

CHRONIC KIDNEY DISEASE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the
award of
Bachelor of Technology degree in Information Technology
By

CHARULATHA K (40120020)

KAVITHA V R (40120044)

**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)
Category-1 University by UGC**

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

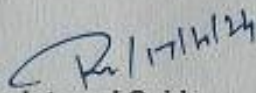
Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

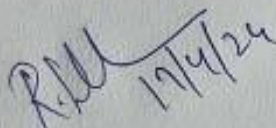
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CHARULATHA K (40120020), KAVITHA V R (40120044)** who carried out the Project phase-2 entitled "**CHRONIC KIDNEY DISEASE PREDICTION USING MACHINE LEARNING**" under my supervision from November 2023 to April 2024


Internal Guide

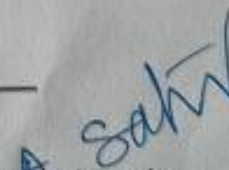
Dr. RAJASEKAR. P, M.E., Ph.D.,


Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 17/04/2024


Internal Examiner


External Examiner

DECLARATION

I, CHARULATHA K (40120020), hereby declare that the Project Phase- 2 Report entitled "CHRONIC KIDNEY DISEASE PREDICTION USING MACHINE LEARNING" done by me under the guidance of Dr. RAJASEKAR. P, M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in **Information Technology**.

DATE: 17/04/2024

PLACE: CHENNAI

Charulatha.K

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing **Dr.R.Subhashini,M.E., Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Rajasekar.P,M.E., Ph.D.**,for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

The term “chronic kidney disease” means lasting damage to the kidneys that can get worse over time. If the damage is very bad, your kidneys may stop working. This is called kidney failure, or end-stage kidney disease (ESRD). Kidney disease patients have the potential to get into the chronic phase and chronic kidney disease (CKD) is a decrease in kidney function gradually. So, doctors can diagnose kidney disease patients. So, predicting whether patients with kidney disease have entered a phase of chronic kidney disease or not by showing the best accuracy result of comparing supervised classification machine learning algorithms. The aim is to investigate machine learning-based techniques for Deforesting by predicting results in the best accuracy. The analysis of the dataset by supervised machine learning technique (SMLT) to capture several information like, variable identification, univariate analysis, bi-variate and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preparation, and data visualization will be done on the entire given dataset. Additionally, to compare and discuss the performance of various machine learning algorithms from the given hospital dataset with an evaluation classification report, identify the confusion matrix and to categorizing data from priority and the result shows that the effectiveness of the proposed machine learning algorithm technique can be compared with the best accuracy with precision, Recall and F1 Score.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF SYMBOLS	x
1	INTRODUCTION	1
	1.1 Data Science	1
	1.2 Data Scientist	1
	1.3 Artificial Intelligence	2
	1.4 Natural Language Processing (NLP)	5
2	LITERATURE SURVEY	6
	2.1 Review of Literature Survey	
3	REQUIREMENT ANALYSIS	10
	3.1 Workflow Diagram	11
	3.2 Use Case Diagram	12
	3.3 Class Diagram	13
	3.4 Activity Diagram	14
	3.4 Sequence Diagram	15
	3.5 Entity Relationship Diagram (ERP)	16
	3.6 Collaboration Diagram	17
4	DESCRIPTION OF PROPOSED SYSTEM	18
	4.1 Advantages	19
5	ALGORITHM AND IMPLEMENTATION	20
	5.1 Data Pre-Processing	20
	5.2 Data Validation / Cleaning / Preparing Process	21
	5.3 Exploration Data Analysis of Visualization	22

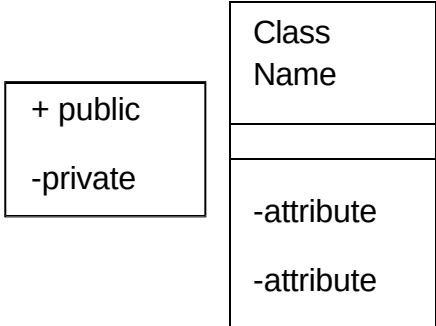
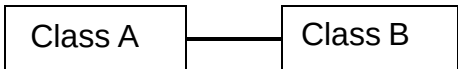
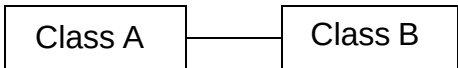
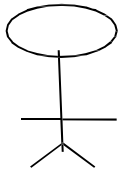
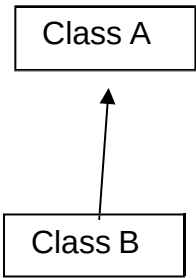
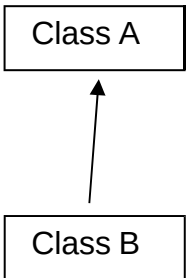
	5.4 Comparing Algorithm with Prediction In the Form of Best Accuracy Result	23
	5.5 Prediction Result by Accuracy	25
	5.6 Algorithm Explanation	27
	5.7 Used Python Packages	27
	5.8 Logistic Regression	28
	5.9 Support Vector Machine	28
	5.10 Gaussian Naïve Bayes	30
	5.11 Random Forest Classifier	31
	5.12 Deploying the Model in Django Framework and Predicting Output	32
	5.13 Django	32
	5.14 Module Diagram	34
6	RESULTS AND DISCUSSION	35
	6.1 Presentation of Findings	35
	6.2 Data Analysis	35
	6.3 Prediction Analysis	36
	6.4 Future Work	37
7	CONCLUSION	38
	REFERENCES	39
	APPENDIX	40
	A.SOURCE CODE	40
	B.SCREENSHOTS	42
	C.RESEARCH PAPER	46

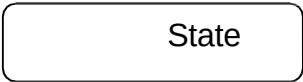
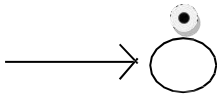
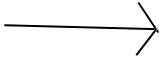
LIST OF FIGURES

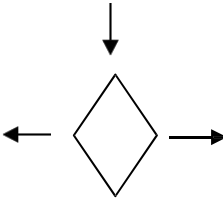
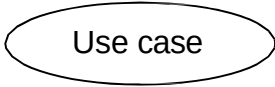
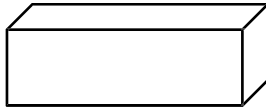
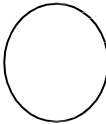
FIGURE NO	FIGURE NAME	PAGE NO
3.1	System architecture	10
3.2	Workflow diagram	11
3.3	Use Case diagram	12
3.4	Class diagram	13
3.5	Activity diagram	14
3.6	Sequence diagram	15
3.7	ER diagram	16
3.8	Collaboration Diagram	17
4.1	Module Diagram for Data Pre-processing	20
4.2	Module Diagram for Data Visualization	21
5.1	Module diagram for implementing Logistic regression	26
5.2	Module Diagram for implementing SVM Algorithm	27
5.3	Module diagram for implementing Gaussian Naïve Bayes Algorithm	29
5.4	Module diagram for implementing Random Forest Classifier Algorithm	30
6.1	Module diagram for disease prediction	33
6.2	Architecture of proposed system	34
8.1	Dataset Used	35
8.2	Prediction of Chronic Kidney Disease	36
8.3	Results for Prediction	36

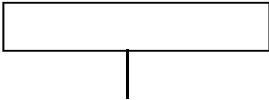
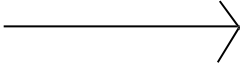
B.1	Data Pre-Processing	42
B.2	Data Visualization	42
B.2.1	Data Visualization 1	43
B.2.2	Data Visualization 2	43
B.3	Logistic Regression	44
B.4	Support Vector Machine	44
B.5	Random Forest Algorithm	45
B.6	Implementation of Views.py	45

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association	NAME  	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation	 	Interaction between the system and external environment

5.	<i>Relation</i> (uses)	uses	Used for additional process communication.
6.	Relation (extends)	EXTENDS 	Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.

12.	Decision box		Represents decision making process from a constraint
13.	Use case		Interact ion between the system and external environment.
14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.

18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

CHAPTER 1

INTRODUCTION

1.1 DATA SCIENCE:

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

The term "data science" was first coined in 2008 by D.J. Patil, and Jeff Hammerbacher, the pioneer leads of data and analytics efforts at LinkedIn and Facebook. In less than a decade, it has become one of the hottest and most trending professions in the market.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which help us in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decisions.

1.2 DATA SCIENTIST:

Data scientists examine which questions need answering and where to find the related data. They have business acumen and analytical skills as well as the ability to mine, clean, and present data. Businesses use data scientists to source, manage, and analyze large amounts of unstructured data.

Required Skills for a Data Scientist:

- **Programming:** Python, SQL, Scala, Java, R, MATLAB.
- **Machine Learning:** Natural Language Processing, Classification, Clustering.
- **Data Visualization:** Tableau, SAS, D3.js, Python, Java, R libraries.
- **Big data platforms:** MongoDB, Oracle, Microsoft Azure, Cloudera.

1.3 ARTIFICIAL INTELLIGENCE:

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals. Leading AI textbooks define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers.

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

AI applications include advanced web search engines, recommendation systems (used by Youtube, Amazon and Netflix), Understanding human speech (such as Siri or Alexa), self-driving cars (e.g. Tesla), and competing at the highest level in strategic game systems (such as chess and Go), As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology.

Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding. AI research has tried and discarded many different approaches during its lifetime, including simulating the brain, modeling human problem solving, formal logic, large databases of knowledge and imitating animal behavior. In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly successful, helping to solve many challenging problems throughout industry and academia.

The various sub-fields of AI research are centered around particular goals and the use of particular tools. The traditional goals of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects. General intelligence (the ability to solve an arbitrary problem) is among the field's long-term goals. To solve these problems, AI researchers use versions of search and mathematical optimization, formal logic, artificial neural networks, and methods based on statistics, probability and economics. AI also draws upon computer science, psychology, linguistics, philosophy, and many other fields.

The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence. These issues have been explored by myth, fiction and philosophy since antiquity. Science fiction and futurology have also suggested that, with its enormous potential and power, AI may become an existential risk to humanity.

As the hype around AI has accelerated, vendors have been scrambling to promote how their products and services use AI. Often what they refer to as AI is simply one component of AI, such as machine learning. AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular.

In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a chatbot that is fed examples of text chats can learn to produce life like exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples.

AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

Learning processes. This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Reasoning processes. This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

Self-correction processes. This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.

AI is important because it can give enterprises insights into their operations that they may not have been aware of previously and because, in some cases, AI can perform tasks better than humans. Particularly when it comes to repetitive, detail-oriented tasks like analyzing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors.

Artificial neural networks and deep learning artificial intelligence technologies are quickly evolving, primarily because AI processes large amounts of data much faster and makes predictions more accurately than humanly possible.

1.4 NATURAL LANGUAGE PROCESSING (NLP):

Natural language processing (NLP) allows machines to read and understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straightforward applications of natural language processing include information retrieval, text mining, question answering and machine translation. Many current approaches use word co-occurrence frequencies to construct syntactic representations of text. "Keyword spotting" strategies for search are popular and scalable but dumb; a search query for "dog" might only match documents with the literal word "dog" and miss a document with the word "poodle". "Lexical affinity" strategies use the occurrence of words such as "accident" to assess the sentiment of a document. Modern statistical NLP approaches can combine all these strategies as well as others, and often achieve acceptable accuracy at the page or paragraph level. Beyond semantic NLP, the ultimate goal of "narrative" NLP is to embody a full understanding of commonsense reasoning. By 2019, transformer-based deep learning architectures could generate coherent text.

CHAPTER 2

LITERATURE SURVEY

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

2.1 REVIEW OF LITERATURE SURVEY

Title: Chronic Kidney Disease and Its Complications

Author: Robert Thomas, MD, Abbas Kanso, MD, John R. Sedor, MD

Year: 2008

Chronic kidney disease (CKD) is recognized as a major health problem affecting approximately 13% of the US population. Numbers of prevalent CKD patients will continue to rise, reflecting the growing elderly population and increasing numbers of patients with diabetes and hypertension. As numbers of CKD patients increase, primary care practitioners will be confronted with management of the complex medical problems unique to patients with chronic renal impairment. As well documented in the literature, the nephrologist rarely manages the medical needs of CKD patients until renal replacement therapy is required. In this article, we define CKD staging and discuss five complications associated with CKD: anemia,

hyperlipidemia, nutrition, osteodystrophy, and cardiovascular risk.

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

Author: Muhammad SalmanI, Amer Hayat KhanII, Azreen Syazril AdnanIII, Syed Azhar Syed SulaimanIV, Khalid HussainV, Naureen ShehzadiVI, Fauziah Jummaat

Year: 2015

Chronic kidney disease (CKD) is an escalating medical and socioeconomic problem worldwide. Information concerning the causes of CKD, which is a prerequisite for reducing the disease burden, is sparse in Malaysia. Therefore, this study aimed to evaluate the attributable causes of CKD in an adult population at a tertiary referral hospital. A total of 851 eligible cases were included. The patients' mean age was 61.18

± 13.37 years. CKD stage V was found in 333 cases (39.1%) whereas stages IV, IIIb, IIIa, and II were seen in 240 (28.2%), 186 (21.9%), 74 (8.7%) and 18 (2.1%), respectively. The percentage of CKD stage V patients receiving renal replacement therapy was 15.6%. The foremost attributable causes of CKD were diabetic nephropathy (DN) (44.9%), hypertension (HPT) (24.2%) and obstructive uropathy (9.2%). The difference in the prevalence of CKD due to DN, HPT and glomerulonephritis between patients ≤ 50 and > 50 years old was statistically significant. Our results suggest that DN and HPT are the major attributable causes of CKD among patients at a Malaysian tertiary-care hospital. Furthermore, the results draw attention to the possibility that greater emphasis on primary prevention of diabetes and hypertension will have a great impact on reduction of hospital admissions due to CKD in Malaysia.

Title: PATHOPHYSIOLOGY AND CLASSIFICATION OF KIDNEY DISEASES

Author: Mirjana Sabljari Matovinovic

Year: 2009

Chronic kidney disease (CKD) is far more prevalent worldwide than was previously assumed. It affects 10 - 15% of the adult population in the western countries, many

of whom require costly treatments or renal replacement therapy. According to the Third National Health and Nutrition Examination Survey and the National Kidney Foundation Kidney Disease report nearly 26 million persons in the USA fall into this category and another 20 millions are at an increased risk for CKD. Moreover, it has been recognized that CKD is a major risk factor for increased cardiovascular disease and death. This knowledge has been incorporated in the recent cardiologic guidelines as well as in the 2007 European Guidelines for the Management of Arterial Hypertension. At the same time, there is an increasing prevalence of diseases that predispose individuals to CKD, such as hypertension, diabetes, obesity and other, rendering the prevention and early detection of CKD a health-care priority in both developed and developing countries.

Title: Practical Approach to Detection and Management of Chronic Kidney Disease for the Primary Care Clinician

Author: Joseph A. Vassalotti, MD, Robert Centor, MD, Barbara J. Turner, MD, MSED, Raquel C. Greer, MD, MHS, Michael Choi, MD, Thomas D. Sequist, MD

Year: 2015

A panel of internists and nephrologists developed this practical approach for the Kidney Disease Outcomes Quality Initiative to guide assessment and care of chronic kidney disease (CKD) by primary care clinicians. Chronic kidney disease is defined as a glomerular filtration rate (GFR) $<60 \text{ mL/min/1.73 m}^2$ and/or markers of kidney damage for at least 3 months. In clinical practice the most common tests for CKD include GFR estimated from the serum creatinine concentration (eGFR) and albuminuria from the urinary albumin-to-creatinine ratio. Assessment of eGFR and albuminuria should be performed for persons with diabetes and/or hypertension but is not recommended for the general population. Management of CKD includes reducing the patient's risk of CKD progression and risk of associated complications, such as acute kidney injury and cardiovascular disease, anemia, and metabolic acidosis, as well as mineral and bone disorder. Prevention of CKD progression requires blood pressure $<140/90 \text{ mm Hg}$, use of angiotensin-converting enzyme inhibitors or angiotensin receptor blockers for patients with albuminuria and hypertension, hemoglobin A1c $\leq 7\%$ for patients with diabetes, and correction of CKD-associated metabolic acidosis. To reduce patient safety hazards from

medications, the level of eGFR should be considered when prescribing, and nephrotoxins should be avoided, such as nonsteroidal anti-inflammatory drugs.

Title: Prevalence of chronic kidney disease and its associated factors in Malaysia; findings from a nationwide population-based crosssectional study

Author: Thamil Arasu Saminathan, Lai Seong Hooi, Muhammad Fadhli Mohd Yusoff, Loke Meng Ong, Sunita Bavanandan, Wan Shakira Rodzlan Hasani, Esther Zhao Zhi Tan, Irene Wong, Halizah Mat Rifin, Tania Gayle Robert, Hasimah Ismail, Norazizah Ibrahim Wong, Ghazali Ahmad, Rashidah Ambak, Fatimah Othman, Hamizatul Akmal Abd Hamid and Tahir Aris

Year: 2020

Our study shows that the prevalence of CKD in Malaysia was 15.48% (95% CI: 12.30, 19.31) in 2018, an increase compared to the year 2011 when the prevalence of CKD was 9.07%. An estimated 3.85% had stage 1 CKD, 4.82% had stage 2 CKD, and

6.48% had stage 3 CKD, while 0.33% had stage 4–5 CKD. Hypertension (aOR 3.72), diabetes mellitus (aOR 3.32), increasing BMI (aOR 1.06), and increasing age (aOR 1.06) were significantly associated with CKD. Our study has shown that CKD has become one of the leading public health issues in Malaysia

Title: CHRONIC KIDNEY DISEASE PREDICTION USING MACHINE LEARNING

Author: Kavitha V R, Charulatha K

Year: 2024

Chronic kidney disease (CKD) is a prevalent and debilitating condition, demanding early detection and effective management to mitigate its progression and associated complications. Machine learning (ML) algorithms, particularly the Random Forest (RF) algorithm, offer a promising approach to enhance CKD prediction and patient risk stratification. By leveraging large and diverse datasets encompassing patient demographics, clinical parameters, and genetic markers, RF models can provide accurate predictions of CKD onset and progression. However, challenges such as model interpretability, data quality assurance, and seamless integration into clinical workflows need to be addressed to realize the full potential of ML in optimizing CKD care delivery and improving patient outcomes

CHAPTER 3

REQUIREMENT ANALYSIS

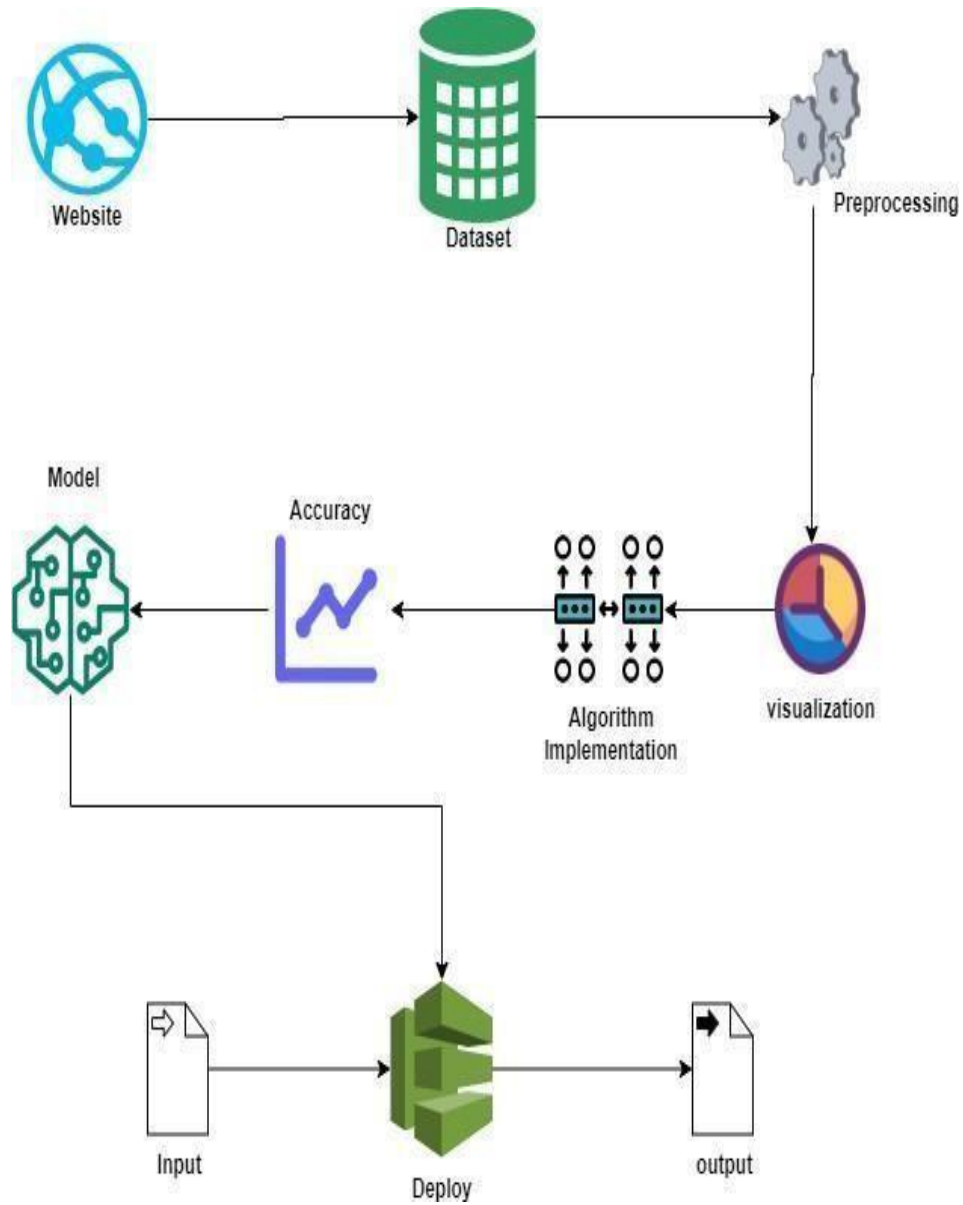


Fig 3.1: System Architecture

3.1 WORKFLOW DIAGRAM:

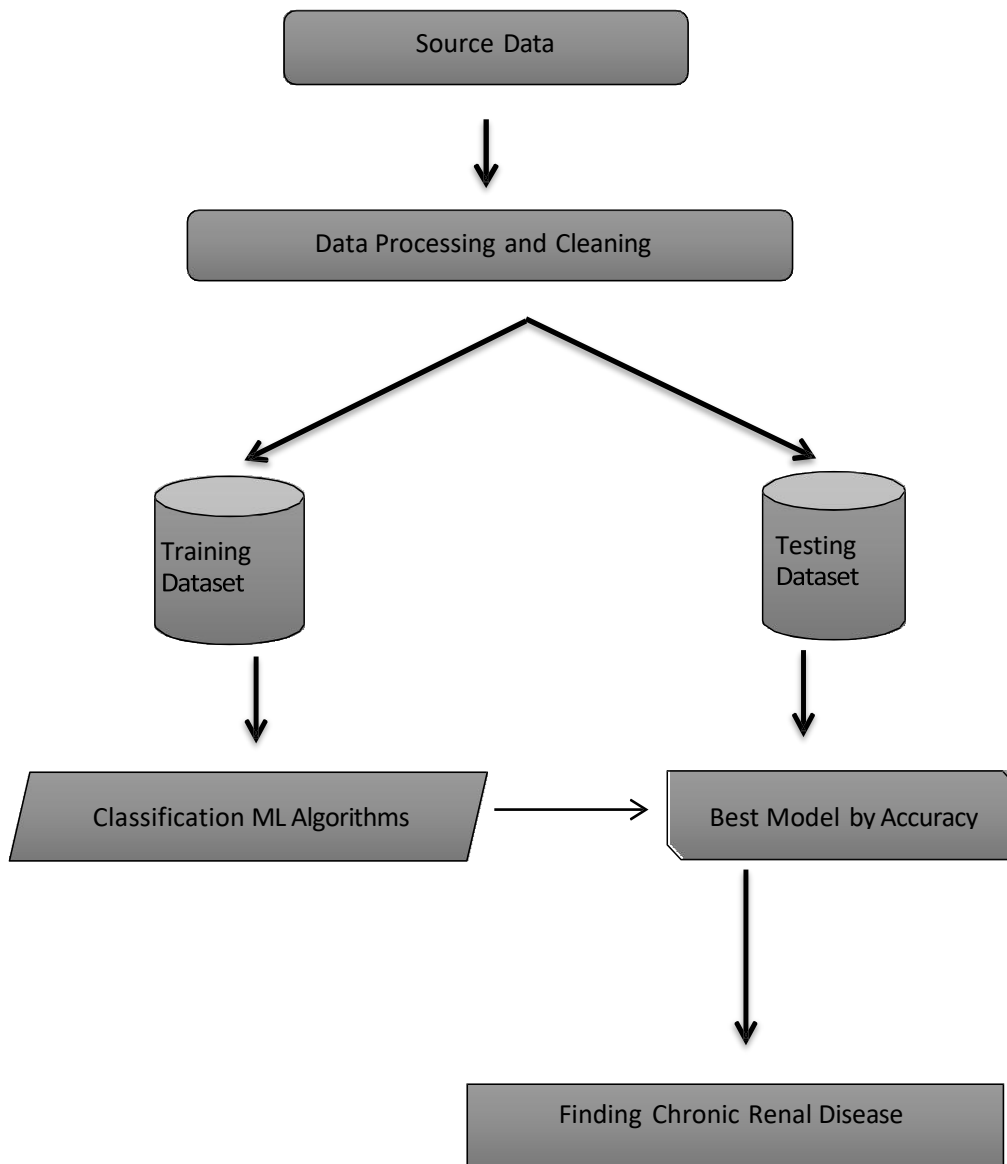


Fig 3.2: Workflow Diagram

3.2 USE CASE DIAGRAM:

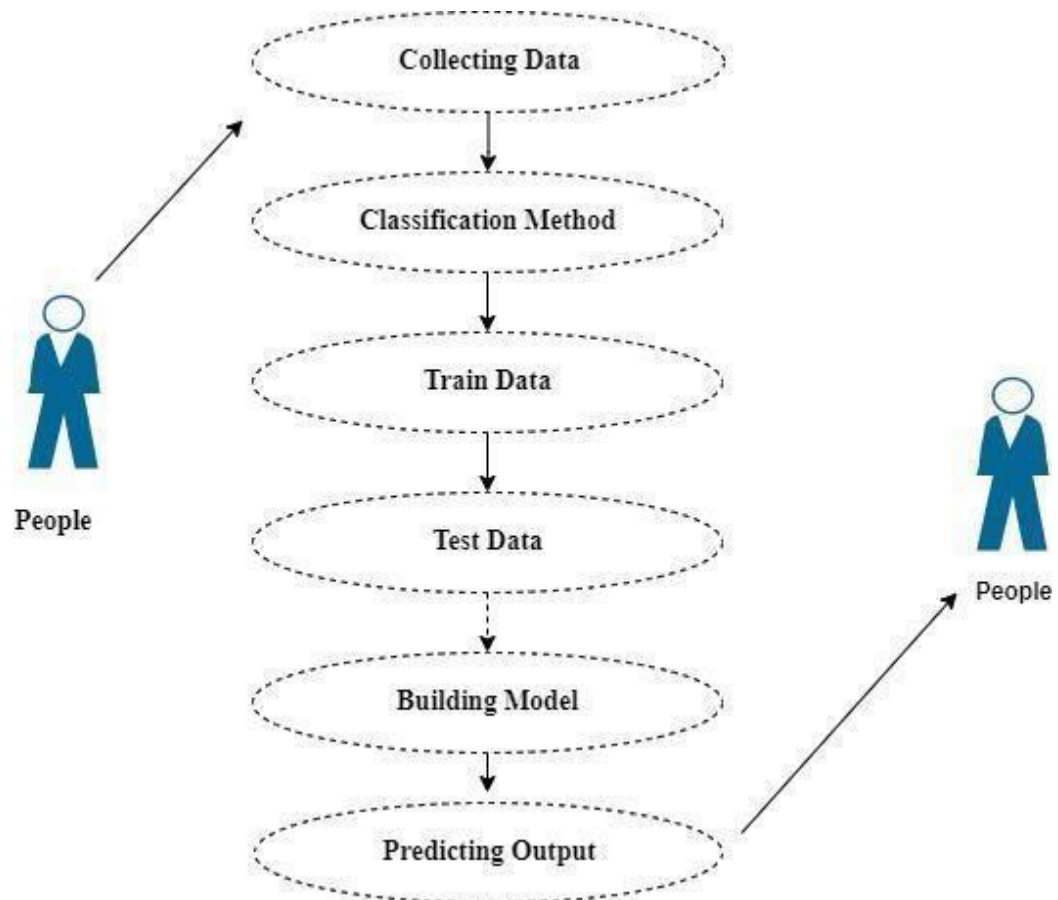


Fig 3.3: Use Case Diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

3.3 CLASS DIAGRAM:

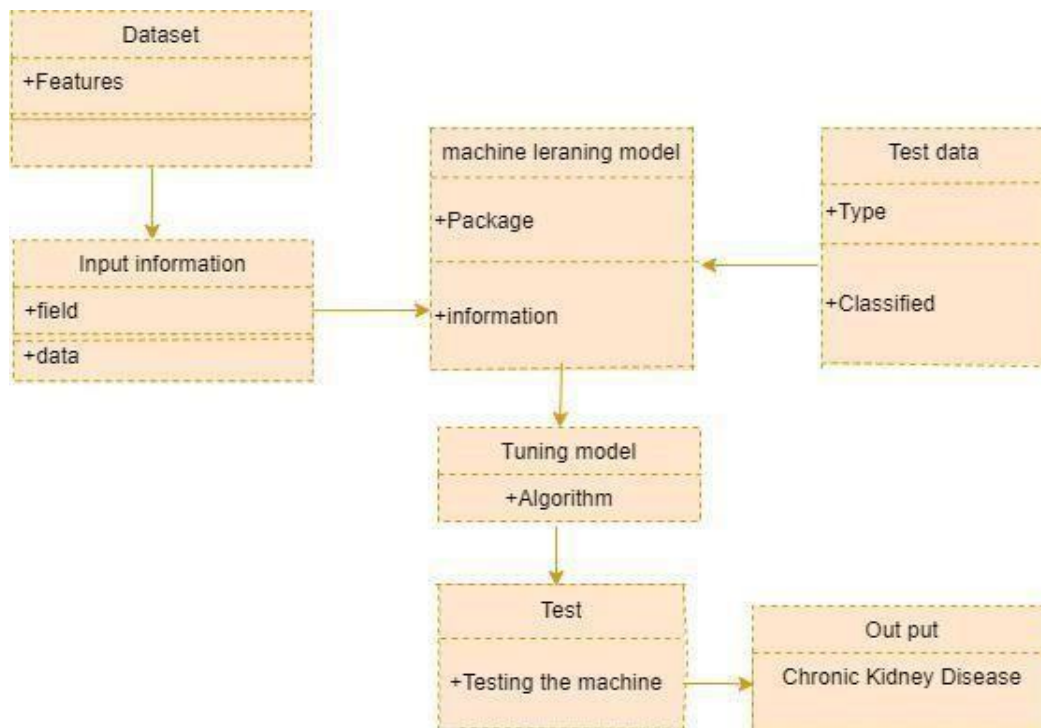


Fig 3.4: Class Diagram

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

3.4 ACTIVITY DIAGRAM:

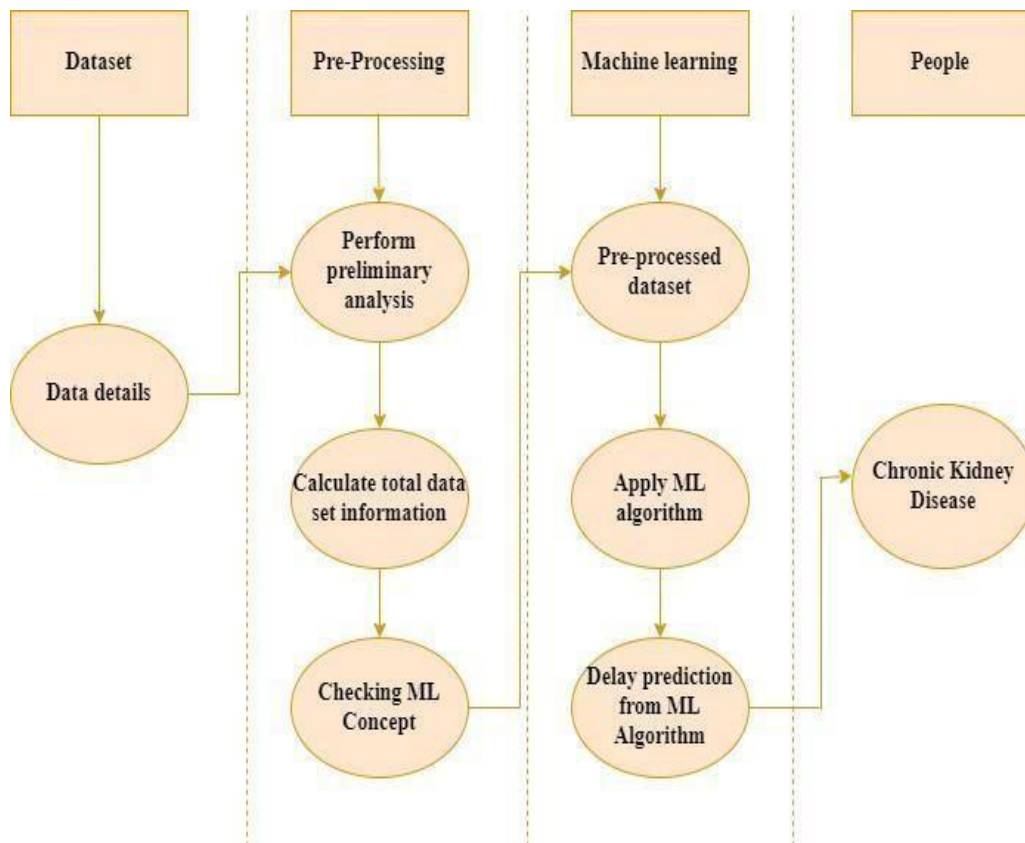


Fig 3.5: Activity Diagram

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams look like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

3.5 SEQUENCE DIAGRAM:

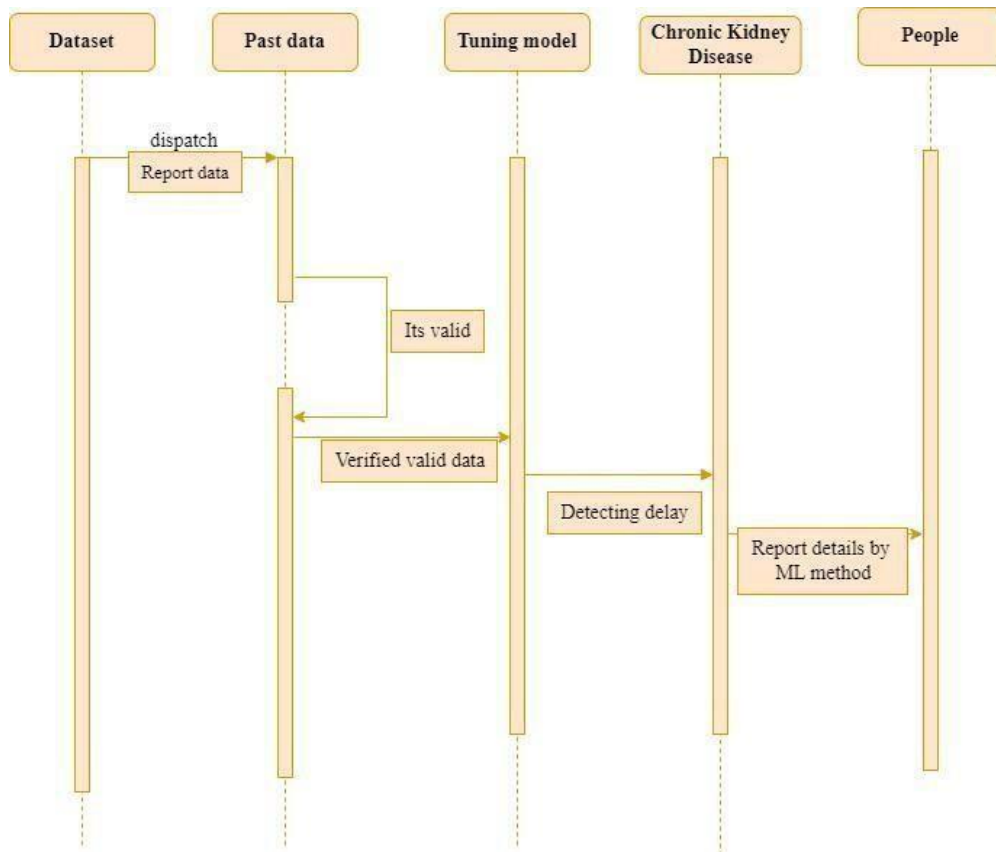


Fig 3.6: Sequence Diagram

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include activity diagramming, communication interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design- level models for modern business application development.

3.6 ENTITY RELATIONSHIP DIAGRAM (ERD):

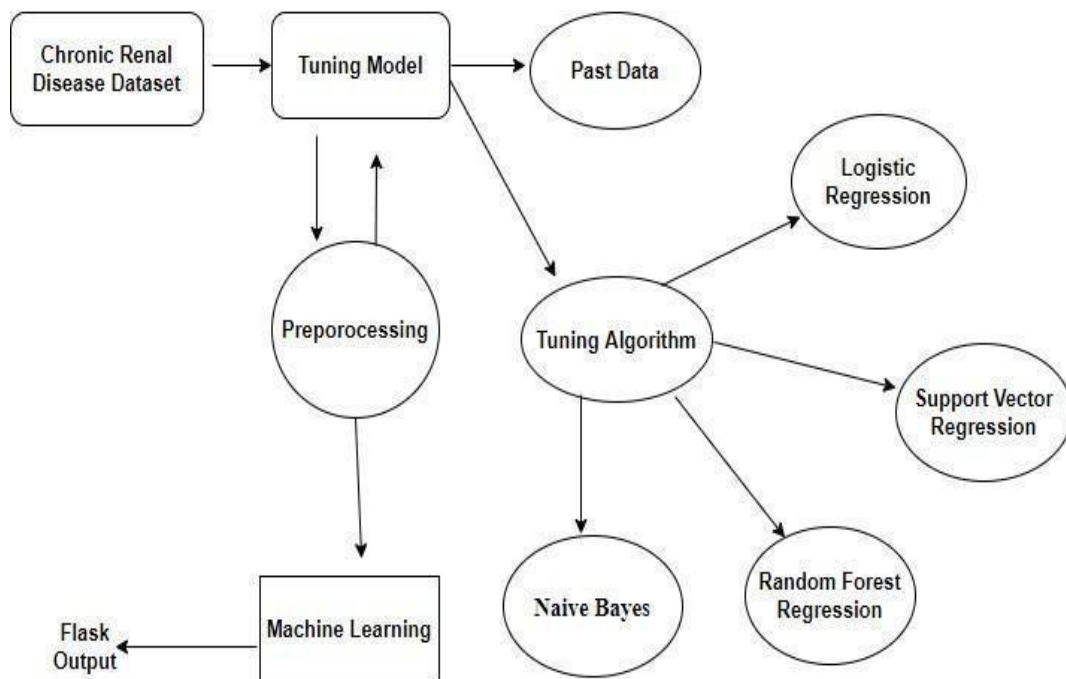


Fig 3.7: ER Diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

3.7 COLLABORATION DIAGRAM:

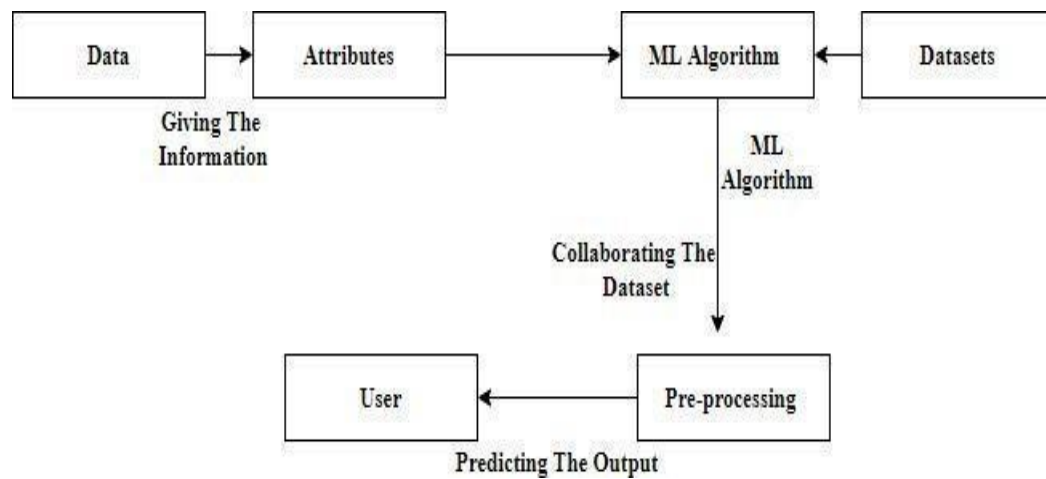


Fig 3.8: Collaboration Diagram

A collaboration diagram is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users (like doctor or patient) can benefit from this collaboration. A collaboration diagram often comes in the form of a visual chart that resembles a flow chart. It can show, at a glance, how a single piece of software complements other parts of a greater system.

CHAPTER 4

PROPOSED SYSTEM

The proposed method is to build a machine learning model for the classification of kidney disease. The process carries from data collection where past data related to kidney disease are collected. Data mining is a commonly used technique for processing enormous data in the healthcare domain. The kidney disease if found before proper treatment can save lives. Machine learning is now applied and mostly used in health care where it reduces the manual effort and a better model makes error less which leads in saving the life. The data analysis is done on the dataset proper variable identification is done that is both the dependent variables and independent variables are found. Then proper machine learning algorithms are applied to the dataset where the pattern of data is learnt. After applying different algorithms a better algorithm is used for the prediction of the outcome

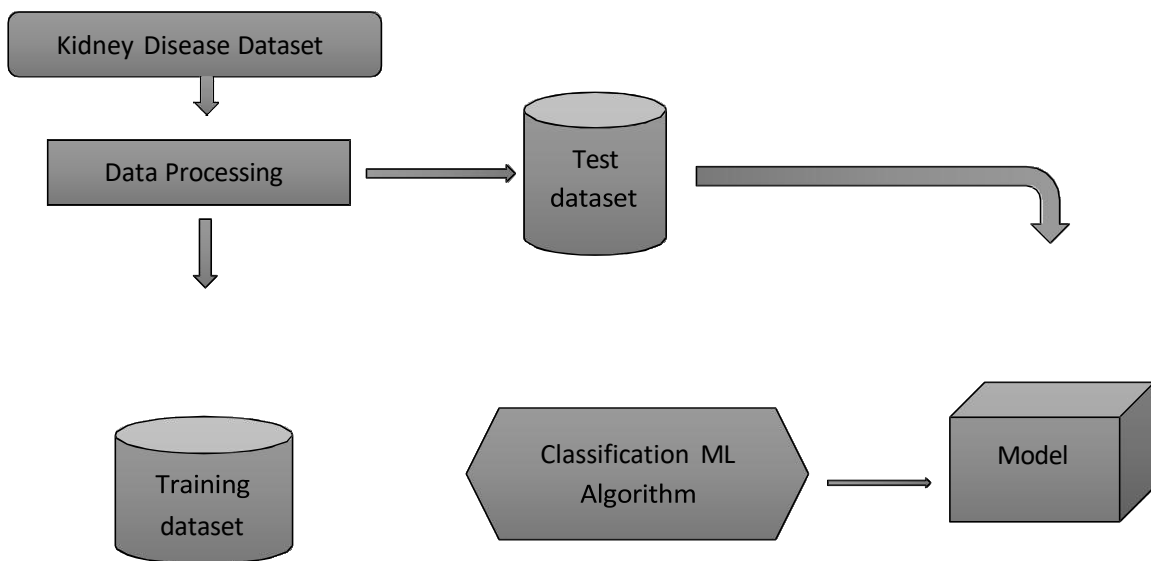


Fig 4.1 Proposed System

4.1 ADVANTAGES:

- We are using Machine Learning Technique to predict kidney disease.
- Algorithms are compared and the best model is evaluated for better prediction.
- Performance metrics of different algorithms are compared and a better prediction is done.

CHAPTER 5

IMPLEMENTATION

5.1 DATA PRE-PROCESSING:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine- tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.
- Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:
- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

5.2 DATA VALIDATION / CLEANING / PREPARING PROCESS:

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

5.2.1 Module Diagram

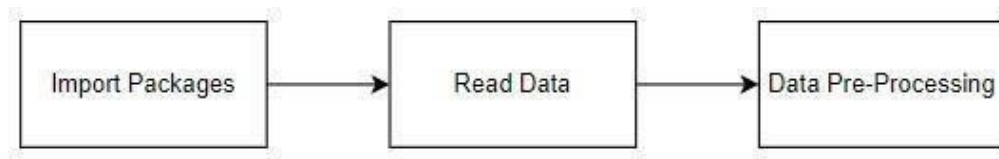


Fig 5.1: Module Diagram for Data Pre-processing

Given Input Expected Output

input: data

output: removing noisy data

5.3 EXPLORATION DATA ANALYSIS OF VISUALIZATION:

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

5.3.1 Module Diagram



Fig 5.2: Module Diagram for Data Visualization

Given Input Expected Output

input: data

output: visualized data

5.4 COMPARING ALGORITHM WITH PREDICTION IN THE FORM OF THE BEST ACCURACY RESULT:

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite

of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

- Logistic Regression
- Support |Vector Machine
- Gaussian Naïve Bayes
- Random Forest Classifier

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

5.5 PREDICTION RESULT BY ACCURACY:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy. True Positive Rate (TPR) = $TP / (TP + FN)$

False Positive rate (FPR) = $FP / (FP + TN)$

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. Precision = $TP / (TP + FP)$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = $TP / (TP + FN)$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

General Formula:

$$F\text{- Measure} = 2TP / (2TP + FP + FN)$$

F1-Score Formula:

$$F1 \text{ Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class is no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

5.6 ALGORITHM EXPLANATION:

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

5.7 USED PYTHON PACKAGES:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train test split, Decision Tree Classifier or Logistic Regression and accuracy score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

5.8 LOGISTIC REGRESSION:

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts $P(Y=1)$ as a function of X . Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

5.8.1 Module Diagram

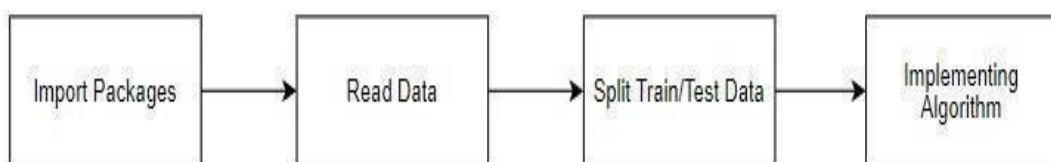


Fig 5.1: Module diagram for implementing Logistic regression

Given Input Expected Output

input: data

output: getting accuracy

5.9 SUPPORT VECTOR MACHINE:

A classifier that categorizes the data set by setting an optimal hyper plane between data. I chose this classifier as it is incredibly versatile in the number of different kernelling functions that can be applied and this model can yield a high predictability rate. Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high- performing algorithm with little tuning.

- How to disentangle the many names used to refer to support vector machines.
- The representation used by SVM when the model is actually stored on disk.
- How a learned SVM model representation can be used to make predictions for new data.
- How to learn an SVM model from training data.
- How to best prepare your data for the SVM algorithm.
- Where you might look to get more information on SVM.

5.9.1 Module Diagram

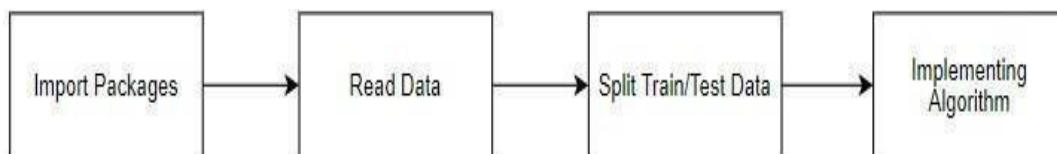


Fig 5.2: Module Diagram for implementing SVM Algorithm

Given Input Expected Output

input: data

output: getting accuracy

5.10 GAUSSIAN NAÏVE BAYES:

- The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically.
- Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
- The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.
- Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.
- Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location.
- Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.

5.10.1 Module Diagram

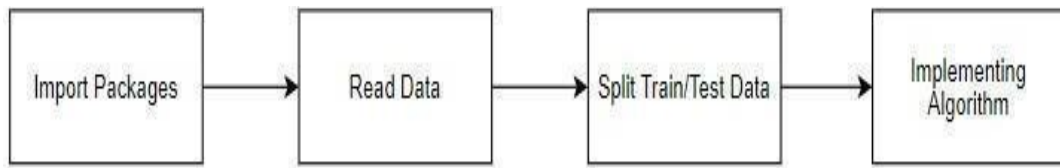


Fig 5.3: Module diagram for implementing Gaussian Naïve Bayes Algorithm

Given Input Expected Output

input: data

output: getting accuracy

5.11 RANDOM FOREST ALGORITHM

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

5.11.1 Module Diagram

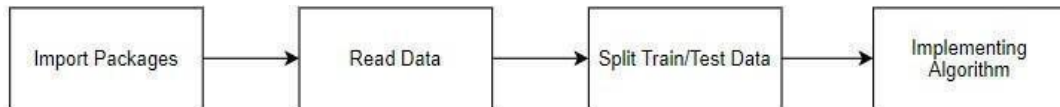


Fig 5.4: Module diagram for implementing Random Forest Classifier Algorithm

Given Input Expected Output

input: data

output: getting accuracy

5.12 DEPLOYING THE MODEL IN DJANGO FRAMEWORK AND PREDICTING OUTPUT

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 output whether the given image is CKD / Not CKD.

5.13 DJANGO

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

Complete

Django follows the "Batteries included" philosophy and provides almost everything

developers might want to do "out of the box". Because everything you need is part of the "product", it all works seamlessly together, follows consistent design principles extensive and up-to-date documentation.

Versatile

Django can be (and has been) used to build almost any type of website - from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is built with Django!

Secure

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.

Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and click jacking (see Website security for more details of such attacks).

Scalable

Django uses a component-based "shared-nothing" architecture (each part of the

architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, databaseservers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, toname just two).

Maintainable

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

Portable

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

5.14 MODULE DIAGRAM

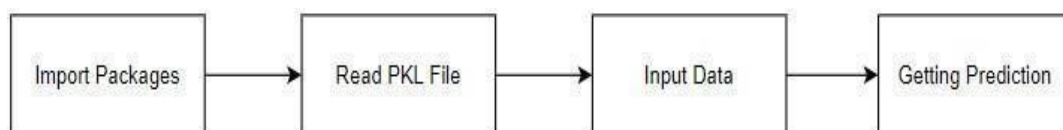


Fig 5.5 : Module diagram for disease prediction Given Input

Expected Output

input: data

output: getting prediction

CHAPTER 6

RESULTS AND DISCUSSION

6.1 PRESENTATION OF FINDINGS

This dataset contains 400 records of features. It is classified into 2 classes CKD and not a CKD with diastolic blood pressure, specific gravity, albumin, sugars, RBC cells, blood urea, serum creatinine, sodium, potassium, hemoglobin level in blood, WBC count in blood, RBC count in blood and hypertension.

6.2 DATA ANALYSIS

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn	Class		
80	1.02	1	0	1	36	1.2	137.53	4.63	15.4	7800	5.2	1	1		
50	1.02	4	0	1	18	0.8	137.53	4.63	11.3	6000	4.71	0	1		
80	1.01	2	3	1	53	1.8	137.53	4.63	9.6	7500	4.71	0	1		
70	1.005	4	0	1	56	3.8	111	2.5	11.2	6700	3.9	1	1		
80	1.01	2	0	1	26	1.4	137.53	4.63	11.6	7300	4.6	0	1		
90	1.015	3	0	1	25	1.1	142	3.2	12.2	7800	4.4	1	1		
70	1.01	0	0	1	54	24	104	4	12.4	8406	4.71	0	1		
76	1.015	2	4	1	31	1.1	137.53	4.63	12.4	6900	5	0	1		
100	1.015	3	0	1	60	1.9	137.53	4.63	10.8	9600	4	1	1		
90	1.02	2	0	0	107	7.2	114	3.7	9.5	12100	3.7	1	1		
60	1.01	2	4	1	55	4	137.53	4.63	9.4	8406	4.71	1	1		
70	1.01	3	0	0	60	2.7	131	4.2	10.8	4500	3.8	1	1		
70	1.015	3	1	1	72	2.1	138	5.8	9.7	12200	3.4	1	1		
70	1.02	1	0	1	86	4.6	135	3.4	9.8	8406	4.71	1	1		
80	1.01	3	2	1	90	4.1	130	6.4	5.6	11000	2.6	1	1		
80	1.015	3	0	1	162	9.6	141	4.9	7.6	3800	2.8	1	1		
70	1.015	2	0	1	46	2.2	138	4.1	12.6	8406	4.71	0	1		
80	1.02	1	0	1	87	5.2	139	3.7	12.1	8406	4.71	1	1		
100	1.025	0	3	1	27	1.3	135	4.3	12.7	11400	4.3	1	1		
60	1.015	1	0	1	31	1.6	137.53	4.63	10.3	5300	3.7	1	1		
80	1.015	2	0	0	148	3.9	135	5.2	7.7	9200	3.2	1	1		
90	1.02	1	0	1	180	76	4.5	4.63	10.9	6200	3.6	1	1		
80	1.025	4	0	1	163	7.7	136	3.8	9.8	6900	3.4	1	1		
70	1.01	0	0	1	57	3.07	137.53	4.63	12.53	8406	4.71	0	1		
100	1.015	4	0	1	50	1.4	129	4	11.1	8300	4.6	1	1		
60	1.025	0	0	1	75	1.9	141	5.2	9.9	8400	3.7	1	1		
80	1.015	0	0	1	45	2.4	140	3.4	11.6	10300	4	1	1		
70	1.01	3	4	1	87	2.7	130	4	12.5	9600	4.1	1	1		
70	1.02	1	3	1	31	1.4	137.53	4.63	12.53	8406	4.71	0	1		

Fig 6.1: Dataset Used

This dataset contains 400 records of features.

6.3 PREDICTION ANALYSIS

The screenshot shows a web application interface for predicting Chronic Kidney Disease. It features a grid of input fields for various medical parameters, a large red banner with the text "Chronic kidney disease", and a background image of human kidneys.

Parameter	Value
Blood Pressure	0.06
Specific Gravity	0.02
Albumin	0.01
Sugar	0.01
Red Blood Cells	0.01
Blood Urea	0.01
Sodium	0.02
Potassium	0.03
Hemoglobin	0.04
White Blood Cell Count	
Red Blood Cell Count	
Hypertension	0.01

Fig 6.2: Prediction of Chronic Kidney Disease

The screenshot shows the results page of the Chronic Kidney Disease prediction application. It features the same input fields as Fig 6.2, a large red banner with the text "Chronic kidney disease", a green "Predict" button, and a yellow banner at the bottom stating "Affected By Chronic Kidney Disease".

Parameter	Value
Red Blood Cells	Rbc
Blood Urea	Bu
Serum Creatinine	Sc
Red Blood Cell Count	Rbcc
Hypertension	0.01

Predict

Affected By Chronic Kidney Disease

Fig 6.3: Results for Prediction

6.4 FUTURE WORK

- Hospitals want to automate the detecting disease persons from eligibility process (real time) based on the account detail.
- To automate this process by show the prediction result in web application or desktop application.
- To optimize the work to implement in Artificial Intelligence environment.

CHAPTER 7

CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be found out. This application can help to find the Prediction of Chronic Renal Disease. Finally, the goal of this research is to utilize ML to anticipate chronic renal illness, which is becoming increasingly common. By comparing and contrasting the Random Forest (RF) techniques, this study hopes to contribute to the progress of healthcare decision-making tools. The importance of early detection and monitoring is critical in overcoming the challenges provided by the progressive nature of chronic renal disease. The findings and insights from this work show promise in improving predictive capacities and influencing effective interventions, marking a significant step toward minimizing the impact of this health burden on individuals and healthcare systems.

REFERENCES

- [1] Haishuai Wang*, Benjamin Bowe*, Zhicheng Cui Hong Yang*, S. Joshua Swamidass, Yan Xie, and Ziyad Al-Aly, "A Deep Learning Approach for the Estimation of Glomerular Filtration Rate", IEEE access, vol-10, pp. 1109/ TNB. 2022.3147957.
- [2] S. N. Induja and C. G. Raji, "Computational methods for predicting chronic disease in healthcare communities," in Proceedings of the 2019 International Conference on Data Science and Communication (IconDSC), pp.1–6, IEEE, Bangalore, India, March 2019.
- [3] B. Manjulatha and P. Suresh, "An ensemble model for predicting chronic diseases using machine learning algorithms," in Smart Computing Techniques and Applications, pp. 337–345, Springer, New York, NY, USA, 2021
- [4] R. Ge, R. Zhang, and P Wang, "Prediction of chronic diseases with multi-labelneural network," IEEE Access, vol. 8, pp. 138210–138216, 2020.
- [5] I. Preethi and K. Dharmarajan, "Diagnosis of chronic disease in a predictive model using machine learning algorithm," in Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), vol. 6, pp. 191–96, IEEE, Bengaluru, India, October 2020.
- [6] S. Y. Yashfi, Md A. Islam, N. Sakib et al., "Risk prediction of chronic kidney disease using machine learning algorithms," in Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), vol. 5, pp. 1–5, IEEE, Kharagpur, India, July 2020.
- [7] S. Ganiger and K. M. M. Rajashekharaiyah, "Chronic diseases diagnosis using machine learning," in Proceedings of the 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), pp. 1–6, IEEE, Kottayam, India, December 2018.
- [8] J. Mishra and S. Tarar, "Chronic disease prediction using deep learning," in Proceedings of the International Conference on Advances in Computing and Data Sciences, vol.4, pp. 201–11, Springer, Valletta, Malta, April 2020.

APPENDIX

A.SOURCE CODE

```
from django.shortcuts import render

from django.shortcuts import render, redirect

import numpy as np

import joblib

model = joblib.load('C:/Users/dell/Downloads/CODE (1) (1)/CODE/rf.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)]

        print(final_features)

        prediction = model.predict(final_features)

        print(prediction)

        output = prediction[0]

        if output == 1:
```

```
        return render(request, 'index.html', {"prediction_text": "Affected By  
Chronic Kidney Disease"})
```

```
    else:
```

```
        return render(request, 'index.html', {"prediction_text": "Not Affected By  
Kidney Disease"})
```

```
    print(output)
```

B. SCREENSHOTS

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The notebook is titled "Module - 1" and "Data Pre-Processing". The code editor contains four cells of Python code. The first cell imports pandas and numpy. The second cell imports warnings and filters them. The third cell reads a CSV file named "kidney.csv". The fourth cell displays the first five rows of the data using data.head().

```
[1] import pandas as p
import numpy as n

[2] import warnings
warnings.filterwarnings('ignore')

[3] data=p.read_csv('kidney.csv')

[4] data.head()
```

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn	Class
0	80.0	1.020	1.0	0.0	1.0	36.0	1.2	137.53	4.63	15.4	7800.0	5.20	1.0	1
1	50.0	1.020	4.0	0.0	1.0	18.0	0.8	137.53	4.63	11.3	6000.0	4.71	0.0	1

Fig B.1: Data Pre-Processing

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The notebook is titled "Module - 2" and "Visualization". The code editor contains four cells of Python code. The first cell imports pandas, numpy, seaborn, and matplotlib. The second cell imports warnings and filters them. The third cell reads a CSV file named "kidney.csv". The fourth cell drops rows with missing values using df.dropna().

```
[1] import pandas as p
import numpy as n
import seaborn as s
import matplotlib.pyplot as plt

[2] import warnings
warnings.filterwarnings('ignore')

[3] data=p.read_csv('kidney.csv')

[4] df=data.dropna()
```

df.columns

Fig B.2: Data Visualization

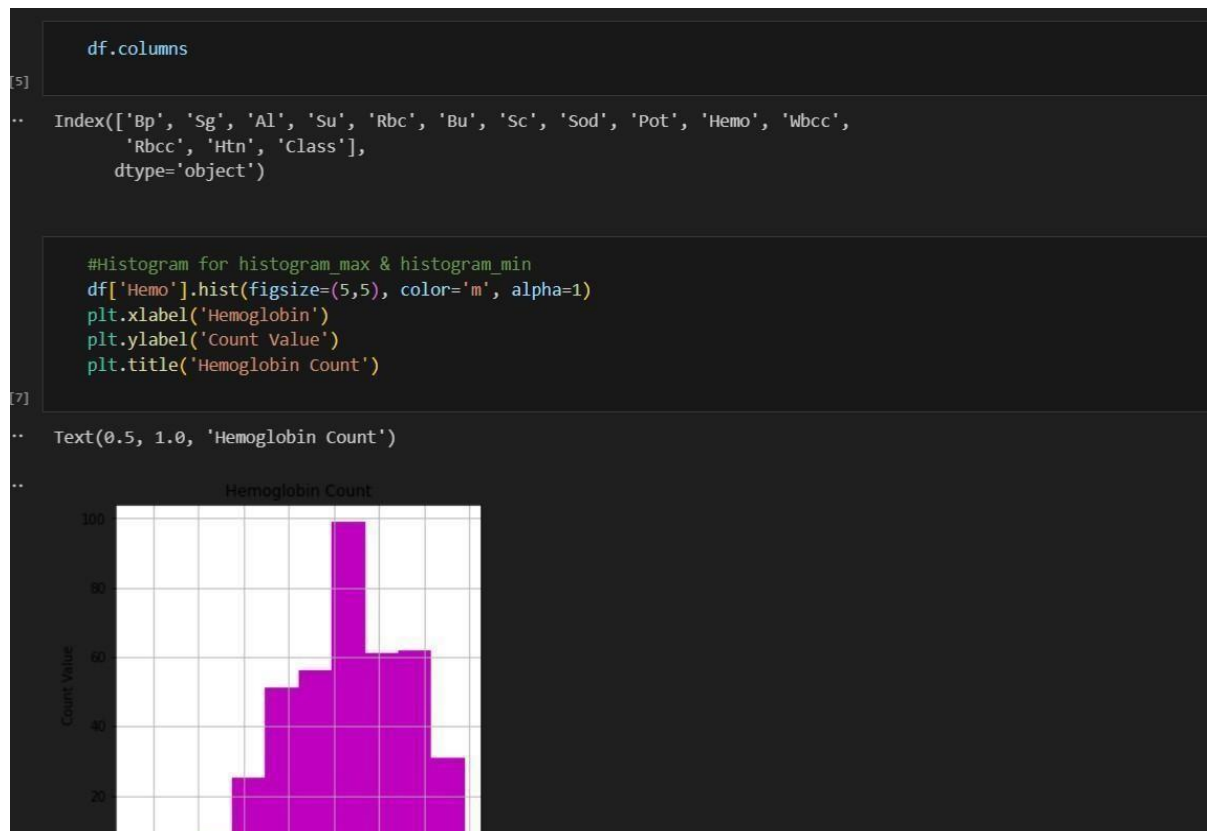


Fig B.2.1: Data Visualization 1



Fig B.2.2: Data Visualization 2

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 400 entries, 0 to 399
Data columns (total 14 columns):
#   Column  Non-Null Count  Dtype
---  -
0    Bp      400 non-null       float64
1    Sg      400 non-null       float64
2    Al      400 non-null       float64
3    Su      400 non-null       float64
4    Rbc     400 non-null       float64
5    Bu      400 non-null       float64
6    Sc      400 non-null       float64
7    Sod     400 non-null       float64
8    Pot     400 non-null       float64
9    Hemo    400 non-null       float64
10   Wbcc    400 non-null       float64
11   Rbcc    400 non-null       float64
12   Htn     400 non-null       float64
13   Class   400 non-null       int64
dtypes: float64(13), int64(1)
memory usage: 46.9 KB

df.columns

Index(['Bp', 'Sg', 'Al', 'Su', 'Rbc', 'Bu', 'Sc', 'Sod', 'Pot', 'Hemo', 'Wbcc',
       'Rbcc', 'Htn', 'Class'],
      dtype='object')

```

Fig B.3: Logistic Regression

Support Vector Machine

```

#import library packages
import pandas as p
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as n

```

```

#Load given dataset
data = p.read_csv("kidney.csv")

```

```

import warnings
warnings.filterwarnings('ignore')

```

```
data.head()
```

	Bp	Sg	Al	Su	Rbc	Bu	Sc	Sod	Pot	Hemo	Wbcc	Rbcc	Htn	Class
0	80.0	1.020	1.0	0.0	1.0	36.0	1.2	137.53	4.63	15.4	7800.0	5.20	1.0	1
1	50.0	1.020	4.0	0.0	1.0	18.0	0.8	137.53	4.63	11.3	6000.0	4.71	0.0	1
2	80.0	1.010	2.0	3.0	1.0	53.0	1.8	137.53	4.63	9.6	7500.0	4.71	0.0	1

Fig B.4: Support Vector Machine

```

from sklearn.ensemble import RandomForestClassifier

rfc= RandomForestClassifier()
rfc.fit(X_train,y_train)
predictRF = rfc.predict(X_test)

print("")
print('Classification report of Random Forest Results:')
print("")
print(classification_report(y_test,predictRF))

print("")
cm1=confusion_matrix(y_test,predictRF)
print('Confusion Matrix result of Random Forest Classifier is:\n',cm1)
print("")

accuracy = cross_val_score(rfc, X, y, scoring='accuracy')
print('Cross validation test results of accuracy:')
print(accuracy)
#get the mean of each fold
print("")
print("Accuracy result of Random Forest Classifier is:",accuracy.mean() * 100)
RF=accuracy.mean() * 100

```

Classification report of Random Forest Results:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	45
1	1.00	1.00	1.00	75

Fig B.5: Random Forest Algorithm

```

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

model = joblib.load('C:/Users/dell/Downloads/CODE (1) (1)/CODE/rf.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)]
        print(final_features)
        prediction = model.predict(final_features)
        print(prediction)
        output = prediction[0]
        if output == 1:
            return render(request, 'index.html', {"prediction_text": "Affected By Chronic Kidney Disease"})
        else:
            return render(request, 'index.html', {"prediction_text": "Not Affected By Kidney Disease"})
        print(output)

```

Fig B.6: Implementation of Views.py

CHRONIC KIDNEY DISEASE PREDICTION USING MACHINE LEARNING

Dr. Rajasekar P,
Sathyabama Institute of Science
and technology
Mail Id :
rajasekar.cse@sathyabama.ac.in

Kavitha VR
B.Tech, Sathyabama Institute of
Science and technology,
Chennai, India
Mail Id :
rangaiahkavitha2003@gmail.com

Charulatha K
B.Tech, Sathyabama Institute of
Science and technology,
Chennai, India
Mail Id :
charukumar297@gmail.com

ABSTRACT

Each year, more people are diagnosed with advanced stages of renal disease. The condition known as chronic kidney disease (CKD), or renal disease, is characterized by a progressive decline in kidney function over time. Patients are more likely to be screened for CKD if they have diabetes, high BP, or a family history of the illness. Early diagnosis is crucial for effective treatment. Early detection and monitoring are crucial to prevent kidney injury or failure. Machine Learning (ML) can effectively support decision-making in healthcare organizations. This project aims to develop and propose a ML strategy for predicting chronic kidney disease. We proposed a Random Forest (RF) algorithm. The components are developed using chronic kidney disease datasets and compared to get the best model for prediction.

Keywords: Chronic Kidney Disease, Machine Learning.

1. INTRODUCTION

Their research has led to improvements in CKD diagnosis. Mean imputation is employed in the aforementioned models to fill in the missing data based on the diagnostic categories of the samples. Their technique cannot be used as the diagnostic results of the samples are unknown. In point of fact, prior to being diagnosed, patients may fail certain tests for a variety of reasons. Furthermore, when missing values are present in categorical categories, mean imputation-derived data may significantly deviate from the actual values.

1.1 CHRONIC KIDNEY DISEASE

CKD is a condition in which kidney function gradually declines over several months or years. At first, there are no symptoms; however, disorientation, leg swelling, fatigue, vomiting, and a lack of appetite may follow. Anemia, high blood

pressure, broken bones, and heart disease are among the complications. Gout, diabetes, polycystic kidney disease, high blood pressure as well as other conditions can all contribute to CKD. A family history of chronic renal disease is one risk factor.

The diagnosis is made with a urine test to measure albumin and a blood test to measure eGFR (Estimated Glomerular Filtration Rate). A kidney biopsy or ultrasound may be used to identify the underlying cause.

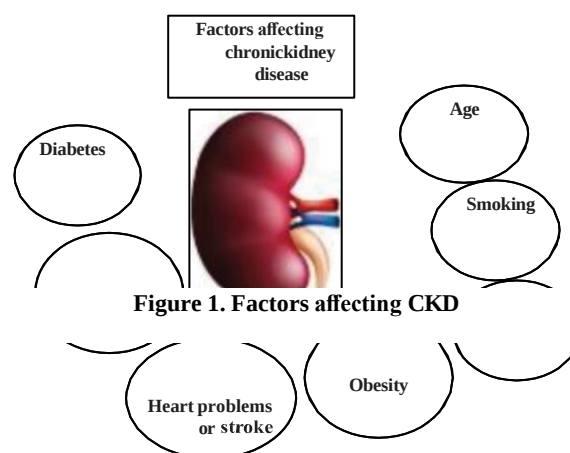


Figure 1. Factors affecting CKD

1.2 MACHINE LEARNING

ML is the research of computer algorithms that automatically get better over time. It is speculated to be a subset of AI (Artificial Intelligence). Sample data, or "training data," is used to build a model by ML methods to make judgments or predictions without being explicitly programmed to do so. A lot of applications, like computer vision and email filtering, use ML methods when it would be impossible or difficult to develop conventional algorithms that can do the job. However, machine learning is not limited to statistical learning. A part of machine learning is linked to computational statistics, which studies how computers can make predictions; However, there are other types of

machine learning as well as statistical learning. The theory, method, and application areas of machine learning are provided by the analysis of mathematical optimization. Another related study is data mining, which uses unsupervised learning to do experimental data analysis. The interaction by which PCs figure out how to finish jobs without being explicitly trained is known as AI.

2. LITERATURE SURVEY

[1] LINTA ANTONY and others present a far-reaching solo structure for the expectation of persistent kidney sickness in this review. offered in countries with a variety of social determinants of health, the prevalence, incidence, and course of CKD have changed over time. High BP and diabetes are the most frequent causes of CKD in most countries. A condition known as CKD is described by indicators of kidney damage and progressive decreases in GFR (“Glomerular Filtration Rate”). People who have CKD are more likely to die young. Doctors need to quickly find a number of disorders linked to CKD because doing so might prevent or even reverse damage to a kidney. Early identification can improve patient care and treatment. In many outlying hospitals and clinics, there are not enough nephrologists or general practitioners to recognize the symptoms.

[2] An AI System for Out-of-Clinic Heart Failure Heartbeat Location in this Work by ERIK ALONSO et al. suggest. Automated pulse detection would make it easier to quickly diagnose cardiac arrest and detect spontaneous circulation recovery during OHCA (“Out-Of-Hospital Cardiac Arrest”). In this work, a reliable pulse identification algorithm was created by combining the majority of defibrillator signals, including ECG & TI. From 187 OHCA patients, 1140 ECG and TI segments were included in the study's dataset. 792 of the segments have been classified as PR (Pulse-Generating Rhythm) and 348 as pulse-less electrical activity (PEA) by an OHCA expert panel. The TI's impedance circulation component and 1st derivative were retrieved via adaptive filtering. The subband components and denoised ECG were created using wavelet decomposition of the ECG.

[3] JIONGMING QIN et al. use machine learning to diagnose CKD. suggest that CKD is a worldwide health problem that is responsible for a large number of other disorders, as well as a significant number of deaths. Because there have been no noticeable

symptoms, patients frequently miss the CKD initial stages. If CKD is detected early, patients can receive prompt treatment to slow its progression. Machine learning models could effectively help doctors attain this aim because of their fast & accurate identification capabilities. A machine-learning approach to CKD diagnosis is proposed in this paper. There are a ton of missing qualities in the CKD informational collection, which was taken from the AI storehouse at the College of California, Irvine (UCI). The missing data for each incomplete sample were processed using KNN imputation by selecting multiple complete samples with the most comparable measurements. Missing values are common in real-world medical scenarios because people may miss certain measures for a variety of reasons.

[4] A Novel Approach to Uneven Data Ensemble Learning for Medical Diagnosis NA LIU et al. With the help of ML (machine learning) approaches pathologists and doctors can avoid potential errors caused by inexperience, exhaustion, stress, and other factors. Medical data can also be analyzed more quickly and thoroughly. However, despite the fact that traditional ML methods like classification were able to achieve good classification accuracy when applied to medical diagnostics, their performance is fatally hampered by an unbalanced dataset, particularly when it comes to the detection of minority groups. A unique ensemble learning paradigm with three phases is presented in this paper for medical diagnosis with unbalanced data: data pre-processing, the training base classifier, and the final ensemble to address shortcomings in existing classification methods. I combine the Cross-Validated Committees Filter (CVCF) with an extension of the SMOTE (Synthetic Minority Oversampling Technique) in the 1st phase of data pre-processing. To perform effectively in the classification process, SMOTE may prepare the sample of minorities, balance the input instances, and filter the noisy examples.

[5] A Review of the ML approaches utilized for Cancer Prediction Gene Expression Analysis in this study, Mahmud Khalsanet and coworkers suggested powerful ML algorithms, which are typically used to create cancer prediction models based on linked gene expression and mutation data. According to this publication, gene expression data from the lung, breast, ovarian, kidney, central nervous system, liver, and gallbladder have been used to predict

survival, identify tumours, and stratify patients. In addition, I provide a synopsis of the various biomarker studies related to cancer. The survey looks at a lot of cancer research that uses machine learning, like how to find biomarker genes, RNA-Seq data, and microarray data, classify and predict cancer, and so on.

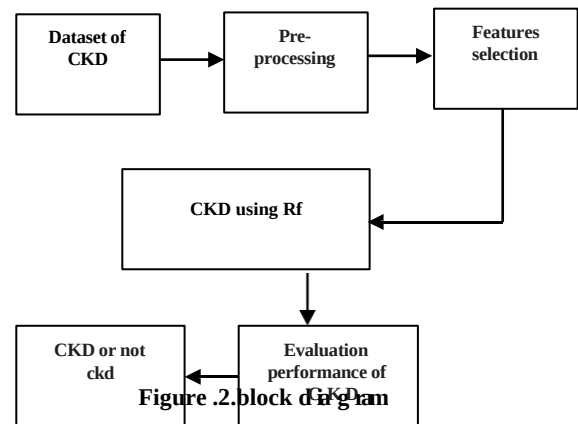
3. RELATED WORK

The prevalence, incidence, and development of CKD have changed throughout time, particularly in nations with diverse social determinants of the health. In most countries, diabetes and hypertension are the CKD leading causes. According to worldwide recommendations, CKD is a disorder which outcomes in decreasing kidney function over time, as seen by GFR (Glomerular Filtration Rate) and kidney damage markers. People with CKD are possible to die at a young age. Doctors must diagnose various CKD-related diseases early on because initial detection can prevent or even reverse renal damage. Better treatment and patient care may result from early detection. There are few nephrologists or general practitioners in many remote hospitals and clinics who are capable of diagnosing the symptoms. As a result, patients are now required to wait longer for a diagnosis. As a result, this study argues that establishing an intelligent system for the classification of patients into 'CKD' or 'Non-CKD' categories will assist doctors in dealing with several patients and providing diagnoses more quickly.

4. METHODOLOGY

The input is the CKD dataset with its various properties. During pre-processing, redundant data and unknown properties are removed. All of the chosen characteristics and features have been selected. To improve classification performance, algorithms like RF (Random Forest) are utilized. Accuracy, recall, precision, and the f-measure will all receive grades. These parameters will be presented as a visual representation. Using a lot of CKD data and the model's accuracy on their test data, they developed a neural organization-based classifier and used image enrolment to find changes in the kidney's morphology. Additionally, the majority of previous studies utilized the UCI AI repository's CKD informational index. This study looks into how machine learning (ML) can be used to diagnose chronic kidney disease. The successful application of ML algorithms, which have been a

driving force in the detection of anomalies in a variety of physiological data, is benefiting a number of categorization tasks.



A. Loading The Dataset

Loading data is the first step in every data analysis or machine learning project. In this case, the Chronic Kidney Disease (CKD) dataset is put into the software. The dataset typically includes a variety of variables and corresponding CKD diagnosis outcomes for each patient. Loading the dataset gives you access to and control over the data, allowing you to conduct additional analysis.

B. Pre-Processing

Data Pre-Processing involves encoding data in a machine-readable format, allowing a dataset to be seen as a collection of objects. The dataset may contain missing values. Missing values must be dealt with first and can be calculated or removed from the dataset. Missing values are typically filled using the feature's mean, median, mode, or constant value. It is necessary to transform the categorical data of the object type into float64 type because the object values cannot be used for the analysis. The value that appears the most frequently in that column is used to replace null values in category attributes. Label encoding represents each distinct value as an integer, converting categorical attributes to numerical values. This automatically converts the attributes to the integer type. The panda's package is useful for data pre-processing.

C. Feature Selection

The procedure of selecting the most pertinent characteristics (features) from a dataset for model training is known as feature selection. The goal is to

reduce dimensionality while maintaining the most useful features. Various approaches, like mutual information, correlation analysis, and feature importance scores, can be used to determine which features are most important for CKD prediction. Choosing the right features can improve model performance while reducing computational overhead.

Features	Mean	Standard	Max	Min
Age	51.483	17.21	90	2
Blood urea	57.426	49.987	391	1.5
Serum creatinine	3.072	4.512	76	0.4
Blood glucose random	148.037	76.583	490	22
Blood pressure	76.469	13.756	180	50
Packed cell volume	38.884	8.762	54	9
Potassium	4.627	2.92	47	2.5
Hemoglobin	12.526	2.815	17.8	3.1
Sodium	137.529	9.908	163	4.5
Red blood cell count	4.707	0.89	8	2.1
White blood cell count	8406.12	2823.35	26400	2200

Table 1. Statistical analysis of the dataset of numerical features

D. Model Fitting and Testing

After feature selection, data is divided into 2 halves in a 4:1 ratio: 80% for training and 20% for testing learned models. The data has been utilized for suggested models, including Random Forest. The models are compared based on their predicted accuracy.

E. Classification Performance

Classification performance is an assessment of how well machine learning models can classify or predict outcomes, such as the presence or absence of CKD. Several measures are routinely used to assess classification model performance, involving precision, F1-score, accuracy, recall, and the confusion matrix. The accuracy ratio measures the model's correctness by comparing correctly predicted instances to the overall number of

examples in the dataset. Precision measures the fraction of real positive predictions between all positive predictions, allowing for the evaluation of the capability of the model for avoiding false positives. Recall (Sensitivity) analyzes the fraction of true positive predictions among all the actual positive examples, showing the capacity of the model to catch all relevant cases. F1-score is the harmonic mean of recall and precision, allowing a fair evaluation of the performance of the model.

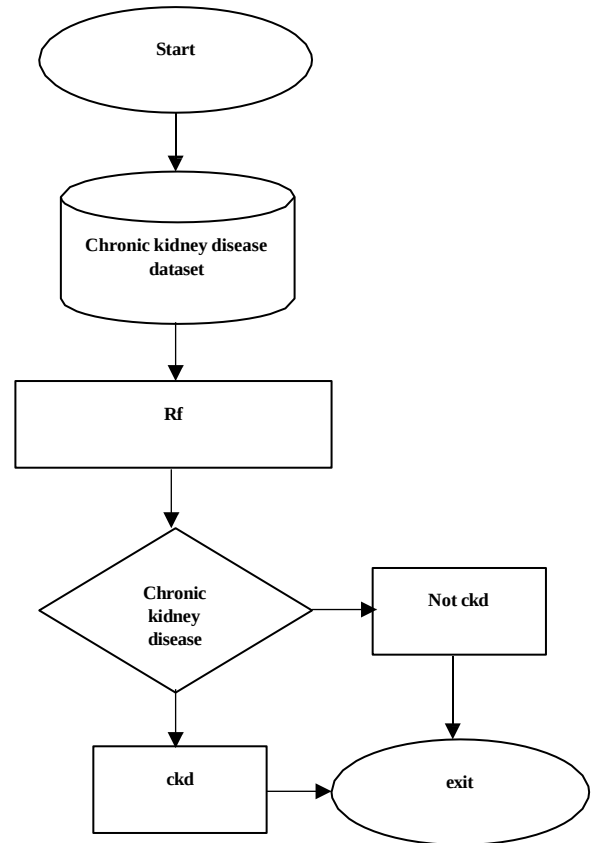


Figure 3. Flow diagram

5. ALGORITHM DETAILS

A. Random Forest Classifier

Random Forest uses hyperparameters similar to Decision Trees and Bagging Classifiers. Combining a decision tree with a bagging classifier is unnecessary when using the random forest classifier-class. Random forest can be used for regression applications by using the repressor of the method. Using a random forest during tree development increases the model's flexibility. When splitting a hub, it selects the optimal component from a diverse collection of supplies, rather than the

principal piece. There are numerous alternatives available, with the majority leading to a superior model. The calculation for separating a hub only examines an irregular subset of items in random timberland. To increase the randomness of trees, consider utilizing irregular bounds for each component instead of searching for the optimal edges. The Random Forest approach combines many decision trees into a classification and regression ensemble. A random subset of the training data sets is used to construct multiple decision trees. Using many decision trees improves the accuracy of results. This approach has a short runtime and can handle missing data. Random forest is used to randomize the procedure rather than the training dataset. Decision trees generate decision classes.

1. Here, the author first displays the pseudocode for creating Random Forests: Choose "K" features at random from the total "m" features, where $k \ll m$.

2. Determine the node "d" among the "K" characteristics by utilizing the optimal split point.

3. Utilizing the best split, divide the node into daughter nodes.

4. Once the "l" number of nodes was obtained, repeat steps a through c.

5. To build a forest, repeat steps a through d a number of times to get a total of "n" trees.

We'll make the prediction in the following step once the RF classifier has been formed. Below is the pseudo-code for the random forest prediction:

1. To predict the outcome, use the test features and the randomly generated decision tree's rules. Next, the anticipated outcome (target) is stored.

2. Calculate the votes for each and every predicted target.

3. The random forest algorithm's highly voted anticipated target serves as the basis for the final forecast. Although the process is simple to comprehend, it is efficient in some ways.

6. RESULT ANALYSIS

The table compares three classifiers RF, K-Nearest Neighbors (KNN), and on the basis of their predictive performance in the setting of chronic renal disease, as described in the abstract. The accuracy metric measures the percentage of

correctly classified instances, and RF has notable high accuracies of 98.33% and 96.67%, respectively, demonstrating their effectiveness in detecting patterns in the dataset. Precision, a measure of the classifier's capability to prevent false positives, reveals the resilience of RF, with scores of 100% and 92%, respectively, with percentages of 97.37% and 94.74%, respectively. The F1-score, a harmonic mean of precision & recall, validates RF's overall superior performance, with scores of 98.67% and 97.3%, respectively. These findings emphasize the potential of RF to provide accurate and reliable predictions for chronic kidney disease, making them intriguing candidates for further investigation and potential clinical use.

Classifiers	KNN	RF
Accuracy %	96.67	98.33
Precision %	92	100
Recall %	94.74	97.37
F1-score%	97.3	98.67

Table .2.comparison table

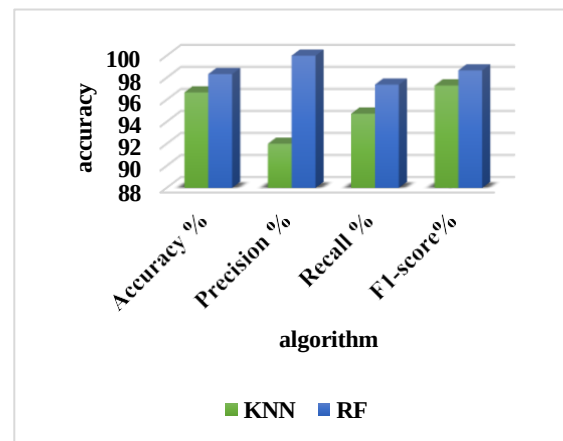


Figure.4. comparison graph

7. CONCLUSION

Finally, the goal of this research is to utilize ML to anticipate chronic renal illness, which is becoming increasingly common. By comparing and contrasting the Random Forest (RF) techniques, this

study hopes to contribute to the progress of healthcare decision-making tools. The importance of early detection and monitoring is critical in overcoming the challenges provided by the progressive nature of chronic renal disease. The findings and insights from this work show promise in improving predictive capacities and influencing effective interventions, marking a significant step toward minimizing the impact of this health burden on individuals and healthcare systems.

8. FUTURE WORK

Future research in this area should concentrate on improving and expanding the machine learning models described in this study to predict chronic renal disease. Further investigation of advanced algorithms and ensemble methods may improve predicted accuracy and robustness. Furthermore, combining different and larger datasets would improve model generalization and real-world application. The addition of more relevant variables, such as genetic markers or lifestyle factors, may increase the models' capacity to capture the complexities of chronic kidney disease risk.

9. REFERENCES

1. "a total solo system for persistent renal sickness expectation," *Appl.*, by J. M. Bolarn, f. Cavas, j. S. Velázquez, j. L. Alió *Sci.*, vol. 10, no. 5, pp. 1874, March 2020.
2. A machine learning framework for the detection of pulses during cardiac arrest outside of a hospital, by M. Abdar, M. Zomorodi-moghadam, x. Zhou, r. Gururajan, x. Tao, and p. D. Barua, among others," *pattern recognition letters*, vol. 132, pp. 123-131, apr. 2020.
3. "a method for identifying chronic renal disease using machine learning," *J. Med.*, by S. Thompson, J. F. Mather, C. M. Waszynski, and R. S. Dicks, along with J. P. Corradi. *Syst.*, 42, no. 12, p. 261, 2020.
4. Springer Singapore, pp., R. A. Shah, M. Zaman, S. A. Khan, N. A. Ganai, M. Ashraf, and others a brand-new approach to ensemble learning for medical diagnosis using unbalanced data." 239-255, 2021.
5. Vol., *Resuscitation*, 147, pages by H. Kwok, j. Coult, j. Blackwood, s. Bhandari, p. Kudenchuk, and t. Rea 104-111, Feb. 2020
6. W. Zhu, L. Xie, J. Han, and X. Guo, "Analysis for Disease Gene Association Using Machine Learning," *Cancers*, vol. 12, no. 3, pp. 603, March 2022.
7. D. Jain and V. Singh, "Covid-19 future forecasting using supervised machine learning models", *International Journal of Computer Applications*, vol. 43, no. 6, pp. 524-536, July 2021.
8. Detecting keratoconus from corneal imaging data using machine learning. Geneva, Switzerland, 2021.
9. F. Petropoulos and S. Makridakis, "Diabetes prediction using assembly of different machine learning classifiers," *PLoS ONE*, vol. 15, no. 3, March 2022.
- 10 S. Kumari, D. Kumar, and M. Mittal, "Efficient prediction of cardiovascular disease using machine learning algorithms with relief and lasso feature selection techniques", *int. J. Cogn. Comput. Eng.*, vol. 2, pp. 40-46, June 2021.
11. J. M. Bolarín, f. Cavas, j. S. Velázquez, and j. L. Alió, "Hybrid machine learning scheme for classification of bectcs and tle patients using eeg brain signals," *Appl. Sci.*, vol. 10, no. 5, pp. 1874, March 2020.
12. M. Abdar, M. Zomorodi-moghadam, X. Zhou, R. Gururajan, X. Tao, P. D. Barua, and R. Gururajan, "Improvement in automated diagnosis of soft tissues tumors using machine learning," *Pattern Recognition Letters*, vol. 132, pp. 123-131, April 2020.
13. N. P. Tigga and S. Garg, "Machine learning approach for prediction of hematic parameters in hemodialysis patients," *Procedia Computer Sci.*, vol. 167, pp. 706-716, 2020.
14. M. S. Oh and M. H. Jeong, "Machine learning for clinical outcome prediction", *Korean Journal of Medicine*, vol. 95, no. 4, pp. 266-275, August 2020.
15. H. Kwok, J. Coult, J. Blackwood, S. Bhandari, P. Kudenchuk, and T. Rea, "Machine learning to identify psychomotor behaviors of delirium for patients in long-term care facilities," *Resuscitation*, vol. 147, pp. 104-111, February 2020

