, 'index.html', {"prediction\_text":"NO CHRONIC KIDNEY DISEASE AND NO KIDNEY STONE"})

elif output == 1:

return render(request, 'index.html', {"prediction\_text":"CHRONIC KIDNEY DISEASE STAGE 1 AND NO KIDNEY STONE"})

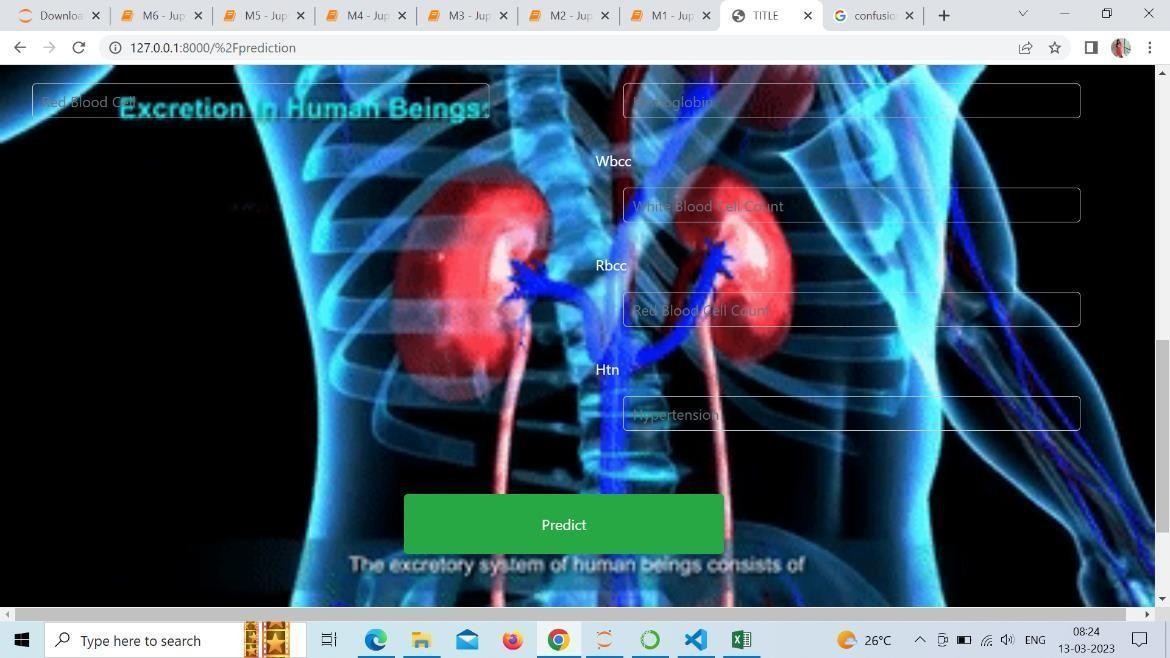
else:

return render(request, 'index.html', {"prediction\_text":"CHRONIC KIDNEY DISEASE STAGE 2 AND KIDNEY STONE"})

print(output)

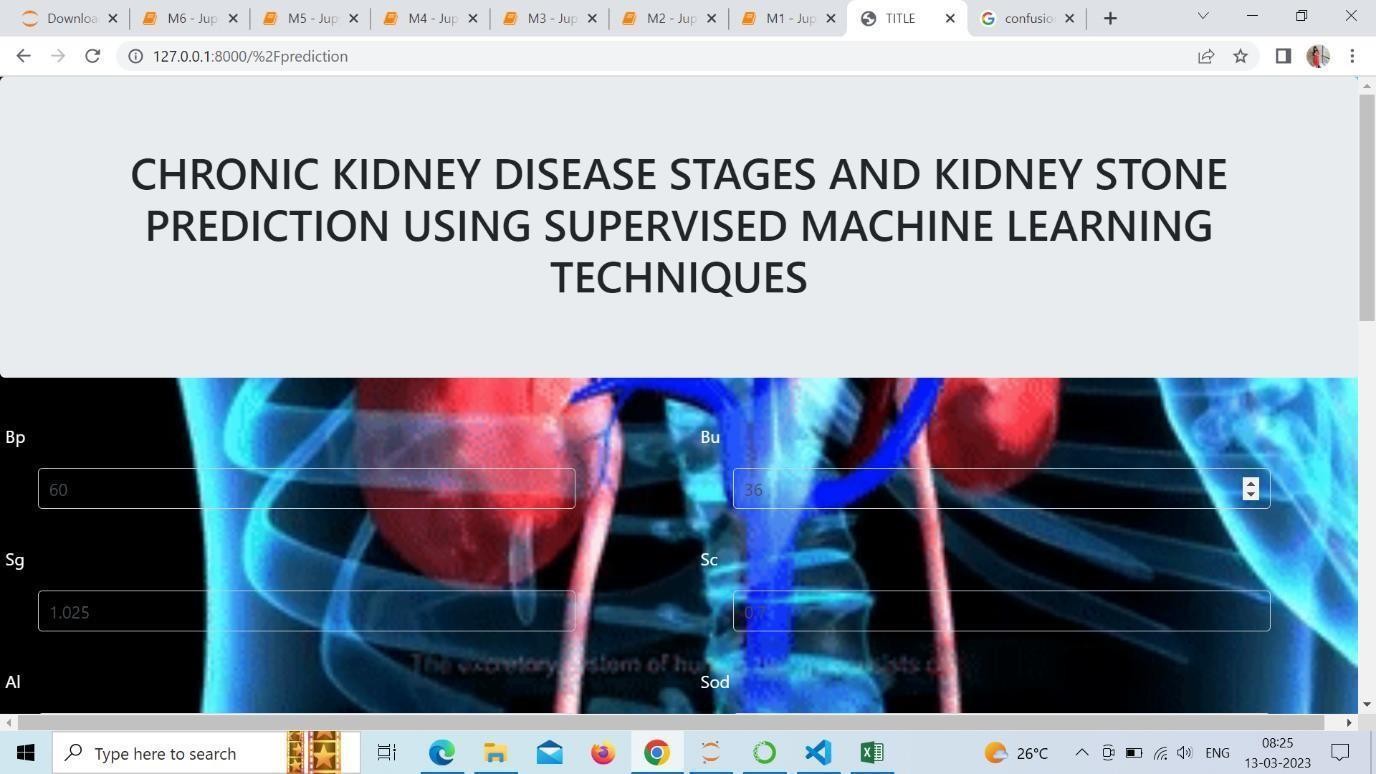
#### OUTPUTS:

**Fig 5.4 Output**

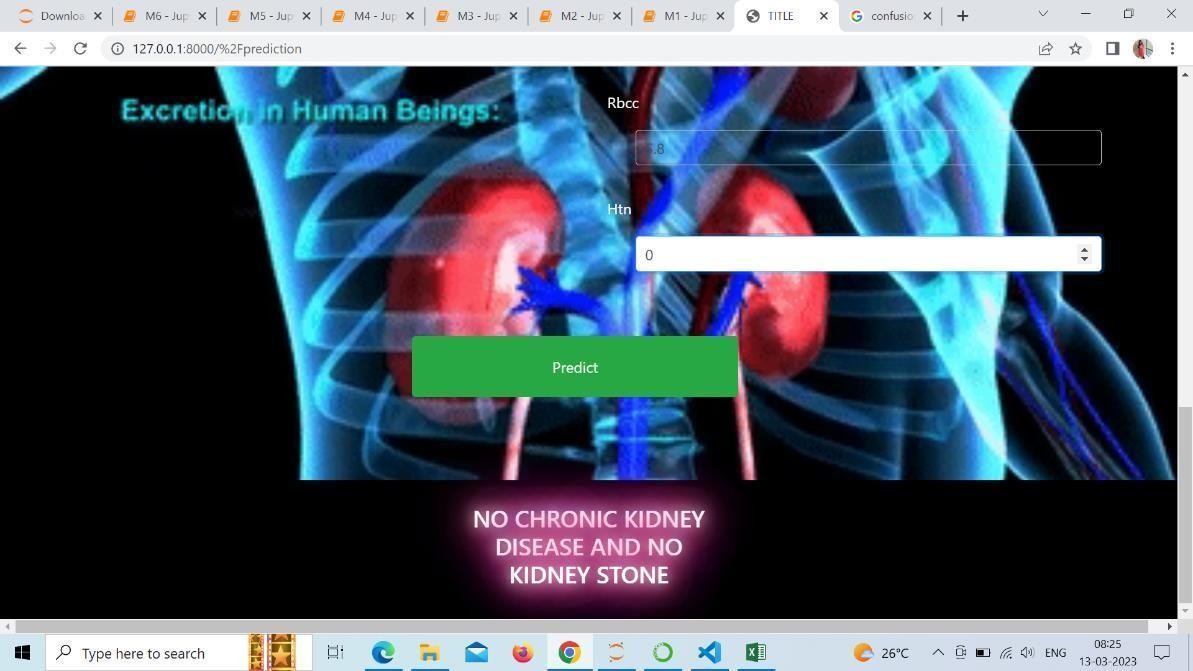


#### Fig 5.5 Output

**STAGE 0:**



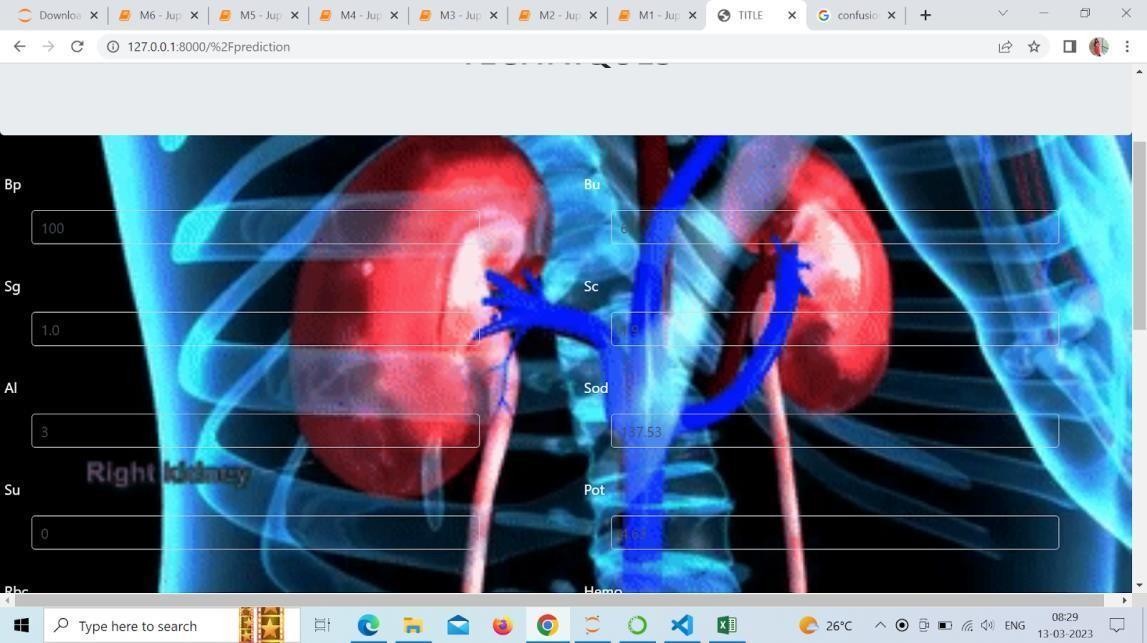
#### Fig 5.6 Stage0 Output



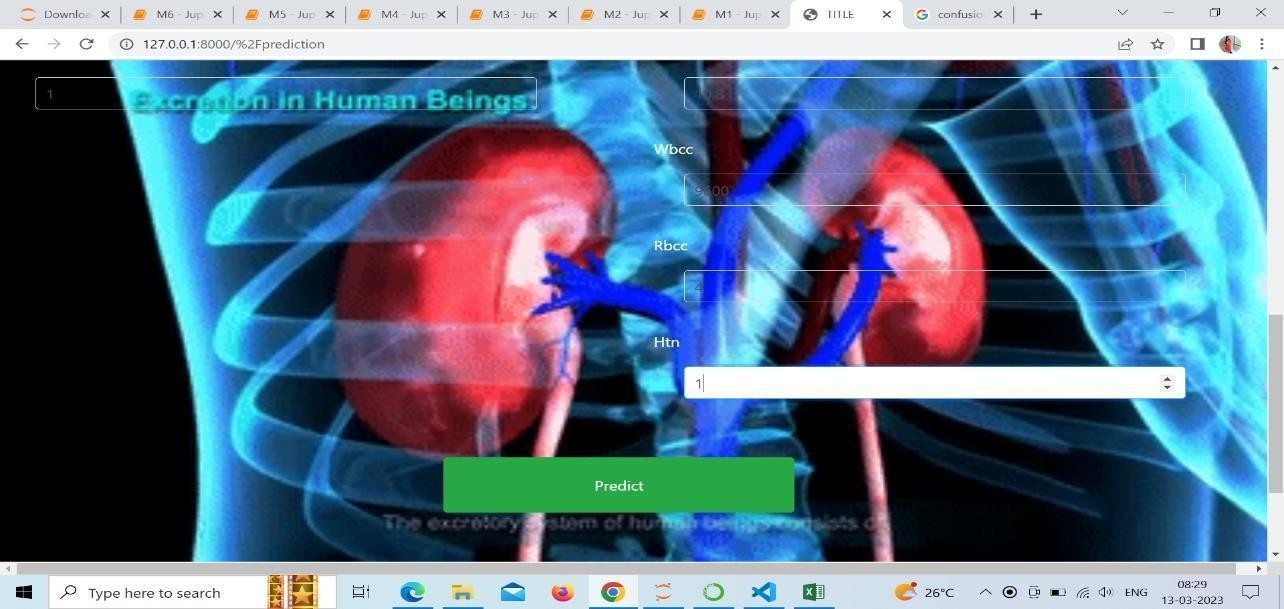
**Fig 5.7 Stage0 output**

#### 44

**STAGE 1:**

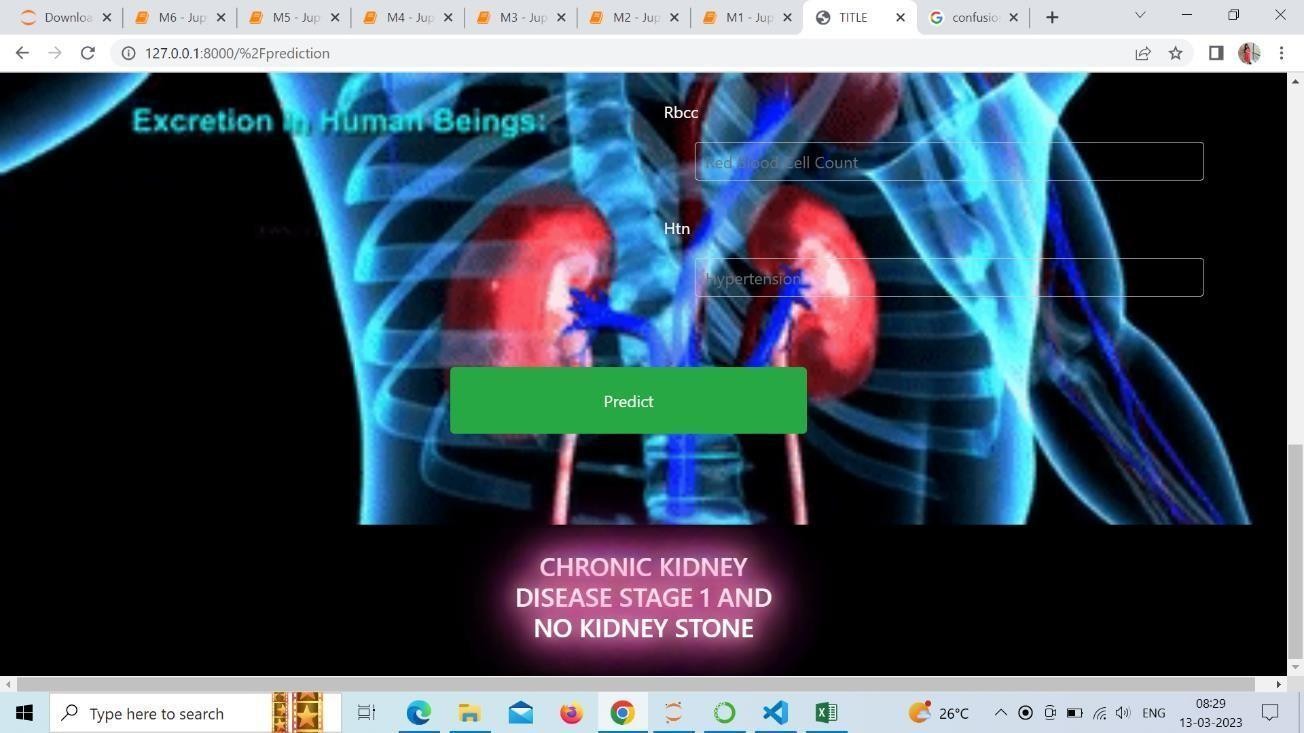


#### Fig 5.7 Stage1 output



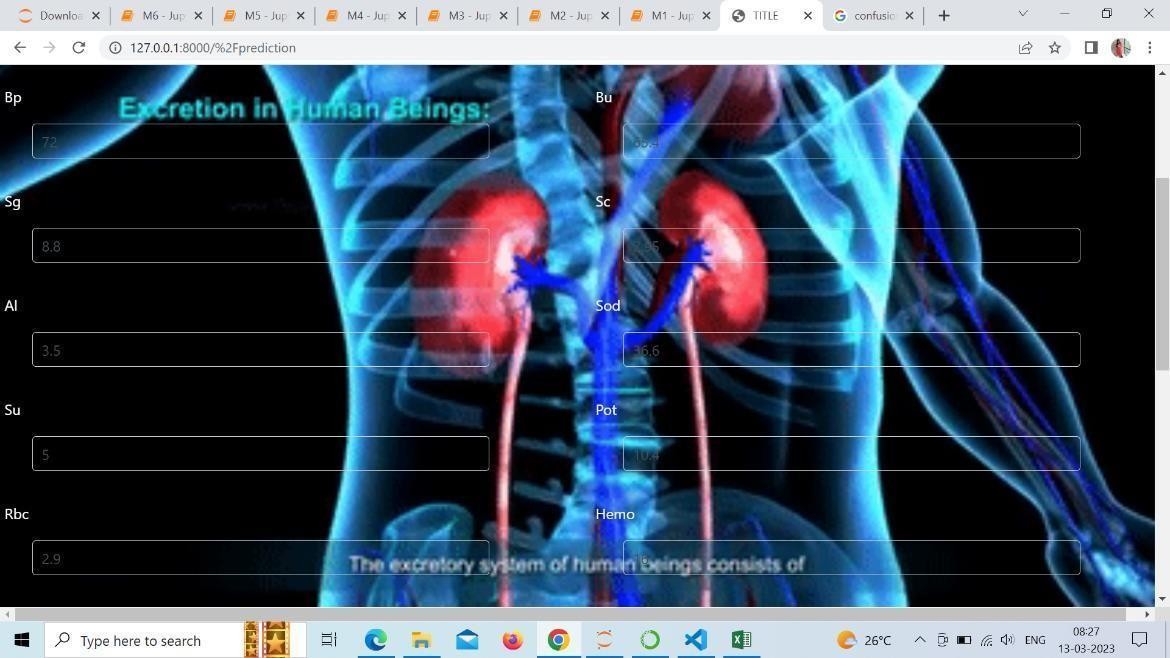
**Fig 5.8 Output**

#### 45

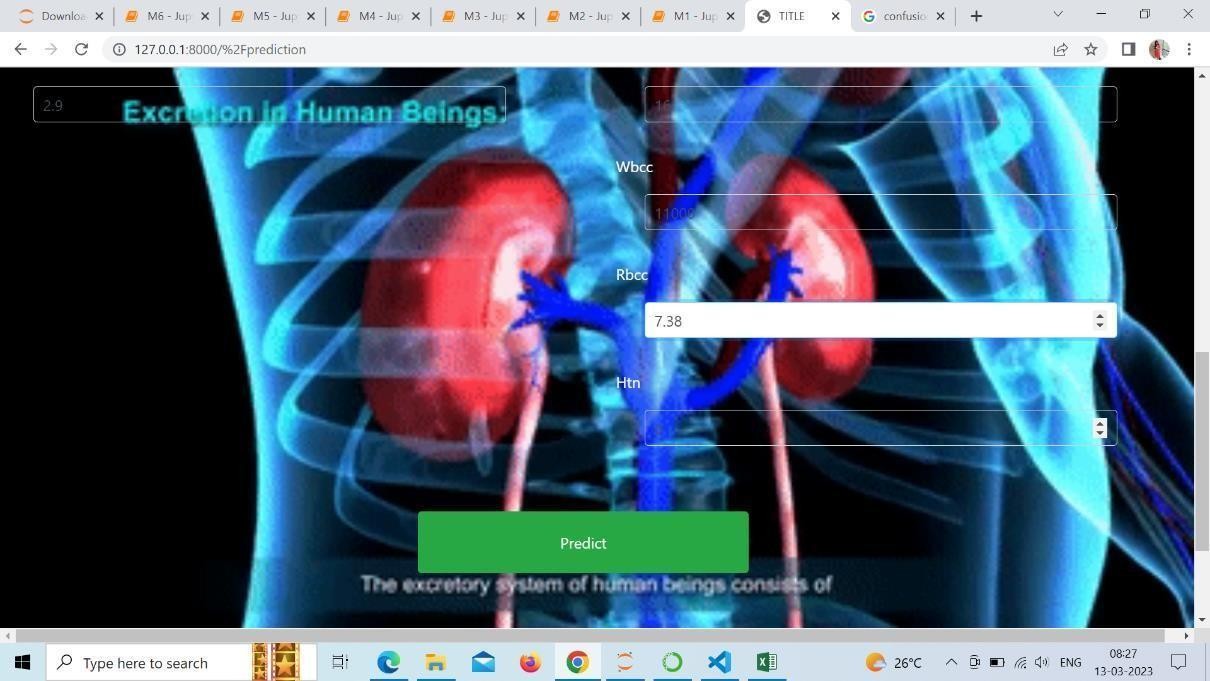


**STAGE 2:**

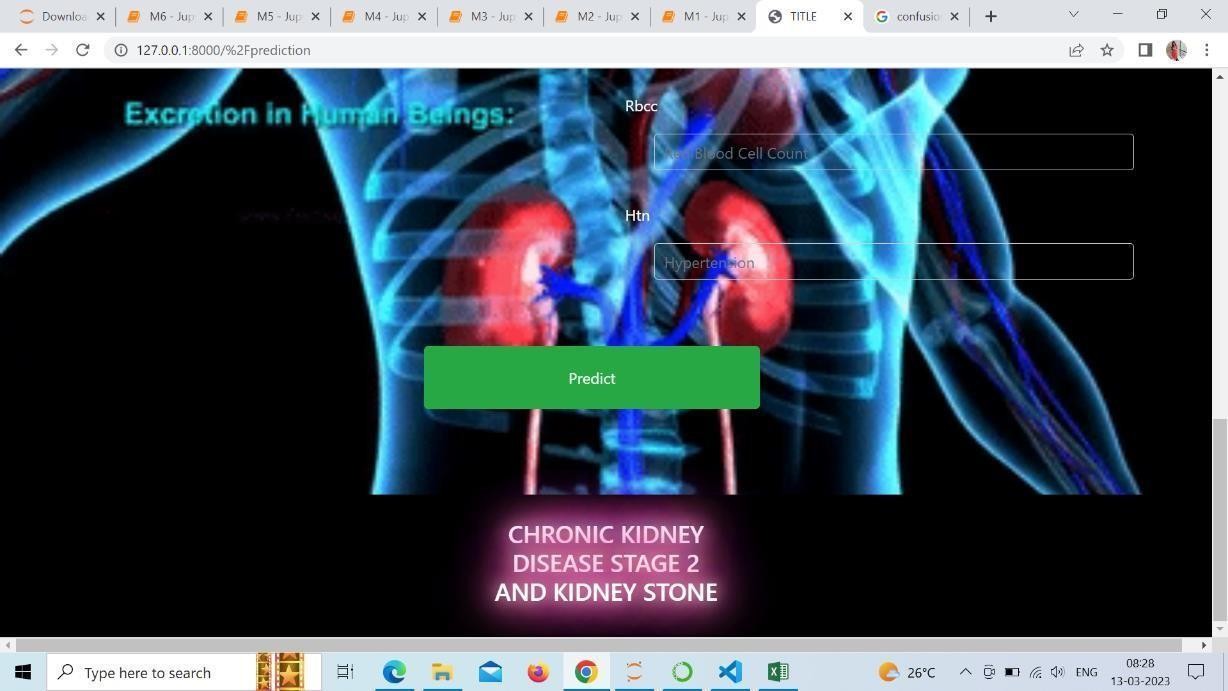
#### Fig 5.9 Output



**46**



#### Fig 5.10 Stage2 Output



**Fig 5.11 Output**

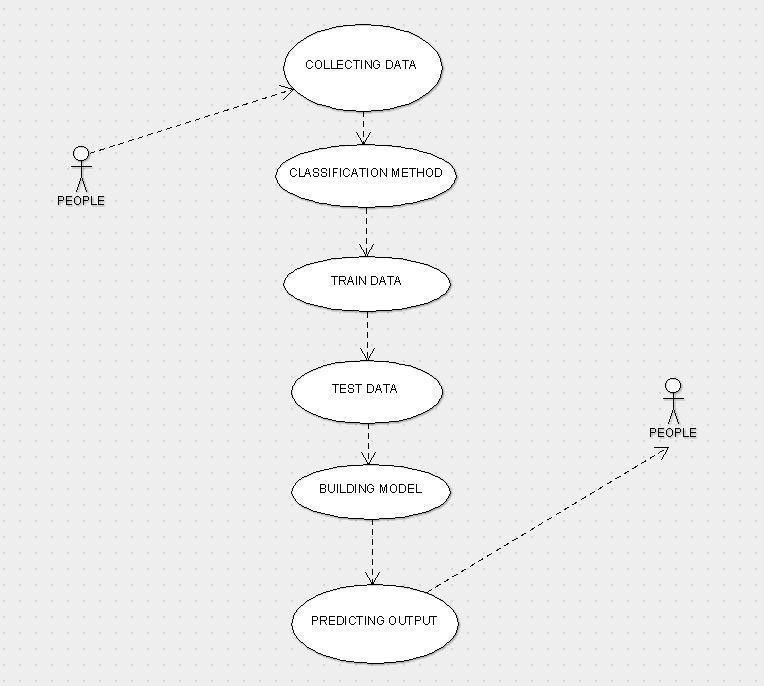
#### 47

**CHAPTER 6**

#### UML DIAGRAMS AND TESTING

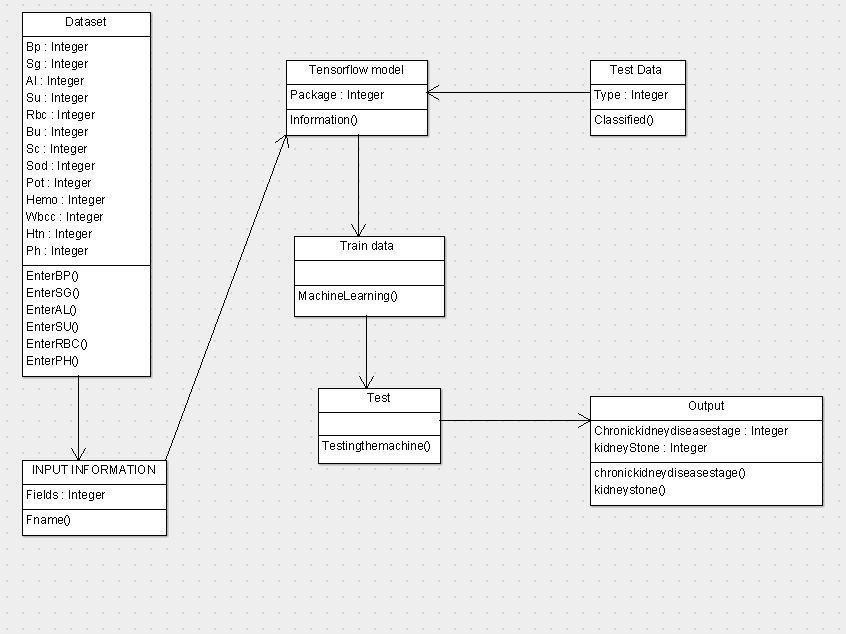
#### UML DIAGRAMS

#### USE CASE DIAGRAM



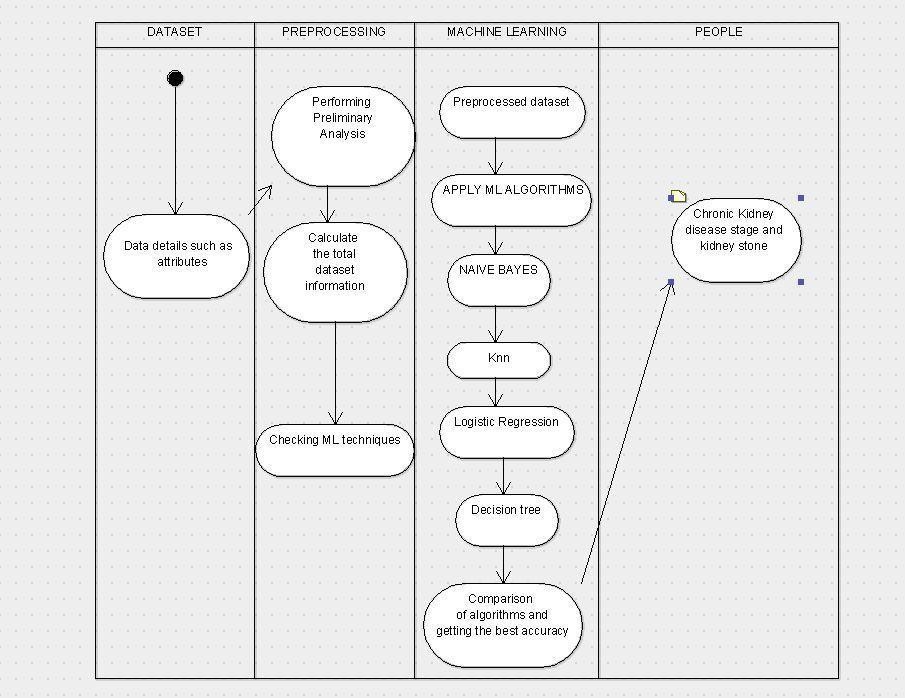
**Fig 6.1 Use case diagram**

#### CLASS DIAGRAM



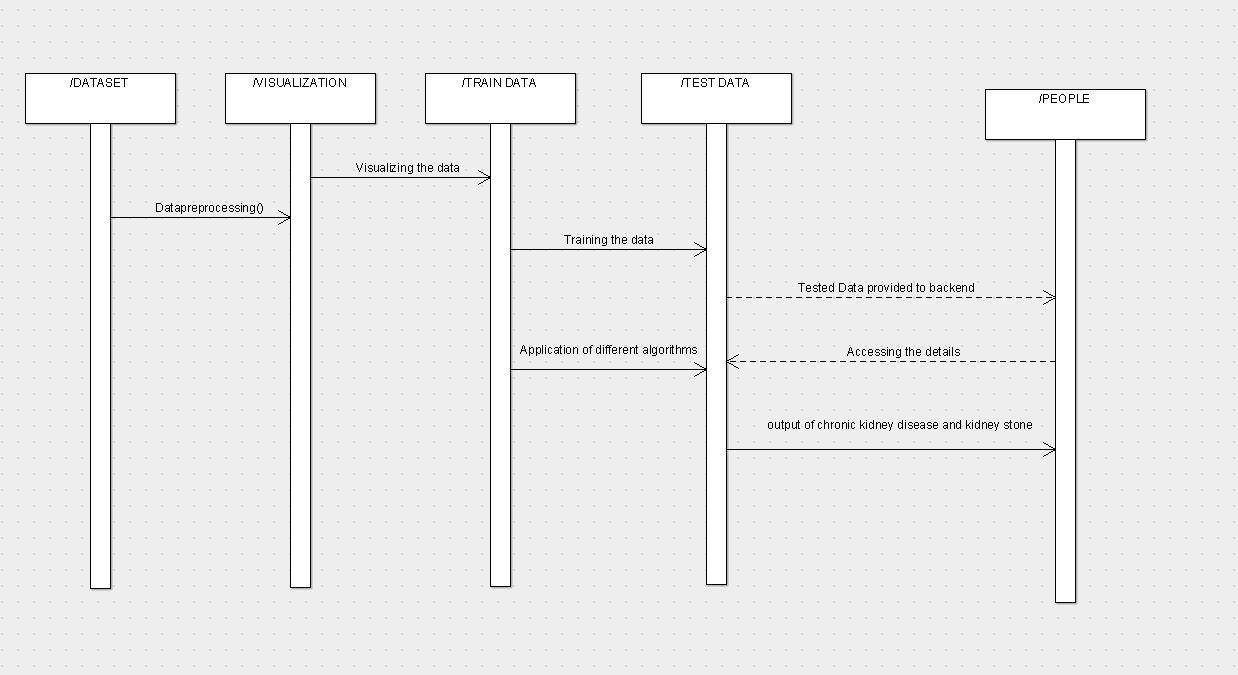
**Fig 6.2 Class diagram**

#### ACTIVITY DIAGRAM



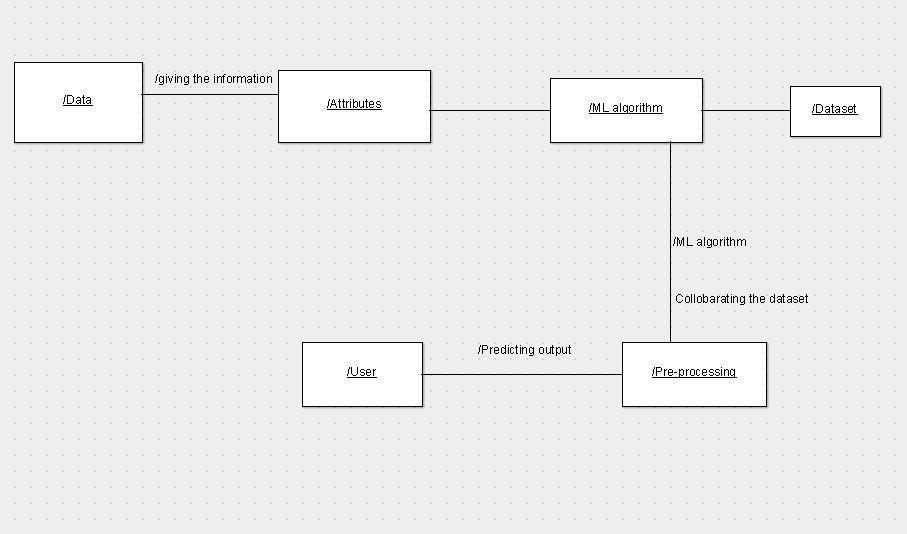
**Fig 6.3 Activity diagram**

#### SEQUENCE DIAGRAM



**Fig 6.4 Sequence Diagram**

#### COLLABORATION DIAGRAM



**Fig 6.5 Collaboration Diagram**

#### TESTING

* + 1. **UNIT TESTING**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

#### SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The Goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walk through. Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

* + 1. Static analysis is used to investigate the structural properties of the Source code.
    2. Dynamic testing is used to investigate the behavior of the source code by executing the program on the test

#### 51

* 1. **TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **MODULE** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **RESULT** |
| TC1 | Collecting the data | Dataset | Attribute segregation | PROCESSED DATA | PASS |
| TC2 | DETECTING THE STAGE | Attribute values like BP, SOD,POT,  PH,etc | STAGE OF KIDNEY DISEASE AND PRESENCE OF KIDNEY STONE | STAGE 0,1  AND 2  KIDNEY STONE PRESENT OR NOT | PASS |
| TC3 | ALGORITHM IMPLEMENTATION | CLEAN DATA AND ATTRIBUTE VALUES | ACCURACY AND AVERAGE | BEST ACCURACY AND AVERAGE | PASS |
| TC4 | WEBPAGE | VALUES FOR ATTRIBUTES PRESENT | RESULT SATGES AND KIDNEY STONE | STAGE 0,1  AND 2  KIDNEY STONE PRESENT OR NOT | PASS |

#### CHAPTER-7

**CONCLUSION AND FUTURE ENHANCEMENT**

#### CONCLUSION

Data preparation and processing, missing value analysis, exploratory analysis, and model construction and evaluation came first in the analytical process. The highest accuracy score on the public test set will be discovered. This software can assist in predicting kidney stones and chronic renal illness.

#### FUTURE ENHANCEMENT

* + 1. Hospitals aspire to automate (in real time) the process of excluding diseased people from eligibility.
    2. To automate this procedure by displaying the prediction outcome in a desktop or web application.
    3. To streamline the work that has to be done in an AI setting.

**54**

**APPENDICES**

**PAPER:**

56