

CHRONIC KIDNEY DISEASE DETECTION USING MACHINE LEARNING



Thesis/Dissertation submitted in the partial fulfillment of the requirements for the award of the
degree of

BACHELOR OF TECHNOLOGY
in
CSE(Data Science)
by

P. ARVIND REDDY	20K91A6734
M. SAI	20K91A6726
P. LAXMI PRIYA	20K91A6731
K. SAI RAM	21K95A6704

Under the guidance of
Mrs. K. KIRANMAYEE
Asst.Prof(Data Science)

DEPARTMENTMENT OF CSE (DATA SCIENCE)
TKR COLLEGE OF ENGINEERING &TECHNOLOGY
(AUTONOMOUS)
(Accredited by NAAC with 'A+' Grade)
Medbowli, Meerpet, Saroornagar, Hyderabad-500097

DECLARATION BY THE CANDIDATES

We, **Mr. P. Arvind Reddy** bearing Hall Ticket Number: **20K91A6734**, **Mr. M. Sai** bearing Hall Ticket Number: **20K91A6726**, **Ms. P. Laxmi Priya** bearing Hall Ticket Number: **20K91A6731**, **Mr. K. Sai Ram** bearing Hall Ticket Number: **21K95A6704** hereby declare that the major project report titled “**CHRONIC KIDNEY DISEASE DETECTION USING MACHINE LEARNING**” under the guidance of **Mrs. K. KIRANMAYEE Asst.prof** in Department of Computer Science and Engineering (Data Science) is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering (Data Science)**.

P.ARVIND REDDY 20K91A6734

M. SAI 20K91A6726

P. LAXMI PRIYA 20K91A6731

K . SAI RAM 21K95A6704

CERTIFICATE

This is to certify that the major project report entitled **CHRONIC KIDNEY DISEASE DETECTION USING MACHINE LEARNING**, being submitted by **Mr. P.Arvind Reddy**, bearing **Roll.No.:20K91A6734**, **Mr. M.Sai**, bearing **Roll.No.:20K91A676**, **Ms. P.Laxmi Priya**, bearing **Roll.No.:20K91A6731**, **Mr. K.Sai Ram**, bearing **Roll.No.:21K95A6704** in partial fulfillment of the academic requirements for the award of the degree of **Bachelor of Technology in CSE(Data Science)**, to the TKR College of Engineering & Technology is a record of bonafide work carried out by them under my guidance and supervision.

Signature of the Guide
Mrs. K. KIRANMAYEE
Asst.Prof(Data Science)

Signature of the HOD
Dr.V.Krishna
Professor

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowned our efforts with success.

We express our sincere gratitude to **Management of TKRCET** for granting permission and giving inspiration for the completion of the project work.

Our faithful thanks to our **Principal Dr. D. V. Ravi Shankar, M.Tech., Ph.D.**, TKR College of Engineering & Technology for his Motivation in studies and completion of the project work.

With our heart full pleasure we thank our **Head of the Department Dr.V.Krishna, M.Tech., Ph.D.**, Professor, Department of CSE (Data Science), TKR College of Engineering & Technology for his suggestions regarding the project.

Thanks to our Project Coordinator **Mr.M.Arokia Muthu M.E.,(Ph.D.)**, Assistant Professor, Department of CSE (Data Science), TKR College of Engineering & Technology for his constant encouragement.

We are indebted to the **Internal Guide, Mrs. K. KIRANMAYEE Asst.prof**, Department of CSE (Data Science), TKR College of Engineering & Technology for his support in completion of the project successfully.

Finally, we express our thanks to one and all that have helped us in successfully completing this project. Furthermore, we would like to thank our family and friends for their moral support and encouragement.

	By,
P.ARVIND REDDY	20K91A6734
M. SAI	20K91A6726
P. LAXMI PRIYA	20K91A6731
K . SAI RAM	21K95A6704

TABLE OF CONTENTS

ABSTRACT	i
LIST OF FIGURES	ii
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Problem definition	2
1.3 Limitations of existing system	2
1.4 Proposed system	3
2 LITERATURE REVIEW	
2.1 Paper-1	4
2.2 Paper-2	5
2.3 Paper-3	6
2.4 Paper-4	7
2.5 Paper-5	8
2.6 Paper-6	9
2.7 Literature Survey Conclusion	10
3 REQUIREMENTS ANALYSIS	11
3.1 Functional Requirements	11
3.2 Non-Functional Requirements	12
4 DESIGN	
4.1 System architecture	14
4.2 Use Case Diagram	15
4.3 Class Duagram	15
4.4 Sequence Diagram	16
5 CODING	
5.1 Code for Kidney Disease Detection	17
5.2 Code for Algorithms	24

6 IMPLEMENTATION & RESULTS	
6.1 Explanation of Key functions	36
6.2 Method of Implementation	37
6.2.1 Forms	37
6.2.1.1 Home Page	37
6.2.1.2 User Sign In	37
6.2.1.3 Disease Detection	38
6.2.1.4 Admin Sign In	38
6.2.1.5 Admin Accessing User Details	39
6.2.1.6 About Project	39
6.2.1.7 Contact Us	40
6.2.2 Output Screens	40
6.2.2.1 Model Prediction Page	40
6.2.2.2 Model Performance	41
6.2.3 Result Analysis	41
7 TESTING & VALIDATION	43
7.1 Design of Test Cases and scenarios	43
7.2 Validation	45
7.3 Conclusion	46
8 CONCLUSION	48
REFERENCES	49

ABSTRACT

Chronic Kidney Disease (CKD) is a prevalent and often asymptomatic condition that poses a significant public health challenge worldwide. Early detection and intervention are crucial to mitigate the progression of CKD and prevent associated complications. This study presents an automated approach for the early detection of CKD using machine learning techniques. The developed model demonstrates promising accuracy, sensitivity, and specificity in distinguishing individuals with CKD from those without. Furthermore, we explore the potential of explainable AI techniques to provide insights into the decision-making process of the model, aiding clinicians in understanding the key factors influencing CKD detection.

The developed model demonstrates promising accuracy, sensitivity, and specificity in distinguishing individuals with CKD from those without. Furthermore, we explore the potential of explainable AI techniques to provide insights into the decision-making process of the model, aiding clinicians in understanding the key factors influencing CKD detection. By facilitating timely interventions and personalized patient care, this approach contributes to the overall improvement of healthcare outcomes and the reduction of the societal burden associated with CKD. Future work may involve the refinement of the model with additional data sources and the development of a user-friendly interface for seamless integration into clinical workflows.

Project initiates and provide a valuable tool for early detection and risk assessment, ultimately leading to better patient outcomes and reduced health care costs . We explore the application of various machine learning algorithms to predict the levels of chronic diseases and recommend appropriate drugs based on patient profiles. Early prediction based on patient conditions and physical symptoms which results in accurate reports. The goal is to improve accuracy , optimise treatment plans , and enhance patient outcomes.

LIST OF FIGURES

4.1 System Architecture	14
4.3 Use case Diagram	15
4.4 Class Diagram	15
4.5 Sequence Diagram	16

INTRODUCTION

1.1 Motivation

The motivation behind this project documentation lies in the urgent need for innovative solutions to address the challenges posed by CKD. Machine learning, with its ability to analyze vast amounts of diverse data, offers a promising avenue for early detection and risk assessment. By leveraging advanced algorithms, we aim to develop a reliable and efficient system capable of identifying CKD at its early stages when intervention is most impactful.

The significance of this project extends beyond the realm of academia and research. Successful implementation of a machine learning-based CKD detection system has the potential to revolutionize clinical practices, providing healthcare professionals with a valuable tool for timely decision-making. Improved diagnostic accuracy will lead to personalized treatment plans, better patient outcomes, and a reduction in the overall healthcare burden associated with CKD.

Furthermore, the documentation of this project serves as a knowledge-sharing resource, contributing to the growing body of literature on the intersection of healthcare and machine learning. By transparently documenting the methodologies, challenges, and outcomes, we aim to facilitate collaboration, inspire further research, and foster a community dedicated to advancing the field of medical diagnostics.

In essence, this project documentation is driven by the commitment to making a meaningful impact on public health, advancing medical science, and providing a practical tool that can be readily adopted by healthcare professionals to enhance the quality of care for individuals at risk of or affected by Chronic Kidney Disease. The silent progression of CKD often leads to late-stage diagnoses, resulting in limited treatment options and increased morbidity and mortality rates.

1.2 Problem definition

The problem definition underscores the critical necessity for an innovative approach to CKD detection. Leveraging machine learning techniques presents an opportunity to analyze complex datasets comprehensively, identify subtle patterns indicative of early-stage CKD, and develop a predictive model that can assist healthcare professionals in making timely and informed decisions. Addressing these challenges will not only improve patient outcomes but also contribute to the overall efficiency and sustainability of healthcare systems grappling with the increasing burden of chronic diseases.

1.3 Limitations of existing system

The existing system for Chronic Kidney Disease (CKD) detection is marred by several limitations that collectively impede its effectiveness. Firstly, the predominant issue lies in the late-stage detection of CKD, where symptoms often become apparent only when the disease has progressed significantly. This delays interventions, diminishing the potential impact of timely preventive measures. Moreover, the reliance on traditional biomarkers, such as serum creatinine and estimated glomerular filtration rate (eGFR), constrains the accuracy of diagnoses, particularly in capturing early signs of kidney dysfunction.

The current diagnostic tools also exhibit a limited capacity for personalization, often failing to account for the diverse range of risk factors and individual variations contributing to CKD. The absence of a personalized risk assessment hampers the development of targeted and effective preventive strategies tailored to individual patient profiles. Furthermore, the high cost and resource intensity associated with laboratory tests and clinical evaluations present a significant barrier, particularly in resource-constrained healthcare settings, limiting the widespread adoption and accessibility of the existing diagnostic approach.

The inability to leverage advanced data analytics is another notable limitation. Traditional diagnostic methods may struggle to efficiently analyze and interpret large and diverse datasets, hindering a comprehensive understanding of CKD risk factors. In this context, the underutilization of recent technological advances, particularly in machine learning and artificial intelligence, is a missed opportunity. Integrating these innovations into the diagnostic paradigm could offer a more sophisticated and automated approach, enhancing both accuracy and efficiency.

Additionally, the existing system may lack robust predictive capabilities, hindering its ability to forecast the likelihood of CKD development in individuals with early risk factors. Predictive capabilities are crucial for implementing proactive measures and interventions to slow or prevent disease progression. Recognizing and addressing these limitations is imperative for the development of a more effective and proactive system for CKD detection. By overcoming these challenges and embracing emerging technologies, we can significantly improve the accuracy, efficiency, and overall impact of CKD diagnosis and management.

1.4 Proposed system

The proposed system aims to overcome the limitations of existing chronic kidney disease detection system by introducing a multi-level classification approach that leverages cutting-edge technologies. The core innovation lies in the integration of machine learning algorithms to create a responsive and adaptive platform.

Chronic Kidney Disease (CKD) detection introduces a paradigm shift by integrating advanced machine learning techniques to address the limitations inherent in the current diagnostic framework. Central to this approach is the utilization of a comprehensive dataset, encompassing a diverse range of clinical, demographic, and lifestyle factors. Through sophisticated feature engineering and selection methods, the system aims to extract and prioritize relevant information critical to CKD detection. State-of-the-art machine learning algorithms, including Support Vector Machines, Random Forests, and Neural Networks, will be implemented for classification, while ensemble methods and deep learning architectures will enhance predictive capabilities, ensuring accurate and early identification of individuals at risk of CKD.

We formulate a multi-class classification task to predict disease levels [eg : high , medium, low]. We also introduce drug recommendation system for the people who affected with the kidney related diseases. We also predict price for those drugs we recommend in a suitable way.

Chapter 2

LITERATURE SURVEY

2.1 PAPER-1

Title: Machine Learning-Based Early Diagnosis of Chronic Kidney Disease Using Clinical Test Attributes

Authors: Md. Rashed-Al-Mahfuz, Abedul Haque, Akm Azad, Salem A. Alyami, Julian M. W. Quinn, and Mohammad Ali Moni

Description: Chronic kidney disease (CKD) poses a significant public health challenge globally, with late-stage diagnosis and limited testing facilities contributing to high morbidity and mortality rates, especially in less developed regions. This study focuses on developing machine learning models utilizing select key pathological categories to identify clinical test attributes crucial for accurate early CKD diagnosis. The results indicate that the optimized datasets effectively facilitate CKD diagnosis, with the random forest (RF) classifier demonstrating superior performance. Overall, the machine learning approach offers promising predictive analytics for CKD screening, potentially serving as a valuable resource for enhancing and streamlining CKD diagnosis and treatment planning processes.

Merits:

1. Early Diagnosis: The study focuses on early diagnosis of CKD, which is crucial for improving patient outcomes and reducing healthcare costs associated with late-stage diagnosis.
2. Machine Learning Models: By leveraging machine learning techniques, the study develops predictive models that can accurately identify key clinical test attributes for CKD diagnosis, aiding in timely intervention and treatment.

Demerits:

1. Data Dependency: The effectiveness of the machine learning models relies on the availability and quality of clinical data, which may vary across different healthcare settings and regions.
2. Model Interpretability: While the machine learning models demonstrate high accuracy, their interpretability may be limited, making it challenging to understand the underlying factors driving CKD diagnosis.
3. Generalizability: The study's findings may be limited in their generalizability to diverse populations and healthcare settings, necessitating further validation and testing across different contexts.

2.2 PAPER-2

Title: Adaptive Hybridized Deep Convolutional Neural Network for Chronic Kidney Disease Prediction on the Internet of Medical Things Platform

Authors: Guozhen Chen, Chenguang Ding, Yang Li, Xiaojun Hu, Xiao Li, Li Ren, Xiaoming Ding, Puxun Tian, and Wujun Xue

Description: Chronic kidney disease (CKD) is a serious lifelong condition stemming from renal disease or impaired kidney function, with early diagnosis and proper therapy being crucial for patient survival. In this study, the authors address the urgent challenge of identifying subtypes of kidney disease accurately through the development of automated tools. They propose an Adaptive Hybridized Deep Convolutional Neural Network (AHDCNN) for the early detection of CKD, leveraging various deep learning methods to enhance efficiency and effectiveness. The AHDCNN model aims to classify kidney disease subtypes efficiently by reducing feature dimensionality using convolutional neural network (CNN) algorithms. Experimental evaluations conducted on the Internet of Medical Things platform (IoMT) demonstrate the potential of machine learning advances in providing intelligent solutions for predicting kidney disease beyond traditional methods.

Merits:

1. Early Detection: The proposed AHDCNN model enables early detection of CKD, facilitating timely intervention and treatment to prevent disease progression.
2. Integration of Deep Learning: By leveraging deep learning techniques, the study enhances classification efficiency and accuracy, offering a promising framework for intelligent solutions in disease prediction.
3. Feature Reduction: The algorithm model developed using CNN helps reduce feature dimensionality, improving the accuracy of the classification system and aiding in efficient diagnosis.

Demerits:

1. Data Dependence: The effectiveness of the AHDCNN model relies heavily on the availability and quality of medical data, which may vary across different healthcare settings and regions.
2. Computational Complexity: Implementing deep learning models like AHDCNN may require significant computational resources and expertise, potentially limiting accessibility and scalability in resource-constrained environments.
3. Interpretability: While the AHDCNN model demonstrates high accuracy, its interpretability may be limited, making it challenging to understand the underlying factors driving CKD prediction outcomes.

2.3 PAPER-3

Title: Automated Sensing of Chronic Kidney Disease Using Deep Learning

Authors: Navaneeth Bhaskar and Suchetha M.

Description: This article presents a novel sensing technique for the automated detection of chronic kidney disease (CKD) by monitoring salivary urea concentration. A new sensing approach is introduced to monitor urea levels in saliva samples, providing a non-invasive and convenient method for disease detection. To analyze the raw signals obtained from the sensor, a one-dimensional deep learning convolutional neural network (CNN) algorithm is implemented, integrated with a support vector machine (SVM) classifier. The use of this CNN–SVM integrated network enhances the classification accuracy of the model. Experimental results demonstrate that the proposed model achieves a high classification accuracy of 98.04% in accurately classifying CKD samples, showcasing its potential for automated CKD sensing.

Merits:

1. **Non-Invasive Sensing:** The proposed sensing technique monitors salivary urea concentration, offering a non-invasive and convenient method for detecting chronic kidney disease.
2. **Deep Learning Integration:** The integration of a one-dimensional CNN algorithm with an SVM classifier enhances the classification accuracy of the model, improving its effectiveness in disease detection.
3. **High Classification Accuracy:** Experimental results demonstrate a high classification accuracy of 98.04%, indicating the effectiveness of the proposed model in accurately classifying CKD samples.
4. **Potential for Automation:** The automated sensing system has the potential to streamline CKD detection processes, leading to early diagnosis and timely intervention for improved patient outcomes.

Demerits:

1. **Limited Validation:** While the proposed model demonstrates high classification accuracy, further validation with larger and diverse datasets is necessary to assess its robustness and generalizability across different populations and settings.
2. **Sensor Reliability:** The reliability and accuracy of the sensing technique may be influenced by factors such as sensor calibration, variability in saliva composition, and environmental conditions, which need to be carefully addressed for real-world application.
3. **Interpretability:** Deep learning models, such as CNN, may lack interpretability, making it challenging to understand the underlying features driving disease classification outcomes. Additional efforts may be required to enhance model interpretability and clinical relevance.

2.4 PAPER-4

Title: Chronic Kidney Disease Prediction Using Machine Learning Techniques

Authors: Pankaj Chittora, Sandeep Chaurasia, Prasun Chakrabarti, Gaurav Kumawat, Tulika Chakrabarti, Zbigniew Leonowicz, Michał Jasiński, Łukasz Jasiński, Radomir Gono, Elżbieta Jasińska, and Vadim Bolshev

Description: Chronic Kidney Disease (CKD) is a prevalent and critical illness requiring timely diagnosis for effective treatment. This article explores CKD prediction using machine learning techniques, leveraging a dataset obtained from the UCI repository. Seven classifier algorithms, including artificial neural network, C5.0, Chi-square Automatic Interaction Detector, logistic regression, linear support vector machine (LSVM) with penalty L1 and L2, and random tree, are applied to the dataset. Feature selection techniques such as correlation-based feature selection, wrapper method feature selection, and least absolute shrinkage and selection operator regression are also employed. Results demonstrate LSVM with penalty L2 achieving the highest accuracy of 98.86% using synthetic minority oversampling technique with full features. Precision, recall, F-measure, area under the curve, and GINI coefficient are computed and compared across various algorithms. Additionally, a deep neural network is applied, achieving the highest accuracy of 99.6%.

Merits:

1. High Accuracy: The machine learning models, particularly LSVM with penalty L2 and deep neural network, demonstrate high accuracy in predicting CKD, indicating their potential for effective disease detection.
2. Comprehensive Analysis: The study evaluates multiple classifier algorithms and feature selection techniques, providing insights into the most effective approaches for CKD prediction.
3. Robust Performance Metrics: Precision, recall, F-measure, and other performance metrics are computed and compared, offering a comprehensive evaluation of model performance beyond accuracy alone.
4. Utilization of Deep Learning: The application of a deep neural network demonstrates the potential of deep learning techniques in enhancing CKD prediction accuracy.

Demerits:

1. Limited Generalizability: The study's findings may be limited to the specific dataset used from the UCI repository, and further validation on diverse datasets is necessary to assess the models' generalizability.
2. Complexity of Deep Learning: While deep neural networks achieve high accuracy, they often require significant computational resources and expertise for training and deployment.

2.5 PAPER-5

Title: Predicting Chronic Kidney Disease Using Machine Learning Classifiers

Authors: Chilakamarthi Prem Kashyap, Gollapudi Sai Dayakar Reddy, M Balamurugan

Description:Chronic kidney disease (CKD) is a prevalent condition characterized by impaired kidney function, leading to complications in blood purification. Without proper diagnosis and management, CKD can progress to irreversible stages, necessitating dialysis or kidney transplantation. To address the challenges associated with late-stage diagnosis and treatment costs, this paper explores the use of machine learning techniques for early CKD detection. Four machine learning algorithms—Support Vector Machine Classifier (SVM), K-Nearest Neighbor Algorithm (KNN), Random Forest Algorithm, and Decision Tree Algorithm—are applied to a dataset sourced from the UCI repository. Preprocessing techniques are employed to enhance the accuracy of CKD prediction by these algorithms.

Merits:

1. Early Diagnosis: Machine learning classifiers offer the potential for early detection of CKD, enabling timely intervention and improved patient outcomes.
2. Multiple Algorithm Evaluation: The study evaluates the performance of four different machine learning algorithms, providing insights into their effectiveness for CKD prediction.
3. Utilization of Public Dataset: By utilizing a dataset from the UCI repository, the study ensures transparency and reproducibility of results, facilitating further research and validation.
4. Data Preprocessing Techniques: The application of data preprocessing techniques enhances the quality of the dataset, leading to improved accuracy in CKD prediction.

Demerits:

1. Algorithm Selection Bias: The choice of machine learning algorithms may influence the study's findings, and alternative algorithms could yield different results.
2. Limited Dataset Scope: While the use of a public dataset ensures accessibility, it may lack diversity and real-world complexity, potentially limiting the generalizability of the findings.
3. Model Interpretability: Some machine learning algorithms, particularly ensemble methods like Random Forest, may lack interpretability, making it challenging to understand the factors driving CKD prediction outcomes.
4. Dependency on Data Quality: The accuracy of CKD prediction models heavily relies on the quality and completeness of the dataset, and errors or biases in the data could affect the reliability of the results.

2.6 PAPER-6

Title: Prediction of Chronic Kidney Disease - A Machine Learning Perspective

Authors: Pankaj Chittora, Sandeep Chaurasia (Senior Member, IEEE)

Description: Chronic Kidney Disease (CKD) presents a significant health challenge, emphasizing the need for timely and accurate diagnosis. Machine learning techniques have emerged as reliable tools for medical diagnosis, enabling early detection and intervention. This paper discusses CKD prediction from a machine learning perspective, utilizing a dataset sourced from the UCI repository. Seven classifier algorithms, including artificial neural network, C5.0, Chi-square Automatic Interaction Detector, logistic regression, linear support vector machine (SVM) with penalty L1 & L2, and random tree, are applied to the dataset. Feature selection techniques are employed to enhance classifier performance. Results show that the linear SVM with penalty L2 achieves the highest accuracy of 98.86% using synthetic minority oversampling technique with full features. Additionally, precision, recall, F-measure, area under the curve, and GINI coefficient metrics are computed and compared across various algorithms. Furthermore, a deep neural network is applied to the dataset, achieving the highest accuracy of 99.6%.

Merits:

1. Comprehensive Analysis: The study evaluates multiple machine learning algorithms and feature selection techniques, providing a comprehensive understanding of CKD prediction.
2. High Accuracy: The utilization of machine learning models yields high accuracy in CKD prediction, essential for early diagnosis and intervention.
3. Metric Comparison: Various evaluation metrics, including precision, recall, and area under the curve, are computed and compared, offering insights into classifier performance.
4. Deep Learning Integration: The application of a deep neural network demonstrates the potential of advanced techniques in achieving superior accuracy in CKD prediction.

Demerits:

1. Model Interpretability: Some machine learning algorithms, particularly neural networks, lack interpretability, hindering the understanding of underlying factors influencing CKD prediction.
2. Dependency on Dataset: The performance of machine learning models heavily relies on the quality and representativeness of the dataset, which may limit generalizability to diverse populations.
3. Computational Complexity: Deep learning models, while achieving high accuracy, often require significant computational resources for training and inference, posing challenges for deployment in resource-constrained environments.

Conclusion for Literature Survey:

In conclusion, the exploration of machine learning techniques for predicting chronic kidney disease (CKD) showcases promising advancements in early detection and intervention. These studies underscore the significance of timely diagnosis in mitigating the burden of CKD on public health systems and improving patient outcomes. By leveraging diverse machine learning algorithms and integrating advanced techniques such as deep learning, researchers have achieved notable successes in accurately identifying CKD and its subtypes. The high accuracy rates attained by these models, coupled with comprehensive evaluation metrics, highlight their potential for clinical application and decision support. However, challenges remain, including the interpretation of complex models, dependence on dataset quality and representativeness, and computational complexity. Addressing these limitations will be crucial for advancing the reliability and accessibility of machine learning-based CKD prediction tools. Overall, these studies offer valuable insights into the role of machine learning in transforming CKD diagnosis and management, paving the way for improved healthcare delivery and patient outcomes in the future.

In further exploration, the collective findings from these studies underscore the growing importance of machine learning in revolutionizing the diagnosis and management of chronic kidney disease (CKD). By harnessing the power of diverse algorithms and innovative techniques, researchers have made significant strides towards early detection and intervention, essential for improving patient prognosis and reducing healthcare burdens. The robust performance metrics and high accuracy rates achieved by machine learning models validate their potential as valuable tools in clinical decision-making.

Chapter 3

REQUIREMENTS ANALYSIS

The requirement analysis for the development of a Chronic Kidney Disease (CKD) detection system. Stakeholder identification and engagement, including healthcare professionals, data scientists, and end-users, are critical to ensuring the system aligns with the diverse needs of the healthcare ecosystem. From a technical perspective, the analysis must encompass the selection and integration of diverse datasets that capture a wide range of patient information to train and validate the machine learning models effectively.

Feature engineering and selection techniques must be employed to extract relevant information, and the choice of machine learning algorithms, such as Support Vector Machines, Random Forests, and Neural Networks, should be guided by the complexity of CKD risk factors. The system should also prioritize interpretability through explainable AI techniques, fostering trust among healthcare professionals. Real-time monitoring capabilities and dynamic model updates are essential for ensuring the adaptability of the system to evolving patient profiles.

Additionally, the development of a user-friendly interface is paramount for seamless integration into clinical workflows, promoting accessibility and ease of use. Regular validation processes and iterative improvements based on real-world performance and user feedback should be incorporated to refine the model continuously. Ethical considerations, including bias mitigation strategies, should be integrated to ensure fairness and equity across diverse patient populations. Overall, a thorough requirement analysis lays the foundation for a robust, effective, and ethically sound CKD detection system that aligns with the needs of both healthcare professionals and the patients it aims to benefit.

3.1 Functional Requirements

The functional requirements for the development of a Chronic Kidney Disease (CKD) detection system using machine learning are multifaceted, covering critical aspects of data processing, model development, user interaction, and system adaptability. The system must begin by integrating diverse datasets, incorporating clinical, demographic, and lifestyle factors to ensure a comprehensive analysis of CKD risk factors. Advanced feature engineering and selection techniques are imperative to extract pertinent information for accurate model training.

Employing a variety of machine learning algorithms, including Support Vector Machines, Random Forests, and Neural Networks, is necessary to develop a robust CKD detection model capable of addressing the complexity of risk factors. Additionally, the integration of explainable AI techniques enhances the interpretability of the model, providing valuable insights into the decision-making process and fostering trust among healthcare professionals. Real-time monitoring of patient data and dynamic updates to the model enable adaptability to evolving patient profiles and emerging risk factors, ensuring the system's ongoing relevance. A user-friendly interface is essential, facilitating seamless integration into clinical workflows and providing healthcare professionals with intuitive access to CKD detection insights.

Rigorous validation protocols, leveraging independent datasets, and iterative improvement processes based on real-world performance and user feedback are crucial for refining the model continuously. Ethical considerations, including the implementation of bias mitigation strategies, are integral to ensuring fair and equitable outcomes across diverse patient populations. These functional requirements collectively establish the foundation for a comprehensive and effective CKD detection system, aligning with the goals of healthcare professionals and contributing to enhanced patient outcomes.

Functional requirements collectively define the capabilities and characteristics necessary for a CKD detection system that meets the needs of healthcare professionals and contributes to improved patient outcomes.

3.2 Non-Functional Requirements

The non-functional requirements for a Chronic Kidney Disease (CKD) detection system using machine learning encompass crucial aspects that govern the system's performance, usability, and overall quality. Firstly, performance requirements mandate that the system should demonstrate responsiveness, handling large and diverse datasets efficiently to ensure timely processing and analysis. The accuracy and reliability of the CKD detection model are paramount, requiring a high level of precision to minimize false positives and false negatives, thus ensuring the trustworthiness of the system in clinical applications.

Scalability is a key consideration, with the system designed to handle an increasing volume of patient data without compromising performance. In terms of usability, the system should adhere to accessibility standards, providing an intuitive and inclusive user interface that accommodates various healthcare professionals, regardless of their technical expertise. Security and privacy requirements demand robust measures to safeguard patient data, adhering to relevant healthcare regulations and standards.

The system must also be designed for interoperability, seamlessly integrating with existing healthcare information systems and Electronic Health Records (EHR) to facilitate streamlined workflows within clinical settings. Reliability and availability considerations dictate that the CKD detection system should operate with minimal downtime, ensuring continuous accessibility for healthcare professionals who rely on its insights. Lastly, the system should be adaptable to varying healthcare environments, accommodating different infrastructures and resource capacities.

These non-functional requirements collectively establish a framework for a CKD detection system that not only delivers accurate and reliable results but also prioritizes usability, security, and adaptability within the dynamic landscape of healthcare.

Chapter 4

DESIGN

4.1 SYSTEM ARCHITECTURE

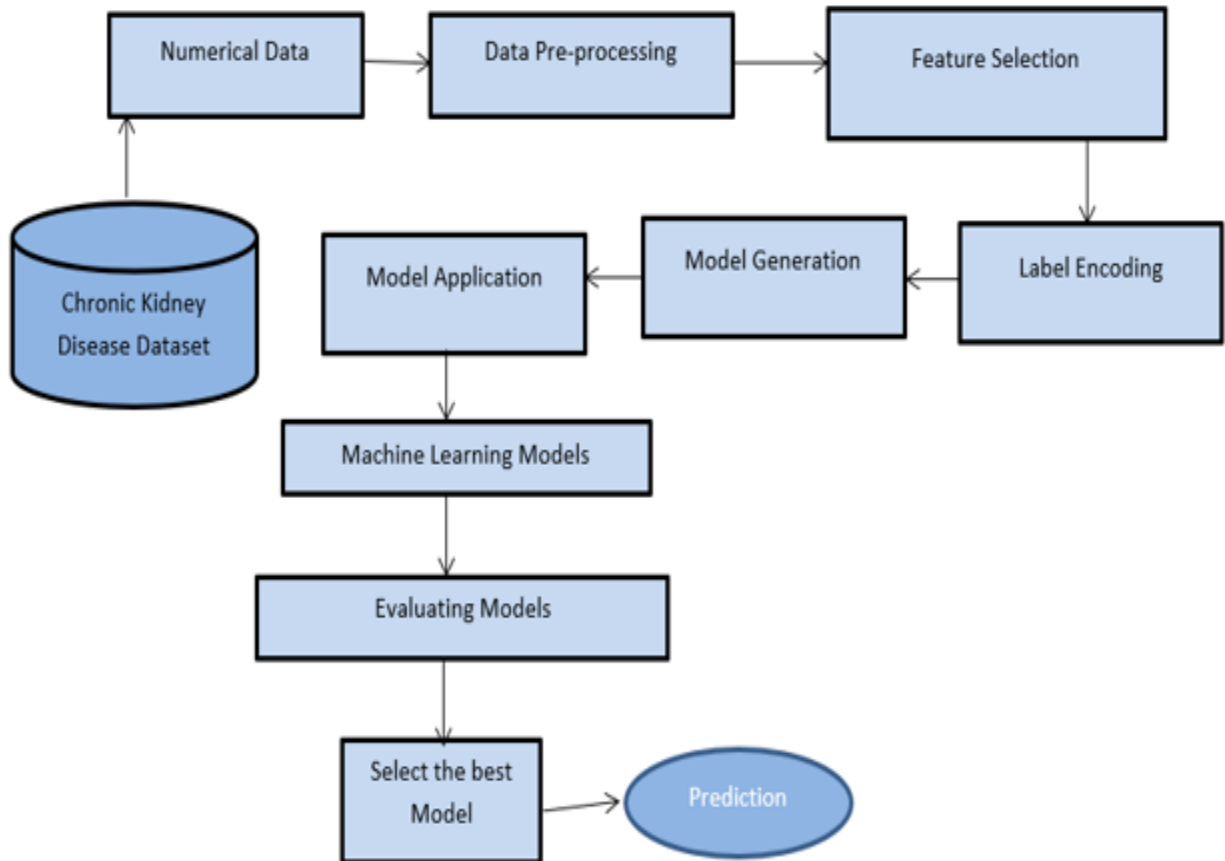


Fig. 4.1 SYSTEM ARCHITECTURE

4.2 USE CASE DIAGRAM

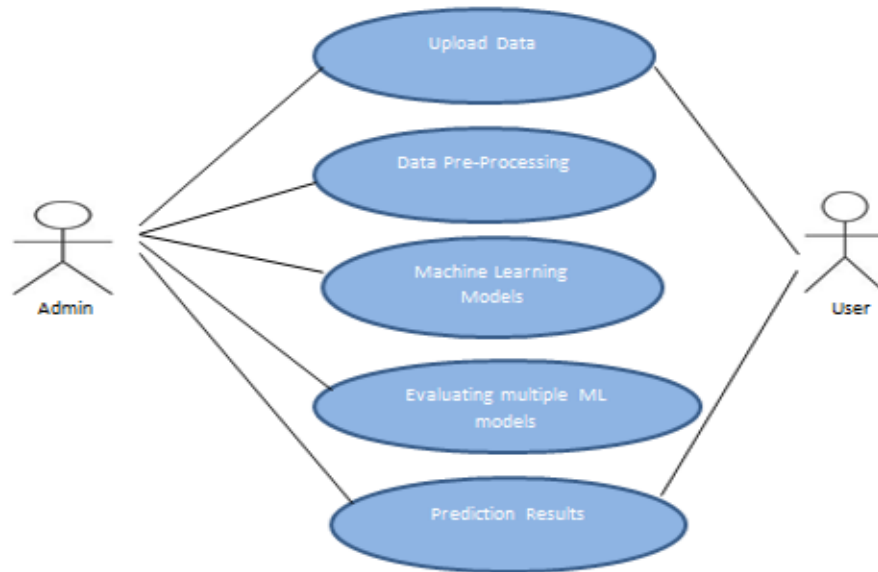


Fig. 4.2 USE CASE DIAGRAM

4.3 CLASS DIAGRAM

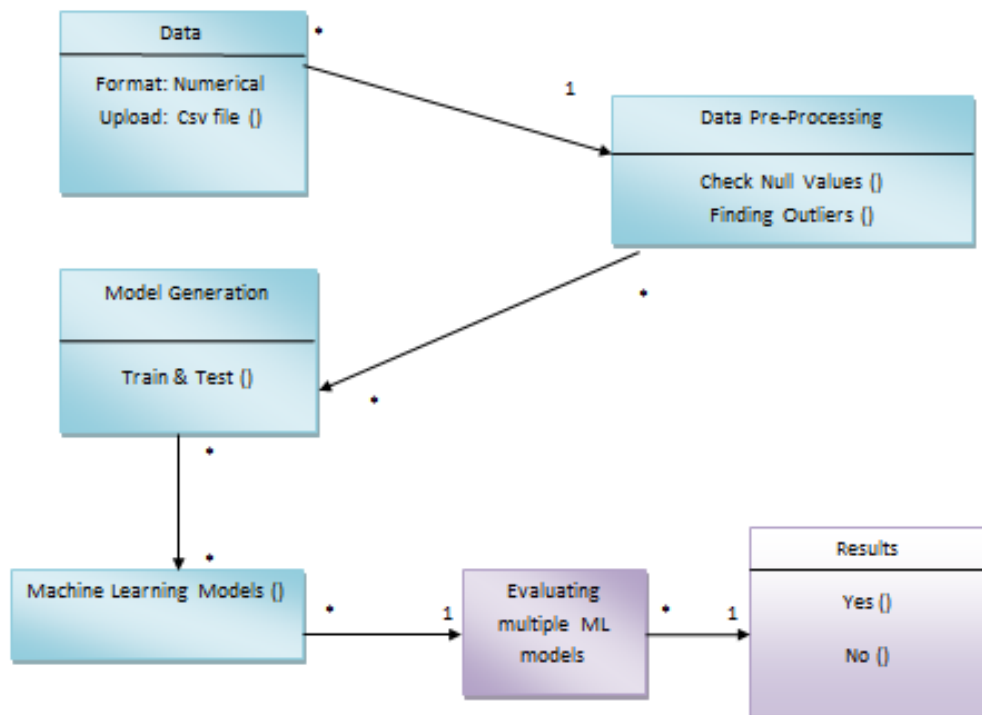


Fig.4.3 CLASS DIAGRAM

4.4 SEQUENCE DIAGRAM

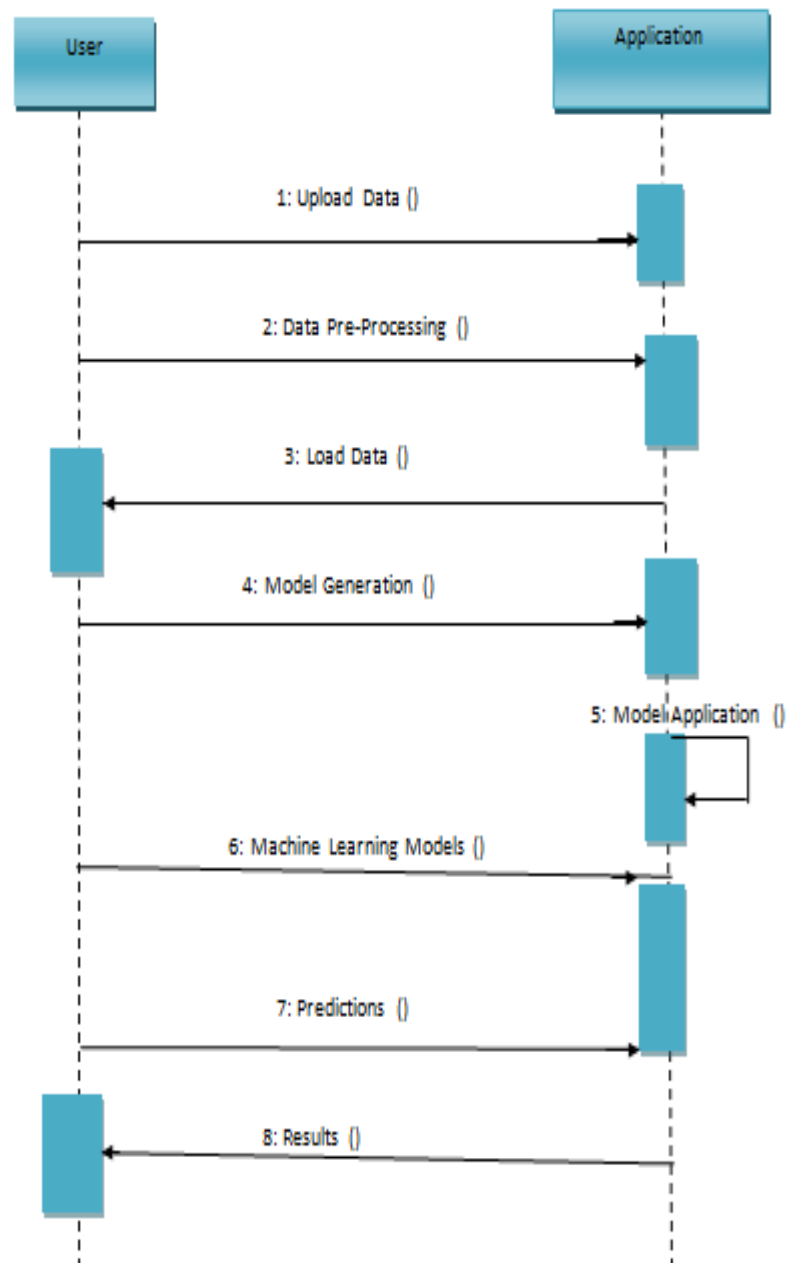


Fig. 4.4 SEQUENCE DIAGRAM

Chapter 5

CODING

5.1. Code for kidney disease detection

PyCharm Community edition supports Jupyter notebooks in read-only mode, to get full support for local notebooks download and try PyCharm Professional now!

Try DataSpell — a dedicated IDE for data science, with full support for local and remnotebooks

Try Datalore — an online environment for Jupyter notebooks in the browser

Also read more about JetBrains Data Solutions on our website

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import warnings
```

```
import pickle
```

```
# Ignore all warnings globally
```

```
warnings.filterwarnings("ignore")
```

```
df=pd.read_csv("../data/kidney_disease.csv")
```

```
df.head()
```

```
df.columns
```

```
Index(['id', 'age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr',  
      'bu', 'sc', 'sod', 'pot', 'hemo', 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad',  
      'appet', 'pe', 'ane'],  
      dtype='object')
```

```
df.columns
```

```
df.columns = ['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','haemoglobin','packed_cell_volume','white_blood_cell_count','red_blood_ce
ll_count','hypertension','diabetes_mellitus','coronary_artery_disease','appetite','peda_edema','aane
mia']
```

```
# checking for null values
```

```
df.isna().sum().sort_values(ascending = False)
```

```
red_blood_cells      152
red_blood_cell_count 131
white_blood_cell_count 106
potassium             88
sodium               87
packed_cell_volume    71
pus_cell              65
haemoglobin           52
sugar                49
specific_gravity      47
albumin               46
blood_glucose_random  44
blood_urea            19
serum_creatinine      17
blood_pressure        12
age                   9
bacteria              4
pus_cell_clumps       4
hypertension          2
diabetes_mellitus      2
coronary_artery_disease 2
appetite              1
peda_edema            1
aanemia              1
dtype: int64
```

```
df[num_cols].isnull().sum().sort_values(ascending = False)
```

```
red_blood_cell_count    131
```

```

white_blood_cell_count  106
potassium                88
sodium                  87
packed_cell_volume      71
haemoglobin              52
sugar                   49
specific_gravity         47
albumin                 46
blood_glucose_random    44
blood_urea               19
serum_creatinine        17
blood_pressure           12
age                      9
dtype: int64

```

```
# Define the MDRD equation for eGFR calculation
```

```
def calculate_egfr(row):
```

```
    creatinine = row['serum_creatinine']
```

```
    age = row['age']
```

```
    systolic_bp = row['blood_pressure']
```

```
    if pd.notna(creatinine) and pd.notna(age) and pd.notna(systolic_bp):
```

```
        return 175 * ((creatinine ** -1.154) * (age ** -0.203) / systolic_bp) * 0.742*100
```

```
    else:
```

```
        return None
```

```
# Apply the eGFR calculation function to create a new column 'eGFR'
```

```
df['eGFR'] = df.apply(calculate_egfr, axis=1)
```

- Stage 0: $\text{eGFR} \geq 90 \text{ mL/min/1.73 m}^2$ Normal range
- Stage 1: $60 \leq \text{eGFR} < 90 \text{ mL/min/1.73 m}^2$
- Stage 2: $30 \leq \text{eGFR} < 60 \text{ mL/min/1.73 m}^2$
- Stage 3: $15 \leq \text{eGFR} < 30 \text{ mL/min/1.73 m}^2$
- Stage 4: $\text{eGFR} < 15 \text{ mL/min/1.73 m}^2$ or end-stage renal disease (ESRD)

```

def label_ckd_stage(row):
    eGFR = row['eGFR']

    if pd.notna(eGFR):

        if eGFR >= 90:
            return 0
        elif 60 <= eGFR < 90:
            return 1
        elif 30 <= eGFR < 60:
            return 2
        elif 15 <= eGFR < 30:
            return 3
        elif eGFR < 15:
            return 4
        else:
            return 'Unknown'
    else:
        return 'Unknown'

# Apply the CKD stage labeling function to create a new column 'CKD_stage'
df['CKD_stage'] = df.apply(label_ckd_stage, axis=1)

df.iloc[5]

```

age	60.000000
blood_pressure	90.000000
specific_gravity	1.015000
albumin	3.000000
sugar	0.000000
red_blood_cells	1.000000
pus_cell	1.000000
pus_cell_clumps	0.000000
bacteria	0.000000
blood_glucose_rand	74.000000
blood_urea	25.000000
serum_creatinine	1.100000
sodium	142.000000
potassium	3.200000
haemoglobin	12.200000
packed_cell_volume	39.000000
white_blood_cell_count	7800.000000
red_blood_cell_count	4.400000
hypertension	1.000000
diabetes_mellitus	1.000000

coronary_artery_disease	0.000000
appetite	0.000000
peda_edema	1.000000
aanemia	0.000000
eGFR	56.294709
CKD_stage	2.000000

Name: 5, dtype: float64

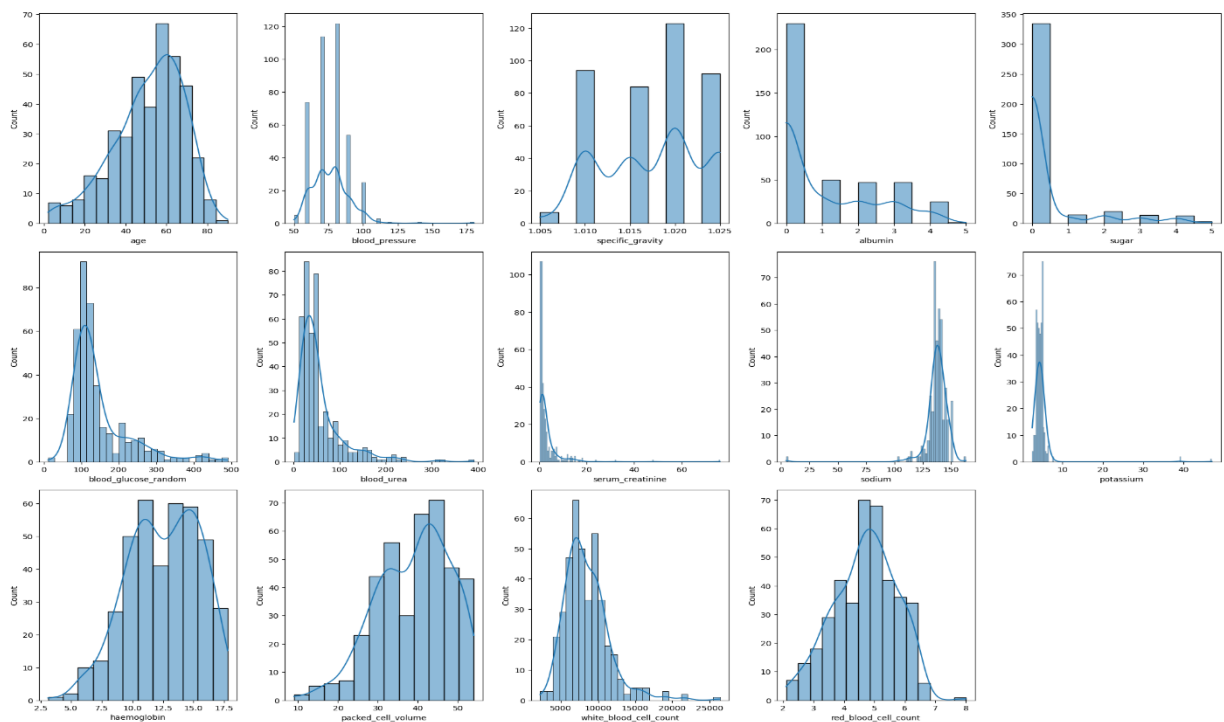
Visualizing the Data

```
plt.figure(figsize = (20, 15))
plotnumber = 1
```

```
for column in num_cols:
    if plotnumber <= 14:
        ax = plt.subplot(3, 5, plotnumber)
        sns.histplot(df[column], kde=True)
        plt.xlabel(column)
```

```
plotnumber += 1
```

```
plt.tight_layout()
plt.show()
```

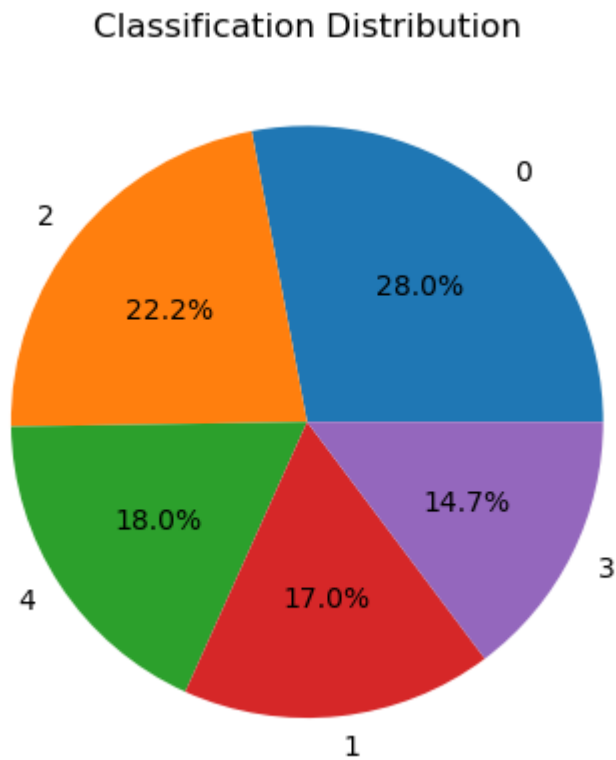


```
# Checking target class
```

```
plt.pie(df['CKD_stage'].value_counts(),labels=df['CKD_stage'].value_counts().index,
autopct='%1.1f%%')

plt.title('Classification Distribution')

plt.show()
```

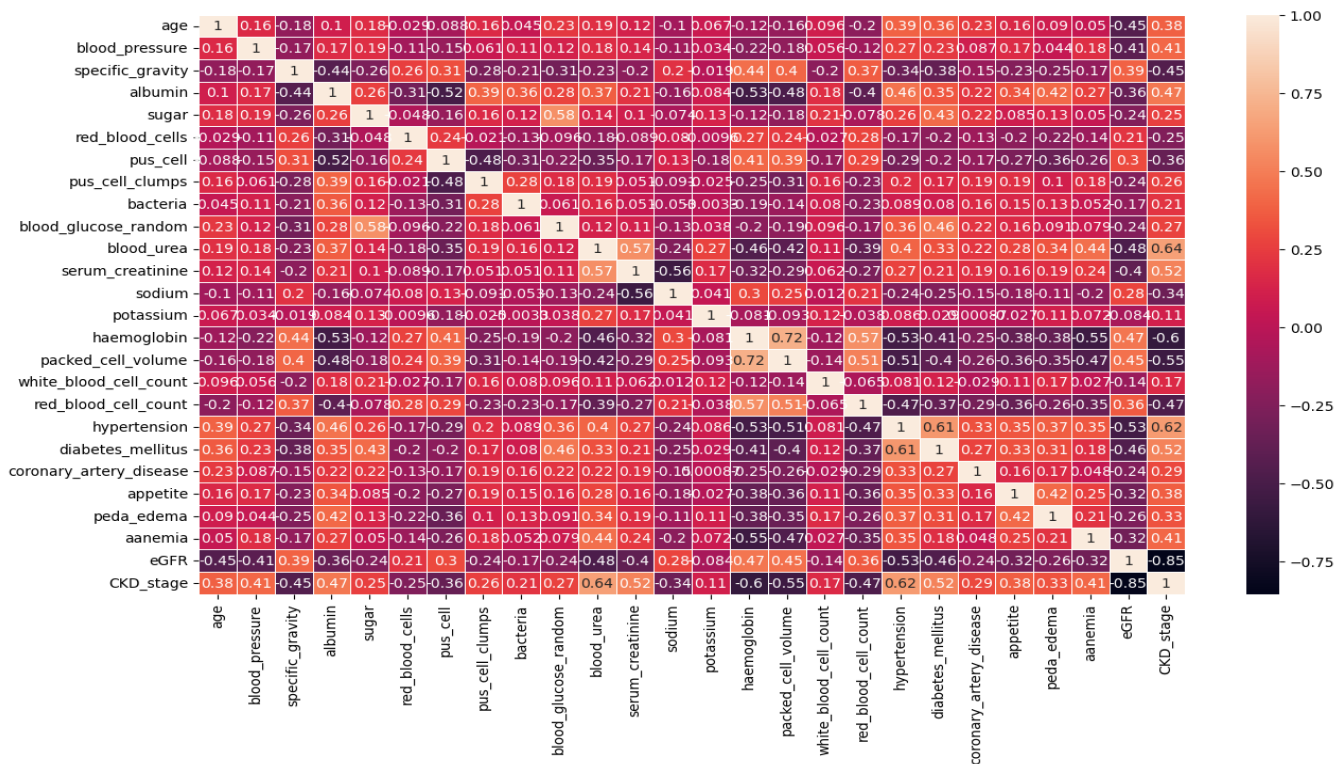


```
# heatmap of the data

plt.figure(figsize = (15, 8))

sns.heatmap(df.corr(), annot = True, linewidths = 0.5)

plt.show()
```



Creating a boxen plot to show features against the target

n_rows, n_cols = (5,3)

figure, axes = plt.subplots(nrows=n_rows, ncols=n_cols, figsize = (20, 10))

figure.suptitle('Numerical Features VS Target Variable')

for index, column in enumerate(num_cols):

i,j = (index // n_cols), (index % n_cols)

bp=sns.boxenplot(y=column, x='CKD_stage', data=df, ax=axes[i,j], showfliers=False)

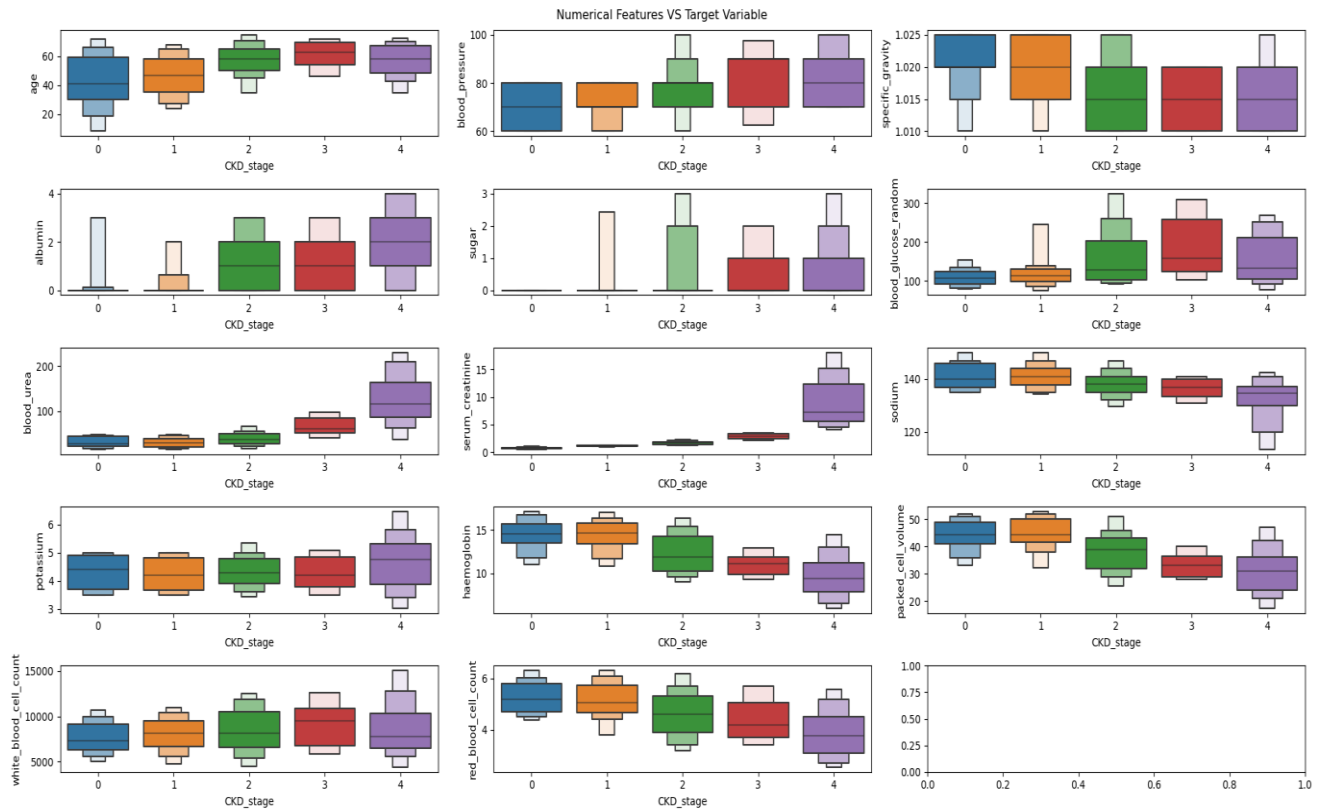
axes[i,j].set_xlabel(axes[i,j].get_xlabel())

axes[i,j].set_ylabel(column)

axes[i,j].set_xticklabels(axes[i,j].get_xticklabels())

plt.tight_layout()

plt.show()



5.2. Code for Algorithms

Logistic Regression

```
from scipy.stats import loguniform
```

```
from sklearn.linear_model import LogisticRegression
```

```
param_grid = dict()
```

```
param_grid['solver'] = ['newton-cg', 'lbfgs', 'liblinear']
```

```
param_grid['penalty'] = ['l2'] # 'none', 'l1', 'l2', 'elasticnet'
```

```
param_grid['C'] = loguniform.rvs(1e-5, 100, size=10)
```

```
grid = GridSearchCV(LogisticRegression(), param_grid, refit=True, verbose=1, cv=5)
```

```
model = grid.fit(X_train, y_train).best_estimator_
```

```
best_params = grid.best_params_
```

```
print(f"Best params: {best_params}")
```



```
y_train_prob = model.predict_proba(X_train)
```

```
y_test_prob = model.predict_proba(X_test)
```

```
print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob, train=True)
```

```
lr_acc, lr_ra = print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob,  
train=False)
```

Fitting 5 folds for each of 30 candidates, totalling 150 fits

Best params: {'C': 79.88515858220363, 'penalty': 'l2', 'solver': 'liblinear'}

Train Result:

=====

Accuracy Score: 98.12%

ROC AUC Score: 99.77%

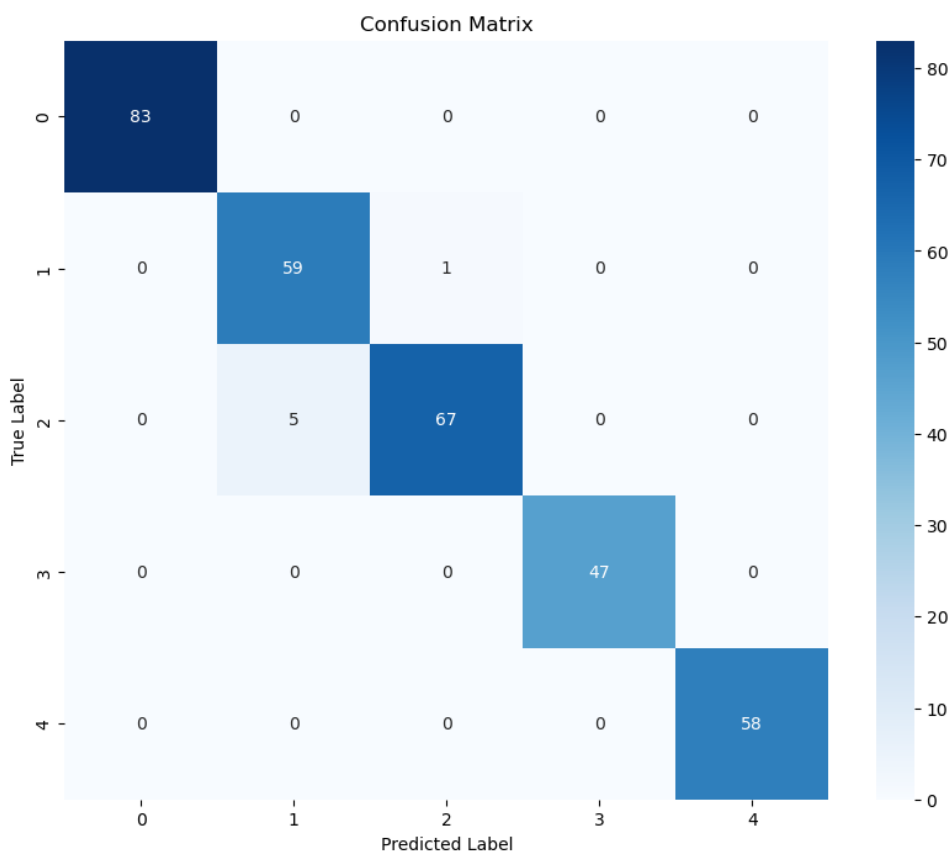
CLASSIFICATION REPORT:

	0	1	2	3	4	accuracy	macro avg \
precision	1.0	0.921875	0.985294	1.0	1.0	0.98125	0.981434
recall	1.0	0.983333	0.930556	1.0	1.0	0.98125	0.982778
f1-score	1.0	0.951613	0.957143	1.0	1.0	0.98125	0.981751
support	83.0	60.000000	72.000000	47.0	58.0	0.98125	320.000000

	weighted avg
precision	0.982043
recall	0.981250
f1-score	0.981285
support	320.000000

Confusion Matrix:

```
[[83  0  0  0  0]  
[ 0 59  1  0  0]  
[ 0  5 67  0  0]  
[ 0  0  0 47  0]  
[ 0  0  0  0 58]]
```



Test Result:

=====

Accuracy Score: 86.25%

ROC AUC Score: 99.77%

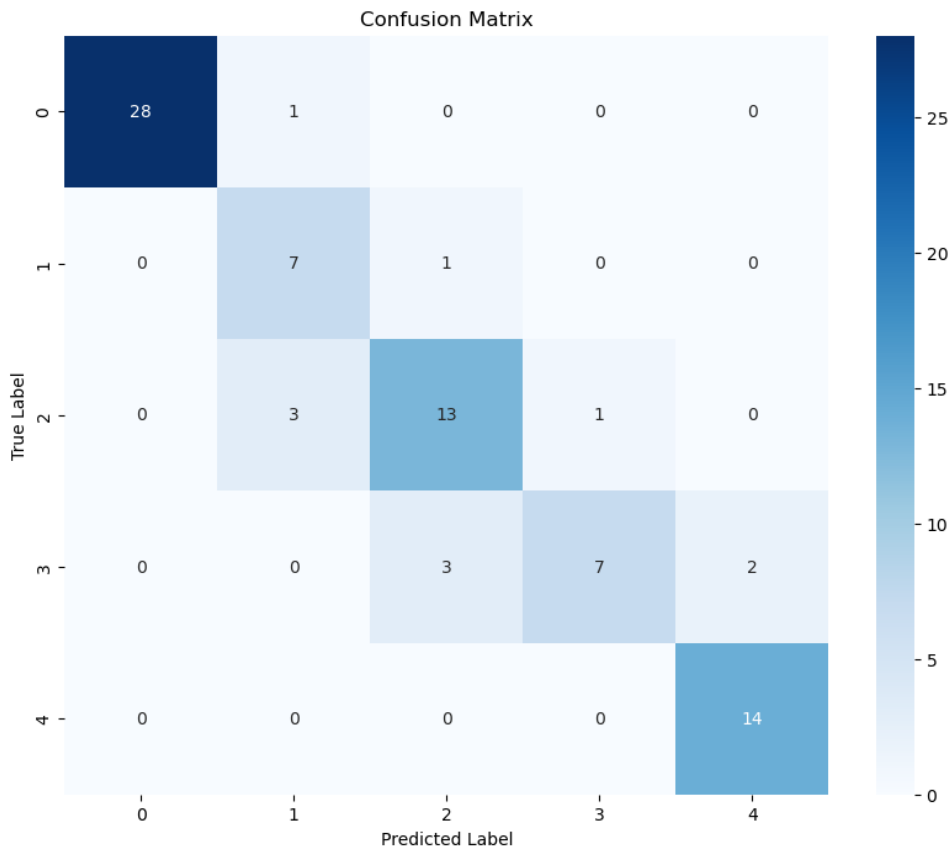
CLASSIFICATION REPORT:

	0	1	2	3	4	accuracy \
precision	1.000000	0.636364	0.764706	0.875000	0.875000	0.8625
recall	0.965517	0.875000	0.764706	0.583333	1.000000	0.8625
f1-score	0.982456	0.736842	0.764706	0.700000	0.933333	0.8625
support	29.000000	8.000000	17.000000	12.000000	14.000000	0.8625

	macro avg	weighted avg
precision	0.830214	0.873011
recall	0.837711	0.862500
f1-score	0.823467	0.860658
support	80.000000	80.000000

Confusion Matrix:

```
[[28  1  0  0  0]
 [ 0  7  1  0  0]
 [ 0  3 13  1  0]
 [ 0  0  3  7  2]
 [ 0  0  0  0 14]]
```



KNN

```
from sklearn.neighbors import KNeighborsClassifier
```

```
param_grid = {"n_neighbors": [i for i in range(1, 50, 10)],
              "weights": ["uniform", "distance"],
              "algorithm": ["ball_tree", "kd_tree", "brute"],
              "leaf_size": [i for i in range(1, 50, 10)],
              "p": [1,2]}
```

```
grid = GridSearchCV(KNeighborsClassifier(), param_grid, refit=True, verbose=1, cv=5)
model = grid.fit(X_train, y_train).best_estimator_
```

```
best_params = grid.best_params_
print(f"Best params: {best_params}")
```

```
y_train_prob = model.predict_proba(X_train)
y_test_prob = model.predict_proba(X_test)
```

```
print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob, train=True)
knn_acc, knn_ra = print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob,
                              train=False)
```

Fitting 5 folds for each of 300 candidates, totalling 1500 fits
Best params: {'algorithm': 'ball_tree', 'leaf_size': 1, 'n_neighbors': 11, 'p': 1, 'weights': 'distance'}
Train Result:

=====

Accuracy Score: 100.00%

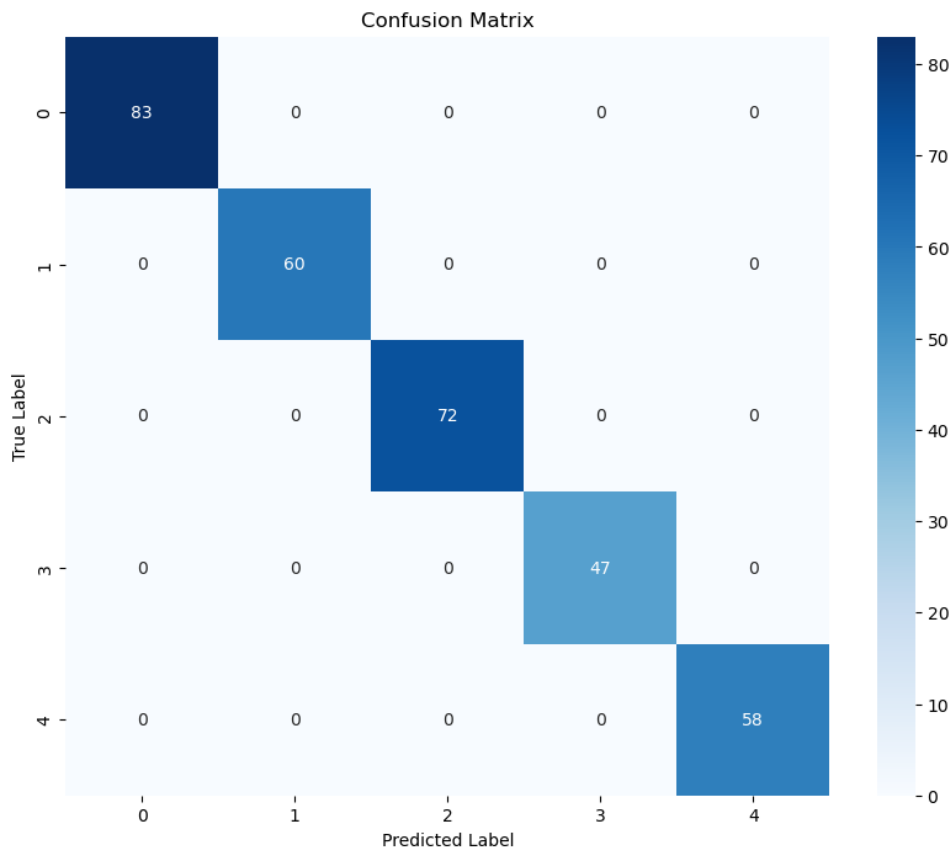
ROC AUC Score: 100.00%

CLASSIFICATION REPORT:

	0	1	2	3	4	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
support	83.0	60.0	72.0	47.0	58.0	1.0	320.0	320.0

Confusion Matrix:

```
[[83  0  0  0  0]
 [ 0 60  0  0  0]
 [ 0  0 72  0  0]
 [ 0  0  0 47  0]
 [ 0  0  0  0 58]]
```



Test Result:

=====

Accuracy Score: 68.75%

ROC AUC Score: 100.00%

CLASSIFICATION REPORT:

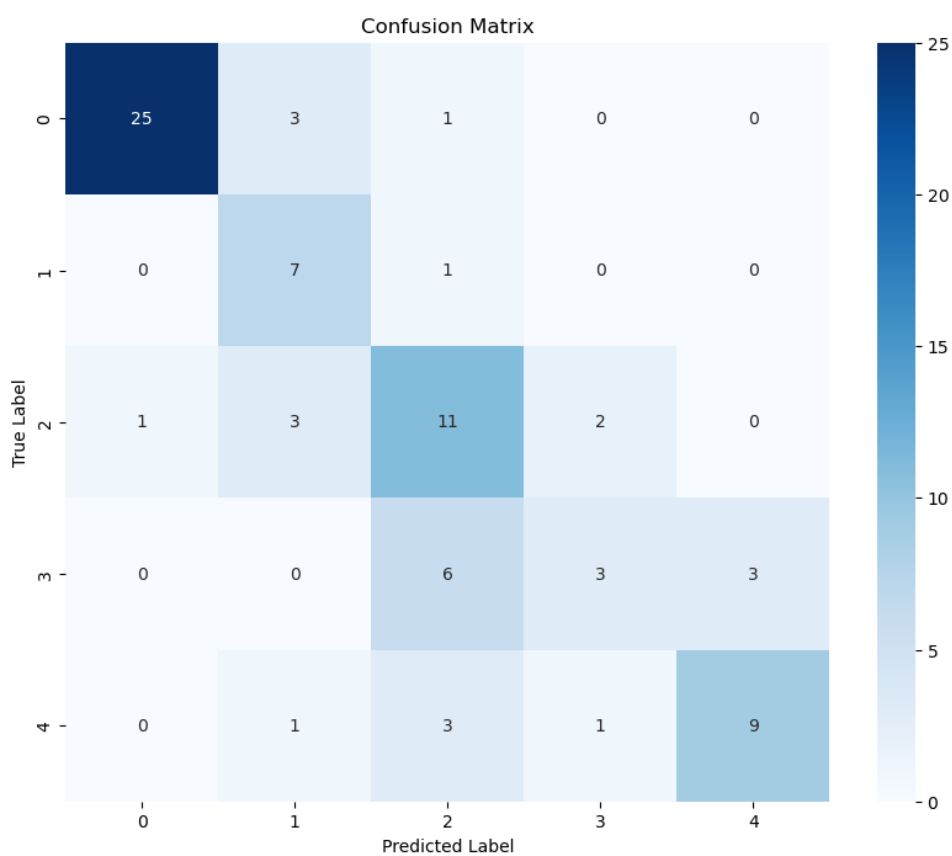
	0	1	2	3	4	accuracy	\
--	---	---	---	---	---	----------	---

precision	0.961538	0.500000	0.500000	0.500000	0.750000	0.6875
recall	0.862069	0.875000	0.647059	0.250000	0.642857	0.6875
f1-score	0.909091	0.636364	0.564103	0.333333	0.692308	0.6875
support	29.000000	8.000000	17.000000	12.000000	14.000000	0.6875

	macro avg	weighted avg
precision	0.642308	0.711058
recall	0.655397	0.687500
f1-score	0.627040	0.684207
support	80.000000	80.000000

Confusion Matrix:

```
[[25  3  1  0  0]
 [ 0  7  1  0  0]
 [ 1  3 11  2  0]
 [ 0  0  6  3  3]
 [ 0  1  3  1  9]]
```



Decision Tree

```
from sklearn import tree
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
param_grid = {"max_depth": [3, 5, 7, 10, 15, 20, None],
              "max_features": [None, 'sqrt', 'log2'],
              "min_samples_leaf": [1, 3, 5, 10, 20],
              "criterion": ["gini", "entropy"]}
```

```

grid = GridSearchCV(DecisionTreeClassifier(), param_grid, refit=True, verbose=1, cv=5)
model = grid.fit(X_train, y_train).best_estimator_

best_params = grid.best_params_
print(f"Best params: {best_params}")

y_train_prob = model.predict_proba(X_train)
y_test_prob = model.predict_proba(X_test)
dump(model, "../model/DT.joblib")
print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob, train=True)
dtc_acc, dtc_ra = print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob,
train=False)
plt.figure(figsize=(20, 18))
tree.plot_tree(model, feature_names = df.columns.tolist()[1:], filled=True, class_names=["0",
"1", "2", "3", "4"])
plt.show()

```

Fitting 5 folds for each of 210 candidates, totalling 1050 fits
Best params: {'criterion': 'gini', 'max_depth': 5, 'max_features': None, 'min_samples_leaf': 1}

Train Result:

=====
Accuracy Score: 100.00%

ROC AUC Score: 100.00%

CLASSIFICATION REPORT:

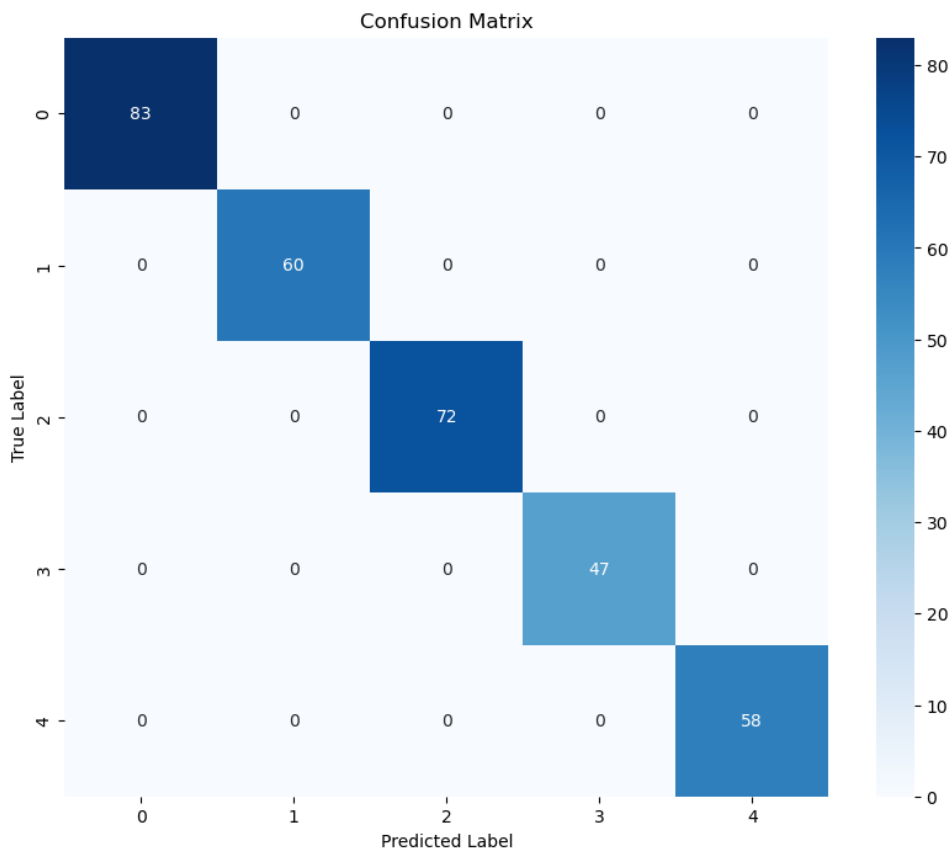
	0	1	2	3	4	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
support	83.0	60.0	72.0	47.0	58.0	1.0	320.0	320.0

Confusion Matrix:

```

[[83  0  0  0  0]
 [ 0 60  0  0  0]
 [ 0  0 72  0  0]
 [ 0  0  0 47  0]
 [ 0  0  0  0 58]]

```



Test Result:

=====

Accuracy Score: 100.00%

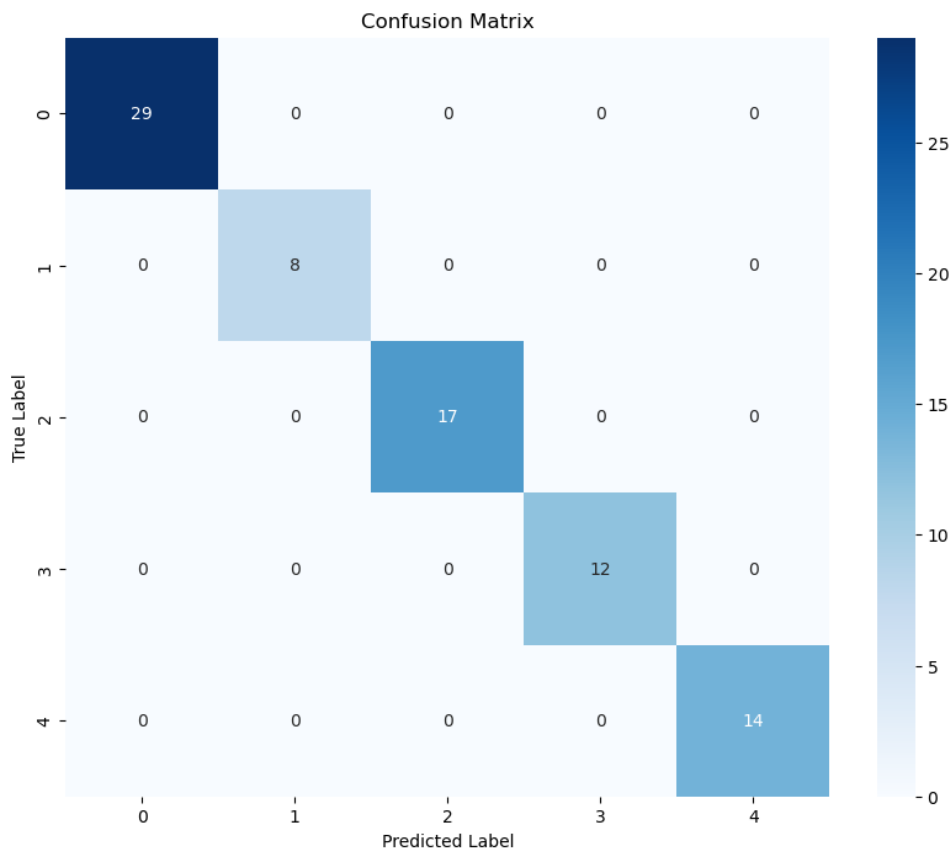
ROC AUC Score: 100.00%

CLASSIFICATION REPORT:

	0	1	2	3	4	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
support	29.0	8.0	17.0	12.0	14.0	1.0	80.0	80.0

Confusion Matrix:

```
[[29  0  0  0  0]
 [ 0  8  0  0  0]
 [ 0  0 17  0  0]
 [ 0  0  0 12  0]
 [ 0  0  0  0 14]]
```



Random Forest

```
from sklearn.ensemble import RandomForestClassifier
```

```
param_grid = {'bootstrap': [True, False],
              'max_depth': [None, 5, 10, 15, 20],
              'max_features': [None, 'sqrt'],
              'min_samples_leaf': [1, 10, 20],
              'min_samples_split': [2, 10, 20],
              'n_estimators': [50, 100]}
```

```
grid = GridSearchCV(RandomForestClassifier(), param_grid, refit=True, verbose=1, cv=5)
model = grid.fit(X_train, y_train).best_estimator_
```

```
best_params = grid.best_params_
print(f"Best params: {best_params}")
```

```
y_train_prob = model.predict_proba(X_train)
y_test_prob = model.predict_proba(X_test)
```

```
print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob, train=True)
```



```
rd_clf_acc, rd_clf_ra = print_score(model, X_train, y_train, X_test, y_test, y_train_prob, y_test_prob,
train=False)
```

```
Fitting 5 folds for each of 360 candidates, totalling 1800 fits
Best params: {'bootstrap': True, 'max_depth': None, 'max_features': None,
'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}
```

Train Result:

```
=====
Accuracy Score: 100.00%
```

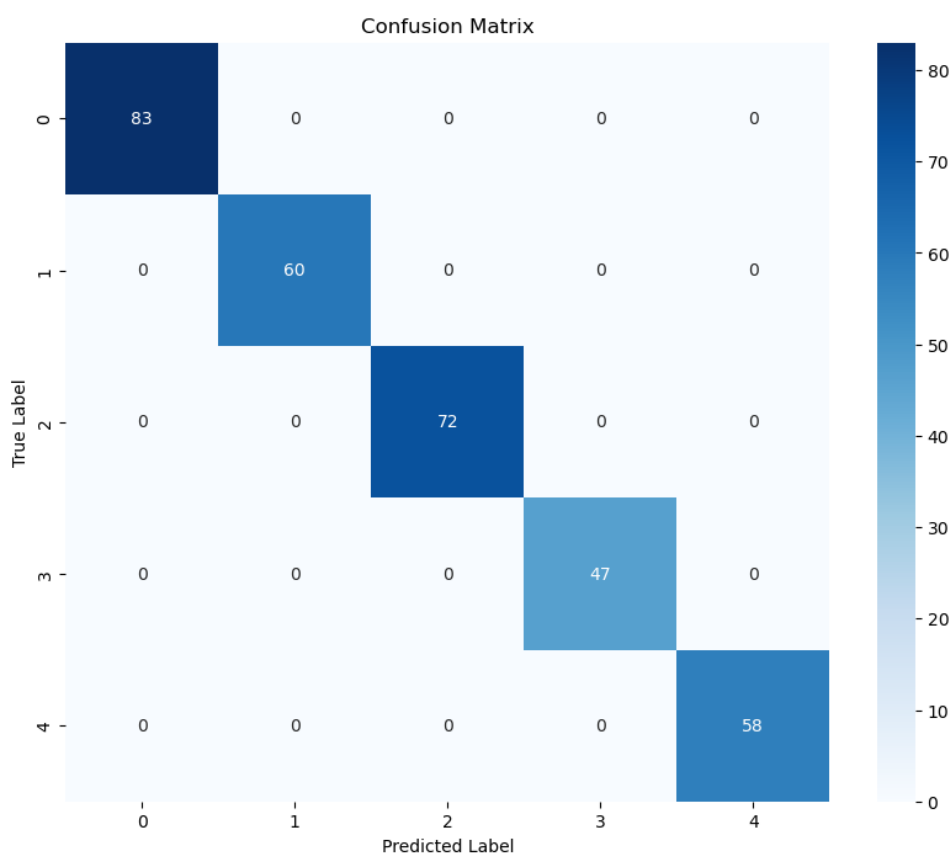
ROC AUC Score: 100.00%

CLASSIFICATION REPORT:

	0	1	2	3	4	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
support	83.0	60.0	72.0	47.0	58.0	1.0	320.0	320.0

Confusion Matrix:

```
[[83  0  0  0  0]
 [ 0 60  0  0  0]
 [ 0  0 72  0  0]
 [ 0  0  0 47  0]
 [ 0  0  0  0 58]]
```



Test Result:

```
=====
Accuracy Score: 100.00%
```

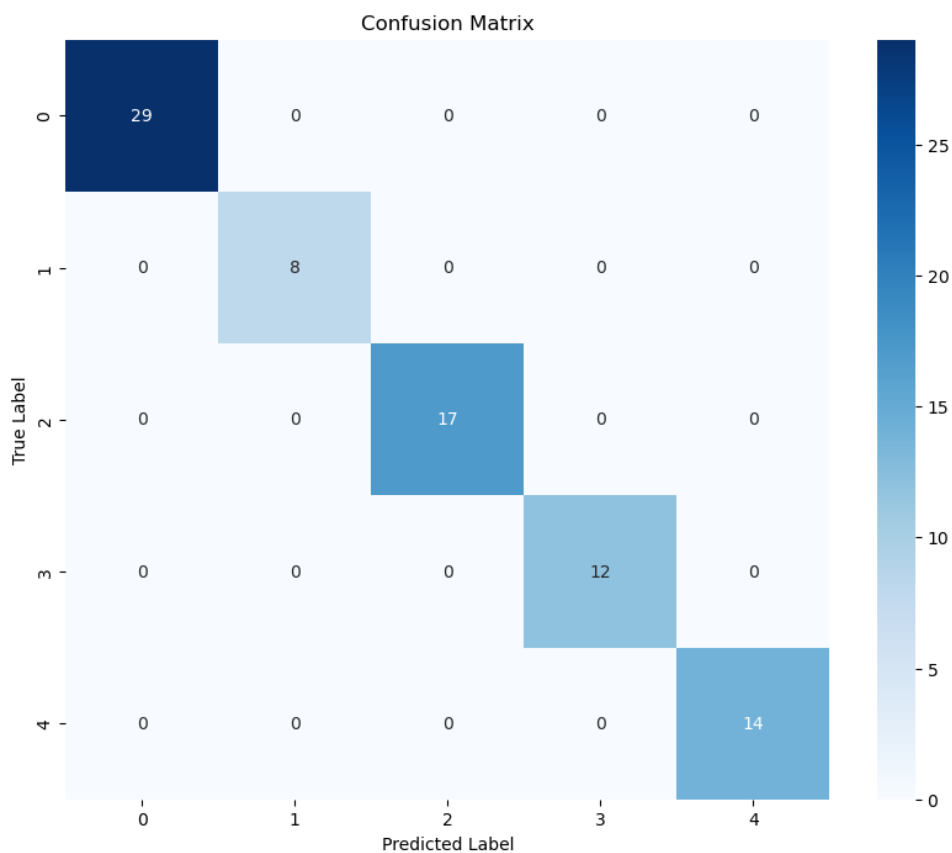
ROC AUC Score: 100.00%

CLASSIFICATION REPORT:

	0	1	2	3	4	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
support	29.0	8.0	17.0	12.0	14.0	1.0	80.0	80.0

Confusion Matrix:

```
[[29  0  0  0  0]
 [ 0  8  0  0  0]
 [ 0  0 17  0  0]
 [ 0  0  0 12  0]
 [ 0  0  0  0 14]]
```



Create a bar plot

```
plt.figure(figsize=(10, 6))
```

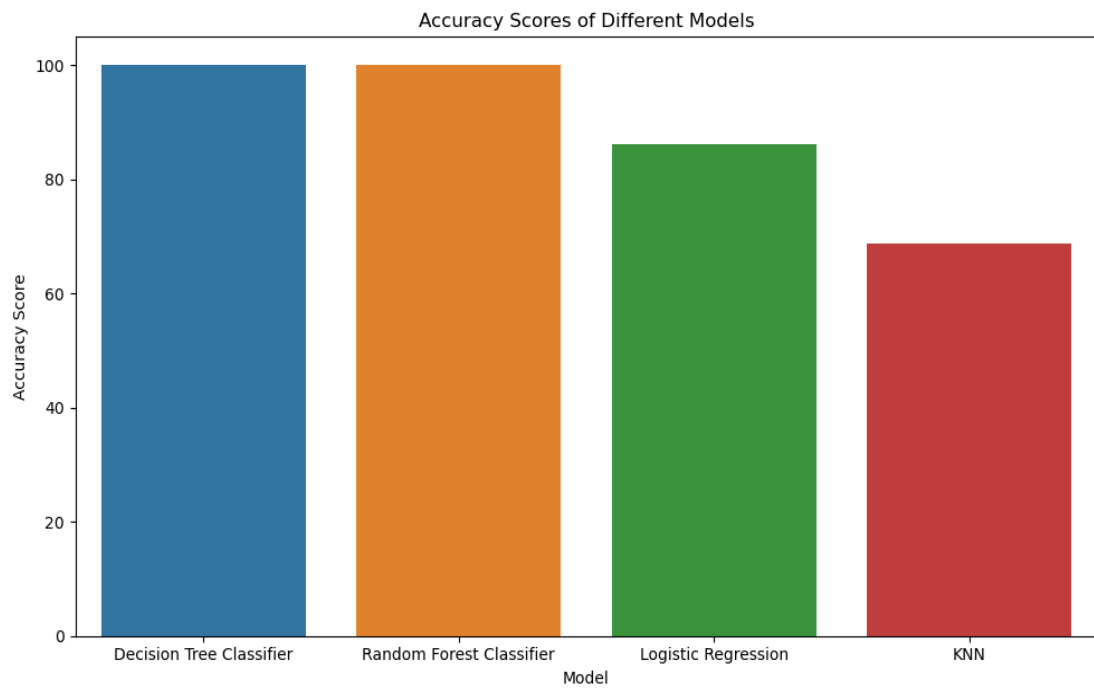
```
sns.barplot(data=models.sort_values(by = 'Accuracy Score', ascending = False), x='Model', y='Accuracy Score')
```

```
plt.title('Accuracy Scores of Different Models')
```

```
plt.ylabel('Accuracy Score')
```

```
plt.tight_layout()
```

```
plt.show()
```



Chapter 6

IMPLEMENTATION & RESULTS

6.1 Explanation of Key functions

Both Flask and Streamlit are popular frameworks used in web development, particularly for building interactive web applications. Flask, a lightweight Python web framework, empowers developers to build versatile web applications with flexibility and control. It's favored for its simplicity in getting started while providing the means to scale up to complex projects, making it suitable for a wide range of web development tasks, including building RESTful APIs and dynamic web pages. On the other hand, Streamlit shines as a Python library tailored specifically for data science and machine learning practitioners.

It streamlines the creation of interactive web applications by enabling developers to write simple Python scripts, abstracting away much of the complexity of web development. Streamlit is particularly valuable for rapidly prototyping and sharing data-centric applications, making it a preferred choice among data scientists and machine learning engineers for showcasing models and visualizations online. In essence, Flask and Streamlit cater to different needs within the web development ecosystem, with Flask offering general-purpose flexibility and Streamlit providing streamlined functionality for data-centric applications.

Flask serves as a versatile foundation for web applications, offering developers a robust set of tools and libraries while allowing for extensive customization. Its minimalistic design philosophy encourages developers to adopt best practices and design patterns suited to their project's unique requirements. Flask's modular architecture facilitates the integration of third-party extensions, enabling developers to extend functionality with ease, whether it be integrating authentication mechanisms, database management systems, or other advanced features. This extensibility makes Flask a popular choice for building diverse web solutions, from simple websites to complex enterprise applications, providing developers with the freedom to tailor their projects to specific needs.

In contrast, Streamlit simplifies the process of creating interactive web applications focused on data visualization and machine learning models. By abstracting away the complexities of web development, Streamlit enables data scientists and machine learning practitioners to quickly prototype and deploy web-based interfaces for their models and analyses. With Streamlit, developers can leverage familiar Python syntax to create interactive components such as sliders, buttons, and plots, allowing users to explore data and manipulate parameters in real-time.

This approach accelerates the development lifecycle, empowering practitioners to rapidly iterate on their ideas and share their findings with stakeholders and collaborators through intuitive web interfaces.

6.2 Method of Implementation

6.2.1 Forms

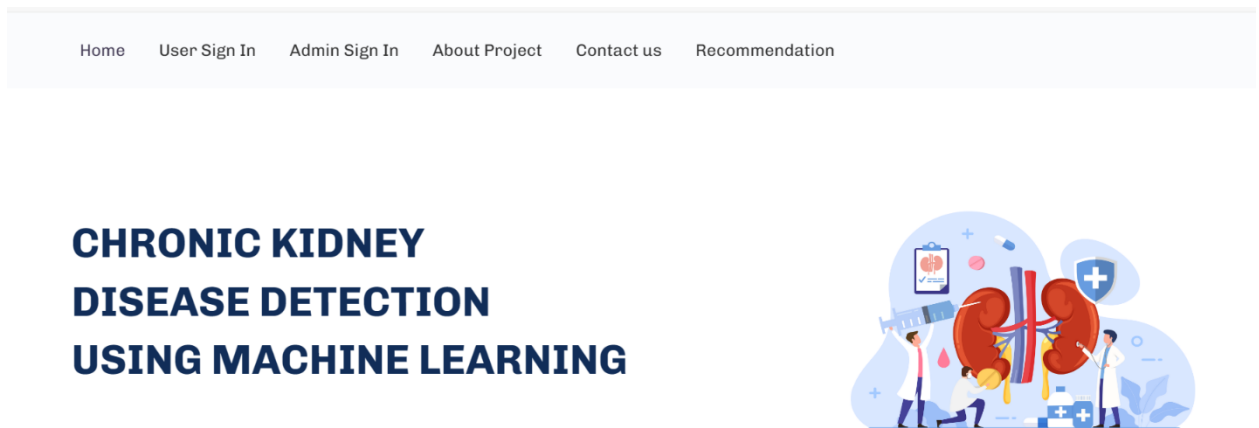


Fig 6.2.1.1:home page

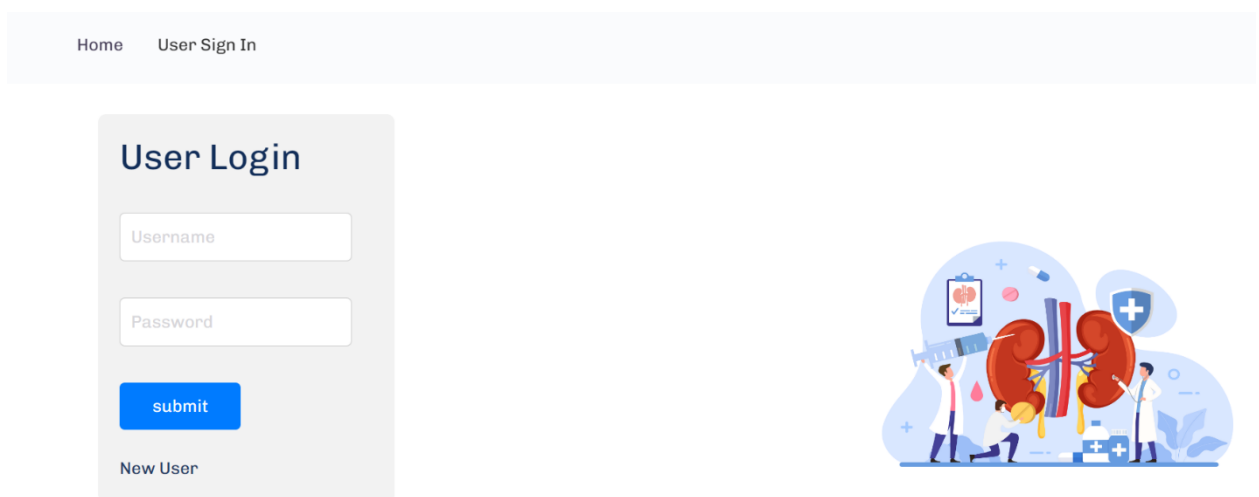


Fig 6.2.1.2 :User Sign In

Prediction Page SIGN OUT

Enter Prediction Values

Enter age in years:	<input type="text"/>	Enter blood pressure in mm/Hg:	<input type="text"/>
Enter Specific gravity in nominal:	<input type="text"/>	Enter albumin in nominal(1-5):	<input type="text"/>
Enter sugar in nominal:	<input type="text"/>	Enter rbc(normal,abnormal):	<input type="text" value="normal"/>
Enter pus cell(normal,abnormal):	<input type="text" value="normal"/>	Enter pus cell clumps(present,not present):	<input type="text" value="present"/>
Enter bacteria(present,not present):	<input type="text" value="present"/>	Enter blood glucose rando:	<input type="text"/>
Enter blood urea:	<input type="text"/>	Enter serum creatinine	<input type="text"/>

Fig 6.2.1.3 :Disease Detection

Home User Sign In Admin Sign In

Admin Login

submit




Fig 6.2.1.4 :Admin Sign In

Users Details

Id	Name	Email	Mobile no:	Status	Active
1	chethana	chethana855855@gmail.com	7675998105	Activated	Activated
2	suresh	suresh@gmail.com	5555555555	Activated	Activated
3	mdk	di@gmail.com	1234567890	Activated	Activated
4	qwe	di@gmail.com	1234567890	Activated	Activated
5	qwer	div@gmail.com	1234567890	Activated	Activated
6	qasd	zxc@gmail.com	1234567890	Activated	Activated
7	san	san@gmail.com	7777777777	Activated	Activated
8	divesh123	div@gmail.com	8978309554	Activated	Activated

Fig 6.2.1.5 :Admin Accessing User Details

BACK

Decision Tree

In this project we are using Image Preprocessing by using Computer Vision.



Preprocessing

1. Fixing typos
2. Encoding methods
3. Imputations



Train & Test Split

1. Train: 80%
2. Test: 20%



Algorithm

1. Accuracy: 100%

Fig 6.2.1.6: About Project

BACK

Who Developed This Project

Student Name	Student ID
P.Arvind	20K91A6734
P.Laxmi Priya	20K91A6731
M.Sai	20K91A6726
K.Sai Ram	21K95A6704

College Name :

TKR College Of Engineering & Technology

Under the Guidance of :

MRS.K.KIRANMAYEE



Fig 6.2.1.7: Contact Us

6.2.2 Output Screens

Prediction Page SIGN OUT

MODEL PREDICTION PAGE

Model Predicted Patient Report:

Stage: 4

1. **Stage Type:** Severe CKD
2. **Description:** CKD Stage 4 shows severe reduction in kidney function, leading to significant accumulation of fluids and waste products in the body.
3. **Diagnosis:** Regular, detailed assessments to plan for potential kidney replacement therapy (dialysis or transplant).
4. **Scientific Drug:** Comprehensive treatment including erythropoiesis-stimulating agents for anemia and medications to manage other complications.
5. **Food Precautions:** Strict dietary restrictions to manage potassium, phosphorus, fluids, and protein intake; often requires the guidance of a dietitian.

Fig 6.2.2.1 :Model Prediction page

Model Performance

Evaluation Details

```

Test Result:
=====
Accuracy Score: 100.00%

ROC AUC Score: 100.00%

CLASSIFICATION REPORT:
      0      1      2      3      4  accuracy  macro avg  weighted avg
precision 1.0  1.0  1.0  1.0  1.0      1.0      1.0      1.0
recall    1.0  1.0  1.0  1.0  1.0      1.0      1.0      1.0
f1-score   1.0  1.0  1.0  1.0  1.0      1.0      1.0      1.0
support   28.0  8.0 18.0 13.0 13.0      1.0      80.0      80.0

Confusion Matrix:
[[28  0  0  0  0]
 [ 0  8  0  0  0]
 [ 0  0 18  0  0]
 [ 0  0  0 13  0]
 [ 0  0  0  0 13]]

```

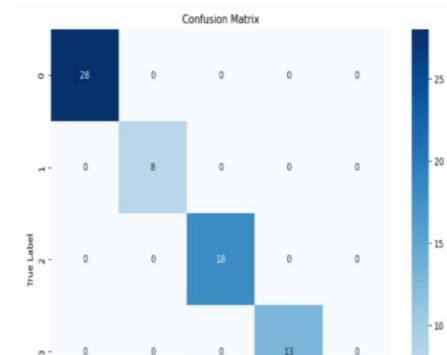


Fig 6.2.2.2 :Model Performance

6.2.3 Result Analysis

Logistic Regression

Train Result:

Accuracy Score: 98.12%

ROC AUC Score: 99.77%

Test Result:

Accuracy Score: 86.25%

ROC AUC Score: 99.77%

KNN

Train:

Accuracy Score: 100.00%

ROC AUC Score: 100.00%

Test Result:

Accuracy Score: 68.75%

ROC AUC Score: 100.00%

Decision Tree

Train Result:

Accuracy Score: 100.00%

ROC AUC Score: 100.00%

Test Result:

Accuracy Score: 100.00%

ROC AUC Score: 100.00%

Random Forest

Train Result:

Accuracy Score: 100.00%

ROC AUC Score: 100.00%

Test Result:

Accuracy Score: 100.00%

ROC AUC Score: 100.00%

Chapter 7

TESTING & VALIDATION

7.1 Design of Test Cases and scenarios

Designing test cases and scenarios for chronic kidney disease detection using machine learning projects involves ensuring that the system functions accurately, efficiently, and reliably. Here's a structured approach to designing test cases and scenarios:

Data Preprocessing:

Test Case 1: Verify that missing values in the dataset are handled appropriately (e.g., imputation, deletion).

Test Case 2: Ensure that categorical variables are encoded properly (e.g., one-hot encoding, label encoding).

Test Case 3: Validate that data scaling or normalization is applied correctly to numerical features.

Feature Engineering:

Test Case 4: Confirm that feature selection techniques (e.g., correlation analysis, feature importance) are applied accurately.

Test Case 5: Validate the creation of new features from existing ones (e.g., feature transformations, interaction terms).

Model Training:

Test Case 6: Check that the appropriate machine learning algorithms are selected based on the problem (e.g., regression, classification).

Test Case 7: Ensure that hyperparameters tuning is performed correctly (e.g., using cross-validation, grid search).

Test Case 8: Validate the splitting of data into training and testing sets and that it is done randomly and consistently.

Model Evaluation:

Test Case 9: Verify the accuracy of the model's predictions against a baseline (e.g., simple heuristics).

Test Case 10: Validate model performance metrics (e.g., accuracy, precision, recall, F1-score).

Test Case 11: Ensure that the model's generalization ability is tested with unseen data (e.g., cross-validation, holdout set).

Deployment and Integration:

Test Case 12: Confirm that the model integration with the airline's data infrastructure is successful.

Test Case 13: Validate the responsiveness of the system when handling real-time data.

Test Case 14: Ensure that the deployed model's predictions align with the business requirements and expectations.

Robustness and Edge Cases:

Test Case 15: Test the model's robustness against outliers and noisy data.

Test Case 16: Validate the model's behavior under extreme conditions

Test Case 17: Check for potential biases in the model predictions

Security and Privacy:

Test Case 18: Ensure that sensitive data is handled securely and anonymized where necessary.

Test Case 19: Validate compliance with data protection regulations.

Performance Testing:

Test Case 20: Measure the computational resources (e.g., memory, processing time) required for model training and prediction.

Test Case 21: Test the scalability of the system to handle large volumes of data efficiently.

User Acceptance Testing:

Test Case 22: Engage stakeholders to validate that the system meets their requirements and expectations.

Test Case 23: Solicit feedback from end-users to identify areas for improvement and enhancement.

Documentation and Maintenance:

Test Case 24: Ensure that comprehensive documentation is provided for the system, including model architecture, data sources, and usage instructions.

Test Case 25: Validate that the system is maintainable and easily upgradable with future enhancements or bug fixes.

By following this structured approach and tailoring it to the specific requirements of your kidney disease project, you can ensure the reliability, accuracy, and efficiency of your machine learning solution.

7.2 Validation

Validating machine learning models for chronic kidney disease detection involves several key steps to ensure the reliability and accuracy of the models. Here's a generalized process you can follow:

Data Collection and Preprocessing:

Gather relevant data from various sources such as disease detection, stage of disease, drug recommendation, etc. Preprocess the data to handle missing values, outliers, and inconsistencies. This may involve techniques like data imputation, normalization, and feature engineering.

Exploratory Data Analysis (EDA):

Conduct EDA to understand the characteristics and patterns in the data. Identify correlations between different variables and their potential impact on flight operations.

Feature Selection:

Select the most relevant features that contribute to the predictive power of the model. Use techniques like correlation analysis, feature importance, or domain expertise to guide feature selection.

Model Selection and Training:

Choose appropriate machine learning algorithms suitable for the problem at hand (e.g., regression for predicting disease, classification for stage of disease). Split the data into training, validation, and test sets. Train the model on the training set and tune hyperparameters using the validation set.

Evaluation Metrics:

Select appropriate evaluation metrics based on the specific problem. For example, accuracy, precision, recall, F1-score for classification tasks, Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) for regression tasks. Ensure the chosen metrics align with business objectives and requirements.

Cross-Validation:

Perform cross-validation to assess the generalization performance of the model. Techniques such as k-fold cross-validation can help provide a more robust estimate of model performance.

Model Performance Testing:

Evaluate the model's performance on the test set, which serves as an independent dataset not used during training or validation. Assess if the model's performance meets the desired criteria and business requirements.

Model Interpretability and Transparency:

Ensure the model's decisions are interpretable and transparent, especially in critical applications like airline operations. Techniques such as feature importance analysis, SHAP values, or model-specific interpretation methods can help explain the model's predictions

Deployment and Monitoring:

Deploy the validated model into production systems. Implement monitoring mechanisms to track the model's performance over time and detect any drift or degradation in performance.

Feedback Loop:

Establish a feedback loop to continuously improve the model based on new data and insights gained from deployment. By following these steps, you can validate machine learning models effectively for chronic kidney disease detection, ensuring their reliability and usefulness in real-world applications.

7.3 Conclusion

In conclusion, the development of a Chronic Kidney Disease (CKD) detection system using machine learning represents a pivotal stride towards advancing early diagnosis and personalized healthcare. The comprehensive approach outlined encompasses critical stages from data collection and preprocessing to the integration of advanced technologies such as explainable AI, real-time monitoring, and dynamic updates. By leveraging diverse datasets, integrating ethical considerations, and emphasizing user-friendly interfaces, the system is designed to meet the complex demands of healthcare professionals and contribute to improved patient outcomes.

The literature review underscores the evolving landscape of CKD detection, highlighting recent advancements in personalized risk assessment, multimodal data integration, ethical considerations, and longitudinal monitoring. Insights from diverse perspectives, including patient engagement and the integration of Electronic Health Records (EHR), offer a nuanced understanding of the challenges and opportunities in the field. These findings not only inform the proposed system but also emphasize the need for a multidimensional and adaptable approach to CKD detection that considers the dynamic nature of healthcare.

The identified functional and non-functional requirements provide a comprehensive framework for the CKD detection system's development, ensuring accuracy, usability, and ethical considerations are embedded in its core. The emphasis on continuous validation, iterative improvement, and bias mitigation strategies aligns the system with the highest standards of reliability and fairness. The technologies and tools highlighted, ranging from machine learning frameworks to containerization and orchestration, reflect a contemporary and robust technological stack essential for the system's successful implementation.

In summary, the proposed CKD detection system emerges as a sophisticated and holistic solution to the challenges in early diagnosis. By integrating cutting-edge technologies, ethical considerations, and user-centric design principles, the system holds the promise of not only enhancing diagnostic accuracy but also fostering a seamless integration into the complex and dynamic landscape of healthcare. The outlined approach, requirements, and technologies collectively contribute to a roadmap for developing an impactful CKD detection system that stands at the intersection of innovation and patient-centric care.

Chapter 8

CONCLUSION

In conclusion, the development and implementation of an advanced airline data analytics system represent a transformative leap in the aviation industry, where operational efficiency, pricing strategies, and customer satisfaction are paramount. The integration of sophisticated algorithms, cutting-edge technologies, and strategic tools converge to create a comprehensive solution that addresses the multifaceted challenges faced by airlines.

The predictive prowess of machine learning algorithms, such as Random Forests, Gradient Boosting, and Recurrent Neural Networks, empowers airlines to anticipate and mitigate flight delays, optimizing crew schedules and enhancing overall punctuality. In parallel, regression models, time series analysis, and ensemble models contribute to optimal fare price estimation, allowing airlines to dynamically adjust ticket prices based on market dynamics, competitor strategies, and real-time demand.

The customer-centric approach is bolstered by sentiment analysis algorithms, decision trees for feedback mining, and collaborative filtering techniques. These enable airlines to decipher passenger preferences, promptly address concerns, and enhance overall satisfaction by tailoring services to individual needs.

In essence, the advanced airline data analytics system embodies a holistic solution that not only revolutionizes how airlines manage their operations but also elevates the passenger experience. The real-time insights derived from data analytics empower airlines to make informed decisions, adapt to market dynamics, and cultivate lasting customer loyalty. As the aviation industry evolves, this integrated framework stands as a testament to the potential of data-driven strategies in shaping the future of air travel.

REFERENCES

1. A. N. Muiru et al., "The epidemiology of chronic kidney disease (CKD) in rural east Africa: A population-based study", *PLoS ONE*, vol. 15, no. 3, Mar. 2020.
2. C. P. Wen et al., "All-cause mortality attributable to chronic kidney disease: A prospective cohort study based on 462 293 adults in Taiwan", *Lancet*, vol. 371, no. 9631, pp. 2173-2182, Jun. 2008.
3. M. A. Hossain, T. A. Asa, M. R. Rahman and M. A. Moni, "Network-based approach to identify key candidate genes and pathways shared by thyroid cancer and chronic kidney disease", *Informat. Med. Unlocked*, vol. 16, Jan. 2019.
4. K. Brück et al., "CKD prevalence varies across the European general population", *J. Amer. Soc. Nephrol.*, vol. 27, no. 7, pp. 2135-2147, 2016.
5. A. S. Allen, J. P. Forman, E. J. Orav, D. W. Bates, B. M. Denker and T. D. Sequist, "Primary care management of chronic kidney disease", *J. Gen. Internal Med.*, vol. 26, no. 4, pp. 386-392, 2011.
6. G. Remuzzi, P. Ruggenenti and N. Perico, "Chronic renal diseases: Renoprotective benefits of renin–angiotensin system inhibition", *Ann. Internal Med.*, vol. 136, no. 8, pp. 604-615, 2002.
7. M. A. Hossain, T. A. Asa, S. M. S. Islam, M. S. Hussain and M. A. Moni, "Identification of genetic association of thyroid cancer with parkinsons disease osteoporosis chronic heart failure chronic kidney disease type 1 diabetes and type 2 diabetes", *Proc. 5th Int. Conf. Adv. Electr. Eng. (ICAEE)*, pp. 832-837, Sep. 2019.
8. O. J. Wouters, D. J. O'donoghue, J. Ritchie, P. G. Kanavos and A. S. Narva, "Early chronic kidney disease: Diagnosis management and models of care", *Nature Rev. Nephrol.*, vol. 11, no. 8, pp. 491, 2015.
9. K.-U. Eckardt et al., "Autosomal dominant tubulointerstitial kidney disease: Diagnosis classification and management—A KDIGO consensus report", *Kidney Int.*, vol. 88, no. 4, pp. 676-683, 2015.
10. T. Fiseha, M. Kassim and T. Yemane, "Chronic kidney disease and underdiagnosis of renal insufficiency among diabetic patients attending a hospital in southern ethiopia", *BMC Nephrol.*, vol. 15, no. 1, pp. 198, Dec. 2014.