

A PREDICTIVE FRAMEWORK FOR CHRONIC KIDNEY DISEASE USING MACHINE LEARNING ALGORITHMS

Project report submitted in partial fulfillment of the requirements

For the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND BUSINESS SYSTEM

Submitted by

Yadla Sai Ram (Y20CB064)

Dulam Balu (Y20CB019)

Mundlapati Ramanjaneyulu (Y20CB043)

Under the Guidance of

Mr. B. Rama Krishna

Assistant Professor, Dept. of CSE(DS)



R.V.R & J.C. COLLEGE OF ENGINEERING (AUTONOMOUS)

Approved by AICTE- New Delhi, Accredited by NAAC A Grade and NBA

Permanently Affiliated to Acharya Nagarjuna University, Guntur

NH-5, Chowdavaram, Guntur

May 2024

R.V.R & J.C. COLLEGE OF ENGINEERING (AUTONOMOUS)

DEPARTMENT OF

COMPUTER SCIENCE AND BUSINESS SYSTEM



CERTIFICATE

This is to certify that this project work titled “A Predictive Framework for Chronic Kidney Disease using Machine Learning Algorithms” is the work done by **Yadla Sai Ram(Y20CB064), Dulam Balu(Y20CB019) and Mundlapati Ramanjaneyulu (Y20CB043)** under my supervision, and submitted in partial fulfillment of the requirements for the award of the degree, B.Tech. in Computer Science and Business System, during the Academic Year 2023-2024.

Mr. B. Rama Krishna
Assistant Professor, Dept. of CSE(DS)
Project Guide

Dr. M. V. P. Chandra Sekhara Rao
Professor & HOD
Head-CSBS

Dr. Lakshmikanth Paleti
Associate Professor, Dept. of CSBS
Project Coordinator

External Examiner

DECLARATION

We **Yadla Sai Ram(Y20CB064)**, **Dulam Balu(Y20CB019)** and **Mundlapati Ramanjaneyulu(Y20CB043)** hereby declare that the project report titled “**A PREDICTIVE FRAMEWORK FOR CHRONIC KIDNEY DISEASE USING MACHINE LEARNING ALGORITHMS**” under the guidance of **Mr. B. Rama Krishna** is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Business Systems. This is a record of bonafide work carried out by us and the result embodied in this project have not been reproduced or copied from any source. The result embodied in this project have not been submitted to any other university for the award of any other degree.

Yadla Sai Ram(Y20CB064)

Dulam Balu(Y20CB019)

Mundlapati Ramanjaneyulu (Y20CB043)

Place: Guntur

Date:

ACKNOWLEDGMENT

The successful completion of any task would be incomplete without proper suggestion, guidance, and environment. Combination of these three factors acts like a backbone to our project work “**A Predictive Framework for Chronic Kidney Disease using Machine Learning Algorithms**”.

We are profoundly grateful to express our deep sense of gratitude and respect towards the management of the **R. V. R. & J. C. College of Engineering**, for providing the resources to complete the project.

We are very much thankful to **Dr. Kolla Srinivas**, Principal of **R. V. R. & J. C. College of Engineering** for allowing us to deliver the project successfully.

We are greatly indebted to **Dr. M. V. P. Chandra Sekhara Rao**, Professor, & Head of the department, Computer Science and Business System, **R. V. R. & J. C. College of Engineering** for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college.

We are also thankful to our Project Coordinator **Dr. Lakshmikanth Paleti** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Guide, **Mr. B. Rama Krishna** and other Faculty Members & Support staff for their valuable suggestions, guidance, and constructive ideas in every step, which was indeed of great help towards the successful completion of our project.

Yadla Sai Ram (Y20CB064)

Dulam Balu (Y20CB019)

Mundlapati Ramanjaneyulu (Y20CB043)

ABSTRACT

Chronic Kidney Disease (CKD) or Chronic Renal Disease (CRD) has turned out into a serious issue with the constant rise in the number of patients. Without kidneys, one can live only for 18 days on average, thus kidney disease treatment is very much in demand. There is a need to have effective ways of early foreseeing CKD. This study intends to develop and validate a predictive model for chronic kidney disease. Machine learning algorithms are often applied in medicine to prognosticate and classify diseases. Gradient Boosting algorithms are used within the proposed system. This proposed model will be useful for predicting future CKD as well as non-CKD patients based on different parameters with an impressive accuracy level of 98.75%. One example would be designing a machine learning algorithm that classifies the Severity Stages of Chronic Kidney Disease (CKD), using eGFR as the dominant metric for staging CKD progressions.

TABLE OF CONTENTS

TITLE	PAGE No.
Abstract	i
Table of Contents	ii
List of Figures	iv
List of Tables	v
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: LITERATURE SURVEY	5
CHAPTER 3: SYSTEM ANALYSIS & FEASIBILITY STUDY	10
3.1 Existing System	11
3.1.1 Disadvantages of Existing System	12
3.2 Proposed System	13
3.2.1 Advantages of Proposed System	14
3.3 Methodologies (or) Algorithm	15
3.3.1 Random Forest	16
3.3.2 Naive Bayes	17
3.3.3 K-NN	18
3.3.4 Gradient Boosting	19
3.4 Feasibility Study	22
3.4.1 Economic Feasibility	22
3.4.2 Operational Feasibility	23
3.4.3 Technical Feasibility	23
CHAPTER 4: SYSTEM REQUIREMENTS	24
4.1 Functional Requirements	25
4.2 Non – Functional Requirements	25
4.2.1 Software Requirements	26
4.2.2 Hardware Requirements	27
CHAPTER 5: DESIGN	28

5.1 System Design	29
5.2 UML Diagrams	31
5.2.1 Class Diagram	32
5.2.2 Usecase Diagram	33
5.2.3 Activity Diagram	36
5.2.4 Sequence Diagram	38
CHAPTER 6: IMPLEMENTATION	40
6.1 Coding (pseudo code)	41
CHAPTER 7: RESULTS	46
CHAPTER 8: SOCIAL IMPACT	57
CHAPTER 9: CONCLUSION & FUTURE WORK	60
CHAPTER 10: BIBLIOGRAPHY	63

LIST OF FIGURES

S. No.	Figure No.	Figure Description	Page No.
1.	3.3.1	Random Forest	16
2.	3.3.2	Naive Bayes	17
3.	3.3.3	K-Nearest Neighbors	19
4.	3.3.4	Gradient Boosting	20
5.	5.1	Work Flow Diagram	30
6.	5.2.1	Class Diagram	32
7.	5.2.2	Use Case Diagram	34
8.	5.2.3	Activity Diagram	36
9.	5.2.4	Sequence Diagram	38
10.	7.1	Dataset	47
11.	7.3	Confusion Matrix	50
12.	7.4	Accuracy Comparison	51
13.	7.7.1	Web Interface	52
14.	7.7.2	User Input	53
15.	7.7.3	Person having CKD	54
16.	7.7.4	Person having CKD with Severity Stage	54
17.	7.7.5	Person without having CKD	55

LIST OF TABLES

S. No.	Table No.	Table Description	Page No.
1.	3.1	Performance Metrics of Existing System	11
2.	7.2	Classification Report	50
3.	7.5	Performance Metrics	51
4.	7.6	Comparison with other Models	52

Chapter 1

Introduction

1. INTRODUCTION

1.1 Introduction

Right now, engineers and medical researchers are trying to come up with machine learning algorithms and models that can detect chronic kidney disease early. This is proving difficult because healthcare data is widely spread out and highly complex necessitating complicated analysis of the same. However, by making use of the technology of data mining, this data can be formatted in the form of a machine learning algorithm. This condition has led to many deaths globally, claiming lives each year as well as resulting in kidney failure [1]. A study done by Global Burden of Disease Study (GBDS) 2010 found Chronic Kidney Disease (CKD) moved from its rank at position 27th in 1990 to an alarming rate of 18th most common cause of global death [2]. Such patients are more likely to proceed to end-stage renal disease (ESRD), which demands costly treatment approaches including dialysis and transplantation [3]. Hence, kidney disease stands out as a major public health problem affecting hundreds of thousands across Ethiopia irrespective of age or sex [4]. The goal of this model is to build and validate predictive models for chronic kidney disease. Above all, the model will assess kidney failure in terms of the need for dialysis or transplant [5]. Despite over 2 million individuals requiring dialysis or kidney transplants due to renal failure, which causes more deaths than breast and prostate cancer combined, a hormone that is produced by kidneys, controls several physiological processes like erythropoiesis, regulation of blood pressure as well as calcium metabolism too [31].

1.2 Problem Statement

Chronic Kidney Disease (CKD) Prediction Using Machine Learning Techniques is to develop and validate a predictive model for the early detection and prognosis of CKD employing machine learning algorithms. The research underscores the necessity for effective and accurate methods in predicting CKD at an early stage. Despite the global prevalence of CKD and its potential for severe consequences, current diagnostic and predictive tools may prove insufficient. The complexity and magnitude of health industry data present challenges in analyzing and interpreting relevant information. The research aims to address this by harnessing advanced machine learning algorithms to develop predictive models for CKD.

1.3 Scope and Objectives

1. Utilize Data Mining Technology:

Scope:

This involves leveraging data mining techniques to extract, process, and translate large and complex health industry data into a format suitable for machine learning algorithms.

Objectives:

Develop algorithms or pipelines for data preprocessing to handle various data types (structured and unstructured). Implement data cleaning, normalization, and feature engineering techniques. Translate diverse healthcare data sources (electronic health records, medical imaging, genetic data, etc.) into a unified format.

2. Assess CKD Severity:

Scope:

Assessing CKD severity involves using features such as estimated glomerular filtration rate (GFR), age, diet, medical conditions, and albuminuria to enhance clinical decision-making.

Objectives:

Identify relevant features associated with CKD severity through literature review and expert consultation. Develop algorithms to calculate estimated GFR and assess albuminuria levels. Explore correlations between different features and CKD progression.

3. Create and Validate Predictive Models for CKD:

Scope:

This involves building predictive models specifically focused on evaluating kidney disease, particularly CKD.

Objectives:

Collect high-quality datasets containing patient information, clinical variables, and CKD diagnosis. Develop machine learning models tailored for CKD prediction, considering various algorithms such as logistic regression, decision trees, and ensemble methods.

Validate the models using appropriate techniques such as cross-validation and external validation on unseen data.

4. Develop and Implement Advanced ML Algorithms:

Scope:

Develop and implement advanced machine learning algorithms like Random Forest, K-Nearest Neighbor, Artificial Neural Networks, and Ensemble Models for early detection of CKD.

Objectives:

Implement state-of-the-art ML algorithms suitable for CKD prediction. Optimize hyperparameters and model architectures to maximize performance. Assess the robustness and scalability of the developed algorithms.

5. Improve Existing CKD Model:

Scope: Enhance the existing CKD model by incorporating reliable patient information, identifying new features, and increasing prediction accuracy through expert collaboration.

Objectives:

Collaborate with domain experts to identify additional relevant features for CKD prediction. Incorporate advanced feature selection techniques to prioritize informative features. Continuously evaluate and refine the model based on feedback from clinicians and domain specialists.

Chapter 2

Literature Survey

2. LITERATURE SURVEY

A neural network framework was built by Vasquez-Morales and his co-authors [6] to predict the risk of getting chronic kidney disease using 40,000 instances in their dataset. Their model indicated a level of accuracy at 95%.

Padmanaban and Parthiban [7] suggested the use of machine learning classifier algorithms to help identify chronic kidney disease in its early stages among diabetic patients. Data was gathered from a diabetes research center located in Chennai and then Naïve Bayes and Decision tree algorithms were applied to it. The Naïve Bayes classifier registered the highest rate of accuracy, which reached to 91% which was obtained by the Weka tool.

In an investigation by Sujata Drall, Gurdeep Singh Drall, Sugandha Singh, Bharat Drall, and colleagues (citation [8]), a CKD dataset from UCI was examined that consisted of 400 instances including 25 attributes. The authors employed data preprocessing wherein missing values were found and substituted with zero; before then transforming and applying it into the data set. Afterward, they identified the top five features using an algorithm that discriminated significant attributes and later applied classification algorithms; namely Naïve Bayes KNN [32]. Analysis showed that the highest accuracy was obtained using the K-Nearest Neighbor algorithm.

The authors of “Chronic Kidney Disease Prediction Using Machine Learning Algorithm” by [9] developed a project where they used the dataset specific to CKD. The dataset was historical, CHD-related consisting 24 attributes and 1 target variable. During their model building process, they employed two supervised machine learning algorithms namely KNN and Naïve Bayes. It is worth mentioning that the peak accuracy of KNN was 97% whereas Naïve Bayes achieved an accuracy level of 91%.

Charleonnann et al. in their comparative analysis study [10] included predictive models such as K-nearest neighbors (KNN), support vector machine (SVM), logistic regression (LR), and decision tree (DT) [32]. These models were built on Indian CKD-specific datasets. Their objective was to determine which classifier works best for CKD prediction. SVM was the best model with an accuracy of 98.3% and a sensitivity of 0.99.

In Tekale et al.'s paper "Prediction of Chronic Kidney Disease Using Machine Learning Algorithm" [11], a data set consisting of 400 instances and 14 attributes was employed.. Here, support vector machines (SVM) and decision trees are the methodologies used. After the data set underwent preprocessing, there were only 14 features instead of 25, after the original 25. Achieving an accuracy rate of 96.75%, Support Vector Machine emerged as the most effective model in terms of accuracy.

Yashfi et al.[12] introduced a methodology where they predicted the risk of Chronic Kidney Disease (CKD), using machine learning algorithms, and data from CKD patients. The analysis was done using Random Forest and Artificial Neural Network (ANN). From 25 features in the beginning, 20 for each were chosen and both RF and ANN were applied. RF showed maximum accuracy which amounted to 97.12%.

Vinod et al.[13] conducted an evaluation of seven supervised machine learning algorithms including K-Nearest Neighbor, Decision Tree, Support Vector Machine, Random Forest, Neural Network, Naïve Bayes and Logistic Regression to identify the best performing model for BCD prediction by various performance metrics. Finally, it turned out that K-NN was the best performer on the BCD dataset with an accuracy rate of 97%.

Pal, Saurabh et al.[14] attempted to discuss machine learning approaches towards predicting chronic kidney disease thereby providing meaningful insights into how biomedical predictive tools are used [33]. The choice of different algorithmic techniques such as decision tree classifier logistic regression support vector machines was employed among others but it is observed that decision trees have the highest accurate diagnosis.

Debal, Adeba, and Sitote, the authors , made the decision to research the application of machine learning algorithms in prognosticating chronic renal disease. Among its techniques were logistic regression, random forests, and gradient boosting [15]. When it came to accurately predicting the risk of chronic kidney disease, the decision tree was the most accurate method, followed by random forests and neural networks in that order (Dritsas & Trigka; [16]).

The forecast of chronic kidney disease was investigated by Almustafa [17] using different classification algorithms thus enhancing medical informatics. Decision trees, k-nearest neighbors and support vector machines were part of the analysis to determine which

algorithm performed better in predicting the disease, indicating that support vector machine had a higher accuracy rate.

According to Ifraz et al [18], his research compared intelligent approaches like artificial neural networks to predict kidney disease with other techniques that are described in computational healthcare literature. This study looked at different algorithms such as decision trees, logistic regression, and artificial neural networks showing that increased accuracy was obtained by artificial neural networks in disease prediction.

Hittora et al [19] gave their perspective on how biomedical informatics has been influenced by the machine learning aspect of CKD prediction. The study used various algorithms such as decision trees, random forests, and gradient boosting where it was found that gradient boosting had the highest accuracy in predicting the disease.

Walse et al [20] also contributed to this area through exploring suitable applications of methods such as Naïve Bayes, decision tree or random forest while analyzing chronic kidney disease thereby promoting machine learning in healthcare informatics [34]. In this regard, among these techniques which randomly select subsets from the original dataset for classification problems; a better performance was noted for random forest during disease analysis.

Nithya et al. [21] also ventured into artificial neural networks and multi-kernel k-means clustering techniques for detection and segmentation of kidney disease in ultrasound images to move medical image analysis forward. They applied artificial neural networks and multi-kernel k-means clustering algorithms with artificial neural networks having higher accuracy than other methods of disease detection and segmentation.

Navaneeth and Suchetha [22] introduced a method that was new based on dynamic pooling in convolutional neural networks for chronic kidney disease detection, thereby expanding the frontiers of deep learning in biomedical signal analysis. For example, the study employed convolutional neural networks where the proposed dynamic pooling methodology proved more efficient than other methods when detecting diseases [35].

Aqlan et al [23] utilized data mining strategies as a way to predict chronic kidney disease which offered some insights into how industrial and systems engineering could be applied

in healthcare. The investigation involved the use of diverse algorithms that may embrace decision trees, logistic regression, and artificial neural networks besides artificial neural network factors among others. In addition to this, probably this research also looked at the estimated glomerular filtration rate (eGFR) as a possible indicator of chronic kidney disease.

Chapter 3

System Analysis & Feasibility Study

3. SYSTEM ANALYSIS & FEASIBILITY STUDY

3.1 Existing System

The Existing work Chronic Kidney Disease Prediction Using Machine Learning Techniques used Machine Learning models have been applied, such as Logistic Regression (LR), Decision Tree (DT), and Support Vector Machine (SVM) and Bagging ensemble method is applied to improve the performance of the developed model Using a comprehensive dataset of clinical attributes, individual models are trained and evaluated. The ensemble method combines the strengths of these models, resulting in improved predictive accuracy [36,37]. Experimental results demonstrate an achieved accuracy of 97%, indicating the effectiveness of the proposed approach in CKD prediction. This research contributes to advancing healthcare informatics and may aid in early CKD detection, facilitating timely intervention and patient management.

Classifier used	Precision	Recall	F1-score	Accuracy achieved (%)
Logistic regression	0.96	0.96	0.96	93.28
Decision tree	0.99	0.98	0.98	95.92
Support vector machines	0.96	0.96	0.96	94.80

Table 3.1 Performance Metrics of Existing Work

3.1.1 Disadvantages of Existing Systems

While the existing system for chronic kidney disease (CKD) prediction using machine learning techniques has achieved high accuracy, there are still some drawbacks and limitations:

- **Limited Generalizability:** The high accuracy achieved in the experimental setup may not necessarily generalize well to new, unseen data. There's a risk of overfitting to the specific dataset used for training, which may not capture the full variability present in real-world clinical settings.
- **Interpretability Issues:** Ensemble methods, such as Bagging, can enhance predictive accuracy but may sacrifice interpretability. It can be challenging to understand the underlying decision-making process of the combined models, hindering the ability to interpret and trust the predictions, especially in critical healthcare decisions.
- **Data Quality and Bias:** The effectiveness of machine learning models heavily relies on the quality and representativeness of the dataset used for training. Biases or inaccuracies in the data, such as missing values, imbalances in class distribution, or errors in labeling, can impact the performance and reliability of the predictive models.
- **Feature Engineering Challenges:** While using a comprehensive dataset of clinical attributes is beneficial, it also poses challenges in feature selection and engineering. Not all features may be equally relevant or informative for CKD prediction, and selecting the most discriminative features can be complex and subjective.

Logistic Regression:

- Assumes linear relationship between features and the log-odds of the outcome, which may not hold true for complex data.
- Prone to overfitting when there are many features or when features are highly correlated.

Decision Trees:

- Tendency to overfit the training data, especially with deep trees, which can result in poor generalization to unseen data.
- Difficulty in handling missing data and categorical variables without preprocessing.

Support Vector Machines (SVMs):

- Choice of kernel function and parameters can significantly affect performance, requiring careful tuning.
- Prone to overfitting when the dataset is noisy or when the number of features is much larger than the number of samples.

3.2 Proposed System

The proposed work aims to develop a novel machine learning-based model for the classification of chronic kidney disease (CKD) involves the usage gradient boosting, one type of machine learning algorithm and voting classifiers Utilizing a dataset sourced from the University of California, Irvine Machine Learning Repository focusing on chronic renal disorders, the study allocated 80% of the data for training and 20% for testing the model. Our research methodology is centered on the creation of a machine learning model specifically designed for classifying the stages of severity of chronic renal disease with eGFR serving as the prime metric for stage classification in CKD progression. Clinical parameters and eGFR measurements are included in a large dataset to train and validate our model. We used machine learning algorithms, such as Gradient Boosting, to find the most effective one for CKD stage classification. When this research is complete, it will produce a powerful, accurate machine learning approach that distinguishes patients' GFR values by their CKD severity stages to enhance healthcare decisions and patient management.

Stages of CKD are:

Stage 1: Kidney damage with normal or increased GFR (>90 mL/min/1.73 m²)

Stage 2: Mild reduction in GFR (60-89 mL/min/1.73 m²)

Stage 3a: Moderate reduction in GFR (45-59 mL/min/1.73 m²)

Stage 3b: Moderate reduction in GFR (30-44 mL/min/1.73 m²)

Stage 4: Severe reduction in GFR (15-29 mL/min/1.73 m²)

Stage 5: Kidney failure (GFR < 15 mL/min/1.73 m² or dialysis)

Advantages:

Gradient boosting offers high accuracy in chronic kidney disease prediction by combining the strengths of multiple weak models into a strong ensemble. It effectively captures non-linear relationships and provides feature importance, while regularization techniques control overfitting and enhance model robustness.

Using CKD stages for prediction enables targeted treatment, progress monitoring, and risk stratification for improved patient outcomes.

High Accuracy: GB and Ensemble Models are known for their high accuracy in classification tasks. They combine multiple weak learners to create a strong predictive model, which can better capture the complexities of CKD progression.

Robustness: Ensemble models are inherently robust. They are less prone to overfitting compared to individual models. By combining multiple models, they reduce the risk of making decisions based on noise or outliers in the data.

Feature Importance: GB models provide feature importance scores, indicating which features (in this case, possibly clinical indicators like blood pressure, serum creatinine, etc.) are most influential in predicting CKD severity stages. This information can help clinicians understand the factors driving CKD progression.

Interpretability: While ensemble models like Voting Classifier may not be as interpretable as simpler models like decision trees, they can still provide insights into the relative importance of different features in predicting CKD severity stages.

3.2.1 Data Preprocessing

Data Preprocessing plays a crucial role in preparing raw data for machine learning classifiers [38]. It involves several steps aimed at refining the data, making it suitable for analysis and modeling. Let's delve deeper into each aspect:

1. **Handling Missing Data:** One common issue with datasets is missing values. Preprocessing involves strategies to deal with these gaps, such as imputation, where missing values are filled in with estimates derived from the available data [39]. Using the median as a replacement for missing values is advantageous, especially for large datasets, as it helps maintain the central tendency of the feature without being influenced by outliers.

2. Categorical to Numeric Conversion: Machine learning algorithms typically work with numerical data, so categorical variables need to be transformed into a numeric format. This conversion enhances interpretability and enables algorithms to effectively process categorical information [40]. Techniques such as one-hot encoding or label encoding are commonly used for this purpose.

3. Rescaling: Attributes in a dataset may have different scales. Rescaling is the process of transforming these attributes to a uniform scale, usually between 0 and 1 or -1 and 1. This step is vital for algorithms that are sensitive to the scale of the features, such as distance-based methods like k-nearest neighbors or algorithms like support vector machines.

4. Binary Transformation: Some algorithms require binary input data. Binary transformation converts categorical or continuous variables into binary format, where each variable is represented as a series of binary digits (0s and 1s)[41]. This transformation is particularly useful in scenarios where only the presence or absence of a feature is relevant.

5. Standardization: Standardization involves transforming the data such that it has a mean of 0 and a standard deviation of 1. This process ensures that the features have comparable scales, which can improve the performance of certain algorithms, especially those based on distance measures or gradient descent optimization.

Overall, data preprocessing plays a critical role in ensuring that the data fed into machine learning models is clean, consistent, and suitable for analysis. By addressing issues such as missing data, categorical variables, varying scales, and data formats, preprocessing enhances the performance and interpretability of machine learning algorithms across various applications.

3.3 Methodologies

3.3.1 Random Forest:

A Random Forest is a powerful ensemble learning technique in machine learning. It operates by creating multiple decision trees during the training phase. Each decision tree is constructed using a random subset of the dataset and a random subset of features in each

partition [42]. This randomness introduces diversity among individual trees, reducing the risk of overfitting and improving overall prediction performance.

The working process of Random Forest can be summarized in the following steps:

Selection of Random Data Points: Randomly select a subset of K data points from the training set.

Construction of Decision Trees: Build decision trees associated with the selected data points. Each decision tree is constructed independently using a different subset of the data.

Number of Trees (N): Choose the desired number N of decision trees to build.

Repeat Steps 1 and 2: Repeat the process of selecting random data points and building decision trees N times.

Prediction: For new data points, obtain predictions from each decision tree. In classification tasks, the final prediction is determined by majority voting among all decision trees. In regression tasks, predictions are typically averaged across all trees.

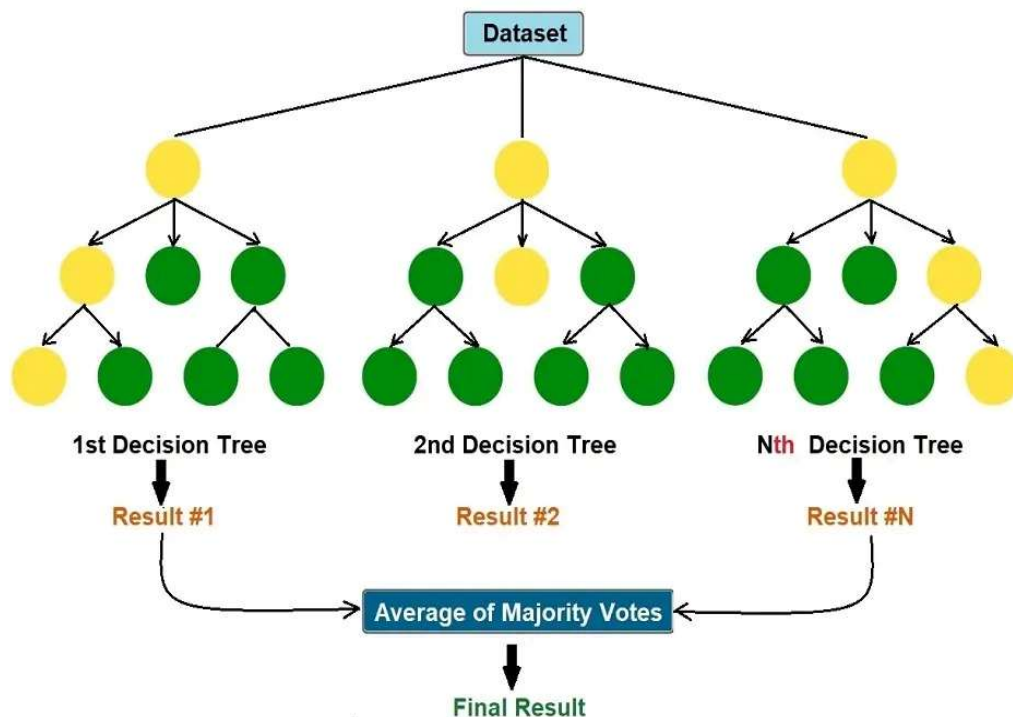


Figure 3.3.1: Random Forest

3.3.2 Naïve Bayes:

The Naïve Bayes classifier is a supervised machine learning algorithm primarily used for classification tasks, particularly in text classification. It operates on the principles of probability to classify data. Belonging to the family of generative learning algorithms, Naïve Bayes models the distribution of inputs for a given class or category. Unlike discriminative classifiers such as logistic regression, Naïve Bayes doesn't learn the importance of individual features in class differentiation; instead, it makes a "naive" assumption of feature independence [43].

Despite this simplification, Naïve Bayes classifiers are widely employed for their simplicity and efficiency in machine learning tasks. Naïve Bayes algorithms are particularly useful in classification problems, especially those involving high-dimensional data like text classification.

One of its primary advantages lies in its speed, making it ideal for applications requiring rapid model development and prediction, even with high-dimensional data.

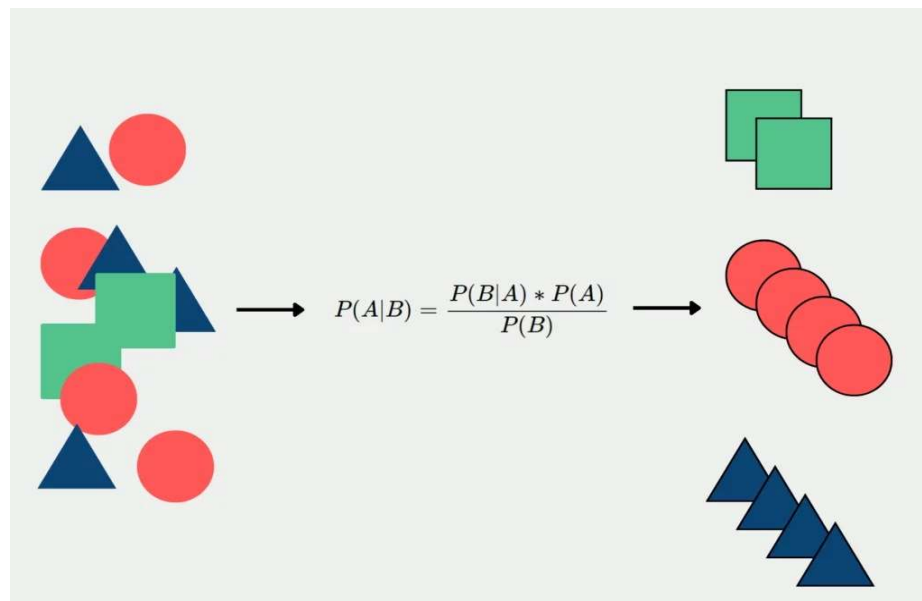


Figure 3.3.2: Naïve Bayes

3.3.3 K-Nearest Neighbors (KNN):

The K-Nearest Neighbors (KNN) algorithm, developed by Evelyn Fix and Joseph Hodges in 1951 and expanded by Thomas Cover, is a supervised machine learning approach used for both classification and regression tasks. KNN operates by assuming similarity between new data points and existing data, assigning the new point to the category most similar to its neighbors. It stores all available data and classifies new data points based on their similarity to existing ones, making it adaptable to changing datasets. The algorithm identifies the k nearest neighbors of a new data point based on a chosen distance metric like Euclidean or Manhattan distance [44,45]. In classification tasks, the predicted class is determined by majority voting among these neighbors.

The working process of KNN can be summarized as follows:

Select the Number of Neighbors (K): Determine the value of K , the number of nearest neighbors to consider.

Calculate Euclidean Distance: Compute the Euclidean distance between the new data point and each existing data point in the dataset.

Select K Nearest Neighbors: Choose the K data points with the shortest Euclidean distances to the new point.

Count Data Points in Each Category: Among the K nearest neighbors, tally the number of data points in each category.

Assign New Data Point to Category: Assign the new data point to the category with the highest count among its K nearest neighbors.

Model Finalization: The model is now ready for use.

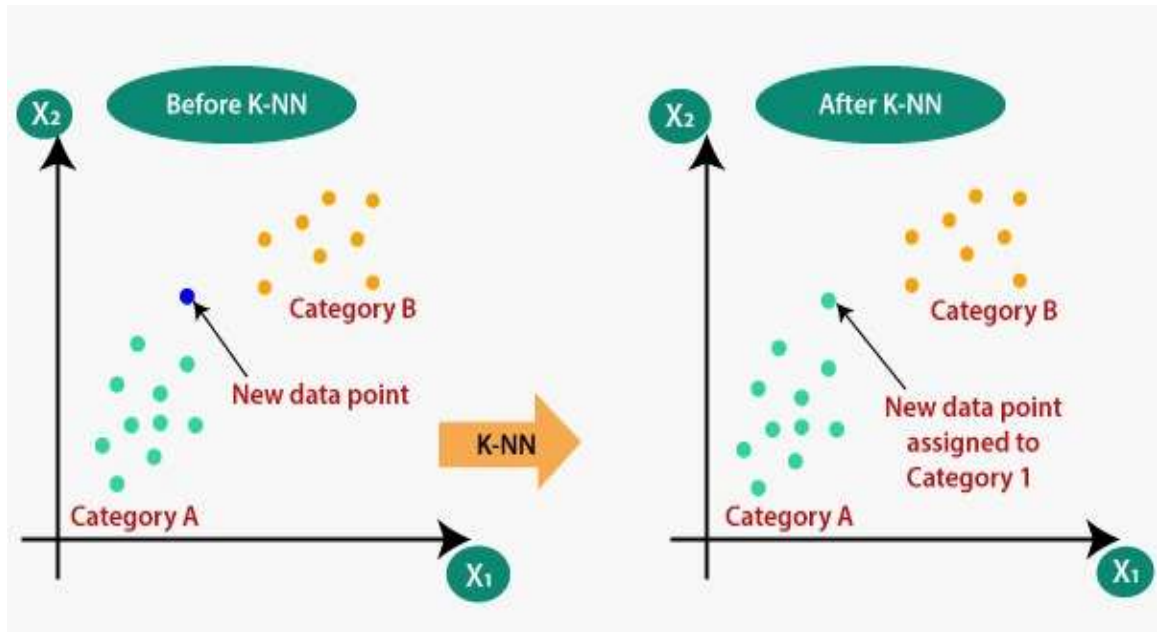


Figure 3.3.3: K-Nearest Neighbors

3.3.4 Gradient Boosting:

Gradient Boosting (GB) classifiers are machine learning methods that combine several weak learners in order to create a powerful predictor, often by using decision trees. The general idea behind it is that combining subsequent models with the earlier ones will reduce the sum of prediction errors [26]. However, a key idea within this approach is about specifying what is expected from subsequent models to minimize these errors. Moreover, the gradient of error in prediction provides targets for each instance. Consequently, at every iteration, models attempt to correct their predictions on individual training cases and decrease predictive mistakes as much as possible [46].

In this the Gradient Boosting approach is formulated as follows:

1. Model Initialization:

a) To initialize the model, commence with a first prediction that is usually set at the average of target values. This model is then saved as an “initial_model”.

2. Boosting Method:

To increase First execute an iterative process that covers every step of the boosting process and does the following:

- Determine residuals for the current model.
- Train a weak learner only on residuals.
- Sum all previous predictions multiplying them by their learning rates from weak learners.
- Incorporate weak learners that have been trained for the ensemble model.

3. The Last Model-Based Predictions:

- It starts with predictions of the first models and goes through each weak learner in the ensemble one after another until
- Adjusted Present Forecasts for a Specific Weak Learner.

4. Last Output of Prediction:

The final prediction is made after an iterative boosting procedure involving all the weak learners in the ensemble.

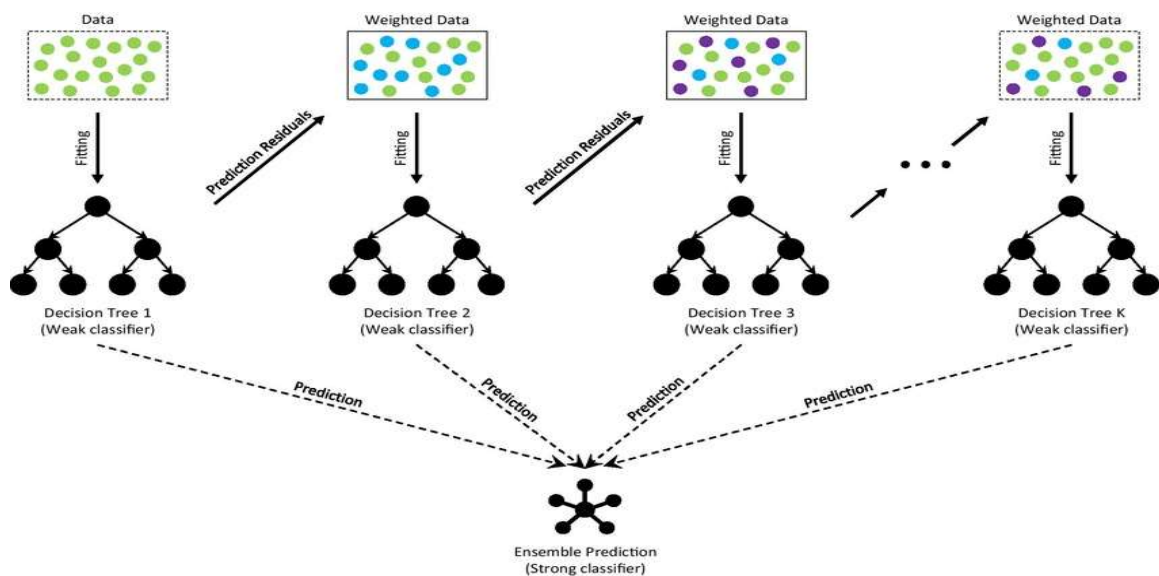


Figure 3.3.4: Gradient Boosting

3.3.5 KIDNEY SEVERITY STAGE CLASSIFICATION

One aspect of renal function is eGFR. It is an approximation of the volume of blood that flows via glomeruli, tiny kidney units, in a minute. Milli litres per minute (mL/min/1.73 m²) is the unit of measurement for eGFR [27]. While there are alternative formulae available, the MDRD and CKD-EPI equations are frequently used to estimate GFR. These algorithms occasionally take into account variables including age, gender, serum creatinine level, and race [47]. The following is the simplified MDRD equation which can be often seen in clinical practice:

$$eGFR = 175 \times Age^{(-0.203)} \times SerumCreatinine^{(-1.154)} \quad \text{-Equation (1)}$$

Where:

- Serum Creatinine: Blood creatinine levels, which are usually measured in milligrammes per deciliter, or mg/dL.
- Age: A person's age expressed in years.

GFR here is estimated through this equation for milliliters per minute and 1.73 m². However, it should be noted that the MDRD equation is commonly used but may not be accurate among different groups like those with extreme muscle mass, elderly people, or some other illnesses [48]. In order to diagnose and stage chronic kidney disease (CKD), the eGFR must be ascertained. Stages of chronic kidney disease are determined by the eGFR measurement.

Stage 1: eGFR \geq 90 mL/min/1.73m²

Stage 2: 89 \geq eGFR \geq 60 mL/min/1.73m²

Stage 3a: 59 \geq eGFR \geq 45 mL/min/1.73m²

Stage 3b: 44 \geq eGFR \geq 30 mL/min/1.73m²

Stage 4: 29 \geq eGFR \geq 15 mL/min/1.73m²

Stage 5: eGFR < 15 mL/min/1.73m²

It is of great importance to establish the extent and progression of renal damage so that these patients may live longer and healthier lives. Bear in mind, however, that eGFR could give information on renal function but it is just a part of the full picture. Other factors for assessing kidney state are urine albumin levels, clinical symptoms, and medical history among others. Therefore, interpretation of eGFR should involve other clinical outcomes.

3.4 Feasibility Study:

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately, this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an ill- conceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case three key considerations involved in the feasibility analysis are:

- Economical Feasibility
- Operational Feasibility
- Technical Feasibility

3.4.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Since the project is Machine learning based, the cost spent in executing this project would not demand cost for software and related products, as most of the products are open source and free to use. Hence the project would consume minimal cost and is economically feasible.

3.4.2 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The main purpose of this project which is based on creating an early prediction system of chronic kidney diseases using basic blood test reports. Our aim is help patients get early treatment based on these predictions which would save many lives. Thus, this is a noble cause for the sake of the society, a small step taken to achieve a secure and healthy future.

3.4.3 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Since machine learning algorithms is based on pure math there is very less requirement for any professional software. Also, most of the tools are open source. The best part is that we can run this software in any system without any software requirements which makes them highly portable. Most of the documentation and tutorials make easy to learn the technology

Chapter 4

System Requirements

4. SYSTEM REQUIREMENTS

4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. The chronic kidney disease (CKD) prediction system is designed to provide healthcare professionals with a comprehensive tool for early detection and risk assessment of CKD in patients [49]. The system facilitates the input of diverse patient data, including demographics, medical history, and laboratory test results, through user-friendly interfaces compatible with various data formats and sources, such as electronic health records and laboratory databases. It employs advanced algorithms for feature selection and model training, allowing for the identification of relevant predictors such as estimated glomerular filtration rate (eGFR), age, and proteinuria. The predictive models, built using machine learning techniques like logistic regression and random forest, enable real-time prediction of CKD risk for individual patients, enhancing clinical decision-making. Integration with existing clinical workflows ensures seamless access to patient data and prediction results, with customizable alerts and notifications to notify healthcare providers of elevated CKD risk levels. Scalability and performance are prioritized to handle large volumes of data and user requests efficiently, while accuracy and reliability are maintained through rigorous testing and validation procedures. Overall, the system aims to improve patient outcomes by facilitating early intervention and personalized management strategies for individuals at risk of developing CKD. In this system following are the functional requirements: -

1. All the data must be in the same format as a structured data.
2. The data collected will be vectorized and sent across to the classifier

4.2 Non – Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent

system properties such as reliability, response time and store occupancy. non-functional requirements to ensure its effectiveness, security, and usability. Security and privacy measures are paramount, with the system implementing robust encryption techniques to safeguard patient data during transmission and storage, and enforcing access controls to prevent unauthorized access. Interoperability is emphasized, enabling seamless integration with existing healthcare information systems and adherence to interoperability standards such as HL7 and FHIR [50]. Usability is prioritized through the provision of a user-friendly interface that accommodates users with varying levels of technical expertise and accessibility needs. Maintaining the system's reliability and accuracy is crucial, with regular updates and rigorous testing procedures to validate predictive models and ensure consistent performance. Additionally, the system is designed to be scalable and responsive, capable of handling increased data volumes and user demands without compromising on processing speed or reliability. Ethical and regulatory compliance is also central, with the system adhering to principles of transparency, fairness, and accountability in its operation and decision-making processes, and complying with regulations such as HIPAA and GDPR to protect patient rights and confidentiality. By meeting these non-functional requirements, the CKD prediction system aims to provide a secure, reliable, and user-friendly platform for healthcare professionals to effectively manage and mitigate the risks associated with chronic kidney disease. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, and the need for interoperability with other software and hardware systems.

4.2.1 Software Requirements

Operating system: Windows 7.

Coding Language: python

Tool: Jupyter Notebook, visual studio code

Libraries: Scikit-learn, Matplotlib

4.2.2 Hardware Requirements

System: Pentium i3 Processor.

Hard Disk: 500 GB.

Monitor: 15" LED

Input Devices: Keyboard, Mouse

Ram: 2 GB

Chapter 5

Design

5. DESIGN

5.1 System Design

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to meet specific requirements. It is a vital step in the development of a system and provides the backbone to handle exceptional scenarios. System design requires a systematic approach to building and engineering systems, and a good system design requires thinking about everything, from infrastructure all the way down to the data and how it's stored.

Systems Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves translating user requirements into a detailed blueprint that guides the implementation phase. The goal is to create a well-organized and efficient structure that meets the intended purpose while considering factors like scalability, maintainability, and performance.

Modules Description

A module is a bounded contiguous group of statements having a single name and that can be treated as a unit. In other words, a single block in file of blocks [49]. Chronic Kidney disease Using Machine Learning can contain the following modules.

User:

View Home page: Here user view the home page of Machine learning Methodology for chronic Kidney disease Web appellation.

View About page: In the about page, users can learn more about Machine learning Methodology for chronic kidney disease.

Select Model: To create a model that predicts disease with better accuracy, this module will help user.

Input Values: The user must provide input values for the certain fields in order to get results.

View Results: User view's the generated results from the model.

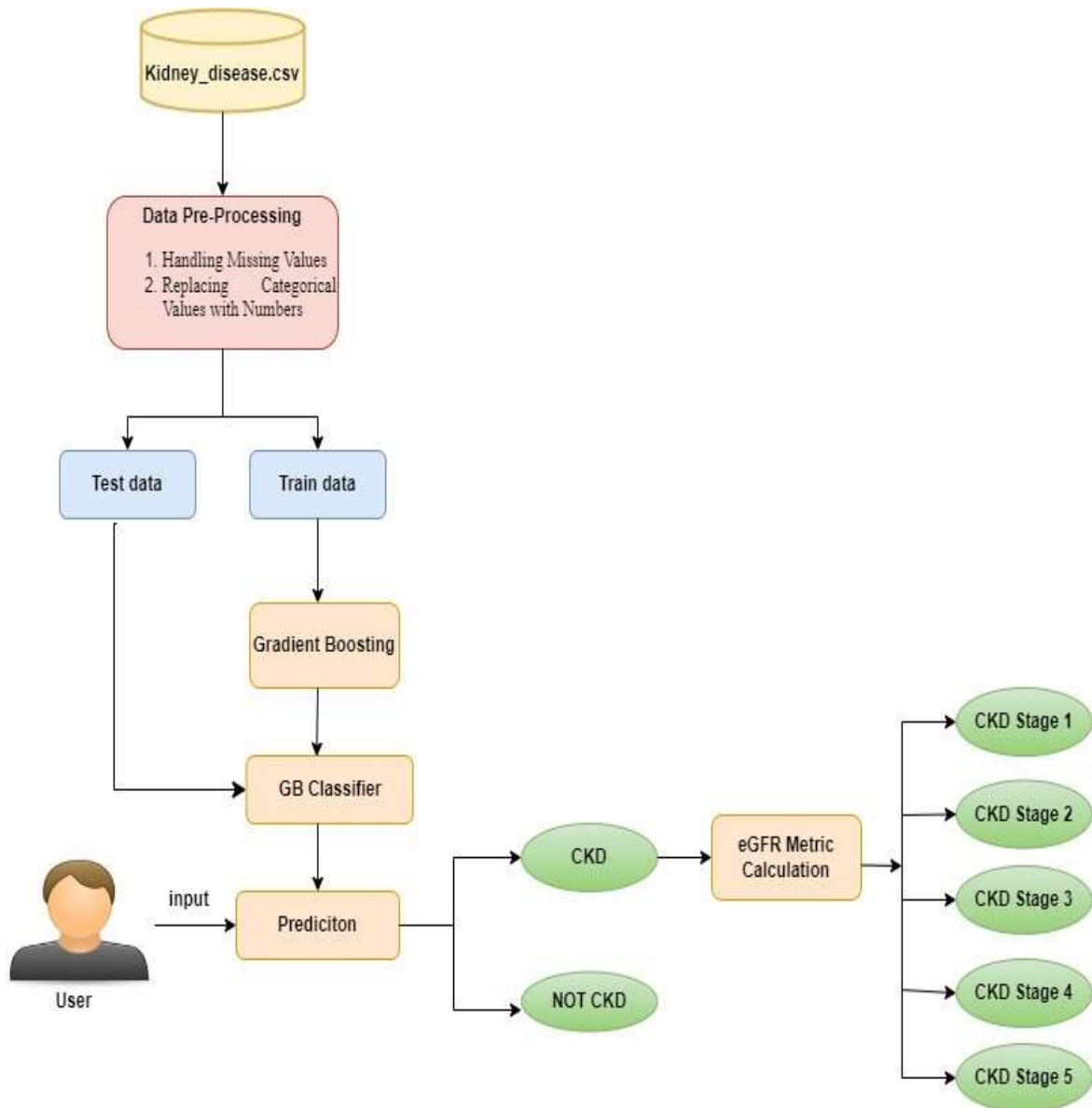


Figure 5.1: Work flow diagram

System:

Working on ckd severity dataset: System checks for data whether it is available or not and load the data in CSV file.

Pre-processing: Data need to be pre-processed according the models it helps to increase the accuracy of the model and better information about the data.

Training the data: After pre-processing, the data will split into two part as train and test data before training with the given algorithms.

Model Building: To create a model that predicts the water is pure are not pure with better accuracy, this module will help user.

Generate Results: We train the machine learning algorithm and calculate which type of Treatments needed for the patient.

5.2 UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artefacts of software system, as well as for business modelling and other non-software systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Here are some key points about UML:

Standardization: UML is standardized by the Object Management Group (OMG), which is an international consortium responsible for defining and maintaining standards for object-oriented modeling and programming.

Types of Diagrams: UML includes several types of diagrams, each serving a different purpose. Some common types include:

Class Diagrams: Represent the static structure of a system, showing classes, attributes, operations, and relationships between classes.

Use Case Diagrams: Represent the interactions between a system and its actors (users or external systems), showing the various ways the system can be used.

Sequence Diagrams: Illustrate how objects interact with each other in a particular scenario of a use case, showing the sequence of messages exchanged.

Activity Diagrams: Represent the flow of control within a system, showing the sequence of activities and decision points.

State Machine Diagrams: Show the different states that an object or system can be in, and how it transitions between those states.

Modeling Software Systems: UML is commonly used for modeling software systems during the design phase of software development. It helps developers to visualize the structure and behavior of a system before implementing it.

5.2.1 Class Diagram

In the UML, classes are represented as compartmentalized rectangles.

1. The top compartment contains the name of the class.
2. The middle compartment contains the structure of the class (attributes).
3. The bottom compartment contains the behavior of the class (operations).



Figure 5.2.1: Class Diagram

System Block

Working on CKD severity Dataset (): This refers to the system retrieving a dataset related to chronic kidney disease (CKD) severity.

Preprocessing (): This refers to the system pre-processing the data, likely cleaning and formatting the data for use in a machine learning model.

Training (): This refers to the system training a machine learning model, likely using the preprocessed data.

Model Building (): This refers to the system building a model, likely a machine learning model, to use for CKD prediction.

Generate Results (): This refers to the system generating results, likely CKD predictions based on the user-entered data.

User Block

View Home page (): This refers to the initial interaction where the system displays the home page to the user.

View About page (): This shows the user clicking to view the about page. This initiates a series of actions within the system.

Select Model (): This refers to the system prompting the user to select a model, perhaps between different pre-trained models or ones the user has built themselves.

Input Values (): This refers to the user entering values, likely features used by the model to make CKD predictions.

View Results (): This refers to the system displaying the generated results to the user.

In conclusion, this block diagram outlines the steps a user would take to interact with a system for CKD prediction and how the system would respond to those actions.

5.2.2 Use Case Diagram

A Use Case Diagram in Unified Modeling Language (UML) serves as a visual representation of the interaction between actors, which can be users or external systems,

and a system under consideration to achieve specific goals. It offers a high-level perspective on the system's functionality by illustrating the various ways users can interact with it.

The diagram consists of actors, use cases enclosed within a system boundary, communication (participation) associations between actors and use cases, and generalization relationships among use cases. It delineates both the external (actors) and internal (use cases) behaviors of the system.

Actors, representing entities external to the system, interact with it by providing input or receiving output. They encompass anyone or anything that engages with the system.

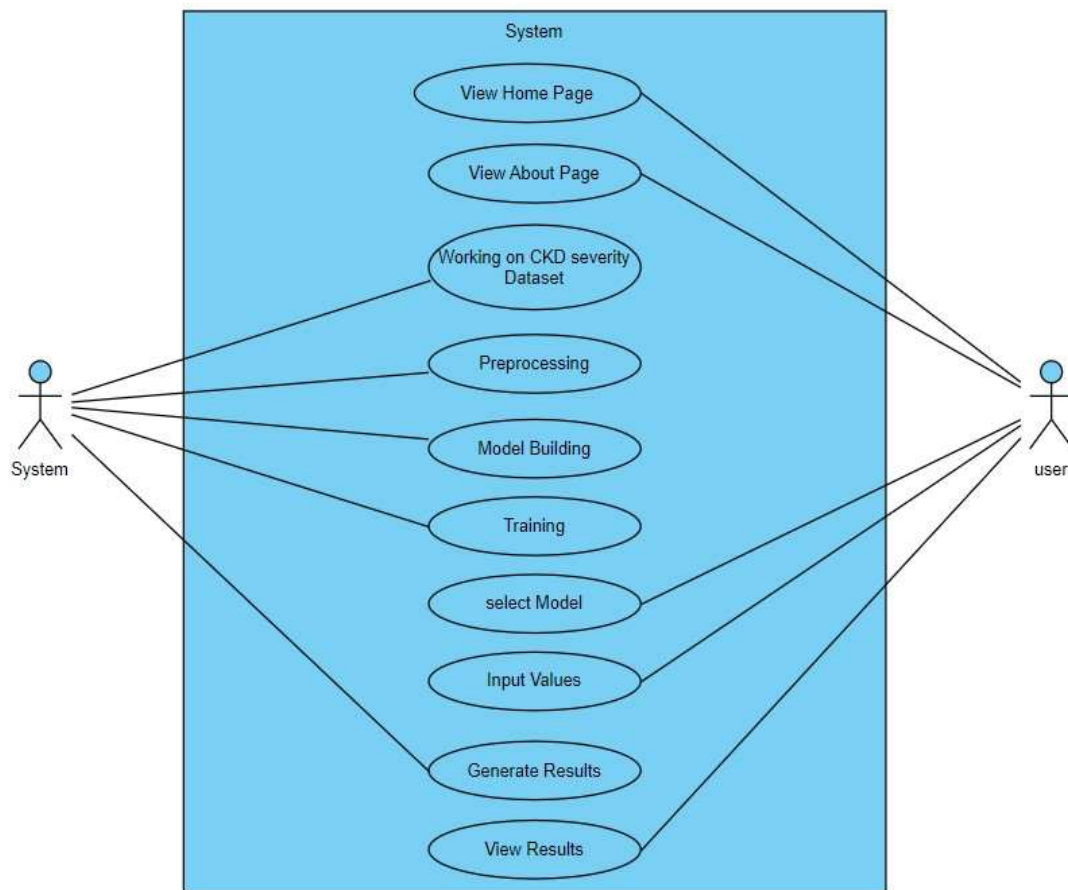


Figure 5.2.2: Use case diagram

It is a use case diagram of a system, likely related to a chronic kidney disease (CKD) prediction system. Here are the elements in the diagram:

System: This is the main entity in the diagram, represented by a rectangle. It represents the computer system that performs various tasks related to CKD prediction.

User: This is another entity, represented by a stick figure, that interacts with the system. It represents the person who uses the system to get predictions about CKD.

Arrows: Arrows show the interactions between the user and the system. They are labelled with a description of the interaction.

Here are the interactions described in the diagram:

View Home Page: This shows the initial interaction where the system displays the home page to the user.

View About Page: This shows the user clicking to view the about page. This initiates a series of actions within the system.

Working on CKD severity Dataset: The system retrieves a dataset related to CKD severity.

Preprocessing: The system preprocesses the data, likely cleaning and formatting the data for use in a machine learning model.

Model Building: The system builds a model, likely a machine learning model, to use for CKD prediction.

Select Model: The system prompts the user to select a model, perhaps between different pre-trained models or ones the user has built themselves.

Input Values: The user enters values, likely features used by the model to make CKD predictions.

Generate Results: The system generates results, which are most likely CKD predictions based on the user-entered data.

View Results: The system displays the generated results to the user.

In conclusion, this use case diagram outlines the steps a user would take to interact with a system for CKD prediction and how the system would respond to those actions.

5.2.3 Activity Diagram

An activity diagram is a variation of a special case of a state machine, in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations. The purpose of the Activity diagram is to provide a view of flows and what is going on inside a use case or among several classes. Activity diagrams contain activities, transitions between the activities, decision points, and synchronization bars.

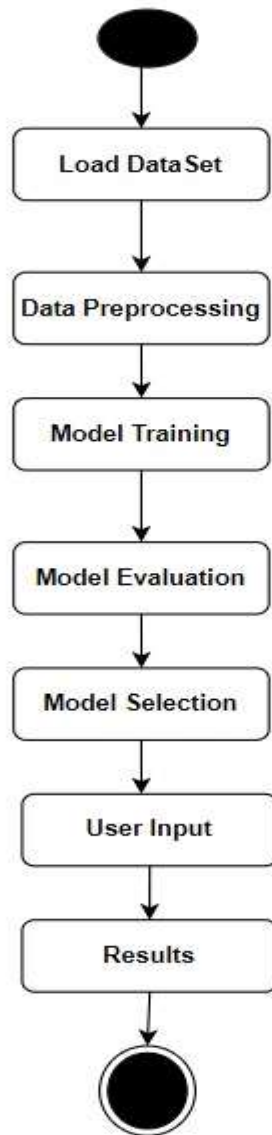


Figure 5.2.3: Activity Diagram

Here's a breakdown of the steps outlined in the diagram:

Home Page: The interaction begins with the System displaying the Home Page to the User.

View About Page: The User then clicks to view the About page. Clicking this initiates a series of steps within the System.

2.1: Load CKD Dataset: The System loads a CKD Dataset. CKD stands for chronic kidney disease, and this dataset is likely used to train a model to identify or predict CKD.

2.2: Pre-processing: The System then preprocesses the data. This likely involves cleaning and formatting the data for use in the Machine Learning model.

2.3: Training: The System trains a model, likely a machine learning model, using the pre-processed data.

2.4: Model Building: Once trained, the System builds a model.

Select Model: The System prompts the User to select a model. This could be a pre-trained model, or one the User has built themselves.

Input Values: The User then inputs values. These are likely the features used by the machine learning model to make predictions about CKD.

Generate Results: Once the User inputs the values, the System generates results. These results are likely CKD predictions based on the User entered data.

View Results: The System displays the generated results to the User.

In conclusion, this sequence diagram outlines the steps a User would take to create a website using a CMS, and how the System would respond to those actions. It also highlights the machine learning aspect of the website, where a model is trained, likely to identify chronic kidney disease.

5.2.4 Sequence Diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams.

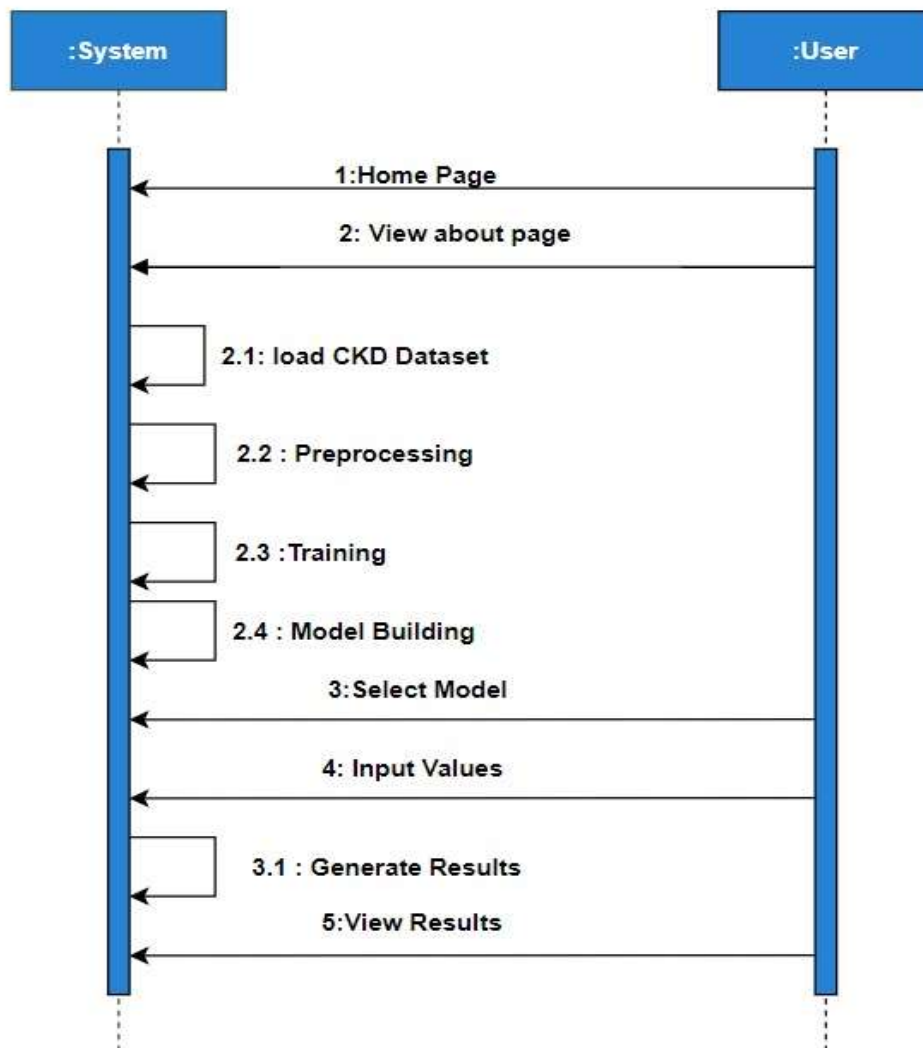


Figure 5.2.4: Sequence Diagram

It outlines the steps involved in choosing the best model for a given task. Here's a breakdown of the flowchart:

Load Data Set: The process starts with loading the data set that will be used to train and evaluate the models.

Data Preprocessing: The data is then preprocessed to ensure it's in a format suitable for training the models. This might involve cleaning the data, handling missing values, and transforming features.

Model Training: Different machine learning models are trained on the preprocessed data.

Model Evaluation: Each trained model is evaluated on a separate test dataset to assess its performance.

Model Selection: The model with the best performance on the test dataset is selected for further use.

User Input: The user can optionally provide input to influence the model selection process. This could involve specifying a preference for a particular type of model or setting a performance threshold.

Results: The final output is the selected model, which can then be used to make predictions or classifications on new data.

Chapter 6

Implementation

6. IMPLEMENTATION

Reading Dataset:

```
dataset = pd.read_csv("Kidney_data.csv")  
  
dataset
```

The provided Python code reads a dataset named "Kidney_data.csv" using the Pandas library and assigns it to the variable dataset. The dataset is then displayed in the output. However, since I can't directly execute code, I can't view the content of the dataset. But I can help you craft a description based on common assumptions about the structure and content of such datasets.

Checking Missing (NaN) Values:

```
dataset.isnull().sum()
```

The provided Python code snippet checks for missing values (NaN) in the dataset using the isnull() function, followed by the sum() function to count the number of missing values in each column. This operation likely generates an output showing the count of missing values for each attribute or feature in the dataset. A description based on this code would include insights into the presence of missing data within the dataset. It would reveal which columns have missing values and how many missing values are present in each column. Understanding the extent of missing data is crucial for data preprocessing steps such as data imputation or removal of incomplete records.

#Replacing Categorical values with numbers:

```
dataset['rbc'].value_counts()  
  
dataset['rbc'] = dataset['rbc'].replace(to_replace = {'normal' : 0, 'abnormal' : 1})  
  
dataset['pc'].value_counts()  
  
dataset['pc'] = dataset['pc'].replace(to_replace = {'normal' : 0, 'abnormal' : 1})  
  
dataset['pcc'].value_counts()  
  
dataset['pcc'] = dataset['pcc'].replace(to_replace = {'notpresent':0,'present':1})
```

```

dataset['ba'].value_counts()

dataset['ba'] = dataset['ba'].replace(to_replace = {'notpresent':0,'present':1})

dataset['htn'].value_counts()

dataset['htn'] = dataset['htn'].replace(to_replace = {'yes' : 1, 'no' : 0})

dataset['dm'].value_counts()

dataset['dm'] = dataset['dm'].replace(to_replace = {'\tyes': 'yes', ' yes': 'yes', '\tno': 'no'})

dataset['dm'] = dataset['dm'].replace(to_replace = {'yes' : 1, 'no' : 0})

dataset['cad'].value_counts()

dataset['cad'] = dataset['cad'].replace(to_replace = {'\tno': 'no'})

dataset['cad'] = dataset['cad'].replace(to_replace = {'yes' : 1, 'no' : 0})

dataset['appet'].unique()

dataset['appet'] = dataset['appet'].replace(to_replace={'good':1,'poor':0,'no':np.nan})

dataset['pe'].value_counts()

dataset['pe'] = dataset['pe'].replace(to_replace = {'yes' : 1, 'no' : 0})

dataset['ane'].value_counts()

dataset['ane'] = dataset['ane'].replace(to_replace = {'yes' : 1, 'no' : 0})

dataset['classification'].value_counts()

dataset['classification'] = dataset['classification'].replace(to_replace={'ckd\t': 'ckd'})

```

The provided code snippet demonstrates the process of replacing categorical values with numerical equivalents in various columns of the dataset. Each categorical attribute undergoes a transformation where specific categories are mapped to corresponding numerical values. For instance, attributes like 'rbc' (Red Blood Cells), 'pc' (Pus Cell), 'pcc' (Pus Cell Clumps), and 'ba' (Bacteria) are converted from categorical to numerical representations, facilitating numerical analysis. Additionally, binary categorical attributes such as 'htn' (Hypertension), 'dm' (Diabetes Mellitus), 'cad' (Coronary Artery Disease),

'appet' (Appetite), 'pe' (Pedal Edema), and 'ane' (Anemia) are encoded into binary values (0 or 1) for computational ease. Finally, the 'classification' column, presumably denoting the disease classification, undergoes standardization by replacing 'ckd\t' with 'ckd', ensuring uniformity. This categorical-to-numerical transformation streamlines subsequent analytical procedures and enhances the dataset's compatibility with machine learning algorithms.

#Handling Null Values

```
features = ['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',  
            'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',  
            'appet', 'pe', 'ane']  
  
for feature in features:  
    dataset[feature] = dataset[feature].fillna(dataset[feature].median())  
  
dataset.isnull().any().sum()
```

The provided code segment demonstrates a method for handling null values within the dataset. It iterates through a predefined list of features, replacing any missing values in each feature with the median value of that particular feature. This approach ensures that missing values across various attributes are imputed with a central tendency measure, maintaining data integrity and completeness. After the imputation process, the code verifies the dataset to ascertain if any null values remain, with the intention of confirming that all missing values have been effectively handled. This systematic approach to null value handling mitigates the risk of bias and loss of information, thereby enhancing the dataset's suitability for subsequent analysis and modeling tasks.

Train Test Split:

```
from sklearn.model_selection import train_test_split  
  
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size=0.2, random_state=33)  
  
print(X_train.shape)
```

```
print(X_test.shape)
```

The code segment demonstrates the implementation of train-test split using the `'train_test_split'` function from the scikit-learn library. It partitions the dataset into training and testing sets, denoted as `X_train`, `X_test`, `y_train`, and `y_test`, where `X` represents the input features and `y` represents the target variable. The parameter `'test_size=0.2'` indicates that 20% of the data is allocated for testing, while the remaining 80% is used for training. Additionally, `'random_state=33'` ensures reproducibility by fixing the random seed. Printing the shapes of the training and testing sets offers insights into the dimensions of the data subsets, providing an overview of the data distribution between training and testing partitions. This train-test split facilitates model development, enabling evaluation of model performance on unseen data, thereby enhancing the robustness and generalization capabilities of the developed models.

#Gradient Boosting

```
from sklearn.ensemble import GradientBoostingClassifier
```

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

```
GB = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3,  
min_samples_split=2, min_samples_leaf=1, random_state=42)
```

```
# Train the classifier
```

```
GB = GB.fit(X_train, y_train)
```

```
# Predictions
```

```
y_pred = GB.predict(X_test)
```

```
# Performance
```

```
print('Accuracy:', accuracy_score(y_test, y_pred))
```

```
print('Confusion Matrix:\n', confusion_matrix(y_test, y_pred))
```

```
print('Classification Report:\n', classification_report(y_test, y_pred))
```

The provided code segment implements a Gradient Boosting Classifier, a popular ensemble learning technique, using the scikit-learn library. The GradientBoostingClassifier is configured with specified hyperparameters including the number of estimators (trees) set to 100, learning rate of 0.1, maximum depth of trees as 3, minimum samples split as 2, minimum samples leaf as 1, and a random state for reproducibility. The classifier is trained on the training data (X_train, y_train) and subsequently used to make predictions on the test data (X_test), which are then evaluated for performance metrics including accuracy, confusion matrix, and classification report. Gradient Boosting is a powerful ensemble learning method that builds a strong predictive model by combining multiple weak learners (decision trees in this case) sequentially, with each subsequent learner focusing on the errors made by the previous ones. By iteratively minimizing the errors, Gradient Boosting enhances the model's predictive performance and robustness. The accuracy score provides a measure of the overall correctness of predictions, while the confusion matrix offers insights into the classifier's performance across different classes. Additionally, the classification report furnishes a detailed breakdown of performance metrics including precision, recall, and F1-score for each class, offering a comprehensive assessment of the classifier's predictive capabilities. Overall, Gradient Boosting serves as a potent tool for tackling classification tasks, particularly when dealing with complex datasets requiring high predictive accuracy.

Chapter 7

Results

7. RESULTS

7.1 Dataset

The CKD dataset, meticulously curated from the Machine Learning Repository at the University of California, Irvine, serves as a foundational cornerstone for refining and assessing our proposed model. Comprising 400 samples meticulously compiled and quality-checked, this dataset represents a pivotal asset in our quest to develop effective algorithms for the prediction and diagnosis of chronic kidney disease (CKD) [51]. By tapping into this repository, we ensure access to standardized, reliable data encompassing diverse patient demographics, clinical parameters, and disease progression indicators. This rich and diverse dataset forms an indispensable resource for training and validating machine learning models, empowering us to derive invaluable insights that hold the potential to significantly enhance CKD management and patient care. Our comprehensive evaluation of the recommended model's performance and predictive accuracy is facilitated by the CKD dataset sourced from the ML Repository at UC Irvine. This dataset encapsulates a wealth of features, including biochemical measurements, patient demographics, and medical history, offering a comprehensive panorama of CKD manifestations.

id																								
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
id	age	bp	sg	al	su	rbc	pc	pcc	ba	bgr	bu	sc	sod	pot	hemo	pcv	wc	rc	htn	dm	cad	appet	pe	
1	0	48	80	1.02	1	0	normal	notpresent	notpresent	121	36	1.2			15.4	44	7800	5.2	yes	yes	no	good	nd	
2	1	7	50	1.02	4	0	normal	notpresent	notpresent		18	0.8			11.3	38	6000		no	no	no	good	nd	
3	2	62	80	1.01	2	3	normal	notpresent	notpresent	423	53	1.8			9.6	31	7500		no	yes	no	poor	nd	
4	3	48	70	1.005	4	0	normal	abnormal	present	notpresent	117	56	3.8	111	2.5	11.2	32	6700	3.9	yes	no	no	poor	ye
5	4	51	80	1.01	2	0	normal	normal	notpresent	notpresent	106	26	1.4			11.6	35	7300	4.6	no	no	no	good	nd
6	5	60	90	1.015	3	0		notpresent	notpresent	74	25	1.1	142	3.2	12.2	39	7800	4.4	yes	yes	no	good	ye	
7	6	68	70	1.01	0	0	normal	notpresent	notpresent	100	54	24		104	4	12.4	36			no	no	no	good	nd
8	7	24		1.015	2	4	normal	abnormal	notpresent	notpresent	410	31	1.1			12.4	44	6900	5	no	yes	no	good	ye
9	8	52	100	1.015	3	0	normal	abnormal	present	notpresent	138	60	1.9			10.8	33	9600	4	yes	yes	no	good	nd
10	9	53	90	1.02	2	0	abnormal	abnormal	present	notpresent	70	107	7.2	114	3.7	9.5	29	12100	3.7	yes	yes	no	poor	nd
11	10	50	60	1.01	2	4		abnormal	present	notpresent	490	55	4			9.4	28		yes	yes	no	good	nd	
12	11	63	70	1.01	3	0	abnormal	abnormal	present	notpresent	380	60	2.7	131	4.2	10.8	32	4500	3.8	yes	yes	no	poor	ye
13	12	68	70	1.015	3	1		normal	present	notpresent	208	72	2.1	138	5.8	9.7	28	12200	3.4	yes	yes	yes	poor	ye
14	13	68	70					notpresent	notpresent	98	86	4.6	135	3.4		9.8			yes	yes	yes	poor	ye	
15	14	68	80	1.01	3	2	normal	abnormal	present	present	157	90	4.1	130	6.4	5.6	16	11000	2.6	yes	yes	yes	poor	ye
16	15	40	80	1.015	3	0		normal	notpresent	notpresent	76	162	9.6	141	4.9	7.6	24	3800	2.8	yes	no	no	good	nd
17	16	47	70	1.015	2	0		normal	notpresent	notpresent	99	46	2.2	138	4.1	12.6			no	no	no	good	nd	
18	17	47	80					notpresent	notpresent	114	87	5.2	139	3.7	12.1				yes	no	no	poor	nd	
19	18	60	100	1.025	0	3		normal	notpresent	notpresent	263	27	1.3	135	4.3	12.7	37	11400	4.3	yes	yes	yes	good	nd
20	19	62	60	1.015	1	0		abnormal	present	notpresent	100	31	1.6			10.3	30	5300	3.7	yes	no	yes	good	nd
21	20	61	80	1.015	2	0	abnormal	abnormal	notpresent	notpresent	173	148	3.9	135	5.2	7.7	24	9200	3.2	yes	yes	yes	poor	ye
22	21	60	90					notpresent	notpresent	180	76	4.5			10.9	32	6200	3.6	yes	yes	yes	good	nd	
23	22	48	80	1.025	4	0	normal	abnormal	notpresent	notpresent	95	163	7.7	136	3.8	9.8	32	6900	3.4	yes	no	no	good	nd
24	23	21	70	1.01	0	0		normal	notpresent	notpresent									no	no	no	poor	nd	
25	24	42	100	1.015	4	0	normal	abnormal	notpresent	present		50	1.4	129	4	11.1	39	8300	4.6	yes	no	no	poor	nd
26	25	61	60	1.025	0	0		normal	notpresent	notpresent	108	75	1.9	141	5.2	9.9	29	8400	3.7	yes	yes	no	good	nd
27	26	75	80	1.015	0	0		normal	notpresent	notpresent	156	45	2.4	140	3.4	11.6	35	10300	4	yes	yes	no	poor	nd
28	27	69	70	1.01	3	4	normal	abnormal	notpresent	notpresent	264	87	2.7	130	4	12.5	37	9600	4.1	yes	yes	yes	good	ye
29	28	75	70		1	3		notpresent	notpresent	123	31	1.4							no	yes	no	good	nd	
30	29	68	70	1.005	1	0	abnormal	abnormal	present	notpresent		28	1.4			12.9	38		no	no	yes	good	nd	
31	30		70					notpresent	notpresent	93	155	7.3	132	4.9					yes	yes	no	good	nd	
32	31	73	90	1.015	3	0		abnormal	present	notpresent	107	33	1.5	141	4.6	10.1	30	7800	4	no	no	no	poor	nd
33	32	64	70	1.01	4	4		normal	notpresent	notpresent	150	30	1.6	133	4.0	11.3	34	8600	4	yes	yes	no	good	nd
Kidney_data																								

Figure 7.1: Dataset

Leveraging advanced machine learning techniques, our aim is to unravel the latent patterns within this dataset and craft a robust model capable of accurately discerning CKD onset, progression, and associated risk factors. Through meticulous scrutiny against this standardized benchmark, we ensure the reliability and efficacy of our model in real-world clinical settings, thus laying the groundwork for the development of more effective CKD management strategies and ultimately fostering improved patient outcomes. The research effort involved experimenting on the UCI CKD dataset, where encoding was done for categorical features while missing values were imputed using various imputation techniques. This section gives a detailed report of the experimental procedure and its results, which makes use of five boosting algorithms to predict CKD cases. 80% of the data sets were used for training gradient boosting algorithm while the remaining 20% were used for testing and validating their performance. In their subsequent evaluations, accuracy, recall, precision and F1-score [30] are meticulously analyzed as performance metrics.

7.2 Performance Evaluation:

Confusion Matrix:

A confusion matrix is a table that shows the performance of a model in classification. It sums up how many predictions are true and false by modeling them into four categories namely: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The structure of a confusion matrix looks like this:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

In simpler terms;

True Positive (TP): Number of instances predicted correctly as positive.

True Negative (TN): Number of instances predicted correctly as negative.

False Positive (FP): Number of instances wrongly predicted as positive (Type I error).

False Negative (FN): Number of instances wrongly predicted as negative (Type II error).

The model's confusion matrix shows how well it performs and is also used to estimate numerous evaluation measures i.e., precision, accuracy, recall and F1-score.

Accuracy: The classification effectiveness is measured by the accuracy level which is obtained from the percentage of instances accurately predicted in relation to the whole dataset.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad - \text{Equation (2)}$$

Recall: Recall measures CKD positive cases that have been correctly labeled as such with respect to all CKD cases in the data set.

$$Recall = \frac{TP}{TP+FN} \times 100 \quad - \text{Equation (3)}$$

Precision: Precision will look at what percentage of people with CKD are truly diagnosed as being so.

$$Precision = \frac{TP}{TP+FP} \times 100 \quad - \text{Equation (4)}$$

F1-Score: F-measure is a harmonic mean of precision and recall; it gives an alternative measure for examining predictive ability of a model vital in research analysis.

$$F1 - score = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + r} \times 100 \quad - \text{Equation (5)}$$

Classification Report:

	Precision	Recall	F1-score	Support
0	1.00	0.98	0.99	41
1	0.97	1.00	0.99	39
Accuracy			0.99	80
Macro avg	0.99	0.99	0.99	80
Weighted avg	0.99	0.99	0.99	80

Table 7.2: Classification Report

Table 7.2 illustrates how diverse configurations done on individual model parameters lead to different outcomes. This table presents the results derived from experiments performed on each model, including measures like accuracy, F1-score, precision, recall as well as presenting confusion matrices.

7.3 Confusion Matrix

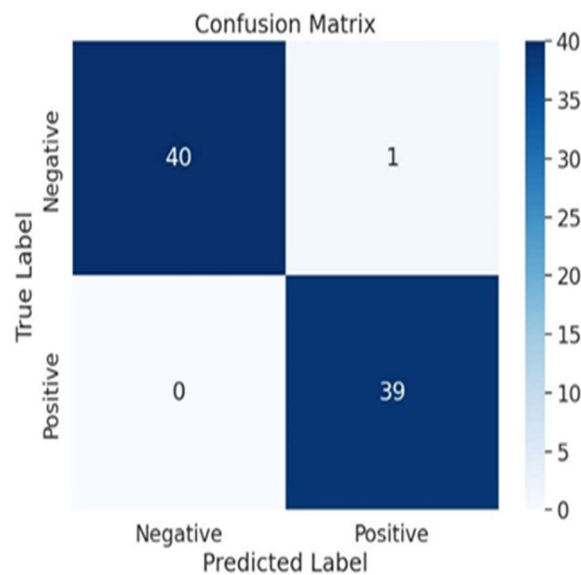


Figure 7.3: Confusion Matrix

7.4 Accuracy Comparison

The bar graph illustrates a comparison of the accuracy of five distinct machine learning algorithms: Random Forest, Naive Bayes, KNN, Gradient Boosting, and ANN. Each bar represents one algorithm and is uniquely colored to distinguish between them. The x-axis

enumerates the algorithms by name, while the y-axis depicts accuracy, ranging from 0.0 to 1.0. The analysis demonstrates that Gradient Boosting achieves the highest accuracy among the five algorithms in this scenario.

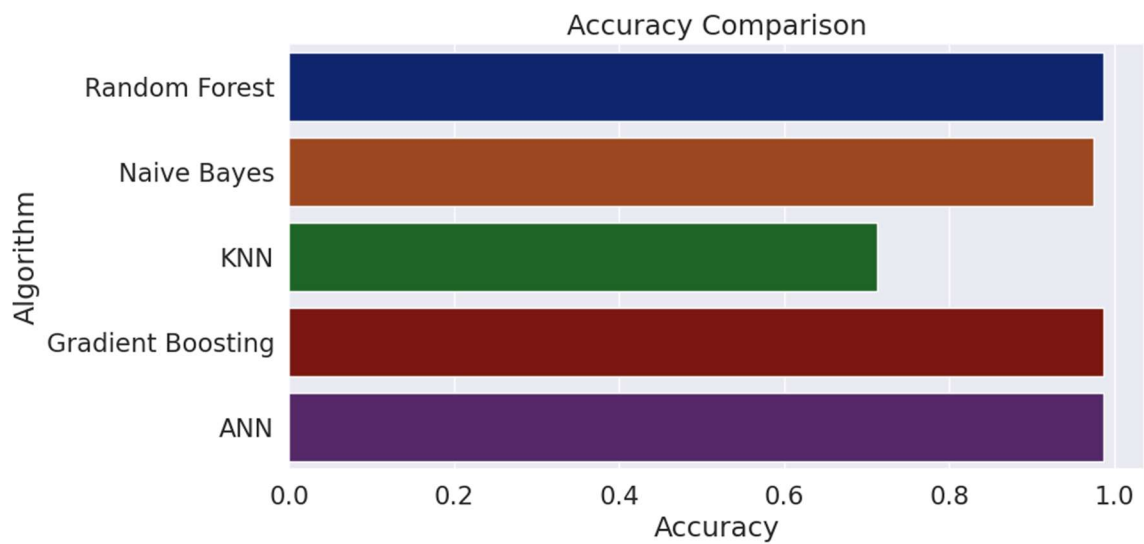


Figure 7.4: Accuracy Comparison

7.5 Performance Metrics

Classifier used	Precision	Recall	F1-score	Accuracy achieved (%)
Random Forest	1.00	0.98	0.99	97
Naive Bayes	0.95	1.00	0.98	96.92
KNN	0.69	0.80	0.74	71.25
Gradient Boosting	1.00	0.98	0.99	98.75

Table 7.5 Performance metrics

7.6 Comparison with Other Models

Reference	Accuracy	Precision	Recall	F1-Score
Pal Saurabh et al. [16]	97%	0.99	0.98	0.98
Debal, Adeba, and Sitote et al. [17]	96%	0.97	0.94	0.96
Dritsas & Trigka et al. [18]	97.4%	0.97	0.97	0.97
Almustafa et al. [19]	95.75%	0.96	0.95	0.95
Chittora et al. [21]	96.4%	0.981	0.913	0.964
Proposed Model	98.75%	1.00	0.98	0.99

Table 7.6: Comparison with Other Models

7.7 Outputs

A Predictive Framework for Chronic Kidney Disease Using Machine Learning Algorithms

Age:

Blood Pressure:

Specific Gravity:

Albumin:

Sugar:

Red Blood Cells:

Pus Cell:

Pus Cell Clumps:

Bacteria:

Blood Glucose Random:

Blood Urea:

Serum Creatinine:

Sodium:

Potassium:

Hemoglobin:

White Blood Cell Count:

Red Blood Cell Count:

Hypertension:

Chronic kidney disease, also called chronic kidney failure, involves a gradual loss of kidney function. Your kidneys filter wastes and excess fluids from your blood, which are then removed in your urine. Advanced chronic kidney disease can cause dangerous levels of fluid, electrolytes and wastes to build up in your body. In the early stages of chronic kidney disease, you might have few signs or symptoms. You might not realize that you have kidney disease until the condition is advanced.




Figure 7.7.1: Web Interface

The Figure 7.7.1 illustrates the framework likely analyzes various factors including age, blood pressure, blood sugar, and red blood cell count to predict CKD. Machine learning algorithms can then be used to identify patterns in the data and predict the likelihood of someone developing CKD.

A Predictive Framework for Chronic Kidney Disease Using Machine Learning Algorithms

Age:	48	Blood Pressure:	80
Specific Gravity:	1.02	Albumin:	1
Sugar:	0	Red Blood Cells:	0
Pus Cell:	0	Pus Cell Clumps:	0
Bacteria:	0	Blood Glucose Random:	121
Blood Urea:	36	Serum Creatinine:	1.2
Sodium:	0	Potassium:	0
Hemoglobin:	15.4	White Blood Cell Count:	7800
Red Blood Cell Count:	5.2	Hypertension:	1

Chronic kidney disease, also called chronic kidney failure, involves a gradual loss of kidney function. Your kidneys filter wastes and excess fluids from your blood, which are then removed in your urine. Advanced chronic kidney disease can cause dangerous levels of fluid, electrolytes and wastes to build up in your body. In the early stages of chronic kidney disease, you might have few signs or symptoms. You might not realize that you have kidney disease until the condition is advanced.



Figure 7.7.2: User Input

The Figure 7.7.2 illustrates CKD prediction, users input relevant medical data such as age, blood pressure, serum creatinine levels, and other pertinent health information. This input is then processed by a predictive model, which analyzes the data to estimate the likelihood of CKD development or progression. The outcome assists healthcare professionals in early diagnosis and treatment planning.

Chronic Kidney Disease Prediction

Oops! 😞

You have CHRONIC KIDNEY DISEASE.

Please Consult Doctor.

CKD Stage Calculator

Parameter	Value
Serum Creatinine Level (mg/dL):	<input type="text"/>
Age (years):	<input type="text"/>

Figure 7.7.3: Person having CKD

Figure 7.7.3 describes Diagnosing CKD necessitates professional medical evaluation. If CKD is a concern, it's crucial to seek guidance from a healthcare provider for accurate diagnosis and appropriate management. A doctor can conduct comprehensive tests, including blood and urine analyses, to assess kidney function and determine the presence of CKD. Seeking timely medical advice ensures proper care and treatment planning.

Chronic Kidney Disease Prediction

Oops! 😞

You have CHRONIC KIDNEY DISEASE.

Please Consult Doctor.

CKD Stage Calculator

Parameter	Value
Serum Creatinine Level (mg/dL):	<input type="text" value="1.2"/>
Age (years):	<input type="text" value="48"/>

The estimated GFR is: 64.62 mL/min/1.73m²

The CKD stage is: Stage 2

Figure 7.7.4: Person having CKD with Severity Stage

Figure 7.7.4 describes the system estimates kidney function based on user-inputted data such as serum creatinine level and age. Using this information, it calculates an estimated glomerular filtration rate (eGFR) and predicts the stage of chronic kidney disease (CKD). This predictive capability aids in early detection and management of CKD, facilitating proactive healthcare interventions.

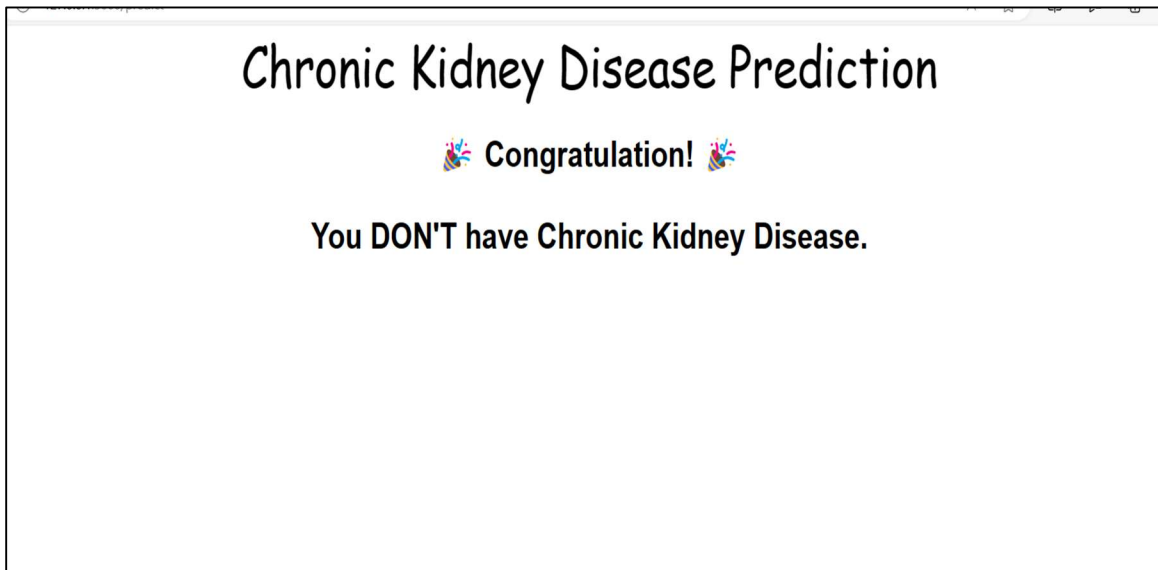


Figure 7.7.5: Person without having CKD

Figure 7.7.5 shows the tool provides a message: "Congratulations! You DON'T have chronic kidney disease." However, it emphasizes that this tool is not a replacement for professional medical advice. If there are concerns about kidney health, it's crucial to consult a doctor for thorough evaluation and guidance.

Chapter 8

Social Impact

8. SOCIAL IMPACT

Introducing a predictive framework for chronic kidney disease (CKD) using machine learning algorithms can have several social impacts:

1. **Early Detection and Prevention:** One of the most significant social impacts is the potential for early detection of CKD. Machine learning algorithms can analyze a wide range of patient data to identify individuals at risk of developing CKD before symptoms manifest. Early detection allows for timely intervention and preventive measures, which can significantly improve patient outcomes and reduce healthcare costs associated with advanced CKD treatment.
2. **Improved Patient Care:** Implementing a predictive framework can enhance patient care by providing healthcare professionals with actionable insights into patient health. With early predictions of CKD risk, healthcare providers can personalize treatment plans, monitor high-risk patients more closely, and intervene proactively to slow disease progression.
3. **Reduced Healthcare Disparities:** Access to predictive frameworks for CKD using machine learning can help reduce healthcare disparities by enabling early detection and intervention in underserved populations. By reaching individuals who may have limited access to healthcare resources or face socioeconomic barriers, the framework can contribute to more equitable healthcare delivery.
4. **Resource Allocation Optimization:** Predictive frameworks can assist healthcare systems in optimizing resource allocation by identifying high-risk patients who require intensive monitoring or intervention. By prioritizing care for patients at higher risk of CKD progression, healthcare facilities can allocate resources more efficiently and effectively, improving overall healthcare system performance.
5. **Empowering Patients:** Providing patients with information about their CKD risk empowers them to take an active role in their healthcare management. Patients can make informed lifestyle changes, adhere to treatment plans more effectively, and engage in

shared decision-making with healthcare providers, leading to better health outcomes and improved quality of life.

6. Research and Public Health Initiatives: Data collected through predictive frameworks can contribute to research efforts aimed at better understanding CKD risk factors, disease progression, and treatment outcomes. This information can inform public health initiatives focused on CKD prevention, early intervention, and population health management strategies.

7. Ethical and Privacy Considerations: Implementing predictive frameworks for CKD requires careful consideration of ethical and privacy implications. Safeguarding patient data, ensuring transparency in algorithmic decision-making, and addressing potential biases are essential to maintaining trust and integrity in healthcare delivery.

Overall, a predictive framework for CKD using machine learning algorithms has the potential to positively impact public health outcomes, enhance healthcare delivery, and empower individuals to manage their kidney health proactively. However, it is crucial to address ethical, social, and privacy considerations to maximize the benefits while minimizing potential risks.

Chapter 9

Conclusion & Future Work

9.CONCLUSION & FUTURE WORK

9.1 Conclusion:

The developed CKD prognosis model utilizing gradient boosting demonstrates promising performance in early detection of chronic kidney disease (CKD) and shows proficiency in utilizing various evaluation metrics such as recall, precision, F1 score, and accuracy. Its ability to effectively incorporate both categorical and non-categorical features enhances its capability to identify individuals at risk of CKD, facilitating prompt intervention and preventive measures. Moreover, the model's potential to continuously improve through feature selection approaches indicates a pathway for enhancing its clinical utility. By selecting an optimal subset of characteristics for model refinement, the system can further enhance its predictive accuracy and efficiency. Furthermore, the integration of guided machine learning techniques with feature selection approaches offers a systematic method for enhancing the model's performance. This iterative process allows for the identification and incorporation of relevant features, thereby refining the model's predictive capabilities. Looking ahead, leveraging unsupervised or deep learning techniques holds promise for identifying performance variations and enhancing the model's robustness. These advanced methodologies can contribute to the continuous evolution and refinement of the CKD prognosis model, ensuring its effectiveness in real-world clinical settings.

9.2 Future Work:

Future work in the development of the CKD prognosis model could focus on several areas to enhance its clinical utility and effectiveness

1. Refinement of Feature Selection Approaches: Further exploration and refinement of feature selection methodologies can help identify the most informative features for model building. This iterative process should aim to enhance the model's predictive accuracy and interpretability.

2.Integration of Unsupervised and Deep Learning Techniques: Incorporating unsupervised or deep learning techniques can offer insights into performance variations and further improve the model's robustness. These advanced methodologies can help uncover complex patterns within the data and enhance the model's predictive capabilities.

3.Real-time Patient Monitoring and Self-assessment: Exploring the feasibility of deploying the CKD prognosis model on mobile devices for real-time patient monitoring and self-assessment could enhance patient engagement and facilitate early intervention. This would require the development of user-friendly interfaces and secure data transmission protocols.

4.Validation in Diverse Patient Populations: Conducting extensive validation studies in diverse patient populations can ensure the generalizability and reliability of the model across different demographic groups and healthcare settings. This would involve collaborating with healthcare providers and institutions to collect comprehensive patient data for model validation. Overall, continued research and development efforts in these areas can contribute to the ongoing improvement and adoption of the CKD prognosis model, ultimately benefiting both patients and medical professionals in the prevention and management of chronic kidney disease.

Chapter 10

Bibliography

10. BIBLIOGRAPHY

- [1] Pal, Saurabh. "Chronic kidney disease prediction using machine learning techniques." *Biomedical Materials & Devices* 1.1 (2023): 534-540.
- [2] Debal, Dibaba Adeba, and Tilahun Melak Sitote. "Chronic kidney disease prediction using machine learning techniques." *Journal of Big Data* 9.1 (2022): 109.
- [3] Dritsas, Elias, and Maria Trigka. "Machine learning techniques for chronic kidney disease risk prediction." *Big Data and Cognitive Computing* 6.3 (2022): 98.
- [4] Abdel-Kader, K. Symptoms with or because of Kidney Failure? *Clin. J. Am. Soc. Nephrol.* 2022, 17, 475–477.
- [5] D. Ramos et al., Using decision tree to select forecasting algorithms in distinct electricity consumption context of an office building. *Energy Rep.* 8, 417–422 (2022)
- [6] H.E. Song et al., Predictive modeling of groundwater nitrate pollution and evaluating its main impact factors using random forest. *Chemosphere* 290, 133388 (2022)
- [7] H.U. Rongyao et al., Multi-task multi-modality SVM for early COVID-19 diagnosis using chest CT data. *Inf. Proc. Manag.* 59(1), 102782 (2022)
- [8] X.U. Ankun et al., Artificial neural network (ANN) modeling for the prediction of odor emission rates from landfill working surface. *Waste Manag.* 138, 158–171 (2022)
- [9] Almustafa, Khaled Mohamad. "Prediction of chronic kidney disease using different classification algorithms." *Informatics in Medicine Unlocked* 24 (2021): 100631.
- [10] Ifraz, Gazi Mohammed, et al. "Comparative analysis for prediction of kidney disease using intelligent machine learning methods." *Computational and Mathematical Methods in Medicine* 2021 (2021).
- [11] Chittora, Pankaj, et al. "Prediction of chronic kidney disease-a machine learning perspective." *IEEE access* 9 (2021): 17312-17334.
- [12] Walse, Rajesh S., et al. "Effective use of naïve bayes, decision tree, and random forest techniques for analysis of chronic kidney disease." *Information and Communication*

Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 1. Springer Singapore, 2021.

[13] Kumar V. Evaluation of computationally intelligent techniques for breast cancer diagnosis. *Neural Comput Appl.* 2021;33(8):3195–208.

[14] C. Bemando, E. Miranda, M. Aryuni, "Machine-Learning-Based Prediction Models of Coronary Heart Disease Using Naïve Bayes and Random Forest Algorithms," in 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), (IEEE, 2021), pp. 232–237

[15] Nithya, A., et al. "Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images." *Measurement* 149 (2020): 106952.

[16] Navaneeth, Bhaskar, and M. Suchetha. "A dynamic pooling based convolutional neural network approach to detect chronic kidney disease." *Biomedical Signal Processing and Control* 62 (2020): 102068.

[17] B. Navaneeth, M. Suchetha, A dynamic pooling based convolutional neural network approach to detect chronic kidney disease. *Biomed. Signal Proce. Control* 62, 102068 (2020)

[18] B. Deepika, "Early prediction of chronic kidney disease by using machine learning techniques", *Amer. J. Comput. Sci. Eng. Survey*, vol. 8, no. 2, pp. 7, 2020

[19] Yashfi SY. Risk Prediction Of Chronic Kidney Disease Using Machine Learning Algorithms. 2020.

[20] A. Ogunleye, Q.-G. Wang, XGBoost model for chronic kidney disease diagnosis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17, 2131–2140 (2020)

[21] R.P. Ram Kumar, SanjeevaPolepaka, Performance comparison of random forest classifier and convolution neural network in predicting heart diseases, in *Proceedings of the Third International Conference on Computational Intelligence and Informatics*. ed. by K. SrujanRaju, A. Govardhan, B. PadmajaRani, R. Sridevi, M. Ramakrishna Murty (Springer, Singapore, 2020)

- [22] R.S. Walse, G.D. Kurundkar, S.D. Khamitkar, A.A. Muley, P.U. Bhalchandra, S.N. Lokhande, Effective use of naïve bayes, decision tree, and random forest techniques for analysis of chronic kidney disease, in International Conference on Information and Communication Technology for Intelligent Systems. ed. by T. Senjyu, P.N. Mahalle, T. Perumal, A. Joshi (Springer, Singapore, 2020)
- [23] A. Nithya, A. Appathurai, N. Venkatadri, D.R. Ramji, C.A. Palagan, Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images. Measurement (2020). <https://doi.org/10.1016/j.measurement.2019.106952>
- [24] G. R. Vasquez-Morales, S. M. Martinez-Monterrubio, P. Moreno-Ger and J. A. Recio-Garcia, "Explainable prediction of chronic renal disease in the colombian population using neural networks and case-based reasoning", IEEE Access, vol. 7, pp. 152900-152910, 2019.
- [25] H. Singh, N. V. Navaneeth, G. N. Pillai, "Multisurface proximal SVM based decision trees for heart disease classification," in TENCON 2019-2019 IEEE Region 10 Conference (TENCON), (IEEE 2019), pp. 13–18
- [26] S.D. Desai, S. Giraddi, P. Narayankar, N.R. Pudakalakatti, S. Sulega on, Backpropagation neural network versus logistic regression in heart disease classification in advanced computing and communication technologies (Springer, Singapore, 2019)
- [27] A. Brunetti, G.D. Cascarano, I. De Feudis, M. Moschetta, L. Gesualdo, V. Bevilacqua, Detection and segmentation of kidneys from magnetic resonance images in patients with autosomal dominant polycystic kidney disease, in International Conference on Intelligent Computing. ed. by D.-S. Huang, K.-H. Jo, Z.-K. Huang (Springer International Publishing, Cham, 2019)
- [28] A. Nishanth, T. Thiruvaran, Identifying important attributes for early detection of chronic kidney disease. IEEE Rev. Biomed. Eng. 11, 208–216 (2018)

- [29] S. Drall, G. S. Drall, S. Singh and B. B. Naib, "Chronic kidney disease prediction using machine learning: A new approach", *Int. J. Manage. Technol. Eng.*, vol. 8, pp. 278-287, May 2018.
- [31] Tekale S, Shingavi P, Wandhekar S, Chatorikar A. Prediction of chronic kidney disease using machine learning algorithm. *Disease*. 2018;7(10):92–6.
- [32] Fatima M., Pasha M. Survey of machine learning algorithms for disease diagnostic. *J Intel Learn Syst Appl*. 2017;9(01)
- [33] Moccia, S.; De Momi, E.; El Hadji, S.; Mattos, L.S. Blood vessel segmentation algorithms—Review of methods, datasets and evaluation metrics. *Comput. Methods Programs Biomed*. 2018, 158, 71–91.
- [34] Aljaaf, A.J. 2018 Early Prediction of Chronic Kidney Disease Using Machine Learning Supported by Predictive Analytics. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. Wellington. New Zealand
- [35] A. Nishanth, T. Thiruvaran, Identifying important attributes for early detection of chronic kidney disease. *IEEE Rev. Biomed. Eng*. 11, 208–216 (2018)
- [36] D.D. Patil, R.P. Singh, V.M. Thakare, A.K. Gulve, Analysis of ecg arrhythmia for heart disease detection using svm and cuckoo search optimized neural network. *Int. J. Eng. Technol*. 7(217), 27–33 (2018)
- [37] Abdullah Al Imran, Md Nur Amin, and Fatema Tuj Johora. Classification of chronic kidney disease using logistic regression, feedforward neural network and wide & deep learning. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pages 1–6. IEEE, 2018.
- [38] Aqlan, Faisal, Ryan Markle, and Abdulrahman Shamsan. "Data mining for chronic kidney disease prediction." *IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE)*, 2017.
- [39] Fatima M., Pasha M. Survey of machine learning algorithms for disease diagnostic. *J Intel Learn Syst Appl*. 2017;9(01)

- [40] F. Aqlan, R. Markle, A. Shamsan, "Data mining for chronic kidney disease prediction." in IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers, (IISE 2017), pp] 1789–1794
- [41] N. Borisagar, D. Barad, P. Raval, Chronic kidney disease prediction using back propagation neural network algorithm. *Proce. Int. Confe. Commun. Netw.* 19–20, 295–303 (2017)
- [42] U. Rajendra Acharya, Oh. Shu Lih, Y. Hagiwara, J.H. Tan, M. Adam, A. Gertych, R.S. Tan, A deep convolutional neural network model to classify heartbeats. *Comput. Biol. Med.* 89, 389–396 (2017)
- [43] K. R. A. Padmanaban and G. Parthiban, "Applying machine learning techniques for predicting the risk of chronic kidney disease", *Indian J. Sci. Technol.*, vol. 9, no. 29, Aug. 2016.
- [44] Charleonnann A, Fufaung T, Niyomwong T, Chokchueypattanakit W, Suwannawach S, Ninchawee N. Predictive analytics for chronic kidney disease using machine learning techniques. *Manag Innov Technol Int Conf MITiCON*. 2016;80–83:2017.
- [45] Rubini, L.; Soundarapandian, P.; Eswaran, P. Chronic Kidney Disease. *UCI Machine Learning Repository*. 2015. (accessed on 10 June 2023).
- [46] García, S.; Luengo, J.; Herrera, F. Data preprocessing in data mining. *CA Cancer J. Clin.* 2015, 72, 59–139.
- [47] Stanifer JW, et al. The epidemiology of chronic kidney disease in sub-Saharan Africa: A systematic review and meta-analysis. *Lancet Glob Heal.* 2014;2(3):e174–81.
- [48] V. Jha, G. Garcia-Garcia, K. Iseki et al., "Chronic kidney disease: global dimension and perspectives," *The Lancet*, vol. 382, no. 9888, pp. 260–272, 2013.
- [49] W. Mula-Abed, K. A. Rasadi and D. Al-Riyami, "Estimated glomerular filtration rate (eGFR): A serum creatinine-based test for the detection of chronic kidney disease and its impact on clinical practice", *Oman Med. J.*, vol. 27, no. 4, pp. 339–340, 2012.

[50] A. S. Levey, R. Atkins, J. Coresh et al., "Chronic kidney disease as a global public health problem: approaches and initiatives—a position statement from kidney disease improving global outcomes," *Kidney International*, vol. 72, no. 3, pp. 247–259, 2007.

[51] M. J. Lysaght, "Maintenance dialysis population dynamics: current trends and long-term implications," *Journal American Society Nephrology*, vol. 13, suppl 1, pp. S37–S40, 2002.